Глобальные компьютерные сети

Процессы взаимодействия вычислительных средств достаточно быстро переросли рамки отдельных фирм и организаций. Тенденции к интеграции и глобализации в современном мире получили адекватное отражение и в сфере компьютерных технологий. Совокупности вычислительных машин, объединенных коммуникационной средой, охватывающей значительные по расстоянию территории, получили название **глобальных компьютерных сетей**. За последние два-три десятилетия все виды организационного, аппаратного и программного обеспечения этих сетей бурно развивались и претерпели массу метаморфоз. Среди сетей, получивших общемировую известность, могут быть названы **SPRINT**, некоммерческая компьютерная сеть **FIDO**, международная система расчетов **S.W.I.F.T**. Однако самой примечательной тенденцией в последние годы стало выделение из всего сообщества компьютерных сетей сети **Интернет** в качестве явного лидера как по размерам, так и по темпам роста функциональных возможностей.

## 1. Основные принципы построения сети Интернет

**Интернет** – это глобальная информационная инфраструктура, представляющая собой механизм распространения данных и среду взаимодействия между пользователями и компьютерами вне зависимости от их географического положения.

Первоначально целью создания Интернета являлось объединение компьютерных сетей различных типов. В процессе создания сети Интернет были разработаны технические принципы функционирования и объединения компьютерных сетей, решена проблема управления глобальными информационными структурами, использованы новые принципы совместной работы над проектами и управления информационными потоками.

Первые исследования в области соединения удаленных компьютеров были проведены в начале 60-х годов ХХ века. В 1965 г. компьютер, находящийся в Массачусетском технологическом институте, был подключен к компьютеру в Калифорнии по телефонной линии. Для соединения использовалась технология *коммутации каналов*, характерная для телефонных линий.

Технология коммутации каналов подразумевает создание непрерывной физической линии связи между двумя абонентами – *канала*. Соединить всех желающих абонентов друг с другом невозможно, поэтому используется коммутация, т.е. возможность предоставления линии связи нескольким абонентам одновременно. Канал состоит из отдельных участков, которые соединяются между собой специальной аппаратурой – *коммутаторами*. Если абонент хочет установить соединение, то он обращается к ближайшему коммутатору, который, в свою очередь, по свободному каналу обращается к следующему и т.д. В конечном итоге устанавливается *прямое соединение двух абонентов*, и они могут обмениваться данными.

В результате эксперимента выяснилось, что коммутация каналов не подходит для создания компьютерных сетей. При использовании технологии коммутации каналов аппаратура абонентов должна работать с одинаковой скоростью, в то время как компьютеры обрабатывали данные с различной скоростью. Кроме того, неэффективно используется канал связи. Когда данные передаются, канал загружен, а когда обрабатываются – свободен. Однако физическое соединение и в том, и в другом случае сохраняется. Все это потребовало применения новой технологии передачи данных – *коммутации пакетов*. При использовании этой технологии все передаваемые в сети сообщения разбиваются на небольшие части, которые называются *пакетами*. Каждый пакет снабжается заголовком, в котором указывается *адрес назначения пакета*. Коммутаторы, используя адрес, передают пакеты друг другу до тех пор, пока он не достигнет места назначения. Если какой-либо коммутатор слишком загружен и не может передать пакет в течение некоторого времени, он помещает его в очередь пакетов и передает позже.

Проект сети компьютеров на основе коммутации пакетов был разработан в Агентстве перспективных разработок Министерства обороны США (**DARPA**). Создаваемая сеть получила название **ARPANET**. В 1969 г. началась реализация проекта, в сеть **ARPANET** были включены четыре удаленных компьютера. Для включения в сеть большего числа компьютеров необходимо было выработать некоторый единый набор правил, определяющих способ взаимодействия узлов сети **ARPANET** (последовательность передачи, формат сообщений) – протокол. В 1971-72 гг. работа над единым протоколом для сети **ARPANET** была завершена. Этот протокол получил название **Network Control Program** (**NCP**). Создание протокола позволило начать разработку прикладных программ для использования в сети. Одной из первых таких программ стала *программа электронной почты*, которая позволила участникам проекта эффективнее обмениваться информацией между собой.

Возникновение Интернета базировалось на *идее*, предполагавшей возможность *объединения нескольких независимых компьютерных сетей с различной архитектурой* на основе **ARPANET**.

В протоколе **NCP** не предполагалось наличие какого-либо механизма взаимодействия с сетями другой, нежели **ARPANET**, архитектуры. Кроме того, объединяя сети, необходимо было учесть возможность *временного нарушения связи* или даже выхода части объединенной сети из строя. При этом оставшаяся часть сети должна продолжать нормально работать. Таким образом, были сформулированы **основные принципы построения** новой сети:

* для включения в глобальную сеть отдельной сети *не должно производиться никаких дополнительных изменений локальной сети*;
* *пакеты передаются на основе принципа негарантированной доставки*: если пакет не смог достигнуть пункта назначения, то через короткое время он должен быть передан снова;
* *для соединения сетей используются маршрутизаторы*, максимально упрощающие прохождение потока пакетов;
* *не должно существовать единого, централизованного управления объединенной сетью*.

Ключом к объединению сетей стал новый протокол, поддерживающий *межсетевое взаимодействие*: **TCP** (*Transmission Control Protocol*), первая версия которого появилась в 1973 г. Он обеспечивал доставку сообщений в Интернете и предоставлял достаточно широкий диапазон транспортных услуг.

**TCP** прекрасно работает при решении таких задач, как передача файлов или удаленная регистрация, но в некоторых случаях (например, при передаче речи) потери пакетов не могут быть исправлены только средствами **TCP**, а эти функции должны быть возложены на приложение. Этот факт привел к разделению **TCP** на два протокола:

* **IP** – для адресации и передачи отдельных пакетов;
* **TCP** – для разделения сообщений на пакеты, обеспечения целостности и восстановления потерянных пакетов.

Для приложений, которые не хотели использовать **TCP**, был добавлен альтернативный протокол, названный **UDP**, который обеспечивал прямой доступ к **IP**. Объединенный протокол принято называть **TCP/IP**.

*Основной движущей силой* при создании **ARPANET** и Интернета была необходимость обеспечить *разделение ресурсов*. Соединить два компьютера вместе оказалось гораздо экономичнее, чем копировать их функции. Однако передача файлов, удаленная регистрация и электронная почта произвели гораздо больший эффект, чем предполагалось в то время.

Одной из ключевых концепций Интернета было не создание одного или нескольких приложений для работы с сетевыми устройствами, а *создание общей инфраструктуры*, в которой могли бы существовать новые приложения.

Следующим сильным толчком к развитию Интернета было *увеличение размеров сети* и связанные с этим проблемы управления. Все большее количество людей использовало сеть Интернет, и все большее количество узлов подключалось к Интернету. В протоколе **IP** адрес узла представлен в виде набора цифр, разделенных точками.

Запомнить огромное множество комбинаций **IP**-адресов практически невозможно. Для того чтобы упростить использование сети людьми, с каждым узлом ассоциировались *имена* и, таким образом, не было необходимости запоминать числовые адреса. Первоначально существовало ограниченное число узлов, и соответствия адресов и имен можно было хранить в одной таблице. Появление большого количества независимо управляемых сетей привело к тому, что хранить адреса в одной таблице было невозможно, тогда появилась *система доменных имен* (**DNS**), представляющая собой масштабируемый, распределенный, иерархический *механизм разрешения имен узлов* и их *адресов*.

Увеличение размеров Интернета потребовало изменения возможностей маршрутизаторов. Первоначально существовал *единый алгоритм маршрутизации*, который исполнялся всеми маршрутизаторами в Интернете. В то время как число хостов росло, возможности исходного алгоритма не могли быть адекватным образом расширены, и он был заменен *иерархической моделью маршрутизации*, включавшей протокол **IGP**, который использовался внутри регионов, и протокол **EGP**, который использовался для связи регионов вместе. Дизайн протоколов позволял использовать различные версии **IGP** в зависимости от требований к системе (стоимость, скорость конфигурации, чувствительность к сбоям, масштабируемость). Требования к маршрутизаторам определялись не только алгоритмами маршрутизации, но и размерами таблиц маршрутизации. Создание иерархической модели маршрутизации определило современную структуру сети Интернет. В 1980 г. протокол **TCP/IP** был принят в качестве стандарта для сети **ARPANET**. Переход на использование **TCP/IP** позволил разделить сеть **ARPANET** на две различные сети:

* **MILNET** – предназначена для военных целей;
* **ARPANET** – использовалась исследовательскими и научными организациями.

В 1986 г. Национальный научный фонд США (**NSF**) начал создание собственной сети, объединяющей крупные научные суперкомпьютерные центры США. В качестве основы для новой сети был выбран протокол **TCP/IP** и другие технологии, опробованные в **ARPANET**. В дальнейшем эта сеть, получившая название **NSFNET**, стала основной магистралью сети Интернет. В 1988 г. было принято решение предоставлять доступ к сети **NSFNET** не только научным и образовательным организациям, но и коммерческим фирмам. В 1994 г. финансирование основной магистрали сети Интернет было полностью передано от **NSF** различным государственным и коммерческим организациям.

Важным фактором в быстром росте сети Интернет стал открытый и свободный доступ к *документации*, в особенности к спецификациям протоколов. Обычный, традиционный способ публикации результатов исследований был слишком медленным для динамичного обмена идеями в сообществе разработчиков Интернета. Поэтому в 1969 г. была создана система документов, названная **RFC** (**Request For Comment**), задачей которой было быстрое распространение идей и результатов в Интернете. Первоначально RFC распространялись с помощью электронной почты.

**Целью** **RFC** было создание *цикла обратной связи* между разработчиками, чтобы идеи и предложения, представленные в одном из RFC, давали начало другим идеям в следующих документах и т.д. Когда будет найден некоторый последовательный ряд идей, они объединяются для создания *спецификации*. Подобная спецификация может быть использована другими разработчиками.

Открытый и бесплатный доступ к документам **RFC** способствует росту Интернета, так как позволяет использовать действующие спецификации в качестве примера для обучения и разработки новых систем. Ранние версии RFC представляли собой обращение одного автора ко всему сообществу Интернета. Сейчас документы **RFC** являются совместным творчеством нескольких разработчиков.

В конце 70-х годов стало понятно, что рост размеров Интернета сопровождается ростом числа заинтересованных групп и, таким образом, *необходима координация*. Было сформировано несколько координирующих органов:

* **IAB** (*Internet Architecture Board*) занимается координацией деятельности в области развития структуры сети Интернет;
* **IETF** (*Internet Engineering Task Force*) состоит из нескольких рабочих групп, разрабатывающих и утверждающих стандарты для сети Интернет (в настоящее время в **IETF** существует 75 рабочих групп, изучающих различные аспекты развития Интернета);
* впоследствии основным органом, осуществляющим информационную поддержку и регулирование в Интернете, стал **Internet Society** (**ISOC**) – общественная организация, базирующаяся на взносах и пожертвованиях спонсоров. **ISOC** проводит ежегодные конференции, выпускает информационные материалы, поддерживает информационные серверы.

В настоящее время основу сети Интернет составляют *высокоскоростные магистральные сети*. Независимые сети подключаются к магистральной сети через *точки сетевого доступа* **NAP** (**Network Access Point**). Независимые сети рассматриваются как автономные системы, т.е. каждая из них имеет собственное административное управление и собственные протоколы маршрутизации. Изменение протоколов маршрутизации внутри автономной системы не влияет на работу остальных систем. Деление сети Интернет на автономные системы позволяет распределить информацию о топологии всей сети и существенно упростить маршрутизацию.

Автономная система должна состоять не менее чем из 32 меньших по размеру сетей. Обычно в качестве автономных систем выступают крупные, независимые, национальные сети. Примерами подобных сетей являются сеть **EUNet**, охватывающая страны центральной Европы, сеть **RUNet**, объединяющая университеты России. Автономные сети могут образовывать компании, специализирующиеся на предоставлении услуг доступа в сеть Интернет, - *провайдеры*.

Внутри автономной системы данные предаются от одной сети к другой, пока не достигнут точки сопряжения с другой автономной системой. Обмен данными возможен только в том случае, если между автономными системами существуют *соглашения о предоставлении транзита*. По этой причине для пользователей разных автономных систем время доступа к одному и тому же ресурсу может существенно различаться.

Сети, включенные в автономные системы, представляют собой *региональные сети*, *сети университетов*, *исследовательских центров* и *коммерческих фирм*, а также сети более мелких региональных провайдеров.

Важным параметром, определяющим качество работы в сети Интернет, является **скорость доступа к ресурсам сети**, которая определяется *пропускной способностью канала связи внутри автономной системы и между автономными системами*. Для модемного соединения, которое используют большинство домашних пользователей персональных компьютеров, пропускная способность канала невелика - от 19,2 до 57,6 Кбит/с; для выделенных телефонных линий, часто используемых для подключения к сети Интернет небольших локальных компьютерных сетей, - от 64 Кбит/с до 2 Мбит/с; для спутниковых и оптоволоконных каналов связи, которые в основном используются для создания автономных сетей, - от 2 Мбит/с и выше.

## 2. Основные протоколы сети Интернет

В основном в сети Интернет используется семейство протоколов **TCP/IP**.

На *канальном* и *физическом* *уровнях* модели OSI **TCP/IP** поддерживает многие из существующих стандартов, определяющих среду передачи данных. Это могут быть, например, технологии **Ethernet** и **FDDI** для локальных компьютерных сетей или **Х.25** и **ISDN** для организации крупных территориальных сетей. На этом уровне также могут использоваться протоколы **РРР** и **SLIP**, предназначенные для установления соединения с использованием аналоговых линий связи.

Основой семейства протоколов **TCP/IP** является *сетевой уровень*, представленный протоколом **IP**, а также различными протоколами маршрутизации. Этот уровень предоставляет *адресное пространство*, обеспечивает *перемещение пакетов* в сети, а также *управляет их маршрутизацией*.

Размеры пакета, параметры передачи, контроль целостности осуществляются на *транспортном уровне* протоколом **TCP**. Протокол **UDP** работает на этом же уровне, но применяется в том случае, когда требования к надежности передачи данных менее жесткие.

*Прикладной уровень* объединяет все службы, которые система предоставляет пользователю. К наиболее важным **прикладным протоколам** относятся *протокол удаленного управления* **TELNET**, *протокол передачи файлов* **FTP**, *протокол передачи гипертекста* **HTTP**, *протоколы для работы с электронной почтой* **SMTP**, **POP**, **IIАР** и **MIME**. На этом уровне работает *система доменных имен* **DNS**, отвечающая за преобразование числовых **IP**-адресов в имена. Кроме того, следует отметить протокол **SNMP**, предназначенный для *управления сетевыми устройствами*.

***Адресация в сети Интернет***

Каждый компьютер, включенный в сеть Интернет, имеет уникальный **IP**-адрес, на основании которого протокол **IP** передает пакеты в сети.

**IP**-адрес состоит из четырех байтов и записывается в виде четырех десятичных чисел, разделенных точками, например, **194.85.120.66**.

Он состоит из двух логических частей: *номера сети* и *номера узла в сети*. Если сеть, в которую включен компьютер пользователя, является частью Интернета, то *номер сети* выдает специальное подразделение Интернета - **InterNIC** (**Internet Network Information Center**) или его представители. *Номер узла* определяет администратор сети.

В зависимости от того, какое количество байтов в **IP**-адресе выделяется для номера сети и номера узла, выделяют несколько классов **IP**-адресов:

* **класс А** – номер сети занимает один байт, а номер узла три байта. Количество узлов в сети может достигать 224, или 16 777 216. Номер сети класса А меняется в диапазоне от **1.0.0.0** до **126.0.0.0**;
* **класс В** – под номер сети и номер узла отводится по два байта. Количество возможных узлов в сети составляет 216, или 65536 узлов. Номер сети класса В меняется от **128.0.0.0** до **191.255.0.0**;
* **класс С** – под номер сети отводится три байта. Количество узлов в сети ограничено 28, или 256. Номер сети меняется от **192.0.1.0** до **223.255.255.0**.

Например, в **IP**-адресе **194.85.120.66**, **0.0.0.66** – это номер узла в сети класса С с номером **194.85.120.0**.

Существует несколько специальных **IP**-адресов. Так, например, адрес **127.0.0.1** определяет локальную машину пользователя и используется для тестирования различных программ. При этом данные по сети не передаются.

***Протокол IP***

Протокол **IP** представляет собой основу протоколов **TCP/IP**. Он относится к типу *протоколов без установления соединения*, т.е. никакой управляющей информации кроме той, что содержится в самом **IP**-пакете, по сети не передается. Кроме того, он не гарантирует надежной доставки сообщений.

Поток данных протокол **IP** разбивает на определенные части – *датаграммы* и рассматривает каждую датаграмму как независимую единицу, не имеющую связи с другими датаграммами.

**Дейтаграмма** – единица данных, которой оперируют протоколы без установления соединения. Основной задачей протокола **IP** является передача датаграмм между сетями. Часто датаграммы, передаваемые с помощью протокола **IP**, называют **IP***-пакетами*.

***Протокол TCP/IP***

Так как протокол **IP** *не гарантирует надежную доставку сообщений*, эту задачу решает протокол **TCP**, который устанавливает логическое соединение между взаимодействующими процессами. Перед передачей данных посылается *запрос на начало сеанса передачи*, а получателем посылается *подтверждение*.

Надежность протокола **TCP** заключается в том, что источник данных повторяет их посылку в том случае, если не получит в определенный промежуток времени от адресата подтверждения их успешного получения. Части, на которые протокол **TCP** разбивает поток данных, принято называть **сегментами**,

Каждый сегмент предваряется *заголовком*, в котором существует поле контрольной суммы. Если при пересылке данные повреждены, то по контрольной сумме протокол **TCP** может это определить. Поврежденный сегмент уничтожается, а источнику ничего не посылается. Если данные не были повреждены, то они пропускаются на сборку сообщения приложения, а источнику отправляется подтверждение.

Для транспортировки сегментов протокол **TCP** использует протокол **IP**. Перед отправкой протокол **TCP** помещает сегменты в оболочку **IP**-пакета.

***Порты и соединения***

Задача протокола **TCP** заключается в передаче данных между любыми прикладными процессами, выполняющимися на компьютерах в сети. На каждом компьютере может выполняться одновременно несколько процессов. Для того чтобы доставить сообщение определенному процессу, необходимо каким-то образом идентифицировать его среди других. *Идентификатором процесса* служит **номера порта**. Номер порта и **IP**-адрес компьютера однозначно определяют процесс, работающий в сети. Набор этих параметров называется **сокет**.

За некоторыми процессами номера портов закреплены. Например, **порт 21** закреплен за *службой удаленного доступа к файлам* **FTP**, **порт 23** – за *службой удаленного управления* **TELNET**.

Для организации надежной передачи данных предусматривается установление логического соединения между прикладными процессами, которое определяется парой сокетов взаимодействующих процессов. В процессе соединения осуществляется подтверждение правильности приема сообщений и при необходимости выполняется повторная передача.

***Система доменных имен DNS***

Человеку крайне неудобно использовать числовые **IP**-адреса, поэтому логичным представляется создание механизма, позволяющего *ставить в соответствие* ***IP****-адресам символьные имена*. В сети Интернет для этой цели используется *система доменных имен* (**DNS**), которая имеет иерархическую структуру. Младшая часть доменного имени соответствует *конечному узлу сети*. Составные части отделяются друг от друга точкой. Например, **mail.econ.pu.ua**. У одного узла может быть несколько имен.

Совокупность имен, у которых несколько старших частей доменного имени совпадают, называется **доменом**. Например, имена **mail.econ.pu.ua** и **www.econ.pu.ua** принадлежат домену **econ.pu.ua**.

Самым главным является **корневой домен**. Далее следуют домены первого, второго и третьего уровней. Корневой домен управляется **InterNIC**. Домены первого уровня назначаются для каждой страны, при этом принято использовать трехбуквенные и двухбуквенные аббревиатуры. Так, например, для Украины домен первого уровня **ua**, для России - **ru**, для США - **us**. Кроме того, несколько имен доменов первого уровня закреплено для различных типов организаций:

* **com** – коммерческие организации (например, **ibm.com**);
* **edu** – образовательные организации (например, **spb.edu**);
* **gov** – правительственные организации (например, **loc.gov**);
* **org** – некоммерческие организации (например, **w3.org**);
* **net** – организации, поддерживающие сети (например, **ripn.net**);

Для каждого имени домена создается свой **DNS**-сервер, который хранит базу данных соответствий **IP**-адресов и доменных имен, расположенных в данном домене, а также содержит ссылки на **DNS**-серверы доменов нижнего уровня. Таким образом, для того чтобы получить адрес компьютера по его доменному имени, приложению достаточно обратиться к **DNS**-серверу корневого домена, а тот, в свою очередь, перешлет запрос **DNS**-серверу домена нижнего уровня. Благодаря такой организации системы доменных имен нагрузка по разрешению имен равномерно распределяется среди **DNS**-серверов.

## 3. Сервисы Интернета

Интернет — это, прежде всего, глобальная компьютерная сеть. Существует большое количество разнообразных услуг (сервисов), которыми мы пользуемся, работая в Сети.

***Электронная почта***

**Система электронной почты** позволяет доставить сообщение на любой компьютер, включенный в сеть Интернет. Сообщение может содержать *текст* или файл практически любого формата – графику, музыку и т.д.

Все пользователи электронной почты имеют *уникальные адреса*. Адрес пользователя зарегистрирован в определенном домене Интернета. С каждым доменом связан *почтовый сервер*, управляющий адресами пользователей.

Пользователь набирает текст письма в специальной программе - **почтовом клиенте**, позволяющем создавать и редактировать новые письма, обрабатывать пришедшие, хранить и систематизировать переписку и т.п. Почтовый клиент помещает письмо в «почтовый ящик» пользователя, расположенный на почтовом сервере. Сервер, в свою очередь, передает письмо на почтовый сервер адресата.

В Интернете принята система **адресов электронной почты**, которая базируется на *доменном адресе машины*, подключенной к сети. Адрес пользователя состоит из двух частей, разделенных символом «**@**», например, **Jones@Registry.org**. Здесь **Jones** – это имя пользователя, а **Registry.org** – адрес – доменное имя почтового сервера.

Почтовое сообщение состоит из трех частей:

* *конверта*;
* *заголовка*;
* *тела сообщения*.

Пользователь видит только заголовок и тело сообщения. Конверт используется программами доставки.

**Заголовок** всегда находится перед телом сообщения и отделен от него пустой строкой. **RFC-822** регламентирует содержание заголовка сообщения: заголовок состоит из полей; поля состоят из имени поля и содержания поля; имя поля отделено от содержания символом двоеточия «:». Минимально необходимыми являются поля **Date**, **From**, **cc** или **То**, например:

Date: 26 Aug 05 1429 EOT

From: Jones@Registry.org

или

Date: 26 Aug 05 1429 EOT

From: Jones@Registry.org

To: Smith@Registry.org

Поле **Date** определяет дату отправки сообщения, поле **From** - отправителя, а поля **сс** и **То** - получателя или нескольких получателей.

Часто заголовок содержит дополнительные поля:

Date: 26 Aug 05 1429 EOT

From: George Jones

Sender: Secy@SHOST

To: Smith@Registry.org

Message -ID: <4231.629.XYzi - What@Registry.org>

В данном случае поле **Sender** указывает, что **George Jones** не является автором сообщения. Он только переслал сообщение, которое получил из **Secy@SHOST**. Поле **Message** **- ID** содержит уникальный идентификатор сообщения и используется программами доставки почты. Следующее сообщение демонстрирует все возможные поля заголовка:

Date: 27 Aug 05 0932

From: Ken Davis

Subject: Re: The Syntax in the RFC

Sender: KSecy@Other - host

Reply to: Sam. Irving@Reg.Organization

To: George Jones

cc: Important folks: Tom Softwood , "Sam Irving"@0ther - Host;

Standard Distribution: /main/davis/people/standard@0ther - Host

Comment: Sam is away on bisiness.

In reply to: George's message

X special action: This is a sample of user - defined field - names. Message - ID:

<4331.629.XYzi - What@0ther - Host

Поле **Subject** определяет тему сообщения, **Reply to** - пользователя, которому отвечают, **Comment** - комментарий, **In reply to** - показывает, что сообщение относится к типу «*В ответ на Ваше сообщение, отвечающее на сообщение, отвечающее …*», **X special action** - поле, определенное пользователем, которое не определено в стандарте.

Формат сообщения постоянно дополняется и совершенствуется. Так в **RFC-1327** введены дополнительные поля для совместимости с почтой **Х.400**. Кроме того, следует обратить внимание на поля некоторых, довольно часто встречающихся заголовков, которые не регламентированы в **RFC-822**. Так, первое предложение заголовка, которое начинается со слова **From**, содержит **UUCP** - путь сообщения, по которому можно определить, через какие машины сообщение «пробиралось». Поле **Received** содержит транзитные адреса почтовых серверов с датой и временем прохождения сообщения. Вся эта информация полезна при разборе трудностей с доставкой почты.

Возможности электронной почты не ограничиваются только пересылкой корреспонденции. По почте можно получить доступ ко многим ресурсам Интернета, которые используют *почтовых роботов*, отвечающих на запросы пользователей. Поэтому имеет смысл более детально изучить программное обеспечение, поддерживающее электронную почту.

Стандарт **MIME** (или, в нотации Интернета, **RFC-1341**) предназначен для описания тела почтового сообщения Интернета. Предшественником **MIME** является Стандарт почтового сообщения **ARPA** (**RFC-822**), который был разработан для обмена текстовыми сообщениями. **RFC-822** не дает возможностей включить в тело сообщения графику, аудио, видео и другие типы информации, а также текстовую информацию, которую нельзя реализовать 7-битовой кодировкой **ASCII**. Ограничения **RFC-822** становятся еще более очевидными, когда речь заходит об обмене сообщениями в разных почтовых системах.

**MIME** ориентирован на описание в заголовке письма структуры тела почтового сообщения и возможности составления письма из информационных единиц различных типов. В стандарте зарезервировано несколько способов представления разнородной информации. Для этого используются специальные поля заголовка почтового сообщения:

* поле версии **MIME**, используемое для идентификации сообщения, подготовленного в новом стандарте;
* поле описания типа информации в теле сообщения, позволяющее обеспечить правильную интерпретацию данных;
* поле типа кодировки информации в теле сообщения, указывающее на тип процедуры декодирования;
* два дополнительных поля, зарезервированных для более детального описания тела сообщения.

Стандарт **MIME** разработан как расширяемая спецификация, в которой подразумевается, что число типов данных будет расти по мере развития форм представления данных. При этом следует учитывать, что анархия типов (безграничное их увеличение) тоже недопустима. Каждый новый тип в обязательном порядке должен быть зарегистрирован в **IANA** (**Internet Assigned Numbers Authority**). Остановимся подробнее на форме и назначении полей, определяемых стандартом.

**Поле версии MIME (MIME - Version)** указывается в заголовке почтового сообщения и позволяет программе рассылки почты определить, что сообщение подготовлено в стандарте **MIME**. Формат поля выглядит так:

MIME - Version: 1.0

Поле версии указывается в общем заголовке почтового сообщения и относится ко всему сообщению целиком. В отличие от **RFC-822** стандарт **MIME** позволяет перемешивать поля заголовка сообщения с телом сообщения. Поэтому все поля делятся на два класса: *общие поля заголовка*, которые записываются в начале почтового сообщения, и *частные поля заголовка*, которые относятся только к отдельным частям составного сообщения и записываются перед ними.

**Поле типа содержания тела почтового сообщения (Content type)** используется для описания типа данных, которые содержатся в теле почтового сообщения. Это поле сообщает программе чтения почты, какого сорта преобразования необходимы для того, чтобы сообщение правильно проинтерпретировать. Эта же информация используется и программой рассылки при кодировании/декодировании почты. Стандарт **MIME** определяет семь типов данных, которые можно передавать в теле письма:

* текст (**text**);
* смешанный тип (**multipart**);
* почтовое сообщение (**message**);
* графический образ (**image**);
* аудиоинформация (**audio**);
* фильм или видео (**video**);
* приложение (**application**).

**Text**. Этот тип указывает на то, что в теле сообщения содержится текст. Основным подтипом типа **text** являются:

* + **plain** – соответствует *планарному* тексту;
  + **richtext** – соответствует *размеченному* текста, т.е. тексту со встроенными в него символами управления отображением;
  + **html** – соответствует *гипертекст*, т.е. тексту, который можно просматривать не последовательно, а произвольно, следуя гипертекстовым ссылкам;

**Richtext** определяет текст со встроенными в него специальными управляющими последовательностями, называемыми **тегами**, в соответствии со стандартом языка разметки документов **SGML** (**Standard Generalized Markup Language**). Теги представляют собой последовательность символов типа <*строка символов*>.

Разметка гипертекста строится по тому же принципу, что и в тексте типа **richtext**. При этом могут применяться теги, позволяющие описать гипертекстовые ссылки. К таким тегам относятся <А HREF="......">.....</А>.

**Multipart**. Этот тип почтового сообщения определяет смешанный документ, который может состоять из данных разного типа. Тип **multipart** имеет ряд подтипов:

* **mixed** – может создавать сообщения, состоящие из нескольких фрагментов, которые разделены между собой границей, задаваемой в качестве параметра подтипа;
* **alternative** – позволяет организовать просмотр почтового сообщения с возможностью выбора в зависимости от типа программы просмотра;
* **digest** – предназначен для многоцелевого почтового сообщения, когда различным частям хотят приписать более детальную информацию, чем просто тип;
* **parallel** – предназначен для составления такого почтового сообщения, части которого должны отображаться одновременно, что предполагает запуск сразу нескольких программ просмотра.

**Message**. Этот тип предназначен для работы с обычными почтовыми сообщениями, которые не могут быть переданы по разного рода причинам, объясняемым подтипами данного типа:

* **partial** – предназначен для передачи одного большого сообщения по частям. Атрибуты подтипа определяют *идентификатор сообщения* (**id**), *номер порции* (**number**) и *общее число порций* (**total**). Каждая часть имеет свое поле **Content type**, что означает – все сообщение может состоять из частей разных типов;
* **external body** – позволяет ссылаться на внешние относительно сообщения информационные источники;
* **rfc822** – определяет типы описания нетекстовой информации по стандарту **RFC-822**. Таких типов имеется четыре:
  + **image** – для описания графических образов (наиболее часто используются файлы форматов **GIF** и **JPEG**);
  + **audio** – для описания аудиоинформации (для воспроизведения сообщения данного типа требуется специальное оборудование);
  + **video** – для передачи видеоизображений (наиболее популярным является формат **MPEG**);
  + **application** – для передачи данных любого другого формата.

Обычно используется для передачи двоичных данных с последующим промежуточным преобразованием. Так, если на машине стоит видеокарта с 512 Кбайт памяти, а графика подготовлена в 256 цветах, то сначала ее следует преобразовать, и здесь может помочь тип **application**. Основной подтип данного типа - **octet stream**, но существуют **ODA** и **postscript**;

**Поле типа кодирования почтового сообщения (Content-Transfer-Encoding)** <pclass=just>. Многие данные передаются по почте в их исходном виде. Это могут быть 7-, 8-битные символы, 64-base символы и т.п. Однако при работе в разнородных почтовых средах необходимо определить механизм их представления в стандартном виде - **US - ASCII**. Для этого существуют процедуры кодирования такого рода данных. Наиболее широко применяемая - **uuencode**. Для того чтобы при получении данные были правильно распакованы, в стандарт введено поле **Content-Transfer-Encoding**. Синтаксис этого поля следующий:

Content-Transfer-Encoding:="BASE64"/"QUOTED-PRINTABLE"/"8ВГГ/ "7ВГГ/"BINARY"/x-token

Каждая из альтернатив применяется в подходящем случае. Альтернативы **8bit**, **7bit**, **BINARY** реально никакого преобразования не требуют, так как почта передается байтами и **SMTP** не делает различия между ними. Однако они введены для строгости описания типов. **BASE64** обычно используется в связке с типом **text/ ISO - 8859 - 1**. Элемент **x-token** позволяет пользователю описать свою процедуру преобразования.

**Дополнительные необязательные поля**. Стандарт определяет еще два дополнительных поля: **Content-ID** и **Content-Description**. Первое поле определяет уникальный идентификатор содержания, а второе служит для комментария. Ни то, ни другое программами просмотра обычно не отображаются.

Стандарт **MIME** позволяет расширить область применения электронной почты, обеспечить доступ к другим информационным ресурсам сети в стандартных форматах.

***Система World Wide Web***

В 1989 г. гипертекст представлял новую многообещающую технологию, которая имела относительно большое число реализаций, с одной стороны, а с другой стороны, делались попытки построить формальные модели гипертекстовых систем, которые носили скорее описательный характер и были навеяны успехом реляционного подхода описания данных. *Идея создания системы WWW заключалась в том, чтобы применить гипертекстовую модель к информационным ресурсам, распределенным в сети*, и сделать это максимально простым способом. Она заложила три краеугольных камня системы из четырех существующих ныне, разработав:

* язык гипертекстовой разметки документов **HTML** (*HyperText Markup Language*);
* универсальный способ адресации ресурсов в сети **URL** (*Universal Resource Locator*);
* протокол обмена гипертекстовой информацией **HTTP** (*HyperText Transfer Protocol*).

Позже команда **NCSA** добавила к этим трем компонентам четвертый:

* универсальный интерфейс шлюзов **CGI** (*Common Gateway Interface*).

Язык программирования **Java** не включается в этот список намеренно, так как область применения этого языка гораздо шире, чем простое «оживление» World Wide Web.

Идея **HTML** – пример чрезвычайно удачного решения проблемы построения гипертекстовой системы при помощи специального средства управления отображением. На разработку языка гипертекстовой разметки существенное влияние оказали два фактора:

1) исследования в области интерфейсов гипертекстовых систем;

2) желание обеспечить простой и быстрый способ создания гипертекстовой базы данных, распределенной в сети.

В 1989 г. активно обсуждалась проблема интерфейса гипертекстовых систем, т.е. способов отображения гипертекстовой информации и навигации в гипертекстовой сети. Значение гипертекстовой технологии сравнивали со значением книгопечатания. Утверждалось, что лист бумаги и компьютерные средства отображения/воспроизведения серьезно отличаются друг от друга, и поэтому форма представления информации тоже должна отличаться. *Наиболее эффективной формой организации гипертекста были признаны контекстные гипертекстовые ссылки*, а кроме того, было признано деление на ссылки, ассоциированные со всем документом в целом и с отдельными его частями.

Обычно гипертекстовые системы имеют специальные программные средства построения гипертекстовых связей. Сами гипертекстовые ссылки хранятся в специальных форматах или даже составляют специальные файлы. Такой подход хорош для локальной системы, но не для распределенной на множестве различных компьютерных платформ. В **HTML** гипертекстовые ссылки встроены в тело документа и хранятся как его часть. Часто в системах применяют специальные форматы хранения данных для повышения эффективности доступа. В **WWW**-документах это обычные **ASCII**-файлы, которые можно подготовить в любом текстовом редакторе. Таким образом, проблема создания гипертекстовой базы данных была решена чрезвычайно просто.

С момента разработки первой версии языка (**HTML 1.0**) он значительно развился: увеличилось число элементов разметки, оформление документов все больше приближается к оформлению качественных печатных изданий, развиваются средства описания нетекстовых информационных ресурсов и способы взаимодействия с прикладным программным обеспечением. Совершенствуется механизм разработки типовых стилей. Фактически, в настоящее время **HTML** развивается в сторону создания стандартного языка разработки интерфейсов как локальных, так и распределенных систем.

Вторым краеугольным камнем **WWW** стала универсальная форма адресации информационных ресурсов (*Universal Resource Identification*, **URI**), представляющая собой довольно стройную систему, учитывающую опыт адресации и идентификации **E-mail**, **Gopher**, **WAIS**, **Telnet**, **FTP** и т.п. Но реально из всего, что описано в **URI**, для организации баз данных в **WWW** требуется только **Universal Resource Locator** (**URL**). Без наличия этой спецификации вся мощь **HTML** оказалась бы бесполезной. **URL** используется в гипертекстовых ссылках и обеспечивает доступ к распределенным ресурсам сети. В **URL** можно адресовать как другие гипертекстовые документы формата **HTML**, так и ресурсы **E-mail**, **Telnet**, **FTP**, **Gopher**, **WAIS**. Различные программы различным образом осуществляют доступ к этим ресурсам. Следует отметить, что программы обработки электронной почты в формате **MIME** также имеют возможность отображать документы, представленные в формате **HTML**. Для этой цели в **MIME** зарезервирован тип **text/html**.

Третьим краеугольным камнем является протокол обмена данными в **World Wide Web** – **http** (*HyperText Transfer Protocol*). Данный протокол предназначен для обмена гипертекстовыми документами и учитывает специфику такого обмена. Так, в процессе взаимодействия клиент может получить новый адрес ресурса сети, запросить встроенную графику, принять и передать параметры и т.п. Управление в **HTTP** реализовано в виде **ASCII**-команд. Реально разработчик гипертекстовой базы данных сталкивается с элементами протокола только при использовании внешних программ или при доступе к внешним относительно **WWW** информационным ресурсам, например базам данных.

Спецификация **CGI** (*Common Gateway Interface*) была специально разработана для расширения возможностей **WWW** за счет подключения внешнего программного обеспечения. Эта технология соответствовала принципам простоты разработки, доступности и наращивания возможностей **WWW**. Предложенный и описанный в **CGI** способ подключения не требовал дополнительных библиотек и буквально ошеломлял своей простотой. Сервер взаимодействовал с программами через стандартные потоки ввода/вывода, что упрощает программирование до предела. При реализации **CGI** чрезвычайно важное место заняли методы доступа, описанные в **HTTP**. И хотя реально используются только два из них (**GET** и **POST**), опыт развития **HTML** показывает, что сообщество **WWW** ждет развития и **CGI** по мере усложнения задач, в которых будет использоваться **WWW**-технология.

**WWW** построена по известной схеме «клиент-сервер». Программа-клиент выполняет функции интерфейса пользователя и обеспечивает доступ практически ко всем информационным ресурсам Интернета. В этом смысле она выходит за обычные рамки работы клиента только с сервером определенного протокола. **Клиент** - это *интерпретатор* **HTML**, который в зависимости от команд (разметки) выполняет различные функции. В круг этих функций входит не только размещение текста на экране, но и обмен информацией с сервером по мере анализа полученного **HTML**-текста, что наиболее наглядно происходит при отображении встроенных в текст графических изображений. При анализе **URL**-спецификации или по командам сервера клиент запускает дополнительные внешние программы для работы с документами в форматах, отличных от **HTML**, например **GIF**, **JPEG**, **MPEG**, **Postscript** и т.п.

Другую часть программного комплекса **WWW** составляет *сервер протокола* **HTTP**, *базы данных документов* в формате **HTML**, управляемые сервером, и *программное обеспечение*, разработанное в стандарте спецификации **CGI**. До самого последнего времени реально использовалось два **HTTP**-сервера: сервер **CERN** и сервер **NCSA**. Но в настоящее время число базовых серверов расширилось.

**База данных HTML-документов** – это часть файловой системы, которая содержит файлы в формате **HTML** и связанные с ними графику и другие ресурсы.

Среди прикладного программного обеспечения, работающего с сервером, можно выделить *программы-шлюзы*. **Шлюзы** – это программы, обеспечивающие взаимодействие серверов различных протоколов, например, через **FTP**. Другие прикладные программы принимают данные от сервера и выполняют какие-либо действия: получение текущей даты, реализацию графических ссылок, доступ к локальным базам данных или просто расчеты.

Все, что было сказано до этого момента, можно отнести к классической схеме **World Wide Web**. В настоящее время следует говорить об ее изменении.

К середине 1996 г. произошли некоторые изменения в архитектуре сервиса **World Wide Web**. Произошел возврат к *модульной структуре сервера* **WWW**. Этот возврат был реализован в виде спецификации **API**. **API** – это спецификация разработки прикладных модулей, которые редактируются совместно с модулями сервера.

В дополнение к **HTML** активно стал применяться еще один язык разметки - **VRML** (*Virtual Reality Modeling Language*), а также язык **XML**.

Изменения коснулись и клиентской части технологии. В настоящее время происходит постепенный переход от простой классической архитектуры клиент-сервер к *архитектуре с сервером приложений*, в роли которого выступает *программа-клиент*. В частности, **NCSA** опубликовала спецификацию **CCI** (*Common Client Interface*) для разработки приложений.

Компоненты архитектуры **World Wide Web** существуют практически для всех типов компьютерных платформ и свободно доступны в сети. Любой, кто имеет доступ в Интернет, может создать свой **WWW**-сервер или, по крайней мере, посмотреть информацию с других серверов.

Для работы с гипертекстовыми документами используются специальные программы просмотра. Наиболее известными из них являются **Microsoft Internet Explorer** (встроен в последние версии операционных систем **Windows**), **Opera** (программа привлекает многих пользователей своей «легкостью» и удобством в работе), **Netscape Communicator** и др.

Программы, используемые для просмотра информации с серверов WWW, могут воспроизводить различные форматы. Первыми допустимыми форматами были **HTML**, **TXT** и два графических — **GIF** и **JPEG**. Практически только эти спецификации следует использовать при оформлении страниц информационного сервера.

Однако в качестве гиперссылки можно указать и документ, созданный в иной программе. Если программы просмотра поддерживают технологию **OLE**, то для показа документа после его загрузки на компьютер будет вызвана соответствующая программа.

Существует специальный формат представления документов, который широко используется в Интернете и средства просмотра которого полностью бесплатны, — **PDF** (*Portable Document Format*) — это файл, который содержит аппаратно независимое описание документа. Независимо от ОС, стоящей на компьютере, параметров устройств отображения, наличия шрифтов и т.п., пользователь увидит одно и то же оформление документа. Формат **PDF** использует в своей основе описание страниц документа на языке **PostScript**. **PDF**-документы просматриваются специальной программой — **Adobe Acrobat**, которая распространяется бесплатно и будет автоматически загружена из Интернета при попытке обращения к такому документу, если она не обнаружена на компьютере.

Основной недостаток использования **PDF**-файлов заключается в их большом объеме. Такие файлы долго загружаются из Сети, обычно их содержимое не включается в поисковые базы.

***Универсальный адрес ресурса***

Для того чтобы получить информацию из Интернета, необходимо знать *адрес*, по которому она расположена. **Универсальный адрес ресурса** (**URL**) – это адрес в системе WWW, с помощью которого однозначно определяется любой документ. В общем случае универсальный адрес ресурса имеет следующий формат:

**протокол://компьютер/путь**

**Протокол** является набором правил, согласно которому должна происходить передача данных. Основным протоколом в системе WWW является **протокол** **HTTP** - протокол передачи гипертекста, поэтому большая часть адресов начинается следующим образом:

**http://**

Тем не менее, могут быть использованы и другие протоколы передачи данных, например протокол передачи файлов - **FTP** или протокол передачи данных в формате **Gopher**. Тогда на первое место в универсальном адресе ресурса ставится название используемого протокола. Например:

**ftp://**

или

**gopher://**

**Компьютер** - это адрес сервера, с которым необходимо установить соединение. Может использоваться как **IP**-адрес, так и имя сервера в доменной системе имен. Например:

**http://www.econ.pu.ua**

или

**ftp://194.85.120.66**

Адреса большей части серверов в системе **WWW** начинаются с префикса **www**. Этот префикс используется просто как удобное обозначение того, что на данном компьютере запущен **Web**-сервер.

**Путь** представляет собой точное указание месторасположения документа на **Web**-сервере. Это может быть название директории и файла, как в следующем примере:

**http://www.econ.pu.ua/info/history/jubilee.htm**

Если ввести в строке «адрес» броузера данный адрес, броузер установит связь с компьютером **www.econ.pu.ua** по протоколу **HTTP** и запросит у него документ с названием **jubilee.htm** из каталога **/info/history**.

Последняя часть универсального адреса ресурса может включать *дополнительную информацию*, которую обычно используют для того, чтобы передать **Web**-серверу параметры запроса пользователя в интерактивных страницах, а также путь и имя той программы на сервере, которая этот запрос будет обрабатывать. Например:

**http://www.econ.pu.ua/sf/cgi-bin/main.bat?object=teachers&id=1**

Получив такой запрос, **Web**-сервер попытается найти программу **main.bat** в каталоге **/sf/cgi-bin/**, запустить ее и передать ей параметры **object** и **id** с соответствующими значениями.

В современных версиях броузеров нет необходимости указывать имя протокола в начале каждого адреса ресурса. Если имя протокола не указано, то броузер попытается самостоятельно определить, какой протокол необходимо использовать.

Если не указано имя файла, а только каталог, в котором он должен находиться, то пользователю будет передан файл, который администратор **Web**-сервера определил как файл, передаваемый по умолчанию. Обычно таковым является файл с названием **index.htm** (**index.html**) или **default.htm** (**default.html**). Если в каталоге нет файла по умолчанию, то будет выдано сообщение об ошибке.

Для указания документов на одном и том же сервере в **HTML**-документах часто используется сокращенное обозначение, называемое *относительным адресом*. Перед отправкой запроса на **Web**-сервер броузер преобразует относительный адрес в полный. Например, если документ по адресу **http://www.econ.pu.ua/info/index.htm** содержит ссылку на документ **history/jublilee.htm**, то броузер преобразует эту ссылку в **http://www.econ.pu.ua/info/history/jubilee.htm**.

***Язык разметки гипертекста (HTML)***

Большая часть документов в системе **World Wide Web** хранится в формате **HTML**. **HTML** – это язык гипертекстовой разметки, используемый для кодирования документов. Язык **HTML** представляет собой *набор команд*, в соответствии с которыми броузер отображает содержимое документа. Команды **HTML** не отображаются. В языке **HTML** реализован *механизм гипертекстовых ссылок*, который обеспечивает связь одного документа с другими. Эти документы могут находиться на том же сервере, что и страница, с которой на них делается ссылка, а могут быть размещены на другом сервере.

Команды в тексте **HTML**-документа называются **тегами**.

**HTML**-тег может содержать *список атрибутов*. Текст тега заключается в угловые скобки (< и >). В простейшем случае тег представляет собой *имя*, заключенное в угловые скобки, например **<TITLE>** или **<В>**. Для более сложных тегов характерно наличие атрибутов, которые могут иметь конкретные значения, определяющие функции тега. Например:

**<Р ALIGN="JUSTIFY">**

В данном случае **<Р>** - это тег, **ALIGN** - один из его атрибутов, a **"JUSTIFY"** - значение этого атрибута. (Данный тег определяет начало абзаца, атрибут **ALIGN** определяет способ выравнивания текста в абзаце, значение **"JUSTIFY"** задает выравнивание по ширине.)

Регистр символов в именах тегов не учитывается. Теги **<Р>** и **<р>** будут интерпретироваться одинаково. Однако в некоторых случаях важно использовать определенный регистр символов в значениях атрибутов.

Большинство **HTML**-тегов имеют парный *закрывающий тег*. Текст, к которому применяется форматирование, заключается между открывающим и закрывающим тегами. Закрывающий тег имеет имя, идентичное открывающему тегу, но перед именем закрывающего тега ставится косая черта. Например, чтобы отобразить текст курсивом, его нужно заключить в теги **<I>** и **</I>**:

**<I> Этот текст будет отображен курсивом</I>**

***Протокол передачи гипертекста (HTTP)***

**Протокол передачи гипертекста** (**HTTP**) – это стандартный протокол для передачи документов между серверами и броузерами в системе **WWW**. Протокол **HTTP** позволяет установить соединение между *клиентом* и *сервером*. Соединение сохраняется только на время обработки сервером запросов клиента. *Запрос клиента и ответ сервера образуют так называемую* **транзакцию**.

Обмен данными по протоколу **HTTP** происходит следующим образом. Клиент устанавливает соединение с сервером по указанному номеру порта. Если в качестве клиента выступает броузер, то номер порта указывается в **URL**-запросе. Если номер не указан, то по умолчанию используется порт 80. Затем клиент посылает запрос на документ, указывая **HTTP**-команду, адрес документа и номер версии **HTTP**. Например:

**GET /index.html НТТР/1.0**

Команды **HTTP**-клиента принято называть **методами**. Метод сообщает серверу о цели запроса. В данном примере используется метод **GET**, который запрашивает файл **index.html**, расположенный в корневом каталоге сервера, используя протокол **HTTP** версии 1.0. Другими часто используемыми методами являются методы **HEAD** и **POST**. **Метод HEAD** аналогичен **GET**, но запрашивает не содержимое файла, а информацию о нем. **Метод POST** позволяет разместить файл на сервере.

Кроме того, клиент может послать информацию, называемую *заголовком*, чтобы сообщить серверу дополнительную информацию о себе. В качестве такой информации может выступать имя и номер версии клиента, информация о типах данных, которые предпочтительны для клиента, и др. Например:

**User - Agent: Mozilla/4.6 [en] (Win98;I)**

**Accept: image/gif, image/x-xbitmap, image/jpeg, \*/\***

Заголовок завершается пустой строкой.

***Протокол передачи файлов FTP***

Для обеспечения перемещения данных между различными ОС, которые могут встретиться в Интернете, используется протокол **FTP** (*File Transfer Protocol*), работающий независимо от применяемого оборудования. Протокол обеспечивает способ перемещения файлов между двумя компьютерами и позволяет абоненту сети Интернет получить в свое распоряжение множество файлов. Пользователь получает доступ к различным файлам и программам, хранящимся на компьютерах, подключенных к сети.

Программа, реализующая этот протокол, позволяет установить связь с одним из множества FTP-серверов в Internet.

**FTP-сервер** - компьютер, на котором содержатся файлы, предназначенные для открытого доступа.

Программа **FTP**-клиент не только реализует протокол передачи данных, но и поддерживает набор команд, которые используются для просмотра каталога **FTP**-сервера, поиска файлов и управления перемещением данных.

Для установки связи с **FTP**-сервером пользователь должен ввести команду **ftp**, а затем адрес или его доменное имя.

Если связь установлена, появится приглашение ввести имя пользователя. Пользователь, не зарегистрированный на сервере, может представиться именем "*anonymus*" и получит доступ к определенным файлам и программам. Если будет запрошен пароль, можно ввести свой адрес электронной почты или свой IP-адрес. Поступившее после выполнения этих процедур приглашение позволяет работать с FTP-сервером.

Основной режим передачи файлов - передача в коде **ASCII**. Для передачи двоичных файлов необходимо ввести команду **binary***.* Для определения активного режима необходимо ввести команду **status***.*

Так как большинство FТР-серверов работает под управлением ОС **Unix**, то технология работы в этой системе требует введения команд из командной строки компьютера и несколько затрудняет действия пользователя в этом режиме. Однако для пользователей, привыкших к графическим интерфейсам Windows, весьма затруднительно работать в командной строке. Поэтому можно обратиться к любой программе, поддерживающей использование данного протокола. Так, по протоколу **FTP** может «работать» и **MS Internet Explorer**. Достаточно перед адресом сайта набрать используемый протокол следующим образом:

**ftp://ftp.server.ru/**

Однако **MS Internet Explorer** не очень удобен при работе с сайтами, требующими использования пароля для доступа к информации. Кроме того, с его помощью невозможно запланировать операцию копирования удаленного файла на компьютер на удобное время, автоматически возобновлять процесс при обрыве связи и т.п.

Такие файловые менеджеры, как **Total Commander**, **FAR** и др., также позволяют организовать соединение с **FTP**-сервером. Причем, например, **Total Commander** дает возможность ограничить скорость пересылки данных, что бывает полезно в условиях офиса (чтобы не монополизировать канал).

При интенсивной работе с серверами **FTP** следует установить одну из специализированных программ, например, **ReGet**, **Get-Right** и др. Эти программы позволяют запланировать необходимые загрузки (пользователь может составить список **URL** нужной информации, указать желаемое время, допустимое количество попыток соединения и т.п.), а также автоматически возобновлять загрузку файлов при обрыве соединения.

***Взаимодействие с другим компьютером (Telnet)***

**Telnet** обеспечивает взаимодействие с удаленным компьютером. Установив такую связь через **Telnet**, пользователь получает возможность работать с удаленным компьютером, как со «своим», т.е. теоретически получить в свое распоряжение все ресурсы, если к ним разрешен доступ. Реально **Telnet** предоставляет открытый доступ, но организация взаимодействия полностью определяется удаленным компьютером.

Два вида услуг Internet требуют подключения к серверам через **Telnet**: *библиотечные каталоги* и *электронные доски объявлений* (**BBS**).

Программа **Telnet** в использовании очень проста. Для установки с ее помощью связи с каким-либо компьютером, подключенным к сети, необходимо знать его полный адрес в Internet. При установлении соединения с нужным компьютером следует указать в команде его адрес, В процессе соединения хост-компьютер запрашивает имя пользователя. Для работы в удаленной системе пользователь должен иметь там права доступа. После успешного подключения к хост-компьютеру пользователь должен указать тип используемого терминала. Для удобства работы пользователя хост-компьютер обычно указывает ему способ вызова справочной информации.

Работа с удаленной системой может вестись в "прозрачном" режиме, когда программы на сервере и у клиента только обеспечивают протокол соединения, и в командном, когда клиент получает в свое распоряжение набор команд сервера.

Следует заметить, что из соображений безопасности намечается тенденция сокращения числа узлов Internet, позволяющих использовать Telnet для подключения к ним.

**Электронные доски объявлений (BBS).** Независимо от Internet существуют маленькие диалоговые службы, предоставляющие доступ к BBS (Bulletin Board System - система электронных досок объявлений).

Это компьютеры, к которым можно подсоединиться с помощью модемов через телефонную сеть. BBS содержат файлы, которые можно переписывать, позволяют проводить дискуссии, участвовать в различных играх и имеют свою систему электронной почты.

Самой крупной и известной системой электронных досок объявлений является система CompuServe. Она насчитывает около двух миллионов пользователей. Для расширения своих возможностей CompuServe подключается к Internet и предоставляет своим пользователям право доступа к службам Internet.

Несмотря на относительную дешевизну обслуживания, ни одна из диалоговых систем BBS не может дать пользователям тех возможностей, которые предоставляет Internet.

***Instant Messaging***

Системы обмена мгновенными сообщениями очень популярны среди пользователей Интернета. Данные программы предназначены для быстрой передачи от одного пользователя другому коротких текстовых сообщений.

Программы, которые позволяют пользователям «напрямик» общаться через Интернет, реально осуществляют доступ из Сети к данному локальному компьютеру. Тем самым в них заложена потенциальная опасность исполнения вредоносных кодов при наличии тех или иных недоработок в программе.

Наиболее известными программами обмена сообщениями являются **ICQ**, **Windows** **Messenger**, **Yahoo Messenger**.

Программа **ICQ** бесплатна. Обычно установочный комплект программы доступен на одном из ближайших **FTP**-серверов, так что не обязательно затрачивать средства на загрузку файлов с сервера **http://www.icq.com/**.

После установки программы необходимо зарегистрироваться и получить уникальный номер (регистрация автоматически запускается при первом вызове программы). Для заполнения предлагается много позиций, однако для получения номера достаточно определить только будущий пароль. Регистрационные данные используются (при наличии согласия пользователя) в справочной информации на сервере. Например, при поиске собеседника по интересам с учетом места жительства, возраста и т.п.

С помощью программы можно отправлять любому пользователю **ICQ** текстовые сообщения, файлы и т.п. Для этого нужно знать номер пользователя (или воспользоваться поиском на основе регистрационных данных). Постоянные пользователи могут быть включены в адресную книгу, что облегчает их быстрый вызов.

Программа **Miranda** позволяет обмениваться только сообщениями с пользователями **ICQ**, поэтому она намного меньше по объему, чем другие подобные программы.

**MSN Messenger** - программа обмена сообщениями, которая поставляется в составе операционных систем Windows. Одна из самых защищенных программ (с точки зрения пользователя) из систем обмена мгновенными сообщениями. Реализованы те же возможности, что и для программы **ICQ**. Основной «недостаток» — программа поддерживает свои списки пользователей, отличающиеся от списков **ICQ**. В результате вы не имеете возможности отправить сообщение абоненту **ICQ**.

При регистрации в качестве пользователя **MSN** создается **Microsoft Passport**, который может быть использован при аутенфикации пользователя как на страницах сайта корпорации Microsoft, так и на других серверах **WWW**, поддерживающих данную технологию.

***Internet Chat***

Существует множество программ, реализующих единый подход: на экране компьютера в окне, напоминающем текстовый редактор, вы можете вводить свой текст и читать сообщения любого количества участников такого «разговора».

Чтобы начать использовать данный сервис, следует установить соответствующее программное обеспечение (в основном бесплатное). После этого необходимо подключиться к одному из серверов, который обеспечивает этот сервис (обычно список содержится в программных файлах).

После выбора одной из тематик вы «попадаете» в комнату для переговоров, в которой высказываете свои мысли и читаете рассуждения других участников. Обычно такие «разговоры» ведутся под псевдонимами, хотя установление адреса реального пользователя принципиально возможно.

Для «оживления» бесед появилось много программ, в которых, например, можно «сконструировать» своего героя и путешествовать с ним по виртуальному миру. Часто можно определить для себя какой-либо образ, назначить ему тип голоса, включить музыку и т.п. Конечно, использование таких возможностей предъявляет повышенные требования к скорости канала связи.

***IP-телефония***

Под **IP-телефонией** понимается технология, позволяющая использовать Интернет или любую другую **IP**-сеть в качестве средства организации и ведения телефонных разговоров и передачи факсов в режиме реального времени. Существует техническая возможность оцифровать звук или факсимильное сообщение и переслать его аналогично тому, как пересылаются цифровые данные. И в этом смысле IP-телефония использует Интернет (или любую другую **IP**-сеть) для пересылки голосовых или факсимильных сообщений между двумя пользователями компьютера в режиме реального времени.

Общий принцип действия телефонных серверов **IP**-телефонии таков: с одной стороны, сервер связан с телефонными линиями и может соединиться с любым телефоном мира. С другой стороны, сервер связан с Интернетом и может связаться с любым компьютером в мире. Сервер принимает стандартный телефонный сигнал, оцифровывает его (если он исходно не цифровой), значительно сжимает, разбивает на пакеты и отправляет через Интернет по назначению с использованием протокола **TCP/IP**. Для пакетов, приходящих из Сети на телефонный сервер и уходящих в телефонную линию, операция происходит в обратном порядке. Обе составляющие операции (вход сигнала в телефонную сеть и его выход из телефонной сети) происходят в режиме реального времени.

Для того чтобы осуществить связь с помощью телефонных серверов, организация или оператор услуги должны иметь серверы в тех местах, куда и откуда планируются звонки. Стоимость такой связи на порядок меньше стоимости телефонного звонка по обычным телефонным линиям. Особенно велика эта разница для международных переговоров.

***Электронная коммерция***

**Электронная коммерция** подразумевает использование технологий глобальных компьютерных сетей для ведения бизнеса. Благодаря широкому распространению технологии WWW сеть Интернет из академической сети превратилась в популярную среду для общения, рекламы и бизнеса.

Электронная коммерция позволяет изменить практически все процессы, происходящие в современном бизнесе, интегрируя их в единое целое. Потребители могут искать, заказывать и оплачивать товары, используя Интернет, обмениваться информацией о товарах и услугах с другими пользователями. Правительственные организации могут использовать Интернет для сбора налоговых деклараций и распространения официальной информации.

Первоначально под термином электронная коммерция понимались *продажи товаров* и *переводы денежных средств* с помощью компьютерных сетей. Всякий раз, когда мы используем кредитную или телефонную карту, мы принимаем участие в электронной коммерции. Если банки осуществляют перевод денежных средств с помощью систем электронных платежей, они также используют электронную коммерцию. Однако банковские системы платежей являются закрытыми. Круг действия подобной системы ограничен банками, принимающими в ней участие, или клиентами конкретного банка. Между тем Интернет является открытой системой и представляет собой совершенной новый тип взаимодействия с пользователем. Поэтому сейчас термин «электронная коммерция» охватывает практически все аспекты ведения бизнеса, которые возможны с использованием Интернета. Можно выделить следующие два аспекта электронной коммерции:

1. **Электронная коммерция как торговля в сети Интернет –** продажа товаров и услуг с использованием Интернета.
2. **Электронная коммерция как рынок**. Электронная коммерция не ограничивается покупкой и продажей товаров в Интернете. Например, виртуальный магазин может не только предлагать свою продукцию пользователям Интернета, но и искать поставщиков продукции, заключать контракты с ними, оплачивать счета, нанимать сотрудников и проводить маркетинговые акции через Интернет. При этом компании, вовлеченные в процесс электронной торговли, могут даже не подозревать об этом. Электронная коммерция оказывает значительное влияние на процессы производства, распространения и обмена товарами, а также на то, каким образом потребитель получает информацию о товаре и производит торговую сделку.

***Системы электронных платежей***

Существуют две **основные области** применения систем электронных платежей:

1) обмен данными (заказы, счета и др.);

2) электронный перевод денежных средств, который осуществляется между банками и имеет достаточно большой объем и относительно небольшое количество передаваемых полей, а также долгосрочные связи.

Существует три **типа** электронных платежных средств:

1. *Комбинация обычных и электронных платежей.* Например, платеж осуществляется традиционным способом, а подтверждение владельцу высылается по электронной почте. Или, напротив, платеж делается электронным способом, а подтверждение приходит по обычной почте.
2. *Расширение традиционного способа передачи денежных средств.* Сюда входит передача номеров кредитных карточек электронным способом и использование смарт-карт, которые могут хранить и передавать всю информацию о своем владельце. В данном случае вся операция может происходить в электронном виде.
3. *Различные виды цифровой наличности и электронных денег.* Разница между первыми двумя типами и третьим заключается в том, что в последнем случае действительно переводятся деньги, а не только информация о них. Например, если передается только номер кредитной карточки - это платеж второго типа, а если сообщение само несет в себе некую сумму денег - это платеж третьего типа.

**Электронные деньги** – это некоторые зашифрованные серийные номера, представляющие реальную сумму денег. В то же время они являются вполне полноценными денежными средствами в том смысле, что могут быть обменены на обычные деньги.

**Смарт-карты** являются небольшими устройствами, которые могут хранить информацию, необходимую для совершения транзакций в рамках электронной коммерции. Смарт-карты могут хранить электронную наличность, информацию о владельце, электронные ключи и другую информацию.

**Микроплатежи** - это особый вид электронных денег, необходимый для оплаты микротранзакций. Стоимость одной транзакции может быть очень мала, и вследствие этого ее может быть невозможно выразить в обычных денежных единицах, поэтому возникла необходимость в микроплатежах. Важно отметить, что микроплатежи являются особенностью электронной коммерции.

**Служба Gopher.** Выполняет функции, аналогичные WWW. Вся информация на **Gopher**-сервере хранится в виде дерева данных (или иерархической системы меню). Начальный каталог **Gopher** является вершиной этого дерева, а все остальные каталоги и файлы представляются элементами меню. Строка главного меню представляет собой либо подменю, либо файл. **Gopher** поддерживает разные типы файлов - текстовые, звуковые, программные и т.д.

**Телеконференции Usenet.** Система **Usenet** была разработана для перемещения новостей между компьютерами по всему миру. В дальнейшем она практически полностью интегрировалась в Internet, и теперь Internet обеспечивает распространение всех ее сообщений. Серверы **Usenet** имеют средства для разделения телеконференций по темам.

**Телеконференции** – дискуссионные группы, входящие в состав **Usenet**. Телеконференции организованы по иерархическому принципу, и для верхнего уровня выбраны семь основных рубрик. В свою очередь, каждая из них охватывает сотни подгрупп. Образуется древовидная структура, напоминающая организацию файловой системы. Из числа основных рубрик следует выделить:

* **сотр** *-* темы, связанные с компьютерами;
* **sci** *-* темы из области научных исследований;
* **news** - информация и новости **Usenet**;
* **soc** *-* социальная тематика;
* **talk** *-* дискуссии.

Существуют, кроме того, специальные рубрики и региональное разделение телеконференций.

Управляют доступом к службе **Usenet** специальные программы, позволяющие выбирать телеконференции, работать с цепочками сообщений и читать сообщения и ответы на них. Эти программы выполняют такую функцию, как подписка на телеконференции. Если пользователь не вводит никаких ограничений, то по умолчанию производится подписка на все телеконференции, с которыми имеет связь его хост-компьютер. Программа также позволяет сделать тематический выбор и обеспечит пользователя сообщениями по интересующему его направлению.

При участии в какой-либо телеконференции любой абонент может направить свое сообщение по интересующей его теме.

Существуют два способа выполнения этой процедуры:

* посылка непосредственного ответа автору статьи по адресу его электронной почты;
* предоставление своего сообщения в распоряжение всех участников телеконференции.

Второй способ обозначается термином "*Follow-up*".

***Видеоконференции***

Перспективы использования глобальных компьютерных сетей практически безграничны. Одной из таких возможностей являются *видеоконференции*.

При организации видеоконференций по Интернету еще острее становятся проблемы виртуального канала реального времени для передачи видеокадров и звукового сопровождения. Поэтому на сегодняшний день, даже имея высокоскоростной канал связи, очень трудно добиться качества изображения, совместимого с телевизионным сигналом. Обычно можно передавать несколько кадров в секунду, тогда как минимальное значение для просмотра непрерывного движения — это 16 кадров в секунду.

Конечно, можно передавать и статические «картинки». Для этого необходимо иметь видеокамеру с передачей информации на компьютер, стандартную звуковую плату с микрофоном.

В последнее время одна из наиболее удачных программ для организации таких видеоконференций — это составная часть полной установки программы **Microsoft Internet Explorer**, которая называется **Microsoft NetMeeting**.

**NetMeeting** позволяет организовать сетевое общение: передавать видеоизображения, организовывать беседы (чат), совместно использовать программы, установленные на локальном компьютере, пересылать по Сети файлы и даже рисовать вместе на одном «листе».

Программа использует **MSN Messenge**r для взаимодействия с сервером, поиска контактов и т.п. Естественно, что для передачи изображений необходимо иметь подключенную видеокамеру, а для разговора — звуковую плату, микрофон и колонки.

Для связи с другими пользователями необходимо ввести их параметры (адрес), можно использовать учетные записи с сервера **Microsoft Directory**. В случае необходимости организации собственного вещания для многих пользователей следует установить специальное серверное программное обеспечение (**NetMeeting Server**).