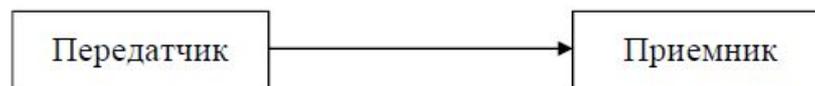


# **Лекция №4**

## **Режимы передачи данных**

**Симплекс (англ. Simplex)** - режим передачи данных, при котором передача ведется только в одном направлении по общему каналу связи.

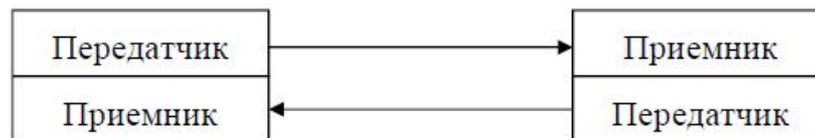
**Полудуплекс (англ. Half Duplex)** - режим передачи данных, при котором передача между устройствами ведется по общему каналу связи в любом направлении, но с разделением по времени.



Однонаправленный режим передачи – симплекс



Попеременный двунаправленный режим передачи – полудуплекс



Двунаправленный режим передачи – полный дуплекс

**Полный дуплекс (англ. Full Duplex, Duplex)** - режим передачи данных, при котором передача данных может вестись одновременно в двух направлениях по разным подканалам связи.

# Система отношений «клиент-сервер»

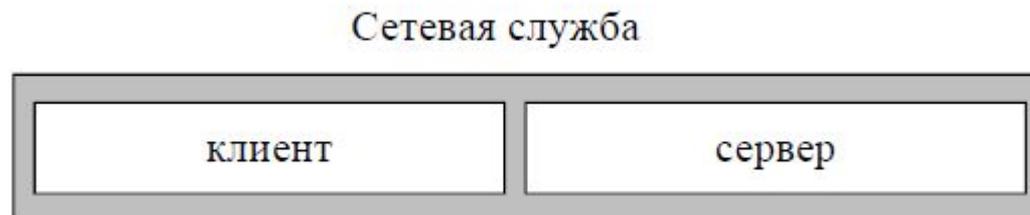
Взаимодействие компьютеров в составе сети – это обмен сообщениями двух типов: **запрос и ответ**.

**Клиент** (англ. **Client**) - это программный модуль, предназначенный для формирования и передачи сообщений-запросов к ресурсам удаленного компьютера от разных приложений с последующим приемом результатов из сети и передачей их соответствующим приложениям.



**Сервер (англ. Server)** - это программный модуль, который постоянно ожидает прихода из сети запросов от клиентов и, приняв запрос, пытается его обслужить, как правило, с участием локальной ОС; один сервер может обслуживать запросы сразу нескольких клиентов (поочередно или одновременно).

**Сетевая служба (англ. Service)** - пара программных модулей клиент–сервер, предоставляющая доступ к конкретному типу ресурса через сеть.

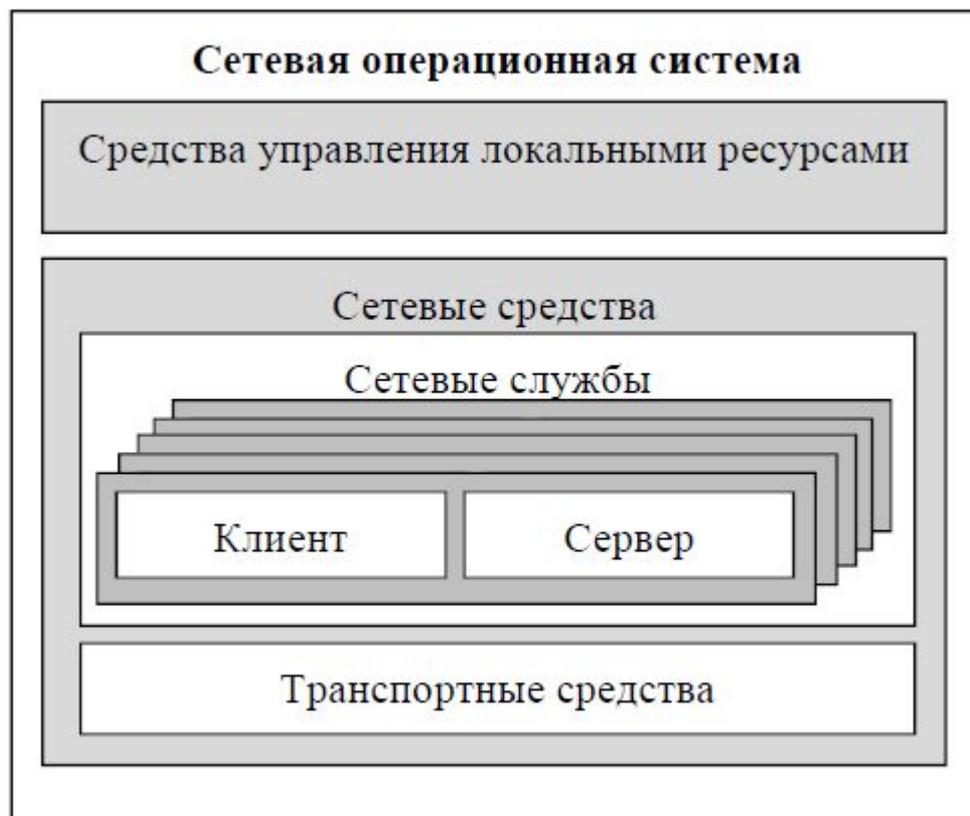


# Сетевая операционная система

**Сетевая операционная система** - ОС компьютера, которая помимо управления локальными ресурсами предоставляет пользователям и приложениям возможность эффективного и удобного доступа к информационным и аппаратным ресурсам других компьютеров сети. То есть, в составе сетевой ОС должны присутствовать программные модели различных сетевых служб.

Удаленный доступ к сетевым ресурсам обеспечивается:

- сетевыми службами;
- средствами транспортировки сообщений по сети (сетевыми адаптерами, их драйверами).



**Функциональные компоненты сетевой операционной системы**

# Одноранговые сети и сети с выделенным сервером

**Клиентская ОС** - такая ОС, в составе которой имеются преимущественно клиентские модули сетевых служб. Такая ОС позволяет компьютеру выступать только в роли клиента сети, то есть использовать удаленные ресурсы других компьютеров и сетевых устройств.

**Серверная ОС** - ОС, ориентированная на обработку запросов из сети к ресурсам своего компьютера и включает в себя расширенный набор серверных модулей сетевых служб (в том числе и служб управления сетью).

**Выделенный сервер** - компьютер, на котором установлена серверная операционная система и использующийся исключительно для обслуживания запросов других компьютеров и управления работой сети.

**Одноранговая сеть** - компьютерная сеть, в составе которой отсутствуют выделенные серверы, а все компьютеры работают под управлением одноранговых ОС.

**Сеть с выделенным сервером** - сеть, в составе которой присутствуют компьютеры не только под управлением одноранговых ОС, но обязательно присутствует хотя бы один компьютер под управлением серверной ОС.

## **Функции выделенного сервера:**

- DHCP-сервер (раздача сетевых адресов в аренду при подключении пользователей к сети);
- сервер учетных записей (контроль идентификации пользователей и ведение групповых политик);
- прокси-сервер (контроль подключения пользователей организации к ресурсам глобальных сетей);
- файловый сервер (централизованное хранилище файлов пользователей с разграничением прав доступа на основе групповых политик);
- сервер баз данных (централизованное хранение БД корпоративной информации);
- сервер печати (при использовании дорогостоящих высокопроизводительных или широкоформатных устройств печати);
- HTTP-сервер организации (может содержать материалы для внешнего или внутреннего потребления);
- сервер электронной почты организации (внутренняя либо внешняя служба) и т.д.

# Сетевые приложения

## Типы приложений:

- **Локальное приложение** - приложение, которое полностью выполняется на данном компьютере и использует только локальные ресурсы. Для такого приложения не требуется никаких сетевых средств, оно может быть выполнено на автономно работающем компьютере.
- **Централизованное сетевое приложение** - приложение, которое выполняется на данном компьютере, но в процессе его выполнения обращается к ресурсам других компьютеров сети. Например, использует файлы или устройства печати других компьютеров.

## **Типы приложений:**

**- Распределенное (сетевое) приложение** - приложение, которое состоит из нескольких взаимодействующих частей, каждая из которых выполняет какую-то определенную законченную работу

по решению прикладной задачи, причем каждая часть выполняется

на отдельном компьютере сети. Части распределенного приложения

взаимодействуют друг с другом, используя сетевые службы.

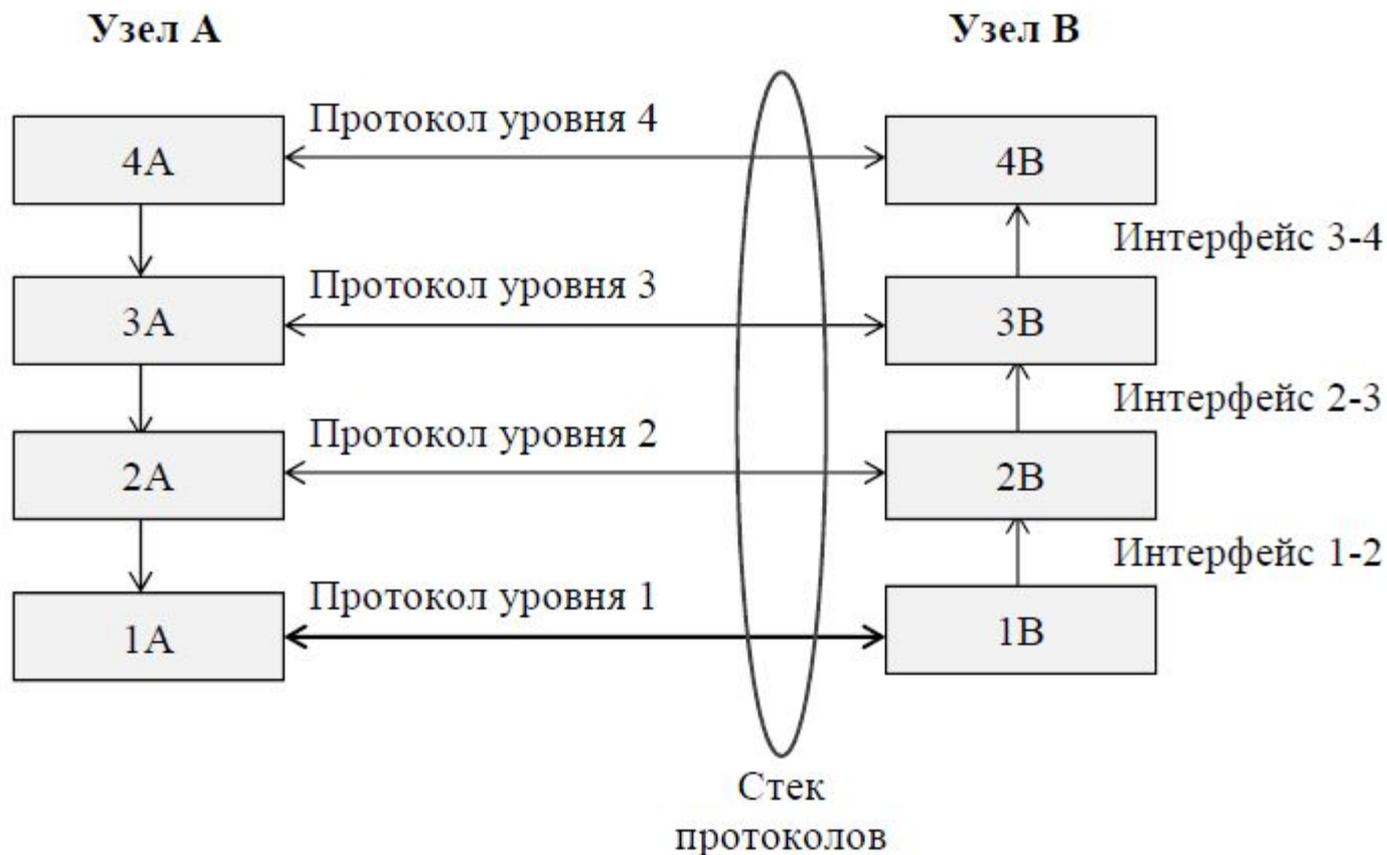
Распределенное приложение имеет доступ ко всем ресурсам сети.

**Терминальное приложение** - приложение, которое хранится и выполняется удаленно (на терминальном сервере).

Терминальные приложения используются как в клиент-серверной так и в терминальной архитектуре сетей.

# Взаимодействие открытых систем

## Протоколы и интерфейсы



**Протокол** - совокупность формализованных правил, определяющих последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах сети.

**Интерфейс** - совокупность формализованных правил и стандартизованных форматов сообщений, определяющих способ взаимодействия модулей соседних уровней в составе одного узла. Интерфейс определяет набор сервисов, предоставляемый данным уровнем соседнему уровню.

**Стек коммуникационных протоколов** - иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети.

# Модель OSI

## Модель взаимодействия открытых систем ISO/OSI

Модель OSI				
	Уровень (layer)	Тип данных (PDU)	Функции	Примеры
Host layers	7. Прикладной (application)		Доступ к сетевым службам	HTTP, FTP, SMTP
	6. Представительский (представления) (presentation)		Представление и шифрование данных	ASCII, EBCDIC, JPEG
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP
	4. Транспортный (transport)	Сегменты (segment)/ Дейтаграммы (datagram)	Прямая связь между конечными пунктами и надежность	TCP, UDP, SCTP
Media layers	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk
	2. Канальный (data link)	Биты (bit)/ Кадры (frame)	Физическая адресация	PPP, IEEE 802.2, Ethernet, DSL, L2TP, ARP
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, витая пара, коаксиальный кабель, оптический кабель

# Уровни модели OSI

**1 уровень.** Физический (*physical*). Единицей нагрузки (*PDU*) здесь является бит. Кроме единиц и нулей физический уровень не знает ничего. На этом уровне работают провода, патч панели, сетевые концентраторы (хабы, которые сейчас уже сложно найти в привычных нам сетях), сетевые адаптеры. Именно сетевые адаптеры и ничего более из компьютера. Сам сетевой адаптер принимает последовательность бит и передает её дальше.

**2 уровень.** Канальный (*data link*). PDU - кадр (*frame*). На этом уровне появляется адресация. Адресом является MAC адрес. Канальный уровень ответственен за доставку кадров адресату и их целостность. В привычных нам сетях на канальном уровне работает протокол ARP. Адресация второго уровня работает только в пределах одного сетевого сегмента и ничего не знает о маршрутизации - этим занимается вышестоящий уровень. Соответственно, устройства, работающие на L2 - коммутаторы, мосты и драйвер сетевого адаптера.

**3 уровень.** Сетевой (*network*). PDU пакет (*packet*). Наиболее распространенным протоколом (далее не буду говорить про “наиболее распространенный” - статья для новичков и с экзотикой они, как правило, не сталкиваются) тут является IP. Адресация происходит по IP-адресам, которые состоят из 32 битов. Протокол маршрутизируемый, то есть пакет способен попасть в любую часть сети через какое-то количество маршрутизаторов. На L3 работают маршрутизаторы.

# Уровни модели OSI

**4 уровень.** Транспортный (*transport*). PDU сегмент (*segment*)/датаграмма (*datagram*). На этом уровне появляются понятия портов. Тут трудятся TCP и UDP. Протоколы этого уровня отвечают за прямую связь между приложениями и за надежность доставки информации. Например, TCP умеет запрашивать повтор передачи данных в случае, если данные приняты неверно или не все. Так же TCP может менять скорость передачи данных, если сторона приема не успевает принять всё (TCP Window Size).

**5 уровень.** Сеансовый (*session*). PDU данные (*data*). Управляет сеансом связи, обменом информации, правами. Протоколы - L2TP, PPTP.

**6 уровень.** Представительский (*presentation*). PDU данные (*data*). Преставление и шифрование данных. JPEG, ASCII, MPEG.

**7 уровень.** Прикладной (*application*). PDU данные (*data*). Самый многочисленный и разнообразный уровень. На нем выполняются все высокоуровневые протоколы. Такие как POP, SMTP, RDP, HTTP и т.д. Протоколы здесь не должны задумываться о маршрутизации или гарантии доставки информации - этим занимаются нижестоящие уровни. На 7 уровне необходима лишь реализации конкретных действий, например получение html-кода или email-сообщения конкретному адресату.

# Соответствие популярных стеков протоколов модели OSI

## Стек протоколов TCP/IP

Протоколы TCP/IP делятся на 4 уровня: \*уровень IV - уровень сетевых интерфейсов - соответствует физическому(1ур)и канальному(2ур) уровням модели OSI. Этот уровень в протоколах TCP/IP не регламентируется, но поддерживает все популярные стандарты физического и канального уровня;\*уровень III - уровень межсетевого взаимодействия, который полностью соответствует сетевому(3ур) уровню модели OSI; \*уровень II – транспортный, полностью соответствует транспортному(4ур) уровню модели OSI; \*уровень I- прикладной. Соответствует трем верхним уровням модели(сеансовый 5 ур; представления 6 ур; прикладной 7ур) OSI.

Протокол HTTP (Hypertext Transfer Protocol)- протокол уровня приложений. Был разработан для передачи Web-страниц. Является основой системы World Wide Web.

Протокол SNMP (Simple Network Management Protocol) - наиболее используемое и известное программное средство для управления сетью. Для получения информации о сети протокол, переходя от одного сетевого устройства к другому, опрашивает каждого из них о его состоянии.

Протокол FTP (File Transfer Protocol) перемещение файлов между компьютерными системами. Реализует удаленный доступ к файлу. Для обеспечения надежной передачи в качестве транспорта исп протокол TCP, т.е прежде, чем получить доступ к файлу, user должен указать login и пароль. FTP предлагает широкий набор услуг, но и является самым сложным для программирования.

**Telnet** — стандартный протокол эмуляции. Протокол Telnet используется для организации удаленного администрирования узла.

**Протокол SMTP** (Simple Mail Transfer Protocol) обеспечивает передачу эл. почты. Гл. цель - надежная доставка эл.сообщений.

**Протокол TCP** (Transmission Control Protocol) — надежный протокол с установлением соединения. Он отвечает за разбиение сообщений на сегменты, их сборку в пункте назначения, повторную отсылку всего, что оказалось не полученным. Каждый TCP сегмент содержит номер порта источника и назначения, с помощью которых идентифицируются отправляющие и принимающие приложения. Примеры портов: HTTP – 80, FTP – 20, 21, Telnet – 23, SMTP – 25.

**Протокол UDP** (User Datagram Protocol ) является "ненадежным". Он используется при передаче потокового видео, игр реального времени, в IP-телефонии, а также некоторых других типов данных, чувствительных к задержкам. Каждый UDP сегмент содержит номер порта источника и назначения, с помощью которых идентифицируются отправляющие и принимающие приложения.

**Протокол IP** (Internet Protocol) обеспечивает маршрутизацию пакетов. Этот протокол ищет лучший способ направить пакет к месту его назначения. В каждом заголовке IP-пакета должен быть достоверный IP-адрес источника и адресата. Без достоверной информации об адресе отправленные пакеты не попадут к узлу назначения.

**Протокол ICMP** (Internet Control Message Protocol) работает на всех хост-машинах, использующих протокол TCP/IP. Сообщения этого протокола переносятся внутри IP-пакетов и используются для посылки управляющих сообщений и сообщений об ошибках.

**Протоколы ARP и RARP.** Протокол ARP (Address Resolution Protocol) использует широковещательные сообщения для определения физического адреса (уровень MAC), соответствующего конкретному IP-адресу.

**Протокол разрешения обратного адреса RARP** (Reverse Address Resolution Protocol) использует широковещательные сообщения для определения IP-адреса, связанного с конкретным физическим адресом.

**Ethernet** — это самый распространенный на сегодняшний день стандарт локальных сетей. Ethernet работает на 2 нижних уровнях модели OSI: канальном и физическом. Ethernet в качестве физической адресации использует MAC-адреса. В зависимости от скорости передачи данных и передающей среды существует несколько вариантов технологии: 10 Мбит/с Ethernet, Быстрый Ethernet (Fast Ethernet, 100 Мбит/с), Гигабитный Ethernet (Gigabit Ethernet, 1 Гбит/с), 10-гигабитный Ethernet, 40-гигабитный и 100-гигабитный Ethernet.

**Token Ring** (маркерное кольцо) - архитектура сетей с кольцевой логической топологией и детерминированным методом доступа, основанная на передаче маркера. Кольцевая топология означает упорядоченную передачу информации от одной станции к другой в одном направлении, строго по порядку включения.

**Технология FDDI** (Fiber Distributed Data Interface- оптоволоконный интерфейс распределенных данных) - это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель.

**Протокол PPP** (Point-to-point Protocol)используется для установления прямой связи между двумя узлами сети, причем он может обеспечить аутентификацию соединения, шифрование и сжатие данных. Используется на многих типах физических сетей: телефонная линия, сотовая связь и т. д.

**Протокол Frame Relay** (Протокол ретрансляции фреймов) - не обладает мощными средствами для поиска ошибок и имеет невысокую надежность; Frame relay обеспечивает множество независимых виртуальных каналов (Virtual Circuits, VC) в одной линии связи. Извещает о перегрузках в сети.

**АТМ (Asynchronous Transfer Mode - Режим асинхронной передачи)**— Международный стандарт для передачи ячеек, в которых различные типы данных передаются в ячейках фиксированной длины (53 байта). Это позволяет обрабатывать их на стационарном оборудовании, сокращая транзитные задержки. АТМ позволяет воспользоваться высокоскоростными передающими средами, такими как ЕЗ, SONET и ТЗ.