Тема 2. Теоретические основы сетевых технологий. Сетевые стандарты и протоколы.

Общие принципы моделирования сетей. Понятие открытой системы.

Телекоммуникационные и вычислительные сети

2.1. Общие понятия, терминология

**Компьютерная сеть или телекоммуникационная вычислительная сеть** представляет собой сеть обмена и распределенной обработки информации, образуемая множеством взаимосвязанных абонентских систем и средствами связи; средства передачи и обработки информации ориентированы в ней на коллективное использование общесетевых ресурсов – информационных, программных, аппаратных.

**Абонентская система** – совокупность абонента-пользователя (объекта, генерирующего и потребляющего информацию) и рабочей станции.

**Рабочая станция** – система оборудования конечного пользователя сети, включающая персональный компьютер (терминал), средства связи с коммуникационной подсетью компьютерной сети, выполняющие прикладные процессы.

**Телекоммуникационная система** – это совокупность физической среды передачи информации, аппаратных и программных средств, обеспечивающих взаимодействие абонентской системы.

**Прикладной процесс** – это различные процедуры ввода, хранения, обработки и выдачи информации, выполняемые в интересах пользователей и описываемые прикладными программами.

Компьютерные сети решили две очень важные проблемы: обеспечение в принципе неограниченного доступа к ПК пользователей независимо от территориального расположения и возможность оперативного перемещения больших массивов информации на любые расстояния, позволяющая своевременно получать данные для принятия тех или иных решений.

2.2. Аппаратные и программные компоненты сети

На тех компьютерах, ресурсы которых должны быть доступны всем пользователям сети, необходимо добавить модули, которые постоянно будут находиться в режиме ожидания запросов, поступающих по сети от других компьютеров. Обычно такие модули называются программными серверами, так как их главная задача - обслуживать запросы на доступ к ресурсам своего компьютера. На компьютерах, пользователи которых хотят получать доступ к ресурсам других компьютеров, также нужно добавить к операционной системе некоторые специальные программные модули, которые должны вырабатывать запросы на доступ к удаленным ресурсам и передавать их по сети на нужный компьютер. Такие модули обычно называют программными клиентами. В современные популярные операционные системы для ПК, все необходимые программные модули для сетевых подключений уже интегрированы.

Модель взаимодействия «клиент-сервер» образуется, когда на устройстве, с которым непосредственно взаимодействует пользователь, начинает выполняться некоторая предварительная обработка информации.

**Сервер (программное обеспечение)** – программное обеспечение, принимающее запросы от клиентов.

**Сервер (аппаратное обеспечение)** – компьютер (или специальное компьютерное оборудование), выделенный и/или специализированный для выполнения определенных сервисных функций.

**Клиент** – аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.

По способу взаимодействия серверов и клиентов определяют два вида сетей: «клиент-сервер» (client-server) и «равный с равным» (peer-to-peer).

Часто модели «клиент-сервер» и «равный с равным» могут одновременно существовать в одной сети. Сети, построенные по принципу «равный с равным», называют также одноранговыми сетями, в которых все компьютеры имеют одинаковый статус – ранг.

Как отмечалось ранее, модели «клиент-сервер» и «равный с равным» могут одновременно существовать в одной сети. Это стало возможным благодаря различным сетевым компонентам, важнейшими из которых можно назвать средства организации канала передачи данных между клиентами и серверами сети. В простейшем случае канал передачи данных строится с использованием двух компонентов:

**среды передачи данных** (проводная или беспроводная - wire или wireless), обеспечивающей доставку информации от одного узла сети к другому;

**сетевых интерфейсных карт** (network interface card, NIC), обеспечивающих взаимодействие компьютера со средой передачи данных.

Сетевые аппаратные и аппаратно-программные средства можно разделить на следующие группы по их основному функциональному назначению:

1. пассивное сетевое оборудование - соединительные разъёмы, кабеля, патч-корды, патч-панели, информационные розетки, и т.п.;

2. активное сетевое оборудование - преобразователи (adapters), модемы (modems), повторители (repeaters), мосты (bridges), коммутаторы (switches), маршрутизаторы (routers), и т.п.

Проводные среды передачи информации создаются с использованием кабельных соединений на основе либо металлических проводников электрических сигналов, либо волоконно-оптических проводников световых сигналов.

При создании сетей передачи данных чаще всего используют именно проводные среды передачи информации.

Беспроводные среды передачи информации предусматривают организацию взаимодействия между компьютерами посредством передачи радиочастотных сигналов.

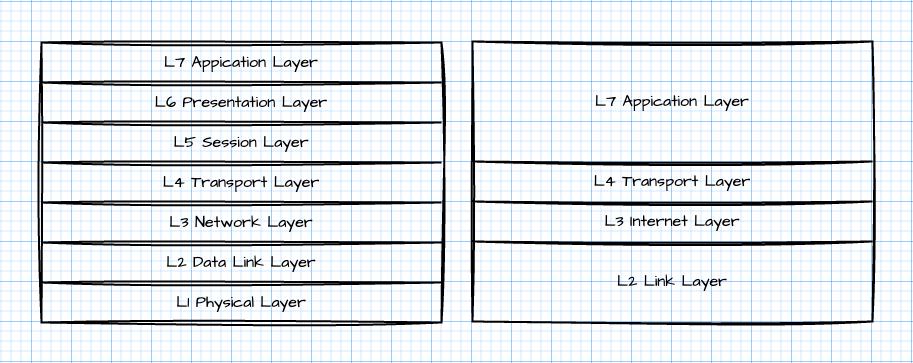
В зависимости от охвата территории компьютерные сети бывают:

1. Персональные — Personal Area Network (PAN).
2. Локальные — Local Area Network (LAN).
3. Городские — Metropolitan Area Network (MAN).
4. Глобальные — Wide Area Network (WAN).

Различные датчики, подключённые к смартфону, образуют сеть PAN. Компьютерная сеть из устройств, подключённых к вашему домашнему роутеру, является LAN-сетью, сеть из абонентов провайдера в городе — это MAN-сеть, а весь интернет, который вам предоставляет провайдер — WAN-сеть.

Сетевые модели

Под сетевой моделью понимаются основы, которые стандартизируют сетевое взаимодействие. Это основные термины, а также назначение и функции сетевых компонентов. Сетевая модель разделяет сетевые компоненты и их функции на уровни (слои (layers)). Каждый слой сетевой модели имеет определённое назначение и функции.  
Сейчас наиболее распространены две сетевых модели. Это семиуровневая OSI-модель и четырехуровневая TCP/IP-модель.

  
Рис 1. Схема моделей и как они соотносятся друг с другом.

Большинство действующих стандартов интернета и протоколов TCP/IP регламентируются документами Request For Comments (RFC). Детально сетевая модель TCP/IP рассмотрена в [RFC 1122 (Requirements for Internet Hosts — Communication Layers )](https://datatracker.ietf.org/doc/html/rfc1122) и [RFC-1123 (Requirements for Internet Hosts — Application and Support)](https://datatracker.ietf.org/doc/html/rfc1123). Модель объясняется и расширяется другими RFC.

## **Cтек протоколов TCP/IP**

Существует ряд протоколов, на которых всё основывается:

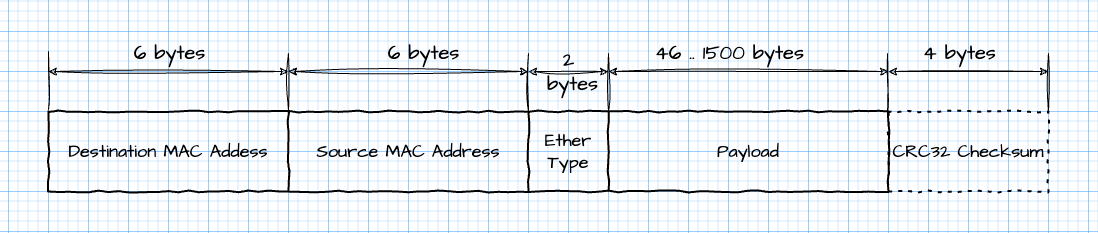
* Ethernet II;
* IP — Internet Protocol;
* ICMP — Internet Control Management Protocol;
* UDP — User Datagram Protocol;
* TCP — Transmission Control Protocol;
* DHCP — Dynamic Host Configuration Protocol;
* DNS — Domain Name Service.

Данные передаются порциями, которые называются Protocol Data Unit (PDU). PDU состоит из заголовка (header) и полезных данных (payload). PDU одного протокола в полезных данных могут содержать PDU другого протокола. Это называется инкапсуляцией. В зависимости от уровня, на котором работает сетевой протокол, PDU могут называться по-разному:

* на канальном уровне — фрейм;
* на сетевом уровне — пакет (IP, ICMP);
* на транспортном уровне — сегмент или датаграмма (TCP, UDP);
* на прикладном уровне — сообщение (DNS, DHCP).

## **Протокол Ethernet**

Ethernet-фрейм, который передаётся или принимается драйвером сетевого адаптера, состоит из заголовка и полезных данных.



 Ethernet-фрейм содержит MAC-адрес получателя, MAC-адрес отправителя, тип фрейма и сами данные.

Протокол IP

Оригинальное описание протокола находится в [RFC 791 Internet Protocol — DARPA Internet Program Protocol Specification](https://datatracker.ietf.org/doc/html/rfc791).

Протокол ICMP

Описание протокола приведено в [RFC 792 (Internet Protocol DARPA Internet Program Protocol Specification)](https://datatracker.ietf.org/doc/html/rfc792). ICMP причисляют к тому же уровню, что и IP — сетевому.

Протокол ARP

Протокол ARP используется для определения МАС-адреса физического интерфейса хоста по его IP-адресу. Описание протокола приведено в [RFC 826 — An Ethernet Address Resolution Protocol](https://datatracker.ietf.org/doc/html/rfc826).

Протокол UDP

Описание протокола приведено в [RFC 768 (User Datagram Protocol)](https://datatracker.ietf.org/doc/html/rfc768).  
  
Протокол позволяет двум процессам обмениваться UDP-датаграммами. Каждая UDP-датаграмма содержит в себе порт отправителя (Source Port), порт получателя (Destination Port), длину дейтаграммы (Length), контрольную сумму (Checksum) и собственно сами передаваемые данные.

При расчёте контрольной суммы добавляется псевдозаголовок, который не передаётся, а только участвует в расчёте контрольной суммы.  
  
Протокол используется в качестве транспортного протокола там, где на транспортном уровне допускается дублирование получаемых данных, пропуск данных или не важен порядок, в котором данные будут доставлены.  
  
Как правило, обработка этих случаев возлагается на протоколы уровня приложений или не осуществляется вовсе. Например, в потоковом видео или аудио данные пропускаются, так как повторная передача данных является в этом случае бессмысленной. Но если вы хотите гарантированную доставку данных на транспортном уровне, то вам необходимо использовать протокол TCP.

### Протокол TCP

Протокол TCP — самый сложный из всех, приведённых в статье. Назначение протокола TCP — создать надёжное виртуальное полнодуплексное соединение между процессами. На данный момент самое свежее описание протокола приведено в [RFC 9293 — Transmission Control Protocol (TCP)](https://datatracker.ietf.org/doc/html/rfc9293).

#### **▍ Структура сообщений**

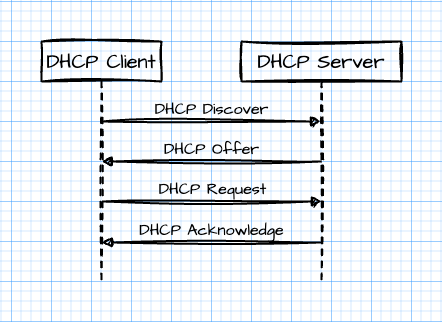
Сообщения, используемые в протоколе TCP, называются TCP-сегментами. Просьба не путать с сегментами сети. Они с ними не имеют ничего общего. При расчёте контрольной суммы для TCP-сегмента как и в UDP используется псевдозаголовок. Но если для UDP расчёт контрольной суммы не является обязательным, то для TCP он обязателен.

### Протокол DHCP

Для работы в сети TCP/IP хост необходимо настроить. Минимально необходимо указать его IP-адрес и маску подсети. Также может понадобиться указать адрес шлюза и адрес DNS-сервера. Протокол DHCP позволяет хосту получить эти данные автоматически из сети.  
Существуют различные варианты использования данного протокола, но мы рассмотрим основной успешный сценарий получения IP-адреса хостом, который состоит из обмена 4 сообщениями.

#### **▍ Получение конфигурации**

1. Изначально хост не имеет IP-адреса и не знает, где расположен DHCP-сервер, который ему эту информацию может предоставить. Поэтому он посылает широковещательное сообщение DHCP Discover в свой сегмент сети.  
  
2. Если в сети присутствует DHCP-сервер, он отвечает unicast-сообщением DHCP Offer, в котором содержится предлагаемая конфигурация для хоста.  
  
3. Хост посылает unicast-сообщение DHCP Request, в котором указывает, назначенный ему IP-адрес  
  
4. Сервер отвечает unicast-сообщением DHCP Acknowledge, которое говорит о том, что конфигурация хосту назначена.  
  
Диаграмма последовательностей приведена ниже.



Описание протокола приведено в [RFC 2131 — Dynamic Host Configuration Protocol](https://datatracker.ietf.org/doc/html/rfc2131).

Протокол DNS

Протокол DNS регламентируется [RFC 1035 DOMAIN NAMES — IMPLEMENTATION AND SPECIFICATION](https://datatracker.ietf.org/doc/html/rfc1035).  
  
Служба DNS используется для преобразования доменного имени хоста в его IP-адрес. IP-адрес хоста — это лишь часть той информации, которую может хранить DNS.  
Программы типа nslookup и функции в Winsock или glibc запутывают в понимании DNS. Провести некоторые эксперименты можно с утилитой dig и анализа трафика (https://www.diggui.com/). DNS — это распределённая иерархическая база данных доменов.  
Чтобы убедиться, что это действительно база данных можете зайти [на сайт](https://trendoceans.com/dns-toys/) и увидеть тому подтверждение.

Понятие открытой системы.

Открытость можно рассматривать на разных уровнях иерархии программного и аппаратного обеспечения системы или ее составных частей. Открытыми, например, могут быть:

* физические интерфейсы, протоколы обмена, методы контроля ошибок, системы адресации, форматы данных, типы организации сети, интерфейсы между программами, диапазоны изменения аналоговых сигналов;
* пользовательские интерфейсы, языки программирования контроллеров, управляющие команды модулей ввода-вывода, языки управления базами данных, операционные системы, средства связи аппаратуры с программным обеспечением;
* конструкционные элементы (шкафы, стойки, корпуса, разъемы, крепежные элементы);
* системы, включающие в себя перечисленные выше элементы.

Под открытостью системы иногда понимают ее соответствие современным промышленным стандартам, которое обеспечивает возможность интеграции с другими открытыми системами. Однако понятие открытости нужно трактовать шире: оно должно подразумевать, что система не только удовлетворяет стандартам, но стандарт является *общепризнанным*, а *в свободной продаже* имеются аналогичные системы*других производителей* по *конкурентоспособным ценам*.

Как следует из определения, необходимыми условиями открытости являются:

* модульность;
* соответствие стандартам (необязательно официальным, но обязательно общепринятым и легко доступным по цене, компенсирующей только затраты на его разработку, поддержку и распространение);
* наличие в свободной продаже аналогичных систем других производителей (подсистем, модулей) по конкурентоспособным ценам.

Требование модульности вытекает из требования возможности замены части системы (т. е. модуля) аналогичными изделиями других производителей. Для этого система должна состоять из модулей.

Соответствие стандартам необходимо для обеспечения совместимости.

Наличие в свободной продаже и конкурентоспособность цен являются требованиями, вытекающими из практического аспекта: без выполнения этого условия открытая система может существовать только "на бумаге".

Для промышленных сетей открытость означает наличие в свободной продаже сетевой аппаратуры от разных производителей по конкурентоспособным ценам, совместимой с открытыми стандартами.

Понятие открытости не подразумевает **открытость программного кода**, как, например, в ОС Linux, хотя открытость кода позволяет добавлять в систему модули других производителей, что является признаком открытости. Однако открытость исходного кода существенно снижает надежность системы вследствие потенциальной возможности появления в ней дополнительных ошибок, внесенных во время модификации и компиляции. Поэтому открытость программного кода является спорным признаком открытости системы.

В отличие от открытых, **закрытые системы** разрабатываются по внутренним стандартам отдельных предприятий. Части (модули) закрытых систем не могут быть заменены аналогичными изделиями других производителей, а заказчик, однажды применив закрытую систему, навсегда оказывается привязанным к ее разработчику.

Разновидностью и предельным случаем открытых систем являются системы, удовлетворяющие идеологии "Plug&Play" ("вставил - и заиграло"), когда вообще не требуется усилий для конфигурирования или настройки модулей после их подключения или замены на модули других производителей