Вопрос 1. Краткая история компьютерных сетей. Кто стоит у истоков создания сетей, какие личности и организации?
Вопрос 2.Охарактеризуйте два нижних уровня модели OSI, перечислите типы проводящей среды, приведите примеры стандартов и характеристики среды передачи данных.
Вопрос 3. Кодирование информации в вычислительных сетях: Основные типы кодов, их достоинства и недостатки. Зачем нужны несколько уровней кодирования?
Вопрос 4. Изложите принципы цифрового и аналогового кодирования, приведите примеры кодирования цифровыми и аналоговыми кодами, поясните, что где применяется.
Вопрос 5. Перечислить виды адресации в сетях, дать краткое описание, на каких уровнях модели OSI они используются.
Вопрос 6. Произвести краткую сравнительную характеристику всех известных сетевых топологий.
Вопрос 7. Перечислить и охарактеризовать алгоритмы доступа к среде передачи данных в компьютерных сетях, назвать их достоинства и недостатки.
Вопрос 8. Протокол Token Ring (High Speed Token Ring), принципы его работы.
Вопрос 9. Алгоритм скользящего окна. Основные параметры и характеристики.
Вопрос 10. Охарактеризовать способ доступа к среде в сетях Ethernet. Привести основные параметры.
Вопрос 11. Протоколы канального уровня Ethernet (10Mbps-10 Gbps). Общность и различия. Как обеспечивается их обратная совместимость?
Вопрос 12. Привести технические параметры каналов связи в компьютерных сетях.
Вопрос 13. Что максимально влияет на пропускную способность каналов связи, приведите формулы с пояснениями.
Вопрос 14. Охарактеризовать алгоритм CSMA/CD. Перечислить его основные параметры.
Вопрос 15. Какими средствами была достигнута преемственность стандартов Ethernet?
Вопрос 16. Алгоритм CSMA/CD и его разновидности. Охарактеризовать, назвать сферы применения.
Вопрос 17. Дать определение домена коллизий. Как он различается на разных устройствах и почему?
Вопрос 18. Классификация методов доступа к среде передачи данных, их характеристики, достоинства и недостатки.

Вопрос 19. Нарисовать структуру кадра Ethernet, дать краткое описание полей кадра.
Вопрос 20. Стандартные сетевые протоколы (в пределах изученных стеков протоколов). Перечислить, поставить в соответствие модели OSI.
Вопрос 21. Протоколы прикладного уровня. Перечислить, назвать основные функции.
Вопрос 22. Транспортные протоколы. Перечислить, назвать основные функции
Вопрос 23. Протоколы сетевого уровня. Перечислить, назвать основные функции.
Вопрос 24. Способы получения адресов узлами. Назвать специальные IP адреса.
Вопрос 25. Стек протоколов TCP/IP. Структура, достоинства, недостатки.
Вопрос 26. Протокол TCP. Назначение, основные поля заголовка, предназначение полей, алгоритм работы.
Вопрос 27. Мультиплексирование и демультиплексирование на транспортном уровне, порты и сокеты.
Вопрос 28. Протокол IP. Назначение, основные поля заголовка, предназначение полей, алгоритм работы, принципы маршрутизации.
Вопрос 29. Правила IP-адресации.
Вопрос 30. Структура и особенности IPv6. Какие функции были в нем оптимизированы?
Вопрос 31. Протоколы канального уровня. Перечислить, произвести сравнительную характеристику.
Вопрос 32. Протоколы ARP, RARP. Назначение. Алгоритм работы.
Вопрос 33. Нарисовать модель соответствия стека TCP/IP модели OSI. Зачем нужно разделение на уровни?
Вопрос 34. Охарактеризовать функции сетевого и канального уровней OSI. Привести примеры протоколов, на них работающих.
Вопрос 35. Модель OSI. Назначение, характеристика уровней. Понятия инкапсуляции и декапсуляции.
Вопрос 36. Назвать назначение и порядок функционирования ICMP.
Вопрос 37. Перечислить виды маршрутизирущих и маршрутизируемых протоколов и их основные характеристики.
Вопрос 38. Принципы работы протокола DNS, структура доменных имен. Зачем нужны хранители ключей?
Вопрос 39. Алгоритм работы «прозрачного» моста.
Вопрос 40. Алгоритмы, которые могут использовать различные коммутаторы в своей работе.
Вопрос 41. Структурные схемы коммутаторов, перечислить, охарактеризовать достоинства и недостатки.

Вопрос 42. Сетевые проблемы, решенные за счет использования коммутаторов.
Вопрос 43. Ограничения коммутаторов (какие сетевые вопросы они не решают)
Вопрос 44. Перечислить виды маршрутизации и способы занесения записей в таблицы.
Вопрос 45. Ключевые моменты алгоритма работы моста с маршрутизацией от источника.
Вопрос 46. Назначение технологии NAT. Способы трансляции адресов, особенности каждого из видов. Пример.
Вопрос 47. Перечислить дополнительные функции коммутаторов.
Вопрос 48. Провести сравнительную характеристику коммутаторов и маршрутизаторов (сферы использования, уровень OSI, типы адресации, алгоритмы работы).
Вопрос 49. Виртуальные частные сети. В каких случаях появляется необходимость в создании виртуальных сегментов? Приведите примеры.
Вопрос 50. Назвать наиболее слабые механизмы в работе коммутаторов и существующие методы борьбы с ними.
Вопрос 51. Перечислить виды конструктивного исполнения сетевых устройств, назвать достоинства и недостатки.
Вопрос 52. Перечислить сетевые устройства, поставить их в соответствие модели OSI.
Вопрос 53. Какие из изученных сетевых технологий могут быть использованы для обеспечения сетевой безопасности и каким образом?
Вопрос 54. Типы организации VLAN. Дать краткую сравнительную характеристику.
Вопрос 55. Разновидности и правила формирования ACL.
Вопрос 56. Отличие технологии VPN от VLAN

1. Краткая история компьютерных сетей. Кто стоит у истоков создания сетей, какие личности и организации?

в 1957-м году по указу президента США Дуайта Эйзенхауэра в подразделении отдела государственной обороны был сформирован агентство по передовым оборонным исследованиям (Defence Advanced Research Projects Agency - DARPA).

История развития сетей началась с того, что в начале 60-х годов основные усилия DARPA были направлены на то, чтобы соединить между собой два мейнфрейма, удаленные друг от друга на большое расстояние (находящиеся в двух разных штатах).

Агентство выделяет большие средства для привлечения к разработке передовых умов различных национальных университетов и образовательных центров. Одной из таких "находок" для DARPA стал Массачусетский Технологический Институт - «MIT».

Именно в стенах университета «MIT» получила свое начало история развития сетей. По низкоскоростным коммутируемым телефонным линиям были соединены между собой два суперкомпьютера (мейнфрейма), один из которых находился в Массачусетсе, а другой - в Калифорнийском университете. По такому же принципу (через телефонные линии связи) к основному компьютеру происходило и подключение удаленных на большие расстояния терминалов.

В 1969-ом году министерство обороны США объединило в одну сеть суперкомпьютеры нескольких оборонных и научно-исследовательских центров:

- University of California (Los Angeles)
- Stanford Research Institute
- University of California (Santa Barbara)
- University of Utah

Эта сеть получила название «ARPANET». В подобной структуре уже присутствовало такое понятие, как распределенная обработка данных. Как вы понимаете, для военных – это критически важный параметр. Устраняется единый центр обработки, который может быть потенциальной мишенью противника.

2. Охарактеризуйте два нижних уровня модели OSI, перечислите типы проводящей среды, приведите примеры стандартов и характеристики среды передачи данных.

На двух нижних уровнях модели OSI располагаются физический и канальный уровни.

Физический уровень отвечает за передачу и приём битов в виде электрических, световых и радиосигналов и определяет характеристики среды передачи данных.

Канальный уровень обеспечивает передачу данных в пределах одного сегмента сети. На этом уровне биты формируются в кадры и выполняется проверка целостности с помощью CRC. В случае обнаружения ошибки кадр отбрасывается. Передача осуществляется на основе MAC-адресов.

Канальный уровень делится на два подуровня: MAC-подуровень, который управляет доступом к среде передачи данных, и LLC-подуровень, обеспечивающий логическое взаимодействие с сетевым уровнем.

Типы проводящей среды:

1. проводная (витая пара, коаксиальный кабель, оптоволокно)
2. беспроводная (вифи, блютуз)

Примеры стандартов и характеристики среды передачи данных:

1. стандарт 100BASE-TX использует витую пару, обеспечивает скорость передачи данных до 100 Мбит/с и максимальную длину сегмента до 100 метров.
2. стандарт 10base-fx использует оптоволоконный кабель и обеспечивает скорость передачи данных до 10 Мбит/с и максимальную длину сегмента 2км

Средой передачи информации называются те линии связи (или каналы связи), по которым производится обмен информацией между компьютерами.

Характеристики среды передачи данных:

1. Полоса пропускания кабеля и затухание сигнала. Два этих параметра тесно связаны между собой, так как с ростом частоты сигнала растет затухание сигнала. Затухание измеряется в децибелах и пропорционально длине кабеля.
2. Помехозащищенность кабеля и обеспечиваемая им секретность передачи информации. Эти параметры показывают, как кабель реагирует на внешние помехи и насколько просто прослушать информацию.
3. Скорость распространения сигнала по кабелю или задержка сигнала. Типичные величины скорости распространения сигнала – от 0,6 до 0,8 скорости распространения света. Типичные величины задержек – от 4 до 5 нс/м.
4. Волновое сопротивление кабеля. Типичные значения – от 50 до 150 Ом.

Витые пары:

Кабель на основе витых пар представляет собой несколько пар скрученных попарно изолированных медных проводов в единой диэлектрической оболочке. Он довольно гибкий и удобный для прокладки.

Неэкранированные витые пары характеризуются слабой защищенностью от внешних электромагнитных помех и от подслушивания.

Экранированная витая пара STP имеет металлическую оплетку-экран для защиты от помех и снижения перекрестных наводок (crosstalk). Экран должен быть заземлен.

Коаксиальный кабель:

Коаксиальный кабель представляет собой электрический кабель, состоящий из центрального медного провода и металлической оплетки, разделенных между собой слоем диэлектрика.

Обладает высокой помехозащищенностью (благодаря металлической оплетке), более широкими полосами пропускания (свыше 1ГГц), а также большими допустимыми расстояниями передачи (до километра).

Оптоволоконный кабель:

Информация по нему передается не электрическим сигналом, а световым. Главный его элемент – это прозрачное стекловолокно, по которому свет проходит на огромные расстояния (диаметром около 1 – 10 мкм) с незначительным ослаблением.

Оптоволоконный кабель обладает исключительными характеристиками по помехозащищенности и секретности передаваемой информации. Теоретически возможная полоса пропускания такого кабеля достигает величины 1000 ГГц.

Беспроводная среда передачи данных:

Используется передача информации по радиоволнам. Скорость передачи достигает десятков мегабит в секунду.

Преимущества:

- не требуется прокладка проводов;
- компьютеры можно легко перемещать.

Недостатки:

- плохая защита от прослушивания;
- слабая помехозащищенность.

3. Кодирование информации в вычислительных сетях: Основные типы кодов, их достоинства и недостатки. Зачем нужны несколько уровней кодирования?

Кодирование - процесс преобразования данных (битов) в электрические, световые и радиосигналы для передачи на физическом уровне.

Декодирование - процесс преобразования электрических, световых и радиосигналов полученных на физическом уровне в биты.

Кодирование делится на три уровня:

1. **Логическое кодирование** — преобразование информации в последовательность битов.
2. **Линейное кодирование** — преобразование битов в изменения сигнала (например, NRZ, RZ, Манчестерский, Бифазный), что позволяет приёмнику корректно определить, какой бит (0 или 1) был передан.
3. **Физическое кодирование** — превращение изменений сигнала в реальные физические сигналы (электрические, световые или радиосигналы) для передачи по среде.

```swift
Данные → (логическое кодирование) → Биты
Биты → (линейное кодирование: NRZ/RZ/Манчестер/Бифазный) → Изменения сигнала
Сигнал → (физическое кодирование) → Реальные электрический/световой/радиосигнал
```

1. Логический уровень → «что передаём»
2. Линейный → «как передаём» (синхронизация, помехоустойчивость)
3. Физический → «передаём реально по среде»

Каждый уровень решает свою задачу, что позволяет передавать данные надёжно и независимо от типа физической среды

Код	Как работает	Достоинства	Недостатки
NRZ (Non-Return-to-Zero)	1 — высокий уровень, 0 — низкий	Простая реализация, высокая скорость передачи	Нет самосинхронизации, длинные последовательности 0 или 1 создают ошибки
RZ (Return-to-Zero)	1 — высокий уровень только половину бита, потом ноль; 0 — низкий	Есть синхронизация, лучше помехоустойчивость	Требует больше полосы частот, сложнее реализовать
Манчестерский	0 — переход от высокого к низкому в середине бита, 1 — от низкого к высокому	Отличная синхронизация, легко обнаружить ошибки	Скорость передачи фактически в 2 раза ниже
Бифазный / Differential Manchester	Переход в середине бита + переход в начале бита при 0 или 1	Хорошая синхронизация, меньше влияния длинных последовательностей	Более сложная реализация, увеличенная ширина спектра

4. Изложите принципы цифрового и аналогового кодирования, приведите примеры кодирования цифровыми и аналоговыми кодами, поясните, что где применяется.

Цифровое кодирование — представление информации в виде **двоичных сигналов** (0 и 1). Применяется в компьютерных сетях, Ethernet, Wi-Fi, цифровой телефонии.

Аналоговое кодирование — представление информации с помощью **непрерывных сигналов** (напряжение, частота, амплитуда). Применяется в телефонных линиях, радио, телевизионных сигналах.

Принципы:

Цифровое → дискретные значения, устойчиво к помехам

Аналоговое → непрерывные значения, передаёт естественные сигналы, чувствительно к шуму и помехам

Пример аналогового кодирования (Суть: сигнал непрерывный, амплитуда/частота/фаза меняются плавно)

Пример	Как кодируется	Где применяется
AM (Amplitude Modulation)	Амплитуда несущей меняется пропорционально сигналу	Радио, старое телевизионное вещание
FM (Frequency Modulation)	Частота несущей меняется пропорционально сигналу	Радио (FM), аудиотрансляции
Телевидение аналоговое	Яркость и цвет меняются непрерывно	Старое эфирное и кабельное телевидение
Телефонная связь (аналог)	Напряжение линии изменяется плавно в зависимости от звука	Старые телефонные сети

Амплитудная модуляция: если говорим тихо – амплитуда маленькая, если громко – большая. Частота постоянна.

Частотная модуляция: если звук громче – частота увеличивается, если тише – уменьшается. Амплитуда постоянна.

Фазовая модуляция: бит 0 – фаза не меняется, бит 1 – сдвиг на 180*. Амплитуда и частота постоянны.

Пример цифрового кодирования: (Суть: сигнал дискретный — биты (0 и 1) преобразуются в физические сигналы)

Пример	Как кодируется	Где применяется
NRZ, RZ, Манчестер, Бифазный	Биты → изменения сигнала → электрические / световые / радиосигналы	Ethernet, Wi-Fi, цифровые линии связи
PCM (Pulse Code Modulation)	Аналоговый звук → оцифровка → двоичный поток	Телефония, VoIP
MPEG, H.264	Видео → сжатие → двоичный поток	Цифровое ТВ, видео онлайн
DVB-T/C/S	Видео и аудио → цифровые биты → модуляция (OFDM, QAM)	Цифровое телевидение

5. Перечислить виды адресации в сетях, дать краткое описание, на каких уровнях модели OSI они используются.

1.Физическая адресация (mac):

Располагается на канальном уровне модели osi. Используется для уникальной идентификации устройства в локальной сети. Нужна для передачи кадров внутри сети.

Пример: 00:1A:2B:3C:4D:5E

Структура адреса:

- OUA (24 бита) – организационно уникальный адрес.
- OUI (22 бита) – организационно уникальный идентификатор.
- UAA (совместно OUA с OUI) – универсально управляемый адрес или IEEE-адрес.
- Два старших бита – управляющие: Первый бит I/G указывает на тип адреса; второй бит U/L определяет, как был присвоен адрес данному сетевому адаптеру.

Структура 48-битного стандартного MAC-адреса

Для широковещательной передачи (то есть передачи всем абонентам сети одновременно) используется адрес, все 48 битов которого установлены в единицу.

Данной системы адресов придерживаются такие популярные сети, как Ethernet, Fast Ethernet, Token-Ring, FDDI, 100VG-AnyLAN.

2.Логическая адресация (ip):

Располагается на сетевом уровне модели osi. Используется для идентификации устройства в глобальной или локальной сети. Нужна для маршрутизации пакетов между сетями.

Примеры:

- **IPv4: 192.168.0.1**
- **IPv6: 2001:0db8::1**

Маршрутизатор считывает из IP-пакета только адрес сети, в которой находится узел - приемник пакета, а затем на основе своей таблицы маршрутизации определяет, каким образом доставить пакет в сеть, в которой расположен адресат.

Такой механизм маршрутизации возможен благодаря делению IP-адреса на два компонента:

- идентификатор сети (network ID) — первая часть IP-адреса, представляющая конкретную сеть в более крупной TCP/IP-сети; (первые 2 октета)
- идентификатор узла (host ID) — вторая часть IP-адреса, определяющая узел TCP/IP. (последние 2 октета)

Идентификаторы сетям и узлам назначают по определенным правилам: нельзя присваивать всем битам идентификаторов сети и узла значение 0 и 1; идентификатор узла должен быть уникален в пределах локальной сети.

Класс IP-адреса определяется по значению первого октета. Определено пять классов адресов A, B, C, D, E, из которых для адресации TCP/IP-узлов используются только классы A, B и C.

Форматы IP-адреса

3.Портовая адресация:

Располагается на транспортном уровне уровне модели osi. Идентифицирует конкретно приложение или процесс на устройстве. Позволяет одному устройству принимать несколько потоков данных одновременно

Примеры портов: HTTP → 80, HTTPS → 443, SSH → 22

4.Доменная адресация:

Располагается на прикладном уровне модели osi. Используется для идентификации ресурса или сервиса. Обеспечивает удобную идентификацию ресурсов для человека. Адреса ресурсов (URL, email), используется для идентификации конкретного сервиса или пользователя.

Примеры: URL (https://example.com), email (user@mail.com)

Доменное имя – это имя компьютера, вида www.sait.com. Адресация в Internet происходит по IP-адресам, однако для человека гораздо удобнее доменные имена.
Существует также URL-адрес, т.е. запись вида http://www.sait.com. Доменное имя является частью URL-адреса (схема_передачи:// доменное_имя : порт / имя файла# внутреняя_ссылка).

Используется служба доменных имен DNS для установления соответсвия между доменным именем и IP-адресом. Алгоритм ее функционирования таков:

1. Пользователь в окне Web-браузера вводит http://www.microsoft.com
2. Компьютер пользователя направляет запрос DNS-серверу на установление IP-адреса по доменному имени.
3. DNS-сервер возвращает IP-адрес или перенаправляет запрос дальше по иерархии серверов (более высокого уровня).

6. Произвести краткую сравнительную характеристику всех известных сетевых топологий.

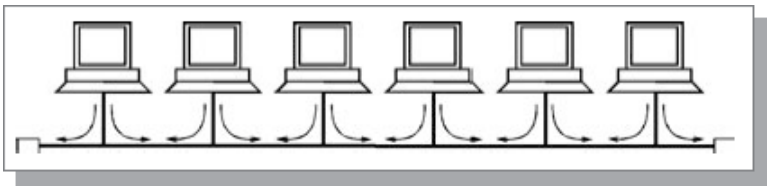
Под топологией компьютерной сети обычно понимается физическое расположение компьютеров сети относительно друг друга и способ соединения их линиями связи. Важно отметить, что понятие топологии относится, прежде всего, к локальным сетям, в которых структуру связей можно легко проследить.

Топология определяет требования к оборудованию, тип используемого кабеля, допустимые и наиболее удобные методы управления обменом, надежность работы, возможности расширения сети.

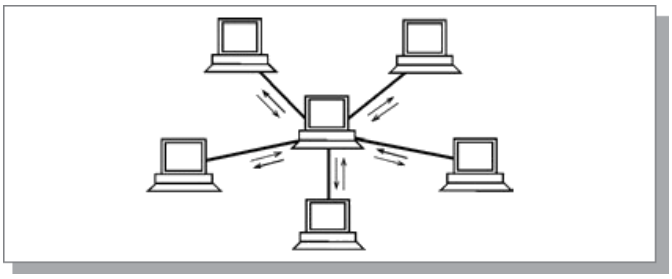
Существует три базовые топологии сети:

- **Шина** (bus) — все компьютеры параллельно подключаются к одной линии связи. Информация от каждого компьютера одновременно передается всем остальным компьютерам.

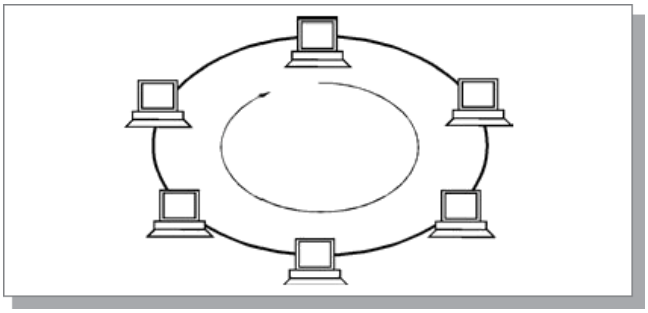
- **Плюсы:** простая реализация, минимум кабеля, дешевая, надежность
- **Минусы**: один обрыв и сеть падает, коллизии, плохо маштабируется
- **Применение**: **старые сети эзернет 10base-2, 10base-5**



- **Звезда** (star) — к одному центральному компьютеру присоединяются остальные периферийные компьютеры, причем каждый из них использует отдельную линию связи. Информация от периферийного компьютера передается только центральному компьютеру, от центрального — одному или нескольким периферийным.
- **Плюсы:** легко маштабируется, отказ одного узла не влияет на остальные, удобно для администрирования
- **Минусы:** отказ центрального устройства и сеть падает, больше кабеля
- **Применение: современные эзернет сети**

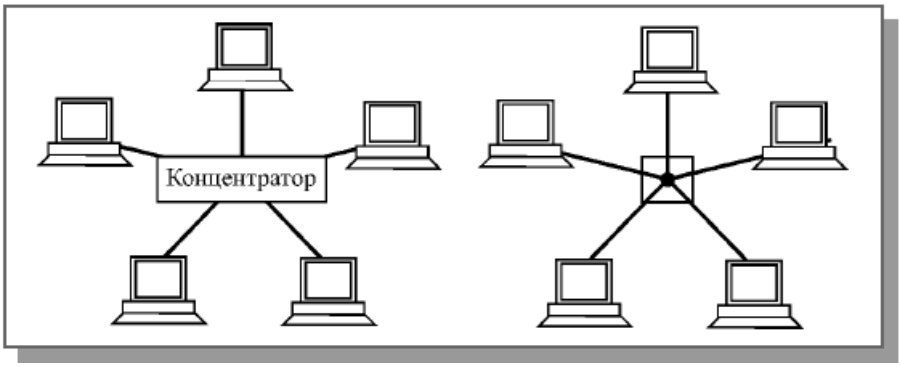


- **Кольцо** (ring) — компьютеры последовательно объединены в кольцо. Передача информации в кольце всегда производится только в одном направлении. Каждый из компьютеров передает информацию только одному компьютеру, следующему в цепочке за ним, а получает информацию только от предыдущего в цепочке компьютера.
- **Плюсы:** нет коллизий, равный доступ к среде
- **Минсуы:** отказ одного узла может нарушить сеть, сложно обслуживать
- **Применение:** токер ринг, FDDI



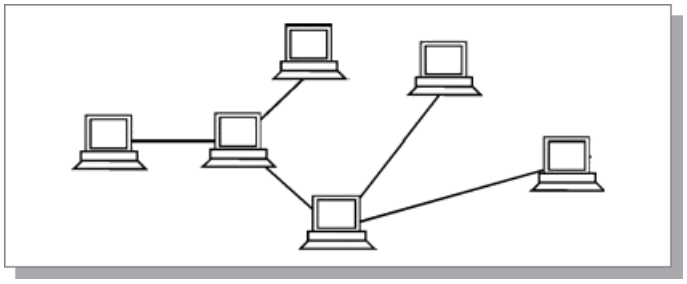
Звезда, показанная ранее, носит название активной или истинной звезды. Существует также топология, называемая пассивной звездой, которая только внешне похожа на звезду. В настоящее время она распространена гораздо более широко, чем активная звезда. Достаточно сказать, что она используется в наиболее популярной сегодня сети Ethernet.

В центре сети с данной топологией помещается не компьютер, а специальное устройство — концентратор или, как его еще называют, хаб (hub), которое выполняет ту же функцию, что и репитер, то есть восстанавливает приходящие сигналы и пересылает их во все другие линии связи.

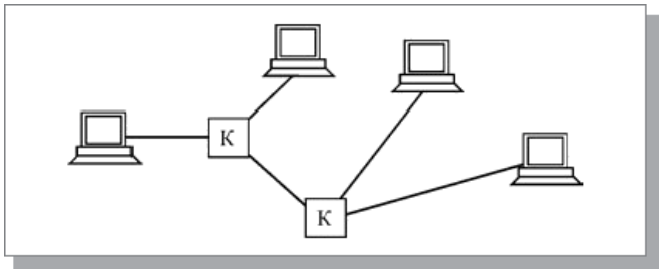


Топология пассивная звезда и ее эквивалентная схема

Кроме трех рассмотренных базовых топологий нередко применяется также сетевая топология дерево, которую можно рассматривать как комбинацию нескольких звезд. Причем, как и в случае звезды, дерево может быть активным или пассивным. При активном дереве в центрах объединения нескольких линий связи находятся центральные компьютеры, а при пассивном — концентраторы (хабы).



Топология активное дерево



Топология пассивное дерево. К — концентраторы

7. Перечислить и охарактеризовать алгоритмы доступа к среде передачи данных в компьютерных сетях, назвать их достоинства и недостатки.

Существует множество алгоритмов доступа. Их выбор зависит от скорости передачи в сети, длины шины, загруженности сети, используемого кода передачи.

Алгоритм CSMA/CD– множественный доступ с контролем несущей и обнаружением коллизий, используемый в сети Ethernet. Его главное достоинство в том, что все абоненты полностью равноправны, и ни один из них не может надолго заблокировать обмен другому (как в случае наличия приоритетов). В этом методе коллизии не предотвращаются, а разрешаются.

Применяется: класический эзерент (10/100), шина, хабы

Плюсы: простота, не требует центрального управления

Минусы: коллизии, падение при высокой загрузке, не подходит под фул-дуплекс

Алгоритм CSMA/CA – множественный доступ с контролем несущей и избежанием коллизий, применяющийся, например, в сети Apple LocalTalk. Абонент, желающий передавать и

обнаруживший освобождение сети, передает сначала короткий кадр запрос на передачу. Затем он заданное время ждет кадр подтверждение запроса от абонента-приемника. Если ответа нет, передача откладывается. Если ответ получен, передается кадр. Коллизии полностью не устраняются, но в основном сталкиваются управляющие кадры. Столкновения информационных кадров выявляются на более высоких уровнях протокола.

Применяется: вифи

Плюсы: подходит для беспроводной сети, минимизирует коллизии

Минусы: больше накладных расходов, ниже скорость по сравнению с эзернет

Маркерный доступ (токен ринг): Компьютер может начать передавать данные в сеть, только если получит от предыдущего компьютера в кольце "маркер" – специальный короткий пакет, свидетельствующий о том, что сеть свободна. Если компьютеру нечего передавать в сеть, то он передает маркер следующему компьютеру в кольце. Если компьютеру есть что передавать, то он уничтожает маркер и передает свой пакет в сеть. Пакет по битам ретранслируется по кольцу от компьютера к компьютеру, адресат получает пакет, устанавливает в пакете биты, подтверждающие, что пакет достиг адресата и передает пакет дальше по кольцу. Наконец, пакет возвращается к отправителю, который уничтожает его и передает в сеть новый маркер. Компьютер может и не передавать в сеть новый маркер, а продолжить передавать кадры данных до тех пор, пока не истечет время удержания маркера. После истечения времени удержания маркера компьютер обязан прекратить передачу собственных данных и передать маркер далее по кольцу. Обычно время удержания маркера по умолчанию равно 10 мс.

Применяется: токен ринг, fddi

Плюсы: нет коллизий, предсказуемое время доступа

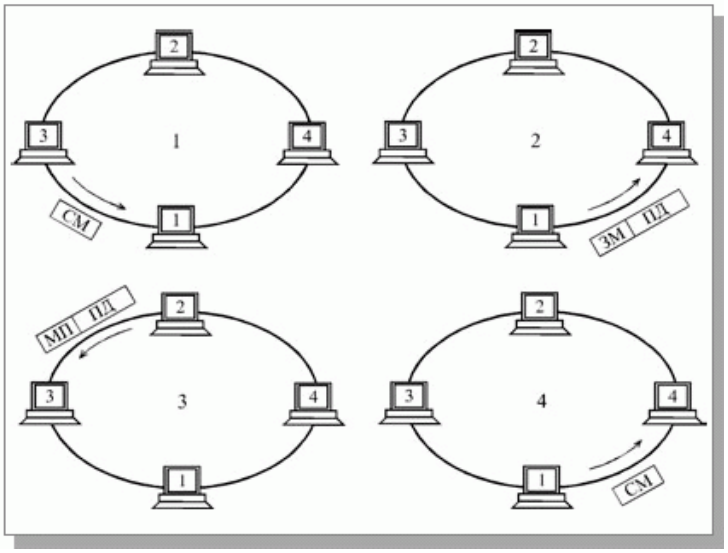
Минусы: сложность реализации, отказ одного узла может повлиять на сеть

«Алгоритмы доступа к среде определяют порядок передачи данных по общей среде и необходимы для предотвращения коллизий и повышения эффективности сети.»

8. Протокол Token Ring (High Speed Token Ring), принципы его работы.

Token Ring — это сетевой протокол и технология, в которой доступ к среде передачи данных осуществляется **по принципу передачи маркера**. **Топология** – кольцо. **Доступ к среде передачи данных** – маркерный.

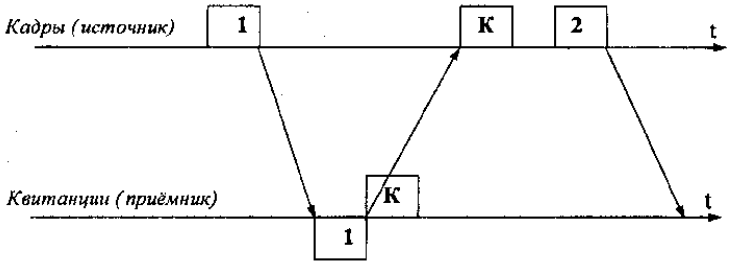
Компьютер может начать передавать данные в сеть, только если получит от предыдущего компьютера в кольце "маркер" – специальный короткий пакет, свидетельствующий о том, что сеть свободна. Если компьютеру нечего передавать в сеть, то он передает маркер следующему компьютеру в кольце. Если компьютеру есть что передавать, то он уничтожает маркер и передает свой пакет в сеть. Пакет по битам ретранслируется по кольцу от компьютера к компьютеру, адресат получает пакет, устанавливает в пакете биты, подтверждающие, что пакет достиг адресата и передает пакет дальше по кольцу. Наконец, пакет возвращается к отправителю, который уничтожает его и передает в сеть новый маркер. Компьютер может и не передавать в сеть новый маркер, а продолжить передавать кадры данных до тех пор, пока не истечет время удержания маркера. После истечения времени удержания маркера компьютер обязан прекратить передачу собственных данных и передать маркер далее по кольцу. Обычно время удержания маркера по умолчанию равно 10 мс.



Маркерный метод управления обменом (СМ—свободный маркер, ЗМ— занятый маркер, МП—занятый маркер с подтверждением, ПД—кадр данных)

9. Алгоритм скользящего окна. Основные параметры и характеристики.

1) Метод с простоями требует, чтобы источник, пославший кадр, ожидал получения квитанции (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр (или квитанция) считается утерянным и его передача повторяется.



При использовании этого метода производительность обмена данными существенно снижается, т.к передатчик обязан ждать прихода квитанции. В некоторых случаях время ожидания квитанции может существенно превышать время посылки сообщения. Снижение производительности этого метода коррекции особенно заметно на низкоскоростных каналах связи.

2) Второй метод - метод "**скользящего окна**".

Скользящее окно – это алгоритм управления передачей данных, который позволяет отправителю посылать несколько пакетов подряд, не дожидаясь подтверждения на каждый из них.

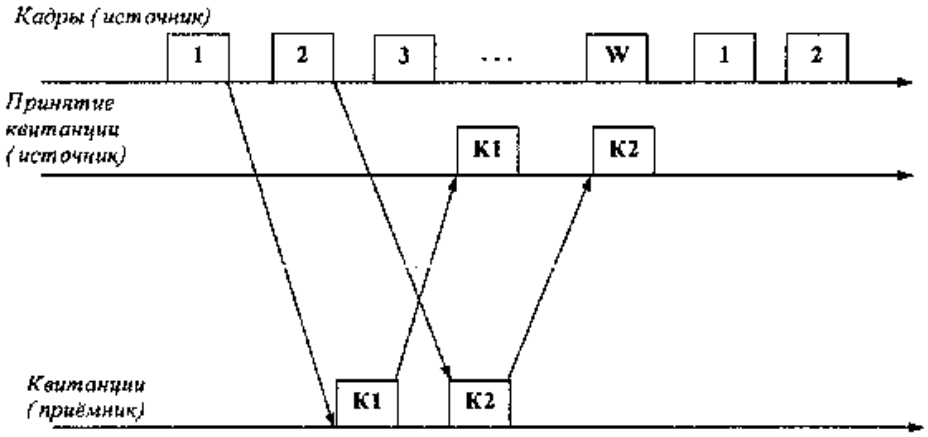
Используется в протоколе TCP на транспортном уровне (и в некоторых канальных протоколах).

Основная идея: у отправителя есть окно – это диапазон пакетов, которые можно отправить, но они ещё не подтверждены получателем. Размер окна ограничен.

ACK – это подтверждение приёма данных, показывающее, какие пакеты уже получены и какой ожидается следующим.

Основные параметры

1. **Размер окна**
   - количество кадров, которые можно отправить без подтверждения
2. **Номера последовательности**
   - каждый кадр имеет номер
   - используется для упорядочивания и повторной передачи
3. **Подтверждения (ACK)**
   - подтверждают получение кадров
4. **Таймер**
   - если подтверждение не пришло → повтор передачи



10. Охарактеризовать способ доступа к среде в сетях Ethernet. Привести основные параметры.

В сетях Ethernet используется метод доступа к среде передачи данных **CSMA/CD** – множественный доступ с прослушиванием несущей и обнаружением коллизий, используемый в сети Ethernet. Устройство проверяет, свободна ли сеть (линия). Если сеть свободна, устройство начинает отправку. Если два устройства начали передачу одновременно: коллизия обнаруживается, устройство отправляет в сеть jam-сигнал и передача прекращается. Устройства ждут случайное время и повторяют попытку.

Применяется в классических сетях эзернет (10base-5, 10base-2, 10base-t) с общей средой передачи данных в полудуплексном режиме.

В современном эзернет CSMA/CD не используется, так как в них используется коммутатор и фул-дуплекс, в которых коллизия отсутствует

Основные параметры:

1. Минимальный размер кадра - 64 байта (Для того, чтобы отправитель успел обнаружить коллизию)
2. jam-сигнал - 32 бита
3. Время двойного оборота – 512(575) битовых интервал. Максимальное время, за которое сигнал проходит до самого дальнего устройства и обратно.
4. Максимальное число попыток передач – 16. После этого кадр отбрасывается.

Время двойного оборота — это время, за которое сигнал от передатчика доходит до самого дальнего узла сети и возвращается обратно к передатчику.
Если ПК А начал передачу кадра к ПК В, а ПК В одновременно начал передачу к ПК А, происходит коллизия. В этот момент сигнал искажается, и это искажённое состояние распространяется обратно к

обоим узлам **пока они ещё передают кадры**. После обнаружения коллизии узлы посылают в сеть **jam-сигнал**, чтобы все узлы узнали о коллизии.
Если же кадр был очень короткий и передача закончилась до того, как искажённый сигнал вернулся, передатчик может ошибочно считать, что кадр успешно доставлен.

Напоминалка: Симплекс - передача только в одну сторону. Полудуплекс - передача в обе стороны, но не одновременно. Полный дуплекс - передача в обе стороны одновременно.

11. Протоколы канального уровня Ethernet (10Mbps-10 Gbps). Общность и различия. Как обеспечивается их обратная совместимость?

10BASE-T, 100BASE-TX, 1000BASE-T, 10GBASE-T

Общность всех Ethernet-протоколов
Независимо от скорости (10, 100, 1000, 10 000 Mbps):

1. Все протоколы используют **Ethernet-кадры** (заголовок + данные + CRC).
2. **MAC-адресация** для идентификации источника и получателя.
3. **CSMA/CD** используется **только в полудуплексном режиме**.
4. Передача данных может идти через **общую среду (кабель, шина)** или через **коммутатор**.

Основные отличия

1. **Скорость передачи данных (10, 100, 1000 Мбит/с).**
2. **Кодирование**
   - 10 Mbps: Manchester
   - 100 Mbps: 4B/5B + MLT-3
   - 1 Gbps: PAM-5
   - 10 Gbps: PAM-16 или другие методы
3. **Полный дуплекс и CSMA/CD**
   - На скоростях ≥1 Gbps почти всегда full-duplex → коллизий нет → CSMA/CD не используется

Ethernet обеспечивает совместимость через несколько механизмов:

1. **Одинаковая структура кадра**
   - Независимо от скорости передачи (10, 100, 1000 Mbps и выше) структура Ethernet-кадра остаётся **одинаковой**: заголовок + данные + CRC.
   - Это позволяет новым устройствам **понимать старые кадры** и наоборот.
2. **Компромис скорости и дуплекса (Auto-Negotiation)**
   - Устройства при подключении **договариваются**, на какой скорости работать и какой дуплекс использовать.
   - Пример: один узел поддерживает 10 Мбит/с, другой — 100 Мбит/с → оба работают на **минимальной скорости** 10 Мбит/с, чтобы связь была стабильной.
3. **Использование совместимых разъёмов и кабелей**
   - RJ-45 для витой пары остаётся стандартом для 10/100/1000 Mbps.
   - Старые кабели могут поддерживать меньшие скорости, новые — более высокие.
4. **Поддержка полудуплексного режима в новых стандартах**
   - Для совместимости со старыми устройствами, использующими CSMA/CD, новые устройства **ещё долго поддерживали полудуплекс**, хотя современные сети обычно full-duplex.

12. Привести технические параметры каналов связи в компьютерных сетях.

1. Полоса пропускания кабеля (частотный диапазон сигналов, пропускаемых кабелем) и затухание сигнала в кабеле. Два этих параметра тесно связаны между собой, так как с ростом частоты сигнала растет затухание сигнала. Надо выбирать кабель, который на заданной частоте сигнала имеет приемлемое затухание. Или же надо выбирать частоту сигнала, на которой затухание еще приемлемо. Затухание измеряется в децибелах и пропорционально длине кабеля.
2. Помехозащищенность кабеля и обеспечиваемая им секретность передачи информации. Эти два взаимосвязанных параметра показывают, как кабель взаимодействует с окружающей средой, то есть, как он реагирует на внешние помехи, и насколько просто прослушать информацию, передаваемую по кабелю.
3. Скорость распространения сигнала по кабелю или, обратный параметр – задержка сигнала на метр длины кабеля. Этот параметр имеет принципиальное значение при выборе длины сети. Типичные величины скорости распространения сигнала – от 0,6 до 0,8 от скорости распространения света в вакууме. Соответственно типичные величины задержек – от 4 до 5 нс/м.
4. Для электрических кабелей очень важна величина волнового сопротивления кабеля. Волновое сопротивление важно учитывать при согласовании кабеля для предотвращения отражения сигнала от концов кабеля. Волновое сопротивление зависит от формы и взаиморасположения проводников, от технологии изготовления и материала диэлектрика кабеля. Типичные значения волнового сопротивления – от 50 до 150 Ом.
5. Пропускная способность — это максимальная скорость передачи данных по заданному каналу связи.
6. Дуплексность — это способ передачи данных, при котором информация передаётся между устройствами одновременно в обоих направлениях.
- Simplex — однонаправленная передача.
- Half-duplex — попеременная двусторонняя передача.
- Full-duplex — одновременная двусторонняя передача.

Полоса пропускания - **сколько данных можно передать за единицу времени** (как труба — чем шире труба, тем больше воды (данных) можно пропустить за раз.)
Помехозащищённость - Способность канала передавать сигнал без ошибок при наличии помех (насколько канал умеет передавать данные, когда вокруг шумно (электромагнитные помехи, другие сигналы).)
Затухание - Потеря мощности сигнала на единицу длины кабеля (если кричать в длинную трубу, звук будет слабее в конце трубы.)
Сопротивление - Электрическое сопротивление проводников, влияет на затухание и качество передачи (чем выше сопротивление, тем сильнее ослабляется сигнал)
Максимальная длина канала - Максимальное расстояние, при котором сигнал ещё **достаточно сильный для корректного приёма**.

13. Что максимально влияет на пропускную способность каналов связи, приведите формулы с пояснениями.

Пропускная способность канала — это сколько данных можно передать за единицу времени.
Факторы влияющие на пропускную способность:

1. Ширина полосы канала - чем шире диапазон частот, тем больше данных можно передать одновременно.
2. Соотношение сигнал/шум - чем чище сигнал относительно помех, тем больше бит можно передавать без ошибок.
3. Кодирование и модуляция - от способа кодирования зависит, сколько битов можно «зашить» в один сигнал.
4. Ошибки передачи - если ошибок много, нужно повторять отправку кадров → эффективная скорость падает.

14. Охарактеризовать алгоритм CSMA/CD. Перечислить его основные параметры.

Алгоритм **CSMA/CD** – множественный доступ с прослушиванием несущей и обнаружением коллизий. Устройство проверяет, свободна ли сеть (линия). Если сеть свободна, устройство начинает отправку. Если два устройства начали передачу одновременно: коллизия обнаруживается, устройство отправляет в сеть jam-сигнал и передача прекращается. Устройства ждут случайное время и повторяют попытку.

Применяется в классических сетях эзернет (10base-5, 10base-2, 10base-t) с общей средой передачи данных в полудуплексном режиме.

В современном эзернет CSMA/CD не используется, так как в них используется коммутатор и фул-дуплекс, в которых коллизия отсутствует

Основные параметры:

1. Минимальный размер кадра - 64 байта (Для того, чтобы отправитель успел обнаружить коллизию)
2. jam-сигнал - 32 бита
3. Время двойного оборота – 512(575 со служебными) битовых интервал. Максимальное время, за которое сигнал проходит до самого дальнего устройства и обратно.
4. Максимальное число попыток передач – 16. После этого кадр отбрасывается.

Время двойного оборота — это время, за которое сигнал от передатчика доходит до самого дальнего узла сети и возвращается обратно к передатчику.
Если ПК А начал передачу кадра к ПК В, а ПК В одновременно начал передачу к ПК А, происходит коллизия. В этот момент сигнал искажается, и это искажённое состояние распространяется обратно к обоим узлам **пока они ещё передают кадры**. После обнаружения коллизии узлы посылают в сеть **jam-сигнал**, чтобы все узлы узнали о коллизии.
Если же кадр был очень короткий и передача закончилась до того, как искажённый сигнал вернулся, передатчик может ошибочно считать, что кадр успешно доставлен.

Напоминалка: Симплекс - передача только в одну сторону. Полудуплекс - передача в обе стороны, но не одновременно. Полный дуплекс - передача в обе стороны одновременно.

15. Какими средствами была достигнута преемственность стандартов Ethernet?

Основные средства достижения преемственности

5. **Одинаковая структура кадра**
   - Независимо от скорости передачи (10, 100, 1000 Mbps и выше) структура Ethernet-кадра остаётся **одинаковой**: заголовок + данные + CRC.
   - Это позволяет новым устройствам **понимать старые кадры** и наоборот.
6. **Компромис скорости и дуплекса (Auto-Negotiation)**
   - Устройства при подключении **договариваются**, на какой скорости работать и какой дуплекс использовать.
   - Пример: один узел поддерживает 10 Мбит/с, другой — 100 Мбит/с → оба работают на **минимальной скорости** 10 Мбит/с, чтобы связь была стабильной.
7. **Использование совместимых разъёмов и кабелей**
   - RJ-45 для витой пары остаётся стандартом для 10/100/1000 Mbps.
   - Старые кабели могут поддерживать меньшие скорости, новые — более высокие.
8. **Поддержка полудуплексного режима в новых стандартах**

- Для совместимости со старыми устройствами, использующими CSMA/CD, новые устройства **ещё долго поддерживали полудуплекс**, хотя современные сети обычно full-duplex.

16. Алгоритм CSMA/CD и его разновидности. Охарактеризовать, назвать сферы применения.

Алгоритм **CSMA/CD** – множественный доступ с контролем несущей и обнаружением коллизий. Устройство проверяет, свободна ли сеть (линия). Если сеть свободна, устройство начинает отправку. Если два устройства начали передачу одновременно: коллизия обнаруживается, устройство отправляет в сеть jam-сигнал и передача прекращается. Устройства ждут случайное время и повторяют попытку.

Разновидности:

1. **1-persistent CSMA\cd (постоянный).** Устройство постоянно прослушивает сеть и начинает передачу сразу, как только сеть становится свободной.
2. **Non-persistent CSMA\cd (непостоянный).** Если сеть занята, устройство ждёт случайное кол-во времени, а затем снова проверяет сеть.
3. **p-persistent CSMA\cd (вероятностный).** Если сеть свободна, устройство передаёт данные с вероятностью p, а с вероятностью (1−p) ждёт следующий временной интервал.

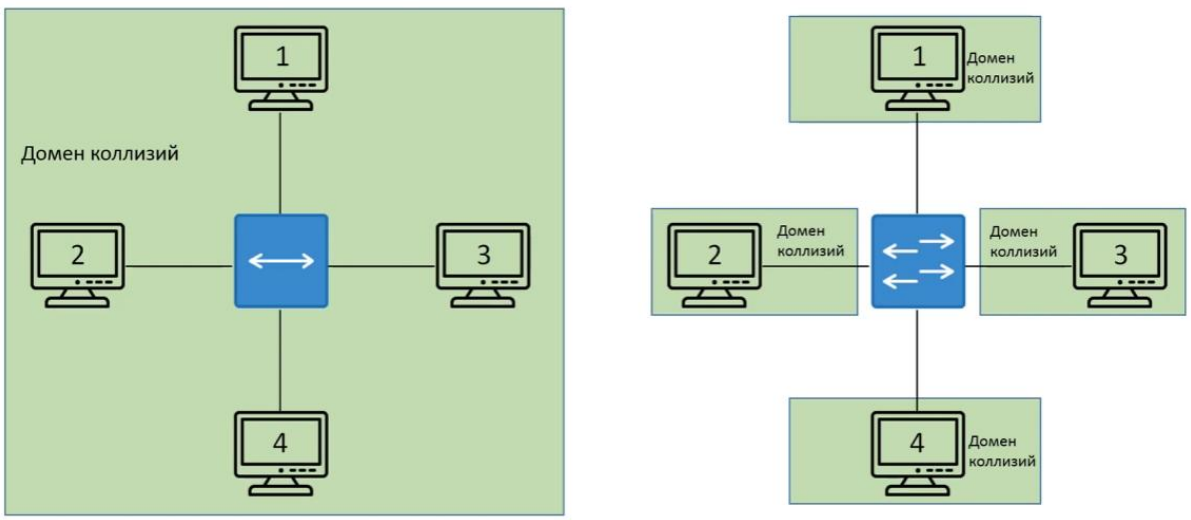
Сферы применения: Применяется в классических сетях эзернет (10base-5, 10base-2, 10base-t) с общей средой передачи данных в полудуплексном режиме.

В современном эзернет CSMA/CD не используется, так как в них используется коммутатор и фул-дуплекс, в которых коллизия отсутствует

Напоминалка: Симплекс - передача только в одну сторону. Полудуплекс - передача в обе стороны, но не одновременно. Полный дуплекс - передача в обе стороны одновременно.

17. Дать определение домена коллизий. Как он различается на разных устройствах и почему?

Домен коллизий — это часть сети, в которой могут возникнуть коллизии при одновременной передаче данных несколькими устройствами.



Различия:
Шина и хаб: один домен коллизий. ПК1 отправляет кадр ПК4, а ПК3 в это же время отправляет кадр ПК1. Происходит **коллизия**, так как используется одна общая среда.
Свитч полудуплекс: у каждого порта свой домен коллизий. Коллизия возможна только между устройствами которые работают через 1 порт. ПК1 ↔ ПК2 работают через один порт.
Свитч фулдуплекс: отсутствует домен коллизий

18. Классификация методов доступа к среде передачи данных, их характеристики, достоинства и недостатки.

Существует множество алгоритмов доступа. Их выбор зависит от скорости передачи в сети, длины шины, загруженности сети, используемого кода передачи.

Случайные методы доступа:

Алгоритм CSMA/CD– множественный доступ с контролем несущей и обнаружением коллизий, используемый в сети Ethernet. Его главное достоинство в том, что все абоненты полностью равноправны, и ни один из них не может надолго заблокировать обмен другому (как в случае наличия приоритетов). В этом методе коллизии не предотвращаются, а разрешаются.

Применяется: класический эзерент (10/100), шина, хабы
Плюсы: простота, не требует центрального управления
Минусы: коллизии, падение при высокой загрузке, не подходит под фул-дуплекс

Алгоритм CSMA/CA – множественный доступ с контролем несущей и избежанием коллизий, применяющийся, например, в сети Apple LocalTalk. Абонент, желающий передавать и обнаруживший освобождение сети, передает сначала короткий кадр запрос на передачу. Затем он заданное время ждет кадр подтверждение запроса от абонента-приемника. Если ответа нет, передача откладывается. Если ответ получен, передается кадр. Коллизии полностью не устраняются, но в основном сталкиваются управляющие кадры. Столкновения информационных кадров выявляются на более высоких уровнях протокола.

Применяется: вифи
Плюсы: подходит для беспроводной сети, минимизирует коллизии
Минусы: больше накладных расходов, ниже скорость по сравнению с эзернет

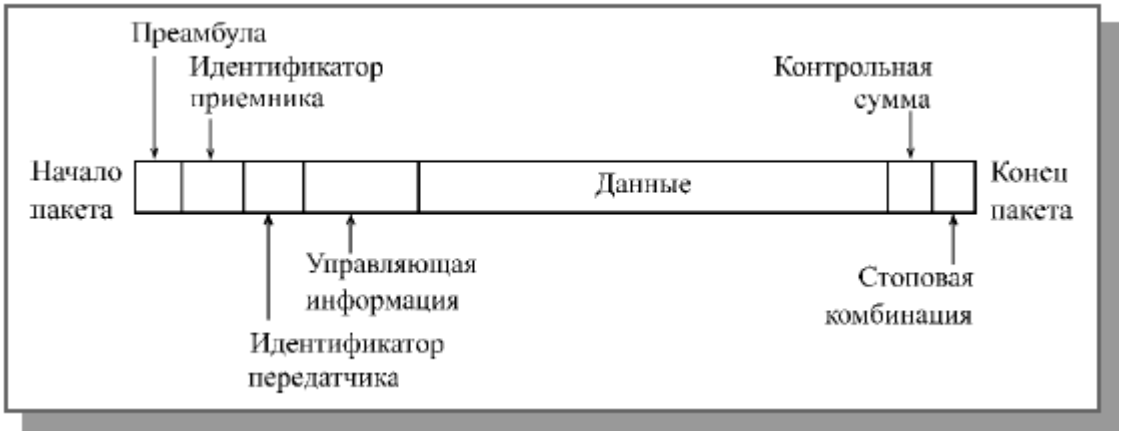
Детерминированные методы доступа (передача по строгим правилам)

Маркерный доступ (токен ринг): Компьютер может начать передавать данные в сеть, только если получит от предыдущего компьютера в кольце "маркер" – специальный короткий пакет, свидетельствующий о том, что сеть свободна. Если компьютеру нечего передавать в сеть, то он передает маркер следующему компьютеру в кольце. Если компьютеру есть что передавать, то он уничтожает маркер и передает свой пакет в сеть. Пакет по битам ретранслируется по кольцу от компьютера к компьютеру, адресат получает пакет, устанавливает в пакете биты, подтверждающие, что пакет достиг адресата и передает пакет дальше по кольцу. Наконец, пакет возвращается к отправителю, который уничтожает его и передает в сеть новый маркер. Компьютер может и не передавать в сеть новый маркер, а продолжить передавать кадры данных до тех пор, пока не истечет время удержания маркера. После истечения времени удержания маркера компьютер обязан прекратить передачу собственных данных и передать маркер далее по кольцу. Обычно время удержания маркера по умолчанию равно 10 мс.

Применяется: токен ринг, fddi
Плюсы: нет коллизий, предсказуемое время доступа
Минусы: сложность реализации, отказ одного узла может повлиять на сеть

«Алгоритмы доступа к среде определяют порядок передачи данных по общей среде и необходимы для предотвращения коллизий и повышения эффективности сети.»

19. Нарисовать структуру кадра Ethernet, дать краткое описание полей кадра.



7 байт	1 байт	6 байт	6 байт	2 байта	46 – 1500, 1504 или 1982 байта		4 байта	
Preamble	**SFP**	**Destination Address**	**Source Address**	**Length/Type**	**Data**	**PAD**	**FCS**	**Extension**
		64-2000 байта						

Стартовая комбинация (Преамбула)

Информирует сетевую карту получателя о том, что сейчас начнется передача данных. Это технический сигнал синхронизации.

Адрес получателя (Destination Address)

Может быть адресован:

- Одному конкретному устройству (личное письмо).
- Группе устройств (например, «всем жильцам подъезда»).
- Всем устройствам в сети (как объявление по радио на весь дом).

Адрес отправителя (Source Address)

Передается для того, чтобы получатель знал, от кого пришла посылка, и смог, если нужно, отправить ответ. Почта (свитч) тоже смотрит на этот адрес, чтобы запомнить, откуда кто шлёт посылки.

Служебная информация (Length/Type)

Объясняет, **что внутри** (IP-пакет, ARP-запрос и т.д.) и **как это обрабатывать**. Благодаря этому поле данных потом передадут нужной программе (браузеру, игре).

Данные (Data)

Полезная информация, ради которой всё и затевалось. Может быть разного размера.

Контрольная сумма (CRC/FCS)

Получатель сверяет, цела ли посылка после дороги. Он сам пересчитывает, что должно быть внутри, и сравнивает с вашей описью. Если цифры не сходятся — значит, данные повредились в пути, и такую посылку **выбрасывают** (и, если нужно, запрашивают заново).

Стоповая комбинация

Информирует получателя о том, что передача данных закончена. Завершает передачу.

20. Стандартные сетевые протоколы (в пределах изученных стеков протоколов). Перечислить, поставить в соответствие модели OSI.

Протоколы – это набор правил и процедур, по которым устройства обмениваются данными. Компьютеры, участвующие в обмене, должны работать по одним и тем же протоколам, чтобы в результате передачи вся информация восстанавливалась в первоначальном виде.

	Уровень (layer)	Тип данных	Функции уровня	Особенность адресации	Примеры протоколов
Уровни хоста (узла)	7. Прикладной (application)	Данные (строки из байтов)	Доступ к сетевым службам	URL	HTTP(S), FTP(S), RPC, POP3
	6. Представительский (представления) (presentation)		Представление (кодировка) и шифрование данных		ASCII, EBCDIC
	5. Сеансовый (session)		Управление сеансом связи		PAP
	4. Транспортный (transport)	Сегменты (segment) / Дейтаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	Порт	TCP, UDP, SCTP, PORTS
Уровни связи (сети)	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IP-адрес	IPv4, IPv6, IPsec, AppleTalk
	2. Канальный (data link)	Биты (bit) / Кадры (frame)	Физическая адресация	MAC-адрес (физический адрес компьютера)	PPP, IEEE 802.22, Ethernet, DSL, ARP, L2TP, сетевая карта.
	1. Физический (physical)	Электрические сигналы, Биты (bit)	Работа со средой передачи, сигналами и двоичными данными		USB, кабель ("витая пара", коаксиальный, оптоволоконный), радиоканал

Протоколы передачи данных (прикладной/транспортный уровень):
HTTP(S) — протокол передачи веб-страниц (HTTP — без шифрования, HTTPS — с шифрованием).
FTP(S) — передача файлов между компьютерами (FTP — без шифрования, FTPS — с шифрованием).
DNS – протокол преобразования имени в ip-адрес
POP3 — получение электронной почты с сервера.

Представительский (Кодировки символов)
ASCII — стандартная кодировка символов для английского алфавита.
EBCDIC — кодировка, использовавшаяся в мейнфреймах IBM.

Сеансовый (Протоколы аутентификации)
PAP — простой протокол аутентификации (пароль передаётся открыто).

Транспортные протоколы
TCP — надёжный протокол с установкой соединения и контролем доставки.
UDP — быстрый протокол без установки соединения, без гарантий доставки.
SCTP — протокол с поддержкой мультиплексирования и повышенной надёжностью.

Сетевые протоколы
IPv4 — основная версия IP-адресации (32-битные адреса).
IPv6 — новая версия IP-адресации (128-битные адреса).
IPsec — набор протоколов для защиты IP-соединений (шифрование, аутентификация).
AppleTalk — устаревший стек протоколов для сетей Apple.

Канальный и физический уровень
PPP — протокол для прямых соединений (например, модемных).
IEEE 802.22 — стандарт беспроводной связи для широкополосного доступа в сельской местности.
Ethernet — самый распространённый стандарт проводных локальных сетей.
DSL — технология высокоскоростного интернета по телефонным линиям.
ARP — протокол определения MAC-адреса по IP-адресу в локальной сети.
L2TP — протокол туннелирования для VPN.

Физические среды передачи
USB — интерфейс для подключения периферии, иногда используется для сети.
Витая пара — кабель из скрученных пар проводов (используется в Ethernet).
Коаксиальный кабель — кабель с центральным проводником и экраном (используется в ТВ, сетях).
Оптоволоконный кабель — передача данных световыми импульсами (высокая скорость, дальность).
Радиоканал — беспроводная передача данных через радиоволны.

21. Протоколы прикладного уровня. Перечислить, назвать основные функции.

Протоколы прикладного уровня описывают взаимодействие клиентской и серверной частью программы.

1. HTTP - протокол передачи гипертекста, работает на 80 порту. Используется в WWW для передачи гипертекстовых HTML страниц. При работе по этому протоколу, каждый элемент HTML – страницы загружается отдельно, причем соединение между загрузками прерывается и никакой информации о соединении не сохраняется. Это сделано для того, чтобы пользователя Web-страниц каждый получал "по чуть-чуть, в порядке общей очереди". В противном случае могла бы создаться ситуация, когда один человек качает страницу с большим количеством рисунков высокого разрешения, а все остальные ждут пока он это закончит.

HTTP — это правила, по которым браузер и сервер «разговаривают».
браузер говорит: *«дай мне страницу»*
сервер отвечает: *«вот HTML, картинки, стили»*

2. FTP – протокол передачи файлов, работает на 20 и 21 порту. Предназначен для копирование файлов между компьютерами. Полностью занимает канал, пока не будет получен файл, сохраняет информацию о соединении. При сбое возможна докачка с того места, где произошел сбой.

FTP нужен, чтобы **загружать и скачивать файлы** между компьютерами.
Ты закидываешь файлы сайта на хостинг — это FTP.

3. DNS – протокол преобразования имени в ip-адрес.

4. IMAP-4, POP3 – почтовые протоколы (электронная почта). IMAP-4 – 143 порт, POP3 – 110 порт. **Отличие**: POP3 и IMAP-4 – протоколы взаимодействия пользователя со своим почтовым ящиком на сервере.

POP3 — забрать письма с почтового сервера на компьютер.
- письма **скачиваются** и часто **удаляются с сервера**
- удобно для одного устройства

📌 **Пример:**
Как сходить на почту, забрать письма и унести их домой.

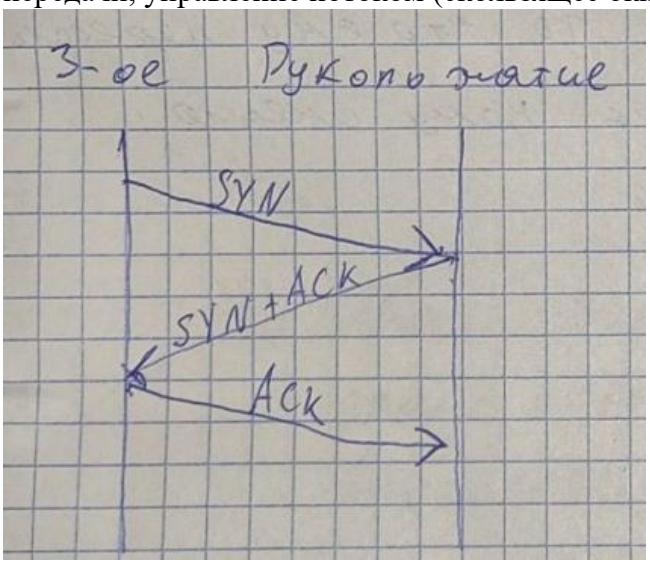
(Для сравнения: IMAP — письма остаются на сервере.)

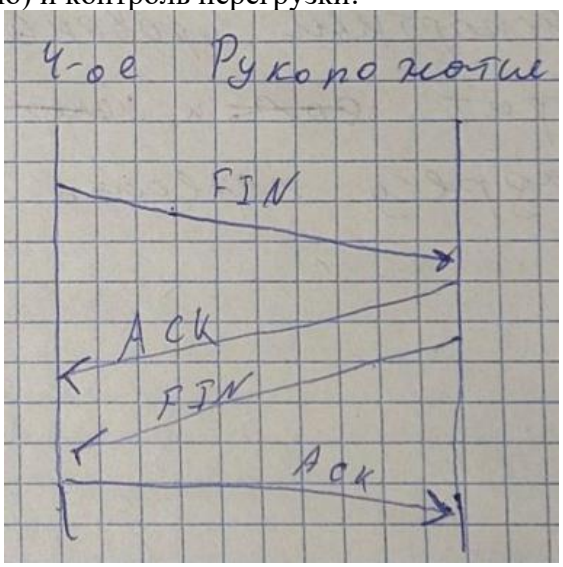
22. Транспортные протоколы. Перечислить, назвать основные функции

Транспортный уровень отвечает за доставку данных между процессами, надежность, порядок, контроль потерь. Работает между приложениями, а не между пк.

Протоколы транспортного уровня представлены двумя протоколами: TCP и UDP.

TCP – надежный, требующий соединения протокол. Гарантирует безопасную доставку данных без ошибок в правильном порядке, без потерь и дубликатов. Для установки соединения использует 3-ное рукопожатие, подтверждение приема (квитанцию), тайм-ауты и повторные передачи, управление потоком (скользящее окно) и контроль перегрузки.





1. Компьютер А посылает компьютеру В пакет, с установленным флагом SYN (синхронизация) и случайным числом (a) => SYN (a).
2. Компьютер В отвечает пакетом, с установленными флагами ACK (подтверждение), с параметром (a+1), и установленным флагом SYN и своим случайным числом (b). <= ACK(a+1), SYN (b)
3. Компьютер А завершает "рукопожатие" пакетом, с флагами ACK(a+1), ACK (b+1).
=> ACK (a+1), ACK (b+1)

UDP - это ненадёжный, не требующий установления соединения транспортный протокол. Не гарантирует доставку, порядок следования или защиту от дублирования пакетов.

Порты. Транспортный уровень вводит порты источника и отправителя, что позволяет нескольким программам на одном пк одновременно работать с сетью.

23. Протоколы сетевого уровня. Перечислить, назвать основные функции.

Протоколы сетевого уровня решают самый главный вопрос, как доставить пакет из одной сети в другую.

IP – протокол обеспечивающий доставку ip-пакета от отправителя к получателя между сетями.

ICMP – это вспомогательный протокол сетевого уровня, используемый для отправки служебных и диагностических сообщений об ошибках и исключительных ситуациях в IP-сети.

IPsec – протокол защиты ip-трафика, шифрование и аутентификация.

Rip - Протокол дистанционно-векторной маршрутизации, при котором роутеры каждые 30 секунд обмениваются своими таблицами маршрутизации с соседями. Каждый роутер запоминает лишь через кого и сколько хопов (шагов) идти до сети-назначения, но не знает полной топологии сети. хопов (максимум 15).

Ospf - Протокол маршрутизации состояния каналов, при котором роутеры обмениваются информацией о состоянии своих каналов. Каждый роутер строит полную топологическую карту всей

области. Рассчитывает оптимальные пути до всех сетей с помощью алгоритма Дейкстры. Обмен происходит только при изменениях в топологии.

Eigrp - это протокол маршрутизации, разработанный Cisco, который использует алгоритм DUAL. Роутеры знают полную топологию сети (хранят топологическую таблицу) и имеют предварительно рассчитанные резервные пути (feasible successors). Обмен происходит только при изменениях.

BGP - протокол динамической маршрутизации, который Обеспечивает маршрутизацию между разными автономными системами (AS), т.е. между разными интернет-провайдерами или огромными компаниями. Рассылает не метрику, а путь (AS_PATH) — список AS, через которые нужно пройти, чтобы достичь сети. Выбор пути основан на политиках и правилах, а не только на "скорости".

24. Способы получения адресов узлами. Назвать специальные IP адреса.

1. **Статическая адресация (вручную).** IP-адрес, маска, шлюз задаются вручную администратором.
2. **Динамическая адресация (DHCP)**. Узел автоматически получает IP от DHCP-сервера.
3. **Автоматическая адресация (APIPA).** Если DHCP недоступен, узел сам выбирает ip

Подсеть	Назначение
0.0.0.0/8	Адрес источника пакета
127.0.0.0/8	Loopback Используется для тестирования сетевого ПО (см. localhost)
169.254.0.0/16	Канальные (link-local) адреса
192.0.2.0/24	Для примеров и документации
198.51.100.0/24	Для примеров и документации
203.0.113.0/24	Для примеров и документации
198.18.0.0/15	Для стендов тестирования производительности
10.0.0.0/8	Для использования в частных сетях
172.16.0.0/12	Для использования в частных сетях
192.168.0.0/16	Для использования в частных сетях
224.0.0.0 – 239.255.255.255	Multicast. Передача группе узлов, а не всем
240.0.0.0/4	Зарезервировано для использования в будущем
255.255.255.255	Широковещательный адрес. Broadcast. Пакет получают все узлы сети
Подсеть 224.0.0.0/4	зарезервирована для многоадресной рассылки

25. Стек протоколов TCP/IP. Структура, достоинства, недостатки.

Стек TCP/IP – набор протоколов контроля передачи данных между сетями.

TCP/IP — **практическая реализация сети**, в отличие от модели OSI, которая более теоретическая.

Структура стека TCP/IP состоит из 4-ех уровней:

1. **Канальный уровень** – обеспечивает передачу данных внутри локальной сети (arp,ppp)

2. **Сетевой уровень** – отвечает за адресацию и маршрутизацию пакетов.

Протоколы:
- IP – логическая адресация и доставка пакетов без гарантии доставки.
- ICMP – передача служебных сообщений об ошибках и диагностика.

Функции:
- логическая адресация (IP-адреса);
- маршрутизация пакетов между сетями;
- выбор маршрута доставки.

3. Транспортный уровень – обеспечивает передачу данных между приложениями.
Протоколы:
- TCP – надёжная передача с установлением соединения, контролем ошибок и потерь.
- UDP – быстрая передача без установления соединения и гарантий доставки.

4. Прикладной уровень – обеспечивает взаимодействие между приложениями.

Достоинства TCP/IP
Стандартизован и повсеместно используется
Независим от типа сети. Ethernet, Wi-Fi, оптоволокно — TCP/IP работает везде
Масштабируемость. Можно объединять миллионы узлов
Гибкость протоколов. Приложения могут использовать TCP или UDP по выбору
- масштабируемость (подходит для сетей любого размера);
- независимость от типа сети и оборудования;
- маршрутизация в сетях с произвольной топологией;
- поддержка динамической маршрутизации (RIP, OSPF, EIGRP);
- высокая надёжность при использовании TCP.

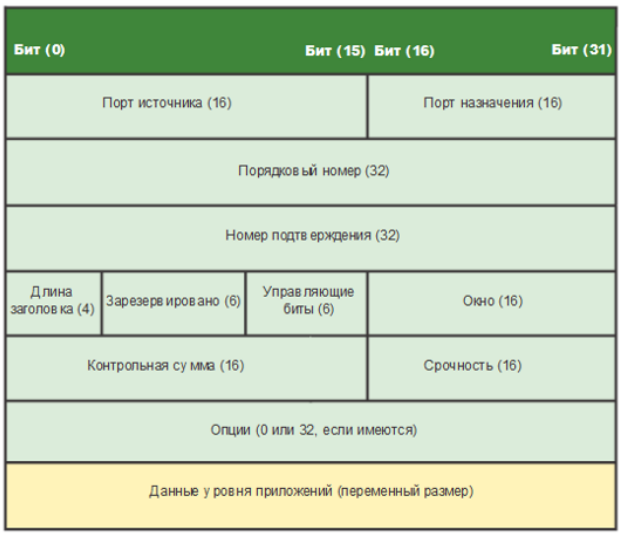
Недостатки TCP/IP
Сложность настройки вручную. Статическая IP-адресация
Безопасность не встроена. Нужны SSL/TLS, IPsec
Большой заголовок у некоторых протоколов (TCP/IP overhead)
Отсутствие гарантии доставки на уровне IP. TCP исправляет это, но UDP — нет
- отсутствие гарантии доставки на уровне IP;
- избыточность служебной информации в TCP;
- сложность администрирования в крупных сетях;
- изначально не предусматривалась защита данных.

26. Протокол TCP. Назначение, основные поля заголовка, предназначение полей, алгоритм работы.

Службы и протоколы верхнего уровня, которые используют механизмы TCP:
- FTP (File Transfer Protocol - протокол передачи файлов);
- HTTP (Hypertext Transfer Protocol - протокол передачи гипертекста);
- SMTP (Simple Mail Transfer Protocol - простой протокол электронной почты);
- DNS (Domain Name System - служба доменных имен).

TCP – это надёжный, ориентированный на соединение протокол. Он гарантирует доставку данных без ошибок, в правильном порядке, без потерь и дубликатов. Для этого использует: установление соединения (3-ное рукопожатие), подтверждение приёма (квитанция), тайм-ауты и повторные передачи, управление потоком (скользящее окно) и контроль перегрузки

TCP сегмент



Порт отправителя
Порт получателя
Порядковый номер – номер, используемый для расположения поступающих данных в правильной последовательности.
Номер подтверждения номер следующего ожидаемого TCP сегмента.
Управляющие биты - служебные функции (например, установка и завершение сеанса).
Окно - количество октетов, с которым отправитель готов согласиться.
Контрольная сумма - расчетная контрольная сумма заголовка и полей данных.
Данные - данные протокола более высокого уровня.

Структура заголовка

Бит	0 — 3	4 — 6	7 — 15	16 — 31
0	Порт источника, **Source Port**			Порт назначения, **Destination Port**
32	Порядковый номер, **Sequence Number (SN)**			
64	Номер подтверждения, **Acknowledgment Number (ACK SN)**			
96	Длина заголовка, (**Data offset**)	Зарезервировано	Флаги	Размер Окна, **Window size**
128	Контрольная сумма, **Checksum**			Указатель важности, **Urgent Point**
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

В заголовке TCP содержатся следующие поля:
•**Source port** (16 бит): порт источника. Порт хоста, от которого исходит запрос.
•**Destination port (**16 бит): порт назначения. Порт хоста, куда направляется запрос.

1. **Порты (как квартиры в доме)**
   - **Порт источника (16 бит) — от какой "программы" на вашем ПК (например, браузер — случайный порт 54321)**
   - **Порт назначения (16 бит) — какой "программе" на сервере (например, веб-серверу всегда порт 80)**
2. **Порядковый номер (32 бита)**
   - **Номер первого байта в этом сегменте. Как номер страницы в книге.**
   - **Пример: если в сегменте 1000 байт, а номер 5001, значит здесь байты с 5001 по 6000.**
3. **Номер подтверждения (32 бита)**
   - **"Я успешно получил все байты ДО этого номера, жду следующий".**
   - **Пример: если подтверждение = 6001, значит все байты до 6000 получены.**
4. **Флаги (управляющие биты) — самые важные!**
   - **SYN — "Привет, давай познакомимся?" (начало соединения)**
   - **ACK — "Я тебя услышал" (подтверждение)**
   - **FIN — "Пока, закончил" (завершение)**
   - **RST — "Всё, срочное отключение!" (аварийный разрыв)**
5. **Размер окна (16 бит)**
   - **"Я могу принять вот столько байт, не больше". Защита от переполнения.**

- **Как сказать: "Присылай не больше 10 страниц за раз, пока я не подтвердил получение".**

6. **Контрольная сумма (16 бит)**
   - **Проверка, не повредились ли данные при передаче.**
   - **Как контрольный код на упаковке.**

27. Мультиплексирование и демультиплексирование на транспортном уровне, порты и сокеты.

Сокет – комбинация ip + порт. Структура (ip-адрес получателя и отправителя, порт получателя и отправителя, протокол tcp\udp). **Сокет** - идентификатор адреса соединения

Мультиплексирование – процесс объеденения независимых потоков данных в единый интерфейс(сегмент, датаграмму) (инкапсуляция)

Несколько приложений на нашем устройстве отправляют независимые друг от друга данные. протокол tcp/udp "сворачивает" их в сегмент\датаграмму дописывая свой заголовок и передает на сеетвой уровень, на котором этот сегмент\датаграмма "оборачивается" в ip-пакет (приложение само решает какой протокол использовать)

Демультиплексирование – процесс распределения независимых потоков данных к соответствующему приложению (декапсуляция)

Порт – идентификатор приложения (некая точка входа)

28. Протокол IP. Назначение, основные поля заголовка, предназначение полей, алгоритм работы, принципы маршрутизации.

IP - протокол обеспечивающий доставку ip-пакета от отправителя к получателя между сетями.

Предназначение полей:
- Версия заголовка – указывает версию протокола (IPv4, IPv6);
- Длина заголовка (Hlen) – определяет размер заголовка IP;
- Идентификатор – используется для сборки фрагментированных пакетов у получателя;
- Флаги – управляют процессом фрагментации пакета;
- Указатель фрагмента – указывает положение фрагмента в исходном пакете;
- Время жизни – ограничивает время существования пакета в сети;

• Протокол – указывает, какому протоколу верхнего уровня передаются данные (TCP, UDP, ICMP и др.)
• Контрольная сумма заголовка –проверка целостности заголовка IP;
• IP-адрес отправителя – адрес узла отправителя;
• IP-адрес получателя – адрес узла получателя;
• Данные – данные протокола более высокого уровня.

Алгоритм работы:
1. Прикладной протокол формирует сегмент данных и передаёт его протоколу IP.
2. IP добавляет к данным свой заголовок, формируя IP-пакет, в котором указываются:
• IP-адрес отправителя
• IP-адрес получателя
• тип протокола верхнего уровня
• время жизни
3. Если размер пакета превышает допустимый для канального уровня, IP выполняет фрагментацию – разбивает пакет на части.
4. IP определяет, находится ли получатель в той же сети:
• если да, пакет передаётся напрямую;
• если нет, пакет отправляется на маршрутизатор.
5. Каждый маршрутизатор по пути:
• уменьшает время жизни,
• проверяет таблицу маршрутизации,
• выбирает следующий узел передачи.
6. Если время жизни становится равным нулю – пакет уничтожается и отправляется сообщение ICMP.
7. У получателя фрагменты собираются в исходный пакет и передаются протоколу верхнего уровня.

Принципы маршрутизации:
1. Иерархическая адресация – IP-адрес состоит из сетевой части и адреса узла, что позволяет строить масштабируемые сети.
2. Таблицы маршрутизации – каждый маршрутизатор хранит таблицу маршрутов, в которой указано, куда передавать пакеты для каждой сети.
3. Выбор наилучшего маршрута – маршрут выбирается по метрике: количеству переходов, пропускной способности, задержке и другим параметрам.
4. Динамическое обновление маршрутов – таблицы могут обновляться автоматически с помощью протоколов маршрутизации (RIP, OSPF, EIGRP).
5. Децентрализованная маршрутизация – каждый маршрутизатор принимает решение самостоятельно, на основе своей таблицы маршрутов.

29. Правила IP-адресации.

Правила IP-адресации:
1. Уникальность адреса
Каждый узел в одной сети должен иметь уникальный IP-адрес.
2. Двухуровневая структура адреса
IP-адрес состоит из:
• сетевой части – идентификатор сети;
• узловой части – идентификатор узла.
3. Использование маски подсети
Маска подсети определяет, какая часть IP-адреса относится к сети, а какая – к узлу.
4. Недопустимые адреса для узлов
• адрес сети;
• широковещательный адрес.

5. Адреса должны принадлежать одной подсети
Для прямого обмена данными узлы должны находиться в одной подсети.
Для обмена между разными сетями используется маршрутизатор.
6. Использование частных и публичных адресов
- частные адреса используются внутри локальных сетей;
- публичные адреса используются в сети Internet.

7. Статическая и динамическая адресация
- статическая – адрес задаётся вручную;
- динамическая – адрес выдаётся автоматически.

30. Структура и особенности IPv6. Какие функции были в нем оптимизированы?

В середине 80-х годов Internet впервые столкнулся с проблемой переполнения таблиц магистральных маршрутизаторов. Решение, однако, было быстро найдено -- подсети устранили проблему на несколько лет. Но уже в начале 90-х к проблеме большого количества маршрутов прибавилась нехватка адресного пространства. Ограничение в 4 миллиарда адресов, заложенное в протокол и казавшееся недосягаемой величиной, стало весьма ощутимым.

В качестве решения проблемы были одновременно предложены два подхода -- один на ближайшее будущее, другой комплексный и долгосрочный. Первое решение -- это внедрение протокола бесклассовой маршрутизации (CIDR), к которому позже присоединилась система NAT.

Долгосрочное решение - это протокол IP следующей версии. Он обозначается, как IPv6, или IPng (Internet Protocol next generation). В этой реализации протокола длина адреса увеличена до 16-ти байтов (128 бит!), исключены некоторые элементы действующего протокола, которые оказались неиспользуемыми.

Преимущества протокола IPv6 перед IPv4

- Большое адресное пространство – длина адресного поля IPv6 **128 бит**, что существенно расширяет адресное пространство.
- Усовершенствованный формат заголовка – IPv6 использует новый формат заголовка, в котором поле "Опции", когда оно необходимо, отделено от основного заголовка. Это упрощает и ускоряет процесс маршрутизации, т.к. большинство из опций роутером не обрабатываются.
- Новые опции – IPv6 имеет новые опции для расширения функциональности.
- Возможности расширения и модификации протокола – IPv6 имеет широкие возможности по расширению и модификации протокола с сохранением преемственности при необходимости поддержки новых приложений и услуг.
- Эффективная поддержка функций распределения ресурсов – в IPv6 удалено поле «Тип сервиса» (Type of Service), но добавлен механизм, который обеспечивает источнику возможность запроса различных способов обработки пакета. Этот механизм может использоваться для поддержки разнородного трафика, в т.ч. аудио и видео трафика реального режима времени.
- Эффективная поддержка функций безопасности – функции шифрования и аутентификации в IPv6 обеспечивают конфиденциальность целостность пакетов.

Общий формат дейтаграммы IPv6

- Каждый пакет состоит из обязательного основного заголовка, за которым следует поле полезной нагрузки.
- Поле полезной нагрузки, в свою очередь, состоит из двух частей
  - Заголовки расширения(опционально);
  - Данные, полученные с верхнего уровня;
- Длина основного заголовка фиксированная и занимает **40** байт
- Размер поля полезной нагрузки (дополнительные заголовки + данные) составляет до 65 535 байт.

Формат пакета IPv6

- Самые важные изменения, отличающие IPv6, очевидны именно в формате дейтаграмм.
- Расширенные возможности адресации. В протоколе IPv6 размер IP адреса увеличивается с 32 до 128 бит. Таким образом гарантируется, что IP-адреса в мире не закончатся никогда. Теперь можно присвоить IP-адрес каждой песчинке на нашей планете.

- В IPv6 появился новый тип адреса, именуемый адресом свободной рассылки, который позволяет доставить дейтаграмму любому хосту из указанной группы. Например, эту возможность удобно применять для отправки HTTP-запроса GET ближайшему сайту-зеркалу из нескольких имеющихся, на котором содержится искомый документ.
- Оптимизированный 40-байтный заголовок. Как будет указано ниже, ряд полей IPv4 были упразднены или сделаны необязательными. В результате получились заголовки, имеющие фиксированную длину в 40 байт и обеспечивающие более быструю обработку IP-дейтаграмм. Новая кодировка возможностей обеспечивает их ускоренную обработку.

Метка потока и приоритет. В протоколе IPv6 существует довольно туманное определение потока. В стандартах RFC 1752 и RFC 2460 записано, что данная сущность обеспечивает «пометку пакетов, относящихся к конкретным потокам, для которых отправитель затребует специальной обработки — например, обслуживания в локальном времени или качества обслуживания, отличающегося от заданного по умолчанию».

Например, передача аудио или видео может интерпретироваться как поток. С другой стороны, при решении более традиционных задач — например, при передаче файлов и электронной почты — сообщаемая информация может и не обрабатываться как потоки. Возможна ситуация, в которой пользовательский трафик с высоким приоритетом также будет обрабатываться как поток. Очевидно, что создатели IPv6 предвидели: рано или поздно потребуется механизм различения потоков, пусть даже точное определение самого термина пока отсутствует. Кроме того, в заголовке IPv6 есть 8-битное поле *класса трафика*. Это поле, как и поле TOS (тип сервиса обслуживания) в IPv4, может применяться для присвоения приоритета определенным дейтаграммам в потоке либо для предпочтения дейтаграмм, исходящих от одних приложений (допустим, от протокола ICMP) дейтаграммам от других приложений (таких, как сетевые новости).

- **Версия:** поле, содержащее 4-битное двоичное значение, которое определяет версию IP-пакета. Для пакетов IPv6 в этом поле всегда указано значение 0110.
- **Класс трафика**: 8-битное поле, соответствующее полю «Дифференцированные сервисы (DS)» в заголовке IPv4.

Оно также содержит 6-битное значение точки кода дифференцированных сервисов (DSCP), которое используется для классификации пакетов, а также 2-битное значение явного уведомления о перегрузке (ECN), используемое для управления перегрузками трафика.

- **Метка потока:** 20-битное поле, предоставляющее специальную службу для приложений реального времени. Используя это поле, маршрутизаторам и коммутаторам передается информация о необходимости поддерживать один и тот же путь для потока пакетов, что поможет избежать их переупорядочивания.
- **Длина полезной нагрузки:** 16-битное поле, соответствующее полю «Общая длина» в заголовке IPv4. Оно определяет размер всего пакета (фрагмента), включая заголовок и дополнительные расширения.
- **Следующий заголовок:** 8-битное поле, соответствующее полю «Протокол» в заголовке IPv4. Оно указывает тип полезной нагрузки данных, которые переносит пакет, что позволяет сетевому уровню пересылать данные на соответствующий протокол более высокого уровня. Это поле также используется в тех случаях, когда в пакет IPv6 добавляются дополнительные заголовки расширений.
- **Предел перехода:** 8-битное поле, заменяющее поле «Время существования» (TTL) в IPv4. Это значение уменьшается на единицу каждым маршрутизатором, пересылающим пакет. Когда счетчик достигает 0, пакет отбрасывается, и на отправляющий узел пересылается сообщение ICMPv6, которое означает, что пакет не достиг своего назначения.

75

Формат пакета IPv6

- **Фрагментация/пересборка**: протокол IPv6 не допускает фрагментацию и пересборку дейтаграмм на промежуточных маршрутизаторах; эти операции могут выполняться лишь на хостах — отправителе и получателе. Если дейтаграмма IPv6, полученная роутером, слишком велика, и ее не удается передать по исходящему соединению, то роутер просто отбрасывает эту дейтаграмму и отсылает отправителю по протоколу ICMP сообщение «Packet Too Big» («Пакет слишком велик») (см. ниже). После этого отправитель может переслать данные, составив более компактную IP-дейтаграмму. На фрагментацию и пересборку дейтаграмм тратится немало времени; при удалении этой функции с промежуточных роутеров и локализации только в начальной и конечной системе удается значительно ускорить передачу инф-ии в сети по прот. IP.
- **Контрольная сумма заголовка**. Поскольку протоколы транспортного уровня (TCP и UDP) и канального уровня (например, стандарт Ethernet) в Интернете выполняют проверку контрольных сумм, разработчики IP, вероятно сочли, что на сетевом уровне эта функциональность является избыточной, и ее можно оттуда удалить. Основное внимание уделялось ускорению обработки IP-пакетов. Как вы помните из обсуждения протокола IPv4, поскольку в заголовке IPv4 есть поле TTL, подобное полю лимита переходов в IPv6, контрольная сумма заголовка IPv4 должна пересчитываться на каждом маршрутизаторе. Как и фрагментация/ пересборка, такой пересчет был признан слишком затратной операцией.

76

31. Протоколы канального уровня. Перечислить, произвести сравнительную характеристику.

1. Ethernet
   Где: Проводные сети (компьютер-роутер, компьютер-свитч)
   Скорость: 10 Мбит → 100 Мбит → 1 Гбит → 10 Гбит → ...
   Как работает: CSMA/CD (слушай канал, если свободно — говори)
   Адресация: MAC-адреса (00:11:22:33:44:55)

2. Arp
   Что делает: Узнает MAC-адрес по IP-адресу
   Пример: "У кого IP 192.168.1.1? Отзовись!" → "У меня! Мой MAC: 00:11:22:33:44:55"

3. PPP
   Где: Модемные подключения, VPN
   Особенность: Только 2 устройства напрямую
   Плюсы: Аутентификация, сжатие, контроль ошибок

Характеристика	Ethernet	ARP	PPP
Что это?	Физическая среда + протокол	Служебный протокол	Протокол связи
Где работает?	Локальные сети (LAN)	Внутри LAN	Точка-точка (WAN)
Адресация	MAC-адреса (00:11:22:33:44:55)	IP → MAC	Нет адресов (всего 2 устройства)
Тип передачи	Много участников (broadcast/multicast/unicast)	Broadcast запрос, unicast ответ	Только между двумя
Пример из жизни	Общая комната, где все слышат друг друга	Кричать: "Вася, ты где?"	Разговор по прямому телефону
Для чего нужен?	Передача данных между многими устройствами	Найти MAC-адрес по IP	Надежное соединение двух устройств

Ethernet, Token Ring, FDDI, ARP

32. Протоколы ARP, RARP. Назначение. Алгоритм работы.

ARP - Работает на стыке канального и сетевого уровней. Его задача — найти MAC-адрес устройства, зная его IP-адрес в локальной сети. Без ARP пакеты не смогли бы уйти с компьютера

1. **Компьютер хочет отправить пакет** на IP-адрес в своей локальной сети
2. **Проверяет свою ARP-таблицу** — есть ли там MAC для этого IP?
3. **Если нет записи:** отправляет **ARP-запрос (broadcast)** на всю сеть: "Кто имеет IP [адрес назначения]? Отзовись!"
4. **Все устройства в сети** получают запрос, но отвечает только тот, у кого есть этот IP
5. **Устройство-получатель** отправляет **ARP-ответ (unicast)**: "У меня! Мой MAC-адрес: [его MAC]"
6. **Отправитель записывает** пару IP-MAC в свою ARP-таблицу
7. **Теперь отправляет пакет** на правильный MAC-адрес

RARP – ищет ip-адрес по мак адресу. Устарел. Вытеснен DHCP

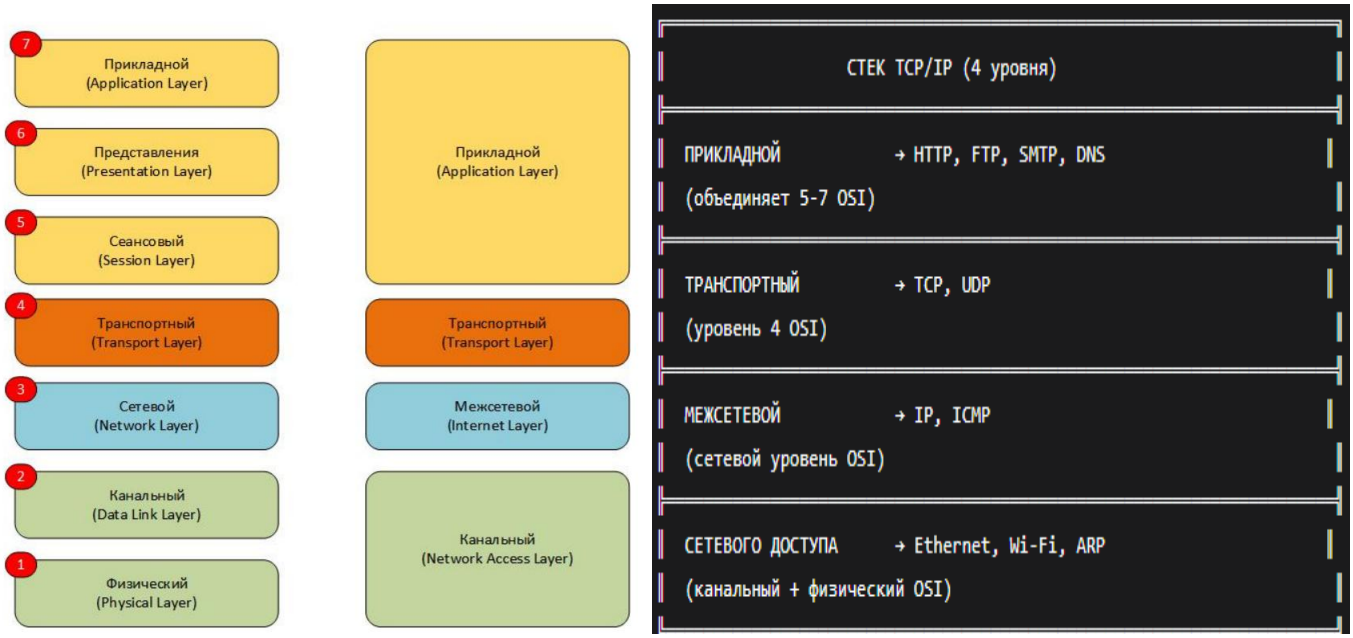
1. **Компьютер без IP-адреса** включается в сеть
2. **Отправляет RARP-запрос (broadcast):** "У меня MAC [его MAC], какой у меня должен быть IP?"

3. **RARP-сервер** (специальный сервер со таблицей MAC-IP) получает запрос
4. **Сервер ищет в своей таблице** IP для этого MAC
5. **Отправляет RARP-ответ:** "MAC [его MAC] должен иметь IP [назначенный IP]"
6. **Компьютер получает IP-адрес** и начинает его использовать

Проблема: RARP давал только IP и маску, но не шлюз, DNS и т.д. **Сейчас заменен на DHCP.**

Аспект	ARP (Address Resolution Protocol)	RARP (Reverse Address Resolution Protocol)
Назначение	Найти **MAC-адрес** по известному **IP-адресу.**	Найти **IP-адрес** по известному **MAC-адресу** (устаревшая функция).
Основной вопрос	"У кого IP-адрес `192.168.1.5` ? Скажи свой MAC!"	"У меня MAC `00:1A:2B:3C:4D:5E` . Какой мне взять IP-адрес?"
Современная замена	Нет, **ARP незаменим** и используется повсеместно.	**Полностью заменён** на **DHCP** и **BOOTP.**
Актуальность	**Критически важен** для работы любой IP-сети.	**Исторический протокол,** не используется в современных сетях.

33. Нарисовать модель соответствия стека TCP/IP модели OSI. Зачем нужно разделение на уровни?



Разделение на уровни нужно для нескольких целей:

- Структурирование сложности. Разделение позволяет инженерам сосредоточиться на конкретных задачах без необходимости понимать всю систему целиком.
- Стандартизация. Модели обеспечивают общий язык и концептуальную основу для разработки сетевых протоколов и оборудования.

- Модульность. Изменения в одном уровне минимально влияют на другие уровни, что упрощает эволюцию технологий.
- Диагностика. Иерархическая структура облегчает поиск и устранение неисправностей, позволяя изолировать проблемы на конкретном уровне.
- Обучение. Модель помогает не запутаться в технологиях и чётко понять, где заканчивается одна задача и начинается другая.
- Максимизация гибкости. У каждого уровня может быть несколько различных реализаций, при этом остальные уровни не будут об этом знать, и всё будет работать, как было задумано.

Прикладной уровень TCP/IP объединяет функции трёх верхних уровней OSI (Прикладного, Представления и Сеансового). Например, протокол HTTP (из стека TCP/IP) выполняет функции всех трёх: задаёт формат данных (Представления), управляет сессией (Сеансовый) и является интерфейсом для браузера (Прикладной).

Транспортный и Сетевой уровни в обеих моделях совпадают один-в-один. Это ядро коммуникации.

Канальный уровень TCP/IP объединяет Канальный и Физический уровни OSI, так как на практике конкретная сетевая технология (например, Ethernet) определяет и формат кадра, и физические параметры.

34. Охарактеризовать функции сетевого и канального уровней OSI. Привести примеры протоколов, на них работающих.

Канальный уровень обеспечивает передачу данных в пределах одного сегмента сети. На этом уровне биты формируются в кадры и выполняется проверка целостности с помощью CRC. В случае обнаружения ошибки кадр отбрасывается. Передача осуществляется на основе MAC-адресов.

1. Ethernet
   Где: Проводные сети (компьютер-роутер, компьютер-свитч)
   Скорость: 10 Мбит → 100 Мбит → 1 Гбит → 10 Гбит → ...
   Как работает: CSMA/CD (слушай канал, если свободно — говори)
   Адресация: MAC-адреса (00:11:22:33:44:55)

2. Arp
   Что делает: Узнает MAC-адрес по IP-адресу
   Пример: "У кого IP 192.168.1.1? Отзовись!" → "У меня! Мой MAC: 00:11:22:33:44:55"

3. PPP
   Где: Модемные подключения, VPN
   Особенность: Только 2 устройства напрямую
   Плюсы: Аутентификация, сжатие, контроль ошибок

Сетевой уровень обеспечивает передачу ip-пакета внутри одного сегмента сети или между сетями. На сетевом уровне поступивший кадр с канального уровня освобождается от MAC-заголовка, извлекается IP-пакет. Происходит проверка IP-заголовка (TTL, контрольная сумма). Если TTL = 0 или контрольная сумма не совпадает, пакет отбрасывается. Если пакет дошел невредимым, происходит проверка IP-адреса назначения: Если IP-адрес в локальной сети → пакет отправляется напрямую (через ARP узнается MAC); Если IP-адрес в другой сети → пакет отправляется на маршрутизатор (роутер). Затем сетевой уровень определяет маршрут через таблицу маршрутизации, уменьшает TTL на 1, пересчитывает контрольную сумму и передает пакет обратно на канальный уровень для отправки следующему роутеру.

Процесс повторяется на каждом роутере, пока пакет не достигнет сети назначения, где будет доставлен конечному устройству.

IP – протокол обеспечивающий доставку ip-пакета от отправителя к получателя между сетями.

ICMP – это вспомогательный протокол сетевого уровня, используемый для отправки служебных и диагностических сообщений об ошибках и исключительных ситуациях в IP-сети.

IPsec – протокол защиты ip-трафика, шифрование и аутентификация.

RIP – это дистанционно-векторный протокол динамической маршрутизации, при котором маршрутизаторы периодически (раз в 30 секунд) обмениваются своими таблицами маршрутов. Каждый роутер знает только расстояние (хопы) и направление (соседа, через которого идти) до сети. Не знает полной топологии.

OSPF – это протокол динамической маршрутизации состояния канала, в котором маршрутизаторы обмениваются информацией о состоянии своих соединений, строят полную карту топологии сети и с помощью алгоритма Дейкстры выбирают кратчайший путь на основе стоимости канала, обеспечивая быструю сходимость и масштабируемость за счёт иерархического деления сети на области.

EIGRP – это протокол маршрутизации, разработанный Cisco, который использует алгоритм DUAL для быстрого расчёта маршрутов, обменивается только изменениями маршрутов с соседями и выбирает оптимальный путь на основе совокупных характеристик канала, таких как пропускная способность и задержка, обеспечивая высокую скорость сходимости и эффективность передачи данных.

BGP - протокол динамической маршрутизации, который Обеспечивает маршрутизацию между разными автономными системами (AS), т.е. между разными интернет-провайдерами или огромными компаниями. Рассылает не метрику, а путь (AS_PATH) — список AS, через которые нужно пройти, чтобы достичь сети. Выбор пути основан на политиках и правилах, а не только на "скорости".

35. Модель OSI. Назначение, характеристика уровней. Понятия инкапсуляции и декапсуляции.

Модель OSI — это эталонная модель, которая показывает, как должны работать сетевые протоколы. Она делит сетевое взаимодействие на 7 уровней, чтобы:

1. **Упростить разработку** — каждый уровень делает свою часть работы
2. **Обеспечить совместимость** — оборудование разных производителей работает вместе
3. **Облегчить поиск проблем** — если что-то не работает, проверяем по уровням

Только КОНЕЧНЫЕ УСТРОЙСТВА (компьютеры, серверы, телефоны) работают на ВСЕХ 7 уровнях OSI! Промежуточные устройства (роутеры, свитчи) работают только на СВОИХ уровнях!

Данные	**Прикладной** (доступ к сетевым службам)	Осуществляет взаимодействие между пользователем и сетью. Взаимодействует с приложениями на стороне клиента.	HTTP, FTP, Telnet, SSH, SNMP
Данные	**Представления** (представление и кодирование данных)	Осуществляет преобразование данных в нужную форму, шифрование/кодирование, сжатие.	MIME, SSL
Данные	**Сеансовый** (управление сеансом связи)	Управляет созданием/поддержанием/завершением сеанса связи.	L2TP, RTCP
Блоки	**Транспортный** (безопасное и надежное соединение точка-точка)	Предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы. Выполняет сквозной контроль передачи данных от отправителя до получателя.	TCP, UDP
Пакеты	**Сетевой** определение пути и IP (логическая адресация)	Его основными задачами являются маршрутизация – определение оптимального пути передачи данных, логическая адресация узлов. Кроме того, на этот уровень могут быть возложены задачи по поиску неполадок в сети (протокол ICMP). Сетевой уровень работает с пакетами.	IP, ICMP, IGMP, BGP, OSPF
Кадры	**Канальный** MAC и LLC (физическая адресация)	Отвечает за доступ к среде передачи, исправление ошибок, надежную передачу данных. *На приеме* полученные с физического уровня данные упаковываются в кадры после чего проверяется их целостность. Если ошибок нет, то данные передаются на сетевой уровень. Если ошибки есть, то кадр отбрасывается и формируется запрос на повторную передачу. Канальный уровень подразделяется на два подуровня: **MAC (Media Access Control)** и **LLC (Locical Link Control)**. MAC регулирует доступ к разделяемой физической среде. LLC обеспечивает обслуживание сетевого уровня. На канальном уровне работают коммутаторы.	IEEE 802.3, IEEE 802.11, PPP, DHCP, ARP
Биты	**Физический** (кабель, сигналы, бинарная передача данных)	Определяет вид среды передачи данных, физические и электрические характеристики интерфейсов, вид сигнала. Этот уровень имеет дело с битами информации.	IEEE 802.11, ISDN

Декапсуляция (получаем данные)

1. Физический уровень: Получает световые, электрические или радиосигналы и преобразует их в биты (0 или 1).

2. Канальный уровень: Преобразует полученные биты в кадр. Просчитывает контрольную сумму кадра (CRC) и сверяет с контрольной суммой в заголовке.

> **Если суммы не совпадают** → кадр отбрасывается (без повторного запроса!)
> **Если сумма совпала** → проверяет MAC-адрес назначения: Если это наш MAC → передает кадр на сетевой уровень. Если не наш → отбрасывает

3. Сетевой уровень: Снимает заголовок кадра, извлекает IP-пакет. Пересчитывает контрольную сумму заголовка IP, проверяет TTL. (собирает фрагментированные пакеты)

> **Если контрольная сумма неверна или TTL = 0** → пакет отбрасывается, отправляется ICMP-сообщение об ошибке
> **Если проверки пройдены** → проверяет IP-адрес назначения: Если это наш IP → передает пакет на транспортный уровень. Если не наш (и мы роутер) → маршрутизирует дальше. Если не наш (и мы не роутер) → отбрасывает.

4. Транспортный уровень: Снимает IP-заголовок, извлекает TCP-сегмент или UDP-датаграмму. Смотрит на порт назначения.

> **Для TCP:** Обеспечивает медленную, но упорядоченную, безопасную отправку сегмента
> **Для UDP:** Принимает без подтверждения
> **Демультиплексирование:** По номеру порта находит, какому приложению передать данные

5. Сеансовый уровень: Управляет сеансом связи (начало, поддержка, завершение диалога). В TCP/IP часто реализован внутри прикладных протоколов.

6. Уровень представления: Преобразует данные в нужный формат — декодирует, расшифровывает, распаковывает. Например, расшифровывает HTTPS или декодирует JPEG.

7. Прикладной уровень: Приложение (браузер, почтовый клиент) получает и отображает данные пользователю.

36. Назвать назначение и порядок функционирования ICMP.

ICMP – это протокол сетевого уровня, используемый для отправки служебных и диагностических сообщений об ошибках и исключительных ситуациях в IP-сети.

ICMP-сообщения автоматически отправляются в определенных случаях, например, TTL передаваемого пакета истек и он отбросился

Порядок функционирования:

1. **Обнаружение ошибки (**TTL истек, = 0)
2. **Формирование ICMP-сообщения**
   Icmp-сообщение = Тип-код-контрольная сумма-данные
3. **Инкапсуляция в ip-пакет и отправка.** (отправляется отправителю)

Типы ICMP- сообщений (сокращенно)

Тип	Назначение сообщения
0	Эхо-ответ
3	Узел назначения недостижим
5	Перенаправления маршрута
8	Эхо-запрос
9	Сообщение о маршрутизаторе
10	Запрос сообщения о маршрутизаторе
11	Истечение времени жизни пакета
12	Проблемы с параметрами
13	Запрос отметки времени
14	Ответ отметки времени

0,8,13,14 – используются при диагностике быстродействия сети.
5 - Сообщение о новом маршруте, который позволяет быстрее попасть к необходимой сети.
9 - Сообщение о маршрутизаторе. Маршрутизаторы сами периодически рассылают такие сообщения, чтобы компьютеры в сети знали, какие есть маршрутизаторы.
10 – сообщение о маршрутизаторе – компьютер может не дожидаться сообщения о маршрутизаторе и может сам запросить информацию о маршрутизаторе.
11 – отправляется, когда отбрасывается пакет, у которого истек TTL
12 – проблемы с параметрами. Маршутизатор не может отправить пакет, т.к. в заголовке ip какая-то ошибка.

37. Перечислить виды маршрутизирущих и маршрутизируемых протоколов и их основные характеристики.

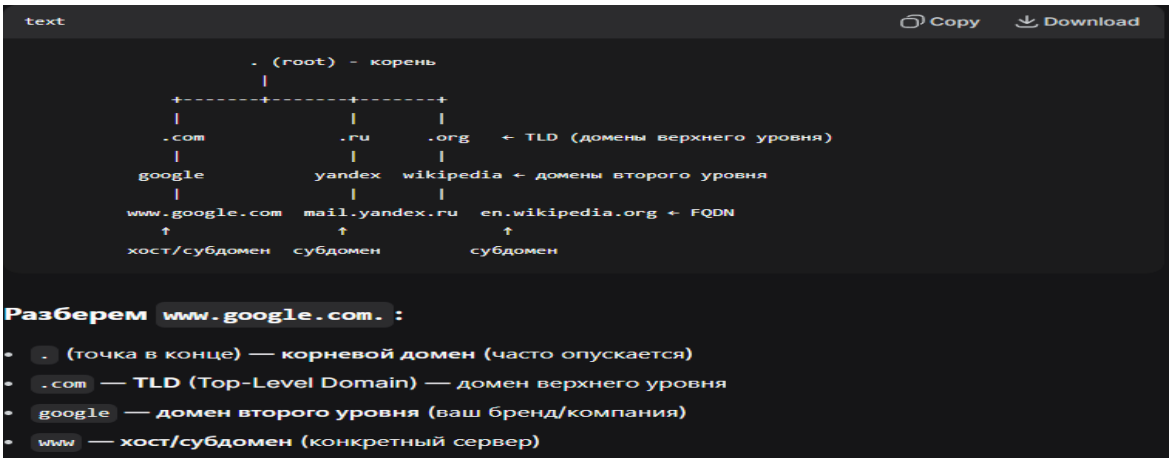
IP – маршрутезируюмый
RIP, OSPF, EIGRP, BGP – маршрутизирующие

Rip - Протокол дистанционно-векторной маршрутизации, при котором роутеры каждые 30 секунд обмениваются своими таблицами маршрутизации с соседями. Каждый роутер запоминает лишь через кого и сколько хопов (шагов) идти до сети-назначения, но не знает полной топологии сети. хопов (максимум 15).

Ospf - Протокол маршрутизации состояния каналов, при котором роутеры обмениваются информацией о состоянии своих каналов. Каждый роутер строит полную топологическую карту всей области. Рассчитывает оптимальные пути до всех сетей с помощью алгоритма Дейкстры. Обмен происходит только при изменениях в топологии.

Eigrp - Протокол Cisco, использующий усовершенствованный дистанционно-векторный алгоритм (DUAL). Роутеры знают полную топологию сети (хранят топологическую таблицу) и имеют предварительно рассчитанные резервные пути (feasible successors). Обмен происходит только при изменениях.

38. Принципы работы протокола DNS, структура доменных имен. Зачем нужны хранители ключей?



Доменное имя – это имя компьютера, вида www.sait.com. Адресация в Internet происходит по IP-адресам, однако для человека гораздо удобнее доменные имена.

Существует также термин URL-адрес (Universal Resource Locator), т.е. запись вида http://www.sait.com, или в полном варианте http://www.sait.com:80/katalog/index.html#glava1. Доменное имя являеется частью URL-адреса (схема_передачи:// доменное_имя : порт / имя файла# внутреняя_ссылка).

DNS — протокол прикладного уровня, который сопоставляет доменное имя IP-адресу**.**
DNS представляет собой распределенную базу данных (телефонную книгу интернета), в которой записано, какому доменному имени соответствует IP-адрес.
Человеку запомнить слова проще, чем комбинацию цифр, поэтому был разработан DNS, позволяющий использовать понятные имена вместо IP-адресов, которые понимают компьютеры.
Пример: При вводе www.google.com в браузере:

1. Устройство сначала проверяет **локальный кэш DNS**
2. Если не найдено → запрашивает у **DNS-сервера провайдера**
3. Если у провайдера нет ответа → начинается **иерархический запрос**:
    - Спрашивает **корневые серверы** (.)
    - Получает адреса серверов **TLD** (.com)
    - Спрашивает серверы .com
    - Получает адреса **авторитативных серверов Google**
    - Спрашивает у **серверов Google** IP для www.google.com
4. Ответ возвращается по обратной цепочке и **кэшируется** на каждом этапе.

Четыре раза в год два десятка человек встречаются, чтобы провести некую церемонию. Эти собрания не отличались от обычных офисных встреч, если бы не беспрецедентные меры безопасности, через которые проходят участники прежде, чем туда попасть: биометрические сканеры, сканеры сетчатки глаза и отпечатков пальцев — это лишь малая часть из них. Причина, по которой они собираются, тоже не самая банальная: некоторые из них являются хранителями уникального ключа от мирового интернета. А вместе их ключи образуют мастер-ключ, контролирующий одну из главных мер безопасности, лежащую в основе работы интернета, — систему доменных имен (DNS)

39. Алгоритм работы «прозрачного» моста.

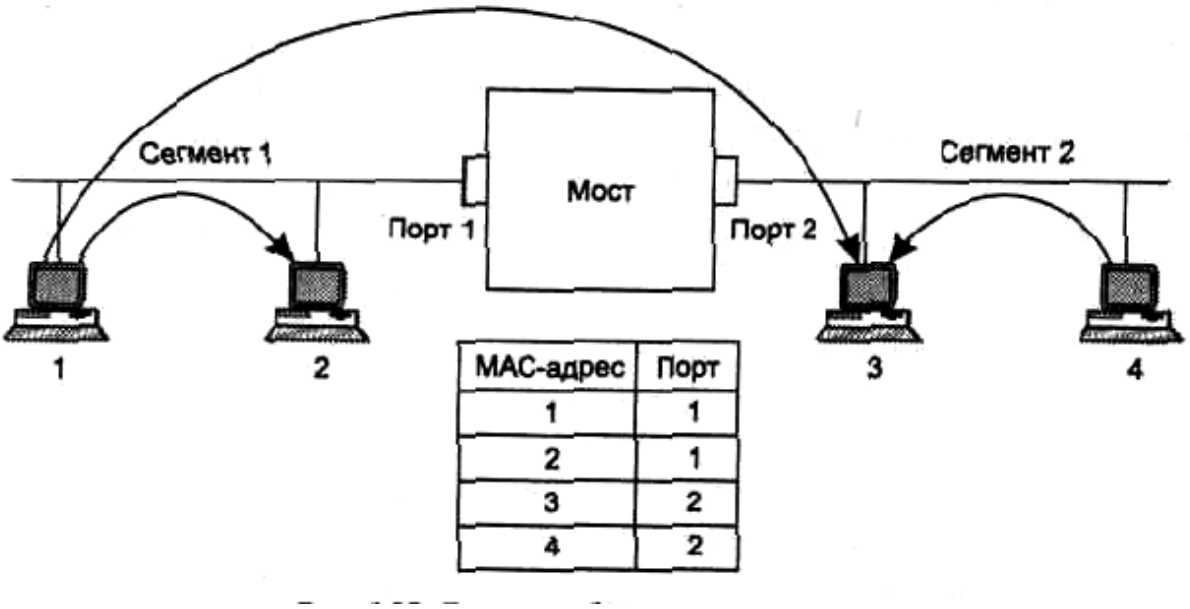
Прозрачный мост – устройство канального уровня, которое соединяет 2 или более сегмента сети между собой, позволяя передавать данные с одного устройства на другое за счет своей таблицы (mac+port).

Называется прозрачным, в связи с тем, что пк его не видят, и думают, что отправляют кадры напрямую

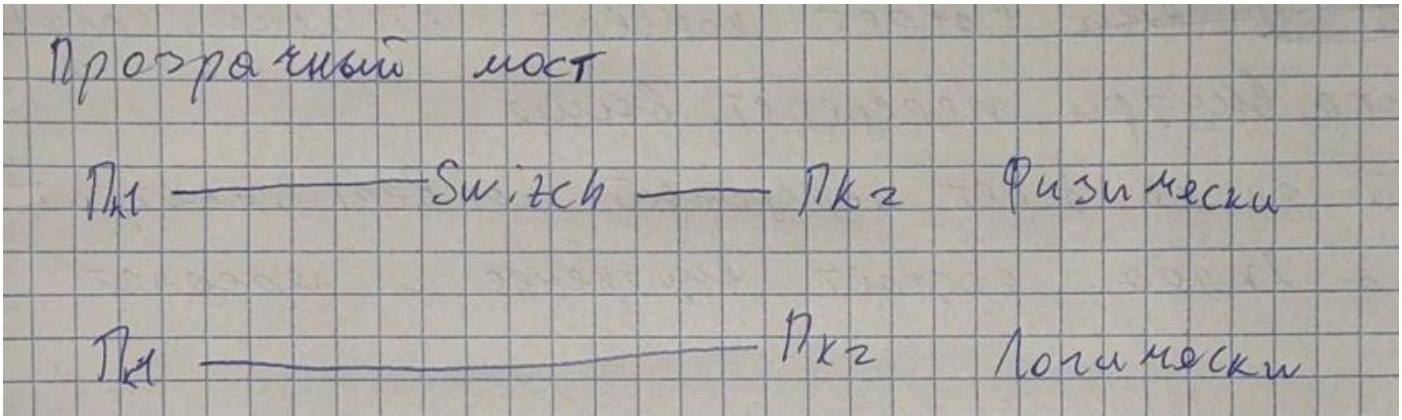
Основное отличие коммутатора от моста заключается в том, что мост обрабатывает кадры последовательно, а коммутатор - параллельно

Алгоритм работы:

1. **При получении кадра** с любого порта мост смотрит на **MAC-адрес отправителя** и запоминает, с какого порта пришел этот MAC (запись в таблицу).
2. **Затем смотрит на MAC-адрес получателя:**
   - **Если получатель есть в таблице** → отправляет фрейм **только на тот порт**, где зарегистрирован этот MAC.
   - **Если получателя нет в таблице** → отправляет фрейм на **все порты, кроме порта-источника** (флуд)
3. **Если отправитель и получатель находятся на одном порту** → мост отбрасывает кадр (не пересылает в ту же сеть).
4. **Записи в таблице стареют** (обычно через 5 минут), если от устройства не было трафика.



MAC-адрес	Порт
1	1
2	1
3	2
4	2



40. Алгоритмы, которые могут использовать различные коммутаторы в своей работе.

1. Алгоритм прозрачного моста + 2. алгоритм работы моста с маршрутизацией от источника:

Этот алгоритм используется в сетях Token Ring и FDDI. Компьютер-отправитель помещает в кадр всю адресную информацию о промежуточных мостах и кольцах, которые кадр должен пройти на пути к компьютеру-адресату. Первоначально компьютер-отправитель не имеет никакой информации о пути к компьютеру адресату. Кадр просто передается в кольцо, в надежде, что адресат находится в одном кольце с отправителем. Если компьютер-адресат в кольце отсутствует, то кадр сделает оборот по кольцу и вернется без установленного признака "кадр получен" (бит "адрес распознан" и бит "кадр скопирован"). В таком случае компьютер-отправитель пошлет одномаршрутный широковещательный кадр-исследователь (SRBF, Single Route Brodcast Frame). Этот кадр распространяется по сети: мосты дублируют кадр на все свои порты, за исключением заблокированных администратором (для избежания петлевых маршрутов и зацикливания кадра). В конце-концов кадр-иследователь будет получен компьютером-адресатом, который немедленно отправит многомаршрутный широковещательный кадр-исследователь (ARBF, All Route Brodcast Frame). Этот кадр распространяется по сети, дублируясь мостами на все порты без исключения (для предотвращения зацикливания, кадр ARBF уже однажды сдублированный мостом на один из своих портов, заново на этот порт не дублируется). В конце-концов, до компьютера-отправителя дойдет множество кадров- ARBF, прошедших через все возможные маршруты от компьютера-адресата до компьютера-исследователя. Полученная информация попадет компьютеру-отправителю и в маршрутные таблицы моста, соединяющего кольцо компьютера-отправителя с остальной сетью. Впоследствии все компьютеры этого кольца могут воспользоваться информацией моста при отправке своих кадров.

41. Структурные схемы коммутаторов, перечислить, охарактеризовать достоинства и недостатки.

1. Коммутаторы на основе коммутационной матрицы
Достоинства:
- Высокая скорость коммутации (параллельная передача между портами)
- Регулярная структура, удобная для реализации на интегральных схемах
- Низкая задержка (latency) при передаче

Недостатки:
- Сложность реализации при большом количестве портов (сложность растет пропорционально квадрату числа портов)
- Отсутствие внутренней буферизации данных: если выходной порт занят, данные должны ждать во входном буфере порта
- Сложность наращивания числа портов

Принцип работы: Временные прямые соединения между входными и выходными портами через матрицу переключателей.

2. Коммутаторы с общей шиной
Достоинства:
- Более простая архитектура по сравнению с матрицей
- Легче наращивать количество портов
- Поддержка псевдопараллельной передачи за счет разбиения кадров на ячейки
- Отсутствие начальных задержек при доступности порта

Недостатки:
- Пропускная способность ограничена скоростью шины
- Шина работает в режиме разделения времени: передача данных происходит поочередно
- Требует высокой производительности шины (равной сумме скоростей всех портов)

Принцип работы: Все порты подключены к общей высокоскоростной шине. Данные передаются в виде небольших ячеек с добавлением тега (номера порта назначения).

3. Коммутаторы с разделяемой памятью
Достоинства:
- Гибкое распределение памяти между портами
- Экономичное использование буферной памяти (общий пул)
- Поддержка приоритетных очередей и QoS
- Удобство для реализации сложных алгоритмов обслуживания очередей

Недостатки:
- Требует высокоскоростной памяти и сложного менеджера очередей
- Ограниченная масштабируемость при росте числа портов
- Риск переполнения общей памяти при перегрузке

Принцип работы: Все входные данные помещаются в общую память, где организуются очереди для каждого выходного порта. Менеджер поочередно обслуживает очереди.

4. Комбинированные коммутаторы
Достоинства:
- Сочетание преимуществ разных архитектур (например, матрицы и шины)
- Высокая производительность и масштабируемость
- Гибкость под разные сценарии нагрузки

Недостатки:
- Высокая сложность проектирования и реализации
- Высокая стоимость
- Требует сложного программного управления

Принцип работы: Например, используется шина для связи групп портов, внутри каждой группы — матричная коммутация.

42. Сетевые проблемы, решенные за счет использования коммутаторов.

1. Устранение коллизий
2. Уменьшение нагрузки на сеть
3. Одинаковая скорость на всех портах
4. Обмен данными без очереди (фул-дуплекс)

43. Ограничения коммутаторов (какие сетевые вопросы они не решают)

1. Топологические ограничения (петли)
- STP/RSTP/MSTP не позволяют использовать ВСЕ альтернативные маршруты
- Агрегирование каналов работает только между двумя соседними коммутаторами
- Многие эффективные топологии неприменимы

2. Плохая изоляция между сегментами
- Слабая защита от широковещательных штормов (broadcast storms)
- VLAN полностью изолируют сети (слишком жестко)
- Узлы разных VLAN не могут взаимодействовать

3. Сложность фильтрации трафика
- Фильтрация только через пользовательские фильтры
- Администратор работает с двоичным представлением пакетов
- Нет высокоуровневых правил фильтрации

4. Одноуровневая система адресации
- Используется только MAC-адрес (L2)

- MAC жестко привязан к сетевому адаптеру
- Недостаточная гибкость для сложных сетей

5. Ограниченная трансляция протоколов
- Не могут транслировать WAN ↔ LAN протоколы
- Причины:
    а) Разные системы адресации
    б) Разные MTU (максимальный размер пакета)
- Проблемы с гетерогенными сетями

6. Невозможность использования альтернативных маршрутов
- STP блокирует резервные пути (только один активный)
- Нельзя распределить нагрузку по нескольким путям
- Ограниченная отказоустойчивость

44. Перечислить виды маршрутизации и способы занесения записей в таблицы.

1. Статическая маршрутизация. Ручное внесение маршрутов администратором.

2. Динамическая маршрутизация. Автоматический обмен маршрутами между роутерами **(RIP, OSPF, EIGRP, BGP)**

3. Маршрутизация от источника. Путь указывается отправителем пакета**.**

4. Бестабличная маршрутизация. Не использует таблицу маршрутизации (например, лавинная маршрутизация) (флуд)

5. **Табличная маршрутизация.** Использует таблицу для принятия решений (основной метод) (rip,ospf)

6. **Гибридная маршрутизация.** Сочетает признаки дистанционно-векторных и link-state протоколов (eigrp)

7. **Внутридоменная (**внутри одной автономной системы**) и Междоменная**

45. Ключевые моменты алгоритма работы моста с маршрутизацией от источника.

1. Источник определяет маршрут
2. Использование поля RIF (в кадр добавляется поле содержащее идентификаторы промежуточных мостов)
3. Роль моста. Он не строит свою таблицу, а лишь перенаправляет по переданной «инструкции»

46. Назначение технологии NAT. Способы трансляции адресов, особенности каждого из видов. Пример.

Nat – технология, которая представляет из себя переводчик адресов.

Назначение NAT: Позволяет внутренним узлам получить доступ в глобалную сеть, параллельно решает проблему нехватки IPv4-адресов.

Nat – делиться на :

Static NAT (Статический): Где каждому локальному адресу ставится строго 1 внешний адрес

Dynamic NAT (Динамический): Есть Пул внешних адресов который nat динамически выделяет внутренним хостам на время сессии.

PAT: Для всех внутренних хостов 1 внешний адрес. Для того что бы различать хосты, к адресу приписываются номера портов. NAT позволяет представить внешнему миру внутреннюю структуру IP-адресации предприятия иначе, чем она на самом деле выглядит.

47. Перечислить дополнительные функции коммутаторов.

1. Поддержка алгоритма Spanning Tree. Автоматическое построение древовидной топологии без петель.
2. Трансляция протоколов канального уровня. Преобразование кадров Ethernet ↔ FDDI, Ethernet ↔ Token Ring.
3. Фильтрация трафика. Ограничение доступа на основе различных критериев.
4. Приоритетная обработка кадров. Пример: По порту (всем кадрам с определённого порта присваивается приоритет)
5. Виртуальные локальные сети. Логическая изоляция групп устройств внутри одной физической сети.

48. Провести сравнительную характеристику коммутаторов и маршрутизаторов (сферы использования, уровень OSI, типы адресации, алгоритмы работы).

Критерий	Коммутатор (Switch)	Маршрутизатор (Router)
Уровень модели OSI	**Канальный (L2)** — коммутаторы L2; **Сетевой (L3)** — коммутаторы L3 (многоуровневые)	**Сетевой (L3)**
Основная функция	Коммутация кадров между портами на основе MAC-адресов	Маршрутизация пакетов между сетями на основе IP-адресов
Тип адресации	**MAC-адреса** (физические, аппаратные)	**IP-адреса** (логические, сетевые)
Таблица	**Таблица MAC-адресов** (CAM-таблица) — изучается автоматически	**Таблица маршрутизации** — статическая или динамическая (RIP, OSPF, BGP)
Область применения	Локальные сети (LAN), сегментация внутри одной сети	Соединение различных сетей (LAN-WAN, Internet), межсетевое взаимодействие
Примеры протоколов	STP, VLAN (802.1Q), LACP, LLDP	RIP, OSPF, BGP, NAT, DHCP, IPSec
Скорость работы	Высокая (коммутация аппаратная)	Ниже (маршрутизация программная или аппаратно-программная)
Алгоритм работы	1. Приём кадра 2. Поиск MAC в таблице 3. Передача на нужный порт или flood	1. Приём пакета 2. Поиск маршрута в таблице 3. Изменение TTL, перезапись MAC 4. Передача

49. Виртуальные частные сети. В каких случаях появляется необходимость в создании виртуальных сегментов? Приведите примеры.

VPN — это технология, которая создает защищенное, зашифрованное логическое ("виртуальное") соединение поверх другой сети (чаще всего поверх публичной и небезопасной, такой как Интернет). Она эмулирует прямое защищенное подключение между устройствами (или между устройством и корпоративной сетью), как если бы они были соединены напрямую или находились в одной локальной сети. Основные компоненты: туннелирование данных, их шифрование и аутентификация сторон. Работает на сетевом и транспортном уровнях.

Что происходит:

1. Ваши данные (запрос к сайту, сообщение) упаковываются в зашифрованный "конверт".
2. Этот "конверт" отправляется на сервер VPN (это вход в туннель).
3. По пути в открытом интернете все видят только, что вы связались с сервером VPN, но не могут прочитать, что внутри конверта.
4. Сервер VPN распаковывает конверт и отправляет ваш запрос дальше к целевому сайту (например, в Google) от своего имени.

Зачем это нужно? Три главные причины:

1. Безопасность в чужих сетях (в кафе, аэропорту): Ваш трафик шифруется, и хакер в той же сети не сможет перехватить ваши пароли.
2. Конфиденциальность (анонимность): Для внешних сайтов и вашего провайдера запрос исходит от сервера VPN, а не от вас. Они видят IPадрес VPN-сервера, а не ваш настоящий. Это может помочь обойти географические блокировки.
3. Доступ к "закрытым" сетям: Сотрудник из дома может через VPN подключиться к внутренней сети офиса, как будто его рабочий компьютер прямо в ней находится (получить доступ к файловым серверам, базам данных).

50. Назвать наиболее слабые механизмы в работе коммутаторов и существующие методы борьбы с ними.

1. Широковещательные штормы (Broadcast Storms)
Проблема: Массовая рассылка широковещательных кадров по сети, приводящая к её полной или частичной перегрузке. Часто возникает из-за сетевых петель, неисправного оборудования или злонамеренных атак.
Методы борьбы:
- Использование VLAN для ограничения широковещательных доменов
- Настройка протоколов STP/RSTP/MSTP для предотвращения физических петель
- Включение функции Storm Control на портах коммутатора
- Применение фильтров (ACL) для блокировки избыточного широковещательного трафика
- Регулярный мониторинг сетевой активности для раннего обнаружения аномалий

2. Петли в топологии (Switching Loops)
Проблема: Наличие нескольких активных путей между коммутаторами, вызывающее бесконечную циркуляцию кадров, их дублирование и перегрузку сети.
Методы борьбы:
- Внедрение Spanning Tree Protocol (STP, IEEE 802.1D)
- Использование Rapid Spanning Tree Protocol (RSTP, IEEE 802.1w) для ускоренной конвергенции
- Настройка Multiple Spanning Tree Protocol (MSTP, IEEE 802.1s) для работы с VLAN
- Активация механизмов Loop Guard и BPDU Guard
- Применение агрегации каналов (LACP) вместо нескольких независимых соединений

3. Переполнение таблицы MAC-адресов (MAC Table Overflow)
Проблема: Злонамеренная атака (MAC-флудинг), при которой коммутатор заполняет свою CAM-таблицу поддельными MAC-адресами, после чего переходит в режим неэффективной рассылки (flooding).
Методы борьбы:
- Настройка Port Security с ограничением числа MAC-адресов на порту
- Использование статических (ручных) записей MAC-адресов для критичных устройств
- Включение Dynamic ARP Inspection (DAI)
- Активация DHCP Snooping для создания базы доверенных MAC-IP-соответствий
- Регулярный аудит и очистка таблиц MAC-адресов

4. Подмена MAC-адресов (MAC Spoofing)
Проблема: Злоумышленник изменяет MAC-адрес своего устройства на адрес легитимного узла с целью перехвата трафика или получения несанкционированного доступа.
Методы борьбы:

- Реализация 802.1X для аутентификации устройств перед доступом к сети
- Настройка Dynamic ARP Inspection (DAI) для проверки ARP-сообщений
- Использование IP Source Guard для контроля соответствия IP и MAC
- Включение Port Security в режиме "sticky" MAC
- Внедрение систем обнаружения вторжений (IDS) для мониторинга аномальной активности

5. Неэффективная обработка multicast-трафика
Проблема: Коммутатор по умолчанию обрабатывает multicast-кадры как broadcast, рассылая их на все порты, что приводит к неоправданной нагрузке на сеть и конечные устройства.
Методы борьбы:

- Активация IGMP Snooping для отслеживания запросов multicast-групп
- Настройка протокола CGMP (если используется оборудование Cisco)
- Использование PIM Snooping в маршрутизируемых multicast-сетях
- Конфигурация multicast-фильтров на портах коммутатора
- Переход на многоадресную маршрутизацию (PIM) на L3-устройствах

6. Отсутствие защиты от атак на STP
Проблема: Злоумышленник может подключить коммутатор с более высоким приоритетом и стать корневым мостом, перенаправляя весь трафик через себя для анализа или модификации.
Методы борьбы:

- Настройка Root Guard на портах, которые не должны становиться корневыми
- Включение BPDU Guard на портах, подключенных к конечным устройствам
- Использование BPDU Filter для игнорирования BPDU-пакетов на определённых портах
- Активация PortFast только на портах, подключенных к хостам (не к другим коммутаторам)
- Регулярный мониторинг топологии STP и изменений корневого моста

7. Ограниченные возможности фильтрации трафика на L2
Проблема: Коммутаторы уровня 2 не анализируют IP-заголовки, что не позволяет реализовывать сложные политики безопасности на основе IP-адресов, портов или протоколов.
Методы борьбы:

- Использование L3-коммутаторов с поддержкой расширенных ACL
- Установка отдельного маршрутизатора или файрвола для фильтрации межсетевого трафика
- Применение MAC-based ACL (где поддерживается)
- Внедрение систем глубокой инспекции пакетов (DPI) на границе сети
- Сегментация сети на VLAN с последующей маршрутизацией между ними

51. Перечислить виды конструктивного исполнения сетевых устройств, назвать достоинства и недостатки.

1. Настольное исполнение (Desktop)
Достоинства:

- Компактный размер и эргономичный дизайн (не занимает много места на столе)
- Низкая стоимость и простота установки (не требует специального монтажа)
- Низкий уровень шума (часто пассивное охлаждение или тихий вентилятор)
- Встроенный блок питания (не нужен отдельный БП)
- Подходит для малых офисов (SOHO) и домашнего использования

Недостатки:

- Ограниченное количество портов (обычно 4–8)
- Ограниченная производительность и функциональность (базовые L2-функции)

- Слабое охлаждение (риск перегрева при высокой нагрузке)
- Отсутствие резервирования (блок питания, вентиляторы)
- Низкая плотность размещения (занимает полезную площадь на столе)

Пример:
Неуправляемый коммутатор D-Link DES-1008A, TP-Link TL-SG108, Netgear GS308.

2. Стоечное исполнение (Rackmount)
Достоинства:
- Высокая плотность размещения (экономия места в телекоммуникационной стойке)
- Лучшее охлаждение (мощные вентиляторы, организованный воздушный поток)
- Масштабируемость (до 48 портов и более в одном устройстве)
- Возможность резервирования (двойные блоки питания, горячая замена вентиляторов)
- Профессиональные функции (управление, VLAN, QoS, зеркалирование портов)

Недостатки:
- Требует телекоммуникационной стойки или шкафа (дополнительные расходы)
- Высокий уровень шума (не подходит для офисных помещений)
- Высокая стоимость (по сравнению с настольными моделями)
- Сложность монтажа и обслуживания (требуется доступ к стойке)
- Большой вес и габариты

Пример:
Cisco Catalyst 2960-X, Huawei S5700-28C-EI, Juniper EX2300-24T.

3. Модульное исполнение на основе шасси (Chassis-based)
Достоинства:
- Высокая гибкость и масштабируемость (модули можно менять и добавлять)
- Высокая плотность портов (сотни портов в одном шасси)
- Надёжность (горячая замена модулей, полное резервирование компонентов)
- Централизованное управление (одно логическое устройство)
- Поддержка различных типов интерфейсов (Ethernet, оптические, WAN)

Недостатки:
- Очень высокая стоимость (шасси + модули)
- Сложность проектирования и обслуживания
- Большие габариты и вес
- Высокое энергопотребление и теплообразование
- Требует квалифицированного персонала

Пример:
Cisco Catalyst 9500-32C, Nexus 7004, Huawei CloudEngine 12800.

4. Стековое исполнение (Stackable)
Достоинства:
- Гибкость (можно начать с одного устройства и наращивать стек)
- Единое логическое устройство для управления (упрощение администрирования)
- Повышенная отказоустойчивость (при выходе одного устройства стек продолжает работу)
- Экономия по сравнению с модульными шасси
- Простота монтажа (можно устанавливать в стойку по отдельности или как блок)

Недостатки:
- Зависимость от пропускной способности стекирующих соединений
- Ограниченное количество устройств в стеке (обычно 8–10)
- Риск полного отказа стека при сбое управления
- Требует специальных стековых модулей или портов
- Ограниченная плотность портов по сравнению с шасси

Пример:
Cisco Catalyst 3750-X, HPE Aruba 2930F-24G, Huawei S6720-30C-EI-24S-AC.

5. Промышленное исполнение (Industrial)

Достоинства:
- Устойчивость к экстремальным температурам (–40°C … +75°C)
- Защита от вибрации, ударов, влаги и пыли (степень защиты IP40, IP67)
- Надёжность в тяжёлых условиях (производство, транспорт, уличное размещение)
- Широкий диапазон питания (DC 12–48V, защита от перенапряжения)
- Компактный и прочный корпус (металл, защищённые разъёмы)

Недостатки:
- Высокая стоимость
- Ограниченная функциональность (часто только базовые L2-функции)
- Меньшее количество портов и интерфейсов
- Высокий уровень шума (может не подходить для офисов)
- Сложность монтажа (специальные крепления)

Пример:
Cisco IE 2000, Moxa EDS-405A, Siemens SCALANCE XB-200.

6. Портативное/компактное исполнение (Portable / Mini)
Достоинства:
- Очень малые габариты и вес (помещается в карман или сумку)
- Низкое энергопотребление, часто питание через USB или PoE
- Низкая стоимость и простота использования (plug-and-play)
- Идеально для временных решений, командировок, полевых условий
- Бесшумная работа (часто полностью пассивное охлаждение)

Недостатки:
- Очень ограниченная производительность
- Малое количество портов (обычно 2–5 портов)
- Отсутствие или очень ограниченные возможности управления
- Нет резервирования компонентов
- Слабая защита от внешних воздействий

Пример:
Неуправляемый коммутатор TP-Link TL-SG105, портативный маршрутизатор GL.iNet GL-AR750S, Raspberry Pi с сетевым HAT-модулем.

52. Перечислить сетевые устройства, поставить их в соответствие модели OSI.

Физический: концентратор, репитер
Канальный: коммутатор, мост
Сетевой: маршрутизатор

53. Какие из изученных сетевых технологий могут быть использованы для обеспечения сетевой безопасности и каким образом?

1. ACL (Access Control Lists)
Технология: Списки управления доступом
Как используется:
- **Фильтрация трафика** по IP-адресам, протоколам и портам
- **Разграничение доступа** между отделами/сетями
- **Защита от атак** (блокировка подозрительных адресов)
- **Контроль доступа к сетевым службам** (разрешение только необходимых портов)
  Пример: access-list 101 deny tcp any any eq 23 — запрет Telnet-доступа

2. NAT (Network Address Translation)
Технология: Преобразование сетевых адресов
Как используется:

- **Скрытие внутренней структуры сети** (трансляция приватных адресов в публичные)
- **Ограничение прямого доступа** к внутренним ресурсам извне
- **Экономия публичных IP-адресов**
- **Базовая защита от сканирования сети**
  Пример: PAT (Port Address Translation) для выхода в Интернет всей локальной сети через один IP

3. VLAN (Virtual Local Area Networks)
Технология: Виртуальные локальные сети
Как используется:

- **Логическая изоляция сегментов сети** на канальном уровне
- **Разделение трафика** между отделами (финансы, производство, гости)
- **Ограничение широковещательных доменов**
- **Создание DMZ** (демилитаризованных зон) для публичных серверов
  Пример: Отдельная VLAN для IP-камер с ограниченным доступом

4. VPN (Virtual Private Network)
Технология: Виртуальные частные сети
Как используется:

- **Шифрование трафика** при передаче через ненадёжные сети (Интернет)
- **Безопасный удалённый доступ** к корпоративной сети
- **Объединение филиалов** в единую защищённую сеть
- **Анонимизация трафика** (для пользователей)
  Пример: IPSec VPN между офисами или SSL VPN для удалённых сотрудников

54. Типы организации VLAN. Дать краткую сравнительную характеристику.

VLAN - Технология, которая позволяет физическую сеть разделить на логические подсети.
Зачем это нужно?

1. **Безопасность**: Устройства из разных VLAN не могут общаться напрямую на канальном уровне. Для обмена данными им нужен маршрутизатор, где можно настроить фильтры (аксес листы).

2. **Уменьшение широковещательных доменов**: Широковещательный трафик (например, ARP) ограничивается рамками одного VLAN и не "зашумляет" всю физическую сеть.

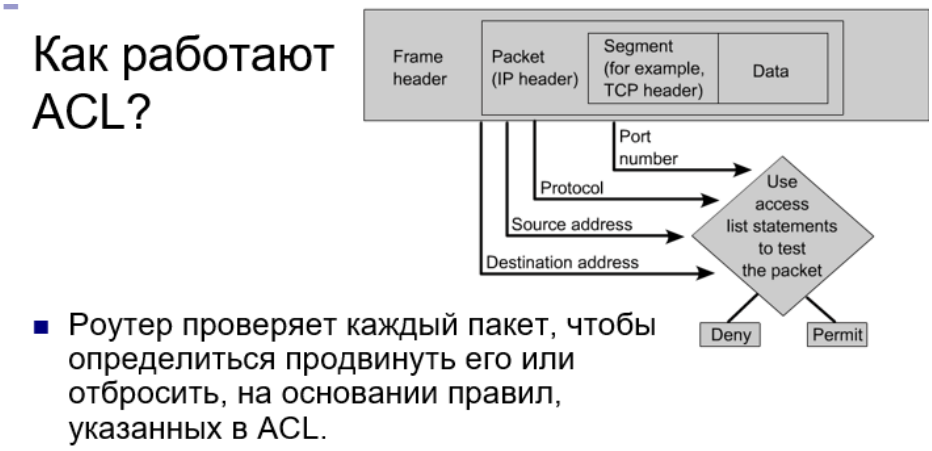
3. **Гибкость**: Устройства можно объединять в сеть по функциональному признаку (например, все IP-камеры в VLAN "Видеонаблюдение"), а не по физическому расположению. Чтобы переместить устройство в другой VLAN, достаточно изменить настройку порта коммутатора, а не перекладывать кабели.

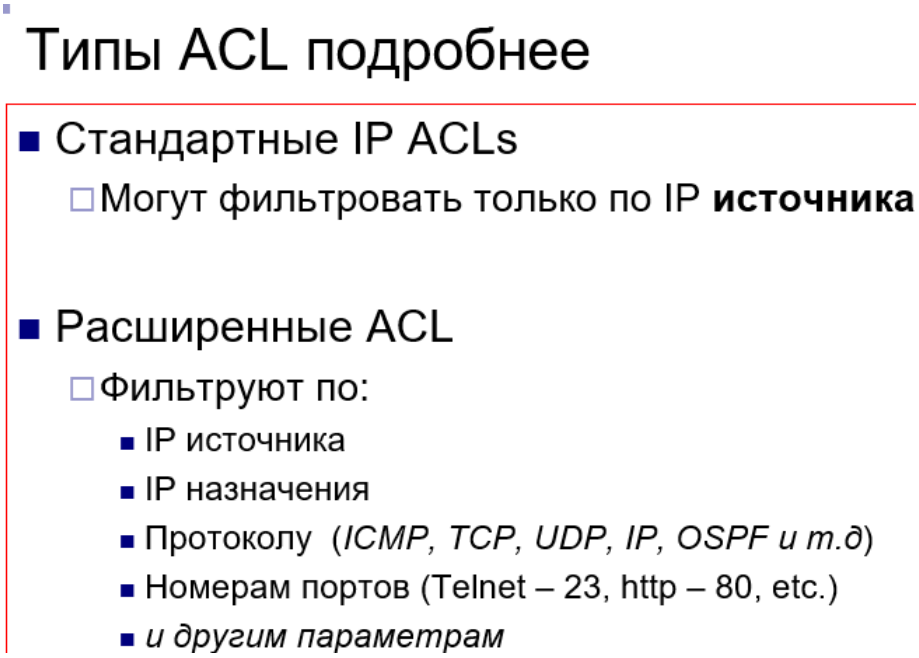
4. **Управление трафиком**

VLAN по портам (Привязка физического порта свитча к определенному VLAN)
VLAN по MAC-адресам (Автоматическое определение VLAN по MAC-адресу устройства)
VLAN по протоколам (Определение VLAN по типу протокола (IP, IPX, AppleTalk)
VLAN по IP-подсетям (Определение VLAN по IP-адресу источника)

55. Разновидности и правила формирования ACL.

ACL — это последовательный список правил, который используется для разрешения изи запрета потока пакетов внутри сети на основании информации, приведенной внутри списка.
ACL применяется к интерфейсам устройства и проверяет каждый проходящий пакет. Работает по принципу «первое совпадение».





Стандартный ACL – выполняет фильтрацию пакетов исключительно на основе адреса отправителя
Расширенный ACL - выполняет фильтрацию пакетов на основе расширенного набора критериев

56. Отличие технологии VPN от VLAN

VPN - Технология создания защищенных виртуальных частных сетей поверх публичных сетей (интернета). Обеспечивает шифрование, аутентификацию и целостность данных при передаче через ненадежные сети.

VLAN - Технология виртуального разделения одной физической сети на несколько логических сетей на уровне канального доступа. Обеспечивает изоляцию трафика и

управление широковещательным доменом в пределах одного кампуса. Использует тегирование фреймов

VLAN — комната в доме, VPN — туннель между домами