**Лекция №1 Введение к сетевым технологиям**

**1.1 Основные понятия**

**Коммуникационная cеть** - система, состоящая из объектов, осуществляющих функции генерации, преобразования, хранения и потребления продукта, называемых пунктами (узлами) сети, и линий передачи (связей, коммуникаций, соединений), осуществляющих передачу продукта между пунктами.

**Информационная сеть** - коммуникационная сеть, в которой продуктом генерирования, переработки, хранения и использования является информация.

**Компьютерная сеть** - это совокупность программных, аппаратных и коммуникационных средств, которые могут обмениваться между собой информацией.

**Компоненты компьютерной сети**:

* компьютер;
* **линии связи** (коммуникационное оборудование) - техника, которая реализует возможность обмена информацией (провода, устройства, иногда - компьютер, выполняющий функцию коммуникационного оборудования);
* операционная система, в особенности, модули ОС, реализующие сетевое взаимодействие;

**Классификация компьютерных сетей по территории**

- **Local Area Network (LAN)** — сеть, имеющая небольшой размер и замкнутую инфраструктуру, основное назначение которой состоит в объединении пользователей (как правило, одной компании или организации) для совместной работы.

- **Metropolian Area Network (MAN)** сеть между учреждениями в пределах одного или нескольких городов, связывающие много локальных вычислительных сетей. Как правило, поддерживают передачу данных в голосовом формате. Иногда объединяется с кабельной ТВ-сетью. Скорость – 10-100 Мб/с. Протяженность – от нескольких километров до сотен километров. - **Региональные** — объединяют компьютеры географической области

( Рунет — региональная сеть).

- **Wide Area Network (WAN)** сеть, охватывающая крупные географические области – страны, континенты. Включает в себя как локальные сети, так и прочие телекоммуникационные сети и устройства. Глобальные сети являются открытыми и ориентированы на обслуживание любых пользователей. Работают на относительно низких скоростях (до 3 Мбит/с) и могут вносить значительные задержки в передачу информации. Пример возможности организации WAN – сети с коммутацией пакетов (Frame relay), через которую могут взаимодействовать между собой различные компьютерные сети. **Распределенные приложения** - программы, которые работают одновременно на разных компьютерах, например, WWW.

**Сервер** - компьютер или программа, предоставляющая некоторые услуги.

**Сервер** – компьютер с выходом в интернет, который обрабатывает запросы клиентов. Обычно находится в состоянии «прослушивания» и обрабатывает запросы только по инициативе клиента.

**Выделенный сервер** - это компьютер, служащий только для обслуживания клиентских машин. Доступ к нему обычно получают администраторы системы для выполнения задач управления, мониторинга и поддержки работоспособности. Часто они выполняют только какую-то одну задачу.

 **Домен (domain)** определенная администратором сети совокупность компьютеров, использующих [единую базу данных и систему защиты](http://www.4stud.info/networking/directory-service.html); каждый домен имеет уникальное имя.

**DNS (Domain Name System, система доменных имен) –** система наименования компьютеров и сетевых служб, организованных в виде

иерархии доменов. Правила наименования DNS используются в сетях TCP/IP, таких как Интернет, для обнаружения компьютеров и служб по именам, удобным для пользователей. Когда пользователь вводит в приложении имя DNS, службы DNS могут сопоставить это имя с другими сведениями, например с IP-адресом.

Большинство пользователей предпочитает использовать для поиска компьютера, являющегося почтовым сервером или веб-сервером сети, запоминающееся имя, с соответствующей смысловой нагрузкой. Компьютеры при связи по сети используют числовые адреса. Чтобы облегчить использование сетевых ресурсов, службы имен, такие как DNS, обеспечивают сопоставление понятного имени компьютера или службы с его числовым адресом.

В соответствии c иерархической структурой в общем виде символьное имя ресурса представляет собой доменное имя, состоящее из отдельных элементов – доменов, разделяемых при записи точками: рис. 1.



Рис.1 – Структура доменного имени

**Узел** – любой компьютер, подключенный к Интернету, имеет свой уникальный IP-адрес, иначе говоря – любая сущность, имеющая уникальный IP-адрес в Интернете. Нет адреса – нет узла. Интернет можно рассматривать как множество узлов, который каждый из которых потенциально может связаться с любым другим.

**Хост** – с точки зрения пользователя тоже что и узел. В общем-то, оба понятия очень смешиваются. Но хост – совсем не обязательно отдельный узел, если это виртуальный хост. Но хост – совсем не обязательно отдельный узел, если это виртуальный хост. Часто хост имеет собственное уникальное доменное имя. Фактически все, что отличает хост от узла – это то, что он может быть виртуальным. Итак любой узел – хост, но не любой хост – узел.

**Виртуальный хост** – это хост не имеющий уникального IP-адреса в сети, но, тем не менее, доступный указанием какого-нибудь дополнительного адреса(например, его DNS-имени)

**Провайдер** – организация, имеющая несколько модемных входов, к которым могут подключаться для доступа в интернет. Все это обычно происходит не бесплатно для пользователей.

**Хостинг-провайдер (хостер)** – это организация, которая может создавать хосты(виртуальные или обычные) в интернете и продавать их различным клиентам, обычно за определенную плату.

**Хостер** – это услуги, которые предоставляют клиентам Хостинг-провайдер.

**Клиент** - это компьютер или программа, запрашивающая услуги.

**Порт** – условный номер ассоциируемый с определенным ПО, которое устанавливается на сервере:

htth - 80

https – 443

pop – 21

smtp – 110

MySQL сервер - 3306

**Протокол** - это набор семантических и синтаксических правил, определяющий поведение функциональных блоков сети при передаче данных. Другими словами, протокол - это совокупность соглашений относительно способа представления данных, обеспечивающего их передачу в нужных направлениях и правильную интерпретацию данных всеми участниками процесса информационного обмена.

**Протокол** определяет:

* формат сообщений;
* очередность сообщений;
* действия, которые необходимо выполнять при получении, приеме сообщений или при наступлении иных событий.

Существует несколько стандартных наборов (стеков) протоколов, получивших сейчас наиболее широкое распространение:

**- TCP/IP;**

**- SPX/IPX;**

**- Novell NetWare;**

**- AppleTalk и т.д.**

**Протоколы перечисленных стеков делятся на три основных типа:**

- прикладные протоколы (выполняющие функции прикладного, представительского и сеансового уровней модели OSI);

- транспортные протоколы (выполняющие функции

- транспортного и сеансового уровней модели OSI);

- сетевые протоколы (выполняющие функции трех нижних уровней OSI – сетевого, канального и физического). **Узел (host)** - подключенное к сети устройство (обычно компьютер), идентифицируемое собственным [адресом](http://www.4stud.info/networking/work1.html) (например, ip-адрес - это уникальное в пределах сети 32-разрядное двоичное число)

**Трафик (traffic)** - поток сообщений в разделяемой среде передачи данных, часто используется для грубой оценки уровня использования передающей среды (тяжелый, средний, легкий трафик).

**Пропускная способность канала связи** (bandwidth) — наибольшая скорость передачи информации по каналу связи. Измеряется числом передаваемых двоичных символов в 1 с. Скорость передачи зависит от физических свойств канала связи, статистических свойств помех, способа передачи, приема сигналов и др.

**По типу функционального взаимодействия ПК можно выделить:**

**Клиент-сервер** – сетевая архитектура, в которой устройства являются либо клиентами (front end), либо серверами (back end). Клиентом является запрашивающая машина (обычно ПК), сервером - машина, которая отвечает на запрос. Термины клиент и сервер могут применяться к физическим устройствам и к программному обеспечению.

**Одноранговые сети** (P2P, Peer-to-Peer Network – «равный к равному») – сетевая архитектура, основанная на равноправии участников. В таких сетях отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и сервером. В отличие от архитектуры клиент-сервер, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов.

**Смешанные (гибридные) сети** – сетевая архитектура, в которой выделены сервера для координации работы, поиска или предоставления информации о существующих машинах сети и их статусе (on-line, off-line и т.д.). Гибридные сети сочетают скорость централизованных сетей и надёжность децентрализованных благодаря гибридным схемам с независимыми индексационными серверами, синхронизирующими информацию между собой. При выходе из строя одного или нескольких серверов, сеть продолжает функционировать.

# 1.2 Архитектура клиент-сервер

Сервером определенного ресурса в компьютерной сети называется **компьютер** (программа), управляющая этим ресурсом, **клиентом** - компьютер (программа), использующий этот ресурс. В качестве ресурса компьютерной сети могут выступать, базы данных, файловые системы, службы печати, почтовые службы. Тип сервера определяется видом ресурса, которым он управляет. Например, если управляемым ресурсом является база данных, то соответствующий сервер называется сервером базы данных.

Структура распределенной ИС, построенной по архитектуре клиент-сервер с использованием сервера баз данных, рассматривается на рис.1. При такой архитектуре сервер базы данных обеспечивает выполнение основного объема обработки данных. Формируемые пользователем или приложением запросы поступают к серверу базы данных в виде инструкции языка SQL. Сервер базы данных выполняет поиск и извлечение нужных данных, которые затем передаются на компьютер пользователя. Достоинством такого подхода в сравнении с файл-сервером является заметно меньший объем передаваемых данных.

Корпоративная база данных создается, поддерживается и функционирует под управлением сервера баз данных, например Microsoft SQL Server. В зависимости от размеров организации и особенностей решаемых задач ИС может иметь одну из следующих конфигураций: компьютер-сервер, содержащий корпоративную и персональную базы; компьютер-сервер и персональные компьютеры с ПБД; несколько компьютеров-серверов и персональных компьютеров с ПБД.

Использование архитектуры клиент-сервер дает возможность постепенного наращивания ИС предприятия, во-первых, по мере развития предприятия; во-вторых, по мере развития самой ИС.

Разделение общей базы данных на корпоративную и персональные позволяет уменьшить сложность проектирования баз данных по сравнению с централизованным вариантом, а значит снизить вероятность ошибок при проектировании и стоимость проектирования.

Важнейшим достоинством применения базы данных в ИС является обеспечение независимости данных от прикладных программ, это дает возможность пользователям не заниматься проблемами представления данных на физическом уровне: размещение данных в памяти, методов доступа к ним.

Такая независимость достигается поддерживаемым СУБД многоуровневым представлением данных в базе данных на логическом (пользовательском) и физическом уровнях. Благодаря СУБД и наличию логического уровня представления данных обеспечивается отделение концептуальной (понятийной) модели базы данных от ее физического представления в памяти ЭВМ. Важнейшим параметром крупной информационной системы является быстродействие при значительном количестве пользователей, а также надежность, масштабируемость и безопасность. Всё это обеспечивает архитектура **"клиент-сервер"**. Такая архитектура позволяет оптимально распределить работу между клиентскими и серверной частями системы: теперь приложение, работающее на рабочей станции, не читает записи базы данных "напрямую", а посылает запросы на сервер, где они принимаются и последовательно отрабатываются специальными программами. В результате на рабочую станцию поступают только обработанные данные, что радикально сокращает информационные потоки в ЛВС.

****

Рис.2 – Архитектура клиент-сервер

Протоколы передачи данных

На прикладном уровне и уровне представления данных модели передачи данных OSI работает множество протоколов, каждый из которых решает свою прикладную задачу. Примерами таких протоколов являются:

* HTTP (***Hyper Text Transfer Protocol***) – протокол, по которому передаются пакеты, содержащие HTML-страницы и прочие WEB-ресурсы;
* FTP (***File Transfer Protocol***) – протокол пересылки файлов;
* SMTP (***Simple Message Transfer Protocol***) - протокол пересылки почтовых сообщений;
* POP3 (***Post Office Protocol***) – почтовый протокол взаимодействия почтового клиента и сервера;
* Telnet и SSH – протоколы для удаленной работы с компьютером;
* SNMP (***Simple Network Manage Protocol***) – протокол управления сетевыми устройствами;
* и т.д.

**Hyper Text Transfer Protocol (HTTP)** – это протокол прикладного уровня, который обеспечивает пересылку, модификацию и прочие действия с разнообразными ресурсами (данными). Для указания ресурсов, к которым применяются упомянутые действия, используется механизм ссылок. Ссылка – это идентификатор ресурса, который определяет его расположение. Структура такого идентификатора (URL – Universal Resource Locator) следующая:

**http://domain:port/directory/file**,

где

http – тип протокола;

domain – доменное имя;

port – номер порта (стандартным для HTTP является порт 80);

directory – каталог, в котором расположен ресурс;

file – файл, в котором расположен ресурс.

****

Рис.2 – модель обмена данных по протоколу http

В http.conf хранится информация о том какое по необходимости задается в ответ на запрос клиента. Для вычисления этого используется номер порта и расширение скрипта, который запрашивает клиент.

**1.3 Структура HTTP запроса**

В таблице 1 рассмотрены основные методы протокола HTTP.

Таблица 1 - Назначение основных методов протокола HTTP

|  |  |
| --- | --- |
| Метод | Назначение |
| GET | Метод служит для получения произвольной информации, которая указана идентификатором. Если URL ссылается на некоторый процесс, то в качестве ответа будут выступать результаты работы процесса. Возможна ситуация, когда данный метод заменяется на условный GET. В таком запросе в заголовке запроса должно быть поле “If-Modified-Since”. Если ресурс не изменялся с указанной даты, то будет возвращено специальное сообщение, в противоположном случае ответ будет аналогичен ответу на обычный запрос GET. |
| HEAD | Метод аналогичный методу GET, но при ответе возвращаются лишь служебные поля ответа, а содержание (сам ресурс) не возвращается. Данный метод может быть использован для получения информации о ресурсе без передачи самого ресурса. |
| PUT | Данный метод дает указание серверу создать новый ресурс или сохранить информацию из запроса в виде ресурса, на который указывает URL. |
| POST | Метод используется для того, чтобы сервер принял информацию из поля “Содержание запроса” как входную. Данная информация может быть использованная как входная информация процессов, программ (скриптов) и как новая информация для добавления в базы данных и т.п. информационные структуры. Реальная функция метода определяется сервером и типом ресурса, на который указывает идентификатор. |
| DELETE | Метод используется для уничтожения ресурса, указанного с помощью URL.  |
| И т.д. |  |

Любой http запрос состоит из двух обязательных поле: заголовок и тело, которое разделяется 2 символами и перевода строки (\n).

Заголовок состоит из команды и атрибутов команды GET и POST + атрибуты.

А

т

р

и

б

у

т

а

З

А

Г

О

Л

О

В

О

к

GET http://www.yandex.ru/news.html HTTP/1.0\r\n

Content-type: text\HTML application\ms

Host: www. yandex.ru\r\n

Content-length:55(обязательно для post запросов)

Cookie: имя = значение

\n

\n

Т

е

л

о

<HTML>

 .

 .

 .

</HTML>

Структура пост запроса

з

а

г

о

л

о

в

о

к

а

т

р

и

б

у

т

а

\n

\n

POST

Cхема передачи данных из html форм методом post

<form metod-post>

 <input type = text name = l>

 <input type = chexBox name = l>

 <input type = text submit = l>

**1.4 Контрольные вопросы**

1. Что такое Компьютерная сеть?
2. Что такое cервер?
3. Что такое **домен**?
4. Что такое DNS?
5. Что такое Хостинг-провайдер?
6. Клиент это?
7. Какие протоколы передачи данных вы знаете.
8. Что значит HTTP протокол?
9. Что значат методы GET и POST?