Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа №1

**Анализ сети.**

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверил: Путевская Ирина Валерьевна

Чебоксары, 2025

**Задачи**:  
    Изучить документацию по сетевым утилитам:

ifconfig/ipconfig, arp, route, netstat или команды iproute2: link, addr, route, neigh, ss

iputil: ping, traceroute/tracert или tracepath, iftop, iperf

nslookup или drill или dig,

wireshark или tcpdump, nmap, whois.

ncat, wget или curl

iptables  
    На подключенных к сети компьютерах собрать информацию об mac- и ip-адресе, маске сети, маршрутах, dns-серверах, запущенных сетевых службах.  
    Просканировать сеть на доступные сетевые службы, адреса доменной зоны.  
    По расположению проводов найти сетевое оборудование, шлюз в Интернет.  
    Построить схему сети в эмуляторе (cisco packettracer, GNS3, Dynamips, UNETLAB(EVE-NG), Common Open Research Emulator, Mininet,Line Network Emulator).

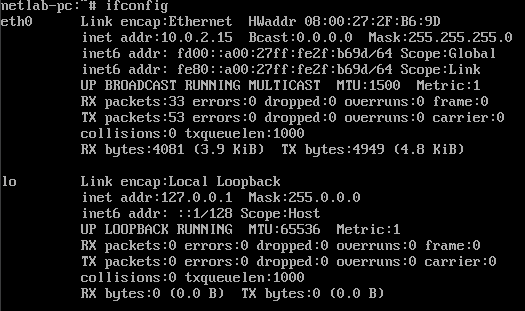
**Выполнение**

**Команда ifconfig/ipconfig.**

**Команда ifconfig (Interface Configuration)** —помогает управлять сетевыми интерфейсами. С помощью ifconfig администраторы могут быстро и легко получать информацию о текущем состоянии сетевых интерфейсов и конфигурировать их параметры: IP-адрес, маску, широковещательный адрес, MTU (максимальный размер пакета данных) и другие. Команда также позволяет изменять состояние интерфейсов, включать или выключать их.

**Основные параметры для работы с утилитой ifconfig:**

* **up** — включить интерфейс;
* **down**— выключить интерфейс;
* **(-)arp** — включить или выключить использование протокола ARP для интерфейса;
* **add <адрес/длина префикса>** — добавить IP-адрес для интерфейса;
* **del <адрес/длина префикса>** — удалить IP-адрес интерфейса;
* **media** — установить тип внешнего протокола;



**Команда arp.**

Команда **arp** предназначена для задействования одноименной утилиты, осуществляющей манипуляции с ARP-кэшем сетевого стека Linux. ARP (Address Resolution Protocol или протокол разрешения адресов) является протоколом компьютерных сетей, предназначенным для определения MAC-адресов удаленных узлов на основе их известных IP-адресов. Данный протокол реализуется лишь в сетях с IPv4 (в случае IPv6 аналогичный функционал реализован в рамках протокола ICMP).

**Параметры**

**-a**

Отображать (все) хосты в альтернативном (BSD) стиле

**-e**

Отображать (все) хосты в стиле по умолчанию (Linux)

**-s, --set**

Установить новую запись ARP

**-d, --delete**

Удалить указанную запись

Изображение выглядит как текст, снимок экрана, Шрифт, черный

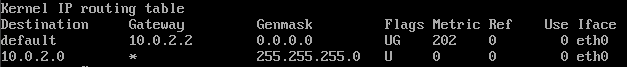
Контент, сгенерированный ИИ, может содержать ошибки.

**Команда route.**

**route** в Linux — это **команда для отображения и настройки таблицы маршрутизации**. Она позволяет изменять таблицы IP-маршрутизации ядра и помогает настраивать статические маршруты к определённым хостам или сетям.

Параметры:

* **del** — удаление маршрута; [1](https://www.it-black.ru/tpost/437xpnfps1-nastroika-setevoi-marshrutizatsii-linux)
* **gw** — шлюз, через который должны достигаться сеть или узел. Задаётся в виде имени узла или точечной записи адреса; [1](https://www.it-black.ru/tpost/437xpnfps1-nastroika-setevoi-marshrutizatsii-linux)
* **mss** — устанавливает значение MTU (максимальную величину пакета) в байтах; [1](https://www.it-black.ru/tpost/437xpnfps1-nastroika-setevoi-marshrutizatsii-linux)



**Команда netstat.**

netstat — это утилита командной строки, которая отображает различную информацию, связанную с сетью, включая:

* **Сетевые соединения (как входящие, так и исходящие)**
* **Таблицы маршрутизации**
* **Статистику сетевых интерфейсов**
* **Соединения маскировки**
* **Членство в мультикастах**
* **Статистику протоколов**

| **Параметр** | **Описание** |
| --- | --- |
| -a | Показывает все прослушивающие порты и активные соединения |
| -t | Отображает TCP-соединения |
| -u | Показывает UDP-соединения |
| -r | Показывает таблицу маршрутизации |
| -i | Показывает статистику сетевых интерфейсов |
| -s | Показывает статистику протоколов |

Изображение выглядит как текст, снимок экрана, Шрифт, черный

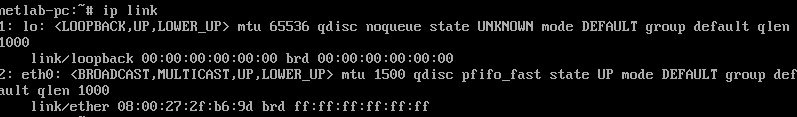
Контент, сгенерированный ИИ, может содержать ошибки.

**Команда ip link.**

Ip link используется для отображения и изменения сетевых интерфейсов.

Параметры:

* **dev**. Определяет имя сетевого интерфейса.
* **up и down**. Изменяют состояние интерфейса на включённый и выключенный.
* **arp on или arp off**. Изменяют флаг NOARP интерфейса.
* **multicast on или multicast off**. Изменяют флаг MULTICAST интерфейса.

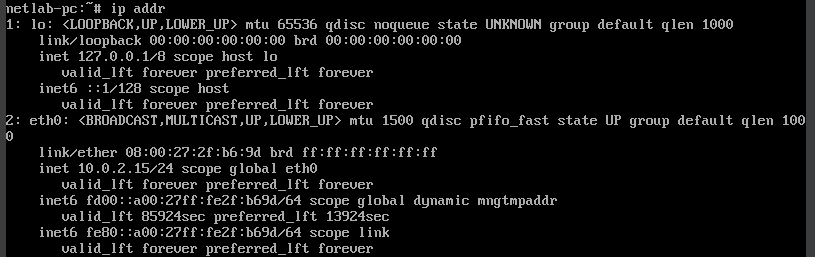


**Команда ip addr.**

Команда ip addr позволяет управлять и просматривать IP-адреса сетевых интерфейсов. Ее альтернативными названиями являются ip address или ip a.

Список доступных подкоманд для данного объекта:

* **add** – Осуществляет добавление нового адреса.
* **show** – Показывает адреса протокола.
* **del** – Осуществляет удаление адреса.
* **flush** – Удаляет адреса в зависимости от заданных критериев.



**Команда ip route.**

Команда ip route предоставляет доступ к таблице IP-маршрутизации и позволяет пользователям настраивать её и выполнять другие важные сетевые задачи, связанные с маршрутизацией.

В качестве подкоманд доступны следующие действия:

* **show** – Показывает таблицу маршрутизации.
* **add**– Осуществляет добавление нового маршрута в таблицу.
* **del** – Осуществляет удаление маршрута из таблицы.
* **change** – Вносит изменения в существующий маршрут.

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Каждая строка в выводе соответствует отдельным маршрутам в таблице.

**Команда ip neigh.**

Ip neigh используется для отображения и управления соседними объектами (таблица ARP)

Параметры:

* **permanent** — запись о соседнем хосте действительна всегда и может быть удалена только административно.
* **noarp** — запись о соседнем хосте действительна. Попыток определения его доступности не производится, но запись может быть удалена по истечению её жизненного цикла.
* **reachable** — запись о соседнем хосте действительна пока не произошёл таймаут при определении доступности хоста.
* **stale** — запись о соседнем хосте действительна, но помечена как подозрительная.

**Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.**

**Команда ss.**

Команда ss – это инструмент командной строки для отображения сетевой статистики. Этот инструмент включен в пакет iproute2 и является более быстрой альтернативой команде netstat.

Параметры:

* **-a, --all**. Выводит все сокеты.
* **-r, --resolve**. Преобразует ip-адреса в имена хостов (домены).
* **-l, --listening**. Выводит сокеты, прослушивающие подключения.
* **-e, --extended**. Выводит подробную информацию о сокетах.



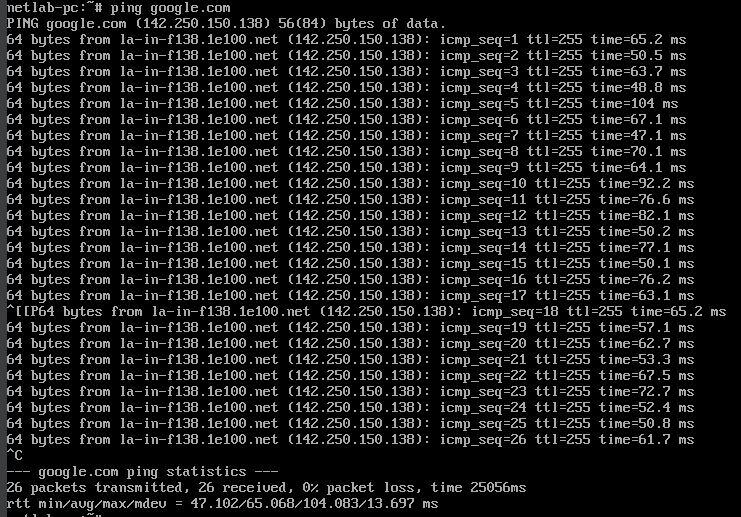
**Команда ping.**

Команда ping – это сетевая утилита для проверки доступности хоста. Она отправляет ICMP-запросы к хосту (компьютеру или серверу) и измеряет время, затраченное на возврат ответа (RTT).

Пинг помогает оценить сетевую задержку между двумя узлами и проверить, доступна ли сеть.

Параметры:

* **-4** — использовать только IPv4 (по умолчанию);
* **-6** — использовать только IPv6;
* **-A** — адаптивный режим, время между отправками пакета адаптируется к времени передачи и приёма пакета, но не меньше чем 200 мс;
* **-b** — разрешить ping широковещательного адреса;



**Команда traceroute.**

Команда traceroute – это инструмент сетевой диагностики, который поддерживается в Linux, macOS и Windows. С её помощью можно отслеживать маршрут, по которому пакеты идут к заданной цели в сети TCP/IP.

С помощью этой команды можно обнаруживать проблемы с маршрутизацией и узкими местами, отображая промежуточные узлы, через которые проходят пакеты при перемещении от отправителя к получателю.

По умолчанию выполнение трассировки составляет 30 прыжков с размером пакета 60 байтов для IPv4 (80 байтов для IPv6).

Параметры:

* **-4** и **-6** — указание протоколов IPv4 и IPv6 соответственно;
* **-I** — использование пакетов ICMP Echo Request вместо UDP;
* **-T** — использование пакетов TCP SYN вместо UDP;
* **-p <порт>** — указание пользовательского номера порта назначения;
* **-q <queries>** — установка количества запросов, отправляемых за один переход;

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

**Команда tracepath.**

Команда tracepath аналогична команде traceroute. Она выявляет маршруты и задержки от отправителя до получателя, отображая маршрутизаторы и сетевые переходы.

Несмотря на то, что traceroute – известная команда с обширными опциями, tracepath представляет собой простой инструмент для отображения сети, доступный на большинстве систем Linux. Для более подробной информации, сравните tracepath и traceroute.

Параметры:

* **-n**. Выводит на экран IP-адреса в числовом виде.
* **-b**. Выводит имена хостов и IP-адреса.
* **-l**. Позволяет установить начальную длину пакета (по умолчанию 65535 для IPv4 и 128000 для IPv6).
* **-m**. Устанавливает максимальное количество переходов (или максимальное TTL) вместо 30.
* **-p**. Указывает начальный порт назначения.
* **-V**. Выводит версию программы и завершается.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

**Команда iftop.**

Команда iftop – это утилита мониторинга сети. С ее помощью можно отслеживать сетевые соединения и использование пропускной способности в режиме реального времени.

Параметр [options] регулирует отображаемую информацию. Кроме того, для мониторинга всего трафика на сетевом интерфейсе необходимы соответствующие права доступа.

Парамеры:

* **-h**. Показывает сводку использования.
* **-n**. Отключает поиск имён хостов.
* **-N**. Отключает преобразование номеров портов в сервисы.
* **-p**. Включает режим, в котором отображается трафик, не проходящий через интерфейс.
* **-P**. Показывает номера портов.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**Команда iperf.**

Команда iperf в Linux предназначена для **тестирования пропускной способности сети** между двумя узлами.

Параметры:

* **-p, --port**. Вручную указать порт на сервере или клиентской машине.
* **-f, --format**. Изменить формат выводимых данных, доступны значения k, m, g, t, K, M, G, T.
* **-i, --interval**. Количество времени между выводом результатов тестирования.
* **-F, --file**. Брать данные для передачи по сети из указанного текстового файла (по умолчанию применяется генератор случайных чисел).

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

**Команда nslookup.**

Команда предназначена для взаимодействия с системой доменных имен.

Параметры:

* **-domain=[доменное имя]**. Позволяет изменить DNS-имя по умолчанию. [1](https://tr-page.yandex.ru/translate?lang=en-ru&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fnslookup-command-in-linux-with-examples%2F)[4](https://phoenixnap.com/kb/nslookup-command)
* **-debug**. Включает отображение отладочной информации. [1](https://tr-page.yandex.ru/translate?lang=en-ru&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fnslookup-command-in-linux-with-examples%2F)[4](https://phoenixnap.com/kb/nslookup-command)
* **-port=[номер порта]**. Указывает номер порта для запросов. По умолчанию nslookup использует порт 53 для запросов DNS.
* **-timeout=[секунды]**. Указывает время, отводимое DNS-серверу для ответа. По умолчанию время ожидания установлено равным нескольким секундам.
* **-type=a**. Позволяет просмотреть информацию о записях DNS A адреса.
* **-type=any**. Просматривает все доступные записи.
* **-type=hinfo**. Отображает информацию об аппаратном обеспечении хоста.

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Контент, сгенерированный ИИ, может содержать ошибки.

**Команда drill.**

Команда также выполняет различные dns-запросы

Параметры:

* **-T**. Трассировка имени от корня вниз. При использовании этой опции аргумент @server не используется.
* **-S**. Следовать за подписью имени до известного ключа или как можно выше в дереве.
* **-I**. Источник адреса для запроса. Адрес должен присутствовать на интерфейсе хоста, на котором запущена drill.
* **-V level**. Быть более verbose. Уровень 5 позволяет увидеть отправленный запрос.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Команда tcpdump.**

Tcpdump используется для анализа пакетов.

Параметры:

* -i <интерфейс>. Указывает сетевой интерфейс для захвата пакетов (например, eth0, wlan0).
* -n. Не разрешает имена хостов, вместо этого отображает IP-адреса.
* -v, -vv, -vvvv. Увеличивает детализацию выходных данных.
* -c <count>. Останавливает захват после получения пакетов count.

Изображение выглядит как текст, снимок экрана, черно-белый, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

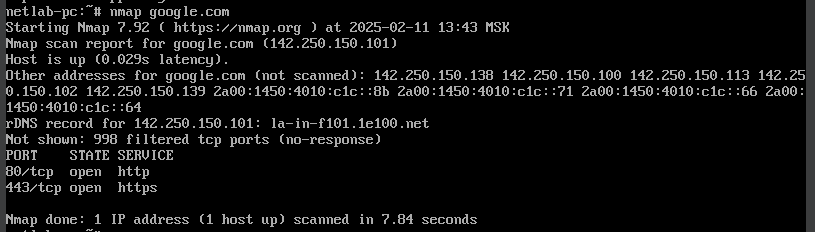
**Команда nmap.**

**Nmap** — это инструмент командной строки Linux для исследования сети и аудита безопасности. Этот инструмент обычно используется хакерами и энтузиастами кибербезопасности, а также сетевыми и системными администраторами. Он используется для следующих целей:

* Информация о сети в режиме реального времени
* Подробная информация обо всех IP-адресах, активированных в вашей сети
* Количество портов, открытых в сети
* Предоставьте список живых хостов
* Сканирование портов, операционной системы и хоста

Параметры:

* **-p**. Указывает, какие порты сканировать. Можно перечислить отдельные порты, разделённые запятыми, или использовать диапазоны, разделённые тире.
* **-sS**. Инициирует скрытое сканирование SYN, которое с меньшей вероятностью будет занесено в журнал.
* **-sV**. Пытается определить версию служб, запущенных на открытых портах.

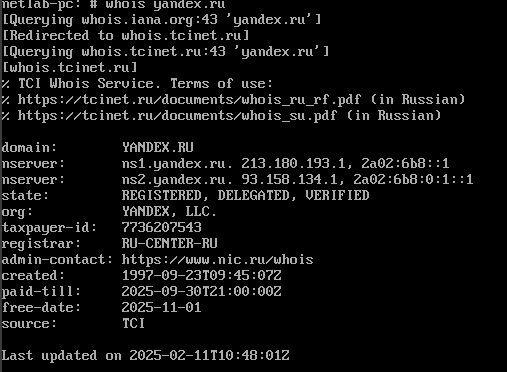


**Команда whois.**

Whois — это утилита командной строки, используемая в системах Linux для получения информации о доменных именах, IP-адресах и сетевых устройствах, зарегистрированных в Интернет-корпорации по присвоению имен и номеров (ICANN). Данные, получаемые с помощью Whois, включают имя и контактную информацию владельца домена или IP-адреса, дату регистрации и срок действия регистрации, регистратора домена и информацию о сервере. Команда Whois может быть очень полезна сетевым администраторам, веб-разработчикам и специалистам по безопасности для выполнения различных задач, таких как проверка подключения к сети или устранение неполадок.

Параметры:

* **-h или --host**. Указывает WHOIS-сервер для использования при поиске.
* **-p или --port**. Указывает номер порта для использования при поиске WHOIS.
* **-r или --raw**. Выводит необработанный ответ WHOIS без форматирования.
* **-6**. Выполняет поиск WHOIS с использованием IPv6.

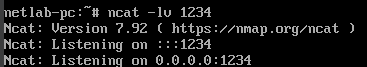


**Команда ncat.**

Ncat — это многофункциональная сетевая утилита, которая считывает и записывает данные в сети из командной строки. Ncat была написана для Nmap Project в качестве сильно улучшенной реинкарнации почтенной программы Netcat. Она использует TCP и UDP для коммуникации и создана быть надёжным движком для мгновенно подключения по сети к другим приложениям и пользователям. Ncat не только работает с IPv4 и IPv6, но и обеспечивает пользователя потенциально безграничным числом вариантов использования.

Параметры:

* **-l** — прослушивание входящих соединений;
* **-u** — использование UDP вместо TCP по умолчанию;
* **-p** — указание исходного порта для использования;
* **-w** — установка таймаута для неактивного соединения;
* **-e** — выполнение заданной команды после подключения;
* **-v** — подробный режим;



**Команда wget.**

Это очень мощная утилита, способная работать по протоколам HTTP, HTTPS и FTP. Кроме того поддерживается работа через прокси. Команда wget linux, может выполнять загрузку файлов даже в фоновом режиме - без участия пользователя, в отличии от большинства веб браузеров.

Параметры:

* **-P, --directory-prefix=ПРЕФИКС**. Указывает каталог, в котором будет сохранён загруженный файл.
* **-O, --output-document-ФАЙЛ**. Указывает имя загружаемого файла.
* **-r, --recursive**. Включает рекурсивную загрузку, которая полезна для загрузки целых веб-сайтов.
* **-np, --no-parents**. Ограничивает загрузку указанным каталогом, предотвращая извлечение файлов из родительских каталогов.

Изображение выглядит как текст, Шрифт, снимок экрана, типография

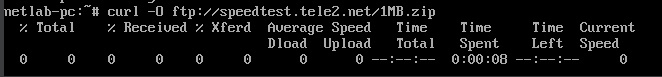
Контент, сгенерированный ИИ, может содержать ошибки.

**Команда curl.**

Среди множества доступных инструментов командной строки `**curl`** выделяется как одна из самых универсальных и мощных утилит. Первоначально разработанная Даниэлем Стенбергом, `**curl`** — это инструмент командной строки и библиотека для передачи данных с помощью URL-адресов. Она поддерживает широкий спектр протоколов, что делает её бесценным инструментом для получения, загрузки и управления данными через Интернет.

Параметры:

* **-O**. Сохраняет скачиваемый файл с его оригинальным именем.
* **-o**. Сохраняет файл с указанным именем.
* **-I**. Получает только HTTP-заголовки ответа сервера.
* **-L**. Следует редиректам. Например, если сайт перенаправляет запрос с [example.com](http://example.com/) на [example.com](https://example.com/), curl с параметром -L автоматически выполнит это перенаправление.
* **-k (SSL)**. Позволяет curl выполнять «небезопасные» SSL-соединения и передачи.



**Команда iptables.**

Команда iptables в Linux — это мощный инструмент, который используется для управления правилами брандмауэра и сетевым трафиком. Она позволяет администраторам настраивать правила, которые помогают фильтровать, преобразовывать или пересылать пакеты. С помощью этой iptables команды вы можете настроить политики безопасности для контроля входящего и исходящего трафика, определить переадресацию портов и реализовать преобразование сетевых адресов (NAT). Это важно для защиты серверов и сетей, поскольку позволяет выборочно разрешать или запрещать определённые типы трафика на основе заданных правил и условий. Такая гибкость делает iptables фундаментальный компонент в сетевых конфигурациях Linux и конфигурациях безопасности.

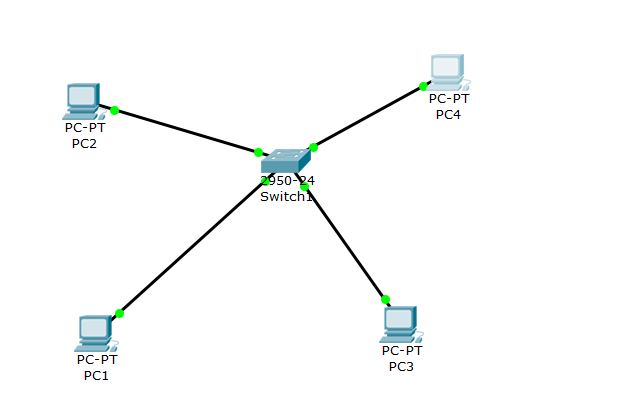
Параметры:

* -A — добавить правило в цепочку; 12
* -C — проверить применяемые правила; 1
* -D — удалить текущее правило; 1
* -I — вставить правило с указанным номером; 1
* -L — вывести правила текущей цепочки; 1
* -S — вывести все активные правила; 1

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Схема сети:



PC-PT PC2:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

PC-PT PC1:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

PC-PT PC3:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

PC-PT PC4:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Ответы на вопросы.

1.как правильно остановить ping

Для остановки ping достаточно надать сочетание клавишь Ctrl+C

2. как получить сетевые параметры компьютера

Сетевые параметры компьютера можно узнать с помощью команы ipconfig. Она предоставляет информацию о таких параметрах, как IP-адреса, маски подсети, шлюзы по умолчанию, серверы DNS и другие.

3. какие сетевые данные можно собрать локально, удаленно

Локально можно собрать Ip и MAC адреса, трафик, информацию о сетевых устройствах. Удаленно можно собрать публичные IP адреса, трафик.

4. как собрать данные о внешних сетях

Можно использовать команды whois, nmap, nslookup, drill, dig.

5. чем логическая топология отличается от физической

* **Физическая топология** описывает **схему прокладки кабеля, расположение узлов и взаимосвязи между ними**. Она определяется возможностями устройств доступа, желаемым уровнем контроля и толерантности к ошибками, а также стоимостью всех необходимых материалов.
* **Логическая топология**, напротив, **описывает поведение сигнала в сети или путь, которым движутся данные в сети от одного устройства к другому**, независимо от их физической взаимосвязи. Она определяет правила передачи данных в существующей среде передачи с гарантированием отсутствия помех, влияющих на корректность передачи данных.

6. зачем нужна схема сети

Схема сети нужна для **отражения взаимодействия всех её компонентов**.

7. что содержит схема сети

Схема сети содержит **все компоненты сети и пути их соединения**.

8. какие стандарты обозначений сетевого оборудования существуют

**Группа стандартов Ethernet IEEE 802.3**

Для подключения клиентских устройств рекомендуется поддержка следующих стандартов:

* IEEE 802.3z (1 Гбит/с);
* IEEE 802.3u (100 Мбит/с);
* IEEE 802.3 (10 Мбит/с).

Для подключения активного сетевого оборудования между собой рекомендуется поддержка следующих стандартов:

* IEEE 802.3z (1 Гбит/с);
* IEEE 802.3ae-2002 (10 Гбит/с);
* IEEE 802.3ba (40 Гбит/с) — да, время пришло.

**Питание по PoE (Power over Ethernet)**

Если для подключения оконечных сетевых устройств нужно active PoE, то необходимо предусмотреть коммутаторы доступа с поддержкой нужных стандартов active PoE. Для наглядности ниже таблица по основным характеристиками active PoE:

|  |  |  |
| --- | --- | --- |
| **Стандарт** | **Мощность на PSE (Power Supply Equipment), Вт** | **Мощность на PD (Powered Device), Вт** |
| IEEE 802.3af (PoE) | До 15.4 | До 12,95 |
| IEEE 802.3at (PoE+) | До 30 | До 25,5 |
| IEEE 802.3bt Type 3 (PoE++) | До 60 | До 51 |

Следует различать две характеристики мощности:

* PSE (Power Supply Equipment) — мощность, которую коммутатор готов выдать на порт;
* PD (Powered Device) — мощность, которая дойдет до оборудования (с учетом потерь).

**VLAN (Virtual Local Area Network)**

Не рекомендуется использовать VLAN 1 для передачи пользовательских данных. Рекомендуется использовать диапазон VLAN'ов с 2 по 1000 включительно.

Деление устройств на VLAN'ы рекомендуется выполнять из соображений получения общего сетевого доступа для последующего применения ACL. В большинстве случаев это существенно упростит задачу конфигурирования межсетевого экрана.