Раздел 5. Интернет технологии

5.1. Общие сведения об Интернет

Интернет (Internet) – это сеть, объединяющая отдельные локальные, региональные, национальные и глобальные сети, это глобальное сообщество мировых ИВС, которые используют для информационного обмена семейство протоколов TCP/IP.

Дословно термин «Internet» означает «между сетей». Это отражает основную функцию Internet - объединение не только отдельных ЭВМ (хост-машин), но и обеспечение связи между различными сетями в глобальном масштабе. Это объединение даёт возможность обмена информацией между всеми ЭВМ, входящими в сети, подключённые к Internet. При этом не важно, в какой операционной системе работают хост-машины (Windows, UNIX и т.п.).

Справка. Определение термина Internet было дано 24.10.1995 года Федеральным советом США по компьютерным сетям (FNC - Federal Networking Couneil): "Internet - это глобальная информационная система, которая логически соединена посредством адресного пространства, основанного на протоколе IP (Internet Protokol) или заменяющих его протоколов, способна поддерживать передачу данных посредством протокола TCP (Transmission Contrel Protocol) или заменяющих его протоколов, обеспечивает, использует или делает доступными услуги по передаче данных и соответствующую инфраструктуру".

История создания сети Интернет

1962 г: Пол Бэран из американского мозгового центра времен «холодной войны» (Rand Corporation) предложил коммутацию пакетов (КП) в качестве надежной сетевой технологии.

Леонард Клейнрок разработал базовые принципы пакетной коммутации, ставшие основой Интернет.

1964 г.: Rand Corp. публикует концептуальные положения будущей надежной сети ПД.

1969 г.: Агентство перспективных исследований Министерства обороны США (DARPA) финансирует проект создания сетей с КП и принимает решение объединить суперкомпьютеры оборонных, научных и управляющих центров в единую сеть, которая получила название ARPANET (Advanced Research Projects Agency Network). Первая действующая информационная сеть ARPANET, объединила компьютерные системы университетов Лос Анджелеса, Стэнфорда, Санта Барбары и Солт Лейк Сити.

1974 г: Винтон Серф и Роберт Канн публикуют основные принципы работы протоколов TCP/IP.

1980 г.: История глобальной сети Интернет начинается примерно с 1980г., когда ARPA стало переводить компьютеры, подключенные к своим исследовательским центрам, на протоколы TCP/IP. Модернизацию начали с ARPANET.

1982 г.: Для сети ARPANET утверждено семейство протоколов TCP/IP.

1983 г.: Штаб-квартира Минобороны США объявила, что все их компьютеры переведены на TCP/IP. В этом же году Минобороны США разделило ARPANET на две независимые сети: научно-исследовательскую – ARPANET и военную – MILNET.

1984 г.: Национальный научный фонд США (NSF) начал инвестировать научную компьютерную сеть NSFNET.

1986 г.: Создание Национальным научным фондом США компьютерной сети NSFNET, которая объединила научные центры и университеты США. В качестве базовых протоколов были выбраны протоколы TCP/IP. К NSFNET примкнули NASA, DOE (Министерство энергетики), DOD (Министерство обороны) и Национальный институт здравоохранения. Появились шесть первых имен доменов: gov, mil, edu, com, org и net.

1986 г. можно считать годом становления глобальной компьютерной сети Интернет с опорной сетью NSFNET.

1989 г.: Последний год ARPANET (руководство ARPANET не сочло возможным войти в проект NSFNET и дальнейшее развитие Интернет (Internet) продолжалось уже без ARPANET).

Конец 1995 г.: 70 тыс. независимых сетей и 200 тыс. сегментов.

Сейчас Интернет составляют более 200 тыс. отдельных сетей, связывая более 2 млн. узловых компьютеров в 150 странах мира. Более 350 млн. пользователей регулярно используют ресурсы Internet.

Сама сеть Интернет не имеет владельца, однако она соединяет множество сетей ЭВМ, которые имеют своих владельцев. Многие из таких сетей ЭВМ (либо отдельные хост-ЭВМ) предоставляют на коммерческой основе различную информацию, полезную во многих сферах жизнедеятельности человека. Эта информация накапливается в информационных банках национальных сетей, а доступ обеспечивается средствами Интернет, что, собственно, и объясняет всемирную популярность Интернет.

У истоков Интернет в России стоят компьютерные сети ОИЯИ (г. Дубна) и Института им. Курчатого И. В. (г. Москва).

Высшая школа – естественный и активный участник работ по развитию Интернет в России. В 1993 году в Госкомвузе РФ были разработаны концепция и программа создания российской университетской компьютерной сети, которая получила название RUNNet (Russian UNiversity Network). Сеть RUNNet необходима для достижения двух целей: формирования единого информационного пространства российской высшей школы и его интеграции в мировую

информационную систему образования, науки и культуры, развивающуюся в рамках глобальной сети Интернет. В 1994-95 годах была создана основа RUNNet – опорная сеть, обеспечивающая магистральную связь между всеми экономическими регионами России и подключение к Интернет через зарубежные академические сети. В эту опорную сеть включены компьютерные узлы крупных научных и учебных центров страны, связанные между собой спутниковыми каналами.

5.1.1. Обобщённая структура сети Интернет

В архитектуре Интернет отдельные сети (ЛВС, региональные и глобальные) соединяются друг с другом специальными устройствами – маршрутизаторами IP-пакетов (IP-шлюзами или IP-маршрутизаторами, или Router).

Шлюз подключается к двум или более сетям, каждая из которых воспринимает этот шлюз как хост-ЭВМ. Поэтому шлюз имеет физический интерфейс и специальный IP-адрес в каждой из подключаемых сетей.

Передача пакетов требует от шлюза определение IP-адреса следующего шлюза или, на последнем участке, IP-адреса хост-машины, которой направляется IP-пакет.

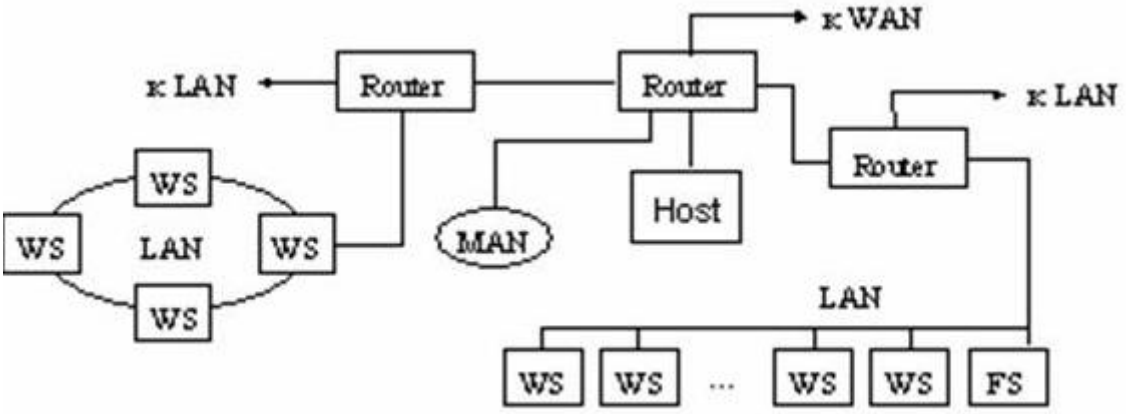


Рис. 5.1. Пример фрагмента сети Интернет

LAN – локальная вычислительная сеть;
MAN – региональная ИВС;
WAN – глобальная ИВС;
WS (Work Station) – рабочая станция ЛВС;
FS (File Server) – файл-сервер;
Host – узловая машина (компьютер, который подключен к сети в качестве узла);
Router – IP-маршрутизатор.

Функция шлюза, которая обычно называется маршрутизацией, основана на анализе специальных маршрутных таблиц (матриц

маршрутов), которые находятся в специальной базе данных. База данных в каждом из шлюзов должна постоянно обновляться, чтобы отражать текущую топологию сети Интернет.

Маршрут – это последовательность маршрутизаторов, которые проходит пакет от отправителя до пункта назначения.

В основе функционирования сети Интернет заложены протоколы TCP/IP.

Основные протоколы семейства TCP/IP приведены на рис. 5.2. Согласно рис. 5.2 одни протоколы верхнего уровня (например, Telnet и FTP) зависят от TCP, а другие (например, TFTP и RPS) — от UDP. Большинство из них используют только один из этих транспортных протоколов, но некоторые (например, DNS) — оба.

На рис. 5.3 показан пример цепочки протоколов TCP/IP.

Данные передаются в пакетах. Пакеты имеют заголовок, который содержит служебную информацию. Данные более верхних уровней вставляются в пакеты нижних уровней. На рис. 5.4 показана передача сообщений в сети Интернет на основе механизма инкапсуляции *(encapsulation).*

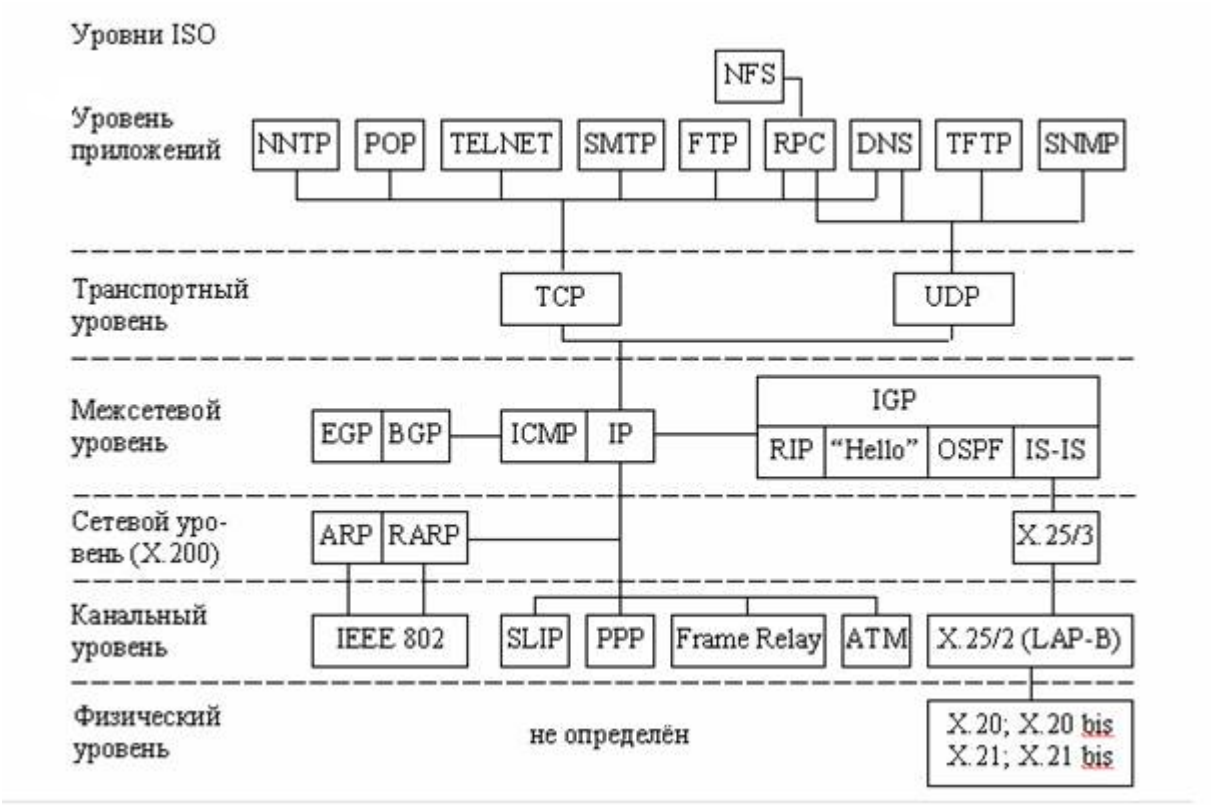


Рис. 5.2. Основные протоколы семейства TCP/IP:

NFS — Network File System — сетевая файловая система;

NNTP — Network News Transfer Protocol — протокол сетевой передачи новостей;

POP — Post Office Protocol — протокол почтового отделения;

TELNET — Terminal Networking — протокол и программные средства, позволяющие подключаться к удалённой машине и работать с ней через эмулируемый терминал;

SMTP — Simple Mail Transfer Protocol — простой протокол электронной почты;

FTP — File Transfer Protocol — протокол передачи файлов;

RPC — Remote Procedure Call — вызов удалённых процедур;

DNS — Domain Name Service — служба именования доменов;

TFTP — Trivial File Transfer Protocol — простейший протокол передачи файлов;

SNMP — Simple Network Management Protocol — простой протокол управления сетью;

TCP — Transmission Control Protocol — протокол управления передачей данных;

UDP — User Datagram Protocol — протокол пользовательских дейтаграмм;

EGP — Exterior Gateway Protocol — протокол внешней маршрутизации;

BGP — Border Gateway Protocol — протокол граничных маршрутизаторов;

IP — Internet Protocol — межсетевой протокол;

ICMP — Internet Control Message Protocol — межсетевой протокол управляющих сообщений;

IGP — Interior Gateway Protocol — внутренний протокол маршрутизации;

RIP — Routing Information Protocol — протокол для передачи маршрутной информации;

“Hello” — реализация протокола внутренней маршрутизации;

OSPF — Open Shortest Path First — открытый протокол предпочтения кратчайшего пути;

IS-IS — Intermediate System to Intermediate System Protocol — протокол маршрутизации, выполняющий маршрутизацию данных IP и MOC;

ARP — Address Resolution Protocol — протокол преобразования адресов;

RARP — Reverse Address Resolution Protocol — протокол обратного преобразования адресов;

X.25/3 — протокол пакетного уровня сети передачи данных;

IEEE 802 — стандарт локальных сетей;

SLIP — Serial Line Internet Protocol — межсетевой протокол для последовательного канала;

PPP — Point-to-Point Protocol — протокол “точка-точка”;

Frame Relay — сетевой механизм для быстрой пересылки кадров;

ATM — Asynchronous Transfer Mode — режим асинхронной пересылки;

X.25/2 (LAP-B) — протокол для управления передачей кадров (Link Access Procedures Balanced — сбалансированные процедуры доступа к каналу);

X.20; X.20 bis — сопряжение оборудования обработки данных с асинхронными модемами;

X.21; X.21 bis — сопряжение оборудования обработки данных с синхронными модемами.

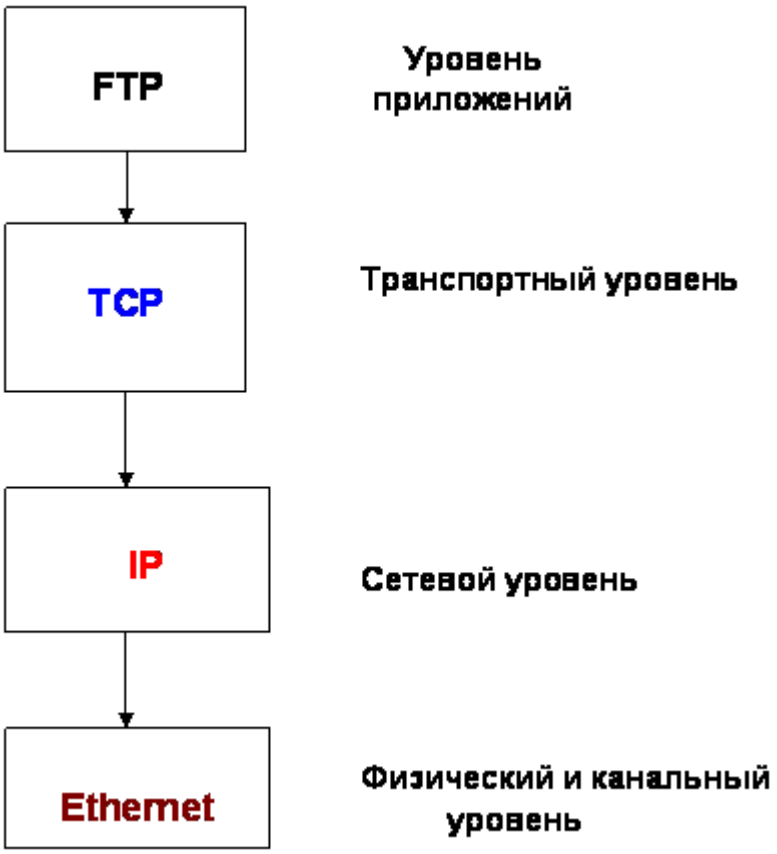
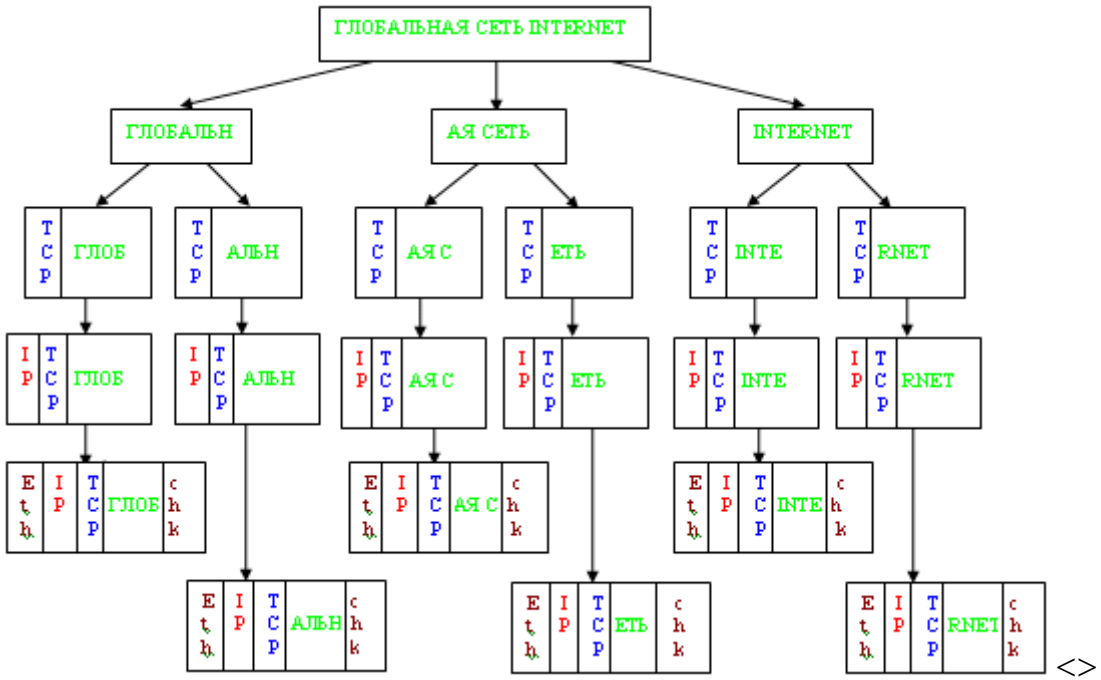


Рис. 5.3. Пример цепочки протоколов TCP/IP



<>

Рис. 5.4. Передача сообщений в сети Интернет на основе механизма инкапсуляции *(encapsulation)*

5.1.2. Стек протоколов TCP/IP

TCP/IP – собирательное название для набора (стека) сетевых протоколов разных уровней, используемых в Интернет. Особенности TCP/IP:

- открытые стандарты протоколов, разрабатываемые независимо от программного и аппаратного обеспечения;
- независимость от физической среды передачи;
- система уникальной адресации;
- стандартизованные протоколы высокого уровня для распространенных пользовательских сервисов.

Стек протоколов TCP/IP делится на 4 уровня:

- прикладной,
- транспортный,
- межсетевой,
- физический и канальный.

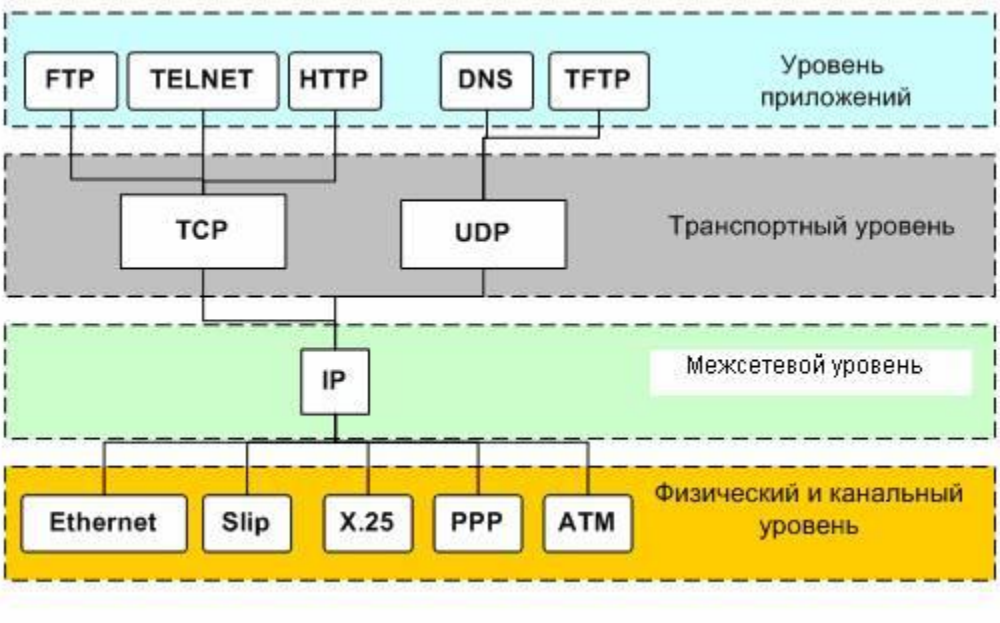


Рис. 5.5. Пример стека протоколов TCP/IP

Каким образом мы попадаем со своего компьютера на удаленный сервер?

Все наши компьютеры объединены в локальную сеть, и имеют локальную IP-адресацию. Пакеты с такой адресацией "путешествовать" в глобальной сети не смогут, т.к. маршрутизаторы их не пропустят.

Поэтому существует шлюз, который преобразовывает пакеты с локальными IP-адресами, давая им свой внешний адрес. И дальше пакеты путешествуют с адресом шлюза.

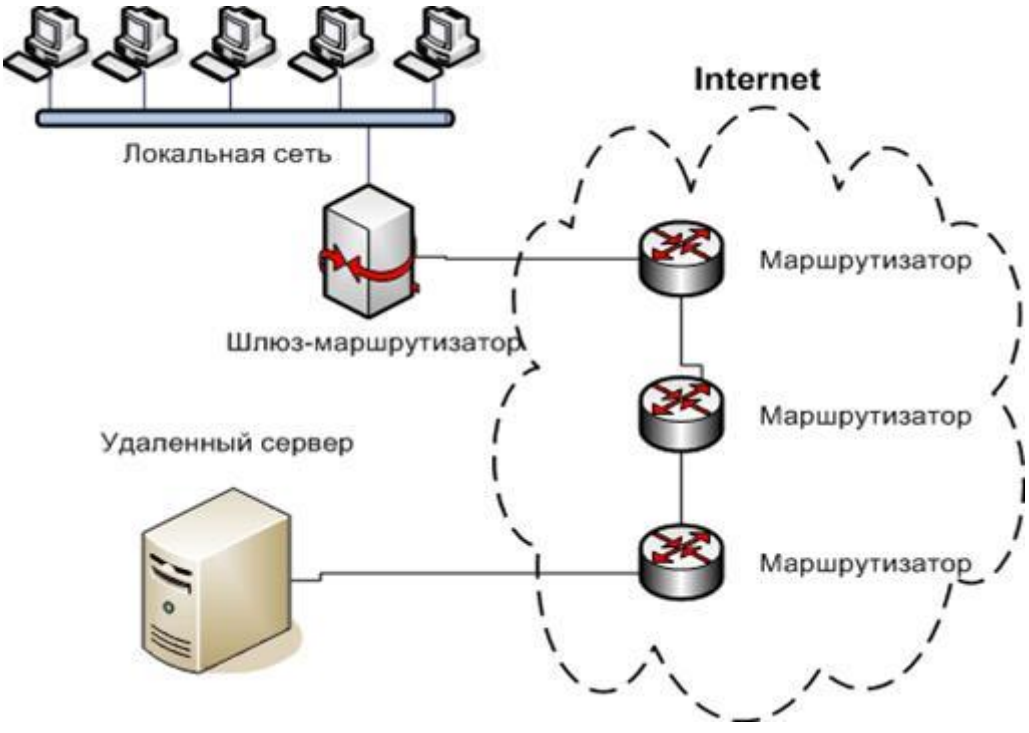


Рис. 5.6. Схема прохождения пакетов из локальной сети к серверу

Локальных сетей слишком много, поэтому реально объединяют автономные системы.

Автономная система (AS – *autonomous system*) – сеть, находящаяся под одним административным контролем, это может быть несколько компьютеров или большая сеть.

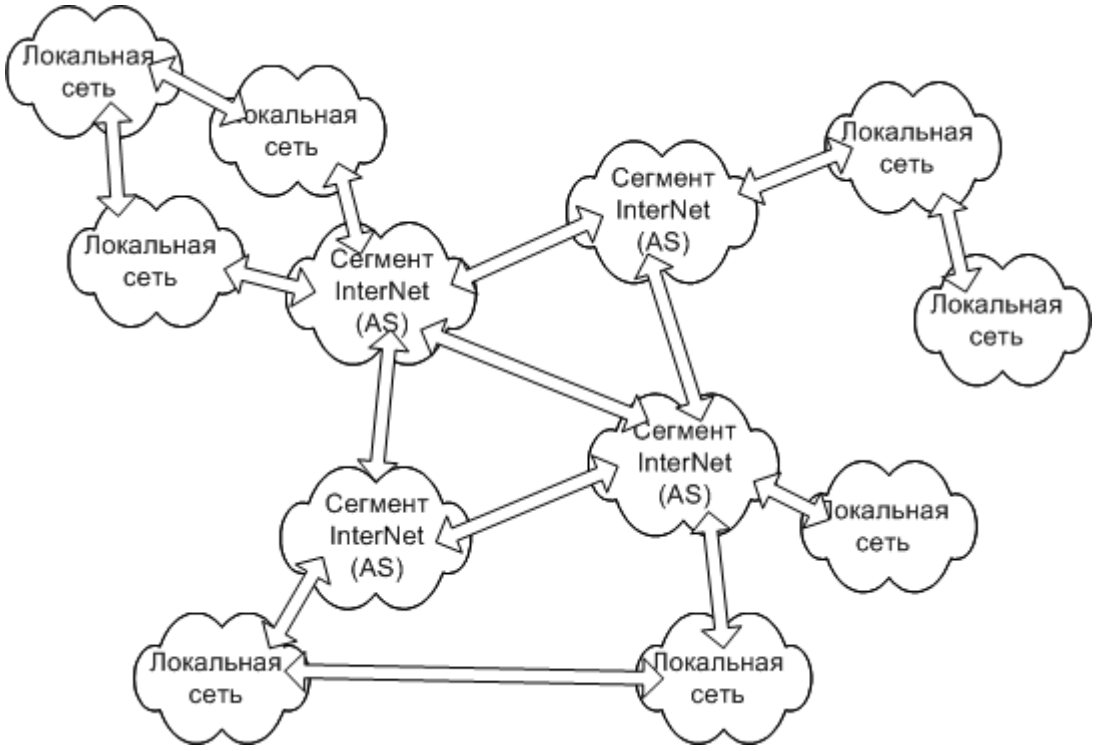


Рис. 5.7. Схема объединения отдельных сетей в общую составную сеть

5.1.3. Организации, отвечающие за развитие Интернет и стандартизацию средств Интернет

Internet Society (ISOC) – профессиональное сообщество, которое занимается общими вопросами эволюции и роста Интернет как глобальной коммуникационной инфраструктуры.

Под управлением ISOC работает – *Internet Architecture Board (IAB)* - организация, в ведении которой находится технический контроль и координация работ для Интернет. IAB координирует направление исследований и новых разработок для стека TCP/IP и является конечной инстанцией при определении новых стандартов Интернет.

В IAB входят две основные группы:

- *Internet Engineering Task Force (IETF)*. Это инженерная группа, которая занимается решением ближайших технических проблем Интернет. Именно IETF определяет спецификации, которые затем становятся стандартами Интернет;

- *Internet Research Task Force (IRTF)*. Координирует долгосрочные исследовательские проекты по протоколам TCP/IP.

Пример разработки стандартов Интернет:

- Сначала в IETF представляется *рабочий проект (draft)* в виде, доступном для комментариев. Он публикуется в Интернет для обсуждения. В него вносятся исправления. Проекту присваивается номер RFC.
- После присвоения номера проект приобретает статус *предлагаемого стандарта*. В течение 6 месяцев предлагаемый стандарт проходит проверку практикой, в результате в него вносятся изменения.
- Если результаты практических исследований показывают эффективность предлагаемого стандарта, то ему присваивается статус *проекта стандарта*. Затем в течение не менее 4-х месяцев проходят его дальнейшие испытания «на прочность», в число которых входит создание не менее двух программных реализаций.
- Если во время пребывания в ранге проекта стандарта в документ не было внесено никаких исправлений, то ему может быть присвоен статус *официального стандарта* Интернет. Список утвержденных официальных стандартов Интернет публикуется в виде документа RFC (Request for Comments) и доступен в Интернет.

Все стандарты Интернет носят название RFC с соответствующим порядковым номером, но не все RFC являются стандартами Интернет - часто эти документы представляют собой комментарии к какому-либо стандарту или просто описания некоторой проблемы Интернет. RFC-документы можно найти по адресу *http://www.rfc-editor.org/* или *http://www.ietf.org/rfc.html* .

В качестве одного из первых составителей и редакторов большой серии документов RFC был Джонатан Постел.

5.1.4. Сравнительная оценка и сфера применения сетевых архитектур ISO и TCP/IP

Главным направлением развития современных распределенных информационных систем (РИС) и информационно-вычислительных сетей (ИВС) являются их глобализация и объединение (интеграция). Это приводит к расширению РИС и ИВС, совместному использованию программного обеспечения (ПО), объединению различных систем и сетей и т.п.

В реальных сетях используется множество сетевых архитектур, таких как TCP/IP, IPX/SPX, XNS XEROX, Apple Talk, SNA, Banyan VINES, ISO, 3COM, DECnet и ряд других.

Однако наибольшее распространение получили два подхода – архитектура TCP/IP американского научно-исследовательского центра DARPA и архитектура сети на базе стандарта ISO. Принципиальные отличия этих архитектур вытекают из учёта качества используемых каналов связи. Так, архитектура TCP/IP ориентирована на применение

достаточно хороших каналов связи с низким коэффициентом ошибок (порядка 10^{-5}), в то время как архитектура ISO допускает использование каналов с вероятностью ошибки порядка 10^{-3}.

Так как основная задача ИВС общего пользования состоит в организации взаимодействия разнородных пользователей на значительных территориях, то главными требованиями к сетевой архитектуре являются:

наличие мощной, открытой и гибкой системы адресации, позволяющей обеспечить обслуживание значительного количества пользователей;

высокая эффективность передачи полезной информации в сети как по времени, как и по верности доставки;

высокая степень адаптации к изменяющимся внешним условиям (неисправности, подключение новых ресурсов или пользователей).

Стек основных протоколов сетевых архитектур ISO и TCP/IP представлен в табл. 5.1.

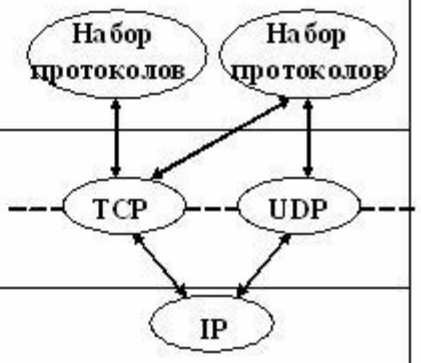
Можно выделить следующие существенные отличия данных архитектур:

Архитектура ISO предусматривает жёсткий набор протоколов на всех уровнях модели, когда на каждом уровне между взаимодействующими объектами сначала устанавливается логическая связь, а уже затем передаются данные. При этом сверху до низу сохраняется последовательность передачи протокольных единиц (блоков, фрагментов, пакетов, кадров) и предпринимаются специальные меры для сохранения целостности этих порций данных. В случае потери или искажения протокольной единицы на каждом уровне (кроме физического) осуществляются перезапрос и повторная передача искажённой протокольной единицы.

Архитектура TCP/IP предусматривает возможность ветвления протоколов и даже добавление новых. За целостностью данных следит транспортный уровень (протокол TCP) либо сам пользователь (протокол UDP).

Различия в идеологии построения сетевых архитектур порождают существенные различия механизма передачи данных на всех уровнях стандарта ISO за исключением физического и канального, где могут применяться протоколы LAP-B и X.21, но могут и другие. Основные отличия в алгоритме передачи данных состоят, во-первых, *в идеологии защиты от ошибок*, и, во-вторых, в реализации *режима коммутации пакетов* (КП).

Таблица 5.1

Уровни стандарта ISO	Стек протоколов стандарта ISO	Стек протоколов TCP/IP
7. Прикладной	Набор протоколов	
6. Представительный	X.226	
5. Сеансовый	X.225	
4. Транспортный	X.224	
3. Сетевой	X.25/3	
2. Канальный	LAP-B(X.25/2)	Произвольный
1. Физический	X.21	Произвольный

Рассмотрим сначала *методы борьбы с ошибками*.

Вопросам защиты данных от ошибок и сбоев уделено много внимания. Для этого выделяется второй (канальный) уровень. Обнаружение ошибок выполняется с помощью мощного помехоустойчивого кода типа БЧХ (Рек. V.42) с минимальным кодовым расстоянием d0=5, что позволяет обнаруживать любую 4-х кратную ошибку. Исправление ошибок выполняется с помощью алгоритмов с обратной связью – РОС-ОЖ или (чаще) РОС-НП. Для борьбы со вставками и выпадениями кадров используются тайм-аут и циклическая нумерация кадров. На сетевом уровне обеспечиваются нумерация пакетов и их перезапрос. Всё это позволяет использовать передающую среду практически любого качества, однако платой за это является высокая степень вносимой избыточности, т.е. падение реальной скорости передачи информации.

В архитектуре TCP/IP первый и второй уровни вообще не оговорены, т.е. передача может вестись даже без защиты от ошибок. Повышение верности возложено на транспортный протокол TCP. Если используются хорошие каналы, например, волоконно-оптические линии связи (ВОЛС), то на транспортном уровне используется протокол UDP, где не предусмотрена защита от ошибок. В этом случае обнаружение и исправление ошибок осуществляются на прикладном уровне специальными программами пользователя. Такой подход становится понятным, т.к. архитектура TCP/IP первоначально была реализована в сети ARPANET, где использовались выделенные высокоскоростные каналы.

Рассмотрим различия в способах *коммутации пакетов*, т.е. в реализации 3-го уровня ISO.

В архитектуре ISO за маршрутизацию (доставку пакетов по адресу) отвечает третий (сетевой) уровень (Рек. X.25). Предусматривается создание виртуальных соединений или каналов от источника до получателя, а затем по этому соединению передаются пакеты. Такой режим называется виртуальным режимом КП и по принципам напоминает традиционную коммутацию каналов (КК). В архитектуре TCP/IP реализуется другой подход, называемый дейтаграммным режимом КП. Этот режим резко упрощает задачу маршрутизации, но порождает проблему сборки сообщений из пакетов, т.к. пакеты одного сообщения могут доставляться по разным маршрутам и поступать к получателю в разное время. Дейтаграммный режим КП по принципам напоминает коммутацию сообщений (КС).

Проведём сравнения виртуального и дейтаграммного методов КП по следующим характеристикам:

- установление соединения;
- адресация;
- процедура передачи пакета по сети;
- управление входным потоком сообщений;
- эффективность использования сетевых ресурсов.

Установление соединения. При виртуальной КП до передачи сообщения устанавливается логическое соединение между взаимодействующими объектами транспортного уровня (а возможно и более высоких уровней ISO). Этот логический канал запоминается в маршрутных таблицах всех центров коммутации пакетов (ЦКП), которые участвуют в соединении. Пакеты передаются только по установленному логическому каналу, поэтому порядок их следования при этом не нарушается.

При дейтаграммной КП логического соединения не устанавливается, поэтому пакеты одного сообщения передаются по тем маршрутам, которые оптимальны в данный момент, т.е. возможно разными маршрутами. Проблема сборки сообщения из пакетов решается на транспортном уровне.

Адресация. При виртуальном режиме КП полный адрес объекта-получателя передаётся только при установлении логического соединения, т.е. с первым пакетом. Получив этот пакет, объект-получатель извещает отправителя о согласии на проведение сеанса связи (или несогласии). Создаётся логическое соединение, и передаются остальные пакеты, содержащие только номер логического канала.

При дейтаграммном режиме КП каждый передаваемый пакет обязательно должен содержать полный адрес получателя (и отправителя) и номер пакета в сообщении.

Процедура передачи пакета по сети. Виртуальный режим КП предусматривает выделение специальной базовой сети передачи данных (ПД) и передачу пакетов в этой сети ПД по готовому

логическому каналу, создаваемому по инициативе транспортного уровня.

При дейтаграммном режиме каждый пакет передаётся по разным маршрутам, что позволяет эффективнее использовать сетевые ресурсы, т.к. в больших сетях загрузка каналов меняется очень быстро, поэтому маршрут доставки желательно корректировать чаще. В данном случае можно построить глобальную сеть без выделения отдельной базовой сети ПД.

Управление входным потоком сообщений. При виртуальном режиме КП управление потоком входящих сообщений (но не пакетов) возможно лишь на входе виртуального канала, т.е. на конкретном центре коммутации пакетов для данного сообщения.

Дейтаграммный режим КП является более гибким и позволяет управлять входящим потоком сообщений практически с любого ЦКП, что улучшает гибкость управления.

Эффективность использования сетевых ресурсов. В виртуальном режиме КП оптимальный маршрут выбирается только в момент установления логического соединения, поэтому при быстром изменении ситуации на сети путь, оптимальный для первого пакета сообщения, может быть не оптимальным для последующих пакетов одного и того же сообщения.

При дейтаграммном режиме коррекция маршрута производится чаще, что позволяет более равномерно загрузить каналы всей сети и, в конечном счёте, уменьшить время доставки сообщения.

Сфера применения архитектур ISO и TCP/IP

Сфера применения архитектур ISO и TCP/IP определяется их свойствами, которые порождают основные достоинства и недостатки используемых сетевых архитектур.

Так, к основным достоинствам архитектуры ISO следует отнести:

- возможность реализации сетей даже на плохих каналах связи за счёт развитой системы защиты от ошибок и сбоев;
- возможность работать в реальном масштабе времени, простота реализации режима диалога и передачи речи в цифровой форме, поскольку задержки в доставке пакетов одного и того же сообщения незначительны;
- высокая степень стандартизации протоколов на всех уровнях, что упрощает построение ИВС заданных размеров с требуемыми показателями качества обслуживания.

Недостатки архитектуры ISO следующие:

- высокая избыточность за счёт большого объёма необходимой служебной информации;
- необходимость реализации большого набора достаточно сложных протоколов взаимодействия, причём отсутствие хотя бы одного протокола приводит к невозможности передачи данных;

- существенные трудности при организации взаимодействия различных сетей, особенно при различной сетевой архитектуре.

Рассмотрим теперь основные достоинства и недостатки архитектуры TCP/IP.

Достоинства архитектуры TCP/IP:

- небольшие затраты на реализацию протоколов взаимодействия за счёт меньшего набора требуемых протоколов;
- существенное упрощение процедуры маршрутизации, что снижает стоимость базовой сети передачи данных за счёт использования более простых ЦКП;
- возможность построения крупномасштабной ИВС с использованием разнотипного оборудования;
- возможность реализации взаимодействия различных сетей с применением простых алгоритмов согласования.

К недостаткам архитектуры TCP/IP можно отнести:

- возможность реализации только при использовании «хороших» каналов связи (желательно выделенных);
- необходимость решения проблемы сборки пакетов, которые могут поступать на транспортный уровень в произвольном порядке;
- возможность потери сообщения из-за несвоевременной доставки одного из пакетов этого сообщения;
- усложнение прикладных программ пользователя за счёт введения процедур контроля и исправления ошибок в получаемых сообщениях.

Теперь, опираясь на проведённый анализ, можно определить сферу применения сетевых архитектур.

Сетевая архитектура ISO эффективна при применении «плохих» каналов связи, необходимости работы в реальном масштабе времени и однородной структуре оборудования, причём основным выступает качество каналов связи.

При построении глобальных сетей, когда решающим фактором выступает простота согласования работы различных национальных сетей, реализуемых, как правило, на разнотипном оборудовании, наиболее эффективно применение архитектуры TCP/IP, данный вывод подтверждается практикой, т.к. в Интернет используют именно архитектуру TCP/IP.

.

5.2. Прикладной уровень. Примеры служб и протоколов

5.2.1. Служба FTP. Протокол FTP

Служба FTP предназначена для обмена файлами и построена по технологии "клиент-сервер".

Взаимодействие клиента и сервера осуществляется по протоколу FTP (*File Transfer Protocol* – протокол передачи файлов).

Клиент посылает запросы серверу, принимает и передает файлы.

Сервер обрабатывает запросы клиента, передает и принимает файлы.

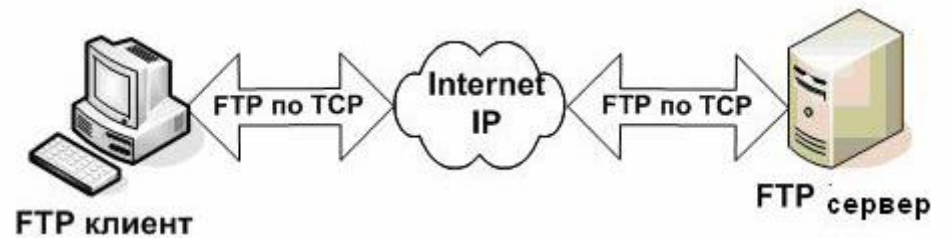


Рис. 5.8. Взаимодействие клиента и сервера по протоколу FTP

FTP-клиент – это программный интерфейс пользователя, реализующий протокол передачи файлов FTP.

Эта программа позволяет пользователю передавать файлы между двумя компьютерами, связанными между собой локальной (LAN) или глобальной (WAN) сетью. При этом компьютерные платформы могут быть различных типов.

FTP-серверы, как правило, доступны только для зарегистрированных пользователей и требуют при подключении ввода идентификатора (*login* – входное имя) и пароля (*password*).

Многие FTP-серверы открыты и для свободного доступа, их часто называют анонимными.

Для таких серверов *login* (входное имя) – *anonymous*, а в качестве пароля (*password*) рекомендуют ввести адрес своей электронной почты.

Большинство Web-браузеров обеспечивают доступ к FTP-серверам без использования специальных FTP-клиентов.

Протокол FTP используется службой FTP для передачи файлов и непосредственно взаимодействует с протоколом транспортного уровня TCP.

FTP отличается от других приложений тем, что он использует два TCP соединения для передачи файла.

Управляющее соединение – соединение для посылки команд серверу и получения ответов от него. Для организации такого соединения используется протокол Telnet. Telnet-соединение устанавливается в один шаг – посылка запроса и ожидание ответа, получение которого свидетельствует о возможности передачи команд

FTP. Канал управления существует на протяжении всей FTP-сессии и закрывается после завершения информационного обмена.

Соединение данных – соединение для передачи файлов. Передача файлов после установленного Telnet-соединения осуществляется через логическое соединение, организуемое протоколом TCP, который проверяет доступность портов, закрепленных за FTP. Канал данных формируется и ликвидируется по мере необходимости.

Протокол FTP предусматривает два возможных режима установления связи для обмена файлами:

- активный режим;
- пассивный режим.

Активный режим

Действия клиента и сервера (рис. 5.9):

1. Клиент устанавливает связь и посылает с нестандартного порта N (N>1024) запрос на 21 порт сервера;
2. Сервер посылает ответ на порт N клиента;
3. Сервер устанавливает связь для передачи данных по порту 20 на порт клиента N+1.

Пассивный режим

Действия клиента и сервера (рис5.10):

1. Клиент устанавливает связь и посылает запрос (сообщает, что надо работать в пассивном режиме) на 21 порт сервера с нестандартного порта N (N>1024);
2. Сервер назначает нестандартный порт P для канала данных (P>1024) и посылает на порт N клиента ответ, в котором сообщает номер порта P;
3. Клиент устанавливает связь для передачи данных по порту N+1 на порт сервера P.

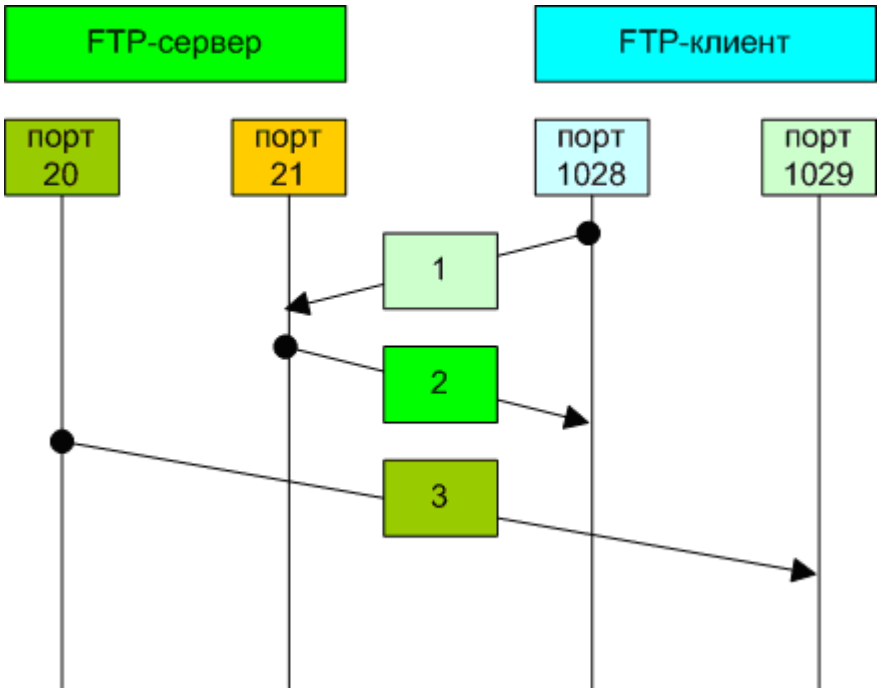


Рис. 5.9. Пример установления связи для обмена файлами в активном режиме

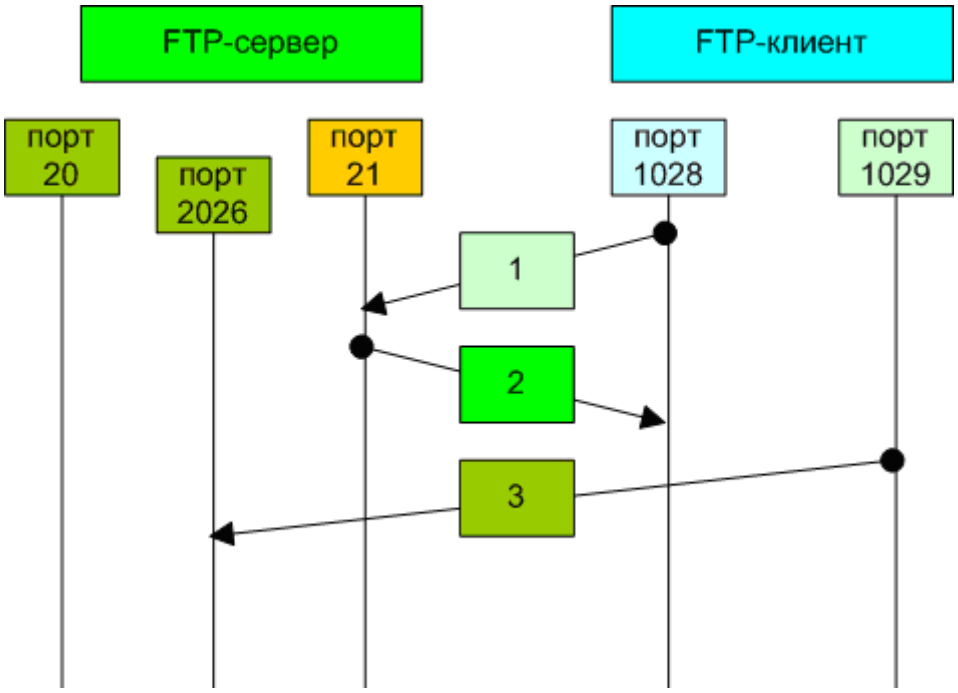


Рис. 5.10. Пример установления связи для обмена файлами в пассивном режиме

Протокол FTP определяет запрос-ответный способ взаимодействия между FTP-клиентом и FTP-сервером.

В этом и в последующих примерах команды клиента помечены буквой C, а ответы сервера – буквой S.

Пример сеанса работы с FTP-сервером

S: 220 ready, dude (vsFTPd 1.0.1: beat me, break me)
C: USER anonymous
S: 331 Please specify the password.
C: PASS emd@pds.sut.ru
S: 230 Login successful. Have fun.
C: PORT 192,168,1,50,4,81
S: 200 PORT command successful. Consider using PASV.
C: NLST
S: 150 Here comes the directory listing.
S: 226 Directory send OK.
C: PORT 192,168,1,50,4,82
S: 200 PORT command successful. Consider using PASV.
C: RETR cyc.txt
S: 150 Opening BINARY mode data connection for cyc.txt (24 bytes).
S: 226 File send OK.
C: QUIT
S: 221 Goodbye.

5.2.2. Служба WWW. Протокол HTTP

World Wide Web или просто Web – это сеть информационных ресурсов.

Служба WWW – представляет собой множество независимых, но взаимосвязанных серверов и предназначена для обмена текстовой, графической, аудио и видеоинформацией. Работая с Web, пользователь последовательно соединяется с Web-серверами и получает информацию.

WWW построена по схеме *"клиент-сервер"* (рис. 5.11).

В качестве *клиента* выступает *браузер*, который является также и интерпретатором языка гипертекстовой разметки документов HTML *(HyperText Markup Language).* Как интерпретатор, браузер в зависимости от команд (тегов) выполняет различные функции: размещение текста на экране, обмен информацией с сервером по мере анализа полученного HTML-текста и др.

Сервер обрабатывает запросы клиента на получение файлов, выполнение программ и др.

Для работы с WEB-ресурсами используется протокол обмена гипертекстовой информацией HTTP *(HyperText Transfer Protocol).* Для взаимодействия WWW-сервера с другими программами, установленными на сервере (например, СУБД) создан универсальный интерфейс шлюзов CGI *(Common Gateway Interface).*

Транспортным протоколом для HTTP является протокол TCP, причем WWW-сервер (Web-сервер) находится в состоянии ожидания соединения со стороны клиента стандартно по порту 80 TCP, а клиент HTTP (браузер Web) является инициатором соединения.

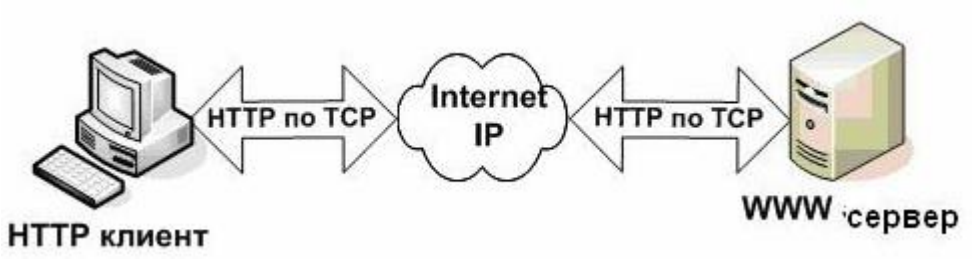


Рис. 5.11. Взаимодействие клиента и сервера по протоколу HTTP

URL *(Universal Resource Locator*) – унифицированный указатель ресурсов – способ адресации ресурсов в сети. Каждый ресурс имеет уникальный для Web адрес, называемый унифицированным (универсальным) идентификатором ресурса (URI – *Universal Resource Identifier*). Например: *http://www.sut.ru* .

5.2.3. Структура и протоколы электронной почты в Интернет

Основным руководящим документом для электронной почты в Интернет является *RFC 2821* (Klensin J., Ed. Simple Mail Transfer Protocol. RFC 2821, April 2001).

В нем рассмотрены:

- протокол SMTP (*Simple Mail Transfer Protocol* – простой протокол электронной почты), используемый для доставки почтовых сообщений от почтовой программы отправителя до электронного почтового ящика получателя;
- основные принципы построения и функционирования электронной почты.

Справка. *Первое электронное письмо было отправлено в 1971 году Реем Томлинсоном – автором программы для обмена сообщениями между компьютерами. Он же предложил использовать значок @ для разделения имени пользователя и компьютера.*

Структура электронного сообщения

В настоящее время для электронных сообщений используется стандарт *RFC 2822* (Resnick P., Ed. Internet Message Format. RFC 2822, April 2001). Сообщение, передаваемое по электронной почте, состоит из трех частей:

- конверт (envelope);
- заголовок (header);
- тело (body).

Сообщение доставляется получателю в виде заголовка и тела.

Заголовок состоит из полей: текстовых строк, состоящих из имени поля, двоеточия и содержимого поля.

В заголовке допускается использование только символов в кодировке ASCII.

В табл. 5.2 описаны некоторые поля заголовка.

Таблица 5.2

Название поля	Значение поля
Date:	Время отправки сообщения
From:	Адрес отправителя
Reply - To:	Адрес для ответа
To:	Адреса получателей
Cc:	Адреса получателей копий
Bcc:	Адреса получателей скрытых копий. Это поле используется в процессе передачи сообщения, при доставке получателю соответствующие поля или часть их содержимого могут быть удалены.
Message - ID:	Уникальный идентификатор сообщения
In-Reply-To:	Уникальный идентификатор сообщения, на которое отвечает данное сообщение
References:	Уникальные идентификаторы всех сообщений в цепочке ответов
Subject:	Тема сообщения
Return - Path:	Адрес отправителя, указанный на конверте сообщения
Received:	Информация о прохождении сообщения. Каждый узел, через который прошло сообщение, должен добавить в заголовок поле " Received :", содержащее имена и адреса *IP* узлов, пославших и принявших сообщение, время прохождения и пр.
MIME - Version:	Используемая версия *Multipurpose Internet Mail Extensions* – многоцелевые расширения электронной почты в Internet – MIME
Content - type:	Тип данных, используемых в теле сообщения
Content-Transfer-Encoding:	Способ кодирования символов не US - ASCII , используемый в тексте сообщения

Тело сообщения, если это не просто текст, записанный латинскими буквами, должно быть закодировано в соответствии со спецификацией MIME, как описано в *RFC 2045* (Freed N., Borenstein N. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045, November 1996). На приемной стороне тело при необходимости декодируется и преобразуется в понятный пользователю вид.

Пример электронного письма

X-AntiVirus: Checked by Dr.Web (http://www.drweb.net)
Return-Path:
Received: from camay.yandex.ru (camay.yandex.ru [213.180.200.33]) by pds.sut.ru (8.12.2/8.12.2/SuSE Linux 0.6) with ESMTP id iA8GKO2Z011039; Mon, 8 Nov 2004 19:20:24 +0300
Received: from YAMAIL (camay.yandex.ru) by mail.yandex.ru id; Mon, 8 Nov 2004 19:15:41 +0300
Date: Mon, 8 Nov 2004 19:15:41 +0300 (MSK)
From: "doronin2004"
Reply-To: doronin2004@yandex.ru
Sender: doronin2004@yandex.ru
Message-Id: <418F9BAD.00001A.28843@camay.yandex.ru>
MIME-Version: 1.0
X-Mailer: Yamail [http://yandex.ru]
Errors-To: doronin2004@yandex.ru
To: emd@pds.sut.ru
Cc: bor@pds.sut.ru
Subject: E-mail
X-source-ip: 213.221.51.66
Content-Type: text/plain; charset="KOI8-R"
Content-Transfer-Encoding: 8bit

Proverka
Проверка

Адреса электронной почты в Интернет

Электронная почта в Internet использует маршрутно-независимую адресацию. Формат электронного адреса:

имя_пользователя@почтовый_домен

где *имя_пользователя* – идентификатор пользователя, уникальный в пределах одного почтового домена;

@ (коммерческое at) – символ-разделитель;

почтовый_домен – уникальный идентификатор почтовой системы.

Имя пользователя может состоять из цифр, латинских букв и символов

! # $ % & " * + - / = ? ^ _ ` { | } ~

Оно может состоять из нескольких полей, разделенных точкой, которая интерпретируется как часть имени пользователя.

Имя почтового домена имеет тот же формат, какой используется в доменных именах Internet. Формат описан в *RFC 1034* (Mockapetris P. Domain names - concepts and facilities. RFC 1034, November 1987).

Адрес может содержать комментарии в виде произвольных текстовых строк до и после значимой части. В этом случае значимую часть адреса заключают в угловые скобки.

комментарий < имя_пользователя@почтовый_домен > комментарий

Например:

Леонид Свердлов <lonk@lonk.pp.ru> (каф. ОПДС)

Для маршрутизации электронной почты в Интернет, как и для установления соответствия между доменными именами узлов сети и их адресами *IP*, используется система *Domain Name System* – система доменных имен – *DNS*. Получив сообщение, предназначенное для отправки, почтовый сервер посылает запрос DNS с указанием имени почтового домена получателя. В ответ почтовый сервер получает список узлов, принимающих почту для данного домена. Список представляется в виде записей MX (Mail eXchange). Одному имени почтового домена могут соответствовать несколько записей MX с различными приоритетами. Приоритеты обозначаются целыми числами, с их помощью определяется, в каком порядке почтовому серверу следует обращаться к узлам, принимающим почту для данного домена. Меньшему числу соответствует больший приоритет.

Структура электронной почты в Интернет

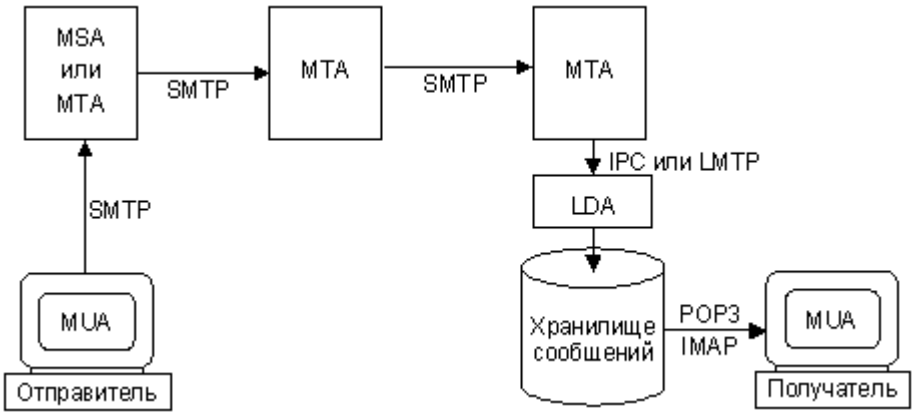


Рис. 5.12. Структура электронной почты в Интернет:

- MUA (Mail User Agent) – пользовательский агент, или клиентская почтовая программа;
- MTA (Mail Transfer Agent) – транспортный агент, или почтовый сервер;
- LDA (Local Delivery Agent) – агент локальной доставки;
- MSA (Message Submission Agent) – агент подачи сообщения.

Довольно большое распространение получили агенты пользователя, использующие интерфейс CGI для доступа оконечного пользователя к его почтовому ящику по протоколу HTTP или более безопасному HTTPS при помощи Web-браузера. Такую реализацию MUA часто называют web-mail.

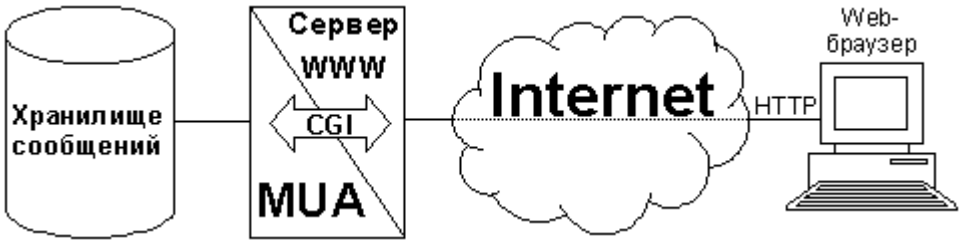


Рис. 5.13. Структура web-mail

Доставка почтового сообщения

Рассмотрим путь почтового сообщения на примере, показанном на рис. 5.14. Порядок следования отдельных событий обозначен числами на стрелках.

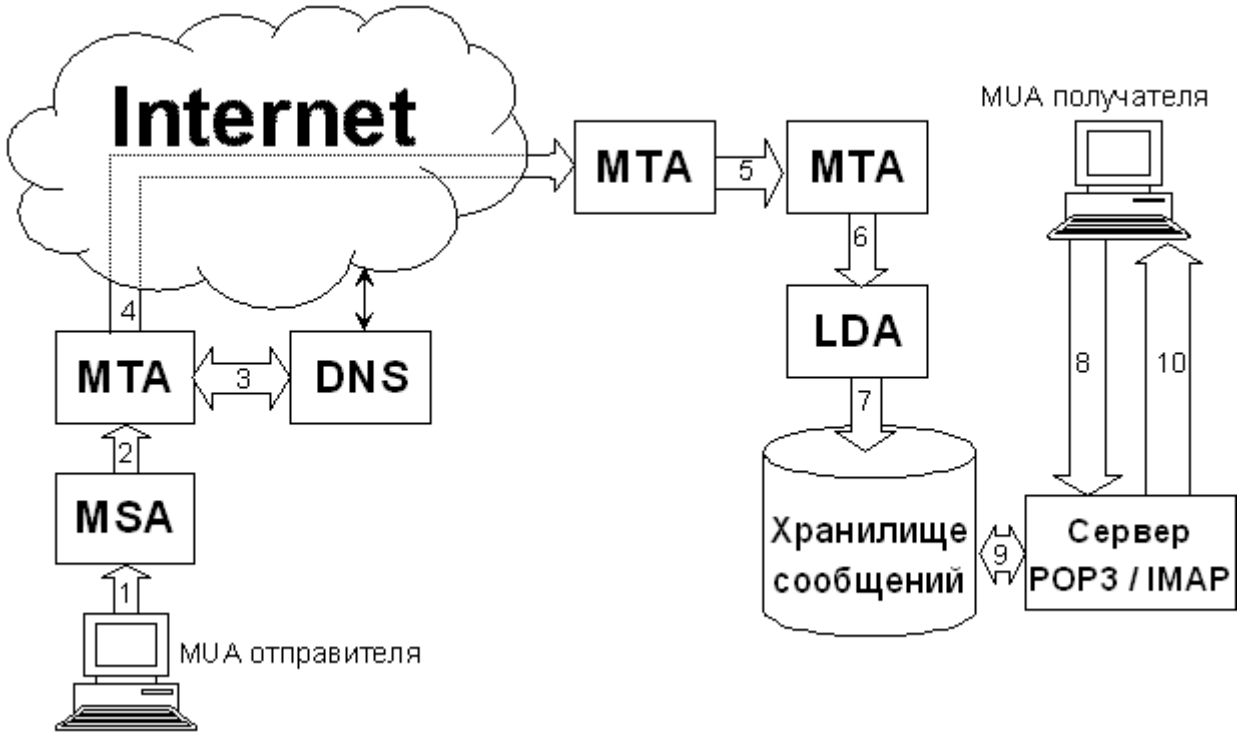


Рис. 5.14. Процесс доставки электронного сообщения от отправителя к получателю

1. Сообщение, сформированное MUA отправителя, по протоколу SMTP посылается MSA. MSA проверяет, имеет ли данный MUA или пользователь право посылать почту из этой почтовой системы. В случае положительного результата, сообщение принимается для дальнейшей доставки.
2. MSA проверяет заголовок сообщения и, при необходимости, исправляет его. Готовое к отправке сообщение по протоколу SMTP отправляется на MTA исходящей почты.
3. MTA исходящей почты анализирует адрес получателя. Если сообщение предназначено для получателя домена, обслуживаемого данной почтовой системой, то оно доставляется получателю (см. пункты 6 – 10), в противном случае MTA запрашивает информацию о почтовом домене, указанном в адресе получателя, сервер DNS.

Получив запрашиваемые данные, сервер DNS сообщает MTA, какие узлы принимают почту для данного домена, их адреса IP и приоритеты.
4. MTA отправителя пытается установить соединение по протоколу с принимающими почту узлами в соответствии с приоритетами, указанными в записях MX, полученных от сервера DNS. Если соединение ни с одним узлом не удается установить, сообщение помещается в очередь, и через некоторое время попытки установить соединение повторяются. Если соединение установлено, то принимающий MTA, удостоверившись, что сообщение предназначено для пользователя его домена, и что почтовый ящик с указанным адресом действительно существует, принимает сообщение.
5. В принимающей почтовой системе сообщение может пройти через несколько промежуточных MTA, выполняющих различные виды обработки входящей почты: проверку на вирусы, фильтрацию спама, перенаправление к нужному хранилищу сообщений и пр.
6. Последний MTA передает сообщение LDA для локальной доставки.
7. LDA помещает сообщение в почтовый ящик адресата.
8. Получатель обращается к серверу POP3 или IMAP, чтобы проверить поступившую почту.
9. Сервер забирает сообщение из почтового ящика.
10. Сервер посылает сообщение пользовательскому агенту получателя.

Таким образом сообщение доставляется от отправителя к получателю.

Протокол SMTP

Про*стой протокол передачи почты – Simple Mail Transfer Protocol* (SMTP) обычно используется на участке от MUA отправителя до ближайшего к получателю MTA. Последняя версия SMTP протокола опубликована в документе – *RFC 2821* (Simple Mail Transfer Protocol J. Klensin, Ed. April 2001).

SMTP может работать с различными протоколами транспортного уровня, но обычно используется TCP. За SMTP закреплен порт TCP 25.

Почта по протоколу SMTP посылается от клиента к серверу. Клиент запрашивает соединение с сервером. После успешного установления соединения сервер сообщает клиенту свое доменное имя. Он также может сообщить тип и версию установленного программного обеспечения. Однако, из соображений безопасности передача этой информации часто блокируется системными администраторами.

Ответ сервера, свидетельствующий о готовности к приему команд клиента, служит сигналом к началу диалога, в котором клиент последовательно посылает серверу команды и ожидает ответы, либо подтверждающие исполнение команд, либо сообщающих о

невозможности исполнения, либо содержащих информацию, запрошенную клиентом.

Рассмотрим пример диалога по протоколу SMTP.

S: 220 pds.sut.ru ESMTP Sendmail 8.12.2/8.12.2/SuSE Linux 0.6; Tue, 19 Oct 2004 20:50:15 +0400 *Сервер представляется как pds.sut.ru и сообщает о готовности к приему команд*
C: helo user *Клиент представляется как user*
S: 250 pds.sut.ru Hello p.pds.sut.ru [192.168.1.7], pleased to mee *Сервер сообщает, что команда выполнена успешно*
C: mail from:emd@pds.sut.ru *Адрес отправителя: emd@pds.sut.ru*
S: 250 2.1.0 emd@pds.sut.ru... Sender ok *Сервер сообщает, что команда выполнена успешно*
C: rcpt to:doronin@yandex.ru *Адрес получателя: doronin@yandex.ru*
S: 250 2.1.5 doronin@yandex.ru... Recipient ok *Сервер сообщает, что команда выполнена успешно*
C: data *Клиент готов передавать сообщение*
S: 354 Enter mail, end with "." on a line by itself *Сервер готов к приему сообщения*
C: Proverka *Клиент передает сообщение*
C: Проверка *Клиент передает сообщение*
. *Сообщение заканчивается строкой, состоящей из одной точки*
S: 250 2.0.0 i9JHEFGu031961 Message accepted for delivery *Сообщение принято*
C: quit *Клиент завершает связь*
S: 221 2.0.0 pds.sut.ru closing connection *Сервер подтверждает завершение связи*

Протокол POP

Протокол POP *(Post Office Protocol)* был разработан в 1984 году, в 1985 году появилась вторая его версия, в 1988 году – третья, которая с существенными модификациями, сделанными в 1991, 1993, 1994 и 1996 годах, используется в настоящее время. Последняя модификация протокола POP3 описана в **RFC 1939** (Myers J., Rose M. Post Office Protocol - Version 3. RFC 1939, May 1996).

Протокол почтового отделения, версия 3 (POP3) предназначен для получения сообщений, находящихся в почтовом ящике пользователя на удаленном сервере электронной почты.

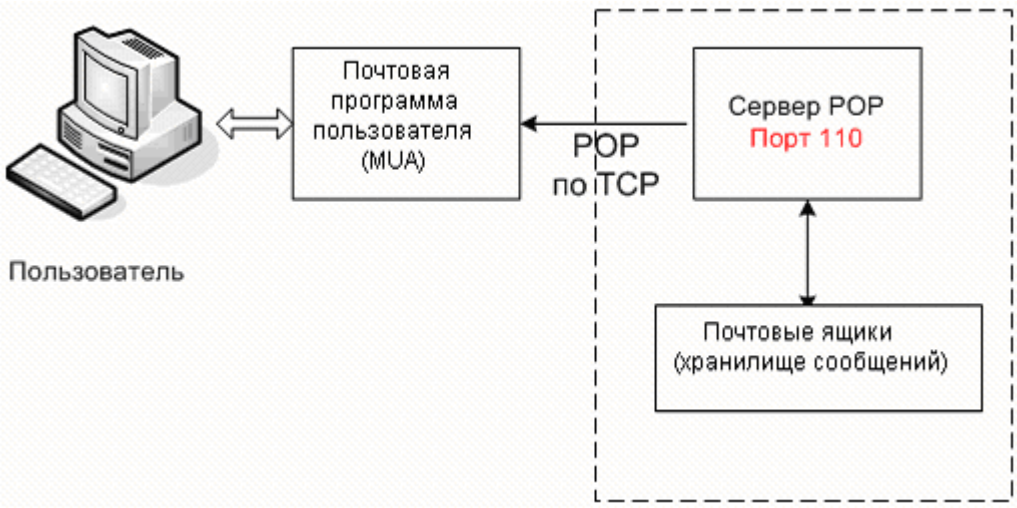


Рис. 5.15. Модель протокола POP

В качестве клиента POP3 выступает MUA пользователя, а сервер должен иметь доступ к хранилищу сообщений. Информация по протоколу POP3 передается от сервера к клиенту.
Порт по умолчанию – 110.

Пользователь может получить доступ к POP-серверу из любой точки доступа к Интернет.

Процесс получения почты по протоколу POP3 состоит из трех этапов (состояний):

- авторизация;
- транзакция;
- обновление (завершение транзакции).

Рис. 5.16. Состояния сеанса POP3

В ходе сеанса клиент посылает серверу команды, а сервер сообщает о результате выполнения каждой из них.

Каждая команда POP3 состоит из ключевого слова и, возможно, из аргументов, разделенных пробелами. Ключевые слова состоят из трех или четырех букв, передаваемых независимо от регистра. Аргументы могут содержать только символы ASCII . Каждый аргумент может состоять не более чем из сорока символов.

Ответ сервера может иметь два значения:

+OK - успешное завершение;

–ERR - неуспешное завершение.

Пример сеанса POP3.

S: + OK POP Ya! v1.0na *Приветствие сервера*
C: user doronin2004 *Имя пользователя: doronin2004*
S: +OK password, please *Имя принято, ожидается ввод пароля*
C: pass educate *Пароль пользователя educate*
S: +OK 2 message(s) 2443 bytes *Доступ к почтовому ящику разрешен. Имеется 2 сообщения общим объемом 2443 октета*
C: list *Запрос списка сообщений*
S: +OK 2 2443 *2 сообщения, 2443 октета*
S: 1 1079 *Размер первого сообщения: 1079 октетов*
S: 2 1364 *Размер второго сообщения: 1364 октета*
S: . *Конец списка*
C: retr 1 *Запрос первого сообщения*
S: +OK
S: X-AntiVirus: Checked by Dr.Web (http://www.drweb.net) *Передается заголовок сообщения*
S: Received: from bingo.yandex.ru ([213.180.200.1]:24968 "EHLO bingo.yandex.ru" smtp-auth: <none>) by mail.yandex.ru with ESMTP id <S998959AbUJSQ4B>; Tue, 19 Oct 2004 20:56:01 +0400
S: Received: from pds.sut.ru ([195.19.219.136]:3202 "EHLO pds.sut.ru" smtp-auth: <none> TLS-CIPHER: "EDH-RSA-DES-CBC3-SHA keybits 168/168 version TLSv1/SSLv3" TLS-PEER-CN1: <none>) by mail.yandex.ru with ESMTP id S862337AbUJSQ4B (ORCPT <rfc822;doronin2004@yandex.ru>); Tue, 19 Oct 2004 20:56:01 +0400
S: Received-SPF: none (bingo.yandex.ru: 195.19.219.136 is neither permitted nor den ied by domain of pds.sut.ru) client-ip=195.19.219.136; envelope-from=emd@pds.sut.ru; helo=pds.sut.ru;
S: Received: from user (p.pds.sut.ru [192.168.1.7]) by pds.sut.ru (8.12.2/8.12.2/SuSE Linux 0.6) with SMTP id i9JGthGu020548 for doronin2004@yandex.ru; Tue, 19 Oct 2004 20:58:12 +0400
S: Date: Tue, 19 Oct 2004 20:55:43 +0400
S: From: Evgeny Doronin <emd@pds.sut.ru>
S: S: Message-Id: <200410191658.i9JGthGu020548@pds.sut.ru>
S: X-AntiVirus: Checked by Dr.Web (http://www.drweb.net)
S: To: undisclosed-recipients:;

S: Proverka *Передается полный текст сообщения*
S: Проверка

S: .
C: dele 1 *Удалить первое сообщение*
S: +OK done. *Сообщение удалено (на самом деле только помечено для удаления)*

C: quit *Конец работы. Будут удалены все помеченные для удаления сообщения*
S: +OK shutting down.

Протокол IMAP

Протокол IMAP4 *(Internet Message Access Protocol)* позволяет клиентам получать доступ и манипулировать сообщениями электронной почты на сервере. Был разработан для замены POP3.

Порт по умолчанию – 143.

Первый принятый стандарт – *RFC 1730* (J. Myers December 1994).

Последний принятый стандарт – *RFC 3501* (Crispin M. INTERNET MESSAGE ACCESS PROTOCOL – VERSION 4rev1. RFC 3501, March 2003).

Протокол позволяет работать с несколькими почтовыми ящиками на одном или нескольких серверах IMAP как с файлами и каталогами на собственной машине пользователя. Сервер IMAP способен анализировать сообщение: выделять заданные поля заголовка и разбирать структуру тела сообщения. Несколько клиентов могут одновременно работать с одним и тем же почтовым ящиком.

5.3. Транспортный уровень. Протоколы TCP и UDP

На транспортном уровне используется два основных транспортных протокола UDP – *User Datagram Protocol*, *RFC 768* (протокол пользовательских дейтаграмм) и TCP – *Transmission Control Protocol*, *RFC 793* (протокол управления передачей). Эти протоколы предоставляют разные услуги прикладным процессам, причём большинство хостов активно используют только один из них.

Если требуется надёжная и эффективная доставка сообщений по протяженному и ненадёжному каналу ПД, то применяется протокол TCP. Если требуется передавать сообщения на высокоскоростных сетях с короткими соединениями, то лучше подходит протокол UDP.

Протокол UDP

Протокол UDP обеспечивает только доставку дейтаграммы и не гарантирует её выполнение. При обнаружении ошибки дейтаграмма просто стирается. Протокол не поддерживает виртуального соединения с удалённым модулем UDP. Чаще всего базируется на принципах динамической маршрутизации (каждая дейтаграмма передаётся по оптимальному маршруту). Основное достоинство – простота.

Формат UDP-дейтаграммы имеет следующий вид:

Видно, что формат протокола UDP размещается в поле данных IP-пакета (или после заголовка IP-пакета) и содержит следующие поля:

Поле «Порт источника» *(Source Port)* указывает порт процесса источника, куда может быть адресован ответ на данное сообщение.

Поле «Порт получателя» *(Destination Port)* идентифицирует принимающий процесс.

Под «портом» понимается адрес (номер) некой точки доступа к услугам другого уровня. В случае архитектуры TCP/IP под портом понимается некий номер области памяти, где размещаются передаваемые в сеть (протоколу UDP или TCP) и принимаемые из сети (поступающие в распоряжение операционной системы) данные. Номера портов на передачу и приём в общем случае могут различаться. На приёмной и передающей сторонах взаимодействие процессов в общем случае может происходить через разные номера

портов, поэтому указание порта в заголовке UDP-дейтаграммы необходимо.

В поле «Длина» *(Length)* указывается размер данной дейтаграммы с учётом длины заголовка в байтах.

Поле «Контрольная сумма» *(Checksum)* обеспечивает контроль правильности данных и заголовка. Суммируются все контролируемые 16-битные слова (с циклическим переносом из старшего разряда в младший). Инвертированное значение результата записывается в поле контрольной суммы. Если UDP-дейтаграмма содержит нечетное число байтов, то недостающий последний байт в таких случаях считается нулевым. Этот байт не передается в области данных.

При подсчете контрольной суммы протокол UDP учитывает 12-байтовый псевдозаголовок *(pseudo header).* Псевдозаголовок включает в себя: *IP*-адрес источника, *IP*-адрес приемника, протокол (код 17) и длину UDP-дейтаграммы.

Если источник проставил контрольную сумму, а адресат при ее проверке обнаружил ошибку, то UDP-дейтаграмма "молчаливо отбрасывается" - не генерируется никакого сообщения об ошибке.

UDP-данные могут отсутствовать.

Прикладные процессы и модули UDP взаимодействуют через UDP-порты, которые нумеруются, начиная с 0. Прикладной процесс ожидает сообщение в порт, специально выделенный для этих услуг. Номер этого порта является общеизвестным и определяется стандартами сети Интернет.

Протокол TCP

Протокол TCP ориентирован на создание виртуальных соединений. Он размещается над сетевым протоколом IP, который даёт возможность TCP посылать и принимать сегменты информации различной длины, вложенные в межсетевые дейтаграммные «конверты» (пакеты).

При организации связи между парой прикладных процессов протокол TCP обеспечивает следующее: надёжную передачу данных; управление потоком данных; мультиплексирование; организацию, поддержание и сброс виртуального соединения (виртуального канала); приоритетную доставку информации и её безопасность.

Подобно модулю UDP, прикладные процессы взаимодействуют с модулем TCP через порты, которые имеют общеизвестные адреса (номера).

Когда прикладной процесс начинает использовать TCP, то этот модуль на клиенте и модуль TCP на сервере приложений начинают взаимодействовать. Эти два оконечных модуля, прежде всего, создают виртуальное соединение, которое является дуплексным и расходует ресурсы обоих оконечных модулей TCP. Протокол TCP разбивает поток двоичных разрядов (поступающих с вышележащего уровня) на

TCP-сегменты, которые передаются по виртуальному соединению. На приёмном конце производится обратная операция.

Протокол TCP требует, чтобы все отправленные сегменты данных были подтверждены с приёмного конца, т.е. используется алгоритм обратной связи. Для повышения эффективности работы используются механизм скользящего окна, тайм-ауты и повторные передачи для обеспечения надёжной доставки. Каждая из принимающих сторон может управлять потоком данных от передающего модуля, чем предотвращается возможность переполнения буферов приёмников. Пользователь при установлении соединения может устанавливать категорию срочности и безопасности. Эти признаки учитываются не только при работе с TCP-сегментами, но и дублируются в поле «Тип сервиса» IP-пакета.

Формат TCP-сегмента включает заголовок и данные и имеет следующий вид:

Поля «Порт источника» и «Порт получателя» указывают номера портов в TCP-модулях (идентифицируют взаимодействующие приложения).

В поле «Последовательный номер (Позиционный номер)» содержится номер, который указывает место в потоке данных от источника до конечного получателя первого байта содержащихся в этом сегменте данных.

В начальном сегменте, посылаемом при установлении соединения, присутствует флаг *SYN*, а в поле «Позиционный номер» содержится так называемый *начальный позиционный номер ISN (initial sequence number)*, выбранный данным хостом для этого нового соединения. Первому байту данных, переданному хостом по новому соединению, будет присвоен позиционный номер, равный *ISN*+1.

Поле «Номер подтверждения (Квитанция)» при установленном флаге АСК содержит значение последовательного номера, который отправитель данного сегмента собирается принимать. (То есть этот номер всегда на единицу больше номера последнего успешно принятого байта). После установления виртуального соединения это

поле обязательно заполняется. С помощью этого поля отмечается байт, с которого начнется окно приёма данных от источника (механизм скользящего окна).

TCP предоставляет приложениям сервис *полнодуплексной передачи.* Это означает, что данные могут передаваться в обоих направлениях независимыми потоками. В процессе двусторонней передачи каждая из сторон должна вести учет позиционных номеров передаваемых и принимаемых ею байтов.

Поле «Смещение данных (Размер заголовка)» определяет число 32-разрядных слов в заголовке TCP-сегмента, так как такое же смещение есть в межсетевом заголовке.

В поле «Резерв» все разряды устанавливаются равными 0.

Содержимое «Поля управляющих флагов»: URG *(urgency)* – указатель срочности; АСК *(acknowledgement)* – подтверждение; PSH *(push)* – указатель немедленной выдачи на верхний уровень; RST *(reset)* – немедленный сброс соединения; SYN *(synchronization)* – синхронизация последовательных номеров; FIN *(final)* – завершение соединения.

Поле «Окно» содержит число байт, равное длине окна, т.е. период времени в байтах, когда отправитель ожидает информацию от приёмника.

Поле «Проверочная сумма». Проверочная сумма подсчитывается для всего TCP сегмента, при этом определяется 16-битное дополнение суммы всех 16-битных слов в заголовке и в поле данных. Если сегмент содержит нечетное количество байтов, то он будет дополнен нулями справа до образования 16-битного слова. Этот выравнивающий байт не передается с сегментом по сети, так как может быть "восстановлен" получателем.

Проверочная сумма учитывает также 96-битный псевдозаголовок, который ставится перед заголовком протокола TCP. Подзаголовок включает следующие поля из заголовка протокола IP: IP-адреса отправителя и получателя, протокол (код 6) и длину сегмента.

С помощью добавления псевдозаголовка протокол TCP защищает самого себя от ошибочной доставки протоколов IP. Так, если протокол IP доставляет сегмент, не предназначенный данному работающему приложению, то модуль протокола TCP на принимающей стороне обнаружит некорректность доставки.

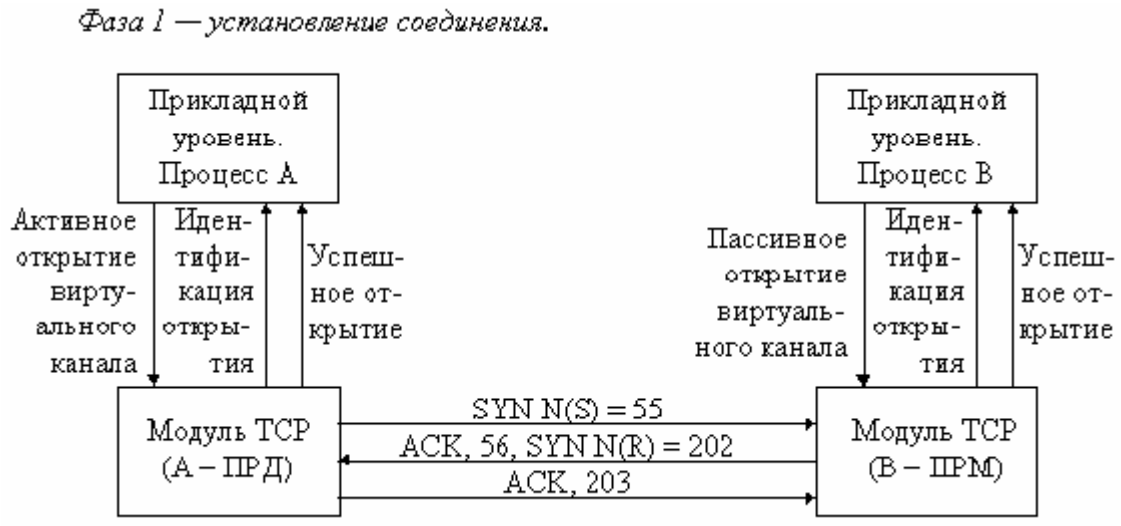
Поле «Указатель срочности (Указатель границы срочных данных)» указывает последовательный номер байта, которым заканчиваются срочные данные. Если установлен флаг URG и это поле равно 0, то весь сегмент считается срочным.

Поле «Модификаторы (Опции)» указывает на дополнительные услуги и может иметь переменную длину, кратную байту.

Поле «Заполнитель» имеет переменную длину и дополняет заголовок до целого числа 32-разрядных слов. Поле заполняется нулями.

Рассмотрим типовой диалог между двумя объектами прикладного уровня с использованием протокола TCP.

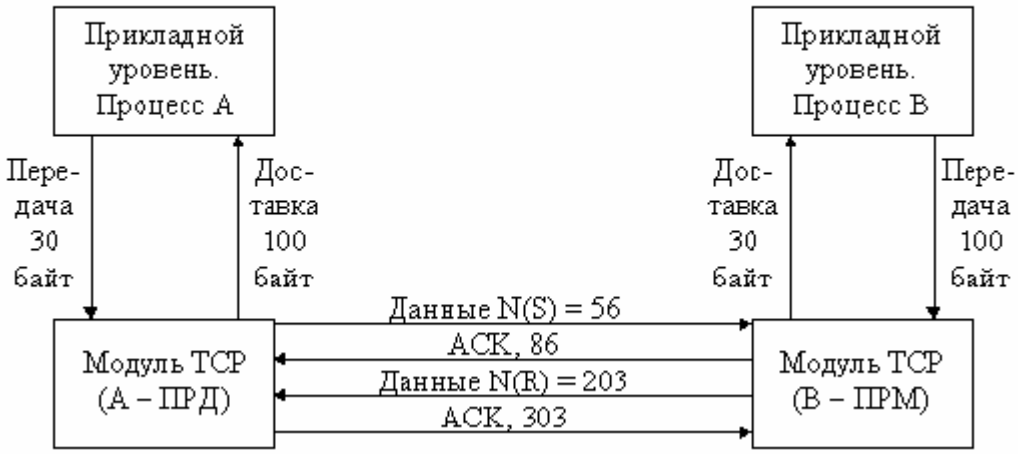
Для открытия виртуального соединения посылается флаг *SYN* в сегменте, с которого начнется передача (N(S)=55). Приёмник отвечает сегменту, в котором флаг *ACK* установлен в 1 и указывает номер байта, с которого он начнёт передавать (N(R)=202). В заголовке этого же сегмента в поле «Номер подтверждения» приёмник указывает, что он ожидает от передатчика байт с номером 56. Здесь же передаётся флаг синхронизации *SYN*. Передатчик (модуль А), получив этот сегмент с подтверждением о готовности приёмника работать, также отвечает сегментом с подтверждением *ACK*, и в поле «Номер подтверждения» передатчик указывает, что он ожидает от приёмника байт с номером 203.



N(S) - номер байта, с которого начнёт передавать передатчик (ПРД), например, 55;
N(R) - номер байта, с которого будет передавать приёмник (ПРМ), например, 202

После этого виртуальное соединение установлено, о чем модули TCP извещают свои прикладные процессы.

Фаза 2 — передача данных.



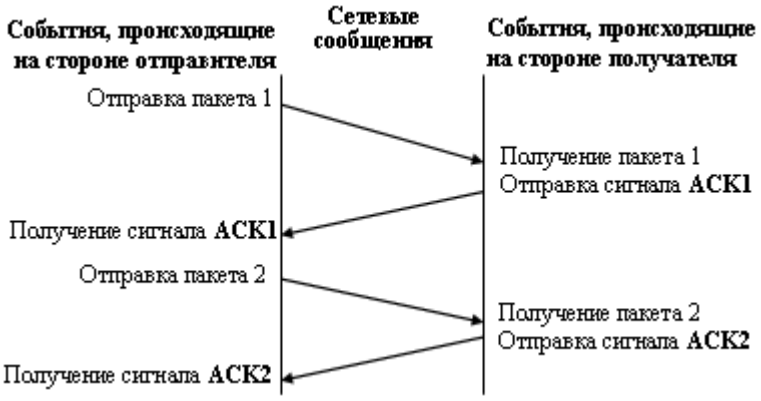
Передатчик по созданному виртуальному соединению передаёт данные (30 байт), начиная с байта под номером 56. Приёмник ожидает байт данных именно с этим номером, поэтому после приёма данных приёмник выдаёт сегмент с флагом подтверждения АСК и номером следующего ожидаемого байта N(S)=56+30=86, кроме того, приёмник отсылает в сторону передатчика 100 байт данных, начиная с номера 203, что и ожидает передатчик. Получив 100 байт от приёмника, передатчик выдаёт сегмент с флагом АСК и номером следующего ожидаемого байта N(R)=203+100=303.

В фазе "передача данных" работают механизмы обеспечения надёжной доставки.

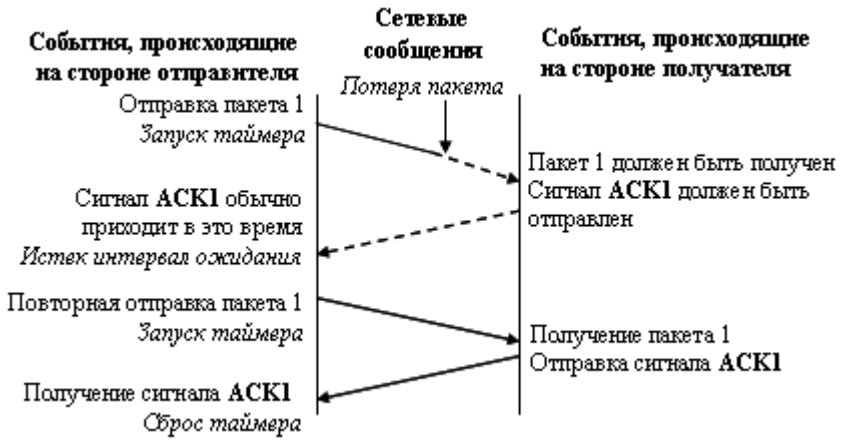
Различают три варианта обратной связи:

- РОС ОЖ
- Оконный режим переспроса
- Адресный режим переспроса

1. *РОС ОЖ.*



На каждый передающийся сегмент ожидается получение квитанции (ACK=1 и номер следующего запрашиваемого октета). Включается таймер ожидания. Если квитанция не приходит до истечения таймера, то осуществляется повторная передача.



При потере пакета через определенный интервал времени выполняется его повторная передача. Пунктирной линией показан процесс нормальной передачи пакета и получения подтверждения.

Причинами потери пакетов могут являться ошибки, возникающие в заголовке IP, истечение времени жизни, переполнение буфера маршрутизатора и т. п. Во всех этих случаях отправляется сообщение ICMP. Время ожидания квитанции зависит от расстояния до получателя (от времени двойного пробега).

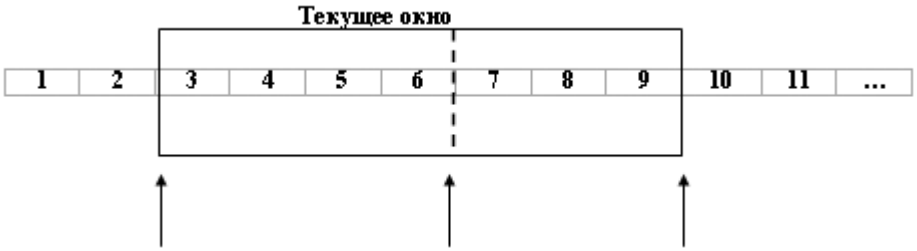
Существует 2 механизма определения времени двойного пробега:

1. Тестирование сети и оценка в режиме «пингования» временной задержки на каждом сегменте. Затем осуществляется набор статистики и её обработка. Для определения времени ожидания квитанции к среднему времени задержки добавляются 4 среднеквадратических отклонения времени задержки от его математического ожидания. Именно за это время и должна придти квитанция. Если квитанция приходит после истечения таймера, передающая сторона считает пакет утерянным.

2. Динамический способ определения таймера. Время двойного пробега определяется в процессе передачи. Первоначально передаются сегменты по одному в режиме РОС ОЖ и определяется время пробега. Время таймера увеличивается до тех пор, пока квитанции не станут не успевать приходить.

К плюсам РОС ОЖ можно отнести простоту реализации. К минусам – непроизводительное использование пропускной способности и, следовательно, низкую эффективную скорость ПД.

2. *Оконный режим переспроса.* Для данного режима характерно некоторое число сегментов, передаваемых непрерывно без ожидания квитанции. Оконный режим использует принцип конвейерной передачи. Таймер включается после передачи последнего сегмента.



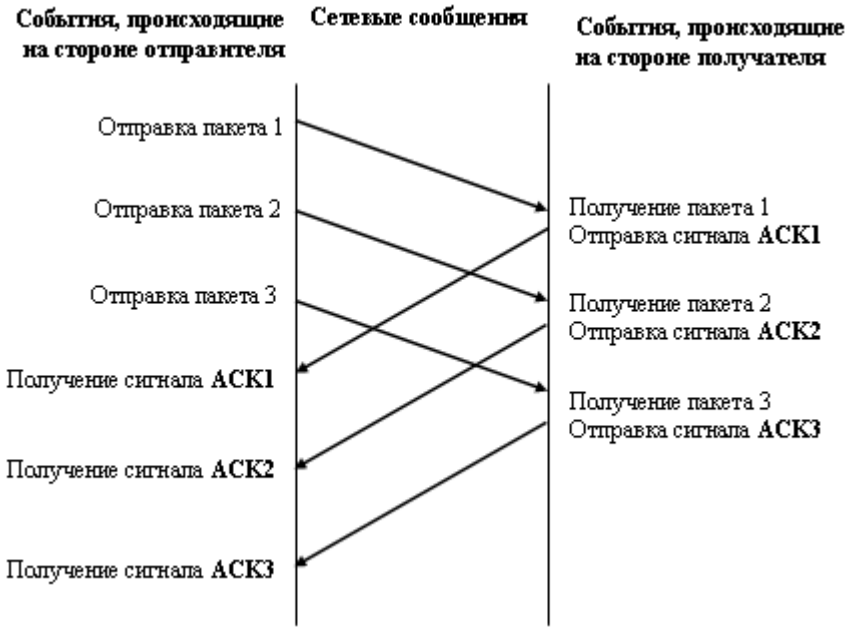
Пример движущегося окна протокола TCP

Октеты 1 и 2 успешно доставлены получателю; октеты 3-6 посланы в сеть, но подтверждение об их доставке еще не получено; октеты 7-9 еще не отправлены, но могут быть отправлены без всяких задержек; октеты с номерами 10 и выше не могут быть посланы в сеть до тех пор, пока не попадут внутрь окна.

Возможны 2 режима окна: фиксированное и скользящее.

Режим фиксированного окна. Переспрос всего окна осуществляется в случае, если хотя бы один сегмент принят с ошибкой (РОС ОЖ, но для всего окна). Эффективность использования лучше, чем при РОС ОЖ.

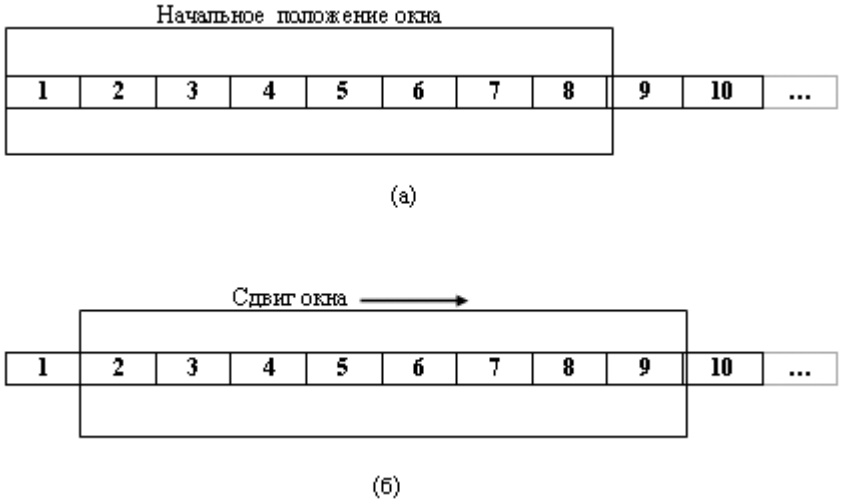
Режим скользящего окна. Переспрос осуществляется с первого ошибочного сегмента. В этом режиме подтверждения, следующие за ошибочным сегментом, не передаются. Окно смещается на число правильно принятых сегментов. Этот режим наиболее эффективен.



Пересылка трех пакетов с использованием метода движущихся окон. Идея заключается в том, что отправитель может послать в сеть сразу все три пакета, не дожидаясь сообщений о подтверждении их приема

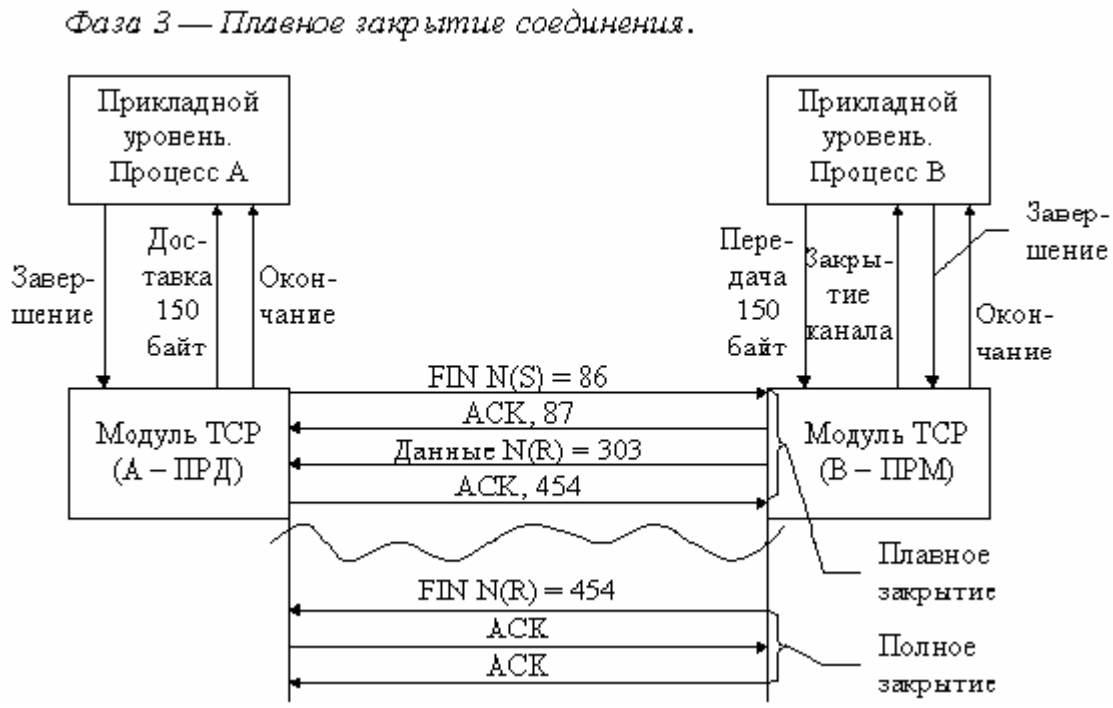
3. Адресный переспрос. Конвейерная передача в размере окна. Все принимаемые сегменты накапливаются и анализируются.

Определяются номера сегментов, которые приняты с ошибкой, и осуществляется повторная передача только этих сегментов.



Движущееся окно, внутрь которого помещено 8 пакетов (а); при получении подтверждения о приеме пакета 1 окно сдвигается на один пакет вправо, и в сеть отправляется 9-й пакет (б). Если для какого-либо пакета, попавшего в окно, не получен сигнал подтверждения, то выполняется повторная передача этого пакета

Режим окна определяется во вспомогательных параметрах TCP заголовка.



Для плавного завершения соединения передатчик отправляет сегмент с флагом *FIN* и номером байта N(S)=86. Приёмник выдаёт сегмент с флагом подтверждения *АСК* и номером ожидаемого байта N(S)=87, но у приёмника ещё остались данные для передачи, которые он и отправляет (150 байт), начиная с байта под номером N(R)=303.

Передатчик отвечает сегментом с флагом подтверждения *ACK* и номером ожидаемого байта 454 (303+150+1).

На этом виртуальное соединение прикладного уровня разрывается, но остается ещё виртуальное соединение транспортного уровня. Для его разрушения приёмник посылает сегмент с флагом *FIN* и номером ожидаемого байта N(R)=454. Передатчик отвечает подтверждением, на что приёмник также отвечает сегментом подтверждения *ACK*. На этом виртуальное соединение на транспортном уровне разрушается.

Более подробное описание протокола TCP можно найти *в RFC-793, RFC-1180*.

Пример вычисления контрольной суммы (TCP checksum) TCP-сегмента

При выполнении FTP-сессии:

```
220 ready, dude (vsFTPd 1.0.1: beat me, break me)
USER anonymous
Please specify the password.
PASS emd@pds.sut.ru
230 Login successful. Have fun.
PORT 192,168,1,50,4,81
200 PORT command successful. Consider using PASV.
NLST
150 Here comes the directory listing.
226 Directory send OK.
PORT 192,168,1,50,4,82
200 PORT command successful. Consider using PASV.
RETR cyc.txt
150 Opening BINARY mode data connection for cyc.txt (24 bytes).
226 File send OK.
QUIT
221 Goodbye.
```

был отфильтрован Ethernet-кадр, который содержит команду FTP-протокола **USER**.

==========

No:	10		
Timestamp:	13:42:53:020		
Frame type:	IP		
Protocol:	TCP-> FTP	Протокол	06
Source IP address:	GULYA	IP-адрес источника	C0 A8 01 32
Dest IP address:	195.19.219.136	IP-адрес назначения	C3 13 DB 88
		Длина TCP-сегмента	00 24
(Общая длина IP-пакета 0038) – (Длина заголовка 5×4) = 0024			
Source port:	1104	Порт источника	04 50
Destination port:	21	Порт получателя	00 15
SEQ:	5007847	Позиционный номер	00 4C 69 E7
ACK:	1006643090	Квитанция	3C 00 27 92
		Размер заголовка	5

Поля управляющих флагов	18 (011000)
Окно (размер окна приема)	22 05
Проверочная (контрольная) сумма	D3 39
Указатель срочности	00 00

Packet size: 70
Packet data:

```
0000: 00 50 FC 1E BF 8D 00 30 4F 0E 89 65 08 00 45 00   .P.....0O..e..E.
0010: 00 38 89 28 40 00 80 06 11 21 C0 A8 01 32 C3 13   .8.(@....!...2..
0020: DB 88 04 50 00 15 00 4C 69 E7 3C 00 27 92 50 18   ...P...Li...'.P.
0030: 22 05 D3 39 00 00 55 53 45 52 20 61 6E 6F 6E 79   "..9..USER anony
0040: 6D 6F 75 73 0D 0A                                 mous..
```

Определяем 16-битное дополнение суммы всех 16-битных слов в псевдозаголовке, заголовке и в поле данных.

C0 A8	01 32
C3 13	DB 88
00 06	00 24
04 50	00 15
00 4C	69 E7
3C 00	27 92
50 18	22 05
KC	00 00
55 53	45 52
20 61	6E 6F
6E 79	6D 6F
75 73	0D 0A

60A1

4448

87DC

2CC6

-

2 C C 6

0010 1100 1100 0110

1101 0011 0011 1001

D 3 3 9

5.4. Адресация в IP-сетях

5.4.1. Типы адресов стека TCP/IP

В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными или физическими), IP-адреса (называемые также сетевыми, логическими или протокольными) и символьные доменные имена.

В терминологии TCP/IP под *локальным* адресом понимается такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной сети. Если подсетью соста вной сети является локальная сеть, то локальный адрес – это MAC-адрес (Media Access Control-адрес).

MAC-адрес назначается сетевым адаптерам и сетевым интерфейсам маршрутизаторов.

MAC-адрес состоит из двух частей – 24-разрядного уникального идентификатора организации (OUI, Organizationally Unique Identifier), назначаемого Комитетом IEEE каждому производителю оборудования, и 24-разрядного номера, назначаемого самим производителем для каждой изготовленной им платы. Например: 00-60-2F-3A-07-BC.

MAC-адрес – это адрес, используемый на канальном уровне.

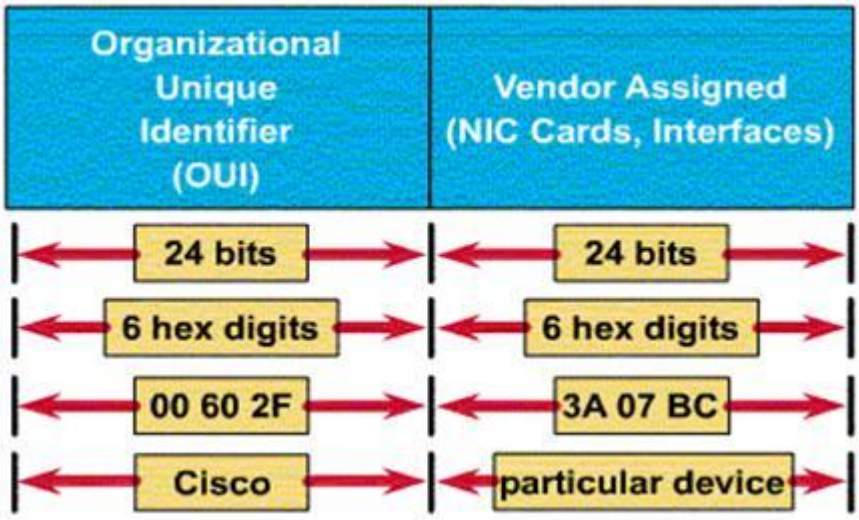


Рис. 5.17. Структура MAC-адреса

Сетевой (IP-адрес) назначается администратором во время конфигурирования компьютеров и маршрутизаторов.

IPv4 – адрес является уникальным 32-битным идентификатором IP-интерфейса в сети Интернет и используется на сетевом уровне. Он состоит из 4 байт.

IP-адрес состоит из двух частей: номера сети и номера узла. Номер узла назначается независимо от локального адреса узла.

IP-адрес принято записывать разбивкой его на октеты, каждый октет записывается в виде десятичного числа, числа разделяются точками.

Например, адрес

10100000010100010000010110000011

записывается как

10100000.01010001.00000101.10000011 = 160.81.5.131

IPv6 – адрес является уникальным 128-битным идентификатором IP-интерфейса в сети Интернет (иногда называют *Internet-2*).

Информацию о поддержке протокола IPv6 в новых версиях операционных систем, маршрутизаторах и других сетевых продуктах можно найти на сайтах:
http://www.ipv6forum.com,
http://www.ipv6.ru,
http://playground.sun.com/pub/ipng,
http://www.ipv6.org.

Набор документов RFC для IPv6 и информацию по вопросам перехода с IPv4 на IPv6 можно найти на сервере IETF по адресам:
http://www.ietf.org/rfc.html,
http://www.ietf.org/html.charters/ipngwg-charter.html,
http://www.ietf.org/html.charters/ngtrans-charter.html.

Символьный адрес. Это идентификатор-имя DNS (*Domain Name System* – доменная система имен), например, *opds.sut.ru*.

На этапе становления Интернет был составлен полный список, в который включили имена всех компьютеров, подсоединенных к сети. Однако из-за быстрого увеличения их количества, с одной стороны, и ежедневных изменений в подсоединенных сетях, с другой стороны, вскоре оказалось невозможным постоянно обновлять такой список. Эти обстоятельства привели к созданию доменной системы имен.

Эта система разделяет адреса по иерархии различных доменов (*domain* – область), представляющих собой определенную группу компьютеров.

Как видно из рассмотренного примера, в полном доменном адресе сначала указывают нужный компьютер: *opds*. Затем следуют домены по мере возрастания их уровня. Домен *sut* включает в себя группу компьютеров, расположенных в Государственном университете телекоммуникаций им проф. М.А. Бонч-Бруевича. Домен *ru* - это компьютеры, расположенные на территории России.

В доменах провайдеры создают так называемые серверы имен. Они представляют собой компьютеры, которые ищут адрес нужного

домена и устанавливают связь с сетью, обслуживающей соответствующий домен.

Таким образом, вместо полного списка всех компьютеров в Интернет имеются частные списки по доменам.

Домены составляются либо по географическим, либо по тематическим признакам.

Примерами доменов, выделенных по тематическим признакам, являются:

com (commercial) – все коммерческие предприятия в Интернет,
edu (educational) – все учебные заведения,
gov (government) – правительственные учреждения разных стран,
org (organization) – некоммерческие организации.

Примеры географических доменов:

jp (Japan) – Япония,
uk (United Kingdom) – Великобритания,
nl (Netherlands) – Нидерланды,
ca (Canada) – Канада.

5.4.2. Классы IP-адресов

Сетевой адрес состоит из двух логических частей - номера сети и номера узла в сети. Класс сети определяется значениями первых битов адреса (рис. 5.18):

• *Сети класса A.* Номер сети занимает один байт, остальные три байта определяют номер узла в сети. Для сетей класса A разрешено иметь номера в диапазоне от 1.0.0.0 до 126.0.0.0. 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. Адреса сетей класса A должны иметь первый бит равный 0.

• *Сети класса B.* Номер сети и номер узла занимают по два байта. Для сетей класса B разрешено иметь номера в диапазоне от 128.0.0.0 до 191.255.0.0. Адреса сетей класса B должны иметь первые два бита равные 10.

• *Сети класса C.* Номер сети и занимает три байта, номер узла - один. Для сетей класса C разрешено иметь номера в диапазоне от 192.0.1.0 до 223.255.225.0. Адреса сетей класса C должны иметь первые три бита равные 110.

• *Сети класса D.* Сети этого класса имеют особый групповой адрес - multicast. Для сетей класса D разрешено иметь номера в диапазоне от 224.0.0.0 до 239.255.225.225. Пакет с адресом, принадлежащим сети класса D, будет получен всеми узлами, имеющими данный адрес. Адреса сетей класса D должны начинаться с последовательности 1110.

• *Сети класса E.* Сети этого класса не используются и зарезервированы для будущих применений. Для сетей класса E разрешено иметь номера в диапазоне от 240.0.0.0 до 247.255.225.225. Адреса сети класса E должны начинаться с последовательности 11110.

Специальные IP адреса

В протоколе IP существуют несколько специальных IP адресов:
• если в поле номера сети стоят 0, то по умолчанию считается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет;
• если в поле номера узла назначения стоят сплошные 1, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером. Такая рассылка называется широковещательным сообщением *(broadcast)*;
• адрес 127.0.0.1 зарезервирован для организации обратной связи при тестировании работы программного обеспечения узла без реальной отправки пакета по сети. Этот адрес имеет название *loopback.*

Пакет, имеющий адрес *multicast*, будет доставлен сразу нескольким узлам, образующих группу с номером, указанным в поле адреса. Такие сообщения, в отличие от широковещательных, называются мультивещательными. Групповой адрес обрабатывается маршрутизатором особым образом и не делится на поля номера сети и узла.

Рис. 5.18. Классы IP-адресов

Существует ряд адресов, которые используются для организации частных сетей, то есть локальных сетей, осуществляющих обмен данными по протоколам TCP/IP (автономные IP-сети). Применение

таких адресов также позволяет легко интегрировать подобную локальную сеть в Интернет при помощи только одного «реального» IP-адреса, выделенного маршрутизатору сети.

Все пакеты, проходящие через этот маршрутизатор, автоматически получают в качестве адреса отправителя адрес маршрутизатора и, таким образом, могут быть корректно обработаны другими маршрутизаторами сети. При этом маршрутизатор, занимающийся преобразованием адресов, ведёт специальную таблицу, в которой записывается, с какого адреса «внутренней» сети на какой адрес «внешней» сети был послан запрос (а также ряд других сведений).

При получении от «внешнего» сервера ответа (пакета с некоторыми данными) маршрутизатор-преобразователь сверяется с таблицей и если находит тот адрес, который запросил пакет, то перенаправляет его получателю. В противном случае пакет уничтожается, и противоположная сторона информируется об этом по протоколу ICMP.

Данный подход может быть также полезен для защиты от несанкционированного доступа как «снаружи» сети, так и «изнутри» (имеется в виду несанкционированная передача некой информации из сети «наружу»).

В соответствии с *RFC 1918*, это диапазоны: в классе A – 10.0.0.0 ÷ 10.255.255.255; в классе B – 172.16.0.0 ÷ 172.31.255.255; в классе C – 192.168.0.0 ÷ 192.168.255.255.

Маски

Маска сети – число, состоящее из четырёх байт. Она представляется десятичными числами, разделёнными точками и используется в паре с IP-адресом. В разрядах IP-адреса, определяющих номер сети, маска содержит десятичные числа 255. Маски позволяют выделять пользователям более узкие диапазоны адресов, чем это разрешается в сетях различных классов. Наименьшим выделяемым диапазоном без использования масок является сеть класса C, т.е. 256 адресов. При использовании маски, запись 192.168.1.253 *mask* 255.255.255.252 определяет адрес 192.168.1.253 в подсети из четырех адресов: 192.168.1.252 - 192.168.1.255.

Примеры распределения IP-адресов

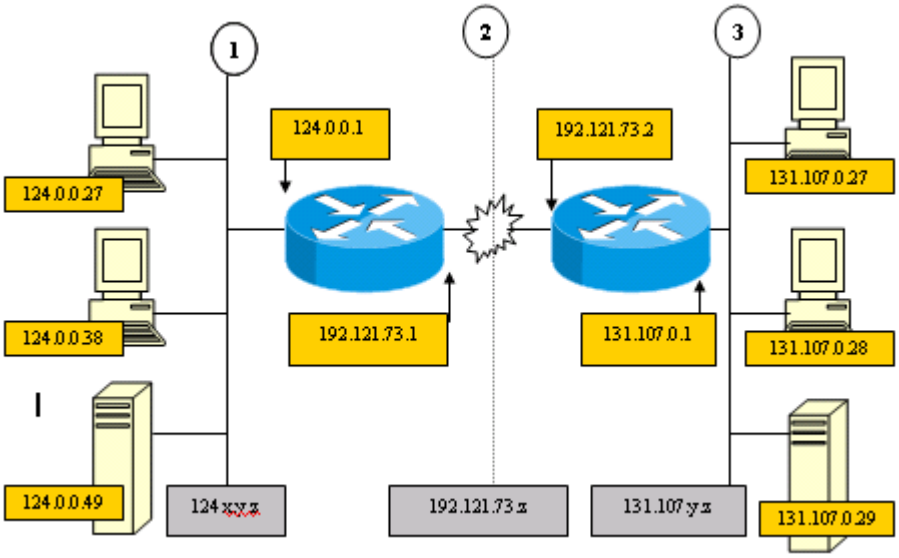


Рис. 5.19. Пример распределения IP-адресов

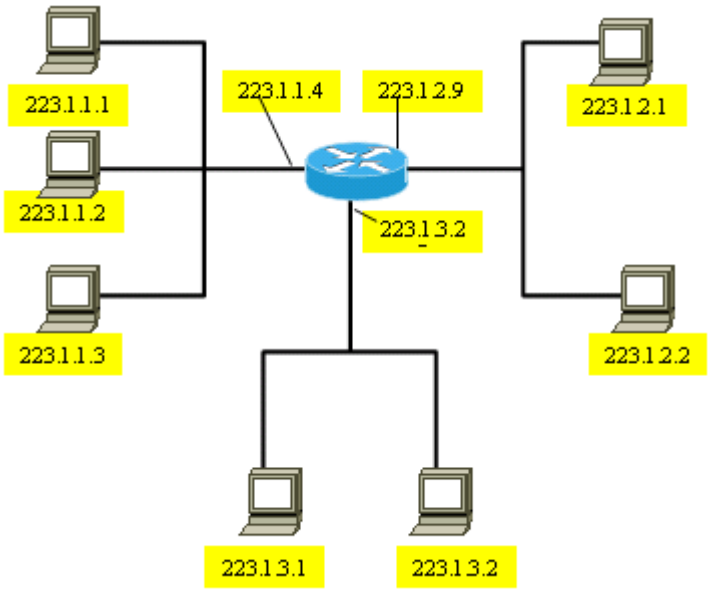


Рис. 5.20. Адреса интерфейсов

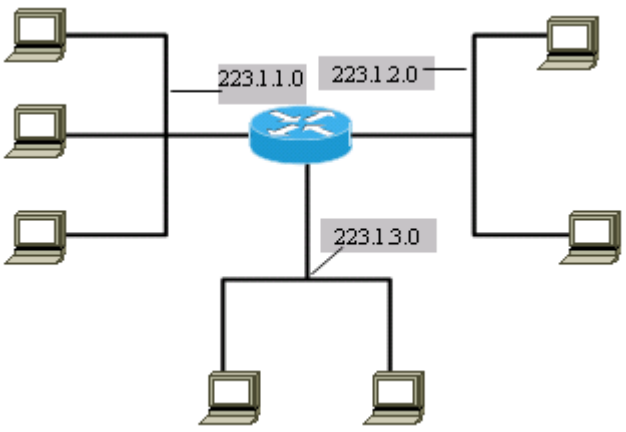


Рис. 5.21. Сетевые адреса

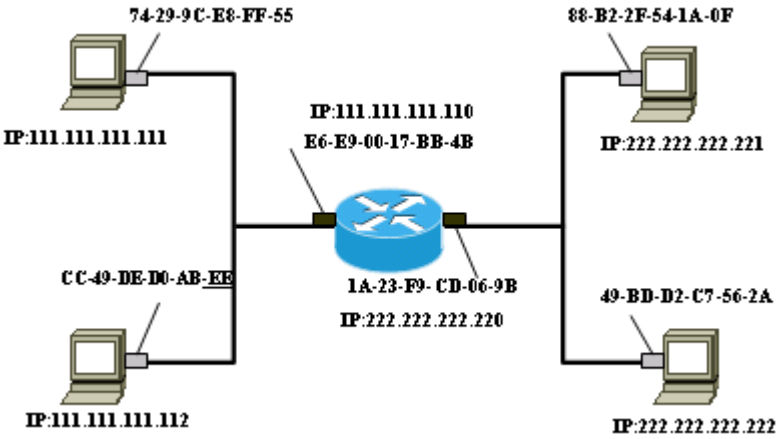


Рис. 5.22. Две локальные сети, соединенные маршрутизатором

Маршрутизатор подключается к двум или более сетям, каждая из которых воспринимает его как хост-ЭВМ. Поэтому маршрутизатор имеет физический интерфейс и специальный IP-адрес в каждой из подключаемых сетей.

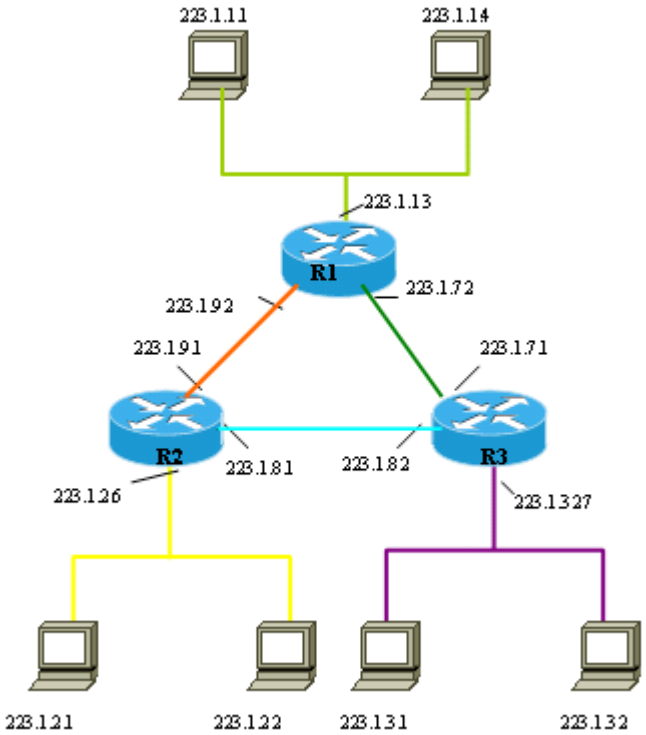


Рис. 5.23. Три маршрутизатора, соединяющие шесть хостов

5.5. Межсетевой уровень и протокол IP (Internet Protocol)

Основу межсетевого уровня (уровня межсетевого взаимодействия) составляет IP-протокол: версия IPv4 – *RFC 791* (1981 г.), версия IPv6 – *RFC 2460* (1998 г.).

Пакет, передаваемый по сети Интернет, называют *IP-дейтаграммой* или *IP-пакетом*.

Основными функциями протокола IP являются:

- перенос между сетями различных типов адресной информации унифицированной форме;
- сборка и разборка пакетов при передаче их между сетями с различным максимальным значением длины пакета.

Основными характеристиками протокола IP являются:

- формат IP пакета;
- способ обработки конфликтных ситуаций;
- способ маршрутизации.

Ненадежность доставки:

- не гарантируется доставка пакетов получателю;
- по пути следования пакет может быть утерян, продублирован, задержан;
- пакеты могут быть доставлены с нарушением порядка следования.

Для доставки пакетов не требуется предварительного установления соединения (т.е. пути следования пакетов), так как каждый пакет считается независимым от остальных. Поэтому пакеты от отправителя до получателя могут проходить по разным маршрутам.

В протоколе IP применяют четыре основных механизма для обеспечения межсетевых услуг: вид обслуживания, время жизни, контрольная сумма заголовка, дополнительные возможности (опции).

Вид обслуживания используется для указания требуемого качества обслуживания межсетевой дейтаграммы (МД) при её передаче через межсетевую систему.

Время жизни является указателем верхней границы времени существования некоторой межсетевой дейтаграммы в сети. Этот указатель задаётся отправителем и уменьшается по мере движения МД по точкам маршрута (по шлюзам), если время МД становится нулевым до того, как она достигнет получателя, то эта дейтаграмма уничтожается.

Контрольная сумма заголовка обеспечивает защиту данных в нём, если модуль обнаруживает ошибку в заголовке, то эта МД уничтожается модулем, который её обнаружит.

Дополнительные возможности обеспечивают выполнение некоторых дополнительных услуг, например, по защите данных, специальной маршрутизации.

Межсетевая дейтаграмма

Структура IP-пакета: заголовок и блок данных.

В заголовок IP-пакета включен *набор правил*, обеспечивающих доставку пакета данных получателю. В этих правилах оговариваются способы обработки пакетов узлами сети и маршрутизаторами, а также условия, при возникновении которых должны генерироваться сообщения об ошибке, а пакеты удаляться из сети.

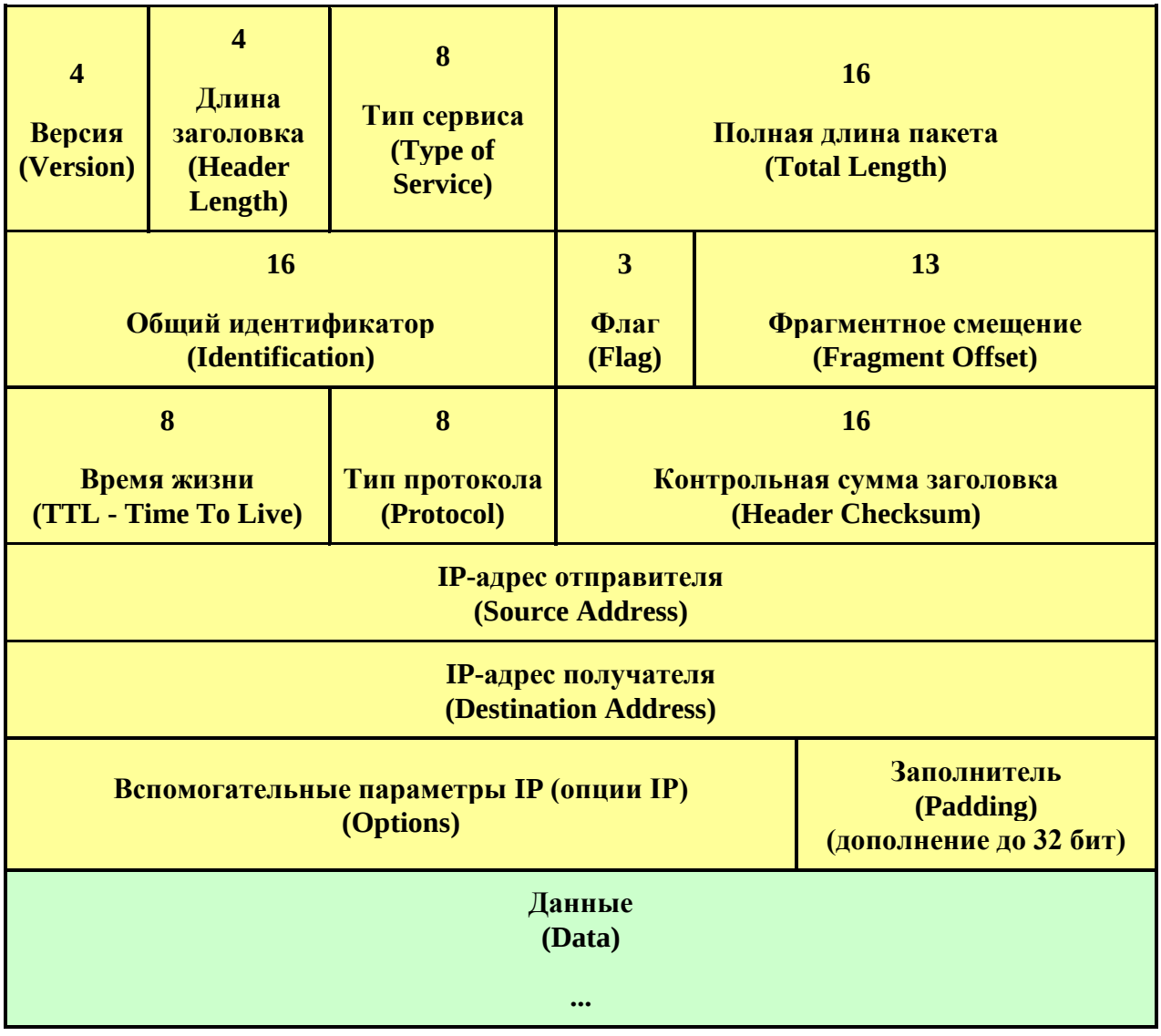


Рис. 5.24. Формат IP-пакета

Функциональное назначение полей заголовка.

Поле **Версия (Version)** указывает номер версии данного протокола межсетевого уровня. В настоящее время наряду с 4-й версией протокола (т.е. в поле — 0100) начинается использование протокола 6-й версии (т.е. в поле — 0110).

Поле **Длина заголовка (Header Length)** указывает длину заголовка межсетевой дейтаграммы в 32-разрядных словах.

Минимальная длина — пять слов, максимальная длина — пятнадцать 32-разрядных слов (на рисунке заголовок имеет шесть слов).

Поле **Тип сервиса (Type of Service)** указывает параметры требуемого качества обслуживания.

9	10	11	12	13	14	15	16
			D	T	R	C	
Приоритет			1	1	1	1	Резерв
0	0	0	низкая задержка	высокая пропускная способность	высокая надёжность	минимальная стоимость	
обычная информация							
0	0	1					
приоритетная информация							
0	1	0					
"Срочно"							
0	1	1					
"Молния"							
1	0	0					
"сверхмолния"							
1	1	1					
сетевой управляющий пакет							

Рис. 5.25. Поле Тип сервиса (Type of Service)

Приоритет *(precedence)* предоставляет возможность присвоить код приоритета каждой дейтаграмме.

Биты: 12 - D *(delay)* — задержка, 13 - T *(throughput)* — производительность (пропускная способность), 14 - R *(reliability)* — надёжность, C *(cost)* — стоимость. Только один бит может быть установлен в 1. По умолчанию все четыре бита равны 0, что означает обычный сервис.

Поле **Полная длина пакета (Total Length)** определяет общую длину дейтаграммы в октетах (байтах), включая заголовок и полезную нагрузку. Полная длина пакета может достигать 65535 байт (2^{16} - 1=65 535). Рекомендуется использовать дейтаграмму длиной 576 байт (т.е. 4608 разрядов) — 552 байта данные плюс 24 байта заголовок.

Поле **Общий идентификатор (Identification)** предназначено для сборки фрагментов межсетевых дейтаграмм.

Поле **Флаг (Flag)** обеспечивает возможность фрагментации дейтаграмм и, при использовании фрагментации, позволяет идентифицировать последний фрагмент дейтаграммы.

17	18	19
0 - резерв	0 - фрагментация разрешена	0 - последний фрагмент
	1 - фрагментация не разрешена	1 - промежуточный фрагмент

<>

Рис. 5.26. Поле Флаг (Flag)

Поле **Фрагментное смещение** указывает место данного фрагмента в межсетевой дейтаграмме. Первый фрагмент имеет смещение, равное нулю.

Для устранения из сети пакетов, задержанных вследствие каких-либо причин, в заголовке в поле **Время жизни (TTL - Time To Live)** указывается время, в течение которого пакет должен существовать в сети. Значение этого времени уменьшается при прохождении пакета по сети, а по его истечении пакет уничтожается с уведомлением отправителя соответствующим ICMP-сообщеним. Такая мера защищает сеть от циклических маршрутов и от перегрузок.

«Время жизни» задаётся в секундах — максимально 255 секунд (приблизительно 4,3 минуты). Однако часто в этом поле указывается максимальное количество хостов, через которые может пройти дейтаграмма. Это является полезным в том случае, когда задержки в сети имеют достаточно большие значения; тогда даже при суммарной задержке более 255 секунд есть вероятность доставки дейтаграммы получателю, если количество транзитных хостов не превысило максимально допустимое значение, определённое в данном поле.

Поле **Тип протокола (Protocol)** идентифицирует протокол верхнего уровня (ICMP - 1, IGMP - 2, TCP - 6, UDP - 17), который будет использован при обработке поля данных межсетевой дейтаграммы.

Протоколы транспортного уровня (протоколы TCP или UDP), пользующиеся сетевым уровнем для отправки пакетов, считают, что максимальный размер поля данных IP-пакета равен 65535, и поэтому могут передать ему сообщение такой длины для транспортировки через сеть. В функции протокола IP входит разбиение слишком длинного для конкретного типа составляющей сети сообщения на более короткие пакеты с созданием соответствующих служебных полей, нужных для последующей сборки фрагментов в исходное сообщение.

В большинстве типов локальных и глобальных сетей определяется такое понятие как максимальный размер поля данных кадра или пакета, в которые должен **инкапсулировать** свой пакет протокол IP (MTU - Maximum Transfer Unit). Так, например, сети Ethernet имеют значение MTU, равное 1500 байт, сети FDDI - 4096 байт, а сети X.25 чаще всего работают с MTU в 128 байт.

Поле **Контрольная сумма заголовка (Header Checksum)**. Для уменьшения вероятности искажения адресной части пакета и, как результат, отправки его не по адресу (и потере) заголовок пакета препровождается проверочной последовательностью — ***контрольной суммой***, занимающей 2 байта и рассчитываемой по всему заголовку.

Для вычисления контрольной суммы IP-заголовка в исходящей дейтаграмме значение этого поля сначала устанавливается в 0. Затем выполняется сложение (с циклическим переносом из старшего разряда в младший) всех 16-разрядных слов заголовка, и инвертированное значение результата записывается в поле контрольной суммы. При получении IP-дейтаграммы вновь вычисляется сумма 16-разрядных слов заголовка. Так как в заголовке принятой дейтаграммы уже

содержится сосчитанная (и инвертированная) отправителем контрольная сумма, в результате должно получиться слово, состоящее только из единиц (если в заголовке ничего не изменилось). Если же получилась другая комбинация (ошибка контрольной суммы), IP-модуль уничтожает дейтаграмму. Никакого сообщения об ошибке не порождается. Обнаружение потери дейтаграммы и повторная передача считаются проблемой, решаемой на вышестоящих уровнях иерархии протоколов.

Поскольку некоторые поля заголовка меняются в процессе движения пакета (например, время жизни), то проверочные разряды пересчитываются в каждой точке обработки межсетевой дейтаграммы. (Алгоритмы подсчета контрольных сумм в протоколах Интернет и их реализация описаны в *RFC 1071*).

Пример вычисления контрольной суммы заголовка (Header Checksum) IP-пакета

Вычисление контрольной суммы заголовка IP-пакета покажем на примере кадра, рассмотренного в п. 5.3 при *вычислении контрольной суммы (TCP checksum)* TCP-сегмента.

==========

No:	10		
Timestamp:	13:42:53:020		
Frame type:	IP		
Protocol:	TCP-> FTP	Протокол	06
Source IP address:	GULYA	IP-адрес источника	C0 A8 01 32
Dest IP address:	195.19.219.136	IP-адрес назначения	C3 13 DB 88
		Длина TCP-сегмента	00 24
(Общая длина IP-пакета 0038) – (Длина заголовка 5×4) = 0024			
Source port:	1104	Порт источника	04 50
Destination port:	21	Порт получателя	00 15
SEQ:	5007847	Позиционный номер	00 4C 69 E7
ACK:	1006643090	Квитанция	3C 00 27 92
		Размер заголовка	5
		Поля управляющих флагов	18 (011000)
		Окно (размер окна приема)	22 05
		Проверочная (контрольная) сумма	D3 39
		Указатель срочности	00 00
Packet size:	70		
Packet data:			

```
0000: 00 50 FC 1E BF 8D 00 30 4F 0E 89 65 08 00 45 00   .P.....0O..e..E.
0010: 00 38 89 28 40 00 80 06 11 21 C0 A8 01 32 C3 13   .8.(@....!...2..
0020: DB 88 04 50 00 15 00 4C 69 E7 3C 00 27 92 50 18   ...P...Li...'.P.
0030: 22 05 D3 39 00 00 55 53 45 52 20 61 6E 6F 6E 79   "..9..USER anony
0040: 6D 6F 75 73 0D 0A                                 mous..
```

Для вычисления контрольной суммы IP-заголовка значение этого поля сначала устанавливается в 0. Затем выполняется сложение (с циклическим переносом из

старшего разряда в младший) всех 16-разрядных слов заголовка, и инвертированное значение результата записывается в поле контрольной суммы.

Заголовок IP-пакета

45 00	00 38
89 28	40 00
80 06	КС=0
C0 A8	01 32
C3 13	DB 88

E E D E
1110 1110 1101 1110

0001 0001 0010 0001
1 1 2 1

IP-адреса, содержащиеся в заголовке (**IP-адрес отправителя (Source Address)**, **IP-адрес получателя (Destination Address)**), являются 32-битовыми идентификаторами объектов сети - оконечных установок и маршрутизаторов.

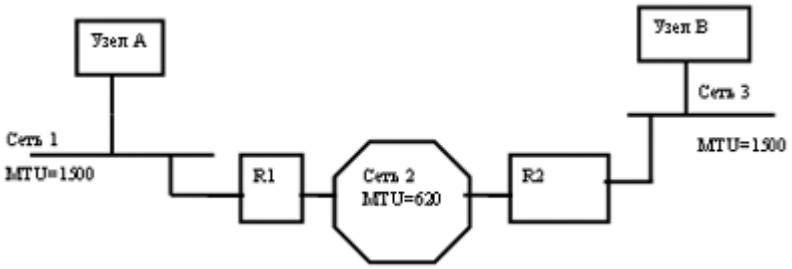
Поле **Вспомогательные параметры IP (опции IP) (Options)** — определяет наличие дополнительных услуг, имеет переменную длину и может присутствовать или отсутствовать в межсетевой дейтаграмме.

Поле **Заполнитель (Padding)** применяется для выравнивания заголовка на 32-разрядную границу.

Фрагментация

IP протокол реализует операции сборки и разборки пакетов, связанные с использованием сетей, в которых применяются форматы меньших длин, чем в пакетах получаемых от транспортного уровня. Формат IP-пакета согласуется с форматами пакетов используемых сетей.

Части, на которые разделяется дейтаграмма (IP-пакет), называются фрагментами, а сам процесс разделения - *фрагментацией.*



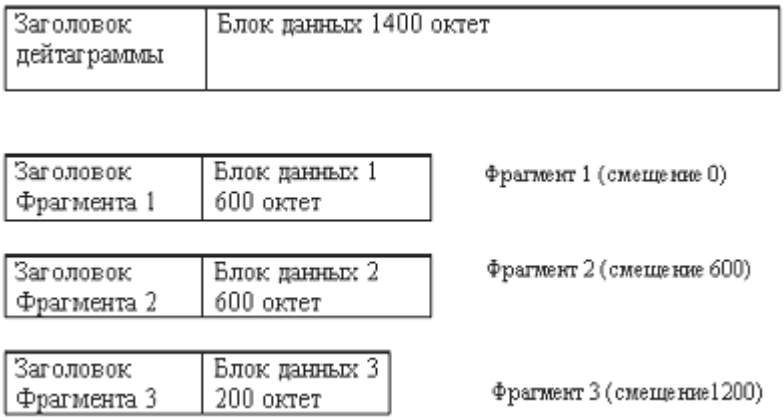


Рис. 5.27. Пример выполнения фрагментации в сети

В процессе передачи данных в сети Интернет может возникнуть необходимость передать некоторые управляющие сообщения отправителю, например, о недостижимости адресата, истечении времени жизни, о возникновении ошибки в заголовке, а также о переадресации межсетевой дейтаграммы. Для этого используется протокол **ICMP**, который является составной частью протокола IP и должен быть реализован в каждом межсетевом модуле IP.

5.6. Принципы и алгоритмы маршрутизации в Интернет

5.6.1. Проблема маршрутизации в сети Интенет

В архитектуре TCP/IP сети соединяются друг с другом коммутаторами IP-пакетов, которые называются шлюзами или IP-маршрутизаторами.

Основная задача IP-маршрутизатора — определение по специальному алгоритму адреса следующего IP-маршрутизатора.

Для решения этой задачи каждый IP-маршрутизатор должен располагать матрицей маршрутов (специальной базой данных, обеспечивающей маршрутизацию), которую необходимо регулярно обновлять.

Алгоритм маршрутизации является тем фундаментом, на котором строится вся работа базовой сети с архитектурой TCP/IP. Обеспечение надёжных сетевых услуг требует определённой динамики маршрутизации. Неожиданные изменения в связности базовой сети должны рассматриваться как обычные явления и соответствующим образом обрабатываться, так же как и перегрузки отдельных направлений и каналов.

Существует ряд требований, которые следует учитывать при выборе приемлемого алгоритма маршрутизации:

алгоритм маршрутизации должен распознавать отказ и восстановление каналов связи или других IP-маршрутизаторов и переключаться на другие, подходящие маршруты. Время переключения маршрутов должно быть меньшим, чем типичный тайм-аут пользователя протокола TCP (примерно 1 мин);

алгоритм должен исключать образование циклов, петель и эффекта «пинг-понг» в назначаемых маршрутах как между соседними IP-маршрутизаторами, так и для удалённых IP-маршрутизаторов. Существование вышеперечисленных эффектов не должно превышать типичного тайм-аута пользователя протокола TCP (примерно 1 минута);

нагрузка, создаваемая управляющими сообщениями, которые необходимы для работы алгоритма маршрутизации, не должна ощутимо ухудшать или нарушать нормальную работу сети. Изменение состояния сети, которое может прервать нормальную работу в некоторой локальной области сети, не должно оказывать воздействия на удалённые участки;

поскольку размеры сети постоянно увеличиваются, необходимо обеспечить эффективное использование сетевых ресурсов, например, изменение матриц маршрутов выполнять по частям, передавая по глобальным сетям только дополнения к базам данных по маршрутизации;

размер базы данных по маршрутизации не должен превышать некоторой константы, не зависящей от топологии сети, умноженной на количество узлов и на среднюю связность сети. Хорошая реализация не должна требовать хранения полной базы данных по маршрутизации в каждом IP-маршрутизаторе;

• если используются метрики, основанные на достижимости узла и задержке в доставке пакета, то они не должны зависеть от прямой связности со всеми другими IP-маршрутизаторами или от использования механизмов широковещательной передачи, специфичных для некоторых сетей. Процедуры опроса не должны вносить существенных дополнительных расходов;

• маршруты по умолчанию следует использовать в качестве первоначальных предположений о маршрутизации, чтобы затем выбирать окончательное направление передачи.

Кроме перечисленных выше задач IP-маршрутизатор должен обеспечивать эффективное распределение собственных ресурсов как по пропускной способности каналов, так и по объёму буферных ЗУ, используемых для хранения пакетов, ожидающих передачу. Самая очевидная стратегия «первым пришёл — первым обслужен» (FCFS — First Come First Served) может оказаться неприемлемой в условиях перегрузки сети.

Так, например, нельзя допустить, чтобы высокоскоростной канал захватил весь объём буферных ЗУ, ничего не оставив низкоскоростному каналу. В хороших алгоритмах обязательно должно учитываться поле «Тип сервиса» заголовка IP-пакета; IP-маршрутизатор может назначить больший приоритет IP-пакетам, передающим управляющую или служебную информацию.

Наконец, алгоритм маршрутизации должен обеспечивать надёжный алгоритм определения состояния каждого канала связи и узла в базовой сети и, если требуется, состояние хост-ЭВМ.

Для этого нужен, по крайней мере, протокол канального уровня, предполагающий периодический обмен кадрами через каждый канал связи.

Однако этого часто оказывается недостаточно, поэтому дополнительно требуется специальный механизм в алгоритмах маршрутизации.

По техническим, административным, географическим, а также иногда и политическим соображениям IP-маршрутизаторы группируются в так называемые «автономные системы».

IP-маршрутизаторы, входящие в одну автономную систему, контролируются одной организацией, обеспечивающей их сопровождение, и используют общие для данной автономной системы алгоритмы маршрутизации.

Определение. Конкретный вариант протокола маршрутизации, действующий внутри одной автономной системы, называется

внутренним протоколом маршрутизации (IGP — Interior Gateway Protocol).

Возможно, что некоторому IP-пакету, чтобы достичь места назначения, придётся пройти через IP - маршрутизаторы двух или более автономных систем. Поэтому автономные системы должны иметь возможность обмениваться информацией о своём состоянии.

Определение. Протокол для обмена служебной информацией между автономными системами называется внешним протоколом маршрутизации (EGP — Exterior Gateway Protocol).

Каждый IP-маршрутизатор должен обеспечить реализацию протоколов физического, канального, межсетевого уровней, а также протоколы доступа к сети. В качестве последних используются протоколы Ethernet, Frame Relay, ATM, SLIP, PPP и ряд других, а для сетей с архитектурой ISO протокол X.25/2 (LAP-B).

Кроме того, IP-маршрутизатору необходима реализация некоторого алгоритма выбора маршрута по таблице маршрутизации, а также алгоритма её обновления.

Процедура выбора пути заложена в протоколе IP, причём IP-уровень не знает всего пути, а владеет лишь информацией о том, какому IP-маршрутизатору передать IP-пакет с конкретным адресом места назначения.

Просмотр таблицы маршрутизации происходит в три этапа:

− ищется соответствие адреса, записанного в IP-пакете, адресу места назначения в маршрутной таблице. В случае успеха пакет посылается соответствующему IP-маршрутизатору или непосредственно хост-ЭВМ. Связи “точка-точка” выявляются именно на этом этапе;

− ищется соответствие адреса, записанного в IP-пакете, адресу некоторой региональной сети места назначения (одна запись в таблице маршрутизации соответствует всем хостам, входящим в данную региональную сеть). В случае успеха система действует так же, как и в предыдущем пункте;

− ищется маршрут «по умолчанию», если таковой предусмотрен; дейтаграмма посылается в соответствующий маршрутизатор.

Существуют **статические** и **динамические** алгоритмы обновления таблицы.

Статический алгоритм есть способ маршрутизации, не изменяющийся при изменении топологии и состояния сети. Примерами являются алгоритмы случайной и лавинной маршрутизации.

Случайная маршрутизация — передача данных из узла в любом, случайным образом выбранном направлении, кроме направления, по которому данные поступили в узел. Данные, совершая «блуждания» по сети с конечной вероятностью когда-либо достигают адресата.

Лавинная маршрутизация — передача данных из узла во всех направлениях, кроме того, по которому поступили данные. Очевидно,

что хотя бы одно направление обеспечит доставку пакета за минимальное время, т.е. лавинная маршрутизация гарантирует малое время доставки.

Шлюзы, входящие в состав одной автономной системы, могут выполнять алгоритм **динамической** маршрутизации — протоколы на основе алгоритма Беллмана-Форда и протоколы на основе алгоритма Дейкстры.

Каждой дуге графа ставится в соответствие действительное число, называемое длиной дуги; тогда длина пути определяется суммой длин составляющих его дуг.

Обычно это число переприёмов или средняя задержка пакетов, но возможны и другие метрики, например, пропускная способность канала связи, надёжность.

Шлюзы, работающие по **алгоритму Беллмана-Форда**, хранят вектор длин кратчайших маршрутов до всех сетей, входящих в состав объединённой сети.

Периодически каждый шлюз передаёт свой вектор соседним шлюзам автономной системы, а элементы вектора, принятого от соседнего шлюза, складываются с длинами исходящих линий связи.

На основе полученной таблицы строится новый вектор длин кратчайших маршрутов — алгоритм Беллмана-Форда (DV — алгоритм Distance Vector).

Протоколы на основе DV-алгоритма достаточно просто реализуются, требуют мало памяти и процессорного времени, однако они обладают рядом общих недостатков. При увеличении количества сетей, входящих в состав автономной системы, резко возрастает количество передаваемой информации, т.к. DV-алгоритм требует, чтобы все шлюзы периодически передавали свои векторы длин маршрутов.

Шлюзы, работающие по **алгоритму Дейкстры** (Shortest Path First — SPF-алгоритм), сначала определяют кратчайшие маршруты по всем сетям автономной системы. Для этого в каждом шлюзе строится полное дерево кратчайших путей с корнем в данном шлюзе.

Процедура построения дерева кратчайших путей использует принцип, согласно которому в дерево кратчайших путей первой включается дуга с наименьшей длиной, поэтому алгоритм Дейкстры часто называют первым кратчайшим путем.

После того как в шлюзе построено дерево кратчайших путей, изменения характеристик линий связи, определяющих длины соответствующих дуг графа, изменения топологии сети приводят к небольшим дополнительным вычислениям для корректирования дерева кратчайших путей.

Шлюзы обмениваются только сведениями о длинах исходящих линий связи, а не векторами длин маршрутов, как в случае алгоритма Беллмана-Форда. Размер корректирующих пакетов со служебной

информацией для маршрутизации мал и не зависит от числа сетей в автономной системе. Каждый шлюз посылает такие пакеты с помощью лавинной маршрутизации. При появлении в сети нового шлюза или включении новой линии связи изменения в топологии сети не учитываются при маршрутизации в течение некоторого времени для того, чтобы информация о происшедших изменениях успела достигнуть всех шлюзов автономной системы.

В целом, алгоритм Дейкстры, по сравнению с алгоритмом Беллмана-Форда, обеспечивает более реальную оценку ситуации в сети, более быструю реакцию на важные изменения в сети (такие, как включение новой линии связи) и уменьшает зацикливание пакетов; однако алгоритм Дейкстры сложнее в реализации и требует в несколько раз больше памяти.

5.6.2. Внутренние протоколы маршрутизации

Протокол GGP (Gateway to Gateway Protocol, RFC 823) был разработан и реализован фирмой BBN для первых экспериментальных шлюзов сети Интернет.
Он до сих пор используется в шлюзах фирмы BBN LSI/11, хотя считается, что GGP имеет серьёзные недостатки и позднее был заменён на алгоритм SPF.

Алгоритм протокола GGP определяет маршрут с минимальным числом переприёмов, т.е. его мерой длины является просто число транзитных участков сети между парами шлюзов.

Он реализует распределённый алгоритм кратчайшего пути, который требует глобальной сходимости маршрутных таблиц после изменений в топологии или связности.

Протокол RIP (Routing Information Protocol, RFC 1058, 1581, 1582, 1724) часто используется для класса протоколов маршрутизации, базирующихся на протоколах XNS (Xerox Network System — сетевая система Xerox) фирмы Xerox.

Реализация протокола RIP для семейства протоколов TCP/IP широко доступна, поскольку входит в состав программного обеспечения ОС UNIX, например, FreeBSD или Linux. В силу своей простоты протокол RIP имеет наибольшие шансы превратиться в **«открытый»** протокол IGP, т.е. протокол, который может использоваться для совместной работы шлюзов, поставляемых разными фирмами.

В качестве метрики маршрутизации RIP использует число скачков (шагов) до цели. Такой вид метрики не учитывает различий в пропускной способности или загруженности отдельных сегментов сети.

Каждому маршруту ставится в соответствие таймер **тайм-аута** и **«сборщик мусора».** Таймер тайм-аута сбрасывается каждый раз, когда маршрут инициализируется или корректируется. Если со времени

последней коррекции прошло 3 минуты или получено сообщение в том, что вектор расстояния равен 16, маршрут считается закрытым, но запись о нём не стирается, пока не истечёт время «уборки мусора» (2 минуты). При появлении эквивалентного маршрута переключение на него не происходит.

Протокол RIP достаточно простой, но не лишённый недостатков:

- требуется много времени для восстановления связи после сбоя в маршрутизаторе (минуты); в процессе установления режима возможны циклы;
- число шагов — важный, но не единственный параметр маршрута, да и 15 шагов — не предел для современных сетей.

Протокол "HELLO". Программное обеспечение Fuzzball для шлюза LSI/11 включает в себя реализацию протокола IGP под названием "HELLO". В отличие от RIP в нём критерием выбора маршрута служит время, а не расстояние, поэтому "HELLO" требует достаточно точной синхронизации служб времени шлюзов.

Протокол OSPF (Open Shortest Path First, RFC 1850, 1583, 1584, 1587) представляет собой протокол состояния маршрута, причём в качестве метрики используется коэффициент качества обслуживания.

Каждый маршрутизатор обладает полной информацией о состоянии всех интерфейсов шлюзов автономной системы. Определяющими являются три характеристики: задержка, пропускная способность и надёжность.

Преимущества OSPF:

- для каждого адреса может быть несколько маршрутных таблиц, по одной на каждый вид IP-операции;
- каждому интерфейсу присваивается безразмерная цена, учитывающая пропускную способность, время транспортировки сообщения; каждой IP-операции может быть присвоена своя цена;
- при существовании эквивалентных маршрутов OSFP распределяет поток равномерно по этим маршрутам;
- при связи «точка–точка» не требуется IP-адрес для каждого из концов;
- применяется мультикастинг вместо широковещательной адресации, что снижает загрузку не вовлечённых в обмен сегментов.

Недостатки OSPF — трудно получить информацию о предпочтительности каналов для узлов, поддерживающих другие протоколы или имеющих статическую маршрутизацию.

Протокол IS-IS. Учитывая, с одной стороны, широкое распространение сетей с архитектурой TCP/IP, с другой стороны, повышенное внимание правительственных и коммерческих организаций к архитектуре ЭМВОС; ожидается, что архитектуры TCP/IP и ЭМВОС будут существовать долгое время вместе. Поэтому возникает необходимость в шлюзах, способных маршрутизировать одновременно IP- и ЭМВОС-трафик.

Протокол IS-IS (Intermediate System to Intermediate System Protocol, RFC 1195) обеспечивает поддержку понятий IP-подсети, переменной маски подсети, маршрутизацию на основе значения поля «Тип сервиса» в заголовке IP-пакета и понятие внешнего маршрута.

Протокол IS-IS является динамическим протоколом маршрутизации, построенным на основе SPF-алгоритма.

5.6.3. Внешние протоколы маршрутизации

Внешние протоколы маршрутизации предназначены главным образом для связи между автономными (независимыми) системами.

Протокол EGP (Exterior Gateway Protocol) один из известных протоколов этого типа (RFC 827, 888, 904, 911, 1092, 1093).

Документ RFC 827, предложивший первую модель шлюза для взаимодействия со шлюзами других автономных систем, и документ RFC 888, представляющий собой развитие этой модели, накладывали существенное ограничение на топологию сети Internet, предполагая древовидную двухуровневую структуру, корнем которой является так называемая «магистральная» автономная система, состоящая из «магистральных» шлюзов.

Главным преимуществом такой модели считалась невозможность образования в древовидной топологической структуре циклических маршрутов между автономными системами.

С помощью протокола EGP шлюзы могут снабжать друг друга информацией о достижимости соседних шлюзов и о маршрутах к соседним шлюзам. При этом динамическое вычисление маршрутов выполняется только шлюзами магистральной автономной системы, и затем результаты могут быть сообщены немагистральным шлюзам.

Немагистральные шлюзы также могут предоставлять маршрутную информацию магистральным и немагистральным шлюзам, но они не имеют права передавать дальше маршруты, вычисленные на основе информации, полученной от других шлюзов. Это ограничение часто называется ограничением на распространение информации «третьей группы».

Протокол EGP включает в себя механизм определения достижимости соседей (соседними называются шлюзы, совместно выполняющие протокол EGP), контроля достижимости и обмена информацией в форме обновляющих сообщений. Цель использования алгоритма достижимости — убедиться в том, что сосед работает и может поставлять надёжную информацию.

Не менее важной является задача фильтрации информации перед тем, как отправлять её другим шлюзам, чтобы избежать лишних изменений базы данных.

Как правило, локальные шлюзы передают по внешнему протоколу только сведения, касающиеся своих автономных систем, чтобы не увеличивать без необходимости трафик в сетях.

Протокол BGP (Border Gateway Protocol, RFC 1267) — это протокол маршрутизации между автономными системами в сети Internet; он построен на основе опыта, накопленного при эксплуатации протокола EGP.

Главная цель BGP — сократить транзитный трафик. Протокол BGP использует расширенное понятие автономной системы. В данном случае внутри автономной системы шлюзы могут использовать несколько различных протоколов маршрутизации и несколько метрик. Однако внутри автономной системы должен существовать единый план маршрутизации, позволяющий рассматривать автономную систему как единое целое.

В зависимости от того, с каким трафиком имеет дело автономная система, она причисляется к одной из следующих категорий:

- тупиковая автономная система, имеющая единственное соединение с другими автономными системами; фактически, тупиковая система имеет дело только с локальным трафиком;
- многовходовая автономная система. Эта система имеет более одного соединения с другими автономными системами, но она отказывается поддерживать транзитный трафик;
- транзитная автономная система, которая имеет более одного соединения с другими автономными системами и предназначена для поддержания обоих видов трафика.

Протокол BGP использует в качестве транспортного протокола протокол TCP.

Хост-ЭВМ, выполняющие протокол BGP, не обязательно должны одновременно являться шлюзами.

Хост-ЭВМ, не являющаяся шлюзом, может обмениваться маршрутной информацией со шлюзами при помощи протокола EGP или внутреннего протокола маршрутизации.

Эта хост-ЭВМ может затем использовать протокол BGP для обмена маршрутной информацией с граничным шлюзом другой автономной системы.

Выводы

1. Интернет (Internet) – это сеть, объединяющая отдельные локальные, региональные, национальные и глобальные сети, это глобальное сообщество мировых ИВС.
2. Сеть Интернет является сетью с пакетной коммутацией и функционирует на основании стека протоколов TCP/IP.
3. Сеть Интернет предоставляет пользователям широкий спектр услуг по передаче и обработки различных видов информации, в том числе трафика реального времени.

4. Распределение информации в сети Интернет осуществляется на основе протоколов внутренней и внешней маршрутизации. Обеспечивающих надежную доставку данных.

Литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб: Питер, 2003.

2. Основы архитектуры Internet / Уч. пособие для ВУЗ. В.В. Камышников, Ю.М. Казаченко, Н.М. Крикунов. – ПГАТИ, 2003.

3. Когновицкий О.С., Доронин Е.М., Свердлов Л.М. Структура и протоколы электронной почты в Internet (спец. 200900, 220200, 220400): Уч. пособие / СПбГУТ. – СПб, 2004.

4. http://www.rfc-editor.org

5. http://opds.sut.ru