6. Архитектура и технологии построения сетей TCP/IP

**TCP/IP** – это промышленный стандарт стека протоколов, используемый в глобальных и в локальных сетях. Аббревиатура TCP/IP включает обозначение двух базовых протоколов, на основе которых строится сеть: **TCP** (***Transmission Control Protocol***) – ***протокол управления передачей*** и **IP** (***Internet Protocol***) – ***межсетевой протокол***. В целом же стек протоколов TCP/IP включает протоколы четырех уровней (прикладного, транспортного, сетевого и уровня доступа или уровня сетевых интерфейсов). Стек TCP/IP использует в качестве транспортной среды между узлами коммутации (шлюзами/маршрутизаторами) другие сети или выделенные каналы. Стек TCP/IP является самым популярным средством организации ассоциативных (объединенных, составных) сетей.

**6.1. КОНЦЕПЦИЯ ТСР/IP**

Хост А

**IP**-уровень межсетев. взаимодей-я

Уровень сетевых интерфейсов

**ТСР** (Т-уровень)

Хост В

**ТСР** (Т-уровень)

**IP**

Уровень сетевых интерфейсов

IP

Уровень

Сетевых интерф

IP

Уровень

Сетевых

интерф

Порт (SAP)

TCP-соединение (лог. канал)

МJ

МF

**Рис. 6.1 Концепция TCP/IP (Transmission Control Protocol)**

Основными элементами сети TCP/IP являются оконечные устройства (компьютеры) и узлы коммутации, а также связывающие их между собой каналы физической среды передачи.

Узлы коммутации представляют собой маршрутизаторы IP-пакетов, а в роли физической среды обычно выступают различные сети, построенные на других сетевых технологиях (LAN или WAN). В протокольные блоки данных (ПБД) таких сетей от уровня звена данных и выше инкапсулируются IP-пакеты для передачи от одного маршрутизатора к другому (или от оконечного устройства к ближайшему маршрутизатору). Для соединения маршрутизаторов между собой и с оконечными устройствами используются также выделенные (или коммутируемые) цифровые каналы в сочетании с вспомогательными протоколами пакетной передачи данных типа «точка-точка», в ПБД которых можно инкапсулировать IP-пакеты.

В сети Интернет различают не только отдельные сети, но и более крупные объединения – автономные системы. ***Автономная система*** – это совокупность сетей под единым административным управлением, обеспечивающим общую для всех входящих в автономную систему маршрутизаторов политику маршрутизации.

Сегодня Интернет представляет собой объединение равноправных автономных систем с произвольной топологией связей.

Вся эта конструкция «склеивается» благодаря протоколу сетевого уровня IР (Internet Protocol — протокол сети Интернет). Его работа заключается в транспортировке дейтаграмм от отправителя к получателю независимо от того, находятся эти машины в одной и той же сети или нет.

Соединение в сети Интернет включает следующие процедуры. Транспортный уровень берет сегмент из потока данных и разбивает его на дейтаграммы (рис.6.2).

Заголов **ДАННЫЕ ≤65535Б**

**≥20 Байт**

ПОТОК ДАННЫХ ПОЛЬЗОВАТЕЛЯ

ПОТОКПП

Сегмент ТСР

Дейтаграмма UDP

Заголов

**≥20 Байт**

IP – пакет

Фрагменты IP- пакетов

1

2

3

1 2 3

1

**Рис. 6.2 Протокольные модули уровней модели TCP/IP**

Теоретически раз­мер каждой дейтаграммы может достигать 64 Кбайт, однако на практике они обычно не более 1500 байт (укладываются в один кадр Ethernet). Каждая дейтаграмма пересылается по Интернету, возможно, разбиваясь при этом на более мел­кие фрагменты, собираемые сетевым уровнем получателя в исходную дейтаграм­му. Затем эта дейтаграмма передается транспортному уровню, вставляющему ее во входной поток получающего процесса. На пути следования дейтограммы может оказаться несколько промежуточных сетей.

Протокол **TCP** отслеживает передаваемые блоки данных. У каждого хоста должен быть уникальный глобальный адрес (межсетевой). У каждого процесса хоста должен быть уникальный адрес в пределах данного хоста. Эти адреса называют портами. Порт – это идентификатор соединения процесса с каналом связи. Совокупность № сети, № узла и № порта в узле называется **гнездом**. Такая адресация позволяет Т-протоколу (ТСР) доставить данные нужному процессу.

Хост А передает сообщение протоколу ТСР с указанием переслать его хосту В (рис. 6.1).

Протокол TCP ориентирован на соединение. Он использует для транспортировки IP-дейтограммы (L3). Дейтаграммы пересылаются посредством протокольных кадров уровня L2.

Между двумя партнерами может быть прямое соединение, а может располагаться большое число сетевых приборов уровней L2 и L3.

Протоколу IP не нужно знать адрес (номер) порта В. Протокол IP передает сообщение уровню доступа к сети 1 (напр. логическому уровню Ethernet) с указанием переслать его маршрутизатору **J.** Управляющая информация передается вместе с данными в заголовках протокольного блока.

Заголовок сегмента (ТСР) содержит номер порта приемника, номер сегмента, контрольную сумму.

Заголовок дейтаграммы (IP) хранит адрес хоста получателя.

Заголовок уровня доступа содержит информацию, необходимую для передачи пакета по подсети (в данном примере адрес маршрутизатора **J**, запрос характеристик).

На маршрутизаторе **J** заголовок пакета удаляется, изучается заголовок IP. IP-модуль маршрутизатора направляет дейтаграмму по подсетям 2, 3 и маршрутизатору **F**  хосту В. Для этого в маршрутизаторах к дейтаграмме добавляется заголовок доступа к сети. На хосте В происходит обратный процесс.

Блок данных на каждом протокольном уровне называют *протокольным модулем данных* (Protocol Data Unit? PDU).

**6.2. СТЕК ПРОТОКОЛОВ TCP/IP**

Стек протоколов TCP/IP включает транспортные протоколы TCP/IP; протоколы сбора маршрутной информации RIP OSPF; протоколы управления сетью ICMP и ARP; прикладные протоколы HTTP, FTP, Telnet, SNMP, SMTP, TFTP, NSP (рис. 6.3).



**Рис.6.3*.* Архитектура сети TCP/IP.**

Рассмотрим более подробно основные протоколы стека TCP/IP на каждом из четырех уровней: прикладном, транспортном, сетевом и уровне доступа.

**6.2.1. Уровень доступа (уровень сетевых интерфейсов)**

Самый нижний уровень стека TCP/IP соответствует уровню звена данных ЭМВОС. Этот уровень в протоколах TCP/IP не регламентируется, но поддерживает все популярные стандарты уровня звена данных как для локальных так и для глобальных сетей. Разработана и специальная спецификация, определяющая использование технологии ATM в качестве транспорта IP-пакетов.

Уровень доступа к сетям (к среде передачи) формирует кадры (фрагменты пакетов), имеющие формат, структура которого зависит от характеристик используемых сетей передачи данных или выделенных каналов. Основное назначение уровня доступа к сетям (сетевого интерфейса) – обеспечение независимости функционирования протоколов TCP/IP от среды передачи.

При использовании выделенных (или коммутируемых) цифровых каналов для соединения маршрутизаторов между собой и с оконечными устройствами применяются вспомогательные протоколы пакетной передачи данных типа «точка-точка», в ПБД которых можно инкапсулировать IP-пакеты. Наиболее популярным протоколом такого типа является протокол PPP.

**PPP** (***Point-to-Point Protocol***) – протокол «точка-точка». При разработке протокола PPP за основу был взят формат кадров HDLC и дополнен собственными полями. Поля протокола PPP вложены в поле данных кадра HDLC, в кадры frame relay и других протоколов глобальных сетей.

Протокол РРР добивается согласованной работы различных устройств с помощью переговорной процедуры, во время которой передаются различные параметры, такие как качество линии, протокол аутентификации и инкапсулируемые протоколы сетевого уровня. Переговорная процедура происходит во время установления соединения.

Протокол РРР основан на четырех принципах: переговорное принятие параметров соединения, многопротокольная поддержка, расширяемость протокола, независимость от глобальных служб.

Одним из важных параметров РРР-соединения является *режим аутентификации*. Для целей аутентификации РРР предлагает по умолчанию протокол **РАР** (Password Authentication Protocol), передающий пароль по линии связи в открытом виде, или протокол **CHAP** (Challenge Handshake Authentication Protocol), не передающий пароль по линии связи и поэтому обеспечивающий большую безопасность сети.

Протокол РРР работает со многими протоколами сетевого уровня (IP, Novell IPX, AppleTalk, DECnet, XNS, Banyan VINES и OSI), и протоколами уровня звена данных LAN.

В РРР используется только один тип кадра HDLC – ненумерованный информационный. Для исправления очень редких ошибок, возникающих в канале, необходимы протоколы верхних уровней – TCP, SPX, NetBUEl, NCP и т.п.

**6.2.2. Сетевой уровень**

Сетевой уровень (чаще называемый «***межсетевым***» от слова ***internet*** или «*уровнем межсетевого взаимодействия*») является стержнем всей архитектуры TCP/IP. Именно он обеспечивает перемещение пакетов от одних оконечных устройств (граничных узлов, хостов) к другим через маршрутизаторы в пределах всей сети.

Основным протоколом сетевого уровня является ***Internet protocol*** (**IP**) – *межсетевой протокол*, в связи с чем сети TCP/IP часто именуют **IP*‑сетями***.

**Протокол IP** обеспечивает только маршрутизацию и доставку пакетов данных и полностью освобожден от задач обеспечения надежности. Функции транспортного и сетевого уровней четко разделены, чем исключено их дублирование.

Протокол IP реализуется программным обеспечением оконечных устройств пользователей (в том числе граничных хостов) и маршрутизаторов и не зависит от характеристик связывающих их WAN и LAN.

Важной особенностью протокола IP является его *способность выполнять динамическую фрагментацию пакетов* (см. рис. 6.6) при передаче их между сетями с различными, максимально допустимыми значениями поля данных кадров (пакетов) MTU. Свойство фрагментации во многом способствовало тому, что протокол IP смог занять доминирующие позиции в сложных составных сетях.

Существует прямая связь между функциональной сложностью протокола и сложностью заголовка пакетов, которые такой протокол использует. Это объясняется тем, что основные служебные данные содержатся в полях заголовка пакетов. На основании этих данных протокол выполняет то или иное действие модулями, реализующими данный протокол на разных сетевых устройствах. Потому очень полезно изучить назначение каждого поля заголовка IP-пакета, что дает не только формальные знания о структуре пакета, но и объясняет все основные режимы работы протокола по обработке и передаче IP-дейтаграмм.

**Формат заголовка дейтограммы (пакета) IP** показан на рис. 6.4.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | D, T, R | | | | | 15 | 16 |  | | | 31 |
| Версия  (4 бита) | | Длина заголовка  (4 бита) | Приоритет  (3 бита) | Критерий  (3 бита) | Резерв  (2 бита) | | Общая длина пакета  (16 бит) | | | | |
| Идентификатор фрагмента  (16 бит) | | | | | | | Флаги  (3 бита) | | | Смещение фрагмента  (13 бит) | |
|  | D | M |
| Время жизни  (8 бит) | | | Протокол верхнего уровня  (8 бит) | | | | Контрольная сумма  (16 бит) | | | | |
| IP-адрес отправителя (32 бита) | | | | | | | | | | | |
| IP-адрес получателя (32 бита) | | | | | | | | | | | |
| Опции | | | | | | | | | Выравнивание | | |

*Рис. 6.4.* Формат заголовка пакета IP.

Поле «***Номер версии***» (*Version*) длиной 4 бита указывает версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4) и готовится переход на версию 6 (IPv6).

Длина заголовка является переменной величиной, для указания которой выделено поле «***Длина заголовка***» (IHL) длиной 4 бита. Значение длины заголовка измеряется в 32-битовых словах. Минимальная (обычная) длина заголовка 20 байт (пять 32-битовых слов). При увеличении объема служебной информации эта длина может быть больше за счет использования дополнительных байт в поле «Опции» (Options). Наибольший заголовок занимает 60 октетов.

Поле «***Приоритет***» (*Precedence*) длиной 3 бита задает приоритетность пакета. Приоритет может иметь значения от самого низкого – 0 (нормальный пакет) до самого высокого – 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь.

Поле «***Критерий***» длиной 3 бита задает критерий выбора маршрута. Выбор осуществляется между тремя альтернативами, каждой из которых соответствует свой 1-битный флаг: малой задержкой – бит **D** (delay), высокой достоверностью – бит **Т** (true) и высокой пропускной способностью – бит **R** (rate).

Поля «Приоритет» и «Критерий» вместе с полем «Резерв» (из 2 бит) часто объединяют в одно поле, называемое «***Тип сервиса***» (Type of Service) длиной 1 байт.

Поле «***Общая длина пакета***» (*Total Length of Packets*) длиной 2 байта содержит длину всей дейтаграммы, включая заголовок и поле данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байт. В настоящее время этот верхний предел достаточен, однако с появлением гигабитных сетей могут понадобиться дейтаграммы большего размера.

Поле «***Идентификатор фрагмента***» (*Identification*) длиной 2 байта позволяет хосту определить, какой дейтаграмме принадлежат принятые им фрагменты. Все фрагменты одного пакета содержат одно и тоже значение этого поля.

Поле «***Флаги***» (*Flags*) длиной 3 бита содержит признаки, связанные с фрагментацией. Установленный бит **D** (*Do not Fragment*) запрещает маршрутизатору фрагментировать данный пакет, а установленный бит **M** (*More Fragments*) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Еще 1 бит поля зарезервирован.

Поле «***Смещение фрагмента***» (*Fragment Offset*) длиной 13 бит указывает положение фрагмента в исходной дейтаграмме. Длина всех фрагментов, кроме длины последнего, должна быть кратна 8. Максимальное количество фрагментов в дейтаграмме равно 8192, что покрывает максимальную длину дейтаграммы 65 535 байт.

Поле «***Время жизни***» (*Time to Live*) занимает 1 байт и означает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни этого пакета измеряется в секундах и задается источником передачи. На маршрутизаторах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица; она вычитается и в том случае, когда время задержки меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше, чем за 1 с, то время жизни можно считать равным максимальному числу узлов, которые разрешено пройти такому пакету до того, как он достигнет места назначения. Если параметр времени жизни станет нулевым до того, как пакет достигнет получателя, этот пакет будет уничтожен.

Поле «***Протокол верхнего уровня***» (Protocol) занимает 1 байт и указывает, какому протоколу верхнего уровня (или своего же – сетевого) принадлежит информация, размещенная в поле данных пакета (например, это могут быть сегменты протокола TCP, дейтаграммы UDP, пакеты ICMP или OSPF).

Поле «***Контрольная сумма***» (*Header Checksum*) длиной 2 байта рассчитывается только по заголовку пакета. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, время жизни), контрольная сумма проверяется и повторно рассчитывается при каждой обработке IP-заголовка. Контрольная сумма подсчитывается как дополнение к сумме всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля «Контрольная сумма» устанавливается в нуль. Если контрольная сумма неверна, то пакет будет отброшен, как только ошибка будет обнаружена.

Поля «***IP-адрес отправителя***» (*Source IP Address*) и «***IP-адрес получателя***» (*Destination IP Address*) длиной по 32 бита задают адресную информацию, необходимую для маршрутизации пакетов (номер сети и номер хоста).

Поле «***Опции***» (IP *Options*) длиной до 40 байт является необязательным и используется обычно только при отладке сети. Оно состоит из нескольких подполей. В этих подполях можно указывать точный маршрут прохождения маршрутизаторов, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности, а также временные отметки.

Поле «***Выравнивание***» (*Padding*) используется для дополнения поля «Опции» (нулями) до величины, кратной 32 битам.

**6.2.3.Адресация в IP-сетях**

В стеке TCP/IP используются три типа адресов: физический (**MAC**-***адрес***); сетевой (**IP**-***адрес***); символьный (**DNS**-***имя***).

**Сетевой IP-адрес**

Длина адреса IP (32 бита, IPv4) разделена на две части. Первая часть обозначает адрес сети, вторая - адрес узла (хоста). Номер узла назначается администратором независимо от локального адреса узла. IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение (конечный узел может входить в несколько IP-сетевых связей).

Сети IP могут также быть разделены на более мелкие единицы, называемые подсетями (*subnets*). Подсети обеспечивают дополнительную гибкость для администратора сети. Число битов, занимаемых для адреса подсети, является переменной величиной.

Адреса IP записываются в формате десятичного числа с проставленными точками, например, 192.228.17.57. Адресация IP обеспечивает пять различных классов сети (табл. 6.1). Самые крайние левые биты обозначают класс сети (выделены жирным шрифтом). Наиболее распространены классы A, B и C. Классы D и E существуют, но обычно не используются конечными пользователями.

Табл. 6.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Класс | Первые биты  IP-адреса | Наименьший  **номер сети** | Наибольший  **номер сети** | Максимальное  число сетей | Максимальное 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 | **191.256.**0.0 | 2 14 – 2 | 2 16 – 2 |
| C | 110 | **192.0.0.**0 | **223.256.256.**0 | 2 21 – 2 | 2 8 – 2 |
| D | 1110 | **224**.0.0.0 | **256.**256.256.255 |  |  |
| E | 1111 | **240**.0.0.0 | **256.**256.256.255 |  |  |

Большие сети используют адреса класса А, средние – класса В, маленькие – класса С.

В версии 4 (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 А равно 126.
2. Маршрутизация пакета в публичной сети всегда производится на основании классического IP-адреса номера сети, согласно табл. 1, поэтому адрес сети не может быть назначен ни одному узлу.
3. Адрес узла со всеми двоичными “1” предназначен 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 либо корпоративных сетей. В классе А – это сеть 10.0.0.0, в классе В – диапазоны сетей от 172.16.0.0 до 172.31.0.0, в классе С – диапазон сетей от 192.168.0.0. до 192.168.256.256.

Основное назначение адресов класса D – распространение информации по схеме “один-ко-многим” для групповой рассылки в Интернет аудио- и видеоинформации. Адреса класса Е зарезервированы для будущих применений.

Номер сети принято обозначать с помощью маски. Маска – это число, которое используется вместе с IP-адресом: двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Для классов сетей маски имеют вид:

• класс А – 11111111.00000000.00000000.00000000 (256.0.0.0)

• класс В – 11111111.11111111.00000000.00000000 (256.2556.0.)

• класс С – 11111111.11111111.11111111.00000000 2556.2556.2556.0).

Создание подсетей обеспечивается путем переназначения части битов узла в качестве битов сети. Процесс заимствования части битов всегда начинается с крайнего левого бита.

Поступивший (в маршрутизатор, узел) IP-адрес в двоичном коде с помощью складывается поразрядно с маской с помощью логической операции **“И”: 0+0=0, 0+1=0, 1+0=0, 1+1=1**. Результат сложения (адрес сети) сравнивается с IP-адресом, записанным в первой строке таблицы маршрутизации. При совпадении адресов поступивший пакет направляется на соответствующий интерфейс. В случае несовпадения сравнение проделывается с последующими строками маршрутной таблицы, если они имеются. Если совпадения нет, поступившая дейтаграмма отбрасывается (фильтруется).

Пример. Имеется глобальная сеть с 150 узлами в трех сетях (в разных городах), соединенных маршрутизатором TCP/IP. У каждой из этих трех сетей 50 узлов. Выделяем сеть класса C 192.168.123.0. Это значит, что адреса с 192.168.123.1 по 192.168.123.254 можно использовать для этих 150 узлов.

Два адреса в данном примере – 192.168.123.0 и 192.168.123.255 нельзя использовать, поскольку двоичные адреса с составляющей узла из одних единиц и нолей недопустимы (см. п.п. 2 и 3). Следует просто запомнить, что первый и последний адрес в любой сети и подсети не может быть присвоен отдельному узлу.

Теперь осталось дать IP-адреса 254 узлам. Это несложно, если все 150 компьютеров являются частью одной сети. Однако в данном примере 150 компьютеров работают в трех отдельных физических сетях. Разбиваем сеть на подсети. С помощью маски «одалживаем» несколько разрядов, обычно применяемых для задания адреса узла, и используем их для составляющей сети в адресе. Маска подсети 256.256.256.192 позволяет создать четыре сети с 62 узлами в каждой. Это возможно, поскольку в двоичном обозначении 256.256.256.192 – то же самое, что и 1111111.11111111.1111111.11000000. Первые две цифры последнего октета становятся адресами сети, поэтому появляются дополнительные сети 00000000 (0), 01000000 (64), 10000000 (128) и 11000000 (192). В этих четырех сетях последние 6 двоичных цифр можно использовать в качестве адресов узлов. Эти четыре сети будут иметь следующие действующие адреса узлов: 192.168.123.1-62; 192.168.123.65-126;

192.168.123.129-190; 192.168.123.193-254.

Не забываем, что двоичные адреса узлов с одними только единицами и нолями недействительны, поэтому нельзя использовать адреса со следующими числами в последнем октете: 0, 63, 64, 127, 128, 191, 192 или 256.

Адреса версии **IPv6** имеют 7 байт на адрес. Адреса назначаются отдельным интерфейсам узлов, а не самим узлам. У одного интерфейса может быть несколько уникальных адресов для целевых передач. Более длинные межсетевые адреса позволяют объединять адреса по сетевым иерархиям, поставщикам услуг, географическому расположению, корпорациям. Получаются таблицы маршрутизации – ускоряется процедура поиска в таблице.

В терминологии TCP/IP под ***физическим*** (именуемым также ***локальным*** или ***аппаратным***) адресом понимается такой тип адреса, который предназначен для доставки данных в пределах подсети. Если подсеть – локальная сеть, то используется МАС-адрес: сетевые адаптеры, интерфейсы маршрутизаторов. Адрес уникальный, назначается производителем (6 байт). Бывают исключения: некоторые компьютеры в локальной сети могут иметь несколько локальных адресов даже при одном адаптере. Однако протокол IP может работать и над протоколами более высокого уровня, например над IPX или Х.26. В этом случае локальными адресами для протокола IP соответственно будут адреса IPX и Х.26.

Глобальные *порты* маршрутизаторов, предназначенные для соединений «точка-точка», не имеют локальных адресов

***Символьные доменные адреса*** строятся по иерархическому принципу: простое имя конечного узла, имя группы узлов (организация), имя более крупной группы (поддомена),…, до имени домена, объединяющие организации по географическому принципу: ru-Россия, vk-Великобритания, su-США. Domain Name System (DNS) – дает необходимое соответствие.

Пример символьного имени: cityline.spb.ru.

Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP применяется специальная распределенная ***служба доменных имен*** **DNS** (***Domain Name System***), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Потому доменные имена называют также **DNS**-***именами***.

**6.3. Отображение IP-адресов на локальные адреса**

Важная задача уровня межсетевых интерфейсов – отображение IP-адресов в локальные адреса. Используется ARP (Address Resolution Protocol – протокол разрешения адреса). Реверсивный ARP (Reverse ARP – RARP) решает обратную задачу – нахождение IP-адреса по известному локальному адресу. Используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера.

ARP –обращение: IP-адрес узла назначения известен модулю IP. Требуется на его основе найти МАС-адрес узла назначения. Работа ARP начинается с просмотра ARP-таблицы. Каждая строка таблицы – соответствие между

IP-адресом и МАС-адресом. Для каждой сети, подключенной к сетевому ада- птеру компьютера или порту маршрутизатора, строится отдельная таблица.

|  |  |  |
| --- | --- | --- |
| IP-адрес | МАС-адрес | Тип записи |
| 194.86.136.75 | 008048ЕВ7ЕG0 | динамический |
| 194.86.60.21 | 008048ЕВ7567 | статический |

Выделяется специальный маршрутизатор, который ведет таблицу ARP для всех маршрутизаторов и узлов ГЛОБАЛЬНОЙ СЕТИ.

**6.4. Фрагментация**

Все сети накладывают ограничения на размер своих пакетов. Эти пределы вызва­ны различными предпосылками, среди которых есть следующие:

1. Аппаратные (например, размер кадра Ethernet).
2. Операционная система (например, все буферы имеют размер 512 байт).
3. Протоколы (например, количество бит в поле длины пакета).
4. Соответствие какому-либо международному или национальному стандарту.
5. Желание снизить количество пакетов, пересылаемых повторно из-за ошибок передачи.
6. Желание предотвратить ситуацию, когда один пакет слишком долгое время занимает канал.

Результатом действия всех этих факторов является то, что разработчики не могут выбирать максимальный размер пакета по своему усмотрению. Максималь­ный размер поля полезной нагрузки варьируется от 48 байт (АТМ-ячейки) до 65 515 байт (1Р-пакеты), хотя на более высоких уровнях размер поля полезной нагрузки часто бывает больше.

Возникает проблема, когда большой пакет хочет пройти по сети, в которой максимальный размер пакетов слишком мал. Единственное решение заключается в разрешении шлюзам разбивать пакеты на фрагменты и посылать каждый фрагмент в виде отдельного межсетевого пакета. Но возникает проблема с восстановлением пакетов из фрагментов.

Для восстановления исходных пакетов из фрагментов применяются две про­тивоположные стратегии. Первая стратегия заключается в том, чтобы «мелкопакетная» сеть имела шлюзы (специализированные маршрутизаторы), предоставляющие интерфейсы другим сетям. Когда на такой шлюз приходит пакет слишком большого размера, он разбивается на фрагменты (рис. 6.5, *а*.). Каждый фрагмент адресуется одному и тому же выходному шлюзу, восстанавливающему из этих фрагментов пакет. Прохождение данных через мелкопакетную сеть остается прозрачным для хостов.

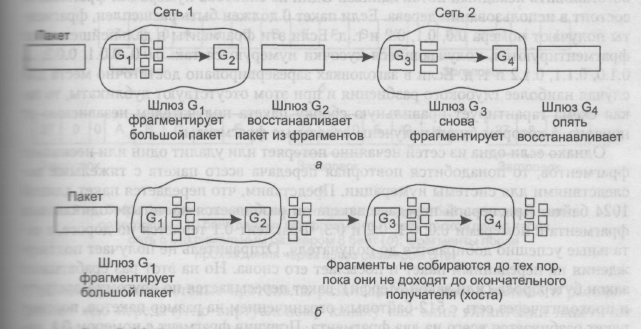


Рис. 6.5. Прозрачная фрагментация (*а*), непрозрачная фрагментация (*б*)

Другая стратегия фрагментации состоит в отказе от восстановления пакета из фрагментов на промежуточных маршрутизаторах. Как только пакет оказывается разбитым на отдельные фрагменты, с каждым фрагментом обращаются как с отдельным пакетом. Все фрагменты проходят через выходной шлюз или несколько (рис. 6.5, *б*)*.* Задача восстановления оригинального пакета возложена на получающий хост. Так работает IР.

Каждый фрагмент должен иметь заголовок. Фрагменты пакета нумеруются таким образом, чтобы можно было восстановить исходный поток данных. Так, например, заголовок межсетевого пакета включает два поля – порядкового номера исходного пакета и порядкового номера фрагмента.

**6.5. Транспортный уровень**

Транспортный уровень обеспечивает обмен сообщениями между прикладными процессами. Идентификация процесса получателя осуществляется по адресу, состоящему из двух частей: IP-адреса, идентифицирующего оконечное устройство, и номера порта, идентифицирующего прикладной процесс. В данном случае «порт» – это не физический разъем или канальный вход/выход, а условный номер прикладного процесса или службы прикладного уровня.

На транспортном уровне стека TCP/IP поддерживаются два режима: с установлением соединения (при использовании протокола TCP) и без него (при применении протокола UDP) (рис.6.6)

Прикладной уровень

**UDP**

**TCP**

**IP**

Линейные протоколы

(сетевые интерфейсы)

в сеть

(кадр, фрейм)

TFTP

NSP

HTTP

FTP

Telnet

SMTP

SNMP

Соответствие уровней стека протоколов

TCP/IP семиуровневой модели OSI

HTTP FTP Telnet SNMP SMTP TFTP NSP

6-7 уровни

TCP UDP

5-4 уровни

IP ICMP RIP OSPF ARP

3- уровень

Не регламентируется

Ehternet, Token Ring, FDDI, X.25, ATM, Fast Ehternet, Gigabit Ehternet, Fibre Cannel

2-1 уровни

Рис.6.6. Уровни стека протоколов TCP/IP

**6.5.1. Протокол TCP**

Протокол **TCP** (Transmission Control Protocol) – протокол управления передачей, реализован только на оконечных системах. Обеспечивает надежную доставку блоков данных с помощью установления между источником и адресатом логического (виртуального) канала.

Установление логического соединения позволяет нумеровать сегменты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню сегменты в том порядке, в котором они были отправлены. Протокол TCP дает возможность объектам одного ранга на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме.

Протокол TCP предоставляет полноценную транспортную службу, которая обеспечивает обмен потоками данных. При этом он не накладывает ограничений на состав потока, освобождая прикладной процесс от функции структурирования данных. Передача данных протоколу TCP аналогична их записи в неструктурированный файл.

Целостность потока данных обеспечивается квитированием, при этом для управления потоком данных используется механизм окна. Управление шириной окна позволяет защищать от перегрузок как промежуточные узлы сети, так и буферную память протокола TCP, принимающего данные. Первую задачу решают маршрутизаторы, направляя протоколам оконечных станций требования об уменьшении размера окна. Вторая задача решается непосредственно протоколом TCP, который декларирует выбранную ширину окна, используя при необходимости и нулевую ширину окна, т.е. запрещая передачу.

Форматы ПБД транспортного и сетевого уровней TCP/IP показаны на рис. 6.7.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Поток данных | | |  | |  | |  |
|  |  |  |  | |  | |  |
| Сегмент TCP/IP  (дейтаграмма UDP) | | Заголовок  ≥ 20 байт | Данные | |  | |  |
|  |  | ≤ 65535 (216–1) байт | | |  | |  |
| IP-пакет | Заголовок  ≥ 20 байт |  |  |  |  | |  |
|  |  |  |  |  |  | |  |
| Фрагменты IP-пакетов | 1 |  | 2 |  | 3 |  |  |

Рис. 6.7. Формат протокольных блоков данных TCP/IP.

Сегмент TCP состоит из полей заголовка и данных. Длина информационного поля данных сегмента может меняться в широких пределах с учетом двух ограничений. Во-первых, каждый сегмент, включая его заголовок и заголовок IP-пакета, не должен превышать 65 535 = 216–1 байт. Во-вторых, в каждой сети есть соответствующая используемым ПБД максимальная единица передачи (**MTU** – ***Maximum Transfer Unit***), в которую должен помещаться сегмент (вместе с заголовком IP-пакета). Впрочем, если сегмент (пакет) проходит через последовательность сетей и попадает в сеть, MTU-единица которой оказывается меньше него, то пограничный маршрутизатор фрагментирует его на несколько частей (см.6.4).

Сегменты могут и не содержать поля данных при передаче квитанций и управляющих сообщений.

Формат заголовка сегмента приведен на рис. 6.В 8. заголовке сегмента 11 полей и 6 служебных бит..

Поля ***«№ порта отправителя***» и ***«№ порта получателя***» (по 16 бит каждое) идентифицируют службы прикладного уровня или прикладные процессы отправителя и получателя.

Поля ***«№ последнего переданного байта***» и «***№ первого ожидаемого на приёме байта***» (по 32 бит каждое) используются для управления потоком данных. Нумеруются не ПБД (сегменты), а байты в сегменте. Причем цикл нумерации составляет 232 байт = 4 Гбайт, т.е. он вполне может применяться не для относительной нумерации (по кругу), а для абсолютной (например, для последовательной нумерации байт в передаваемых файлах).

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 |  | | | | | | | 15 | 16 |  | | 31 |
| № порта отправителя | | | | | | | | | № порта получателя | | | |
| № последнего переданного байта | | | | | | | | | | | | |
| № первого ожидаемого на приеме байта | | | | | | | | | | | | |
| Длина  заголовка  (4 бита) | | Резерв  (6 бит) | U  R  G | A  C  K | P  S  H | R  S  T | S  Y  N | F  I  N | Ширина скользящего окна | | | |
| Контрольная сумма | | | | | | | | | Указатель на срочные данные | | | |
| Дополнительная информация | | | | | | | | | | | Выравнивание | |

Рис. 6.8*.* Формат заголовка сегмента TCP

Поле «***Длины заголовка***» (4 бита) определяет длину заголовка сегмента TCP, измеренную в 32-битовых блоках (словах, соответствующих типовой адресуемой единице в 32-разрядных компьютерах). Длина заголовка не фиксирована, она зависит от числа бит в поле «***Дополнительная информация***», которое округляется с помощью пустого поля «***Выравнивание***» в большую сторону до кратной 32 битам величины. Минимальный размер заголовка при отсутствии дополнительной информации 20 байт. Максимальный размер, который можно записать в поле из 4 бит, составляет 32 бит·**(**24 – 1) = 480 бит = 60 байт.

Следующие 6 бит зарезервированы для последующего использования.

Далее следуют 6 служебных бит (1-битных флагов). Активному состоянию служебных бит соответствует значение 1. Флаги используются в качестве следующих признаков (указателей):

**URG** – указатель наличия срочных данных в сегменте;

**ACK** – указатель наличия осмысленных данных в поле «***№ первого ожидаемого на приёме байта***» (квитанция) на ранее принятые данные. При ACK = 0) указанное поле игнорируется;

**PSH** – указатель получателю, чтобы он доставил данные в сегменте сразу прикладному процессу, а не хранил его в буфере;

**RST** – указатель запроса на сброс и переустановку соединения (из-за сбоя хоста или другой тупиковой ситуации);

**SYN** – признак *служебных сегментов* «***Connection Request***» – запрос соединения (при ACK = 1) и «***Connection Accepted***» – согласие на соединение (при ACK = 0);

**FIN** – признак того, что у отправителя нет больше данных для передачи. Используется для разрыва соединения.

У сегментов с битами FIN и SYN есть порядковые номера, что гарантирует правильный порядок их выполнения.

Поле «***Ширина скользящего окна***» (16 бит) используется для управления потоком данных в виртуальном соединении и сообщает, сколько байт может быть послано после получившего подтверждения байта. Нулевое значение данного поля является командой на приостановку передачи (что напоминает служебный кадр RNR в протоколе HDLC). Командой на продолжение передачи служит ненулевое значение поля «ширина скользящего окна» при такой же величине поля ««***№ первого ожидаемого на приеме байта***», как у команды на приостановку передачи (служебный кадр RR в протоколе HDLC).

Поле «***Контрольная сумма***» (16 бит) содержит проверочную последовательность, позволяющую обнаруживать ошибки в сегменте. Заполняется и проверяется данное поле без использования циклического кодирования. Алгоритм вычисления контрольной суммы просто складывает все 16-ричные слова в дополнительном до 1 коде. Затем рассчитывает дополнение для всей суммы. Получатель считает контрольную сумму всего сегмента, включая поле «Контрольная сумма», результат должен быть равен нулю.

Поле «***Указатель на срочные данные***» содержит смещение в байтах от текущего порядкового номера байта до места расположения срочных данных. Поле проверяется в случае, когда флаг URG установлен в 1. Так в протоколе TCP/IP реализуются прерывающие сообщения прикладного уровня.

Использование скользящего окна протоколом TCP сочетается с процедурой выборочного повтора, аналогичную процедуре РОС-АП, поддерживаемой стандартом HDLC..

В целом протокол TCP поддерживает множество процедур, обеспечивающих эффективное управление TCP-соединением, передачей данных, борьбой с перегрузками, таймерами.

**6.5.2. Протокол UDP**

Протокол **UDP** (User Datagram Protocol) – предназначен для передачи дейтаграмм на уровне абонент – абонент без установления соединения, ориентирован на транзакции, не гарантирует доставку пакета или отсутствие его дубликата.

Прикладные программы через протокол UDP получают доступ к сетевому уровню почти без обработки на транспортном уровне. Многие приложения «клиент-сервер», для того чтобы обменяться одним запросом и ответом, предпочитают не устанавливать соединения, а пользоваться протоколом UDP (см. протоколы прикладного уровня, применяющие транспортные услуги UDP).

Как и сегмент TCP, дейтаграмма UDP состоит из заголовка и блока данных (см. рис. 2.17). Длина информационного поля данных дейтаграммы, как и сегмента, может достигать *W*С.И = 65 535 байт. А заголовок дейтаграммы намного проще, чем заголовок сегмента, и включает всего четыре 16-битных поля (рис. 6.9).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 |  | 15 | 16 |  | 31 |
| № порта отправителя | | | № порта получателя | | |
| Длина дейтаграммы | | | Контрольная сумма | | |

Рис. 6.9. Формат заголовка дейтаграммы UDP.

Поля заголовка ***«№ порта отправителя***» и ***«№ порта получателя***» (по 16 бит каждое) так же, как аналогичные поля в заголовке сегмента, идентифицируют службы прикладного уровня или прикладные процессы отправителя и получателя.

Поле «***Длина дейтаграммы***» включает суммарный размер 8-байтового заголовка и поля данных дейтаграммы.

Поле «***Контрольная сумма***» вычисляется и проверяется таким же способом, как и в сегменте TCP. Контрольная сумма может не рассчитываться. Тогда это поле содержит нули.

Протокол UDP самостоятельно не может управлять потоком, следить за порядком следования дейтаграмм и переспрашивать искаженные или потерянные дейтаграммы (хотя обнаруживать искажения может).

Таким образом, основные функции протокола UDP – мультиплексирование и демультиплексирование (распределение по портам) потока дейтаграмм между приложениями. Кроме этого, использование контрольной суммы позволяет контролировать достоверность данных.

**6.5.3. Модели реализации протокола TCP**

Протокол ТСР функционирует нормально при выполнении ряда условий.

1. Вероятность ошибки доставки невелика, и потеря пакета вероятнее всего происходит из-за переполнения буфера. Если потеря пакета из-за его искажения существенна, уместно поискать оптимальное значение MTU.
2. Время доставки достаточно стабильно. Для его оценки можно использовать простые линейные аппроксимации и модель виртуального канала. Смена порядка прихода пакетов маловероятна.
3. Сеть имеет фиксированную полосу пропускания и не допускает скачкообразных ее вариаций
4. Буферы сетевых устройств используют схему первый\_вошел-первым\_вышел (FIFO). Предполагается, что размер этих буферов соответствует произведению RTT\*B (B - полоса пропускания, RTT - сумма времен транспортировки сегмента от отправителя к получателю и времени движения отклика от получателя к отправителю). Если это условие нарушено, пропускная способность неизбежно понизится и будет определяться размером буфера, а не полосой пропускания канала.
5. Чтобы минимизировать влияние избыточности, связанной с заголовком (20 байт IP +20 байт ТСР + МАС-заголовок), используемое поле данных должно иметь большой объем. Для узкополосных каналов, где MTU мало, нарушение данного требования делает канал низкоэффективным. По этой причине выявление допустимого MTU в начале сессии должно приветствоваться.
6. Взаимодействие с другими ТСР-сессиями не должно быть разрушительным, приводящим к резкому снижению эффективности виртуального канала.

В настоящее время предложено и опробовано несколько разновидностей протокола TCP.

Анализируя различные модели работы протокола ТСР, следует учитывать, что в сети Интернет могут встречаться участки с разными протоколами L2 (Ethernet, ATM, SDH, Frame Relay, PPP и т.д.). Эти технологии имеют разные алгоритмы обработки ситуаций перегрузки (или не иметь их вовсе), а отправитель и получатель, как правило, не имеют данных о том, какие протоколы уровня L2 реализуют виртуальное соединение (L4).

Модификации (модели) протокола ТСР связаны с алгоритмами регулирования размера ***окна*** в зависимости от частоты перегрузок и времени «обращения» сегмента данных.

(TCP-reno, TCP Vegas, TCP-Tahoe, TCP Westwood, Модель TCP Hybla, Модель BIC TCP, Модель CUBIC TCP, [TCP-Illinois](http://www.ews.uiuc.edu/~shaoliu/tcpillinois/index.html), [TCP-Veno](http://wwwcsif.cs.ucdavis.edu/~wangw/Performance%20Study%20of%20TCP%20Veno%20over%20WLAN%20and%20RED%20Router.pdf))

**6.6. Прикладной уровень**

Прикладной уровень (*уровень приложений – application layer*) объединяет службы, предоставляющие телекоммуникационные услуги различным пользовательским приложениям. Протоколы прикладного уровня ориентированы на конкретные службы. Они определяют как процедуры по организации взаимодействия конкретного типа между прикладными процессами, так и форму представления информации при таком взаимодействии.

Основными протоколами прикладного уровня являются:

**Telnet** (*tele* – далеко, *net* – сеть) – протокол удаленного доступа, обеспечивающий посимвольный (побайтный) обмен информацией (в кодах ASCII – American Standard Code for Information Interchage) между терминалами и внутрисетевыми элементами (узлами, компьютерами-хостами). Передача осуществляется с использованием протокола TCP. Работа с протоколом на компьютере напоминает работу с программой Hyper Terminal в ОС Windows.

**FTP** (*File Transfer Protocol*) – протокол передачи файлов. Является одним из старейших протоколов семейства TCP/IP. Как и Telnet, пользуется транспортными услугами TCP.

**TFTP** (*trivial FTP*) – простейший протокол передачи файлов. Использует транспортные услуги UDP. Является одним из старейших протоколов семейства TCP/IP.

**SMTP** (*Simple Mail Transfer Protocol*) – простой протокол передачи почты. Поддерживает передачу сообщений (электронной почты) между узлами сети (почтовыми серверами). Имея механизмы промежуточного хранения почты и механизмы повышения надежности доставки, SMTP допускает применение различных транспортных служб. Он может работать даже в сетях, не использующих протоколы семейства TCP/IP.

**NFS** (*Network File System*) – сетевая файловая система. Использует транспортные услуги UDP и позволяет монтировать в единое целое файловые системы нескольких машин с ОС UNIX. Бездисковые рабочие станции получают доступ к дискам файл-сервера так, как будто это их локальные диски.

**SNMP** (*Simple Network Management Protocol*) – простой протокол управления сетью. Использует транспортные услуги UDP. Он позволяет управляющим станциям собирать информацию о положении дел в сети Internet. Протокол определяет формат данных, их обработка и интерпретация остаются на усмотрение управляющих станций или менеджера сети.

**X-Window** – протокол многооконного отображения графики и текста на растровых дисплеях рабочих станций (соответствует оконному интерфейсу в операционных системах UNIX и LINUX). Использует транспортные услуги TCP.

**HTTP** (*Hyper Text Transport Protocol*) – протокол передачи гипертекстовой информации. Основа формирования всемирной информационной службы (всемирной информационной паутины) – **WWW** (*World Wide Web*). Использует транспортные услуги TCP.

**RTP** (*Real-time Transport Protocol*) – протокол передачи данных в реальном времени. Разработан для обеспечения передачи аудио- и видеосигналов по сети Интернет с ограниченной допустимой задержкой (**IP*-телефония***). Использует транспортные услуги UDP. Тесно связан с еще одним протоколом прикладного уровня – **RTCP** (*Real-time Transport Control Protocol*) – протоколом управления передачей в реальном времени. С помощью данного протокола прикладные программы могут приспосабливаться к изменению нагрузки на сеть. Например, в случае перегрузки, получив сигнал RTCP, алгоритм кодирования речи может увеличить сжатие (снизив качество).

**ПРИЛОЖЕНИЕ 6. Дополнительные протоколы сетевого уровня**

Ряд вспомогательных служебных протоколов, дополняющих протокол IP множеством вспомогательных функций, можно поделить на три группы:

– сигнализации (IСMP);

– адресования и маршрутизации (ARP, RARP, IGMP, BGP, RIP, IGRP, OSPF и др.);

– обеспечения качества (RSVP, IntServ, DiffServ, MPLS и др.).

**IСMP** (***Internet Control Message Protocol***) – *протокол обмена управляющими сообщениями*. Позволяет маршрутизатору сообщить конечному узлу об ошибках, с которыми машрутизатор столкнулся при передаче какого-либо IP-пакета от данного конечного узла.

Сообщения протокола ICMP передаются по сети внутри пакета IP. Пакеты IP с сообщениями ICMP маршрутизируются точно так же, как и любые другие пакеты, без приоритетов, поэтому они тоже могут теряться. Кроме того, в загруженной сети они могут вызывать дополнительную загрузку маршрутизаторов. Для того чтобы не вызывать лавины сообщения об ошибках, потери пакетов IP, переносящие сообщения ICMP об ошибках, не могут порождать новые сообщения ICMP.

Протокол ICMP – это протокол сообщения об ошибках, а не протокол коррекции ошибок. Конечный узел может предпринять некоторые действия для того, чтобы ошибка больше не возникала, но они протоколом ICMP не регламентируются.

Протокол ICMP предоставляет сетевым администраторам средства для тестирования достижимости узлов сети.

Маршрутизаторы могут использовать сообщения протокола ICMP для корректировки маршрутно-адресных таблиц в оконечных устройствах (компьютерах), которые требуют согласования с изменяющимися (динамически) маршрутно-адресными таблицами маршрутизаторов.

**ARP** (***Address Resolution Protocol***) – *протокол разрешения адреса*. Используется для преобразования IP-адреса в локальный адрес.

**RARP** (***Reverse Address Resolution Protocol***) – *реверсивный ARP протокол*. Используется для преобразования локального адреса в IP-адрес.

**IGMP** (***Internet Group Management Protocol***) – *межсетевой протокол управления группами*. Используется для динамической регистрации отдельных хостов в многоадресной группе. Групповая IP-адресация позволяет уменьшать трафик за счет доставки одного потока информации сразу многим адресатам.

**BGP** (***Border Gateway Protocol***) – *пограничный межсетевой протокол*. Выполняет функции протокола маршрутизации между автономными системами. Является развитием протокола **EGP** (***Exterior Gateway Protocol***) – протокола внешних шлюзов. Может использоваться и внутри автономных систем в роли протокола внутренних шлюзов **IGP** (***Interior Gateway Protocol***), к которым также относятся протоколы RIP и IGRP. Имеет большой перечень опций, влияющих на выбор предпочтительных маршрутов.

**RIP** (***Routing Information Protocol***) – *протокол маршрутной информации* или протокол маршрутизации *по вектору расстояния*. Основан на подсчете числа промежуточных ретрансляций в маршрутизаторах и не учитывает реальную пропускную способность каналов передачи данных между маршрутизаторами. Требует обмена между соседними маршрутизаторами сообщений об обновлении маршрутно-адресных таблиц. Это самый старый протокол маршрутизации, постоянно критикуемый, но широко используемый в разных версиях.

**IGRP** (***Interior Gateway Routing Protocol***) – *протокол маршрутизации внутренних шлюзов*. Является развитием протокола RIP. Использует составную метрику, вычисляемую на основе взвешенных математических значений задержки, скорости передачи, надежности и загрузки. Допускает многотрактовую маршрутизацию (с резервированием).

**OSPF** (***Open Shortest Path Firs***) – *открытый протокол предпочтительного выбора кратчайшего пути* или протокол маршрутизации *по состоянию канала*. Оптимизация маршрутов осуществляется обычно на основе измерения времени задержки пакетов. Требует отправки маршрутизаторами сообщений о состоянии прилегающих каналов всем остальным маршрутизаторам сети или ее части.

**RSVP** (***Resource reSerVation Protocol***) – *протокол резервирования ресурсов*. Позволяет Internet-приложениям обеспечивать различное качество обслуживания QoS (Quality of Service) для разных потоков данных. Особенностью RSVP является то, что запросы на резервирование ресурсов (скорости передачи или, иначе, *полосы пропускания*) отправляются от получателей данных к отправителям, а не наоборот.

**IntServ** (***Integrated Service***) – протоколы рабочей группы IETF по интегрированному обслуживанию. Предполагают сигнализацию из конца в конец на основе RSVP. Определяют три класса обслуживания: по мере возможности; с контролируемой загруженностью; с гарантированным обслуживанием.

**DiffServ** (***Differentiated Services***) – протокол по дифференцированному обслуживанию. Предлагает более простой по сравнению с RSVP и IntServ масштабируемый метод обеспечения QoS для приложений реального времени. Основан на переопределении 8-битного поля в заголовке IP «Тип сервиса» в поле «Дифференцированное обслуживание». Классифицирует и объединяет однотипные потоки. Требует модернизации микропрограммного обеспечения маршрутизаторов.

**MPLS** (***Mult Protocol Label Switching***) – многопротокольная коммутация меток. Благодаря данному протоколу сетевые устройства определяют маршруты на основании требований приложений к QoS и реальной пропускной способности промежуточных сетей на 2-м уровне ЭМВОС (относительно сетевого уровня TCP/IP). Добавляет к пакетам IP при входе в сеть (и удаляет при выходе из сети) 32-разрядные метки для информирования коммутаторов и маршрутизаторов о природе трафика. Привносит в дейтаграммные пакетные сети элементы виртуальных каналов с установлением соединений.

Приведенные выше протоколы обеспечения качества являются элементами целой системы стандартов построения мультисервисных сетей с коммутацией пакетов, способных предоставлять комплекс мультимедийных услуг по передаче данных, речевой, видео и графической информации.