Вопросы по дисциплине

«Основы интернет-технологий»

1. Основные понятия компьютерных сетей (пакет, шлюз, сервер и др.).

**Компьютерные сети** — это совокупность компьютеров и других устройств, соединенных между собой для обмена данными и ресурсами. Ниже приведены основные понятия, связанные с компьютерными сетями:

**Пакет** — это блок данных, передаваемый через сеть. В сетях, использующих коммутацию пакетов, данные разделяются на небольшие части, называемые пакетами. Каждый пакет содержит адрес источника и назначения, а также данные и управляющую информацию (например, номера последовательностей и контрольные суммы). Пакеты могут путешествовать разными маршрутами и в итоге собираться в исходное сообщение на принимающем устройстве.

**Шлюз** — это сетевое устройство, которое служит "входом" в другую сеть, часто используемую для подключения локальной сети (LAN) к глобальной сети (интернет). Шлюзы выполняют преобразование сетевых протоколов и маршрутизацию трафика между различными сетями. Они могут также обеспечивать функции безопасности, такие как брандмауэр.

**Сервер** — это компьютер или устройство в сети, предоставляющее ресурсы, данные, услуги или программы другим компьютерам, называемым клиентами. Существует множество типов серверов, включая:

Веб-сервер: обслуживает веб-страницы.

Файловый сервер: предоставляет доступ к файлам.

Почтовый сервер: управляет электронной почтой.

База данных сервер: управляет доступом к базам данных.

**Протокол** — это набор правил и соглашений для передачи данных между устройствами. Примеры сетевых протоколов включают:

HTTP/HTTPS: протоколы для передачи гипертекстов (веб-страниц).

FTP: протокол передачи файлов.

TCP/IP: базовый набор протоколов для интернета.

**IP-адрес (Internet Protocol address)** — уникальный адрес устройства в сети, использующийся для его идентификации и связи. Существует два основных типа IP-адресов:

IPv4: состоит из четырех десятичных чисел, разделенных точками (например, 192.168.0.1).

IPv6: состоит из восьми групп шестнадцатеричных чисел, разделенных двоеточиями (например, 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

**MAC-адрес (Media Access Control address)** — уникальный идентификатор, присвоенный сетевому интерфейсу устройства для использования в сетях, основанных на технологии Ethernet. Он представляет собой 48-битное число, обычно записываемое в шестнадцатеричном формате (например, 00:1A:2B:3C:4D:5E).

**Коммутатор** — это сетевое устройство, которое соединяет устройства в локальной сети (LAN) и направляет данные между ними на основе MAC-адресов. Коммутаторы работают на уровне канального слоя (уровень 2 модели OSI).

**Маршрутизатор** — это устройство, которое направляет пакеты данных между различными сетями. В отличие от коммутаторов, маршрутизаторы работают на сетевом уровне (уровень 3 модели OSI) и используют IP-адреса для принятия решений о маршрутизации.

**Брандмауэр (Firewall)** — это система безопасности, которая контролирует и фильтрует сетевой трафик на основе заданных правил безопасности. Брандмауэры могут быть аппаратными или программными, и их основная задача — защищать сеть от несанкционированного доступа и атак.

**DNS (Domain Name System)** — это система, которая преобразует понятные человеку доменные имена (например, www.example.com) в IP-адреса, которые компьютеры используют для взаимодействия друг с другом. DNS-серверы хранят и обрабатывают эти преобразования.

**LAN (Local Area Network)** — локальная сеть, соединяющая устройства в пределах ограниченной географической области, такой как дом, офис или небольшая группа зданий. LAN обеспечивает высокую скорость передачи данных и низкие задержки.

**WAN (Wide Area Network)**— глобальная сеть, охватывающая большие географические области, такие как страны или континенты. Интернет является крупнейшей и самой известной WAN.

1. Топологии сетей.

**1. Шина (bus)**

В шинной топологии все устройства подключены к одной общей линии передачи данных, называемой шиной или магистралью. Данные передаются по шине, и все устройства в сети получают эти данные, но только целевое устройство обрабатывает их.

Преимущества:

* Простота установки и расширения.
* Экономичность (меньше кабеля требуется).
* При неисправности любого компьютера сеть работает.

Недостатки:

* Если шина выходит из строя, вся сеть перестает работать.
* Ограниченная пропускная способность и снижение производительности при увеличении числа узлов.
* Трудности с выявлением и устранением неисправностей.
* Низкий уровень безопасности.

**2. Звезда (star)**

В звездной топологии все устройства подключены к центральному узлу, обычно серверу или концентратору. Данные передаются от источника к центральному узлу, а затем к целевому устройству.

Преимущества:

* Легкость управления и выявления неисправностей.
* Если один кабель выходит из строя, это не влияет на работу всей сети.
* Высокая производительность при большом количестве устройств.

Недостатки:

* Центральный узел является критической точкой отказа.
* Большое количество кабелей, необходимых для подключения каждого устройства к центральному узлу.

**3. Кольцо (ring)**

В кольцевой топологии каждое устройство подключено к двум другим, формируя кольцо. Данные передаются по кольцу от одного устройства к другому до достижения целевого устройства. Доступ к линии связи осуществляется путем передачи от узла к узлу логических знаков - «маркеров», давая им возможность переслать пакет данных.

Преимущества:

* Все устройства имеют равные права на доступ к сети, что предотвращает коллизии.
* Простота установки и настройки.

Недостатки:

* Если одно устройство или соединение выходит из строя, это может нарушить работу всей сети.
* Трудности с расширением сети.

Не вошедшие в классику: дерево, ячеистая, полносвязная, снежинка.

Комбинированные (гибридные): звездно-шинная, звездно-кольцевая.

Структура локальной вычислительной сети включает в себя: пользователей ЛВС, коммутаторы и файловый сервер.

1. Сетевые устройства, их отличия и функции.

**1. Коммутатор (Switch)**

Функции:

* Соединяет устройства в локальной сети (LAN).
* Работает на канальном уровне (уровень 2 модели OSI).
* Пересылает данные на основе MAC-адресов.
* Уменьшает коллизии и повышает производительность сети за счет создания отдельных доменов коллизий для каждого порта.

Отличия:

* В отличие от концентраторов, коммутаторы направляют данные только нужному устройству, а не всем узлам в сети.
* Интеллектуальные коммутаторы (управляемые коммутаторы) позволяют администрировать и конфигурировать сеть, обеспечивая высокую гибкость и контроль.

**2. Маршрутизатор (Router)**

Функции:

* Соединяет разные сети, такие как локальная сеть (LAN) и глобальная сеть (WAN).
* Работает на сетевом уровне (уровень 3 модели OSI).
* Маршрутизирует пакеты данных на основе IP-адресов.
* Обеспечивает возможность межсетевой маршрутизации и выбора оптимального пути для передачи данных.

Отличия:

* В отличие от коммутаторов, которые работают внутри одной сети, маршрутизаторы соединяют разные сети.
* Может выполнять функции межсетевого экрана и NAT (Network Address Translation).

**3. Концентратор (Hub)**

Функции:

* Соединяет несколько Ethernet-устройств, работая как многопортовый повторитель.
* Работает на физическом уровне (уровень 1 модели OSI).
* Передает данные, полученные на одном порту, на все остальные порты.

Отличия:

* Не имеет логики для фильтрации или управления трафиком.
* В отличие от коммутаторов, создает один большой домен коллизий, что снижает производительность сети при увеличении числа устройств.

**4. Точка доступа (Access Point)**

Функции:

* Соединяет беспроводные устройства с проводной сетью.
* Работает на канальном уровне (уровень 2 модели OSI).
* Обеспечивает доступ к сети по стандартам Wi-Fi.

Отличия:

* В отличие от маршрутизаторов, точка доступа сама по себе не выполняет маршрутизацию между сетями.
* Может быть частью маршрутизатора или отдельным устройством.

**5. Модем (Modulator-Demodulator)**

Функции:

* Преобразует цифровые сигналы компьютера в аналоговые сигналы для передачи по телефонным линиям и наоборот.
* Позволяет подключаться к интернету через телефонные линии, кабельные системы или другие аналоговые среды.

Отличия:

* В отличие от маршрутизаторов и коммутаторов, модемы работают на физическом уровне и обеспечивают доступ к интернету через провайдера.

**6. Шлюз (Gateway)**

Функции:

* Соединяет сети, использующие разные протоколы или архитектуры.
* Работает на различных уровнях модели OSI в зависимости от функций.
* Преобразует данные между разными сетевыми протоколами.

Отличия:

* Может выполнять функции маршрутизатора, брандмауэра и других устройств.
* Обеспечивает связь между разными сетевыми средами, такими как IP и Bluetooth.

**7. Брандмауэр (Firewall)**

Функции:

* Контролирует и фильтрует входящий и исходящий сетевой трафик на основе заданных правил безопасности.
* Может работать на различных уровнях модели OSI.
* Защищает сеть от несанкционированного доступа и кибератак.

Отличия:

* Может быть программным (на уровне операционной системы) или аппаратным (отдельное устройство).
* В отличие от других сетевых устройств, фокусируется на безопасности и управлении доступом.

**8. Сетевые адаптеры (Network Interface Cards, NICs)**

Функции:

* Обеспечивают физическое подключение устройства к сети.
* Преобразуют данные в формат, подходящий для передачи по сети.

Отличия:

* Могут быть встроенными в материнскую плату или отдельными картами.
* Поддерживают различные типы соединений, такие как Ethernet, Wi-Fi и другие.

1. Варианты реализации компьютерных сетей (LAN, WAN, MAN).

Компьютерные сети могут быть реализованы в различных масштабах и с разными целями. Основные типы сетей включают локальные (LAN), глобальные (WAN) и городские (MAN) сети. Рассмотрим их более подробно:

**1. Локальная сеть (LAN)**

Локальная сеть (Local Area Network, LAN) — это сеть, охватывающая небольшую географическую область, такую как офис, здание или кампус.

Характеристики:

* Географический охват: ограничен небольшими расстояниями, обычно не превышает одного здания или комплекса.
* Скорость: высокая скорость передачи данных (до 10 Гбит/с и выше).
* Компоненты: коммутаторы, маршрутизаторы, точки доступа, сетевые адаптеры.
* Применение: совместное использование ресурсов (файлы, принтеры), внутренняя коммуникация, локальные приложения.

Преимущества:

* Высокая скорость и надежность.
* Простота управления и администрирования.
* Низкие затраты на установку и обслуживание.

Недостатки:

* Ограниченный географический охват.
* Может требовать более сложной настройки и управления в больших сетях.

**2. Городская сеть (MAN)**

Городская сеть (Metropolitan Area Network, MAN) — это сеть, охватывающая город или большой кампус, объединяющая несколько локальных сетей (LAN).

Характеристики:

* Географический охват: охватывает город или крупный населенный пункт.
* Скорость: обычно ниже, чем у LAN, но выше, чем у WAN (до 1 Гбит/с).
* Компоненты: маршрутизаторы, оптоволоконные линии, беспроводные мосты.
* Применение: соединение офисов одной компании в пределах города, предоставление доступа к интернету и другим ресурсам.

Преимущества:

* Объединение нескольких локальных сетей для создания единой инфраструктуры.
* Возможность обеспечения высокоскоростного доступа в пределах города.
* Экономия за счет совместного использования ресурсов и услуг.

Недостатки:

* Более высокая стоимость установки и обслуживания по сравнению с LAN.
* Сложности с управлением и безопасностью на большом расстоянии.

**3. Глобальная сеть (WAN)**

Глобальная сеть (Wide Area Network, WAN) — это сеть, охватывающая большие географические области, такие как страны или континенты, объединяющая несколько локальных и городских сетей.

Характеристики:

* Географический охват: охватывает большие расстояния, вплоть до всего мира.
* Скорость: варьируется от низкой до средней (до 100 Мбит/с и выше, в зависимости от технологии и провайдера).
* Компоненты: маршрутизаторы, модемы, спутниковые и оптоволоконные связи, телекоммуникационные каналы.
* Применение: соединение филиалов компании в разных городах и странах, предоставление глобального доступа к интернету.

Преимущества:

* Объединение сетей на больших расстояниях.
* Возможность глобального взаимодействия и обмена данными.
* Широкий доступ к интернет-ресурсам и услугам.

Недостатки:

* Высокая стоимость установки и обслуживания.
* Более низкая скорость и надежность по сравнению с LAN и MAN.
* Сложности с управлением, безопасностью и задержками в передаче данных.

1. Коммутация каналов и пакетов

**Коммутация каналов и пакетов** – это два основных метода передачи данных в сетях. Оба метода имеют свои особенности, преимущества и недостатки, что делает их подходящими для различных типов сетей и приложений.

**Коммутация каналов**

Коммутация каналов (Circuit Switching) – это метод передачи данных, при котором устанавливается фиксированный путь или канал между двумя точками, и этот канал остается зарезервированным на все время сеанса связи.

*Основные характеристики:*

* **Установление соединения:** перед началом передачи данных требуется установить соединение между отправителем и получателем. Это соединение остается фиксированным на протяжении всего сеанса.
* **Резервирование ресурсов:** ресурсы (полоса пропускания, каналы) резервируются на весь период сеанса связи.
* **Постоянный поток:** данные передаются в непрерывном потоке, что подходит для приложений, требующих постоянной скорости передачи данных (например, голосовая связь).

*Примеры:*

Телефонные сети: Традиционные телефонные системы используют коммутацию каналов, обеспечивая непрерывное соединение между абонентами.

*Преимущества:*

* **Гарантированное качество связи:** поскольку канал зарезервирован, качество связи и пропускная способность постоянны.
* **Предсказуемая задержка:** минимальная и предсказуемая задержка передачи данных.

*Недостатки:*

* **Низкая эффективность использования ресурсов:** Ресурсы зарезервированы, даже если передача данных временно не происходит.
* **Длительное установление соединения:** Время на установление и разрыв соединения увеличивает задержки.

**Коммутация пакетов**

Коммутация пакетов (Packet Switching) – это метод передачи данных, при котором сообщения разбиваются на небольшие пакеты, и каждый пакет передается независимо через сеть. Пакеты могут следовать разными маршрутами и могут быть собраны в правильном порядке на принимающей стороне.

*Основные характеристики:*

* **Нет необходимости в установлении соединения:** пакеты передаются независимо друг от друга, что позволяет сети быть более гибкой.
* **Динамическое использование ресурсов:** ресурсы используются только в момент передачи пакета, что повышает эффективность.
* **Маршрутизация:** пакеты могут следовать разными маршрутами в зависимости от загруженности сети и других факторов.

*Примеры:*

Интернет: Протоколы IP и TCP/IP используют коммутацию пакетов для передачи данных между узлами сети.

*Преимущества:*

* **Эффективное использование ресурсов**: Ресурсы используются только тогда, когда это необходимо, что позволяет обслуживать большее количество пользователей.
* **Гибкость и масштабируемость**: Сеть может легко адаптироваться к изменениям трафика и отказам узлов.
* **Лучше подходит для передачи данных**: Коммутация пакетов более эффективна для передачи данных, которые могут быть переданы с непостоянной скоростью (например, веб-сайты, файлы, электронная почта).

*Недостатки:*

* **Возможные задержки:** Задержки и вариации времени доставки (джиттер) могут быть выше, особенно в загруженных сетях.
* **Перегрузка и потеря пакетов:** Возможны ситуации перегрузки сети, что приводит к потере пакетов и необходимости их повторной передачи.



1. Основы Маршрутизации

**Маршрутизация** – это процесс определения пути, по которому данные (пакеты) должны пройти из одной сети в другую. Она играет ключевую роль в функционировании компьютерных сетей, особенно в крупных и сложных сетях, таких как интернет. Основы маршрутизации включают понимание маршрутизаторов, протоколов маршрутизации, таблиц маршрутизации и различных методов маршрутизации.

**1. Маршрутизаторы**

Маршрутизаторы – это устройства, которые соединяют разные сети и управляют потоком данных между ними. Они принимают решения о направлении пакетов на основе информации, содержащейся в таблицах маршрутизации.

*Основные функции маршрутизаторов:*

* Перенаправление пакетов между различными сетями.
* Поддержка и обновление таблиц маршрутизации.
* Применение правил и политик маршрутизации.
* Обеспечение безопасности путем фильтрации трафика.

**2. Таблицы маршрутизации**

Таблицы маршрутизации содержат информацию о возможных путях в сети. Каждая запись в таблице маршрутизации включает:

* **Адрес сети назначения:** сетевой адрес, к которому направляются пакеты.
* **Маска сети**: определяет размер сети.
* **Шлюз или следующий узел:** указывает, через какой узел или интерфейс данные должны быть отправлены.
* **Метрика:** стоимость пути, может учитывать расстояние, задержку и другие параметры.
* **Интерфейс**: сетевой интерфейс, через который должен быть отправлен пакет.

**3. Методы маршрутизации**

*Статическая маршрутизация:*

* Администратор вручную конфигурирует маршруты.
* Проста в настройке для небольших сетей.
* Не адаптируется автоматически к изменениям в сети.

*Динамическая маршрутизация:*

* Маршрутизаторы автоматически обмениваются маршрутной информацией и адаптируются к изменениям.
* Требует настройки протоколов маршрутизации.
* Более сложна в конфигурации, но более гибкая и масштабируемая.

*Политики маршрутизации:*

Администраторы могут применять политики, чтобы контролировать путь прохождения трафика (например, для обеспечения безопасности или качества обслуживания).

**4. Основные понятия**

Основные понятия:

* **Хоп (Hop):** Один переход от одного маршрутизатора к другому.
* **Метрика:** Числовая величина, используемая для оценки стоимости пути.
* **Конвергенция:** Процесс достижения согласованного состояния маршрутных таблиц всеми маршрутизаторами в сети.

1. Плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy, PDH)

**Плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy, PDH)** – это технология передачи данных, используемая для мультиплексирования цифровых сигналов в телекоммуникационных сетях. PDH позволяет передавать большие объемы данных по различным каналам и объединять их в один высокоскоростной поток.

**Основные концепции и характеристики PDH**

**1. Плезиохронность:**

* Термин "плезиохронный" означает "почти синхронный". В PDH сигналы передаются с номинально одинаковой, но не точно синхронизированной частотой.
* Каждый мультиплексируемый сигнал может иметь небольшие отклонения в скорости передачи данных, что требует дополнительных механизмов для компенсации разницы в скоростях.

**2. Мультиплексирование:**

* В PDH используется иерархическое мультиплексирование. Низкоскоростные цифровые сигналы (например, 2 Мбит/с) объединяются для создания более высокоскоростных потоков (например, 8 Мбит/с, 34 Мбит/с и 140 Мбит/с).
* Мультиплексирование осуществляется с использованием временного разделения (TDM), при котором каждый низкоскоростной сигнал занимает определенное время в высокоскоростном потоке.

**3. Иерархические уровни:**

PDH определяет несколько уровней иерархии, каждый из которых соответствует определенной скорости передачи данных.

*Основные уровни иерархии включают:*

* E1: 2.048 Мбит/с (основной уровень в Европе)
* E2: 8.448 Мбит/с
* E3: 34.368 Мбит/с
* E4: 139.264 Мбит/с

**Преимущества и недостатки PDH**

*Преимущества:*

* Простота реализации и использования в ранних цифровых сетях.
* Широкое применение в период становления цифровых телекоммуникаций.

*Недостатки:*

* Ограниченная масштабируемость и сложность декомплексирования сигналов.
* Недостаток механизмов синхронизации и управления.
* Отсутствие стандартных методов для включения новых сервисов и гибкости управления.

1. Синхронная цифровая иерархия (Synchronous Digital Hierarchy, SDH)

**Синхронная цифровая иерархия (Synchronous Digital Hierarchy, SDH)** – это стандарт для передачи данных по оптическим и электрическим телекоммуникационным сетям, который преодолел многие ограничения плезиохронной цифровой иерархии (PDH). SDH обеспечивает высокую пропускную способность, точную синхронизацию и улучшенные возможности управления и мониторинга сети. Этот стандарт широко используется в телекоммуникационных сетях по всему миру.

**Основные концепции и характеристики SDH**

**1. Синхронизация:**

* В SDH все элементы сети синхронизированы с высокой точностью с использованием единого опорного тактового сигнала.
* Это обеспечивает стабильность и надежность передачи данных, устраняя проблемы, связанные с асинхронностью, как в PDH.

**2. Мультиплексирование:**

* SDH использует иерархическое мультиплексирование с временным разделением (TDM), где низкоскоростные потоки данных объединяются в высокоскоростные потоки.
* Основное различие от PDH заключается в синхронности и более простой структуре мультиплексирования.

**3. Иерархические уровни:**

SDH определяет несколько уровней скорости передачи данных, называемых синхронными транспортными модулями (STM).

*Основные уровни включают:*

* STM-1: 155.52 Мбит/с
* STM-4: 622.08 Мбит/с
* STM-16: 2.488 Гбит/с
* STM-64: 9.953 Гбит/с

Эти уровни позволяют легко масштабировать сеть, добавляя новые каналы и увеличивая пропускную способность.

**4. Канальные контейнеры и виртуальные контейнеры:**

* Данные на различных уровнях SDH структурированы в канальные контейнеры (C-n) и виртуальные контейнеры (VC-n).
* **Канальные контейнеры** определяют основные блоки данных, которые передаются через сеть.

**5. Оверхед (Overhead):**

* SDH включает несколько уровней оверхеда для управления, мониторинга и обеспечения целостности данных.
* Основные категории оверхеда включают секционный (Section Overhead, SOH), линейный (Line Overhead, LOH) и трактовый (Path Overhead, POH) оверхед.
* Оверхед используется для синхронизации, управления ошибками, сигнализации и управления сетью.

**Преимущества SDH**

1. Высокая пропускная способность (делает её подходящей для современных телекоммуникационных приложений.)

2. Точная синхронизация (обеспечивает надежность и стабильность передачи данных.)

3. Гибкость и масштабируемость (позволяет легко добавлять новые каналы и увеличивать пропускную способность сети без значительных изменений в инфраструктуре.)

4. Улучшенные возможности управления (механизмы мониторинга и диагностики в SDH позволяют оперативно выявлять и устранять проблемы в сети.)

1. Асинхронный режим передачи (ATM)

Асинхронный режим передачи (Asynchronous Transfer Mode, ATM) — это технология коммутации пакетов, разработанная для обеспечения высокой производительности и гибкости в сетях передачи данных. ATM объединяет преимущества коммутации каналов и пакетов, предлагая фиксированный размер ячеек и возможности для передачи различных типов данных, включая голос, видео и данные.

**Основные концепции и характеристики ATM**

**1. Структура ячеек:**

* В ATM данные передаются в виде ячеек фиксированной длины. Каждая ячейка состоит из 53 байт: 5 байт заголовка и 48 байт данных.
* Фиксированный размер ячеек позволяет легко обрабатывать и коммутировать данные на высокой скорости.

**2. Коммутация ячеек:**

* Ячейки коммутируются на основе информации, содержащейся в их заголовках, которая включает идентификаторы виртуальных каналов и путей (VCI/VPI).
* Коммутационные устройства (ATM-коммутаторы) обрабатывают ячейки независимо от их содержимого, что позволяет эффективно передавать данные с различными характеристиками.

**Преимущества ATM**

**1. Высокая производительность:**

Фиксированный размер ячеек и аппаратная коммутация обеспечивают высокую скорость передачи данных и низкие задержки.

**2. Поддержка различных типов трафика:**

ATM может передавать данные, голос и видео, обеспечивая требуемое качество обслуживания для каждого типа трафика.

**3. Гибкость и масштабируемость:**

ATM поддерживает различные скорости передачи данных и может быть использован в сетях различного масштаба, от локальных до глобальных.

**4. Управление качеством обслуживания:**

Возможность задания параметров качества обслуживания позволяет гарантировать требуемые характеристики передачи данных для критически важных приложений.

**Недостатки ATM**

**1. Сложность:**

ATM-сети сложны в настройке и управлении по сравнению с другими технологиями передачи данных.

**2. Стоимость:**

Внедрение и поддержка ATM-сетей могут быть дорогими из-за необходимости специализированного оборудования и программного обеспечения.

**3. Конкуренция с другими технологиями:**

Появление более простых и дешевых технологий, таких как Ethernet и IP, привело к снижению популярности ATM в новых сетях.

1. Беспроводная технология Wi-Fi

**Wi-Fi (Wireless Fidelity)** — это технология беспроводной локальной сети (WLAN), позволяющая устройствам подключаться к интернету или друг к другу без использования проводов. Wi-Fi основан на стандартах, разработанных Институтом инженеров электротехники и электроники (IEEE), в частности, на серии стандартов IEEE 802.11. Эта технология широко используется в домах, офисах, публичных местах и на предприятиях для предоставления удобного и быстрого доступа к сети.

**Основные концепции и характеристики Wi-Fi**

**1. Частотные диапазоны и каналы:**

* Wi-Fi работает в двух основных частотных диапазонах: 2.4 ГГц и 5 ГГц.
* Частотный диапазон 2.4 ГГц имеет большее покрытие, но более подвержен помехам от других устройств.
* Частотный диапазон 5 ГГц обеспечивает более высокую скорость передачи данных, но имеет меньшее покрытие и лучше подходит для плотных сетей.

**2. Безопасность:**

Wi-Fi использует несколько методов шифрования для обеспечения безопасности данных:

* WEP (Wired Equivalent Privacy): устаревший метод шифрования, считающийся небезопасным.
* WPA (Wi-Fi Protected Access): улучшенная безопасность по сравнению с WEP.
* WPA2: более надежный метод шифрования, использующий протокол AES.
* WPA3: последняя версия с улучшенными характеристиками безопасности, такими как более сильное шифрование и защита от атак по словарю.

**3. Точки доступа и роутеры:**

* Точки доступа (Access Points, AP): Устройства, обеспечивающие беспроводное подключение к сети для клиентов.
* Роутеры: Устройства, которые соединяют локальные сети (LAN) с внешними сетями, такими как интернет. Многие роутеры имеют встроенные точки доступа Wi-Fi.

**Преимущества Wi-Fi**

1. Удобство и мобильность (Устройства могут подключаться к сети без проводов, что обеспечивает мобильность и удобство использования.)

2. Легкость развертывания (Простота установки и конфигурации сетей Wi-Fi делает их доступными для использования в различных средах, от домашних до корпоративных.)

3. Расширяемость (Wi-Fi позволяет легко добавлять новые устройства в сеть без необходимости прокладывать дополнительные кабели.)

4. Совместимость (Современные стандарты Wi-Fi обеспечивают совместимость устройств различных производителей, что упрощает интеграцию оборудования.)

**Недостатки Wi-Fi**

1. Ограниченное покрытие (Диапазон покрытия Wi-Fi может быть ограничен физическими препятствиями и интерференцией от других устройств.)

2. Безопасность (Хотя современные методы шифрования обеспечивают высокий уровень безопасности, сети Wi-Fi могут быть уязвимы к различным атакам, если не применяются соответствующие меры защиты.)

3. Скорость и стабильность (Скорость и стабильность подключения могут варьироваться в зависимости от расстояния до точки доступа, количества подключенных устройств и внешних помех.)

1. 1-й уровень OSI — физический (L1, physical layer)

* Здесь происходит обмен оптическими, электрическими или радиосигналами между устройствами отправителя и получателя.
* На этом уровне железо не распознает данные в классическом для нас виде (картинки, текст, видео), но оно понимает биты (единицы и нули) и работает только с сигналами. Таким оборудованием выступают концентраторы, медиаконвертеры или репитеры. Здесь информация или биты передаются либо по проводам, кабелям, либо без них, например через Bluetooth, Wi-Fi.
* Когда возникает проблема с сетью, многие специалисты сразу же обращаются к физическому уровню, чтобы проверить, например, не отключен ли сетевой кабель от устройства.

1. 2-й уровень OSI — канальный (L2, data link layer)

**Задача** – проверить целостность полученных данных и исправить ошибки.

Этот уровень «поумнее» предыдущего: он уже понимает, что разные амплитуды напряжений отвечают разным битам - нулям и единицам. А ещё канальный уровень умеет кодировать сигналы в биты и передавать их дальше.

**МАС-адрес** (от англ. Media Access Control - надзор за доступом к среде, также Hardware Address, также физический адрес) — уникальный идентификатор, присваиваемый каждой единице сетевого оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet.

С помощью команды **ifconfig** можно узнать МАС-адрес вашего Macbook или компьютера на Linux. B Windows нужно ввести команду ipconfig.

* Уровень управления логическим каналом — LLC (logical link control);
* Уровень управления доступом к среде - тот самый МАС (media access control).

**Logical Link Control** (общепринятое сокращение - LLC) - верхний подуровень управления логической связью - по стандарту IEEE 802 - верхний подуровень канального уровня модели OSI, осуществляет:

* управление передачей данных,
* обеспечивает проверку и правильность передачи информации по соединению.

1. 3-й уровень OSI — сетевой (L3, network layer)

Этот уровень отвечает за маршрутизацию данных внутри сети между компьютерами. Здесь уже появляются такие термины, как «маршрутизаторы» и «IP-адреса».

Данные на сетевом уровне представляются в виде пакетов. Такие пакеты похожи на фреймы канального уровня, но используют другие адреса получателя и отправителя - IP-адреса.

1. 4-й уровень OSI — транспортный (L4, transport layer)

Два главных протокола - ТСР и UDP.

**TCP (Transmission Control Protocol)** – это протокол, который гарантирует доставку данных в корректном виде. Он жёстко следит за каждым битом информации, но работает гораздо медленнее UDP.

Для этого и нужен ТСР - чтобы данные доходили в правильном виде.

**Сегментация** важна для надёжности. Один большой пакет может потеряться или направиться не тому адресату. А маленькие пакеты снижают риск подобных ошибок и даже позволяют проверять их количество. Если какой-то сегмент не получилось доставить, протокол ТСР может запросить его у отправителя снова. Так обеспечивается надёжность.

1. 5-й уровень OSI — сеансовый (L5, session layer)

Уровни с пятого по седьмой уже работают с чистыми данными.

И здесь за дело берутся не сетевые инженеры, а разработчики.

Сеансовый уровень, исходя из названия, отвечает за поддержание сеанса или сессии. Он координирует коммуникацию между приложениями и отвечает за установление, поддержание и завершение связи, синхронизацию задач и сам обмен информацией. Примером для пятого уровня можно назвать созвон в Zoom или прямой эфир на YouTube. Во время сессии необходимо обеспечивать синхронизированную передачу аудио и видео для всех участников, а также поддерживать саму связь. За это как раз отвечают протоколы сеансового уровня (RPC, H.245, RTCP).

1. 6-й уровень OSI — уровень представления данных (L6, presentation layer)

Шестой уровень подготавливает информацию для последнего и преобразует (сжимает, кодирует, шифрует) их в понятный язык для пользователя или машины. Например, если вы отправляете картинку, то она сначала приходит в виде битов, а потом трансформируются в JPEG, GIF или другой формат.

1. 7-й уровень OSI — прикладной (L7, application layer)

Верхний уровень модели OSI – это прикладной. С помощью своих протоколов он отображает данные в понятном конечному пользователю формате. Сюда входят такие технологии, как НТТР, DNS, FTP, SSH и многое другое. Почти каждый человек ежедневно взаимодействует с протоколами прикладного уровня.

1. Технология Ethernet

**Ethernet** – это технология, которая соединяет проводные локальны сети (LAN) и позволяет устройству взаимодействовать друг с другом по протоколу, являющемуся общим сетевым языком

Основные концепции и характеристики Ethernet

**1. Физические среды:**

Ethernet может работать на различных типах кабелей:

* **Витая пара (Twisted Pair):** широко используемая среда передачи для Ethernet-сетей. Включает категории кабелей Cat5, Cat5e, Cat6 и Cat6a.
* **Оптическое волокно (Fiber Optics):** используется для высокоскоростных и длинных соединений. Включает одномодовые и многомодовые волокна.
* **Коаксиальный кабель (Coaxial Cable):** ранее использовался в ранних версиях Ethernet, таких как 10BASE2 и 10BASE5.

**2. Метод доступа к среде:**

Ethernet использует метод доступа к среде CSMA/CD (Carrier Sense Multiple Access with Collision Detection):

* **Carrier Sense:** устройство проверяет, свободен ли канал, прежде чем начать передачу.
* **Multiple Access:** несколько устройств могут использовать один и тот же канал.
* **Collision Detection:** если два устройства начинают передачу одновременно, возникает коллизия, и устройства повторяют попытку передачи через случайное время.

**3. Коммутация и маршрутизация:**

* **Коммутаторы (Switches):** устройства, которые соединяют узлы в сети и направляют данные только на целевые устройства, уменьшая количество коллизий и улучшая производительность сети.
* **Маршрутизаторы (Routers):** устройства, которые соединяют разные сети и направляют пакеты данных между ними, используя информацию о маршрутах.

**Преимущества Ethernet**

1. Высокая производительность:

Ethernet обеспечивает высокую скорость передачи данных, что позволяет использовать его для различных приложений, от простого обмена файлами до потоковой передачи видео высокой четкости.

2. Надежность:

Ethernet-сети устойчивы к помехам и обеспечивают надежную передачу данных благодаря использованию проверенных технологий и методов коррекции ошибок.

3. Масштабируемость:

Ethernet легко масштабируется, поддерживая различные скорости передачи данных и различные типы кабелей, что позволяет расширять и модернизировать сеть по мере необходимости.

4. Широкое распространение и совместимость:

Ethernet является широко распространенным стандартом, что обеспечивает совместимость оборудования и программного обеспечения различных производителей.

**Недостатки Ethernet**

1. Ограничения расстояния:

Максимальная длина кабеля для Ethernet ограничена, особенно для медных кабелей, что требует использования коммутаторов и повторителей для расширения сети на большие расстояния.

2. Коллизии в сети:

В сети с большим количеством устройств могут возникать коллизии, хотя использование коммутаторов значительно уменьшает эту проблему.

3. Стоимость развертывания:

Развертывание Ethernet-сети может потребовать значительных затрат на кабели, оборудование и установку, особенно в крупных сетях.

4. Низкая безопасность:

Данные в разделяемой среде доступны всем.

1. MTU, (maximum transmission unit)

**MTU (Maximum Transmission Unit)** — это максимальный размер пакета данных, который может быть передан по сети без фрагментации. Размер MTU включает в себя размер заголовка и полезную нагрузку данных. В сетевых технологиях, правильная настройка MTU имеет критическое значение для эффективности передачи данных и производительности сети.

**Основные концепции и характеристики MTU**

**1. Размер MTU:**

* Размер MTU варьируется в зависимости от типа сети и протоколов, которые используются.
* В Ethernet, стандартный размер MTU составляет 1500 байт. Это означает, что максимальный размер пакета, включая заголовок, не должен превышать 1500 байт.

**2. Фрагментация:**

* Если размер пакета превышает установленный MTU, он должен быть разделен на более мелкие фрагменты, которые затем пересылаются по сети и собираются обратно на приемной стороне.
* Процесс фрагментации и повторной сборки пакетов может привести к увеличению задержки и снижению производительности сети.

**3. MTU в различных сетях:**

* Ethernet: обычно 1500 байт.
* IPv4: теоретически может быть до 65535 байт, но на практике ограничено MTU подлежащей сети.
* IPv6: минимальный MTU для IPv6 составляет 1280 байт.
* Туннели и VPN: обычно имеют уменьшенный MTU из-за дополнительных заголовков (например, 1400-1450 байт).

**4. Path MTU Discovery (PMTUD):**

* Метод определения максимального MTU на пути между источником и получателем.
* PMTUD помогает предотвратить фрагментацию пакетов, определяя наименьший MTU по всему маршруту и уменьшая размер отправляемых пакетов соответственно.

**Преимущества:**

1. Повышенная производительность:

Правильная настройка MTU может уменьшить количество фрагментаций, тем самым увеличивая эффективность передачи данных.

2. Снижение накладных расходов:

Оптимальный размер MTU уменьшает накладные расходы на фрагментацию и сборку пакетов.

3. Улучшение стабильности сети:

Снижение вероятности потери пакетов и ошибок, связанных с фрагментацией.

**Недостатки:**

1. Сложности настройки:

Определение оптимального MTU может потребовать тщательного тестирования и настройки, особенно в сложных сетях.

2. Проблемы с несовместимостью:

Разные сегменты сети могут поддерживать разные значения MTU, что может привести к проблемам с фрагментацией.

3. Влияние на пропускную способность:

Слишком малый MTU может привести к увеличению накладных расходов из-за большого количества заголовков относительно полезной нагрузки.

1. Методы рассылки сообщений

Метод рассылки **"один к одному"** называется **одноадресным** (unicast). Это означает, что у сообщения есть только один адресат.

Если узел рассылает сообщения методом **"один ко многим"**, это **многоадресная рассылка** (multicast).

Многоадресная рассылка предусматривает одновременную отправку одного и того же сообщения группе узлов.

Если всем сетевым узлам необходимо получить сообщение в одно и то же время, используется **широковещательная рассылка** (broadcast). Это метод рассылки сообщений **"один ко всем".** Кроме того, для узлов предусмотрены правила рассылки сообщений с подтверждением и без него.

1. Протоколы TCP/IP

**TCP/IP (Transmission Control Protocol/Internet Protocol)** — это набор протоколов, который используется для связи и обмена данными в Интернете и других сетях. Он включает множество протоколов, каждый из которых выполняет определенную функцию в процессе передачи данных. TCP/IP можно рассматривать как набор взаимосвязанных протоколов, работающих на разных уровнях модели OSI.

**Основные уровни модели TCP/IP**

**1. Уровень канала передачи данных (Link Layer):**

* Отвечает за физическое соединение и передачу данных между устройствами в одной сети.
* Примеры протоколов: Ethernet, Wi-Fi.

**2. Сетевой уровень (Internet Layer):**

* Обеспечивает маршрутизацию данных между сетями.
* Примеры протоколов: IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol).

**3. Транспортный уровень (Transport Layer):**

* Обеспечивает надежную передачу данных между конечными узлами сети.
* Примеры протоколов: TCP (Transmission Control Protocol), UDP (User Datagram Protocol).

**4. Прикладной уровень (Application Layer):**

* Предоставляет приложениям доступ к сетевым услугам.
* Примеры протоколов: HTTP, FTP, SMTP, DNS.

**Основные протоколы TCP/IP**

**1. IP (Internet Protocol):**

* Основной протокол сетевого уровня, ответственный за адресацию и маршрутизацию пакетов данных.
* IPv4: широко используемая версия IP с 32-битными адресами.
* IPv6: новая версия IP с 128-битными адресами, разработанная для решения проблемы исчерпания адресов IPv4.

**2. TCP (Transmission Control Protocol):**

* Протокол транспортного уровня, обеспечивающий надежную и упорядоченную передачу данных.
* Обеспечивает установку соединения, контроль потока, обнаружение и коррекцию ошибок.
* Подходит для приложений, требующих гарантированной доставки данных, таких как веб-серфинг, электронная почта.

**3. UDP (User Datagram Protocol):**

* Протокол транспортного уровня, обеспечивающий передачу данных без установления соединения и без гарантии доставки.
* Быстрее и эффективнее, но менее надежен, чем TCP.
* Используется для приложений, требующих быстрой передачи данных, таких как потоковое видео и игры в реальном времени.

**4. ICMP (Internet Control Message Protocol):**

* Протокол сетевого уровня, используемый для диагностики и управления сетевыми соединениями.
* Используется утилитами ping и traceroute для проверки доступности и маршрутизации узлов в сети.

**5. ARP (Address Resoчlution Protocol):**

* Протокол, используемый для сопоставления IP-адресов с физическими MAC-адресами в локальных сетях.
* Позволяет устройствам находить физические адреса других устройств по их IP-адресам.

**6. DNS (Domain Name System):**

* Протокол прикладного уровня, используемый для преобразования доменных имен в IP-адреса.
* Обеспечивает удобство использования Интернета, позволяя обращаться к ресурсам по именам, а не по числовым адресам.

**7. HTTP (HyperText Transfer Protocol):**

* Протокол прикладного уровня, используемый для передачи веб-страниц и других данных через Интернет.
* HTTPS: Защищенная версия HTTP, использующая шифрование для защиты данных.

**8. FTP (File Transfer Protocol):**

* Протокол прикладного уровня, используемый для передачи файлов между клиентами и серверами.
* Позволяет загружать и скачивать файлы, а также управлять файловой системой удаленного сервера.

**9. SMTP (Simple Mail Transfer Protocol):**

* Протокол прикладного уровня, используемый для отправки электронной почты между серверами.
* Работает вместе с протоколами POP3 и IMAP для доставки почты конечным пользователям.

1. Сетевые интерфейсы (порт, протокол, сокет, статика)

**Порт** — это логический номер, который используется для идентификации конкретного процесса или службы на устройстве в сети. Порты работают на транспортном уровне модели OSI (уровень 4) и помогают различать различные приложения и службы, работающие на одном устройстве.

**Основные концепции:**

* Диапазон портов: Номера портов варьируются от 0 до 65535.
* Известные порты (0-1023): зарезервированы для системных или широко используемых служб (например, HTTP использует порт 80, HTTPS — порт 443).
* Регистируемые порты (1024-49151): обычно используются для пользовательских или менее известных служб.
* Динамические/частные порты (49152-65535): назначаются динамически для временных коммуникаций (например, для исходящих соединений).

**Использование портов:**

* Порты позволяют нескольким сетевым службам работать одновременно на одном IP-адресе, обеспечивая возможность параллельной работы различных приложений.

**Протокол** — это набор правил и соглашений, которые определяют, как данные передаются и обрабатываются в сети. Протоколы могут работать на разных уровнях модели OSI и обеспечивают различные функции, от физической передачи данных до обеспечения безопасности и управления соединениями.

**Основные типы протоколов:**

*Протоколы канального уровня (Layer 2):*

* Ethernet: обеспечивает передачу данных в локальных сетях.
* PPP (Point-to-Point Protocol): используется для прямых соединений между двумя узлами.

*Протоколы сетевого уровня (Layer 3):*

* IP (Internet Protocol): основной протокол для маршрутизации пакетов данных между узлами в сети.
* ICMP (Internet Control Message Protocol): используется для передачи диагностических сообщений и сообщений об ошибках.

*Протоколы транспортного уровня (Layer 4):*

* TCP (Transmission Control Protocol): обеспечивает надежную передачу данных с установлением соединения.
* UDP (User Datagram Protocol): обеспечивает быструю, но ненадежную передачу данных без установления соединения.

*Протоколы прикладного уровня (Layer 7):*

* HTTP (HyperText Transfer Protocol): используется для передачи веб-страниц.
* FTP (File Transfer Protocol): используется для передачи файлов.
* SMTP (Simple Mail Transfer Protocol): используется для отправки электронной почты.

**Сокет** — это программный интерфейс, который позволяет приложениям обмениваться данными через сеть. Сокет сочетает в себе IP-адрес и номер порта, что позволяет идентифицировать конкретное сетевое соединение.

**Основные концепции:**

**1. Типы сокетов:**

* **Сокеты потока (Stream Sockets):** используют протокол TCP для надежной передачи данных.
* **Датаграммные сокеты (Datagram Sockets):** используют протокол UDP для быстрой и ненадежной передачи данных.

**2. Адресация сокетов:**

* Сокет идентифицируется комбинацией IP-адреса и порта (например, 192.168.1.1:80).

**3. Использование сокетов:**

* Приложения создают сокеты для установления соединений и обмена данными. Серверы слушают на определенных портах, принимая входящие соединения, в то время как клиенты инициируют соединения.

**Статика** в контексте сетевых технологий обычно относится к статической адресации и статической маршрутизации.

**Статическая адресация:**

* При статической адресации IP-адреса назначаются устройствам вручную и не изменяются автоматически.
* Используется в ситуациях, когда необходимо, чтобы устройство всегда имело один и тот же IP-адрес, например, для серверов, принтеров или других сетевых устройств.

**Преимущества:**

* Простота управления и предсказуемость.
* Не требуется DHCP-сервер для назначения адресов.
* Удобство для настройки серверов и других постоянных ресурсов.

**Недостатки:**

* Трудоемкость при масштабировании сети.
* Возможность конфликтов адресов при неправильном управлении.

**Статическая маршрутизация:**

* В статической маршрутизации маршруты к определенным сетям прописываются вручную в конфигурации маршрутизаторов.
* Используется в небольших или простых сетях, где топология редко изменяется.

**Преимущества:**

* Простота настройки и отсутствие необходимости в сложных протоколах маршрутизации.
* Прозрачность маршрутов и легкость их диагностики.

**Недостатки:**

* Непрактичность в крупных и динамических сетях.
* Невозможность автоматического реагирования на изменения в топологии сети.

1. Виды и требования к адресации в компьютерных сетях.

**Виды адресации**

* **MAC-адреса (Media Access Control):** уникальные аппаратные адреса, присваиваемые каждому сетевому интерфейсу.
* **IP-адреса (Internet Protocol):** логические адреса, используемые для идентификации устройств в сети.
* **URL (Uniform Resource Locator):** адреса, используемые для доступа к ресурсам в Интернете. Состоит из схемы (протокола), доменного имени, пути и иногда порта (например, https://www.example.com:443/path/to/resource).

**Требования к адресации**

**1. Уникальность:**

Каждый адрес в сети должен быть уникальным, чтобы избежать конфликтов и обеспечить корректную маршрутизацию данных. В локальных сетях уникальность MAC-адресов обеспечивает производитель оборудования. В глобальных сетях уникальность IP-адресов обеспечивается через системы распределения адресов, такие как IANA и региональные интернет-регистраторы (RIR).

**2. Иерархичность:**

Адресация должна поддерживать иерархическую структуру для эффективной маршрутизации. В IP-адресации это достигается с помощью сетевых префиксов и масок подсети, что позволяет группировать адреса и упрощать маршрутизацию.

**3. Глобальность и локальность:**

Адреса могут быть глобальными (уникальными в глобальном масштабе, как IP-адреса в Интернете) и локальными (уникальными только внутри определенной сети, как приватные IP-адреса).

**4. Скалируемость:**

Система адресации должна быть масштабируемой, чтобы поддерживать рост сети. Это особенно важно для глобальных сетей, таких как Интернет, где количество устройств постоянно увеличивается. Переход от IPv4 к IPv6 был необходимым шагом для обеспечения долгосрочной масштабируемости.

**5. Простота управления:**

Управление адресами должно быть простым и понятным. В небольших сетях это можно делать вручную, но в крупных сетях часто используются автоматизированные системы управления адресами, такие как DHCP для динамического назначения IP-адресов.

**6. Поддержка мобильности:**

Современные системы адресации должны поддерживать мобильность устройств, обеспечивая возможность смены сети без потери связи. Это особенно важно для мобильных устройств, таких как смартфоны и ноутбуки.

1. IP адреса, виды, версии,

IP-адрес (Internet Protocol address) — это уникальный числовой идентификатор, присваиваемый каждому устройству, подключенному к сети, использующей протокол IP для связи. IP-адреса позволяют устройствам находить и общаться друг с другом в сети.

**Виды IP-адресов**

**1. Публичные (глобальные) IP-адреса:**

* Используются для идентификации устройств в глобальной сети Интернет.
* Примеры: 8.8.8.8 (Google DNS).

**2. Приватные (локальные) IP-адреса:**

* Используются для идентификации устройств внутри частных сетей и не маршрутизируются в Интернете.
* Диапазоны (IPv4):

10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

**3. Статические IP-адреса:**

* Назначаются вручную и остаются постоянными.
* Использование: Сервера, сетевые устройства, требующие постоянного адреса.

**4. Динамические IP-адреса:**

* Назначаются автоматически через DHCP и могут изменяться.
* Использование: Домашние компьютеры, мобильные устройства.

**Версии IP-адресов**

**1. IPv4 (Internet Protocol version 4):**

* Формат: 32-битный адрес, обычно записывается в виде четырех десятичных чисел, разделенных точками (например, 192.168.1.1).
* Диапазон: 0.0.0.0 - 255.255.255.255.
* Объем: 4,294,967,296 (4.3 миллиарда) уникальных адресов.

**2. IPv6 (Internet Protocol version 6):**

* Формат: 128-битный адрес, записывается в виде восьми групп шестнадцатеричных чисел, разделенных двоеточиями (например, 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
* Диапазон: 3.4 x 10^38 уникальных адресов.
* Особенности: решает проблему исчерпания адресов IPv4, включает улучшенные функции безопасности и автоконфигурации.

1. Протокол DHCP

**DHCP** – это протокол, используемый в компьютерных сетях для автоматического назначения IP-адресов и других сетевых параметров устройствам, подключенным к сети.

**Функциональность:** DHCP позволяет устройствам автоматически получать IP-адрес, маску подсети, шлюз по умолчанию, адрес DNS-сервера и другие параметры конфигурации сети.

**Процесс работы:** когда устройство присоединяется к сети, оно отправляет DHCP-запрос на локальную сеть. DHCP-сервер отвечает на запрос, предоставляя устройству доступные сетевые настройки.

**Преимущества:**

* Упрощение процесса настройки сетевых параметров для клиентских устройств.
* Эффективное использование IP-адресов в сети за счет динамического переиспользования адресов.

**Недостатки:**

* **Одна точка отказа:** если DHCP-сервер не работает, новые устройства не могут получить сетевые настройки.
* **Задержка при запуске:** большое количество устройств может вызвать задержку при подключении из-за ожидания ответа от DHCP-сервера.
* **Риск конфликтов IP-адресов:** неправильная конфигурация DHCP-сервера может привести к конфликту IP-адресов.
* **Уязвимость к атакам:** неправильная настройка DHCP или нарушение безопасности сети могут стать объектом атак.
* **Сложное управление:** управление несколькими DHCP-серверами и их настройками может быть сложным в крупных сетях.

**Использование:** DHCP широко применяется в домашних и корпоративных сетях, а также в общественных Wi-Fi сетях, чтобы обеспечить автоматическое назначение сетевых настроек для подключенных устройств.

1. Структура и особенности МАС-адреса.

**Структура:**

* МАС-адрес представляет собой уникальный идентификатор сетевого интерфейса, присваиваемый производителем оборудования.
* Формат МАС-адреса: шесть байт (или 48 бит) в шестнадцатеричном виде, разделенных двоеточиями или тире (например, 00:1A:2B:3C:4D:5E).

**Особенности:**

* Уникальность: Каждый МАС-адрес должен быть уникальным. Производители сетевого оборудования получают уникальные идентификаторы благодаря регистрации в IEEE.
* Структура: МАС-адрес состоит из двух частей: идентификатора производителя (OUI - Organizationally Unique Identifier) и идентификатора устройства, присвоенного производителем.
* Распознавание производителя: Первые три байта МАС-адреса (или первые 24 бита) представляют собой OUI, который идентифицирует производителя оборудования.
* Стандарты: Стандарты Ethernet определяют формат МАС-адресов. Например, IEEE 802.3 определяет структуру МАС-адреса для Ethernet.

1. Классы компьютерных сетей. Диапазоны IP-адресов.

Компьютерные сети классифицируются на основе их размера и охвата. Существуют три основных **класса компьютерных сетей:**

* **LAN (Local Area Network):** Локальная сеть, охватывающая небольшую территорию, такую как офис, здание или кампус. LAN часто используется для связи между компьютерами, принтерами и другими устройствами внутри ограниченной области.
* **MAN (Metropolitan Area Network):** Магистральная сеть, которая охватывает город или большой географический район. MAN может включать в себя несколько LAN и обеспечивать связь между ними.
* **WAN (Wide Area Network):** Глобальная сеть, которая охватывает большие географические расстояния и может включать в себя города, страны и даже континенты. Интернет является примером WAN, который связывает компьютеры и сети по всему миру.

**Диапазоны IP-адресов:**

IP-адреса делятся на публичные и приватные и принадлежат к различным диапазонам:

**1. Публичные IP-адреса:**

Эти адреса уникальны в глобальной сети Интернет и используются для идентификации устройств во всем мире.

* Диапазон IPv4: от 0.0.0.0 до 255.255.255.255.
* Диапазон IPv6: широкий диапазон адресов, начинающийся с 2000::.

**2. Приватные IP-адреса:**

Эти адреса предназначены для использования внутри частных сетей и не маршрутизируются в Интернете. Они могут быть повторно использованы в разных сетях, поскольку являются локальными.

* Диапазон IPv4:

10.0.0.0 - 10.255.255.255 (CIDR: 10.0.0.0/8).

172.16.0.0 - 172.31.255.255 (CIDR: 172.16.0.0/12).

192.168.0.0 - 192.168.255.255 (CIDR: 192.168.0.0/16).

* Диапазон IPv6: Существует несколько диапазонов приватных IPv6-адресов, включая fd00::/8.

1. Протокол преобразования адресов ARP

**ARP (Address Resolution Protocol)** – это протокол сетевого уровня, используемый для определения MAC-адреса устройства по его IP-адресу в локальной сети.

**Функциональность:** когда устройство нуждается в отправке пакета данных на определенный IP-адрес в локальной сети, оно отправляет ARP-запрос для определения MAC-адреса устройства с этим IP-адресом.

**Процесс работы:**

* Устройство, отправляющее данные, создает ARP-запрос, содержащий IP-адрес целевого устройства.
* ARP-запрос широковещательно отправляется по всей локальной сети.
* Устройство с указанным IP-адресом отвечает на ARP-запрос, сообщая свой MAC-адрес отправителю.
* После получения ответа отправитель кэширует соответствие IP-адреса и MAC-адреса устройства в своей ARP-таблице для последующего использования.

**Преимущества:**

* Обеспечивает эффективное преобразование IP-адресов в MAC-адреса для локальной сети.
* Помогает снизить количество широковещательного трафика в сети.

**Использование**:

Протокол ARP широко используется в локальных сетях для определения MAC-адресов других устройств в сети, необходимых для успешной передачи данных.

1. Network Address Translation (NAT)

**NAT** – это процесс преобразования IP-адресов и портов пакетов данных при их прохождении через сетевое устройство, такое как маршрутизатор или брандмауэр.

**Функциональность:** NAT позволяет скрыть внутренние IP-адреса устройств в локальной сети за одним или несколькими общими публичными IP-адресами при доступе в интернет.

**Типы NAT:**

* **Static NAT (статический NAT):** преобразует один внутренний IP-адрес в один внешний IP-адрес.
* **Dynamic NAT (динамический NAT):** преобразует несколько внутренних IP-адресов в пул внешних IP-адресов по мере необходимости.
* **NAT Overload (PAT - Port Address Translation):** преобразует внутренние IP-адреса и порты в пул внешних IP-адресов и портов. Это самый распространенный тип NAT, позволяющий использовать один публичный IP-адрес для обращения к интернету из нескольких устройств.

**Преимущества:**

* Увеличение безопасности сети за счет скрытия внутренних IP-адресов.
* Экономия публичных IP-адресов, так как несколько устройств в сети могут использовать один и тот же внешний IP-адрес.

**Недостатки:**

* Ограничения в некоторых приложениях, таких как VoIP и онлайн-игры, которые могут испытывать проблемы с NAT из-за необходимости прослушивания входящего трафика.
* Сложность настройки и поддержки приложений, использующих NAT.

**Использование:** NAT широко применяется в корпоративных и домашних сетях для обеспечения доступа в Интернет и защиты внутренних сетей от внешних угроз.

1. DNS. Компоненты, Структура доменного имени. Алгоритм работы системы DNS.

**Компоненты DNS:**

**1. Информационный ресурс:**

Это база данных, которая содержит информацию о доменных именах и их соответствующих IP-адресах, а также другие сведения, такие как записи MX для электронной почты. Информационный ресурс состоит из зон, которые хранят информацию о доменах и их записях.

**2. DNS-клиент (resolver):**

Это приложение или устройство, которое инициирует запросы к DNS-серверам для разрешения доменных имен в IP-адреса. DNS-клиент может быть частью операционной системы или прикладного программного обеспечения.

**3. Сервер имен (name server), или DNS-сервер:**

Это сервер, который содержит информацию о доменных именах и отвечает на запросы DNS-клиентов для разрешения этих имен. DNS-серверы могут быть разделены на различные типы, такие как корневые DNS-серверы, авторитетные DNS-серверы для конкретных зон, рекурсивные DNS-серверы, которые выполняют запросы от клиентов и передают их по цепочке до получения ответа, и т. д.

**4.Протокол DNS:**

Это набор стандартов и правил, которые определяют форматы сообщений DNS, методы обмена сообщениями между DNS-клиентами и серверами, а также процессы разрешения доменных имен. Протокол DNS работает поверх протоколов транспортного уровня, таких как UDP (User Datagram Protocol) и TCP (Transmission Control Protocol), и использует различные типы записей DNS для хранения информации о доменах.

**Структура доменного имени:**

Структура доменного имени представляет собой иерархическую систему, состоящую из последовательности поддоменов, разделенных точками. Например, в доменном имени "www.example.com" "com" является верхним уровнем домена (TLD), "example" - вторым уровнем домена (SLD), а "www" - поддоменом.

**Алгоритм работы системы DNS:**

* Клиент отправляет запрос DNS для разрешения доменного имени (например, "www.example.com") на IP-адрес.
* Если запрашиваемая информация уже находится в кэше DNS-клиента, то он возвращает ее клиенту. В противном случае, клиент отправляет запрос DNS-серверу.
* Если запрашиваемая информация находится в кэше DNS-сервера, то он возвращает ее клиенту. В противном случае, сервер начинает поиск информации в соответствующих зонах.
* Если запрашиваемая информация найдена в зоне, DNS-сервер возвращает ее клиенту и может также кэшировать ее для будущих запросов.
* Если информация не найдена в локальной зоне, DNS-сервер делегирует запрос вышестоящим серверам, начиная с корневых DNS-серверов и двигаясь по иерархии до достижения DNS-сервера, ответственного за соответствующий домен.
* Когда найдена соответствующая информация, она возвращается клиенту и кэшируется на всех уровнях DNS-системы для ускорения последующих запросов.

1. Классификация доменных имён.

**1. Верхний уровень домена (TLD):**

* **Географические TLD (ccTLD):** состоят из двух символов, обозначающих страну или территорию (например, .ru для России, .uk для Великобритании).
* **Международные TLD (gTLD):** обозначают различные области и секторы (например, .com для коммерческих сайтов, .org для некоммерческих организаций).

**2. Вторичный уровень домена (SLD):**

Это часть доменного имени перед TLD, определяющая конкретный домен (например, в "google.com" "google" является SLD).

**3. Субдомены:**

Это дополнительные уровни доменов, которые могут быть созданы внутри основного домена для организации и иерархии (например, "mail.google.com" - "mail" является субдоменом Google).

**4. Продвинутые домены:**

* Домены верхнего уровня страны внутри доменов верхнего уровня (ccSLD): Некоторые страны разрешают создание своих собственных поддоменов (например, .co.uk для Великобритании).
* Домены верхнего уровня с топологическим ориентиром (TLD): Это специальные TLD, такие как .net, .edu, .gov, которые указывают на определенные типы организаций или секторы.

**5. IDN (Internationalized Domain Names):**

Это домены, содержащие символы из различных алфавитов, таких как кириллица, китайские и японские иероглифы, арабские буквы и т. д. (например, преобразованный в Юникод домен "россия.рф").

1. Формирование подсетей

**Формирование подсетей в компьютерных сетях** – это процесс разделения большой сети на более мелкие подсети для повышения эффективности управления сетью и оптимизации использования адресного пространства IP. Вот шаги, которые обычно используются при формировании подсетей:

* **Определение требований сети:** определите количество устройств, которые будут подключены к сети, а также их местоположение и взаимосвязь. Это позволит определить, сколько подсетей нужно создать и каким образом они будут связаны.
* **Выбор подходящего блока IP-адресов:** определите диапазон IP-адресов, который будет использоваться для создания подсетей. Размер блока адресов зависит от количества устройств в каждой подсети и предполагаемого роста сети.
* **Разработка плана подсетей:** определите размеры и адреса каждой подсети в соответствии с требованиями сети. при разработке плана подсетей учитывайте такие факторы, как количество устройств, предполагаемый рост сети, необходимость изоляции устройств и оптимизация трафика.
* **Применение маски подсети:** для каждой подсети укажите маску подсети, которая определяет количество бит, используемых для идентификации сети и устройств в этой подсети. Маска подсети позволяет разделить IP-адрес на две части: сетевую и хостовую.
* **Назначение адресов:** назначьте адреса каждой подсети, включая адрес сети и шлюз по умолчанию. Убедитесь, что каждый адрес уникален в пределах сети и соответствует выбранной маске подсети.
* **Настройка маршрутизаторов и коммутаторов:** настройте маршрутизаторы и коммутаторы для маршрутизации трафика между подсетями и обеспечения связности сети.
* **Тестирование и отладка:** после настройки подсетей тщательно протестируйте их, чтобы убедиться, что все устройства могут связываться друг с другом и обмениваться данными. При необходимости вносите коррективы.

1. Протокол ICMP

**Протокол ICMP (Internet Control Message Protocol)** – это протокол сетевого уровня, используемый для передачи сообщений об ошибках и контроля состояния сети в сетях TCP/IP. Вот основные аспекты протокола ICMP:

* **Обнаружение ошибок и уведомление об ошибках:** ICMP используется для обнаружения и уведомления об ошибках, возникающих в процессе передачи данных. Например, когда маршрутизатор не может доставить пакет на назначенный адрес, он отправляет сообщение об ошибке ICMP отправителю.
* **Управление сообщениями и состоянием сети:** ICMP также используется для отправки запросов и получения ответов для проверки состояния сети и управления сетевыми устройствами. Например, команда ping использует ICMP для проверки доступности узла и измерения задержки.
* **Типы сообщений:** ICMP включает различные типы сообщений, включая эхо-запросы (ping), эхо-ответы, сообщения об ошибках, информационные запросы и ответы. Каждый тип сообщения имеет свое назначение и используется для определенных задач сетевого управления и диагностики.
* **Интеграция с протоколом IP:** ICMP работает поверх протокола IP и использует его для передачи сообщений в сети. ICMP-сообщения встраиваются в полезную нагрузку IP-пакетов и используются для обмена информацией между узлами сети.

1. Основные понятия web-сети

Основные понятия веб-сети включают:

* **Web-сайт**: набор логически связанных Web страниц, изображений (images), videos или других цифровых данных, которые доступны web серверу. Web сайт имеет некоторую начальную страницу (homepage), используя которую можно перейти к другим страницам сайта. Страницы web сайта обычно разделены по папкам, которые создают иерархическую структуру сайта. Если начальная страница сайта в URL. не указывается, то web сервер использует страницу со стандартным именем (default или index).
* **Web-страница:** Отдельный документ, отображаемый в браузере. Он состоит из HTML-кода, который определяет структуру и содержимое страницы, а также может содержать CSS-стили для оформления и JavaScript для интерактивности. Web-страницы могут быть статическими и динамическими.
* **Браузер:** Программное обеспечение, позволяющее пользователям просматривать веб-страницы и взаимодействовать с ними. Популярные браузеры включают Chrome, Firefox, Safari, Edge и Opera.
* **Сервер:** Компьютерное устройство или программное обеспечение, которое предоставляет ресурсы, такие как веб-страницы, изображения и файлы, другим устройствам в сети, обычно по запросу.
* **Хостинг:** Услуга, предоставляемая провайдерами хостинга, которая позволяет размещать веб-сайты и другие онлайн-ресурсы на серверах, доступных через Интернет.
* **Web приложение или webapp** - клиент-серверное приложение, в котором клиеш взаимодействует с веб-сервером при помощи браузера. Это также компьютерная прикладная программа, пользовательский интерфейс (UI) которой кодируется на языках понятных браузерам (например, HTML, CSS, JavaScript и пр.) и которая доверяет обычному браузеру возможность работы с ней.
* **Web клиент** – специализированная программа, которая обрабатывает запросы клиентов к веб-приложениям.

1. Клиент-серверная архитектура

**Клиент-серверная архитектура** – это распространенная модель в компьютерных сетях, в которой компьютеры (клиенты) обращаются к серверам для получения ресурсов или услуг. Вот основные аспекты клиент-серверной архитектуры:

* **Клиенты:** это устройства или программы, которые инициируют запросы к серверам для получения данных или услуг. Клиенты могут быть компьютерами, мобильными устройствами, веб-браузерами, приложениями и т. д.
* **Серверы:** это компьютеры или программы, которые предоставляют ресурсы или услуги клиентам. Серверы обрабатывают запросы от клиентов и отправляют им необходимые данные или предоставляют услуги.
* **Коммуникация:** Клиенты и серверы обмениваются данными посредством сетевых протоколов, таких как HTTP, FTP, SMTP и т. д. Клиенты отправляют запросы серверам, а серверы отвечают на эти запросы, предоставляя запрашиваемые данные или услуги.
* **Распределение обязанностей:** В клиент-серверной модели серверы отвечают за предоставление ресурсов или обработку запросов, тогда как клиенты отвечают за инициирование запросов и обработку ответов от серверов.
* **Масштабируемость и гибкость:** Клиент-серверная архитектура обеспечивает гибкость и масштабируемость систем, так как серверы могут быть легко масштабированы или заменены без изменения клиентского программного обеспечения.

**Достоинства**

* Сохранность информации.
* Устойчивость к сбоям.
* Масштабируемость (способность к расширению).
* Большая защищенность информации от несанкционированного доступа
* Меньшая нагрузка сети одним пользователем, что обеспечивает большую пропускную способность сети и возможность.
* Большая гибкость системы.

**Недостатки клиент-серверной архитектуры:**

* **Зависимость от сервера:** клиент не может работать без сервера. Если сервер(а) недоступен или имеет проблемы, все клиенты будут неработоспособны или испытывать проблемы с функциональностью.
* **Затраты на инфраструктуру:** клиент-серверная архитектура требует наличия серверного и сетевого оборудования и поддержки, что может потребовать затрат на инфраструктуру и обслуживание.
* **Зависимость от сети:** клиент-серверная архитектура требует постоянного подключения к сети. Если сеть недоступна или имеет проблемы, это может существенно ограничить возможности работы системы.
* **Ограниченность:** при использовании клиент-серверной архитектуры возникают ограничения на количество одновременно подключенных клиентов и на пропускную способность сети. Это может привести к ограничениям в расширении системы и обработке большого количества запросов.

1. URI, URL, URN

**URI** — символьная строка, позволяющая идентифицировать какой-либо физический или абстрактный ресурс, который не обязательно должен быть доступен через сеть Интернет: документ, изображение, файл, службу, ящик электронной почты и т. д.

**URL** – это подтип URI, который указывает местоположение (адрес) ресурса и способ доступа к нему. URL обычно состоит из протокола доступа (например, http, https, ftp), доменного имени и пути к ресурсу. Например, "https://www.example.com/page.html" – это URL, который указывает на веб-страницу "page.html" на сервере "www.example.com".

**Uniform Resource Locator** - уникальный идентификатор ресурса (URL - адрес)

URL представляет собой набор информации, необходимый для того, чтобы определить: узел сети, на котором расположен информационный объект; расположение информационного объекта на узле; метод получения доступа к объекту.

**URN** – это еще один подтип URI, который представляет собой уникальное и постоянное имя ресурса, независимо от его местоположения или способа доступа к нему. URN часто используется для идентификации ресурсов, которые не имеют постоянного местоположения в Интернете. Например, "urn:isbn:0451450523" - это URN, который идентифицирует книгу по ее ISBN.

1. Hypertext Transfer Protocol (HTTP)

**HTTP (Hypertext Transfer Protocol)** – это протокол прикладного уровня, используемый для передачи гипертекстовых документов в сети Интернет. Вот основные аспекты протокола HTTP:

* **Основа Веба:** HTTP является основным протоколом для обмена данными между веб-серверами и клиентами, такими как веб-браузеры. Он обеспечивает передачу различных типов данных, таких как HTML-страницы, изображения, файлы стилей, скрипты и другие ресурсы.
* **Клиент-Серверная модель:** HTTP основан на клиент-серверной архитектуре, где клиент (например, браузер) отправляет запросы к веб-серверу для получения ресурсов, а сервер отвечает на эти запросы, отправляя запрошенные данные.
* **Stateless протокол:** HTTP является без состояния, что означает, что каждый запрос-ответ считается независимым от предыдущих. Сервер не сохраняет информацию о предыдущих запросах, что облегчает масштабирование и обработку большого количества запросов.
* **Методы запроса:** HTTP определяет различные методы запроса, такие как GET, POST, PUT, DELETE, OPTIONS и другие. Каждый метод имеет свою собственную функциональность и используется для выполнения определенных операций на сервере.
* **Заголовки:** HTTP-запросы и ответы содержат заголовки, которые содержат метаданные о запросе или ответе. Заголовки могут содержать информацию о типе содержимого, кэшировании, аутентификации и других аспектах, связанных с запросом или ответом.
* **Состояние протокола HTTP/2 и HTTP/3:** HTTP постоянно развивается. Версия HTTP/2 внедряет множество оптимизаций для улучшения производительности, в то время как HTTP/3 использует новый протокол передачи QUIC для улучшения скорости и безопасности передачи данных.

1. SMTP (Simple Mail Transfer Protocol)

**SMTP (Simple Mail Transfer Protocol)** – это стандартный протокол передачи электронной почты в компьютерных сетях. Он используется для отправки электронных сообщений от одного почтового сервера к другому почтовому серверу или клиенту. Вот основные аспекты SMTP:

* **Передача сообщений:** SMTP обеспечивает механизм передачи электронной почты между почтовыми серверами или между почтовым клиентом и сервером. Он работает на основе клиент-серверной модели, где почтовый клиент (например, электронная почта) отправляет сообщения по SMTP-протоколу на почтовый сервер, который затем пересылает их по сети к их адресатам.
* **Очередь сообщений:** SMTP использует очередь сообщений для временного хранения и обработки электронных сообщений. Когда почтовый сервер не может мгновенно доставить сообщение адресату, оно помещается в очередь и повторно пытается отправиться позже.
* **Аутентификация и безопасность:** SMTP поддерживает методы аутентификации для проверки подлинности отправителя и защиты от нежелательной почты. Некоторые методы аутентификации включают SMTP AUTH, которая позволяет отправителям предоставлять учетные данные перед отправкой сообщений.
* **Порты и протоколы:** стандартный SMTP-порт - 25. Однако для защищенной передачи данных часто используются альтернативные порты, такие как 465 (SMTPS) или 587 (Submission), которые используют SSL/TLS для шифрования данных.
* **Протоколы расширения:** существуют различные расширения SMTP, такие как SMTP AUTH для аутентификации и STARTTLS для защищенной передачи данных, которые расширяют функциональность и безопасность протокола SMTP.

1. POP 3 (Post Office Protocol Version 3)

**POP3 (Post Office Protocol Version 3)** – это протокол приема почты, который используется для получения электронной почты с почтового сервера на клиентский компьютер. Вот основные аспекты POP3:

* **Получение почты:** POP3 предоставляет механизм для получения электронной почты с почтового сервера на клиентский компьютер. Клиенты (почтовые программы) могут использовать POP3 для загрузки электронных сообщений с сервера на локальный компьютер пользователя.
* **Протокол доступа к сообщениям:** POP3 позволяет пользователям получать доступ к электронным сообщениям, хранящимся на почтовом сервере, и загружать их на свой компьютер. После загрузки сообщения с сервера они обычно удаляются с сервера, хотя существуют настройки для сохранения сообщений на сервере.
* **Протокол без состояния:** POP3 работает в режиме без состояния, что означает, что сервер не сохраняет состояние пользовательских сессий. Каждое подключение к серверу рассматривается как отдельное сеанс, и после получения сообщений они удаляются с сервера.
* **Простой и эффективный:** POP3 – это простой протокол, который предоставляет основные функции для получения электронной почты. Это делает его легким в реализации и эффективным в использовании для пользователей, которым не требуются сложные функции управления почтой.
* **Безопасность:** POP3 не предоставляет встроенной поддержки шифрования данных, поэтому обычно рекомендуется использовать дополнительные меры безопасности, такие как SSL/TLS или VPN, для защиты конфиденциальности при передаче данных между клиентом и сервером.

1. IMAP (Internet Message Access Protocol)

**IMAP (Internet Message Access Protocol)** – это протокол доступа к электронной почте, который позволяет клиентским устройствам получать доступ к электронным сообщениям, хранящимся на удаленном почтовом сервере. Вот основные аспекты IMAP:

* **Доступ к сообщениям на сервере:** IMAP позволяет пользователям просматривать и управлять электронными сообщениями, хранящимися на почтовом сервере, без их загрузки на локальный компьютер. Это позволяет пользователям получать доступ к своей почте с различных устройств и приложений.
* **Синхронизация состояния:** IMAP поддерживает синхронизацию состояния между клиентским устройством и сервером, что означает, что изменения, сделанные на одном устройстве (например, прочтение, удаление или перемещение сообщений), отображаются на других устройствах.
* **Множественные папки и метки:** IMAP позволяет пользователям создавать множество папок и меток для организации электронной почты. Это делает его более гибким по сравнению с POP3, который обычно ограничен одной папкой "Входящие".
* **Поддержка больших почтовых ящиков:** IMAP позволяет пользователям работать с почтовыми ящиками большого размера, поскольку сообщения остаются на сервере. Это особенно полезно для пользователей, которые получают большой объем электронной почты.
* **Поддержка SSL/TLS:** IMAP поддерживает использование SSL/TLS для защищенной передачи данных между клиентом и сервером, что обеспечивает конфиденциальность и целостность электронной почты в пути через Интернет.

1. DOM (Document Object Model)

**DOM (Document Object Model)** – это стандартизированный способ представления структуры и содержимого веб-страницы в виде объектов, которые могут быть изменены и манипулированы с помощью программного кода (обычно JavaScript). Вот основные аспекты DOM:

* **Иерархическая структура:** DOM представляет веб-страницу как иерархическое дерево объектов, где каждый элемент HTML (такой как тег <div>, <p>, <img> и т. д.) представлен как узел дерева.
* **Объектная модель:** Каждый элемент веб-страницы, включая HTML-теги, атрибуты, текстовое содержимое и другие элементы, представлен в виде объекта в DOM. Эти объекты могут быть доступны и изменены с помощью JavaScript.
* **Манипуляция содержимым:** DOM позволяет динамически изменять содержимое и структуру веб-страницы. Это включает добавление, удаление и изменение элементов, а также обработку событий, таких как клики мыши и отправка форм.
* **Навигация по дереву:** С помощью DOM можно легко перемещаться по структуре веб-страницы и находить конкретные элементы. Это делает возможным поиск элементов, изменение их стилей, атрибутов или содержимого, и многое другое.
* **Поддержка различных языков:** DOM доступен для множества языков программирования, но наиболее распространенным является его использование с JavaScript в браузерной среде. Однако он также может быть использован с помощью других языков, таких как Python (с помощью библиотеки lxml), PHP (с помощью библиотеки DOM), и других.

1. Структура HTML-документа и теги. Атрибуты тега body. Теги заголовков и абзаца

**HTML (HyperText Markup Language)** – это стандартный язык разметки, используемый для создания веб-страниц. Вот основные элементы структуры HTML-документа и некоторые из наиболее часто используемых тегов:

Структура HTML-документа:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<!-- Содержимое страницы -->

</body>

</html>

**Тег <body>:** Этот тег используется для определения тела веб-страницы, которое содержит ее содержимое, такое как текст, изображения, ссылки и т. д. Некоторые атрибуты тега <body>:

* bgcolor: Задает цвет фона страницы.
* background: Задает фоновое изображение.
* text: Задает цвет текста страницы.
* link: Задает цвет ссылок.
* vlink: Задает цвет посещенных ссылок.
* alink: Задает цвет активных ссылок.

**Теги заголовков <h1>-<h6>:** Эти теги используются для определения заголовков различного уровня на веб-странице. <h1> обычно является самым крупным и наиболее важным заголовком, а <h6> - наименьшим.

<h1> Заголовок 1</h1>

<h2> Заголовок 2</h2>

<!-- ... -->

<h6> Заголовок 6 </h6>

**Тег <p>:** Этот тег используется для определения абзацев текста.

<p>Это текст в первом абзаце. </p>

1. Ссылки, виды типы

* **Гиперссылки (Hyperlinks):** эти ссылки используются для перехода на другие веб-страницы или ресурсы. Они создаются с помощью тега <a> (anchor) и атрибута href (hypertext reference). Пример:

<a href="https://www.example.com"> Ссылка на внешний ресурс</a>

* **Якорные ссылки (Anchor links):** эти ссылки позволяют переходить к определенной части текущей страницы. Они также создаются с помощью тега <a>, но в качестве значения атрибута href указывается ID элемента, к которому нужно прокрутить страницу. Пример:

<a href="#section1">Перейти к разделу 1</a>

<!-- ... -->

<h2 id="section1">Раздел 1</h2>

* **Электронная почта (Email links):** Эти ссылки используются для создания ссылок для отправки электронной почты по определенному адресу. Они также создаются с помощью тега <a>, а адрес электронной почты указывается в атрибуте href, предварительно обернутом в префикс mailto:. Пример:

<a href="mailto:example@example.com"> Написать по электронной почте</a>

* **Картинки (Image links):** Картинки могут также использоваться как ссылки, например, для перехода на другую страницу или для загрузки большого изображения. Они создаются с помощью тега <img>, а адрес ссылки указывается в атрибуте src. Пример:

<a href="https://www.example.com">

<img src="image.jpg" alt="Изображение">

</a>

1. Как работает CSS, Принцип каскада, Таблица стилей, Селекторы, Псевдоклассы, Способы добавления/подключения стилей

**CSS (Cascading Style Sheets)** – это язык таблиц стилей, используемый для оформления веб-страниц. Он определяет внешний вид элементов HTML, таких как цвета, шрифты, отступы, размеры и т. д. Вот основные концепции и принципы работы CSS:

* **Принцип каскада (Cascade):** CSS использует принцип каскада для определения, какие стили будут применены к элементам HTML. Если для одного элемента определены несколько стилей, они могут быть унаследованы от родительских элементов или переопределены с помощью более специфичных правил CSS.
* **Таблица стилей (Stylesheet):** тчаблица стилей – это файл, содержащий набор правил CSS, которые определяют внешний вид элементов HTML. Она может быть встроена непосредственно в HTML-документ или подключена к нему внешне с помощью тега <link>.
* **Селекторы (Selectors):** селекторы в CSS используются для выбора элементов, к которым будут применены стили. Существует множество видов селекторов, включая селекторы элементов, классов, идентификаторов, а также псевдоклассы и псевдоэлементы.
* **Псевдоклассы (Pseudo-classes):** псевдоклассы позволяют применять стили к элементам в определенных состояниях или контекстах, таких как наведение курсора, посещенные ссылки, фокусировка элемента и т. д. Пример:

a:hover {

color: red; }

**Способы добавления/подключения стилей:**

* **Встроенные стили:** CSS правила могут быть встроены непосредственно в тег <style> внутри HTML-документа.
* **Внешние таблицы стилей:** Файлы CSS могут быть созданы отдельно от HTML и подключены к документу с помощью тега <link> внутри <head> документа.
* **Inline стили:** Стили могут быть применены непосредственно к отдельным HTML-элементам с помощью атрибута style.

1. Преимущества и недостатки JS

**JavaScript (JS)** – это высокоуровневый язык программирования, который широко используется для создания интерактивных веб-страниц и веб-приложений. Вот основные преимущества и недостатки JavaScript:

**Преимущества JavaScript:**

* **Интерактивность:** JavaScript позволяет создавать интерактивные элементы на веб-страницах, такие как анимации, динамические формы, слайдеры и многое другое, что улучшает пользовательский опыт.
* **Универсальность:** JavaScript поддерживается практически всеми современными браузерами и может выполняться на различных платформах, включая мобильные устройства и серверы.
* **Легкость обучения и использования:** JavaScript имеет простой и понятный синтаксис, который легко изучить даже для новичков. Большое количество библиотек и фреймворков также делает его использование еще более простым и удобным.
* **Богатство функций:** JavaScript предлагает множество встроенных функций и возможностей, которые позволяют создавать мощные и сложные приложения, включая работу с асинхронными запросами, манипуляцию DOM, валидацию форм и многое другое.
* **Скорость выполнения:** JavaScript выполняется непосредственно в браузере пользователя, что обеспечивает быстрое и отзывчивое выполнение кода.

**Недостатки JavaScript:**

* **Зависимость от браузера:** из-за различий в реализации JavaScript в разных браузерах могут возникать проблемы совместимости, что требует дополнительного тестирования и обработки.
* **Безопасность:** JavaScript выполняется на стороне клиента, поэтому он может быть уязвим для атак, таких как внедрение кода или кросс-сайтовый скриптинг (XSS), если не приняты соответствующие меры безопасности.
* **Производительность:** в некоторых случаях JavaScript может снижать производительность веб-страницы из-за большого количества вычислений или использования сложных алгоритмов.
* **Невозможность отключения:** Пользователи могут отключить выполнение JavaScript в своих браузерах, что может привести к некорректной работе функционала, зависящего от JavaScript.
* **SEO-оптимизация:** Некоторые поисковые системы имеют ограниченную возможность индексации контента, созданного с помощью JavaScript, что может негативно сказаться на SEO-оптимизации веб-страницы.

1. Java-библиотеки и фреймворки

Наиболее популярные Java-библиотеки и фреймворки:

**Spring Framework:** это один из наиболее популярных фреймворков Java, предназначенный для создания масштабируемых и надежных приложений. Spring предоставляет широкий набор функций, включая управление зависимостями, внедрение зависимостей, аспектно-ориентированное программирование (AOP), а также модули для веб-разработки, безопасности, доступа к данным и т. д.

**Apache Struts:** Struts – это фреймворк для создания веб-приложений на основе шаблона проектирования MVC (Model-View-Controller). Он предоставляет механизмы для разделения бизнес-логики, представления и управления веб-приложением.

**JavaServer Faces (JSF):** JSF – это фреймворк для создания пользовательских интерфейсов веб-приложений на основе компонентов. Он предоставляет набор компонентов интерфейса пользователя, управление жизненным циклом компонентов и возможности для взаимодействия с бэкендом.

**Play Framework:** Play – это легковесный фреймворк для создания веб-приложений на Java и Scala. Он основан на асинхронной модели выполнения и предоставляет инструменты для создания масштабируемых и реактивных приложений.

1. Виды сценариев JavaScript

Несколько распространенных видов сценариев JavaScript:

* **Обработка событий:** JavaScript используется для обработки различных событий, которые происходят на веб-странице, таких как клики мыши, наведение курсора, отправка формы, загрузка страницы и т. д. С помощью сценариев JavaScript можно привязывать функции к этим событиям и выполнять определенные действия в ответ на них.
* **Динамическое обновление содержимого:** JavaScript позволяет динамически изменять содержимое веб-страницы без необходимости перезагрузки страницы. Это может включать в себя добавление, удаление или изменение элементов DOM, обновление данных на странице, анимации и т. д.
* **Валидация форм:** JavaScript можно использовать для проверки вводимых пользователем данных в формах на веб-странице. Это позволяет выполнять проверки на корректность заполнения полей формы до их отправки на сервер, что улучшает пользовательский опыт и снижает вероятность ошибок.
* **Обработка AJAX-запросов:** JavaScript широко используется для отправки асинхронных запросов на сервер и обновления содержимого страницы без перезагрузки. Это позволяет создавать динамические веб-приложения, которые могут взаимодействовать с сервером без перезагрузки страницы.
* **Работа с куками и локальным хранилищем:** JavaScript позволяет управлять куками (cookies) и локальным хранилищем (localStorage и sessionStorage) браузера, что позволяет сохранять данные на стороне клиента и использовать их между разными сеансами работы пользователя с веб-приложением.

1. Технология Ajax

**Технология AJAX (Asynchronous JavaScript and XML)** – это подход к созданию интерактивных веб-приложений, который позволяет отправлять и получать данные с сервера асинхронно без перезагрузки всей веб-страницы. Она основана на следующих ключевых концепциях:

* **Асинхронность (Asynchronicity):** AJAX позволяет отправлять запросы на сервер и получать ответы асинхронно, то есть без блокировки пользовательского интерфейса. Это позволяет создавать динамические и отзывчивые веб-приложения.
* **Обмен данными (Data Exchange):** AJAX использует объект XMLHttpRequest (XHR) для отправки запросов на сервер и получения ответов. Этот объект позволяет отправлять различные типы запросов (GET, POST и другие) и обрабатывать ответы сервера.
* **Обновление части страницы (Partial Page Update):** вместо полной перезагрузки веб-страницы, AJAX позволяет обновлять только часть страницы, которая нуждается в обновлении. Это повышает производительность и улучшает пользовательский опыт.
* **Использование форматов данных (Data Formats):** AJAX может работать с различными форматами данных, такими как XML, JSON (JavaScript Object Notation), HTML и другими. JSON является наиболее распространенным форматом для обмена данными в веб-приложениях, благодаря своей простоте и легкости в чтении и записи JavaScript.

Применение технологии AJAX позволяет создавать интерактивные веб-приложения, такие как динамические формы, чаты, поиск в реальном времени, загрузка данных по требованию и многое другое.