

# **Отчёт по лабораторной работе 3**

## **Анализ трафика в Wireshark**

Суннатилло Махмудов

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Теоретические сведения по работе</b>	<b>6</b>
<b>3 Выполнение лабораторной работы</b>	<b>8</b>
3.1 MAC-адресация . . . . .	8
3.2 Анализ кадров канального уровня в Wireshark . . . . .	10
3.3 Анализ handshake протокола TCP в Wireshark . . . . .	16
<b>4 Вывод</b>	<b>19</b>

# Список иллюстраций

3.1 Вывод команды ipconfig . . . . .	8
3.2 Вывод команды ipconfig /all . . . . .	9
3.3 Фильтрация ARP и ICMP пакетов . . . . .	11
3.4 Анализ ICMP эхо-запроса . . . . .	12
3.5 Ping ya.ru . . . . .	13
3.6 Анализ ICMP-запроса к внешнему ресурсу . . . . .	13
3.7 HTTP-запросы и ответы . . . . .	14
3.8 DNS-запросы и ответы . . . . .	15
3.9 QUIC-трафик . . . . .	16
3.10 Фиксация TCP пакетов . . . . .	17
3.11 График TCP потока 1 . . . . .	18
3.12 График TCP потока 2 . . . . .	18

# **Список таблиц**

# **1 Цель работы**

Изучение посредством Wireshark кадров Ethernet, анализ PDU протоколов транспортного и прикладного уровней стека TCP/IP.

## 2 Теоретические сведения по работе

Компьютерные сети представляют собой совокупность узлов, объединённых для обмена данными по определённым правилам — протоколам. Для организации взаимодействия используется многоуровневая модель **OSI**, где каждый уровень отвечает за свою часть передачи данных.

**MAC-адресация** применяется на канальном уровне и позволяет однозначно идентифицировать сетевой интерфейс.

MAC-адрес состоит из 48 бит (6 байт), представленных в шестнадцатеричной системе. Первые 24 бита (OUI) определяют производителя устройства, а оставшиеся 24 бита — уникальный идентификатор интерфейса.

MAC-адреса бывают:

- **Уникальные (unicast)** — для обмена между двумя устройствами;

- **Групповые (multicast/broadcast)** — для передачи сразу нескольким получателям;

- **Глобально администрируемые** (назначены производителем);

- **Локально администрируемые** (назначены пользователем или администратором).

**ARP (Address Resolution Protocol)** используется для преобразования IP-адреса в MAC-адрес, необходимый для доставки пакета в пределах локальной сети.

Запрос ARP (Who has ...?) рассыпается всем узлам, а ответ содержит соответствующий MAC-адрес.

**ICMP (Internet Control Message Protocol)** служит для передачи диагностических и служебных сообщений. Наиболее часто используется для проверки до-

ступности узла с помощью команд **ping** (эхо-запросы и эхо-ответы).

**TCP (Transmission Control Protocol)** работает на транспортном уровне и обеспечивает надёжную доставку данных. Характерной особенностью является процедура установления соединения — *трёхстороннее рукопожатие (three-way handshake)*, включающее обмен сегментами SYN, SYN+ACK и ACK.

**UDP (User Datagram Protocol)** — более простой транспортный протокол, не требующий установления соединения. Он используется для приложений, где важна скорость, а не надёжность (например, DNS-запросы, потоковое видео).

**QUIC (Quick UDP Internet Connections)** — современный транспортный протокол, работающий поверх UDP. Он сочетает преимущества UDP (скорость) и TCP (надёжность), а также использует встроенное шифрование на основе TLS 1.3.

**Wireshark** — это инструмент анализа сетевого трафика, позволяющий исследовать кадры и пакеты на разных уровнях модели OSI. С его помощью можно детально рассматривать заголовки протоколов (Ethernet II, ARP, ICMP, TCP, UDP, QUIC) и отслеживать логику обмена данными.

### **3 Выполнение лабораторной работы**

### **3.1 МАС-адресация**

1. С помощью команды **ipconfig** в ОС Windows была выведена информация о текущих сетевых соединениях.

На скриншоте отображены параметры двух сетевых адаптеров, включая IPv4-адрес, маску подсети и основной шлюз.

Рис. 3.1: Вывод команды ipconfig

2. Для получения более подробной информации о сетевом интерфейсе была использована команда:

ipconfig /all

В результате отобразились дополнительные параметры:

- **Описание адаптера** (Realtek Gaming 2.5GbE Family Controller);

- **Физический адрес (MAC): 10-FF-E0-21-D6-B4;**
- DHCP и автоконфигурация;
- Адресация DNS и DHCP-сервера.

```

Адаптер Ethernet Ethernet:

DNS-суффикс подключения . . . . . : 
Описание . . . . . : Realtek Gaming 2.5GbE Family Controller
Физический адрес . . . . . : 10-FF-E0-21-D6-B4
DHCP включен. . . . . : Да
Автонастройка включена. . . . . : Да
Локальный IPv6-адрес канала . . . . . : fe80::b707:d865:dc39:d9f5%17(Основной)
IPv4-адрес. . . . . : 192.168.1.10(Основной)
Маска подсети . . . . . : 255.255.255.0
Аренда получена. . . . . : 24 сентября 2025 г. 9:26:02
Срок аренды истекает. . . . . : 25 сентября 2025 г. 9:26:02
Основной шлюз. . . . . : 192.168.1.1
DHCP-сервер. . . . . : 192.168.1.1
IAID DHCPv6 . . . . . : 101777376 ↴
DUID клиента DHCPv6 . . . . . : 00-01-00-01-30-47-94-1A-10-FF-E0-21-D6-B4
DNS-серверы. . . . . : 192.168.1.1
NetBIOS через TCP/IP. . . . . : Включен

```

Рис. 3.2: Вывод команды ipconfig /all

### 3. Определение структуры MAC-адреса:

MAC-адрес устройства:

10-FF-E0-21-D6-B4

В шестнадцатеричной записи:

10:FF:E0:21:D6:B4

- Первые 3 октета (**10:FF:E0**) – **OUI (Organizationally Unique Identifier)**, определяющий производителя сетевого адаптера (в данном случае – Realtek).
- Последние 3 октета (**21:D6:B4**) – уникальный идентификатор устройства внутри организации-производителя.
- Данный адрес является **индивидуальным (unicast)**, так как первый байт (0x10) имеет младший бит равный **0**.

- Адрес является **глобально администрируемым**, так как второй младший бит первого байта равен **0**.

## 3.2 Анализ кадров канального уровня в Wireshark

1. На устройство был установлен анализатор сетевого трафика **Wireshark**.  
После запуска программы был выбран активный сетевой интерфейс, начался захват пакетов.
2. В консоли Windows с помощью команды **ipconfig** были определены параметры сетевого подключения:
  - IPv4-адрес устройства: 192.168.1.10
  - Маска подсети: 255.255.255.0
  - Шлюз по умолчанию: 192.168.1.1
3. С помощью команды **ping 192.168.1.1** был выполнен обмен эхо-запросами и эхо-ответами с шлюзом по умолчанию.
4. В программе Wireshark был установлен фильтр **arp or icmp**, что позволило отобразить только пакеты ARP и ICMP, сгенерированные командой ping.

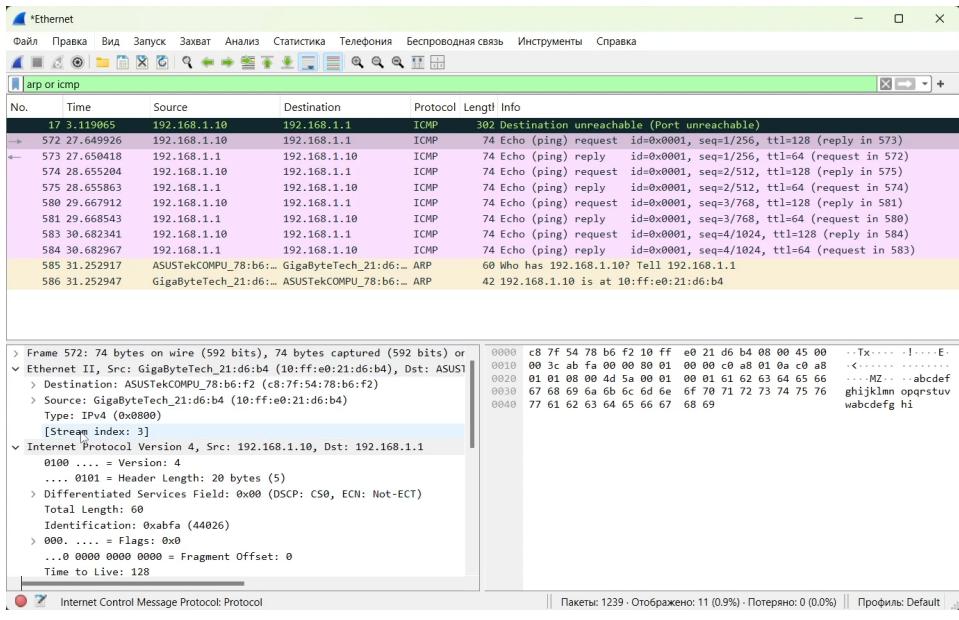


Рис. 3.3: Фильтрация ARP и ICMP пакетов

## 5. Анализ ICMP-пакетов:

### Эхо-запрос (Echo Request):

- Длина кадра: 74 байта
- Тип Ethernet: **Ethernet II**
- MAC-адрес источника: **10:ff:e0:21:d6:b4**
- MAC-адрес шлюза (назначения): **c8:7f:54:78:b6:f2**
- Тип адресов: индивидуальные, глобально администрируемые

### Эхо-ответ (Echo Reply):

- Длина кадра: 74 байта
- Тип Ethernet: **Ethernet II**

- MAC-адрес источника: c8:7f:54:78:b6:f2
- MAC-адрес назначения: 10:ff:e0:21:d6:b4
- Тип адресов: индивидуальные, глобально администрируемые

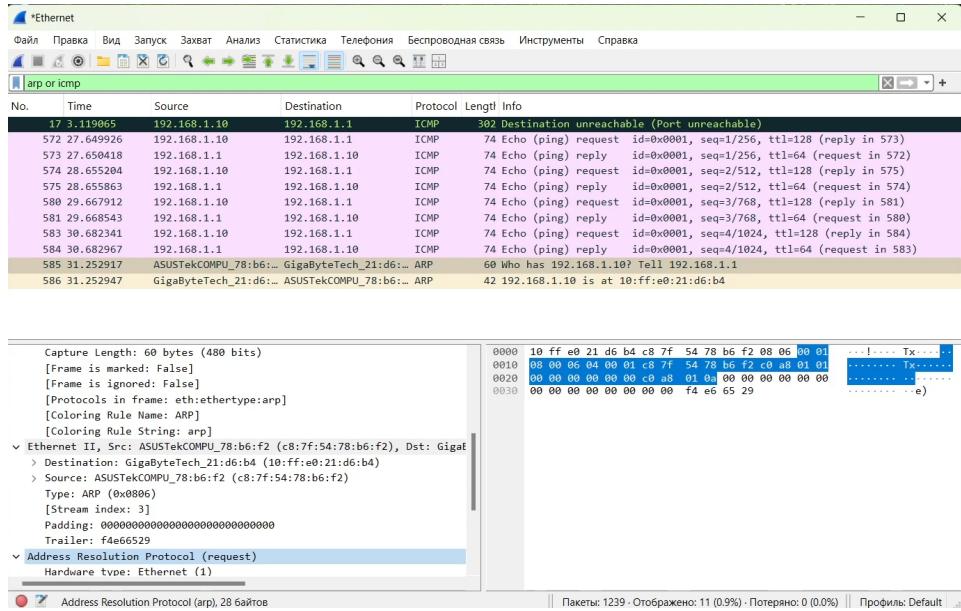


Рис. 3.4: Анализ ICMP эхо-запроса

## 6. Анализ ARP-кадров:

- Запрос: *Who has 192.168.1.10? Tell 192.168.1.1*
- Ответ: *192.168.1.10 is at 10:ff:e0:21:d6:b4*
- В кадрах Ethernet II определены MAC-адреса источника и назначения.

## 7. Для анализа внешнего взаимодействия был выполнен ping доменного имени **ya.ru**.

В процессе обмена пакетами ICMP в Wireshark были зафиксированы как ARP-запросы, так и ICMP-запросы/ответы.

```

PS C:\Users\smahmudov\Documents> ping ya.ru

Обмен пакетами с ya.ru [5.255.255.242] с 32 байтами данных:
Ответ от 5.255.255.242: число байт=32 время=4мс TTL=247

Статистика Ping для 5.255.255.242:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
        (0% потеря)
Приблизительное время приема-передачи в мс:
    Минимальное = 4мсек, Максимальное = 4 мсек, Среднее = 4 мсек
PS C:\Users\smahmudov\Documents>

```

Рис. 3.5: Ping ya.ru

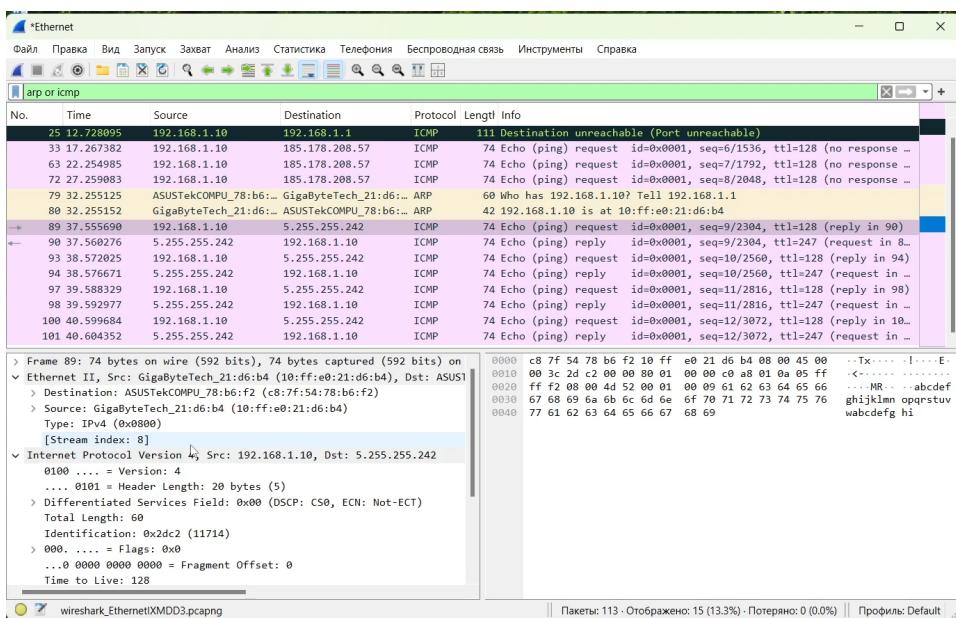


Рис. 3.6: Анализ ICMP-запроса к внешнему ресурсу

## Анализ протоколов транспортного уровня в Wireshark

1. На устройстве был запущен анализатор трафика **Wireshark**.

Для захвата пакетов был выбран активный сетевой интерфейс.

2. В браузере был открыт сайт **CERN** (<http://info.cern.ch/>), работающий по протоколу **HTTP**.

Для генерации большего количества пакетов были выполнены переходы по ссылкам.

### 3. В Wireshark был применён фильтр **http**.

На скриншоте отображены пакеты, относящиеся к HTTP-запросам и ответам:

- Пакет **GET / HTTP/1.1** отправлен от клиента **192.168.1.10** к серверу **188.184.67.127**.
- В ответ сервер передал несколько кадров с кодом **200 OK** и данными HTML-страниц.
- Анализ TCP-полей показал:
  - Исходный порт клиента: **50094**
  - Порт назначения сервера: **80**
  - Используется управление соединением через номера последовательности (Sequence) и подтверждения (ACK).

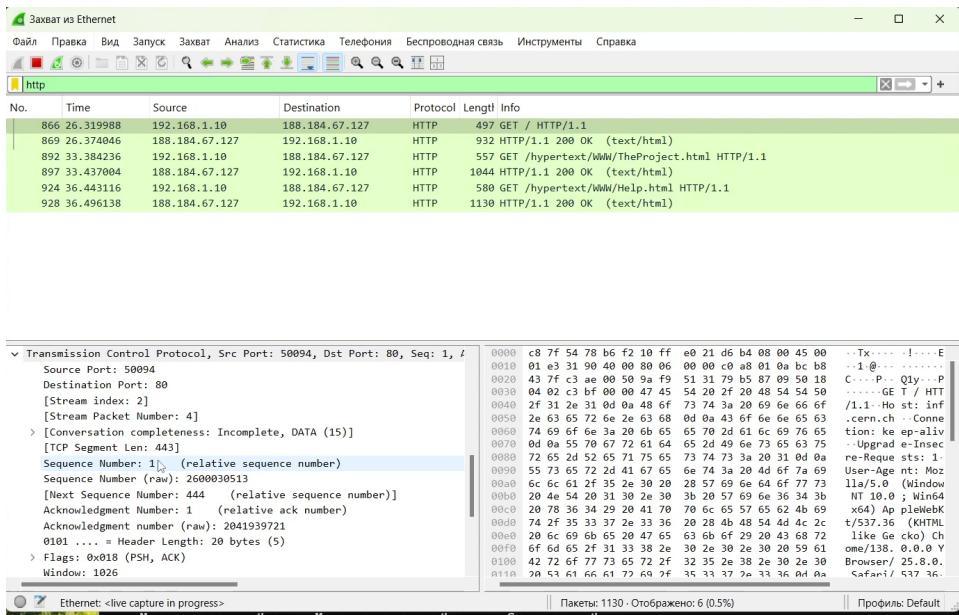


Рис. 3.7: HTTP-запросы и ответы

### 4. Далее был применён фильтр **dns**.

На скриншоте видно взаимодействие клиента (192.168.1.10) с DNS-сервером (192.168.1.1):

- Отправлялись стандартные **DNS-запросы** (A и HTTPS) к домену `slacvx.slac.stanford.edu`.
- Сервер возвращал ответы: как положительные (с IP-адресами), так и отрицательные (`No such name`).
- Транспортным протоколом является **UDP**, порт источника динамический (например 61235), порт назначения фиксированный – 53.

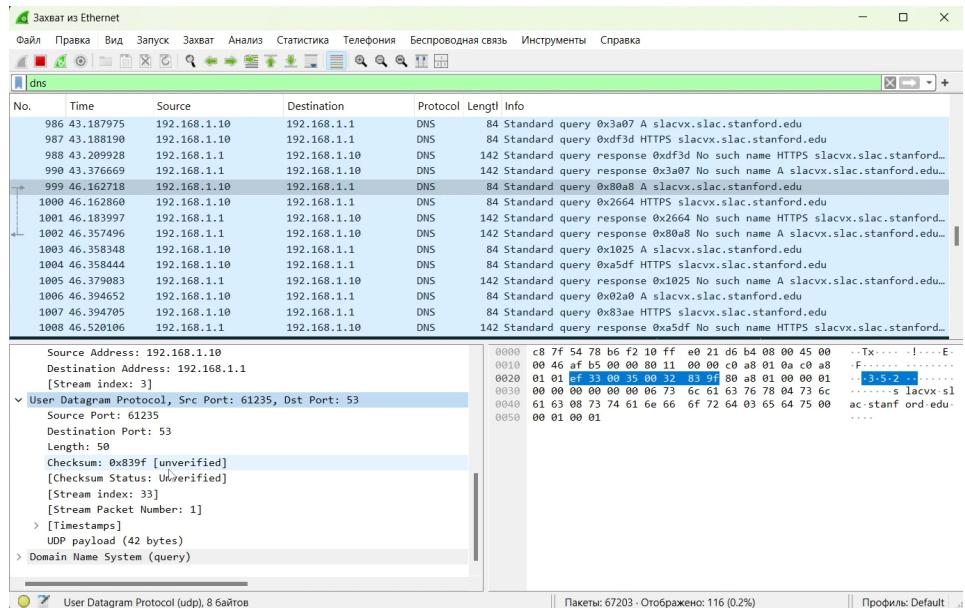


Рис. 3.8: DNS-запросы и ответы

## 5. Для анализа протокола **QUIC** в Wireshark был применён фильтр **quic**.

На скриншоте видны зашифрованные обмены с сервером 173.194.221.100:

- Используются пакеты UDP с портом назначения 443.
- На этапе установления соединения был зафиксирован кадр **QUIC Handshake** с идентификатором соединения (DCID).
- Дальнейшие пакеты передавали **Protected Payload**, так как QUIC обеспечивает встроенное шифрование данных.

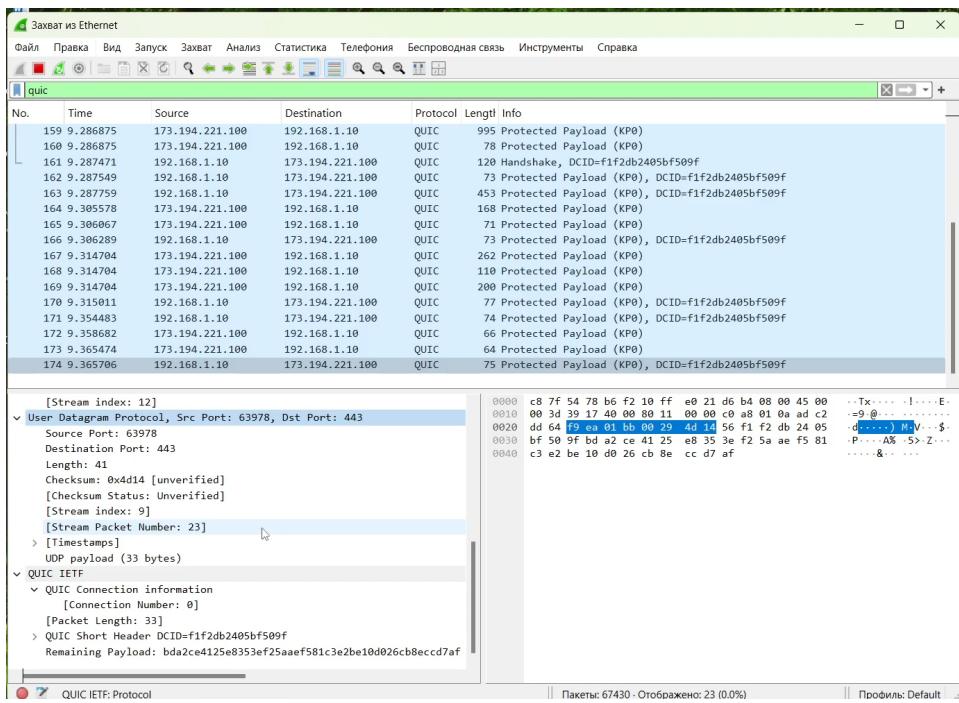


Рис. 3.9: QUIC-трафик

### 3.3 Анализ handshake протокола TCP в Wireshark

- На устройстве был запущен анализатор **Wireshark** и выбран активный сетевой интерфейс для захвата пакетов.
- Для генерации TCP-трафика было установлено соединение с веб-сайтом по протоколу **HTTP**.
- В Wireshark был зафиксирован процесс установки TCP-соединения (трёхстороннее рукопожатие – *three-way handshake*). Этот процесс включает следующие шаги:
  - Клиент отправляет сегмент **SYN** с начальным номером последовательности (Sequence Number).
  - Сервер отвечает сегментом **SYN, ACK**, подтверждая получение первого сегмента и предлагая свой номер последовательности.

- Клиент подтверждает сегментом **ACK**, устанавливая соединение.

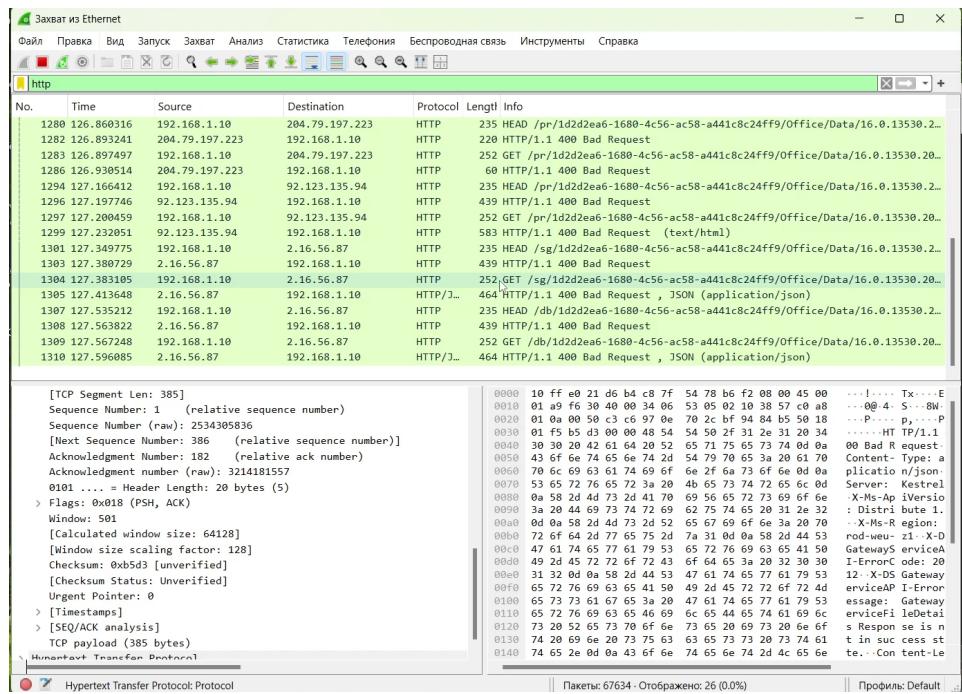


Рис. 3.10: Фиксация TCP пакетов

4. Для анализа эволюции значений полей TCP был использован инструмент

### Статистика → Поток.

На диаграмме потоков виден обмен пакетами между клиентом **192.168.1.10** и сервером.

- Первый пакет (SYN) содержит **Seq = 0**.
- Ответ сервера (SYN, ACK) указывает собственный **Seq = 0**, а также подтверждает номер клиента **Ack = 1**.
- Третий пакет (ACK) подтверждает данные сервера, устанавливая **Ack = 1**.

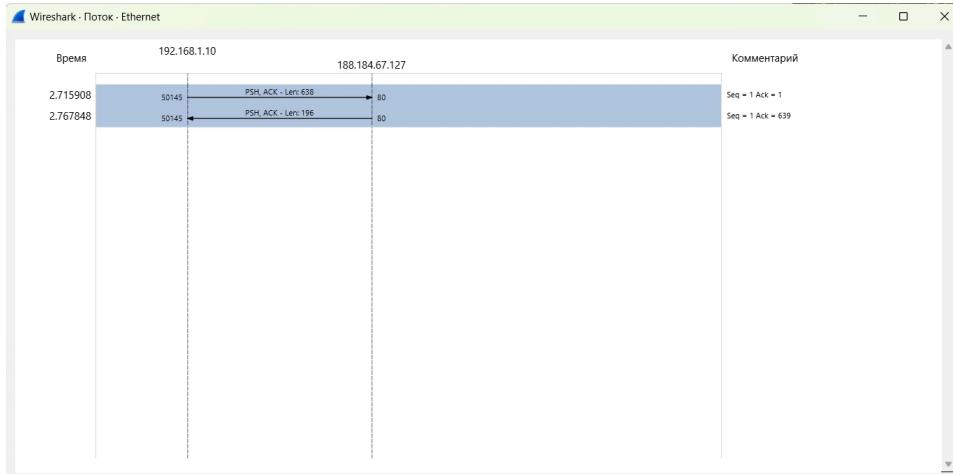


Рис. 3.11: График TCP потока 1

5. На другом примере TCP-потока можно наблюдать детализацию:

- После установки соединения начинают передаваться пакеты с полезной нагрузкой (PSH, ACK).
- Для каждого сегмента фиксируются номера последовательности и подтверждения.
- На графике также видны этапы завершения соединения (FIN, ACK и RST).



Рис. 3.12: График TCP потока 2

## 4 Вывод

В ходе лабораторных работ были изучены и проанализированы основные механизмы работы сетевых протоколов и сервисов.

С помощью утилиты **ipconfig** были определены параметры сетевых интерфейсов и проведён анализ структуры **MAC-адресов**.

С использованием анализатора трафика **Wireshark** были исследованы кадры уровней **ARP** и **ICMP**, подтверждена работа протокола разрешения адресов и обмен эхо-запросами/ответами.

На транспортном уровне были проанализированы протоколы **TCP**, **UDP** и **QUIC**, определены их особенности в передаче данных и работе с соединениями.

Отдельное внимание удалено механизму установления соединения по **TCP** (**three-way handshake**), который был подробно изучен и визуализирован средствами Wireshark.