**Теоретический раздел**

**Сетевой уровень**, также известный как уровень сети или уровень 3 модели OSI (Open Systems Interconnection), это уровень, который отвечает за передачу данных от одной сети к другой. Этот уровень имеет дело с пакетами данных, а не с отдельными битами или байтами. Сетевой уровень также отвечает за определение маршрута, по которому пакеты будут перемещаться от исходного устройства к целевому.

1. **Сетевой уровень. Транзит пакетов: виды, особенности, способы. Принципы работы маршрутизатора**

**Транзит пакетов** обычно относится к процессу передачи пакетов данных через серию сетевых узлов или маршрутизаторов. Это необходимо, когда пакеты должны быть доставлены от источника к назначению через одну или несколько сетей.

Существуют разные типы транзита пакетов, включая:

* **Прямая доставка:** когда источник и назначение находятся в одной и той же сети, пакеты прямо доставляются от источника к назначению.
* **Непрямая доставка:** когда источник и назначение находятся в разных сетях, пакеты передаются через один или несколько маршрутизаторов.

**Маршрутизатор** — это устройство, работающее на сетевом уровне и определяющее лучший путь для каждого пакета данных. Маршрутизаторы имеют таблицы маршрутизации, которые используются для определения куда и как передать каждый пакет. Эти таблицы маршрутизации могут быть статическими (заданными вручную) или динамическими (автоматически обновляемыми с помощью протоколов маршрутизации).

Принцип работы маршрутизатора:

* При получении пакета данных, маршрутизатор сначала проверяет его заголовок, чтобы узнать, куда пакет должен быть отправлен.
* Затем маршрутизатор смотрит в свою таблицу маршрутизации, чтобы определить, через какой интерфейс отправить пакет.
* Пакет отправляется через выбранный интерфейс к следующему маршрутизатору на пути к его назначению.
* Этот процесс повторяется, пока пакет не достигнет своего конечного назначения.

1. **Сетевой уровень. Балансировка нагрузки. Ограничения и особенности данного уровня**

**Балансировка нагрузки — это** процесс распределения работы между несколькими компьютерами, процессами, дисками или другими ресурсами с целью оптимизации использования системы, уменьшения времени ответа и предотвращения перегрузки одного из ресурсов.

Балансировка нагрузки может производиться на разных уровнях, включая уровень сети, аппаратного обеспечения и приложений.

На сетевом уровне балансировщики нагрузки распределяют входящий сетевой трафик по нескольким серверам, чтобы обеспечить равномерное использование ресурсов и предотвратить сбои отдельных серверов. Балансировщики нагрузки также могут использоваться для обеспечения отказоустойчивости, так как в случае отказа одного сервера трафик может быть перенаправлен на другой работающий сервер.

Однако балансировка нагрузки также имеет свои ограничения и особенности:

* **Состояние сессий**: некоторые приложения требуют, чтобы все запросы от одного пользователя направлялись на один и тот же сервер в течение одной сессии. Балансировщик нагрузки должен быть способен обрабатывать такие требования.
* **Распределение нагрузки**: не все балансировщики нагрузки распределяют нагрузку равномерно между серверами. Некоторые модели балансировки могут предпочесть одни серверы другим в зависимости от их текущей нагрузки, производительности или других параметров.
* **Отказоустойчивость**: балансировщик нагрузки сам по себе может стать узким местом. Если балансировщик нагрузки выйдет из строя, все запросы могут быть прерваны. Чтобы предотвратить это, обычно используются несколько балансировщиков нагрузки в режиме высокой доступности.
* **Сложность настройки:** балансировка нагрузки может быть сложной для настройки и управления, особенно в сложных сетевых средах с большим количеством серверов и различными типами трафика.

1. **Сетевой уровень. Объединение сетей. Магистрали. Туннелирование. Преобразование адресов. Фрагментирование**

* **Объединение сетей:** это процесс связывания двух или более физических сетей в одну большую логическую сеть. Объединение может быть осуществлено с помощью мостов, маршрутизаторов или шлюзов, которые позволяют устройствам в разных сетях общаться друг с другом.
* **Магистрали**: В контексте сетей, магистраль (backbone) — это главный канал связи, который соединяет несколько локальных сетей (LAN) или широкополосных сетей (WAN). Все данные, передаваемые между этими сетями, должны проходить через магистраль.
* **Туннелирование**: это метод, позволяющий передавать данные из одной сети в другую через промежуточную сеть. При этом данные "упаковываются" в другие пакеты данных, которые могут быть переданы через промежуточную сеть. При достижении конечного пункта данные "распаковываются" и доставляются до конечного назначения. Это часто используется для создания виртуальных частных сетей (VPN).
* (VPN).
* **Преобразование адресов (NAT - Network Address Translation):** это процесс изменения IP-адресов в пакетах данных во время их передачи. Это обычно делается для того, чтобы несколько устройств могли делить один публичный IP-адрес или для того, чтобы обеспечить безопасность сети, скрывая внутренние IP-адреса от внешнего мира.
* **Фрагментирование**: это процесс разделения больших пакетов данных на меньшие фрагменты, которые могут быть переданы через сеть, которая имеет ограничения на максимальный размер пакета (MTU - Maximum Transmission Unit). Каждый фрагмент передается отдельно и затем собирается обратно в исходный пакет на получающей стороне.

**Транспортный уровень**, или уровень 4 в модели OSI, отвечает за обеспечение надежной передачи данных между системами. Он отвечает за установление соединения, контроль над передачей данных, управление потоками и обеспечение надежности и упорядоченности пакетов. Основные протоколы на этом уровне — это TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

1. **Транспортный уровень. Сетевые службы и порты. Сокеты**

**Сетевые службы** - это различные типы служб, которые используются в сетях для обмена информацией и выполнения различных задач. Примеры сетевых служб включают в себя веб-серверы (HTTP), почтовые серверы (SMTP, IMAP, POP3), серверы файлового обмена (FTP, SFTP), серверы баз данных и другие.

**Порты** используются для идентификации конкретных процессов или служб, которые работают на устройстве. Каждая сетевая служба на устройстве обычно ассоциирована с определенным портом. Например, веб-серверы обычно работают на порту 80 (для HTTP) или 443 (для HTTPS), а почтовые серверы обычно используют порты 25 (SMTP), 110 (POP3) или 143 (IMAP).

**Сокеты** - это программный интерфейс, который позволяет приложениям на одном устройстве общаться с приложениями на другом устройстве через сеть. Сокет представляет собой конечную точку коммуникации и состоит из IP-адреса и порта. Приложение создает сокет для отправки и приема данных через сеть.

Для обеспечения надежной передачи данных между сокетами, транспортный уровень использует протоколы, такие как TCP, которые обеспечивают надежность, упорядоченность и контроль потока данных. Для ненадежной, но быстрой передачи данных может использоваться протокол UDP.

1. **Транспортный уровень. Транспортный протокол. Особенности подключения, разрыва, отправки и получения данных**

**Транспортный протокол** в сетевых коммуникациях определяет, как и в каком порядке данные передаются между системами. Основными протоколами на транспортном уровне являются TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

**TCP** — это надежный протокол, который обеспечивает упорядоченную и безошибочную доставку пакетов данных.

* Подключение: перед тем как данные могут быть переданы, TCP устанавливает соединение между отправителем и получателем с помощью процесса, известного как "трехстороннее рукопожатие" (three-way handshake). Это гарантирует, что оба устройства готовы к обмену данными.
* Разрыв: после передачи данных, соединение закрывается с помощью процедуры, известной как "четырехстороннее рукопожатие" (four-way handshake).
* Отправка и получение данных: когда данные отправляются, TCP разбивает их на пакеты и присваивает каждому пакету номер в последовательности. Если пакеты приходят в неправильном порядке или теряются, TCP обеспечивает их повторную отправку.

**UDP** — это более простой и быстрый протокол, который не обеспечивает гарантии доставки или упорядоченности пакетов.

* Подключение и разрыв: UDP не устанавливает или не закрывает соединения. Это означает, что данные могут быть отправлены без предварительной установки соединения.
* Отправка и получение данных: когда данные отправляются через UDP, они также разбиваются на пакеты, но номера в последовательности не присваиваются. Если пакеты приходят в неправильном порядке, они не будут переупорядочены, и если пакеты теряются, они не будут повторно отправлены.

В зависимости от требуемого уровня надежности и быстродействия, приложения могут выбирать использовать TCP или UDP. Например, веб-серверы обычно используют TCP, чтобы гарантировать правильную доставку данных, в то время как стриминговые медиасервисы могут использовать UDP для обеспечения быстрой доставки данных, даже если это может привести к потере некоторых пакетов.

1. **Транспортный уровень. Контроль ошибок. Восстановление соединения.**

В сетевых коммуникациях контроль ошибок и восстановление соединения играют ключевую роль в обеспечении надежности и устойчивости работы сети.

**Контроль ошибок** — это процесс обнаружения и исправления ошибок, которые могут возникнуть при передаче данных по сети. Ошибки могут возникнуть из-за многих причин, включая помехи, потерю пакетов или повреждение данных.

* В TCP контроль ошибок осуществляется через механизм подтверждения приема пакетов (acknowledgements). Когда пакет данных достигает получателя, он отправляет подтверждение обратно отправителю. Если отправитель не получает подтверждение в течение определенного времени, он считает, что пакет был потерян, и отправляет его снова.
* Кроме того, TCP использует контрольную сумму для обнаружения ошибок в пакетах. Контрольная сумма — это небольшое значение, которое вычисляется из данных пакета и передается вместе с пакетом. Получатель вычисляет свою собственную контрольную сумму из данных и сравнивает ее с полученной. Если они не совпадают, это указывает на ошибку в данных.

**Восстановление соединения** — это процесс восстановления разорванного сетевого соединения без потери данных.

* В TCP восстановление соединения осуществляется с помощью механизма повторной передачи. Если соединение разрывается, TCP автоматически пытается его восстановить и повторно отправлять потерянные пакеты.
* Кроме того, TCP использует механизм контроля перегрузки для предотвращения переполнения сети пакетами. Если TCP обнаруживает, что сеть перегружена (например, из-за потери многих пакетов), он уменьшает скорость отправки данных, чтобы предотвратить дальнейшие перегрузки.

Эти механизмы контроля ошибок и восстановления соединения делают TCP особенно подходящим для приложений, где важна надежность передачи данных, таких как веб-браузеры или электронная почта. В то же время, протоколы, такие как UDP, которые не обеспечивают контроль ошибок или восстановление соединения, могут быть более подходящими для приложений, где скорость важнее

1. **Транспортный уровень. Контроль перегрузок**

**Контроль перегрузок** — это механизм, который предназначен для предотвращения переполнения сетевых ресурсов при передаче данных. Это особенно важно в случаях, когда трафик в сети превышает ее возможности по передаче данных.

Одним из наиболее известных примеров контроля перегрузок является алгоритм контроля перегрузки TCP. TCP контролирует скорость, с которой данные передаются между отправителем и получателем, чтобы избежать перегрузки сети.

Вот как это работает:

* **Медленный старт (Slow Start):** начинается с малого размера окна отправки и удваивает его с каждым подтверждением получения пакета. Это продолжается до достижения определенного порога или до тех пор, пока не произойдет потеря пакета.
* **Уклонение от перегрузки (Congestion Avoidance):** когда размер окна достигает порогового значения, рост окна становится линейным (т.е., окно увеличивается на 1 за каждый круг обмена данными). Если происходит потеря пакета, порог устанавливается в половину текущего размера окна, и процесс повторяется.
* **Быстрое восстановление (Fast Recovery):** если TCP обнаруживает потерю пакета, оно устанавливает порог в половину текущего размера окна, затем входит в фазу быстрого восстановления, увеличивая размер окна с каждым подтверждением, пока не получит подтверждение для потерянного пакета. После этого он переходит обратно в режим избегания перегрузки.

В целом, этот механизм позволяет TCP адаптироваться к изменениям в сети и поддерживать высокую производительность передачи данных при минимальном количестве потерь пакетов. Он важен для поддержания устойчивости всей сети и предотвращения проблем производительности из-за перегрузок.

1. **Уровень приложений. Клиент-серверная и распределённая модели. Сетевые службы и сервисы**

**Уровень приложений** (или уровень приложений в модели OSI) — это уровень, на котором работают приложения, которые взаимодействуют с пользователем, например, веб-браузеры, почтовые клиенты, мессенджеры и т.д. Протоколы на этом уровне обеспечивают обмен данными между различными приложениями, используя протоколы более низких уровней для передачи данных по сети.

**Клиент-серверная модель** — это архитектура, в которой клиенты (обычно персональные компьютеры или мобильные устройства) отправляют запросы на серверы, которые затем обрабатывают эти запросы и возвращают результаты. Веб-браузинг, электронная почта и большинство других сетевых служб основаны на клиент-серверной модели.

**Распределенная модель** — это архитектура, в которой задачи и службы распределяются по множеству узлов в сети, и каждый узел может служить и как клиент, и как сервер. Это обеспечивает более высокую отказоустойчивость, поскольку даже если один узел выйдет из строя, другие узлы могут продолжить предоставлять службу. Примером такой модели является технология блокчейна, где все участники сети имеют равный статус и хранят копию всего блокчейна.

**Сетевые службы** — это программы или процессы, которые предоставляют функциональность по работе с данными через сеть. Они могут включать в себя веб-серверы, почтовые серверы, базы данных, файловые серверы и другие.

**Сетевые сервисы** обычно относятся к тем же вещам, что и сетевые службы, но могут также относиться к услугам, предоставляемым интернет-провайдерами или другими компаниями, которые предоставляют инфраструктуру или услуги для работы в сети, такие как облачные услуги или VPN.

1. **World Wide WEB. Принципы организации и работы**

**World Wide Web (WWW или просто Web)** — это информационная система, где документы и другие веб-ресурсы идентифицируются по URL, причем каждый ресурс связан с другими посредством гипертекстовых ссылок. Все эти ресурсы доступны через интернет.

Вот основные принципы организации и работы World Wide Web:

* **URL:** Каждая веб-страница имеет свой уникальный URL (Uniform Resource Locator), который можно использовать для доступа к странице.
* **HTTP и HTTPS:** Веб-страницы обычно передаются по протоколу HTTP (HyperText Transfer Protocol) или HTTPS (HTTP Secure), который является защищенной версией HTTP.
* **HTML:** Веб-страницы обычно написаны на языке разметки HTML (HyperText Markup Language), который определяет структуру и формат страницы.
* **Серверы и клиенты:** Веб работает по модели клиент-сервер. Веб-сервер хранит веб-страницы и отправляет их в ответ на запросы от веб-клиентов, обычно веб-браузеров.
* **Гипертекст:** Веб использует гипертекст для создания ссылок между веб-страницами. Это означает, что на одной веб-странице можно нажать на ссылку, чтобы перейти к другой веб-странице.
* **Веб-сервисы:** на веб-серверах также могут работать веб-сервисы, которые обеспечивают интерактивные функции веб-страниц, такие как поиск, регистрация пользователей, корзины для интернет-магазинов и т.д. Эти веб-сервисы обычно используют технологии, такие как JavaScript, PHP, Ruby, Python и другие.
* **CSS и JavaScript:** CSS (Cascading Style Sheets) используется для контроля над внешним видом веб-страницы, в то время как JavaScript обеспечивает интерактивность страницы.
* **Поисковые системы:** Веб использует поисковые системы, такие как Google, для индексации веб-страниц и предоставления результатов поиска пользователям.

Это общие принципы организации и работы World Wide Web, хотя в реальности веб является сложной системой, включающей множество различных технологий и стандартов.

**Прикладные вопросы**

1. **Полезиохронная цифровая иерархия (PDH). Каналы T1-T4 (E1-E4)**

Полезиохронная цифровая иерархия (PDH, Plesiochronous Digital Hierarchy) - это технология, используемая в телекоммуникационных сетях для передачи больших объемов данных по цифровым транспортным каналам. В PDH данные разделяются на кадры, которые передаются по сети.

Каналы T1-T4 и E1-E4 относятся к уровням в иерархии PDH, каждый из которых имеет разные скорости передачи данных:

* **T1:** это канал с скоростью передачи 1,544 Мбит/с, который часто используется в Северной Америке. Он может передавать 24 цифровых голосовых канала.
* **E1:** это аналог T1, используемый в Европе. Его скорость передачи составляет 2,048 Мбит/с, и он может передавать 30 цифровых голосовых каналов.
* **T2, T3 и T4:** это каналы со скоростью передачи 6,312 Мбит/с, 44,736 Мбит/с и 274,176 Мбит/с соответственно. Они используются в Северной Америке.
* **E2, E3 и E4:** это каналы со скоростью передачи 8,448 Мбит/с, 34,368 Мбит/с и 139,264 Мбит/с соответственно. Они используются в Европе.

Однако стоит отметить, что PDH сейчас уже устарела и заменена более современной технологией SDH (Synchronous Digital Hierarchy) или в США SONET (Synchronous Optical Network), которые обеспечивают большую гибкость и надежность.

1. **Синхронная цифровая иерархия. Каналы SDH (SONET)**

Синхронная цифровая иерархия (SDH) и Синхронная оптическая сеть (SONET) - это стандарты для передачи информации через оптоволоконные кабели. Они разработаны для замены полезиохронной цифровой иерархии (PDH), которая использовалась в телекоммуникациях в прошлом.

В отличие от PDH, в SDH и SONET используются синхронные кадры данных, что облегчает мультиплексирование и демультиплексирование различных сигналов и обеспечивает большую пропускную способность и надежность.

Основные каналы SDH и SONET:

* **STM-1/OC-3:** это базовый уровень в SDH с пропускной способностью 155,52 Мбит/с. В SONET аналогичный уровень называется OC-3 (Optical Carrier 3).
* **STM-4/OC-12:** Этот уровень имеет пропускную способность 622,08 Мбит/с.
* **STM-16/OC-48:** Этот уровень имеет пропускную способность 2,488 Гбит/с.
* **STM-64/OC-192:** Этот уровень имеет пропускную способность 9,953 Гбит/с.

Преимущество SDH и SONET в том, что они могут гибко объединять и разделять сигналы различных источников, а также предоставлять функции защиты и восстановления. Они являются ключевой технологией в современных телекоммуникационных сетях, особенно в сетях связи длинного действия и международных соединений.

1. **Технологии CDMA и TDMA**

**CDMA (Code Division Multiple Access)** и **TDMA (Time Division Multiple Access)** - это две различные технологии, которые используются для того, чтобы несколько пользователей могли одновременно использовать одну и ту же частоту без взаимного вмешательства.

**TDMA**: В TDMA, пространство и частота остаются прежними, но время делится на слоты. Каждый пользователь получает определенный слот времени для передачи и получения данных, то есть доступ к каналу предоставляется по очереди. В каждый момент времени канал используется только одним пользователем, но этот пользователь периодически меняется.

**CDMA**: В CDMA, пространство и время остаются прежними, но каждому пользователю присваивается уникальный код. Все пользователи могут одновременно передавать данные по одной и той же частоте, и их сигналы разделяются с помощью уникальных кодов. Это позволяет CDMA эффективно использовать доступное пространство частот, и это обеспечивает большую пропускную способность, чем TDMA, особенно в условиях неравномерного трафика.

Обе технологии имеют свои преимущества и недостатки. TDMA обычно проще для реализации и управления, но может быть менее эффективным при неравномерном трафике. CDMA обеспечивает большую пропускную способность и лучше справляется с неравномерным трафиком, но требует более сложных алгоритмов для кодирования и декодирования сигналов.

1. **Протоколы доступа (SLIP, PPP)**

SLIP (Serial Line Internet Protocol) и PPP (Point-to-Point Protocol) — это протоколы, которые используются для установления прямого соединения между двумя узлами в сети. Они обычно используются для подключения к Интернету через модемы или другие последовательные линии.

* **SLIP** — это старый и простой протокол, который был одним из первых методов подключения к Интернету. Он просто берет пакеты IP и отправляет их по последовательной линии. Он не поддерживает функции, такие как аутентификация или сжатие данных. Кроме того, он предполагает, что каждая сторона соединения знает IP-адрес другой стороны заранее.
* **PPP** — это более современный и сложный протокол, который заменил SLIP в большинстве приложений. Он поддерживает функции, такие как аутентификация, сжатие данных и динамическое присвоение IP-адресов. Он может также передавать пакеты не-IP протоколов.

Таким образом, PPP предлагает больше возможностей и гибкости, чем SLIP, и он обычно предпочтительнее для современных соединений. Однако оба протокола все еще могут использоваться в некоторых специфических ситуациях или старых системах.

1. **Высокопроизводительные и магистральные сетевые технологии (ATM, MPLS, PBB)**

ATM (Asynchronous Transfer Mode), MPLS (Multiprotocol Label Switching) и PBB (Provider Backbone Bridges) — это высокопроизводительные сетевые технологии, предназначенные для передачи больших объемов данных через сеть.

* **ATM** — это технология, которая предоставляет возможность одновременной передачи голосовых, видео и данных через одну и ту же сеть. ATM использует фиксированный размер ячейки для передачи данных, что обеспечивает высокую скорость передачи и низкую задержку. Это особенно важно для интерактивного видео и голосовых вызовов.
* **MPLS** — это технология, которая упрощает процесс маршрутизации пакетов в сети. Вместо того, чтобы анализировать каждый пакет, MPLS использует короткие метки для идентификации пакетов, что ускоряет процесс маршрутизации и обеспечивает лучшую производительность. MPLS также поддерживает "инженерию трафика", что позволяет провайдерам контролировать пути, по которым проходит трафик, для оптимизации работы сети.
* **PBB** (Provider Backbone Bridges, также известны как Mac-in-Mac) — это технология, используемая для расширения сетей Ethernet на протяжении больших расстояний. PBB использует дополнительный MAC-адрес (называемый "B-MAC") для идентификации конечных точек соединения, что обеспечивает более эффективное масштабирование, чем традиционные технологии Ethernet.

Эти технологии имеют свои особенности и могут быть выбраны в зависимости от специфических требований сетевой инфраструктуры.

1. **Сетевое оборудование: виды, функции, характеристики, производители**

Сетевое оборудование служит для создания, управления и поддержания сетей данных. Оно обеспечивает передачу, маршрутизацию и фильтрацию пакетов данных между различными устройствами в сети. Вот некоторые основные типы сетевого оборудования:

* **Маршрутизаторы (Routers):** Маршрутизаторы служат для соединения двух или более сетей и для маршрутизации пакетов данных между ними. Они могут работать на нескольких уровнях сетевой модели OSI, но основная их функция связана с сетевым уровнем (уровнем 3).
* **Сетевые коммутаторы (Switches**): Коммутаторы работают на уровне передачи данных (уровне 2) и служат для соединения устройств внутри одной сети. Они передают пакеты данных конкретному устройству, а не широковещательно, что делает их более эффективными, чем обычные хабы.
* **Межсетевые экраны (Firewalls):** Межсетевые экраны служат для защиты сети от нежелательного внешнего трафика. Они могут блокировать или разрешать трафик на основе набора правил безопасности.
* **Беспроводные точки доступа (Wireless Access Points):** Беспроводные точки доступа обеспечивают беспроводной доступ к проводной сети. Они часто используются в офисных или общественных сетях Wi-Fi.
* **Сетевые мосты (Bridges):** Сетевые мосты служат для соединения двух сегментов сети на уровне передачи данных. Они фильтруют трафик и позволяют только необходимым пакетам переходить из одного сегмента в другой.
* **Модемы:** Модемы преобразуют сигналы между формами, подходящими для передачи по телефонным или кабельным линиям, и формами, подходящими для обработки цифровыми устройствами, такими как компьютеры.

Ведущие производители сетевого оборудования включают такие компании, как Cisco, Juniper Networks, Huawei, Arista Networks, Hewlett Packard Enterprise, и многие другие.

1. **Сетевое оборудование CISCO/Mikrotik**

Cisco и MikroTik - два ведущих производителя сетевого оборудования, каждый со своими сильными сторонами и предпочтительными областями применения.

* **Cisco** — это одна из ведущих мировых компаний по производству сетевого оборудования. Cisco предлагает широкий спектр продуктов, включая маршрутизаторы, коммутаторы, системы безопасности и оборудование для беспроводных сетей. Эти устройства обычно отличаются высоким качеством и надежностью, и они широко используются в корпоративных и предприятий сетях. Компания также предлагает обширный набор сертификаций для профессионалов в области IT, которые хотят углубить свои знания и навыки работы с оборудованием Cisco.
* **MikroTik** - компания, основанная в Латвии, которая производит сетевое оборудование для провайдеров интернет-услуг, небольших и средних предприятий и домашних пользователей. Продукты MikroTik обычно более доступны по цене, чем оборудование Cisco, и они отличаются большой гибкостью благодаря операционной системе RouterOS, которая позволяет настраивать очень сложные сетевые конфигурации. MikroTik производит широкий ассортимент оборудования, включая маршрутизаторы, коммутаторы и беспроводное оборудование.

Выбор между Cisco и MikroTik обычно зависит от специфических требований и бюджета. Если вам нужно надежное и проверенное оборудование для большого предприятия, Cisco может быть хорошим выбором. Если вы ищете более доступное и гибкое решение, MikroTik, может быть, стоит рассмотреть

1. **Сети Ethernet на разделяемой среде**

**Сети Ethernet на разделяемой среде** — это один из ранних типов Ethernet сетей, в которых все устройства в сети использовали общую среду передачи данных. Это обычно означало, что все устройства были подключены к одной и той же коаксиальной линии (или "шине"), и каждое устройство могло "слушать" трафик, передаваемый другими устройствами.

В таких сетях применялся метод доступа CSMA/CD (Carrier Sense Multiple Access with Collision Detection). С его помощью устройства "слушали" среду перед началом передачи, чтобы убедиться, что никто другой в данный момент не передает. Если среда была занята, устройство ожидало, пока она освободится. Если два устройства начинали передачу одновременно, происходила "коллизия", которую устройства могли обнаружить и попытаться передать данные снова после случайного периода времени.

Сети Ethernet на разделяемой среде были простыми и недорогими в установке, но они имели ограничения в отношении пропускной способности и масштабируемости. Коллизии становились все более вероятными по мере увеличения числа устройств в сети, что снижало общую пропускную способность.

С течением времени эти сети в большинстве случаев были заменены сетями Ethernet на базе коммутаторов, в которых каждое устройство подключено к собственному порту коммутатора и имеет полный доступ к пропускной способности сети без возникновения коллизий.