1. Перечислить все ***сетевые утилиты***.

Сетевые утилиты - это программы, предназначенные для работы с сетью.

* **Ping** - используется для проверки доступности хоста в сети.
* **Tracert** - используется для определения маршрута прохождения пакетов от исходного узла до целевого.
* **Ipconfig** - используется для просмотра и изменения настроек сетевых интерфейсов компьютера.
* **Netstat** - используется для просмотра сетевой статистики и списка активных соединений.
* **Nslookup** - используется для выполнения запросов DNS-имен и просмотра соответствующих IP-адресов.
* **Route** - утилита route используется для манипулирования таблицами сетевой маршрутизации вручную.
* **Arp** - Утилита ***arp***предоставляет возможностьпросматривать и модифицировать таблицу соответствия IP адресов и физических адресов вашего компьютера.
* **Host-name** – позволяет получить текущее название компьютера
* **Nbstat** - Windows-утилита, работающая из командной строки, которая может быть использована для отображения информации о NetBIOS-соединениях и таблицах имен. NBTSTAT может собирать такую информацию, как MAC-адреса, имена NetBIOS, доменные имена, и о других активных пользователях.
* **Net** - это сетевая утилита, доступная в операционных системах Windows и Linux, которая предоставляет различные инструменты для администрирования сетей.

1. Перечислить все ***уровни модели OSI/ISO***. Описать назначение каждого уровня.

Уровни модели OSI/ISO:

* Физический уровень - обеспечивает физическую передачу данных через канал связи. Физический уровень определяет свойства среды передачи данных (коаксиальный кабель, витая пара, оптоволоконный канал и т.п.) и способы ее соединения с сетевыми адаптерами: технические характеристx ики кабелей (сопротивление, емкость, изоляция и т.д.), перечень допустимых разъемов, способы обработки сигнала и т.п.
* Канальный уровень - обеспечивает обработку ошибок и управление доступом к среде передачи данных. Формирует кадры из пакетов и готовит их к передаче на физический уровень. Проверяет доступность физической среды передачи и исправляет ошибки. Для обеспечения корректности передачи помещает в начало и конец каждого кадра специальную последовательность бит и высчитывает контрольную сумму. Определяет правила использования физического канала. При большом размере блока данных делит его на кадры меньшего размера. Функции этого уровня реализуются сетевыми адаптерами и их драйверами. Делится на два подуровня: LLC (logical link control)-подуровень - логический контроль связи

MAC (media access control)-подуровень — контролирует доступ к физической среде

LLC находится выше MAC.

* Сетевой уровень - обеспечивает маршрутизацию, адресацию и передачу пакетов данных между различными сетями.
* Транспортный уровень - обеспечивает установление и контроль соединения, разбиение и сборку данных. Основным назначением процедур транспортного уровня является подготовка и доставка пакетов данных между конечными точками без ошибок и в правильной последовательности. Восстанавливает передачу после ошибок, разбивает сообщение на удобные блоки, распределяет приоритеты передачи и подтверждает передачу. Контроль передачи данных, разбиение данных на блоки, индексация блоков данных.
* Сеансовый уровень - обеспечивает управление соединением между устройствами и синхронизацию передачи данных, которые работают в сети. Следует отметить, что сеансовый уровень - это точка взаимодействия программ и компьютерной сети.

Обеспечивает управление диалогом, определяя, какой из сторон является активной. Координирует сеанс (диалог), может устанавливать контрольные точки

Определяет, какой будет передача между двумя прикладными процессами: (Дуплексный\полудуплексный). Установка, поддержание и разрыв сеанса.

* Представительский уровень - обеспечивает преобразование данных в удобный для передачи формат. Процедуры этого уровня описывают способы шифрования, сжатия и преобразования наборов символов данных.
* Прикладной уровень - обеспечивает работу прикладных программ. Основное назначения уровня: определить способы взаимодействия пользователей с системой (определить интерфейс).

1. Поясните понятие ***сетевой протокол***.

Сетевой протокол - это набор правил и процедур, определяющих формат и порядок передачи данных между соседними уровнями модели ISO/OSI.

1. Указать, где в OSI/ISO проходит ***граница между аппаратным и программным обеспечением***.

Граница между аппаратным и программным обеспечением проходит на канальном уровне модели OSI/ISO (LLC и MAC)

LLC – подуровень управления логической связью

MAC – управление доступом к среде (нижний уровень канального)

1. Определить понятие ***CSMA/CD***.

CSMA/CD (Carrier Sense Multiple Access/Collision Detection) - это метод доступа к среде передачи данных, при котором устройство перед началом передачи данных прослушивает среду на наличие других передач, и в случае обнаружения коллизии (одновременной передачи) прекращает передачу.

Устанавливает следующий порядок: если рабочая станция «хочет» воспользоваться сетью для передачи данных, она сначала должна проверить состояние канала: начинать передачу рабочая станция можно, если канал свободен.

В процессе передачи рабочая станция продолжает прослушивание сети для обнаружения возможных конфликтов (коллизий). Если возникает конфликт из-за того, что два узла попытаются занять канал, то обнаружившая конфликт интерфейсная плата соответствующего компьютера выдает в сеть специальный сигнал, и обе станции одновременно прекращают передачу. Принимающая рабочая станция отбрасывает частично принятое сообщение, а все рабочие станции, желающие передать сообщение, в течение некоторого, случайно выбранного промежутка времени выжидают, прежде чем начать сообщение.

Все сетевые интерфейсные платы запрограммированы на разные псевдослучайные промежутки времени ожидания. Если конфликт возникнет во время повторной передачи сообщения, этот промежуток времени будет увеличен.

1. Как называется ***программное обеспечение реализующий подуровень LLC канального уровня***.

Подуровень LLC канального уровня реализуется программным обеспечением под названием LLC (Logical Link Control). **Драйвер.**

1. Что определяет ***спецификация NDIS***?

Спецификация NDIS (network driver interface specification) определяет интерфейс между сетевыми адаптерами и драйверами сетевых протоколов в операционной системе Windows.

1. Свойства ***ненадежных протоколов***. Примеры ненадежных и ***надежных*** протоколов

Ненадежные протоколы - это протоколы, которые не гарантируют доставку сообщений. Они могут быть использованы в условиях, когда потеря сообщения не сильно повлияет на работу приложения. Ненадежные протоколы обычно используются для передачи данных в реальном времени, например, видео- и аудиопотоков.

Примеры ненадежных протоколов: UDP(User Datagram Protocol ), ICMP(Internet Control Message Protocol).

1) отсутствие механизмов обеспечения надежности;

2) отсутствие гарантий доставки;

3) отсутствие обработки соединений;

4) отсутствие буферизации;

5) отсутствие фрагментации;

6) отсутствие соединения.

1. Не гарантируют доставку данных.

2. Не контролируют целостность информации.

3. Не обеспечивают проверку подлинности.

4. Могут вызвать ошибки в передаче данных.

5. Могут привести к дублированию информации.

6. Могут привести к потере части информации.

7. Не предоставляют механизмов обнаружения и коррекции ошибок в передаче данных.

8. Не обеспечивают контроль порядка доставки пакетов данных.

9. Могут быть подвержены атакам и злоупотреблениям.

10. Не могут гарантировать минимальную скорость передачи данных.

11. Могут быть менее эффективными в условиях высоких нагрузок на сеть.

12. Могут потреблять больше ресурсов системы при передаче данных.

13. Не могут обеспечивать безопасность передаваемых данных, так как отсутствует шифрование и защита от несанкционированного доступа.

14. Могут быть подвержены сбоям в работе, что может привести к сбою в работе сети в целом.

15. Не могут обеспечивать гарантированную доставку в режиме реального времени, что может быть критично для некоторых приложений.

16. Не могут быть использованы для передачи больших объемов данных, так как отсутствует механизм фрагментации и сборки пакетов.

17. Могут вызвать конфликты и перегрузку в сети, особенно при использовании в условиях высокой нагрузки на сеть.

Надежные протоколы - это протоколы, которые гарантируют доставку сообщений. Они обычно используются для передачи данных, которые не могут быть потеряны, например, файлов и электронной почты.

Примеры надежных протоколов: TCP(Transmission Control Protocol) , SMTP(Simple Mail Transfer Protocol).

1. Перечислить все уровни ***модели TCP/IP***. Описать назначение каждого уровня. Привести примеры протоколов каждого уровня.

Модель TCP/IP состоит из 4 уровней:

1. Уровень доступа к сети (Network Access Layer): Определяет физические способы передачи данных по сети. Примеры протоколов: Ethernet, Wi-Fi, Token Ring.
2. Интернет-уровень (межсетевой) (Internet Layer): Обеспечивает маршрутизацию пакетов между сетями. Примеры протоколов: IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol), RARP (Reverse ARP).
3. Транспортный уровень (Transport Layer): Обеспечивает передачу данных между приложениями на разных узлах сети. Примеры протоколов: TCP, UDP.
4. Прикладной уровень (Application Layer): Обеспечивает взаимодействие между приложениями и пользователями. Примеры протоколов: HTTP, FTP, SMTP, DNS, DHCP.
5. Поясните понятия ***хост***, ***адрес хоста***, ***имя хоста***.

Хост - это любое устройство, подключенное к компьютерной сети. Хост может быть компьютером, маршрутизатором, сервером и т.д.

Адрес хоста - это уникальный идентификатор хоста в сети. В IP-сети это обычно IP-адрес.

Имя хоста - это текстовый идентификатор, используемый для обращения к хосту в сети. Имя хоста может быть связано с адресом хоста при помощи DNS (системы доменных имен).

Имя хоста – это символьное имя, которое может использоваться для идентификации устройства в сети вместо его адреса.

1. Какая ***организация поддерживает сетевые*** протоколы Internet. Как называются ***документы, описывающие эти протоколы***.

Сетевые протоколы интернета поддерживаются и разрабатываются Internet Engineering Task Force (IETF). Документы, описывающие эти протоколы, называются RFC (Request for Comments).

1. Что такое ***МАС-адрес***? Структура ***Ethernet МАС-адреса***.

МАС-адрес (Media Access Control address) - это уникальный идентификатор, присваиваемый сетевому интерфейсу на физическом уровне. МАС-адрес используется для идентификации устройства в локальной сети. Структура Ethernet МАС-адреса состоит из 6 байт, принадлежащих двум различным категориям: OUI (Organizationally Unique Identifier) и номер устройства.

В качестве стандартного выбран 48-битный формат адреса, что соответствует примерно 280 триллионам различных адресов.

Младшие 24 разряда кода адреса называются OUA– уникальный адрес. Именно их присваивает каждый из зарегистрированных производителей сетевых адаптеров.

Следующие 22 разряда кода называются OUI – уникальный идентификатор. IEEE присваивает один или несколько OUI каждому производителю сетевых адаптеров. UAA (Universally Administered Address) – универсально управляемый адрес или IEEE-адрес.

Два старших разряда адреса управляющие, они определяют тип адреса, способ интерпретации. **Старший бит I/G (Individual/Group) указывает на тип адреса. Если он установлен в 0, то индивидуальный, если в 1, то групповой (многопунктовый или функциональный)**. **Второй управляющий бит U/L (Universal/Local) называется флажком универсального/местного управления и определяет, как был присвоен адрес данному сетевому адаптеру. Обычно он установлен в 0. Установка бита U/L в 1 означает, что адрес задан не производителем сетевого адаптера, а организацией, использующей данную сеть.**

1. Как ***посмотреть MAC-адрес сетевой карты*** на компьютере?

Чтобы посмотреть MAC-адрес сетевой карты на компьютере, можно использовать команду ipconfig /all в командной строке на Windows или ifconfig на Unix-подобных системах. **Физический адрес.**

1. Основное назначение ***межсетевого уровеня***.

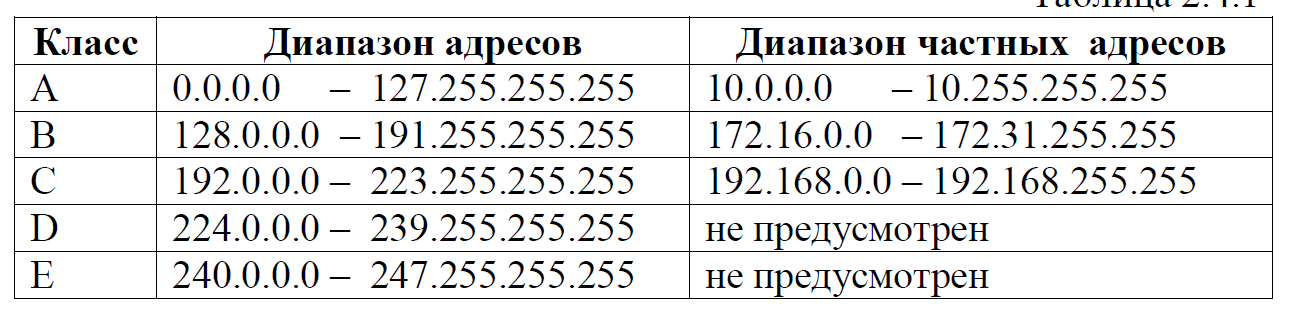
Основное назначение межсетевого уровня - маршрутизация пакетов между различными сетями в компьютерной сети. **Доставка дейстаграмм.** Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и заторов в сети.

1. Структура ***IP-адреса***.

IP-адрес - это уникальный идентификатор устройства в сети. Он состоит из 32-битного числа, разделенного на 4 байта (октета), каждое из которых может принимать значения от 0 до 255. Состоит из подсети и номера узла.

1. Типы ***IP-адресации***. ***Классы адресов Internet***.

Типы IP-адресации включают статическую и динамическую адресацию. Классы адресов Internet включают A, B, C, D и E.





1. Поясните понятия ***публичный IP-адрес*** и ***частный IP-адрес***.

Публичный IP-адрес - это IP-адрес, который используется для связи с другими устройствами в Интернете. Частный IP-адрес - это IP-адрес, который используется для связи с устройствами в локальной сети.

1. Как посмотреть ***IP-адрес компьютера***.

Чтобы узнать IP-адрес компьютера, можно использовать команду ipconfig в командной строке на Windows или ifconfig на Unix-подобных системах.

1. Как протестировать ***IP-соединение в локальной сети*** ?

Для тестирования IP-соединения в локальной сети можно воспользоваться утилитой ping, которая отправляет ICMP-пакеты на указанный IP-адрес и ожидает ответа. Для этого нужно открыть командную строку (в Windows), ввести команду "ping" и IP-адрес целевого компьютера. Например, для проверки соединения с компьютером с IP-адресом 192.168.1.2 нужно ввести "ping 192.168.1.2". Если соединение работает, то в ответ на запрос должны прийти ICMP-пакеты. Если пакеты не доходят до целевого компьютера, то возможно проблемы с сетевым соединением или настройками компьютера.

1. Как получить ***перечень сетевых узлов*** между двумя хостами?

Чтобы получить перечень сетевых узлов между двумя хостами, можно воспользоваться командой traceroute (или tracert в Windows), указав в качестве аргументов IP-адреса источника и назначения. Эта команда отправляет пакеты с увеличивающимся значением TTL (Time To Live) и получает ответы от каждого промежуточного маршрутизатора, пока не достигнет конечной точки.

1. Перечислите ***параметры настройки TCP/IP***.

Некоторые параметры настройки TCP/IP:

* IP-адрес: уникальный идентификатор компьютера в сети
* Маска подсети: определяет, какие биты IP-адреса относятся к сети, а какие - к хосту
* Шлюз по умолчанию: IP-адрес маршрутизатора, через который осуществляется доступ к другим сетям
* DNS-серверы: серверы, которые преобразуют доменные имена в IP-адреса
* DHCP: протокол, который автоматически назначает IP-адреса компьютерам в сети

1. Поясните понятие ***маска подсети***.

Маска подсети - это число, которое определяет, какие биты в IP-адресе относятся к сети, а какие - к хосту, состоят из 32 битов. Она записывается в виде последовательности битов, где единицы обозначают биты, относящиеся к сети, а нули - к хосту. Маска подсети позволяет разбить IP-адрес на сетевую часть и хостовую часть, что упрощает маршрутизацию пакетов в сети.

1. Основные ***отличия между IPv4 и IPv6***.

IPv4 и IPv6 - это протоколы сетевого уровня, которые используются для передачи данных в Интернете. Основные отличия между ними:

* IPv4 использует 32-битные адреса, а IPv6 - 128-битные адреса, что позволяет получить огромное количество уникальных адресов.
* IPv4 использует классовую адресацию, а IPv6 - нет.
* IPv4 использует ARP для разрешения MAC-адресов, а IPv6 - NDP (Neighbor Discovery Protocol).
* IPv6 поддерживает механизм автонастройки адресов, что упрощает настройку сети.

1. Поясните понятие ***сетевой порт***. На каком уровне модели TCP/IP это понятие определено.

Сетевой порт - это номер, который идентифицирует конкретный процесс или приложение, которое использует протокол TCP или UDP, который обрабатывает входящие сетевые запросы или отправляет их. Это понятие определено на транспортном уровне модели TCP/IP.

Процесс, получающий или отправляющий данные с помощью Транспортного уровня, идентифицируется номером, который называется номером порта.

1. Как ***классифицируются сетевые порты***.

Сетевые порты классифицируются на три категории:

* Известные порты (well-known ports) – это порты, которые привязаны к известным службам, таким как HTTP (80), FTP (21), Telnet (23) и др. Они имеют номера от 0 до 1023.
* Зарегистрированные порты (registered ports) – это порты, которые зарезервированы для приложений, которые не относятся к известным службам. Они имеют номера от 1024 до 49151.
* Динамические или частные порты (dynamic/private ports) – это порты, которые используются приложениями для временной коммуникации с другими узлами в сети. Они имеют номера от 49152 до 65535.

1. Как посмотреть какими программами заняты сетевые порты на компьютере?

Чтобы посмотреть, какие программы заняты сетевыми портами на компьютере, можно использовать командную строку или специальные утилиты, такие как netstat. Например, чтобы узнать, какие порты открыты на компьютере, можно в командной строке ввести команду netstat -a, и это покажет список всех открытых портов и связанных с ними программ.

1. Поясните понятие архитектура ***клиент/сервер***.

Архитектура клиент/сервер – это модель взаимодействия компьютерных программ, где одна программа (клиент) запрашивает ресурсы или услуги у другой программы (сервер). Сервер обрабатывает запрос и возвращает ответ клиенту. Такая модель широко используется в сетевых приложениях, таких как веб-серверы, базы данных, почтовые серверы и другие.

1. Что такое ***сетевая служба***. Приведите примеры сетевых служб.

Сетевая служба - это программа, которая предоставляет доступ к определенной функциональности в компьютерной сети. **Программная реализация протокола сетевого уровня.**

Примеры сетевых служб включают веб-серверы, почтовые серверы, FTP-серверы, служба разрешения имен(DNS), служба автоматического выделения адресов(DHCP) и т.д.

1. Поясните понятие ***интерфейс внутренней петли***.

Интерфейс внутренней петли (loopback interface) - это виртуальный интерфейс сетевой карты, который используется для маршрутизации сетевых пакетов на том же компьютере. Он имеет IP-адрес 127.0.0.1 и используется для проверки работоспособности сетевых приложений, установки и проверки сетевых настроек и тестирования сетевых устройств.

1. Назначение сетевых служб ***DNS*** и ***DHCP***.

Сетевые службы DSN (Domain Name System) и DHCP (Dynamic Host Configuration Protocol) предназначены для облегчения настройки и управления сетевыми устройствами. DSN обеспечивает преобразование доменных имен в IP-адреса, что позволяет пользователям легко идентифицировать устройства в сети. DHCP автоматически назначает IP-адреса устройствам в сети и обеспечивает дополнительную конфигурационную информацию, такую как маску подсети и адрес шлюза по умолчанию.

1. Организация, ведающая распределением ***IP-адресов***, поддержкой ***сетевых доменов Internet верхнего уровня***, ***регистрацией портов***.

Организацией, которая ведает распределением IP-адресов, поддержкой сетевых доменов Internet верхнего уровня и регистрацией портов, является ICANN (Internet Corporation for Assigned Names and Numbers). Эта организация управляет инфраструктурой, необходимой для функционирования сети Интернет, и определяет стандарты и правила, связанные с регистрацией доменных имен и IP-адресов.

1. Поясните понятие ***сетевой сокет***.

Сетевой сокет - это абстрактное понятие, которое представляет комбинацию IP-адреса и номера порта для установления соединения между двумя устройствами в сети. Сокеты используются для создания конечных точек, которые обеспечивают взаимодействие между клиентом и сервером в сети.

1. Назначение ***стандарта POSIX***.

Стандарт POSIX (Portable Operating System Interface) определяет интерфейс между операционной системой и прикладным программным обеспечением для обеспечения переносимости между различными операционными системами. POSIX определяет стандарты для программирования взаимодействия с файловой системой, процессами, сокетами и другими основными системными ресурсами.

1. ***Структура TCP/IP TCP-сервера***. Все функции и все параметры функций.

Структура TCP/IP TCP-сервера включает в себя следующие функции:

* создание сокета с помощью функции **socket()**;
* связывание сокета с IP-адресом и портом с помощью функции **bind()**;
* ожидание входящего соединения с помощью функции **listen()**;
* принятие входящего соединения и создание нового сокета для обмена данными с помощью функции **accept()**;
* обмен данными между клиентом и сервером с помощью функций **read()** и **write()**.

1. ***Структура TCP/IP TCP-клиента***. Все функции и все параметры функций.

Структура TCP/IP TCP-клиента включает в себя несколько функций для установления соединения, передачи данных и завершения соединения:

1. **socket()**: функция для создания нового сокета клиента.
   * Параметры функции:
     + **domain** - домен сокета, который должен использоваться (обычно **AF\_INET** для IPv4 или **AF\_INET6** для IPv6).
     + **type** - тип сокета, который должен использоваться (обычно **SOCK\_STREAM** для TCP или **SOCK\_DGRAM** для UDP).
     + **protocol** - протокол, который должен использоваться (обычно **0** для автоматического выбора соответствующего протокола).
2. **connect()**: функция для установления соединения с удаленным сервером.
   * Параметры функции:
     + **sockfd** - идентификатор сокета клиента, который должен быть использован для установления соединения.
     + **addr** - структура, содержащая IP-адрес и порт сервера, с которым нужно установить соединение.
3. **send()**: функция для отправки данных на удаленный сервер.
   * Параметры функции:
     + **sockfd** - идентификатор сокета клиента, который должен быть использован для отправки данных.
     + **buf** - буфер с данными, которые должны быть отправлены.
     + **len** - размер буфера данных.
4. **recv()**: функция для приема данных от удаленного сервера.
   * Параметры функции:
     + **sockfd** - идентификатор сокета клиента, который должен быть использован для приема данных.
     + **buf** - буфер, в который будут записаны принятые данные.
     + **len** - максимальный размер данных, которые могут быть приняты.
5. **close()**: функция для завершения соединения с удаленным сервером и освобождения ресурсов, связанных с сокетом.
   * Параметры функции:
     + **sockfd** - идентификатор сокета клиента, который должен быть закрыт.