

# **Отчёт по лабораторной работе 3**

## **Анализ трафика в Wireshark**

Заур Мустафаев

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение работы</b>	<b>6</b>
2.1 Получение информации о сетевых интерфейсах . . . . .	6
2.1.1 Определение MAC-адресов . . . . .	7
2.1.2 Структура MAC-адреса . . . . .	8
2.2 Анализ кадров канального уровня в Wireshark . . . . .	8
2.2.1 Запуск Wireshark и подготовка к анализу . . . . .	8
2.2.2 Определение IP-адреса и шлюза по умолчанию . . . . .	9
2.2.3 Проверка доступности шлюза . . . . .	9
2.2.4 Анализ пакетов ICMP и ARP в Wireshark . . . . .	10
2.2.5 Анализ ICMP-запроса (Echo Request) . . . . .	11
2.2.6 Анализ ICMP-ответа (Echo Reply) . . . . .	12
2.2.7 Анализ ARP-запросов . . . . .	12
2.3 Анализ протоколов транспортного уровня в Wireshark . . . . .	13
2.3.1 Анализ HTTP (TCP) . . . . .	13
2.3.2 Анализ DNS (UDP) . . . . .	15
2.3.3 Анализ QUIC (UDP) . . . . .	16
2.4 Анализ handshake протокола TCP в Wireshark . . . . .	18
2.4.1 Визуализация TCP Handshake . . . . .	19
<b>3 Вывод</b>	<b>21</b>

# Список иллюстраций

2.1 Вывод команды ipconfig . . . . .	6
2.2 Ping шлюза . . . . .	9
2.3 Ping альтернативного шлюза . . . . .	10
2.4 HTTP-трафик . . . . .	14
2.5 Детализация TCP-пакета . . . . .	15
2.6 DNS-запросы и ответы . . . . .	16
2.7 QUIC-трафик . . . . .	17
2.8 TCP пакеты в Wireshark . . . . .	18
2.9 График TCP потока . . . . .	19

# **Список таблиц**

# **1 Цель работы**

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

## 2 Выполнение работы

### 2.1 Получение информации о сетевых интерфейсах

- Для просмотра параметров сетевых адаптеров была использована команда **ipconfig** в операционной системе Windows.

Команда без параметров отображает общую информацию о подключениях: IPv4-адрес, маску подсети, основной шлюз, локальные IPv6-адреса.

```
Адаптер Ethernet 2:
DNS-сервер по дому . . . . . :
Локальный IPv6-адрес сканала . . . . : fe80::bae0:cc2e:9d1e:b2d4%19
IPv4-адрес . . . . . : 192.168.56.1
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . :

Адаптер беспроводной локальной сети По дому по локальной сети * 1:
Состояние среды . . . . . : Среда передачи не до ступна.
DNS-сервер по дому . . . . . :

Адаптер беспроводной локальной сети По дому по локальной сети * 2:
Состояние среды . . . . . : Среда передачи не до ступна.
DNS-сервер по дому . . . . . :

Адаптер беспроводной локальной сети Беспроводная сеть:
DNS-сервер по дому . . . . . :
Локальный IPv6-адрес сканала . . . . : fe80::da3b:4057:9ef4:1e28%7
IPv4-адрес . . . . . : 192.168.101.83
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . : 192.168.101.1

C:\Users\zmustafaev>
```

Рис. 2.1: Вывод команды ipconfig

- В результате можно выделить несколько сетевых интерфейсов:

- **Ethernet Ethernet 2** с адресом 192.168.56.1, относящийся к виртуальному адаптеру.

- **Беспроводная сеть** с адресом 192.168.101.83, использующая основной шлюз 192.168.101.1.
  - Дополнительные интерфейсы отображаются как “среда передачи недоступна”, что указывает на их неактивность.
3. Для получения более подробных сведений можно использовать ключ **/all**, который выводит расширенную информацию, включая:
- **MAC-адрес (физический адрес)** сетевого интерфейса.
  - DHCP-настройки (включение/отключение).
  - DNS-серверы.
  - Срок аренды IP-адреса.

Это позволяет не только увидеть текущую конфигурацию TCP/IP, но и определить уникальные идентификаторы сетевых карт.

### **2.1.1 Определение MAC-адресов**

1. MAC-адреса были получены из расширенного вывода `ipconfig /all`. Каждый сетевой адаптер имеет собственный физический адрес в формате шестнадцатеричной записи (например: D4-3B-05-7A-9E-F4).
2. Данный адрес используется для уникальной идентификации сетевого интерфейса в локальной сети и отображается в разделе “Физический адрес”.

## 2.1.2 Структура MAC-адреса

1. **MAC-адрес** состоит из 48 бит (6 байт), обычно представляется как шесть пар шестнадцатеричных чисел.

Пример: D4-3B-05-7A-9E-F4.

- Первые 3 байта (D4-3B-05) — это **OUI (Organizationally Unique Identifier)**, который указывает на производителя устройства.
- Последние 3 байта (7A-9E-F4) — это уникальный идентификатор интерфейса, назначенный производителем.

2. Определение типа адреса:

- Если первый бит первого байта равен **0**, адрес является **индивидуальным (unicast)**.
- Если равен **1**, адрес является **групповым (multicast)**.
- Второй бит указывает на тип администрирования: **0 — глобально администрируемый, 1 — локально администрируемый**.

В данном случае адрес D4-3B-05-7A-9E-F4 начинается с D4 (в двоичной системе 11010100). Первый бит равен 1 → это **индивидуальный (unicast)** адрес. Второй бит равен 0 → адрес **глобально администрируемый**.

## 2.2 Анализ кадров канального уровня в Wireshark

### 2.2.1 Запуск Wireshark и подготовка к анализу

1. На устройство была установлена программа **Wireshark** для анализа сетевых пакетов.

2. В интерфейсе программы выбран активный сетевой адаптер (беспроводная сеть). После выбора начался захват трафика.

### 2.2.2 Определение IP-адреса и шлюза по умолчанию

1. С помощью команды **ipconfig** был определён IP-адрес устройства 192.168.101.83 и шлюз по умолчанию 192.168.101.1.

### 2.2.3 Проверка доступности шлюза

1. Для проверки связи был выполнен запрос ping 192.168.101.1.

Ответа от шлюза получено не было — все пакеты потеряны (100%).

```
C:\Users\zmustafaev>ping 192.168.101.1

Обмен пакетами с 192.168.101.1 по с 32 байтами данных:
Превышен интервал ожидания для запроса.

Статистика Ping для 192.168.101.1:
Пакетов: отправлено = 4, получено = 0, потеряно = 4
(100% потерян)

C:\Users\zmustafaev>ping 192.168.1.1

Обмен пакетами с 192.168.1.1 по с 32 байтами данных:
Превышен интервал ожидания для запроса.

Статистика Ping для 192.168.1.1:
Пакетов: отправлено = 4, получено = 0, потеряно = 4
(100% потерян)

C:\Users\zmustafaev>ping ya.ru
```

Рис. 2.2: Ping шлюза

2. Аналогичная проверка с IP-адресом 192.168.1.1 также завершилась неудачно. Видимо особенность защиты роутера.

```
C:\Users\zmustafaev>ping 192.168.1.1

Обмен пакетами с 192.168.1.1 по с 32 байтами данных:
Превышение интервала ожидания для запроса.

Статистика Ping для 192.168.1.1:
Пакетов: отправлено = 4, получено = 0, потеряно = 4
(100% потерян)

C:\Users\zmustafaev>ping ya.ru

Обмен пакетами с ya.ru [5.255.255.242] с 32 байтами данных:
Ответ от 5.255.255.242: чисто байт =32 время =4мс TTL=247
Ответ от 5.255.255.242: чисто байт =32 время =7мс TTL=247
Ответ от 5.255.255.242: чисто байт =32 время =5мс TTL=247
Ответ от 5.255.255.242: чисто байт =32 время =4мс TTL=247

Статистика Ping для 5.255.255.242:
Пакетов: отправлено = 4, получено = 4, потеряно = 0
(0% потеря)
Привилегированное время приема-передачи в мс:
Минимальное = 4мсек, Максимальное = 7мсек, Среднее = 5мсек

C:\Users\zmustafaev>
```

Рис. 2.3: Ping альтернативного шлюза

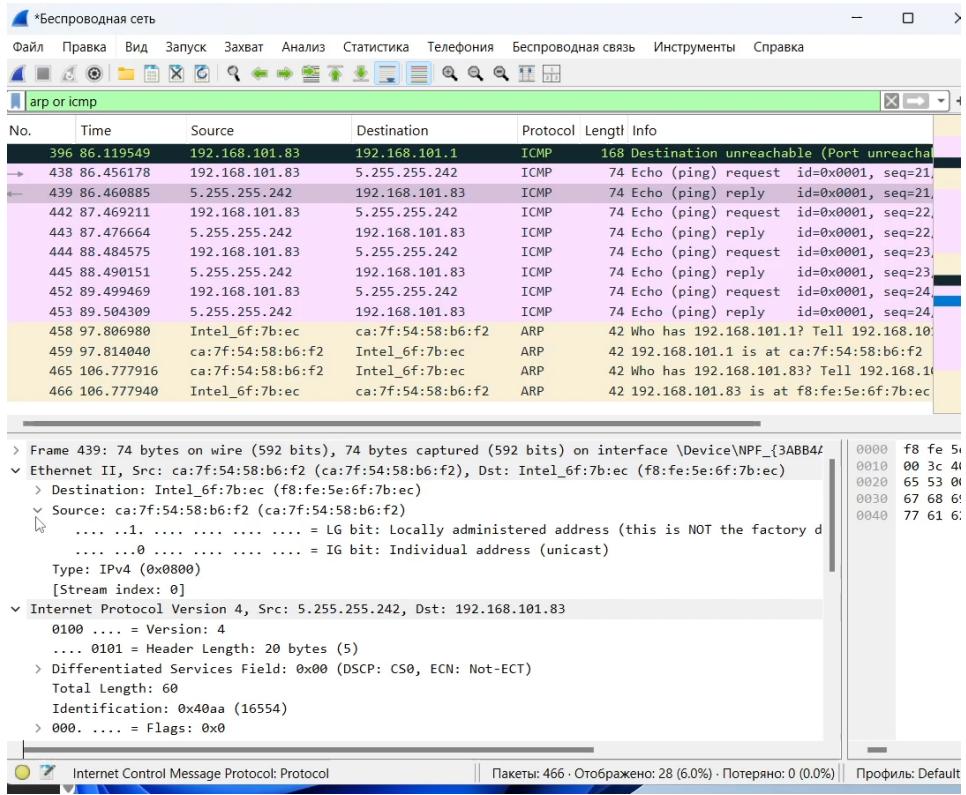
3. При выполнении ping ya.ru соединение успешно установлено: получены ответы от сервера 5.255.255.242 с временем отклика от 4 до 7 мс.

#### 2.2.4 Анализ пакетов ICMP и ARP в Wireshark

1. В Wireshark применён фильтр **arp or icmp**, что позволило отобразить только ICMP и ARP пакеты.

В списке пакетов видно:

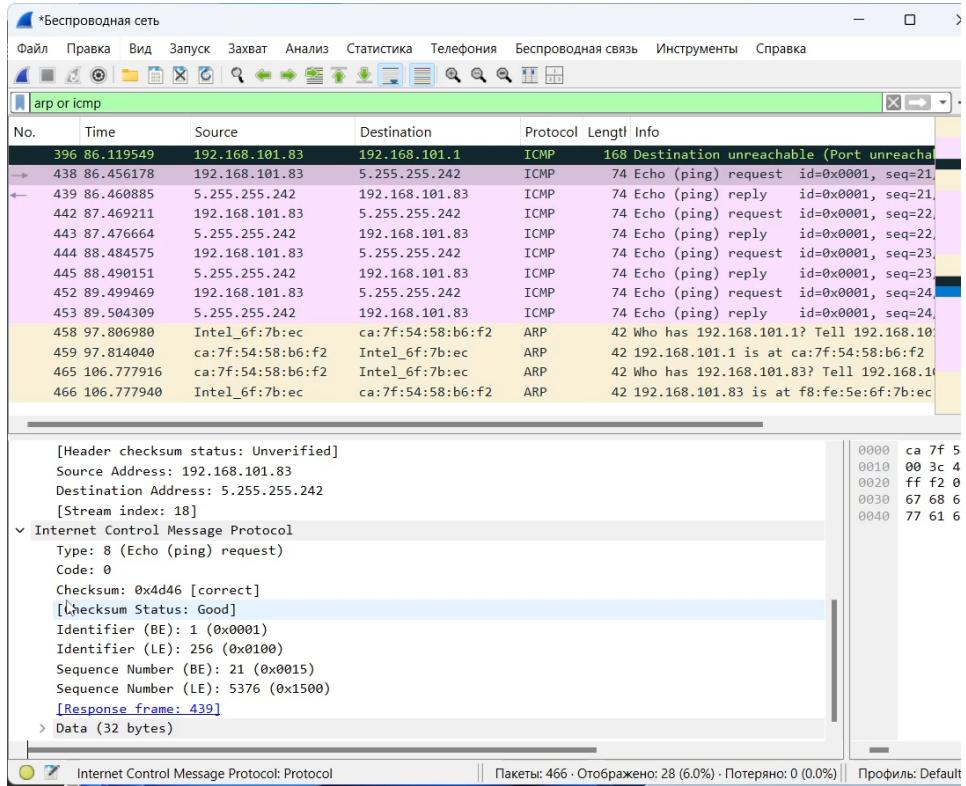
- ICMP-запросы и ICMP-ответы (ping).
- ARP-запросы для определения MAC-адреса шлюза.



## 2.2.5 Анализ ICMP-запроса (Echo Request)

1. Выбран кадр ICMP-запроса:

- Длина кадра – **74 байта**.
- Тип Ethernet – **Ethernet II**.
- MAC-адрес источника: **ca:7f:54:58:b6:f2**.
- MAC-адрес назначения: **f8:fe:5e:6f:7b:ec**.
- Адрес является **индивидуальным (unicast)** и **локально управляемым**.



## 2.2.6 Анализ ICMP-ответа (Echo Reply)

1. В ответном ICMP-кадре:

- MAC-адрес источника совпадает с адресом шлюза.
- MAC-адрес назначения – адрес устройства ca:7f:54:58:b6:f2.
- Адреса также являются **индивидуальными (unicast)**.

(анализ сделан по структуре ICMP-пакетов в Wireshark)

## 2.2.7 Анализ ARP-запросов

![ARP-запрос в Wireshark] (04.png){ #fig:006 width=80% }

1. При работе сети были зафиксированы пакеты ARP:

- Запрос **Who has 192.168.101.1? Tell 192.168.101.83.**
  - Ответ **192.168.101.1 is at f8:fe:5e:6f:7b:ec.**
2. В заголовке Ethernet II содержатся:
- **MAC-адрес отправителя:** ca:7f:54:58:b6:f2.
  - **MAC-адрес получателя:** f8:fe:5e:6f:7b:ec.
  - Тип протокола: ARP (0x0806).

## **2.3 Анализ протоколов транспортного уровня в Wireshark**

### **2.3.1 Анализ HTTP (TCP)**

1. В браузере был открыт сайт, работающий по протоколу **HTTP**.
2. В Wireshark применён фильтр **http**, что позволило выделить только HTTP-запросы и ответы.

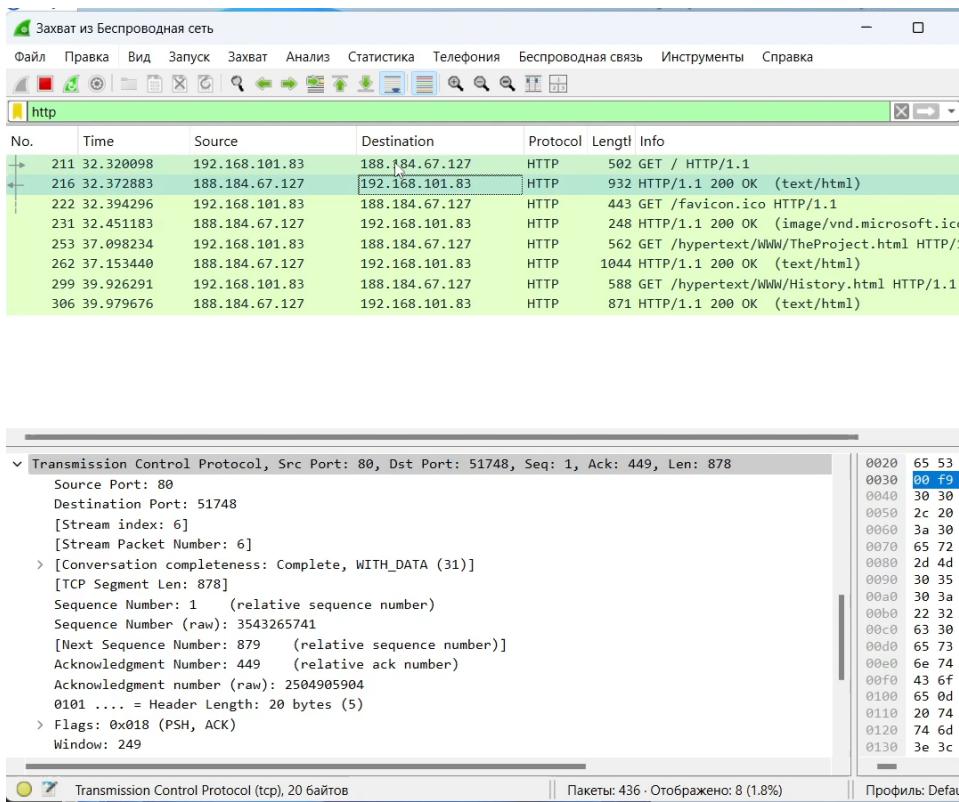


Рис. 2.4: HTTP-трафик

3. На захваченных пакетах видно:

- Запрос GET / HTTP/1.1 от клиента (адрес 192.168.101.83) к серверу (188.184.67.127).
- Ответ сервера HTTP/1.1 200 OK с передачей данных (например, страницы HTML или файла favicon.ico).
- Передача данных выполняется поверх TCP.
- На транспортном уровне в TCP-заголовках отображаются: номера последовательности (Sequence Number), подтверждения (Acknowledgment Number), а также порт назначения – 80.

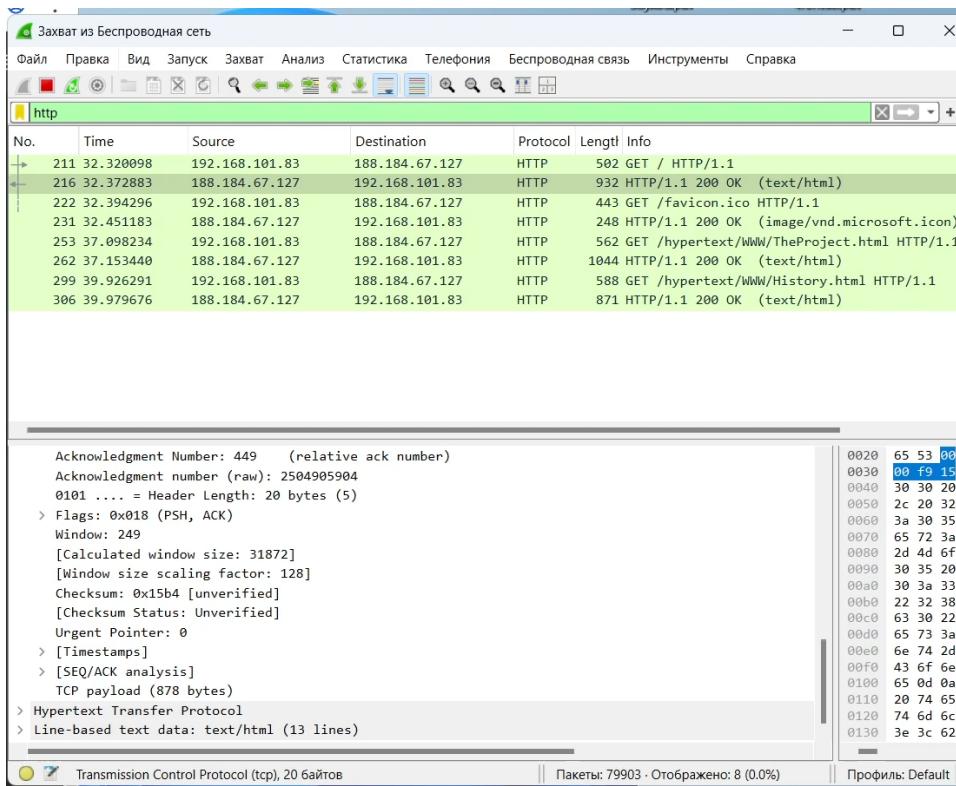


Рис. 2.5: Детализация TCP-пакета

### 2.3.2 Анализ DNS (UDP)

1. Для анализа DNS-запросов был применён фильтр **dns**.
2. В захваченных пакетах видно:
  - Клиент (адрес 192.168.101.83) отправляет DNS-запрос на сервер (192.168.101.1) по **UDP-порту 53**.
  - Запрос имеет тип **Standard query** (например, на домены `yandex.ru`, `cern.ch`).
  - Сервер отвечает сообщением **Standard query response**, где возвращается IP-адрес запрашиваемого ресурса.

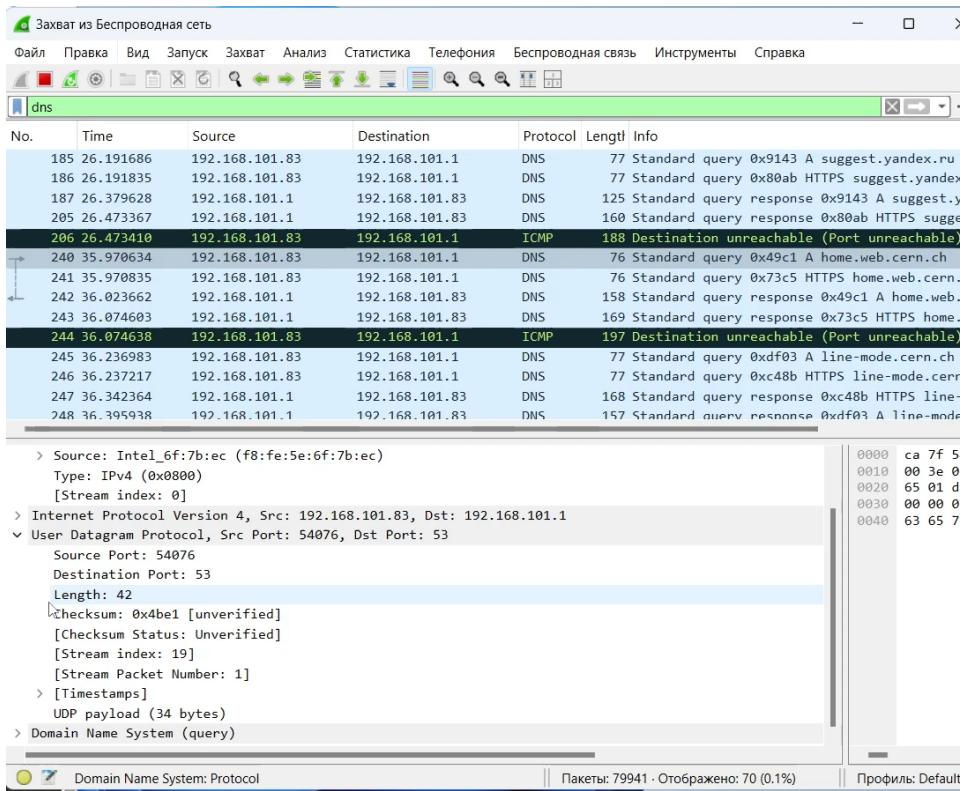


Рис. 2.6: DNS-запросы и ответы

3. На транспортном уровне видно, что протокол DNS использует **UDP**, в заголовке отображаются порты источника и назначения, а также длина полезной нагрузки.

### 2.3.3 Анализ QUIC (UDP)

1. Для анализа работы протокола **QUIC** был применён фильтр **quic**.
2. В захваченных данных отображается взаимодействие с сервером по защищённому каналу через порт **443**.

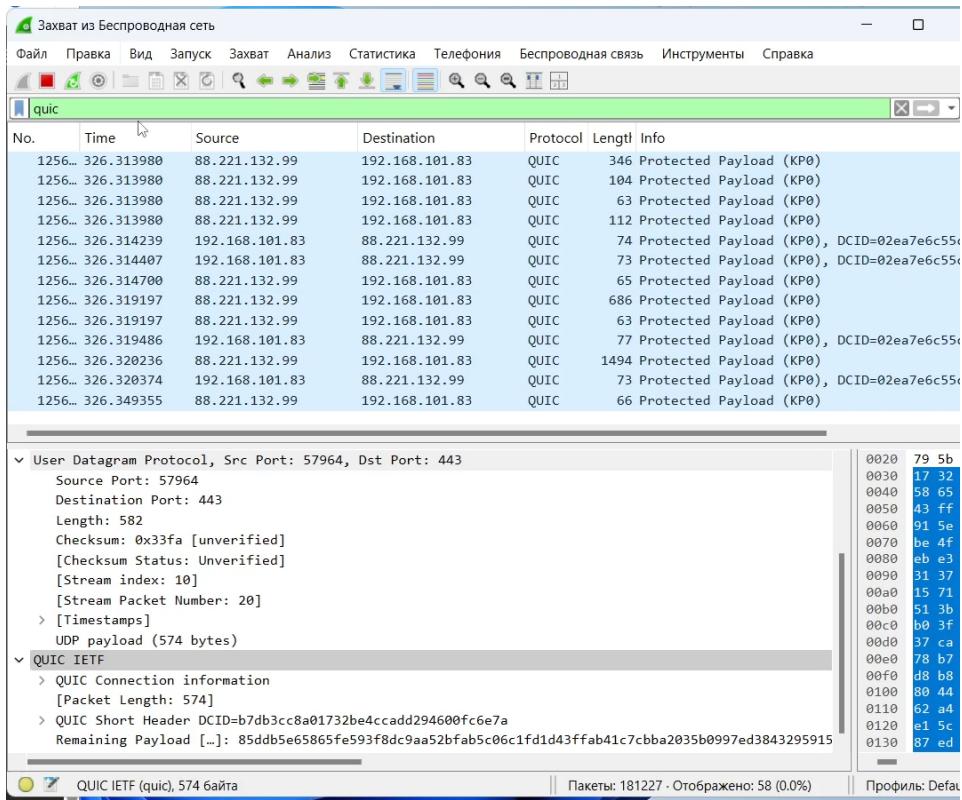


Рис. 2.7: QUIC-трафик

### 3. Особенности QUIC:

- Работает поверх **UDP**, но обеспечивает функции, схожие с TCP (надёжная доставка, контроль последовательности).
- Используется для работы **HTTP/3**.
- В заголовках Wireshark можно наблюдать:
  - Source Port (порт источника).
  - Destination Port (порт назначения = 443).
  - Длину пакета.

- Идентификатор соединения (Connection ID).

## 2.4 Анализ handshake протокола TCP в Wireshark

1. В захваченных данных с фильтром **tcp** отображаются пакеты установления соединения.

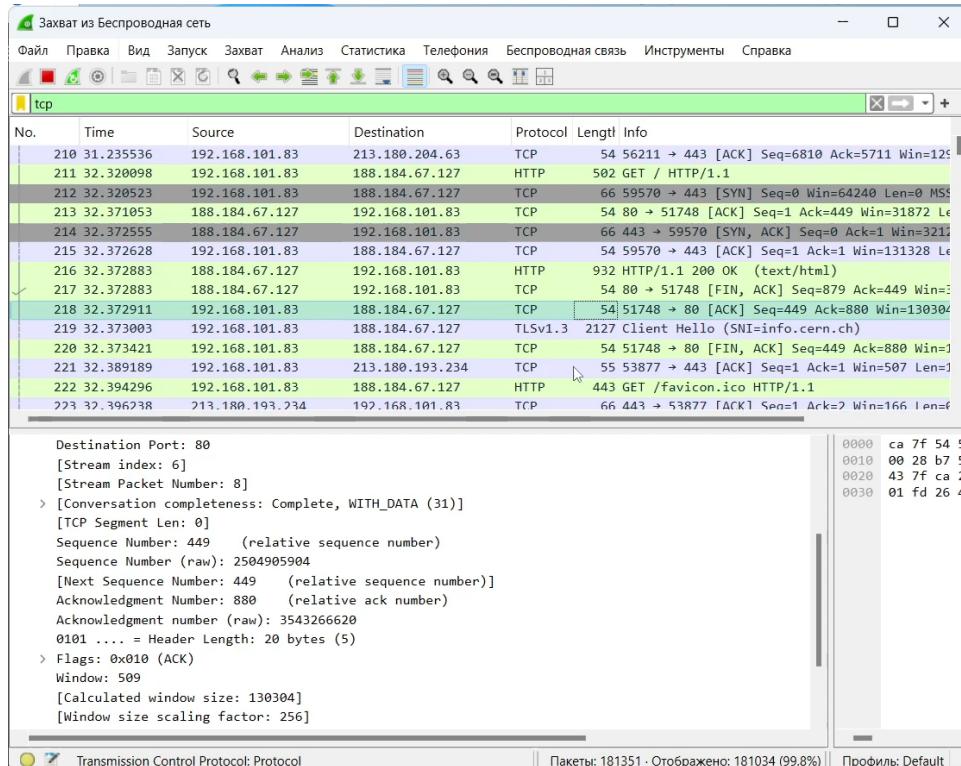


Рис. 2.8: TCP пакеты в Wireshark

2. Процесс установки TCP-сессии происходит в три этапа:

- **SYN** — клиент (192.168.101.83) отправляет пакет с установленным флагом SYN серверу (213.180.204.63) для инициализации соединения. В пакете указывается начальный номер последовательности (**Sequence Number**).
- **SYN, ACK** — сервер отвечает пакетом, в котором установлен флаг SYN и ACK. Сервер подтверждает получение первого SYN от клиента и сообщает о своем начальном номере последовательности.

щает свой начальный номер последовательности.

- **ACK** – клиент отправляет пакет с подтверждением (ACK) для завершения установки соединения. После этого соединение считается установленным, и начинается передача данных.
3. На скриншоте видно, что далее идут пакеты PSH, ACK, которые содержат полезную нагрузку (данные HTTP).

#### 2.4.1 Визуализация TCP Handshake

1. В меню **Статистика → График потока** был построен график TCP Flows.

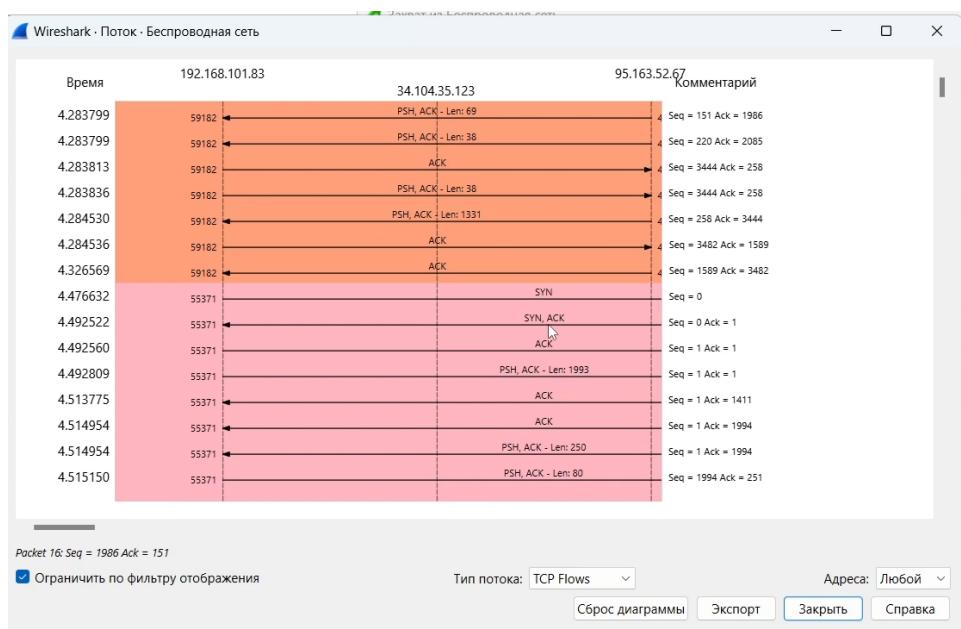


Рис. 2.9: График TCP потока

2. Диаграмма показывает последовательный обмен пакетами между клиентом (192.168.101.83) и сервером (95.163.52.67):

- Сначала выполняются SYN → SYN, ACK → ACK.
- Затем идёт передача данных с использованием PSH, ACK.

- Каждый пакет сопровождается подтверждением (**Acknowledgment**), что подтверждает надёжную доставку.

## **3 Вывод**

В ходе работы был проанализирован процесс установления TCP-соединения с помощью Wireshark. Рассмотрено трёхстороннее рукопожатие ( $\text{SYN} \rightarrow \text{SYN/ACK} \rightarrow \text{ACK}$ ) и подтверждена надёжность протокола за счёт обмена номерами последовательностей и подтверждений. Построен график потока, наглядно отражающий этапы установления соединения и передачу данных.