**1. Коммутация каналов (КК)**

Коммутация каналов — это метод, при котором для связи между устройствами создается выделенный канал. Этот канал занят на все время передачи данных.

* **Плюсы**: постоянная пропускная способность, гарантированная задержка.
* **Минусы**: неэффективное использование ресурсов, если канал простаивает.
* **Примеры**: телефонные сети, ISDN.

**2. Коммутация сообщений и коммутация пакетов (КП)**

* **Коммутация сообщений**: данные передаются целиком как единое сообщение. Сообщение хранится на каждом узле до передачи.
* **Коммутация пакетов**: данные делятся на пакеты, которые передаются независимо. Каждый пакет содержит адрес назначения.
  + **Плюсы**: эффективное использование ресурсов, устойчивость к отказам.
  + **Минусы**: возможны задержки и потери пакетов.

**3. Коммутация пакетов. Датаграммный и виртуальный режимы**

* **Датаграммный режим**: пакеты передаются независимо, без установления соединения. Пример: IP.
  + **Плюсы**: простота, устойчивость к отказам.
  + **Минусы**: возможны потери и нарушение порядка пакетов.
* **Виртуальный режим**: перед передачей устанавливается соединение. Пакеты передаются по одному пути. Пример: TCP.
  + **Плюсы**: гарантированная доставка, сохранение порядка.
  + **Минусы**: сложнее в реализации, требует больше ресурсов

**4. Сравнение методов КК и КП**

| **Характеристика** | **Коммутация каналов (КК)** | **Коммутация пакетов (КП)** |
| --- | --- | --- |
| Выделение ресурсов | Выделенный канал | По требованию |
| Пропускная способность | Фиксированная | Динамическая |
| Задержки | Минимальные | Могут быть большими |
| Использование ресурсов | Неэффективно | Эффективно |
| Примеры | Телефонные сети | Интернет (IP) |

**5. Модель OSI. Функции уровней**

Модель OSI состоит из 7 уровней:

1. **Физический**: передача битов по кабелю.
2. **Канальный**: управление доступом к среде и обнаружение ошибок.
3. **Сетевой**: маршрутизация пакетов.
4. **Транспортный**: гарантированная доставка данных.
5. **Сеансовый**: управление сеансами связи.
6. **Уровень представления**: преобразование форматов данных.
7. **Прикладной**: взаимодействие с пользователем (HTTP, FTP).

**1. Физический уровень**

* Отвечает за передачу битов (единиц и нулей) по физической среде (кабели, оптические волокна, радио).
* Управляет физическими соединениями (разъемы, разрывы).
* Примеры: кабели, сетевые адаптеры.

**2. Канальный уровень**

* Управляет доступом к среде передачи и обнаруживает ошибки.
* Формирует кадры (блоки данных) и добавляет адреса (MAC-адреса).
* Примеры: Ethernet, Wi-Fi.

**3. Сетевой уровень**

* Маршрутизирует пакеты между сетями.
* Использует IP-адреса для идентификации устройств.
* Примеры: IP, маршрутизаторы.

**4. Транспортный уровень**

* Гарантирует доставку данных между приложениями.
* Управляет потоком данных и восстанавливает порядок пакетов.
* Примеры: TCP, UDP.

**5. Сеансовый уровень**

* Управляет сеансами связи между приложениями.
* Устанавливает, поддерживает и завершает соединения.
* Примеры: NetBIOS, RPC.

**6. Уровень представления**

* Преобразует форматы данных для передачи.
* Выполняет сжатие, шифрование и кодирование данных.
* Примеры: SSL/TLS, JPEG.

**7. Прикладной уровень**

* Предоставляет сервисы для пользователя (веб, почта, файлы).
* Обрабатывает запросы и ответы от приложений.
* Примеры: HTTP, FTP, DNS.

**6. Технология TCP/IP. Стек протоколов**

**TCP/IP** — это набор протоколов, который обеспечивает работу Интернета и локальных сетей. Каждый уровень выполняет свои задачи:

1. **Канальный**: передача данных по физической среде.
2. **Сетевой**: маршрутизация пакетов между сетями.
3. **Транспортный**: гарантированная или быстрая доставка данных.
4. **Прикладной**: взаимодействие пользователя с сетью через приложения.

**1. Канальный уровень**

* Отвечает за передачу данных по физической среде (кабели, радиоканалы).
* Определяет, как данные будут передаваться между устройствами в одной сети.
* **Примеры протоколов**: Ethernet (используется в проводных сетях), Wi-Fi (беспроводные сети).
* **Примеры устройств**: сетевые карты, коммутаторы.

**2. Сетевой уровень**

* Отвечает за маршрутизацию пакетов между разными сетями.
* Использует IP-адреса для идентификации устройств и определения пути передачи данных.
* **Основной протокол**: IP (Internet Protocol).
* **Примеры функций**:
  + Определение оптимального пути для передачи пакетов.
  + Обработка фрагментации пакетов, если их размер слишком большой.
* **Примеры устройств**: маршрутизаторы.

**3. Транспортный уровень**

* Отвечает за доставку данных между приложениями на разных устройствах.
* Предоставляет два основных протокола:
  + **TCP (Transmission Control Protocol)**:
    - Гарантирует доставку данных без потерь и в правильном порядке.
    - Использует подтверждения (ACK) и повторную передачу при потере пакетов.
    - Подходит для приложений, где важна надежность (например, веб-сайты, электронная почта).
  + **UDP (User Datagram Protocol)**:
    - Быстрый, но без гарантий доставки.
    - Не использует подтверждения, что делает его быстрее, но менее надежным.
    - Подходит для приложений, где важна скорость (например, видеостриминг, онлайн-игры).

**4. Прикладной уровень**

* Отвечает за взаимодействие пользователя с сетью через приложения.
* Предоставляет различные сервисы и протоколы для работы с данными:
  + **HTTP/HTTPS**: передача веб-страниц (браузеры).
  + **FTP**: передача файлов.
  + **SMTP/POP3/IMAP**: работа с электронной почтой.
  + **DNS**: преобразование доменных имен в IP-адреса.
  + **Telnet/SSH**: удаленное управление устройствами.

**7. Технология TCP/IP. Передача пакетов через составную сеть**

**TCP/IP** позволяет передавать данные через составную сеть, используя пакеты и маршрутизаторы.

1. **Источник разбивает данные на пакеты.**
2. **Каждый пакет содержит IP-адрес назначения.**
3. **Пакеты передаются через маршрутизаторы, которые выбирают оптимальный путь.**
4. **На стороне получателя пакеты собираются в исходный поток данных.**

**1. Разбиение данных на пакеты**

* Источник (например, компьютер или сервер) разбивает большой блок данных на небольшие части — **пакеты**.
* Каждый пакет содержит:
  + **IP-адрес назначения**: указывает, куда нужно доставить пакет.
  + **IP-адрес источника**: указывает, откуда пакет был отправлен.
  + **Данные**: часть исходного сообщения.
  + **Заголовки**: служебная информация для маршрутизации и обработки.

**2. Маршрутизация пакетов**

* Пакеты передаются через сеть, используя **маршрутизаторы**.
* Маршрутизаторы — это устройства, которые:
  + Определяют оптимальный путь для передачи каждого пакета.
  + Используют таблицы маршрутизации, чтобы выбрать следующий узел (маршрутизатор или конечное устройство).
* Пакеты могут проходить через несколько маршрутизаторов, прежде чем достигнут назначения.

**3. Передача через составную сеть**

* **Составная сеть** — это сеть, состоящая из нескольких подсетей, соединенных маршрутизаторами.
* Пример:
  + Пакет отправляется из локальной сети (например, домашней) через маршрутизатор в интернет.
  + Затем он проходит через несколько промежуточных маршрутизаторов, принадлежащих интернет-провайдерам.
  + Наконец, пакет достигает конечного устройства (например, веб-сервера).
* Пакеты могут проходить разными путями, что делает сеть устойчивой к отказам.

**4. Сборка пакетов на стороне получателя**

* На стороне получателя (например, сервера или компьютера) пакеты собираются в исходный поток данных.
* Это делается с помощью протокола транспортного уровня (например, TCP), который:
  + Убеждается, что все пакеты доставлены.
  + Восстанавливает порядок пакетов, если они пришли не по порядку.
  + Отправляет подтверждения (ACK) обратно отправителю.
* Если какие-то пакеты потерялись, отправитель повторно их отправляет.

**8. Протокол IP**

Протокол IP работает на сетевом уровне. Основные функции:

* Маршрутизация пакетов.
* Адресация с использованием IP-адресов.
* Не гарантирует доставку пакетов.

**Основные функции**

1. **Маршрутизация**:
   * Определяет путь передачи пакетов между устройствами.
   * Использует IP-адреса назначения и таблицы маршрутизации.
2. **Адресация**:
   * Использует IP-адреса для идентификации устройств.
   * Поддерживает IPv4 (32 бита) и IPv6 (128 бит).
3. **Не гарантирует доставку**:
   * Пакеты могут быть потеряны, искажены или доставлены не по порядку.
   * Эти задачи выполняются протоколами транспортного уровня, такими как TCP.

**Заголовок**: содержит информацию о маршрутизации (IP-адреса, TTL, контрольная сумма).

**9. Технология TCP/IP. Адресация**

* **IP-адрес** — это уникальный идентификатор устройства в сети.
* Он используется для маршрутизации пакетов между устройствами.
* Существует две версии IP-адресов:
  + **IPv4**: 32-битные адреса, например, 192.168.1.1.
  + **IPv6**: 128-битные адреса, например, 2001:0db8:85a3::8a2e:0370:7334.

IP-адреса в IPv4 разделены на классы, которые определяют размер сети и количество устройств в ней.

**1. Класс A**

* **Диапазон**: 1.0.0.0 – 126.0.0.0.
* **Маска подсети**: 255.0.0.0.
* **Количество сетей**: 126 (127 зарезервирован для специальных целей).
* **Устройства в сети**: до 16 777 214 устройств.
* **Пример**: 10.0.0.1.

**2. Класс B**

* **Диапазон**: 128.0.0.0 – 191.255.0.0.
* **Маска подсети**: 255.255.0.0.
* **Количество сетей**: 16 384.
* **Устройства в сети**: до 65 534 устройств.
* **Пример**: 172.16.0.1.

**3. Класс C**

* **Диапазон**: 192.0.0.0 – 223.255.255.0.
* **Маска подсети**: 255.255.255.0.
* **Количество сетей**: 2 097 152.
* **Устройства в сети**: до 254 устройств.
* **Пример**: 192.168.1.1.

**4. Классы D и E**

* **Класс D**: 224.0.0.0 – 239.255.255.255 (используется для многоадресной рассылки).
* **Класс E**: 240.0.0.0 – 255.255.255.255 (зарезервирован для экспериментальных целей).

**10. Организация подсетей. Маска подсети**

Маска подсети определяет, какая часть IP-адреса относится к сети, а какая — к узлу. Она помогает разделить IP-адрес на две части:

1. **Сетевая часть**: идентифицирует сеть.
2. **Хостовая часть**: идентифицирует конкретное устройство в сети.

Например:

* IP-адрес: 192.168.1.1.
* Маска: 255.255.255.0.
* Сеть: 192.168.1.0.
* Узел: 1.

Маска подсети позволяет:

1. Разделять большую сеть на несколько подсетей.
2. Определять, к какой сети принадлежит устройство.
3. Упрощать маршрутизацию пакетов.

**11. Маски переменной длины (VLSM)**

**VLSM (Variable Length Subnet Mask)** — это метод, который позволяет использовать разные маски подсети для разных частей сети.

* Это помогает более эффективно использовать адресное пространство.
* Пример:
  + Сеть 192.168.1.0/24 можно разделить на подсети с разными масками:
    - Подсеть 1: 192.168.1.0/26 (64 адреса).
    - Подсеть 2: 192.168.1.64/27 (32 адреса).
    - Подсеть 3: 192.168.1.96/28 (16 адресов).

**Преимущество**: можно выделять ровно столько адресов, сколько нужно для каждой подсети.

**12. Суммирование адресов**

**Суммирование адресов (Route Aggregation)** — это метод, который позволяет объединять несколько сетей в одну запись в таблице маршрутизации.

* Это уменьшает размер таблицы и упрощает маршрутизацию.
* Пример:
  + Есть сети: 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24.
  + Их можно объединить в одну запись: 192.168.0.0/22.

**Преимущество**: уменьшает нагрузку на маршрутизаторы.

**13. Протоколы транспортного уровня. Механизмы, гарантирующие доставку**

**TCP (Transmission Control Protocol)**:

* Гарантирует доставку данных.
* Использует подтверждения (ACK) и повторную передачу при потере пакетов.
* Подходит для приложений, где важна надежность (например, веб-сайты, электронная почта).

**UDP (User Datagram Protocol)**:

* Не гарантирует доставку.
* Быстрый, но без подтверждений.
* Подходит для приложений, где важна скорость (например, видеостриминг, онлайн-игры).

**14. Протокол TCP. Использование портов отправителя и получателя**

TCP использует порты для идентификации приложений.

**Порты** — это номера, которые используются для идентификации приложений на устройстве.

Пример:

* + Отправитель: 192.168.1.1:8080 (IP-адрес и порт отправителя).
  + Получатель: 192.168.1.2:80 (IP-адрес и порт получателя).

**Как работает**:

* + Порты помогают различать разные приложения на одном устройстве.
  + Например, браузер может использовать порт 8080, а почтовый клиент — порт 25.

**15. Работа протокола TCP**

1. **Установление соединения (3-way handshake)**:
   * SYN (запрос на соединение).
   * SYN-ACK (подтверждение и запрос).
   * ACK (подтверждение соединения).
2. **Передача данных**:
   * Пакеты передаются с подтверждениями (ACK).
   * Если пакет потерян, он повторно отправляется.
3. **Завершение соединения (4-way handshake)**:
   * FIN (запрос на завершение).
   * ACK (подтверждение запроса).
   * FIN (подтверждение завершения).
   * ACK (подтверждение завершения).

**16. Локальные сети. Топологии. Уровневая модель**

**Топологии**:

* + **Шина**: все устройства подключены к одному кабелю.
  + **Звезда**: все устройства подключены к центральному узлу (например, коммутатору).
  + **Кольцо**: устройства соединены по кругу.
  + **Дерево**: иерархическая структура с центральным узлом.

**Уровневая модель**:

* + Физический, канальный, сетевой, транспортный, прикладной.

**17. Технология Ethernet. Форматы кадров. Адресация**

**Формат кадра Ethernet**:

* + Преамбула: сигнал для синхронизации.
  + MAC-адрес назначения: куда отправляется кадр.
  + MAC-адрес источника: откуда отправлен кадр.
  + Данные: полезная нагрузка.
  + CRC: контрольная сумма для проверки ошибок.

**Адресация**:

* + Используются MAC-адреса (48-битные уникальные идентификаторы устройств).

**18. Алгоритм CSMA/CD**

**CSMA/CD (Carrier Sense Multiple Access with Collision Detection)** — алгоритм доступа к среде:

1. Устройство прослушивает среду (кабель).
2. Если среда свободна, устройство начинает передачу.
3. Если обнаружена коллизия, передача останавливается.
4. После паузы устройство повторно пытается передать данные.
5. При коллизии — повторная передача.

**19. Ограничения механизма CSMA/CD**

**CSMA/CD (Carrier Sense Multiple Access with Collision Detection)** — это алгоритм доступа к среде передачи данных, используемый в сетях Ethernet. Однако у него есть несколько ограничений:

* 1. **Работает только в полудуплексном режиме**:
     + Устройства не могут одновременно передавать и принимать данные.
     + Это ограничивает возможности сети, так как не позволяет использовать полный потенциал пропускной способности.
  2. **Неэффективен на высоких скоростях передачи**:
     + При увеличении скорости передачи (например, Gigabit Ethernet) время обнаружения коллизий сокращается, что делает CSMA/CD менее эффективным.
  3. **Требует времени для обнаружения коллизий**:
     + Если два устройства начинают передавать данные одновременно, возникает коллизия.
     + Для её обнаружения требуется время, что может привести к задержкам.

**20. Протокол ARP**

**ARP (Address Resolution Protocol)** — это протокол канального уровня, который используется для определения MAC-адреса устройства по его IP-адресу.

**Как работает ARP**:

* 1. Устройство с IP-адресом (например, 192.168.1.1) хочет отправить данные устройству с IP-адресом 192.168.1.2.
  2. Оно отправляет широковещательный запрос (ARP-запрос): "Кто имеет IP-адрес 192.168.1.2? Ответьте своим MAC-адресом."
  3. Устройство с IP-адресом 192.168.1.2 отвечает своим MAC-адресом (например, 00:1A:2B:3C:4D:5E).
  4. Отправитель сохраняет эту информацию в своей ARP-таблице для последующих передач.

**Зачем нужен ARP**:

Позволяет преобразовывать IP-адреса в MAC-адреса, что необходимо для передачи данных на канальном уровне.

**21. Коммутаторы. Построение таблицы МАС-адресов**

Коммутаторы создают таблицу MAC-адресов, чтобы передавать данные только на нужный порт.

**Как работает**:

* 1. Когда устройство подключается к коммутатору, оно отправляет кадр с собственным MAC-адресом.
  2. Коммутатор запоминает MAC-адрес устройства и порт, к которому оно подключено, в таблице MAC-адресов.
  3. При передаче данных коммутатор использует эту таблицу, чтобы отправить кадр только на нужный порт.

**Преимущества**:

* 1. Уменьшает коллизии, так как данные передаются только на нужный порт.
  2. Повышает производительность сети.

**22. Протокол STP**

**STP (Spanning Tree Protocol)** — это протокол, который предотвращает петли в сети, создавая логическое дерево.

**Зачем нужен STP**:

В сетях с несколькими путями (например, кольцевая топология) могут возникать петли, которые приводят к дублированию пакетов и перегрузке сети.

**Как работает STP**:

* Сеть анализируется, и создается логическое дерево без петель.
* Некоторые связи отключаются, чтобы избежать дублирования пакетов.
* Если одна из связей выходит из строя, STP автоматически активирует резервную связь.

**Преимущества**:

* + Повышает надежность сети.
  + Предотвращает перегрузку и дублирование пакетов.

**23. Виртуальные локальные сети (VLAN)**

VLAN позволяют логически разделять сеть на несколько подсетей, даже если они физически находятся на одном коммутаторе.

**Как работает VLAN**:

* 1. Устройства в одной VLAN могут обмениваться данными, как будто они подключены к одной физической сети.
  2. Устройства из разных VLAN не могут обмениваться данными без использования маршрутизатора.

**Пример**:

* 1. Физически все устройства подключены к одному коммутатору.
  2. Логически их можно разделить на:
     + VLAN 1: для сотрудников.
     + VLAN 2: для гостей.
     + VLAN 3: для серверов.

**Преимущества**:

* 1. Упрощает управление сетью.
  2. Повышает безопасность, так как устройства из разных VLAN не могут взаимодействовать напрямую.
  3. Экономит ресурсы, так как не требуется несколько физических сетей.

### **24. Статические и динамические VLAN**

\*\*VLAN (Virtual Local Area Network)\*\* — это технология, которая позволяет логически разделить сеть на несколько изолированных сегментов, даже если устройства физически подключены к одному коммутатору. Это улучшает управление сетью, безопасность и снижает широковещательный трафик.

#### Статические VLAN

- \*\*Статические VLAN\*\* настраиваются вручную на коммутаторе.

- Администратор назначает порты коммутатора конкретным VLAN.

- Устройства, подключенные к этим портам, автоматически становятся членами соответствующего VLAN.

- \*\*Преимущества\*\*: Простота настройки, стабильность.

- \*\*Недостатки\*\*: Требуется ручное управление при изменении конфигурации.

#### Динамические VLAN

- \*\*Динамические VLAN\*\* автоматически назначают устройства в VLAN на основе их MAC-адресов.

- Используется сервер VLAN (например, DHCP-сервер или специальный сервер управления VLAN).

- Устройства регистрируются в VLAN при подключении к сети.

- \*\*Преимущества\*\*: Автоматизация, удобство масштабирования.

- \*\*Недостатки\*\*: Сложность настройки, зависимость от сервера.

---

### **25. Взаимодействие между VLAN**

По умолчанию устройства из разных VLAN не могут взаимодействовать напрямую, так как VLAN изолируют трафик. Для взаимодействия между VLAN используются:

1. \*\*Маршрутизатор (Router)\*\*:

- Трафик между VLAN передается через маршрутизатор, который настраивается для каждого VLAN.

- Каждый VLAN имеет свой IP-адрес на маршрутизаторе.

2. \*\*Коммутатор с функцией маршрутизации (Layer 3 Switch)\*\*:

- Коммутаторы Layer 3 выполняют функции маршрутизации между VLAN.

- Упрощает управление и снижает стоимость по сравнению с использованием внешнего маршрутизатора.

3. \*\*Транковые порты (Trunk Ports)\*\*:

- Используются для передачи трафика между коммутаторами.

- Поддерживают несколько VLAN одновременно.

---

### **26. Взаимодействие между терминалами**

Взаимодействие между устройствами (терминалами) в сети происходит через:

1. \*\*IP-адреса\*\*: Устройства используют IP-адреса для обмена данными.

2. \*\*Протоколы канального уровня (MAC-адреса)\*\*: Устройства обмениваются кадрами с использованием MAC-адресов.

3. \*\*Маршрутизаторы\*\*: Если устройства находятся в разных сетях, трафик передается через маршрутизатор.

4. \*\*VLAN\*\*: Если устройства находятся в разных VLAN, взаимодействие происходит через маршрутизатор или коммутатор Layer 3.

---

### **27. Протокол DHCP**

\*\*DHCP (Dynamic Host Configuration Protocol)\*\* — это протокол, который автоматически назначает IP-адреса и другие параметры сети (маску подсети, шлюз по умолчанию, DNS) устройствам в сети.

#### Основные функции:

1. \*\*Автоматизация настройки\*\*: Устройства получают IP-адреса без ручной настройки.

2. \*\*Управление адресами\*\*: Адреса выделяются из пула, и их можно переназначать.

3. \*\*Поддержка аренды (Lease)\*\*: Устройства получают IP-адреса на ограниченное время, после чего их нужно обновлять.

#### Процесс работы DHCP:

1. \*\*DHCP Discover\*\*: Устройство отправляет широковещательный запрос в сеть.

2. \*\*DHCP Offer\*\*: Сервер предлагает IP-адрес.

3. \*\*DHCP Request\*\*: Устройство запрашивает адрес.

4. \*\*DHCP Ack\*\*: Сервер подтверждает назначение адреса.

---

### **28. NAT и NAT Overload**

\*\*NAT (Network Address Translation)\*\* — это технология, которая позволяет преобразовывать частные IP-адреса в публичные для выхода в Интернет.

#### NAT:

- Преобразует частный IP-адрес устройства в один публичный IP-адрес.

- Используется для экономии публичных IP-адресов.

- Работает на маршрутизаторе.

#### NAT Overload (PAT):

- Позволяет множеству устройств использовать один публичный IP-адрес.

- Порты используются для различения трафика от разных устройств.

- Наиболее распространенная реализация NAT.

---

### **29. Протоколы маршрутизации “Distance Vector” и “Link-State”**

#### Distance Vector:

- Маршрутизаторы обмениваются таблицами маршрутизации с соседями.

- Используют метрику (например, количество "прыжков") для выбора пути.

- Примеры: \*\*RIP\*\*, \*\*IGRP\*\*.

- \*\*Недостатки\*\*: Медленная сходимость, проблемы с петлями.

#### Link-State:

- Маршрутизаторы создают топологическую карту сети.

- Используют алгоритм Дейкстры для выбора оптимального пути.

- Примеры: \*\*OSPF\*\*, \*\*IS-IS\*\*.

- \*\*Преимущества\*\*: Быстрая сходимость, устойчивость к петлям.

---

### **30. Протокол RIP**

\*\*RIP (Routing Information Protocol)\*\* — это протокол маршрутизации типа \*\*Distance Vector\*\*.

#### Основные характеристики:

- Использует метрику "количество прыжков" (максимум 15).

- Обновления передаются каждые 30 секунд.

- Простой в настройке, но не подходит для больших сетей.

---

### **31. Протокол OSPF. Принцип работы. Алгоритм Dijkstra**

\*\*OSPF (Open Shortest Path First)\*\* — это протокол маршрутизации типа \*\*Link-State\*\*.

#### Принцип работы:

1. Маршрутизаторы обмениваются информацией о состоянии связей (Link-State Advertisements, LSA).

2. Создается топологическая карта сети.

3. Используется алгоритм Дейкстры для выбора оптимального пути.

#### Алгоритм Дейкстры:

- Находит кратчайший путь от одного маршрутизатора к остальным.

- Основан на метрике (например, пропускной способности).

### **32. Иерархическая маршрутизация в OSPF**

OSPF использует иерархическую структуру:

1. \*\*Backbone Area (Area 0)\*\*: Центральная зона, через которую проходят все маршруты.

2. \*\*Другие зоны (Area 1, Area 2, ...)\*\*: Логические сегменты сети.

3. \*\*Абстракция\*\*: Уменьшает объем информации, передаваемой между зонами.

---

### **33. Протокол внешней маршрутизации BGP. Принцип работы**

\*\*BGP (Border Gateway Protocol)\*\* — это протокол внешней маршрутизации, используемый для обмена маршрутами между автономными системами (AS).

#### Принцип работы:

- Маршрутизаторы обмениваются информацией о доступных маршрутах.

- Использует политики для выбора путей.

- Обеспечивает маршрутизацию в Интернете.

### **34. Требования к сети при передаче речи**

При передаче голоса через сеть (например, VoIP) важны следующие параметры:

1. \*\*Задержка (Latency)\*\*:

- Максимальная допустимая задержка: \*\*150 мс\*\* (односторонняя) или \*\*300 мс\*\* (двусторонняя, включая эхо).

- Задержка свыше 150 мс ухудшает восприятие разговора.

2. \*\*Потеря пакетов (Packet Loss)\*\*:

- Допустимая потеря пакетов: \*\*менее 1%\*\*.

- Потеря пакетов приводит к искажениям звука и потере слов.

3. \*\*Джиттер (Jitter)\*\*:

- Разброс задержек между пакетами.

- Допустимый джиттер: \*\*менее 30 мс\*\*.

- Джиттер вызывает неравномерность звука.

4. \*\*Пропускная способность (Bandwidth)\*\*:

- Для одного голосового канала требуется \*\*64 Кбит/с\*\* (G.711) или \*\*20-30 Кбит/с\*\* (кодеки с потерями, например, G.729).

5. \*\*Стабильность соединения\*\*:

- Сеть должна обеспечивать постоянную доступность и низкую вероятность разрыва соединения.

---

### **35. Влияние задержки на качество передачи речи. Компоненты задержки**

#### Влияние задержки:

- \*\*Задержка до 150 мс\*\*: Разговор воспринимается естественно.

- \*\*Задержка 150-400 мс\*\*: Возникает эхо, сложности с общением.

- \*\*Задержка более 400 мс\*\*: Разговор становится невозможным.

#### Компоненты задержки:

1. \*\*Сетевая задержка\*\*:

- Задержка в сети (например, в маршрутизаторах, коммутаторах).

- Включает задержку на передачу данных по каналам связи.

2. \*\*Задержка кодирования\*\*:

- Время, затрачиваемое на кодирование и декодирование аудио.

- Зависит от используемого кодека (например, G.711, G.729).

3. \*\*Задержка буферизации\*\*:

- Время, необходимое для сглаживания джиттера.

- Используется буфер для хранения пакетов перед воспроизведением.

4. \*\*Задержка эхокомпенсации\*\*:

- Время, затрачиваемое на обработку эха (особенно в двусторонних соединениях).

---

### **36. Протокол RTP**

\*\*RTP (Real-time Transport Protocol)\*\* — это протокол, используемый для передачи аудио и видео в реальном времени.

#### Основные функции:

1. \*\*Синхронизация\*\*:

- Обеспечивает порядок доставки пакетов и синхронизацию аудио/видео.

2. \*\*Маркировка времени\*\*:

- В каждом пакете указывается временная метка для синхронизации.

3. \*\*Использование UDP\*\*:

- RTP работает поверх UDP, что обеспечивает низкую задержку.

#### Структура пакета RTP:

- \*\*Версия (2 бита)\*\*: Текущая версия — 2.

- \*\*Маркер (1 бит)\*\*: Используется для обозначения важных событий.

- \*\*Номер последовательности (16 бит)\*\*: Позволяет восстановить порядок пакетов.

- \*\*Временная метка (32 бита)\*\*: Указывает время создания пакета.

- \*\*Идентификатор источника (32 бита)\*\*: Уникальный идентификатор источника.

#### Применение:

- Используется в VoIP, видеоконференциях, потоковом вещании.

---

### **37. Полоса, требуемая при передаче речи**

Для передачи голоса через сеть требуется определенная пропускная способность, зависящая от используемого кодека:

1. \*\*G.711 (PCM)\*\*:

- Скорость: \*\*64 Кбит/с\*\*.

- Высокое качество звука, но высокие требования к пропускной способности.

2. \*\*G.729\*\*:

- Скорость: \*\*8 Кбит/с\*\* (сжатие до 8 Кбит/с).

- Снижение пропускной способности, но с потерей качества.

3. \*\*G.723.1\*\*:

- Скорость: \*\*6.3 Кбит/с\*\* или \*\*5.3 Кбит/с\*\*.

- Очень низкая пропускная способность, но низкое качество.

#### Пример:

- Для одного голосового канала с использованием G.711 требуется \*\*64 Кбит/с\*\*.

- Для 100 одновременных звонков потребуется \*\*6.4 Мбит/с\*\*.

---

### **38. Механизмы повышения эффективности передачи речи**

Для повышения эффективности передачи речи используются следующие механизмы:

1. \*\*Компрессия аудио (Кодеки)\*\*:

- Использование кодеков с потерями (например, G.729) для снижения пропускной способности.

2. \*\*Буферизация\*\*:

- Сглаживание джиттера с помощью буферов.

3. \*\*QoS (Quality of Service)\*\*:

- Приоритезация голосового трафика для снижения задержек и потерь пакетов.

4. \*\*Эхокомпенсация\*\*:

- Устранение эха в двусторонних соединениях.

5. \*\*Динамическое управление трафиком\*\*:

- Ограничение пропускной способности для других типов трафика во время передачи голоса.

---

### **39. Виды трафика в мультисервисной сети**

Мультисервисная сеть поддерживает несколько типов трафика:

1. \*\*Голосовой трафик (VoIP)\*\*:

- Передача речи в реальном времени.

- Требования: низкая задержка, низкая потеря пакетов.

2. \*\*Видеотрафик\*\*:

- Передача видео (видеоконференции, потоковое вещание).

- Требования: высокая пропускная способность, низкая задержка.

3. \*\*Данные\*\*:

- Веб-трафик, email, файлообмен.

- Требования: высокая пропускная способность, но без жестких ограничений по задержке.

4. \*\*Игровые сервисы\*\*:

- Трафик от онлайн-игр.

- Требования: низкая задержка, стабильность соединения.

5. \*\*Трафик IoT\*\*:

- Данные от устройств интернета вещей.

- Требования: низкая пропускная способность, но высокая надежность.

---

### **40. Качество обслуживания (QoS). Параметры QoS. Рекомендации Y.1540, Y.1541**

\*\*QoS (Quality of Service)\*\* — это набор механизмов для управления трафиком и обеспечения качества обслуживания.

#### Параметры QoS:

1. \*\*Задержка (Latency)\*\*:

- Максимально допустимая задержка для голосового трафика — 150 мс.

2. \*\*Потеря пакетов (Packet Loss)\*\*:

- Допустимая потеря пакетов для голосового трафика — менее 1%.

3. \*\*Джиттер (Jitter)\*\*:

- Разброс задержек между пакетами.

- Допустимый джиттер — менее 30 мс.

4. \*\*Пропускная способность (Bandwidth)\*\*:

- Обеспечение необходимой полосы для каждого типа трафика.

#### Рекомендации Y.1540, Y.1541:

- \*\*Y.1540\*\*: Определяет параметры качества для IP-сетей (задержка, потеря пакетов, джиттер).

- \*\*Y.1541\*\*: Устанавливает классы обслуживания (Class of Service) для различных типов трафика.

---

### **41. Правила политики / Формирование трафика**

Политика (Policy) и формирование трафика (Traffic Shaping) используются для управления трафиком в сети.

#### Основные функции:

1. \*\*Приоритезация трафика\*\*:

- Голосовой трафик получает более высокий приоритет.

2. \*\*Ограничение пропускной способности\*\*:

- Ограничение полосы для менее важного трафика (например, файлообмен).

3. \*\*Формирование трафика\*\*:

- Сглаживание пиков трафика для предотвращения перегрузки.

4. \*\*Маркировка пакетов\*\*:

- Использование меток (например, DSCP) для идентификации типа трафика.

---

### **42. Механизмы управления перегрузкой: Tail Drop и RED**

#### Tail Drop:

- Пакеты отбрасываются, если буфер заполнен.

- Простой в реализации, но может вызывать резкие скачки задержки.

#### RED (Random Early Detection):

- Пакеты отбрасываются раньше, чтобы избежать перегрузки.

- Снижает вероятность резких скачков задержки.

---

### **43. Взаимодействие между терминалами**

Взаимодействие между терминалами (устройствами) в сети происходит через:

1. \*\*IP-адреса\*\*:

- Устройства используют IP-адреса для обмена данными.

2. \*\*MAC-адреса\*\*:

- На канальном уровне устройства обмениваются кадрами с использованием MAC-адресов.

3. \*\*Маршрутизаторы\*\*:

- Если устройства находятся в разных сетях, трафик передается через маршрутизатор.

4. \*\*VLAN\*\*:

- Если устройства находятся в разных VLAN, взаимодействие происходит через маршрутизатор или коммутатор Layer 3.

---

### **44. Протокол DHCP**

\*\*DHCP (Dynamic Host Configuration Protocol)\*\* — это протокол для автоматического назначения IP-адресов и других параметров сети.

#### Основные функции:

1. \*\*Автоматизация настройки\*\*:

- Устройства получают IP-адреса без ручной настройки.

2. \*\*Управление адресами\*\*:

- Адреса выделяются из пула и могут быть переназначены.

3. \*\*Аренда (Lease)\*\*:

- Устройства получают IP-адреса на ограниченное время, после чего их нужно обновлять.

#### Процесс работы DHCP:

1. \*\*DHCP Discover\*\*: Устройство отправляет широковещательный запрос.

2. \*\*DHCP Offer\*\*: Сервер предлагает IP-адрес.

3. \*\*DHCP Request\*\*: Устройство запрашивает адрес.

4. \*\*DHCP Ack\*\*: Сервер подтверждает назначение адреса.

---

### **45. NAT и NAT Overload**

\*\*NAT (Network Address Translation)\*\* — это технология для преобразования частных IP-адресов в публичные.

#### NAT:

- Преобразует частный IP-адрес устройства в один публичный IP-адрес.

- Используется для экономии публичных IP-адресов.

#### NAT Overload (PAT):

- Позволяет множеству устройств использовать один публичный IP-адрес.

- Порты используются для различения трафика от разных устройств.

---

### **46. Протокол ICMP**

\*\*ICMP (Internet Control Message Protocol)\*\* — это протокол для передачи сообщений об ошибках и управления сетью.

#### Основные функции:

1. \*\*Сообщения об ошибках\*\*:

- Уведомление отправителя о проблемах (например, недоступность хоста).

2. \*\*Управление сетью\*\*:

- Проверка доступности устройств (Ping).

- Определение маршрута (Traceroute).

#### Примеры сообщений ICMP:

- \*\*Echo Request/Reply (Ping)\*\*: Проверка доступности устройства.

- \*\*Destination Unreachable\*\*: Уведомление о недоступности хоста.

- \*\*Time Exceeded\*\*: Уведомление о превышении времени жизни пакета (TTL).

---