Федеральное государственное бюджетное образовательное учреждение высшего

образования

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

Кафедра вычислительной техники

Лабораторная работа №6

**Отладка сети.**

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

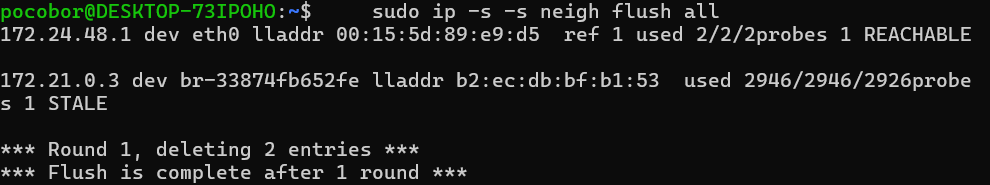
Проверил: Путевская Ирина Валерьевна

Чебоксары, 2025

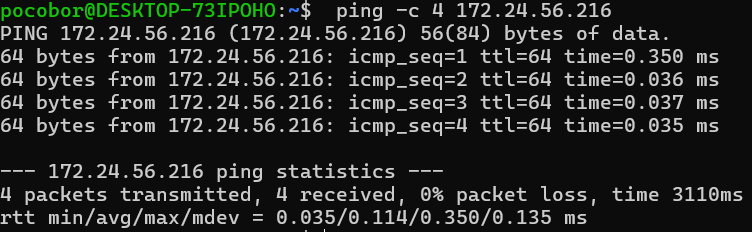
Составить фильтр: ping или  arp

**sudo tcpdump -i any 'arp or icmp' -w arp\_ping.pcap**

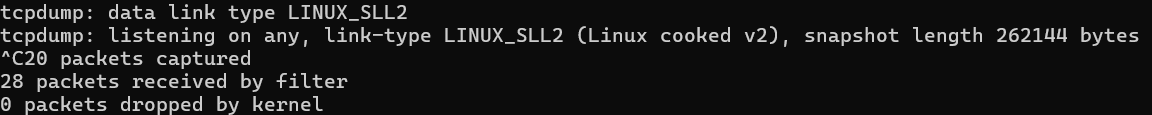
на другом узле очистить arp таблицу



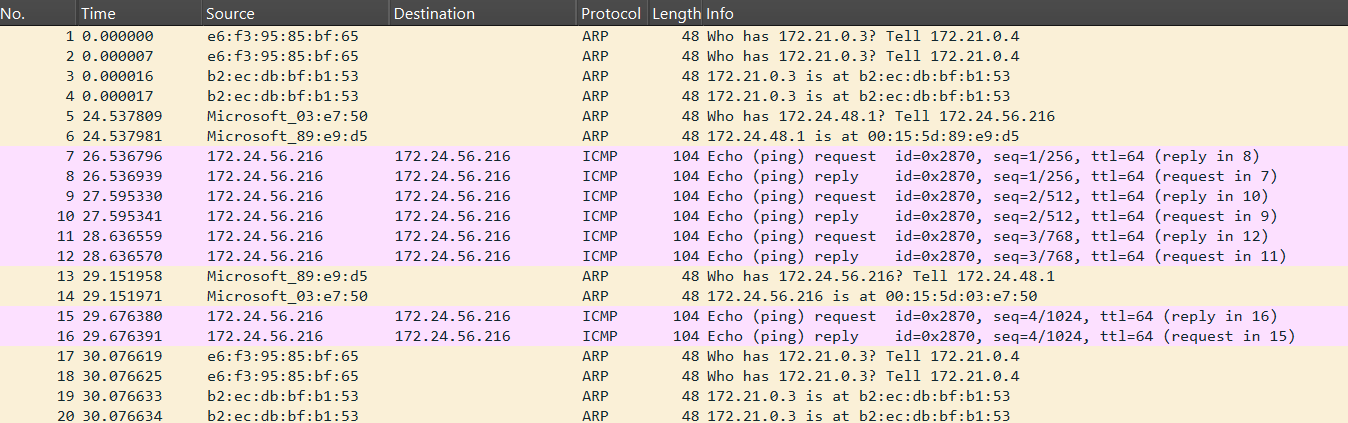
выполнить ping -с 4 первого узла



Результат tcpdump



Apr\_ping.pcap в Wireshark:



Протокол ARP:

1. Request Who has 172.21.0.37? Tell 172.21.0.4

Устройство с IP 172.21.0.4 (MAC: e6:f3:95:85:bf:65) пытается найти MAC-адрес для IP 172.21.0.37

Отправляется широковещательно

1. Ответ 172.21.0.3 is at b2:ec:db:bf:bi:53

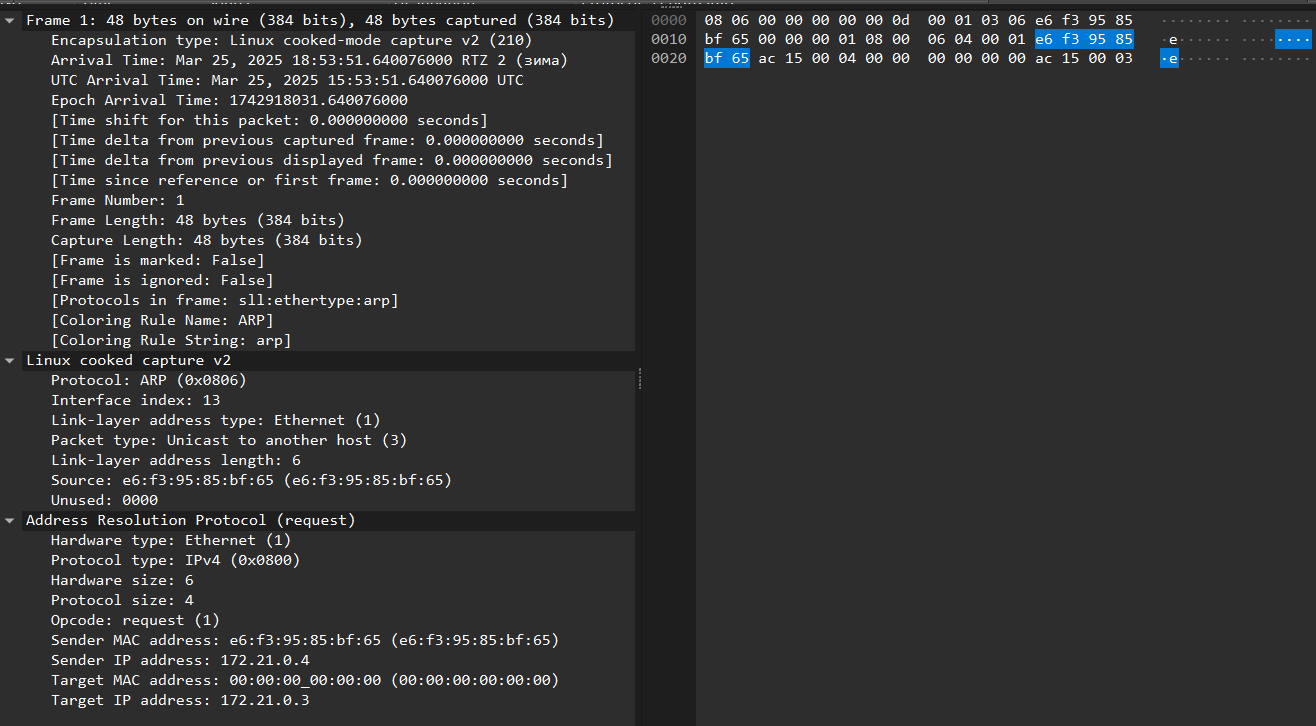
Устройство 172.21.0.3 (MAC: b2:ec:db:bf:bi:53) отвечает, хотя запрашивался 172.21.0.37

1. Request Who has 172.24.48.12? Tell 172.24.56.216

Устройство 172.24.56.216 (MAC: Nicrosoft\_03:e7:50) ищет MAC для 172.24.48.12

1. Ответ 172.24.48.1 is at 00:15:54:89:e9:05

Ответ от 172.24.48.1, хотя запрашивался 172.24.48.12



Байты 00-01 – 08 06 – Протокол АРП

Байты 02-03 – 00 00 – Зарезервировано

Байты 04-07 – 00 00 00 0d – Interface index – 11 (индекс сетевого интерфейса в ядре Linux)

Байты 08-09 – 00 01 – Link-layer address type: Ethernet (1)

Байты 0A – 03– Packet type: unicast to other host

Байты 0B – 06– Link-layer address length: 6 (длина MAC-адреса)

Байты 0C - 11 – E6 F3 95 85 BF 65 – Source: E6:F3:95:85:BF:65

Байты 12 - 13 – 00 00– Зарезервировано

Байты 14 - 15 – 00 01– Hardware type: Ethernet (1)

Байты 16 - 17 – 08 00 – Protocol type: IPv4 (0x0800)

Байты 18 – 06 – Hardware size: 6 (длина MAC)

Байты 19 – 04 – Protocol size: 4 (длина IPv4-адреса)

Байты 1A-1B – 00 01 – Opcode: request (1)

Байты 1C-21– E6 F3 95 85 BF 65 – Sender MAC address: E6:F3:95:85:BF:65

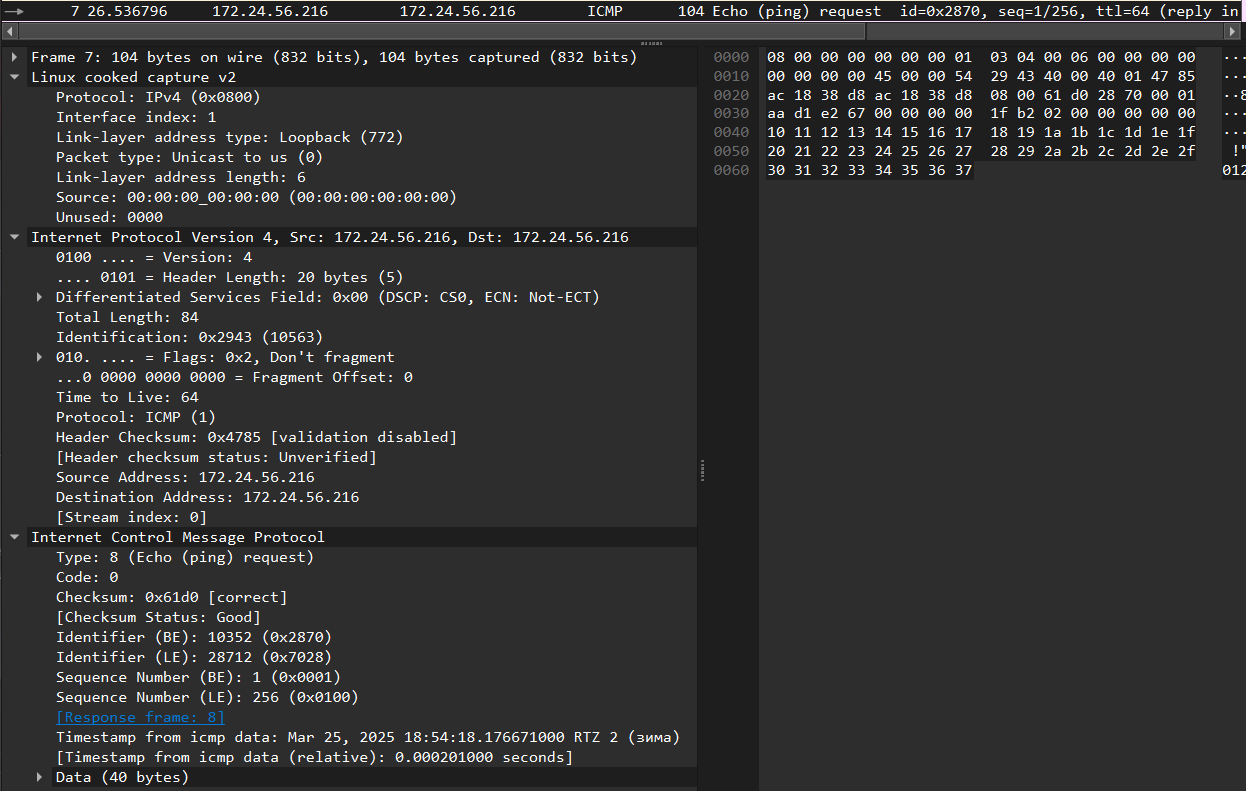
Байты 22-25– Ac 15 00 04– Sender IP address: 172.21.0.4

Байты 26-2B– 00 00 00 00 00 00– Target MAC address: 00:00:00:00:00:00

Байты 2C-2F – Ac 15 00 03– Target IP address: 172.21.0.3

Протокол icmp:

1. Запрос Echo request id=0x2870, seq=1/256, ttl=64
2. Ответ Echo reply id=0x2870, seq=1/256, ttl=64
3. Повторение запрос-ответ



Байты 00-01 – 08 00 – Protocol: IPv4 (0x0800)

Байты 02-03 – 00 00 – Зарезервировано

Байты 04-07 – 00 00 00 01 – Interface index: 1

Байты 08-09 – 03 04 – Link-layer address type: Loopback (772)

Байты 0A – 00 – Packet type: Unicast to us (0)

Байты 0B – 06 – Link-layer address length: 6 (но не используется для loopback)

Байты 0C-11 – 00 00 00 00 00 00 – Source MAC (все нули для loopback)

Байты 12-13 – 00 00 – Зарезервировано

Байты 14 – 45 – Version: 4, Header Length: 5 (5\*4=20 байт)

Байты 15 – 00 – Differentiated Services Field: 0x00

Байты 16-17 – 00 54 – Total Length: 84 (0x54)

Байты 18-19 – 29 43 – Identification: 0x2943 (10563)

Байты 20-21 – 40 00 – Flags: Don't fragment (0x4000)

Байты 22-23 – 40 00 – Fragment Offset: 0

Байты 24 – 40 – Time to Live: 64 (0x40)

Байты 25 – 01 – Protocol: ICMP (1)

Байты 26-27 – 47 85 – Header checksum: 0x4785

Байты 28-31 – AC 18 38 D8 – Source IP: 172.24.56.216

Байты 32-35 – AC 18 38 D8 – Destination IP: 172.24.56.216

Байты 36 – 08 – Type: 8 (Echo request)

Байты 37 – 00 – Code: 0

Байты 38-39 – 61 d0 – Checksum: 0x6140

Байты 40-41 – 28 70 – Identifier: 0x2870 (10352 в BE)

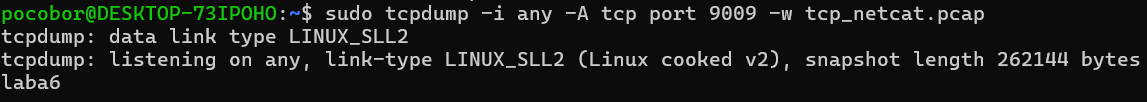
Байты 42-43 – 00 01 – Sequence Number: 0x0801 (1 в BE)

Байты 44-83 – [Данные ping, включая timestamp]

Запустить в фоне netcat в режиме прослушивания



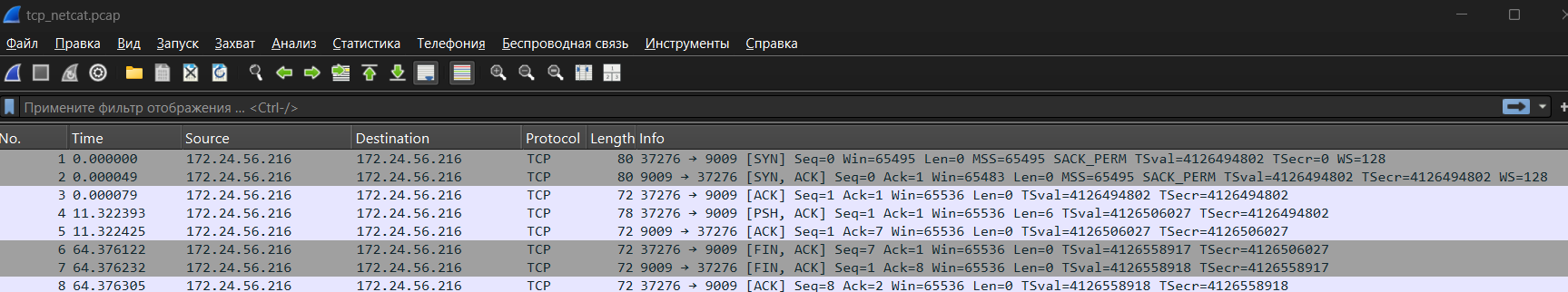
Запустить tcpdump с просмотром содержимого и фильтром:



Подключиться и отправить сообщение



Файл tcp-netcat.pcap в Wireshark



Байты 00-01 – 08 00 – Протокол IPv4

Байты 02-03 – 00 00 – Зарезервировано

Байты 04-07 – 00 00 00 01 – Interface index – 1

Байты 08-09 – 03 04 – Link-layer address type: Loopback (772)

Байты 0A – 00– Packet type: Unicast to us (0)

Байты 0B – 06– Link-layer address length: 6 (длина MAC-адреса)

Байты 0C - 11 – 00 00 00 00 00 00 – Source: 00:00:00\_00:00:00 (00:00:00:00:00:00)

Байты 12 - 13 – 00 00– Зарезервировано

Байты 14 – 45 – 0100 .... = Version: 4

Байт 15 – 00 - Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Байты 16 - 17 – 00 3с – Total Length: 60

Байты 18-19 – 3f a8 – Identification: 0x3fa8 (16296)

Байты 19 – 40 – Identification: 0x3fa8 (16296)

Байт 20 – 00 - ...0 0000 0000 0000 = Fragment Offset: 0

Байт 21 – 40 - Time to Live: 64

Байт 22 – 06 - Protocol: TCP (6)

Байт 23-24 – 31 33 - Header Checksum: 0x3133 [validation disabled]

Байт 25 – 28 – ac 18 38 d8 - Source Address: 172.24.56.216

1. **Пакет 1: [SYN]**
   * Источник: 37276 → Назначение: 9009
   * Флаг: SYN (запрос на соединение)
   * Параметры:
     + Seq=0 (начальный номер последовательности)
     + MSS=65495 (максимальный размер сегмента)
     + SACK\_PERM (поддержка Selective ACK)
     + TSval=4126494802 (метка времени)
2. **Пакет 2: [SYN, ACK]**
   * Ответ от сервера (9009 → 37276)
   * Флаги: SYN + ACK (подтверждение запроса)
   * Ack=1 (подтверждение получения SYN от клиента)
   * Seq=0 (начальный номер последовательности сервера)
3. **Пакет 3: [ACK]**
   * Клиент (37276 → 9009) подтверждает соединение
   * Флаг: ACK
   * Seq=1, Ack=1 (подтверждение SYN сервера)
4. **Пакет 6: [FIN, ACK]** *(начало закрытия соединения)*
   * Источник: 37276 → 9009
   * Флаг: FIN (запрос на завершение) + ACK
   * Seq=7, Ack=8
5. **Пакет 7: [FIN, ACK]** *(ответный FIN)*
   * Сервер (9009 → 37276) также отправляет FIN
   * Seq=1, Ack=8
6. **Пакет 8: [ACK]** *(финальное подтверждение)*
   * Клиент подтверждает закрытие (37276 → 9009)
   * Seq=8, Ack=2

Ответы на вопросы:

1. Что такое и зачем нужен Promiscuous mode?

Promiscuous mode — режим работы сетевого устройства, например, сетевого адаптера, который позволяет ему видеть и «слушать» все сетевые пакеты, проходящие через сеть, даже если они не предназначены для него. В обычном режиме сетевое устройство обрабатывает только пакеты, адресованные ему лично, игнорируя остальные.

Нужен для анализа сетевого трафика или отладка проблем соединения. Например, он помогает администраторам сети мониторить сетевую активность и обнаруживать сетевые атаки.

1. Для чего нужны снифферы?

Снифферы нужны для отслеживания и анализа сетевого трафика. Они помогают понять, как приложения обмениваются данными, и выявить возможные ошибки.

1. Что показывают сниферы?

Снифферы показывают, какие данные циркулируют между приложением и внешними сервисами, и насколько эти данные совпадают с ожидаемыми.

1. Какая структура сообщения перехваченного снифером?

* Служебные данные. Первые 14 байт кадра, которые могут иметь размер от 60 до 1514 байт. Если нужно передать больше данных, они разбиваются на фрагменты и последовательно направляются в сеть.
* Поле структуры type. Оно определяет тип и версию «полезного груза» в кадре.
* Поле структуры data. Оно несёт полезный груз, чаще всего это IP-пакет или ARP-пакет.
* Завершающая секция кадра. Она служит для проверки целостности передаваемых данных и использует код циклического контроля (CRC32 — cyclic redundancy check).

1. Что происходит при подключении узла к сети?

При подключении узла к сети происходит следующее:

* Пользователь вводит доменное имя, присвоенное для конкретного узла в сети. Например, example.com.
* Отправляется запрос на сервер для правильной маршрутизации.
* Доменное имя находится в системе, и компьютер получает связанный с ним IP-адрес.
* Компьютер использует полученный идентификатор для создания подключения.
* При помощи установленной между узлами связи происходит обмен данными, передача информации и выполнение запросов.

1. Как пакет доходит до интернета и как меняется?

Пакет доходит до интернета через маршрутизаторы и коммутаторы. Если хосты расположены в одной подсети с одинаковой маской, информация передаётся напрямую. В противном случае пакет проходит множество промежуточных звеньев до достижения конечной точки.