Отчёт по лабораторной работе 3

Анализ трафика в Wireshark

Заур Мустафаев

Содержание

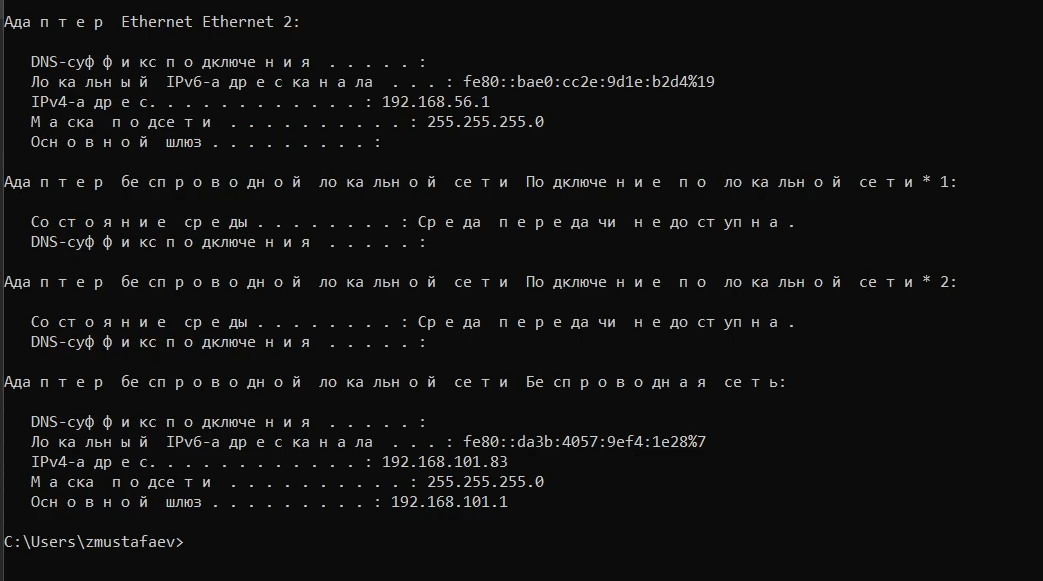
# 1 Цель работы

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

1. 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.

* 
* Рис. 1: Вывод команды ipconfig

1. 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.
2. 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

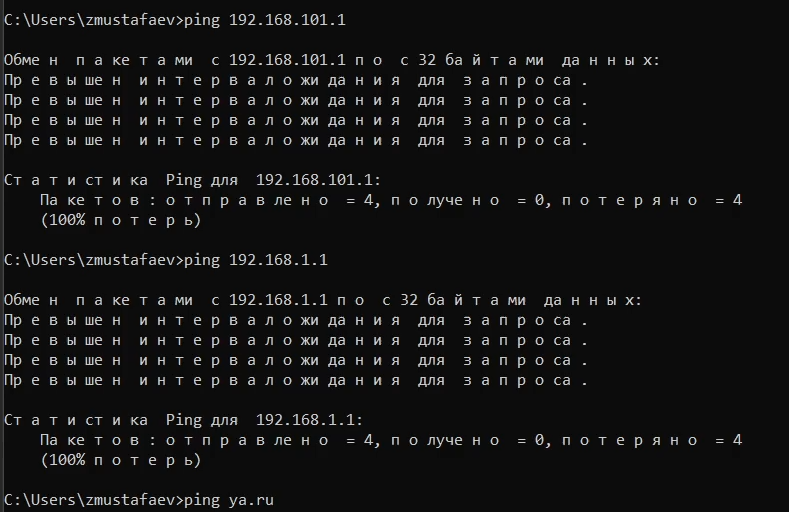
1. На устройство была установлена программа **Wireshark** для анализа сетевых пакетов.
2. В интерфейсе программы выбран активный сетевой адаптер (беспроводная сеть). После выбора начался захват трафика.

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

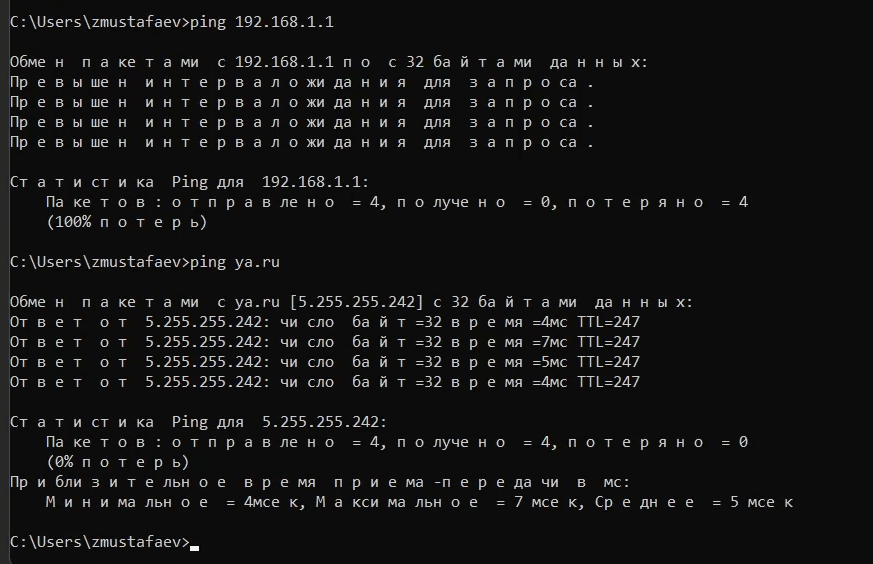
1. С помощью команды **ipconfig** был определён IP-адрес устройства 192.168.101.83 и шлюз по умолчанию 192.168.101.1.

### 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%).

* 
* Рис. 2: Ping шлюза

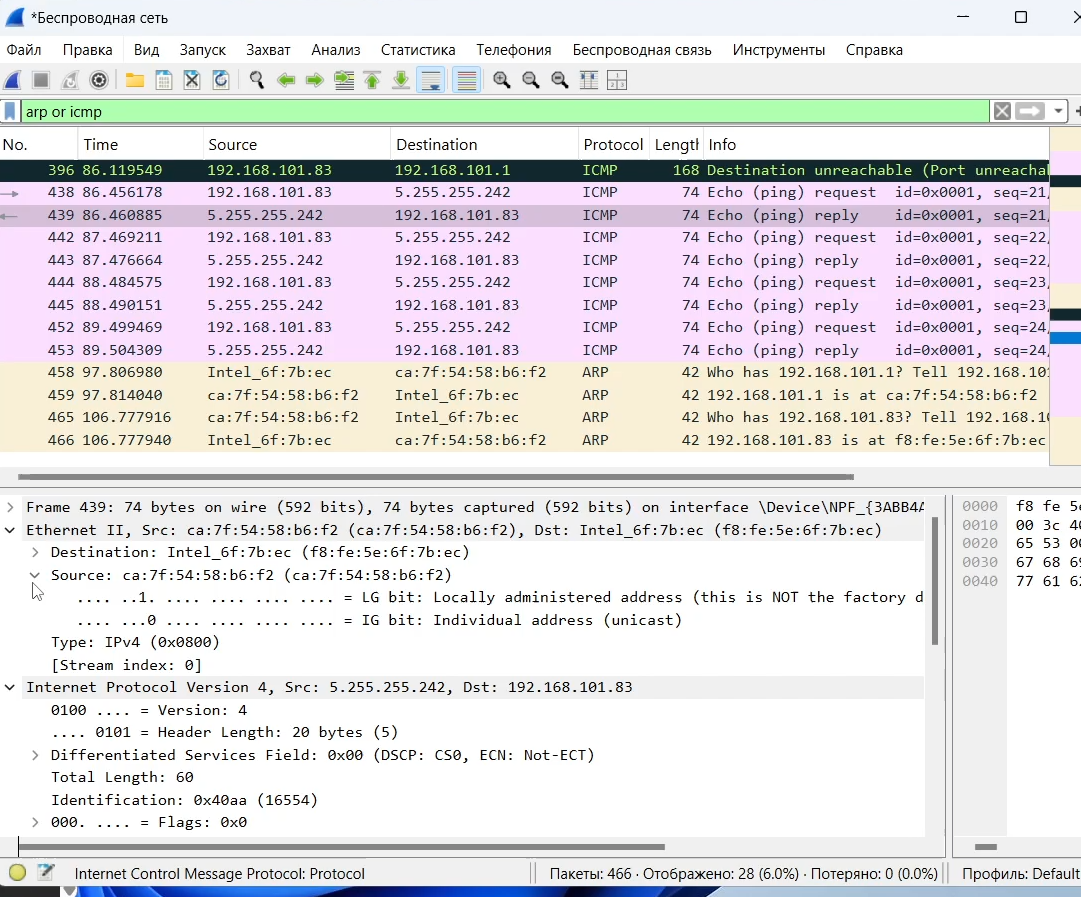
1. Аналогичная проверка с IP-адресом 192.168.1.1 также завершилась неудачно. Видимо особенность защиты роутера.

* 
* Рис. 3: Ping альтернативного шлюза

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

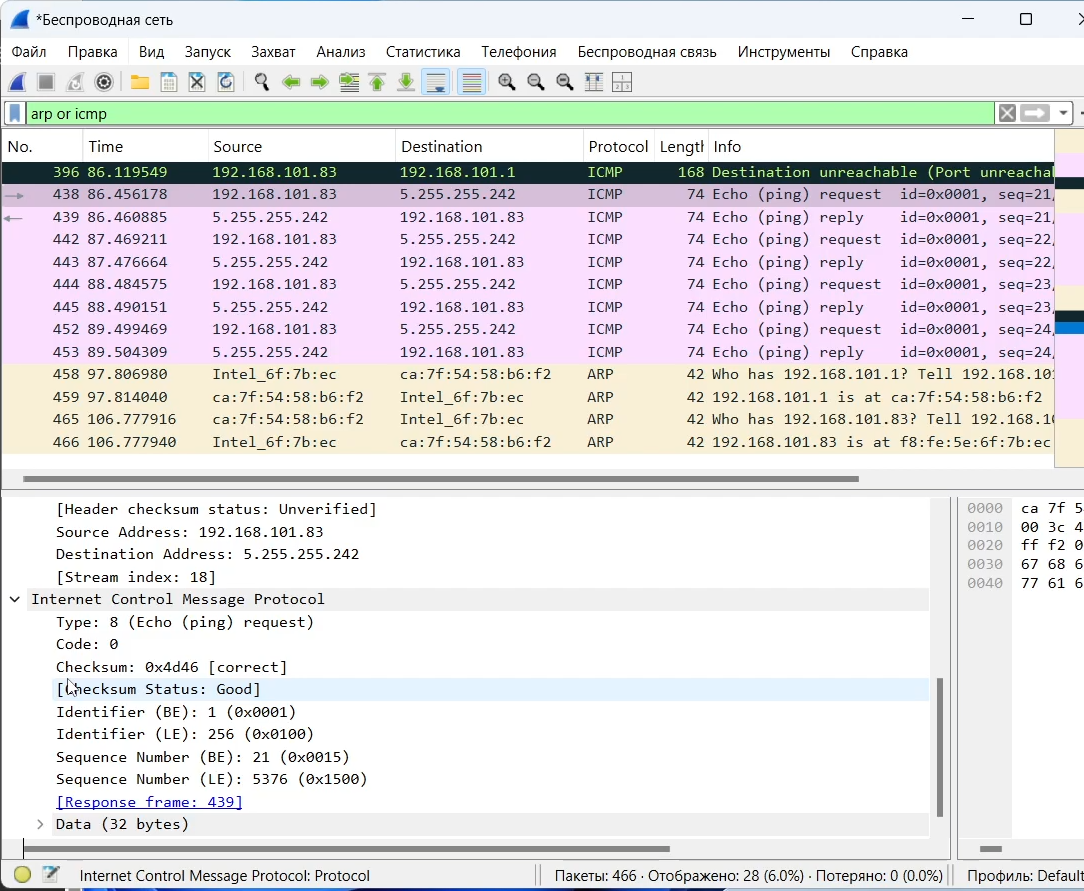
### 2.2.4 Анализ пакетов ICMP и ARP в Wireshark

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.

* 
* Рис. 4: Фильтрация пакетов в Wireshark

### 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 администрируемым**.

* 
* Рис. 5: ICMP-запрос в Wireshark

### 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

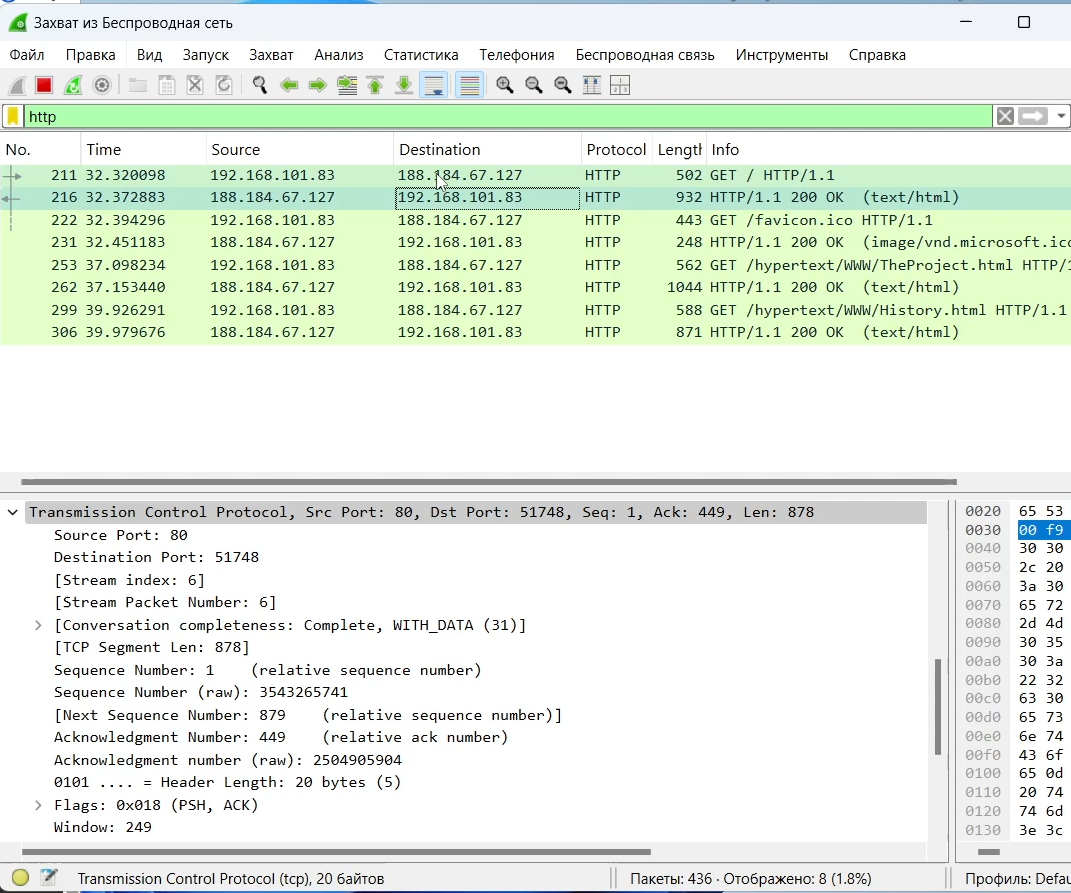
![ARP-запрос в Wireshark](04.png){ #fig:006 width=80% }

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).

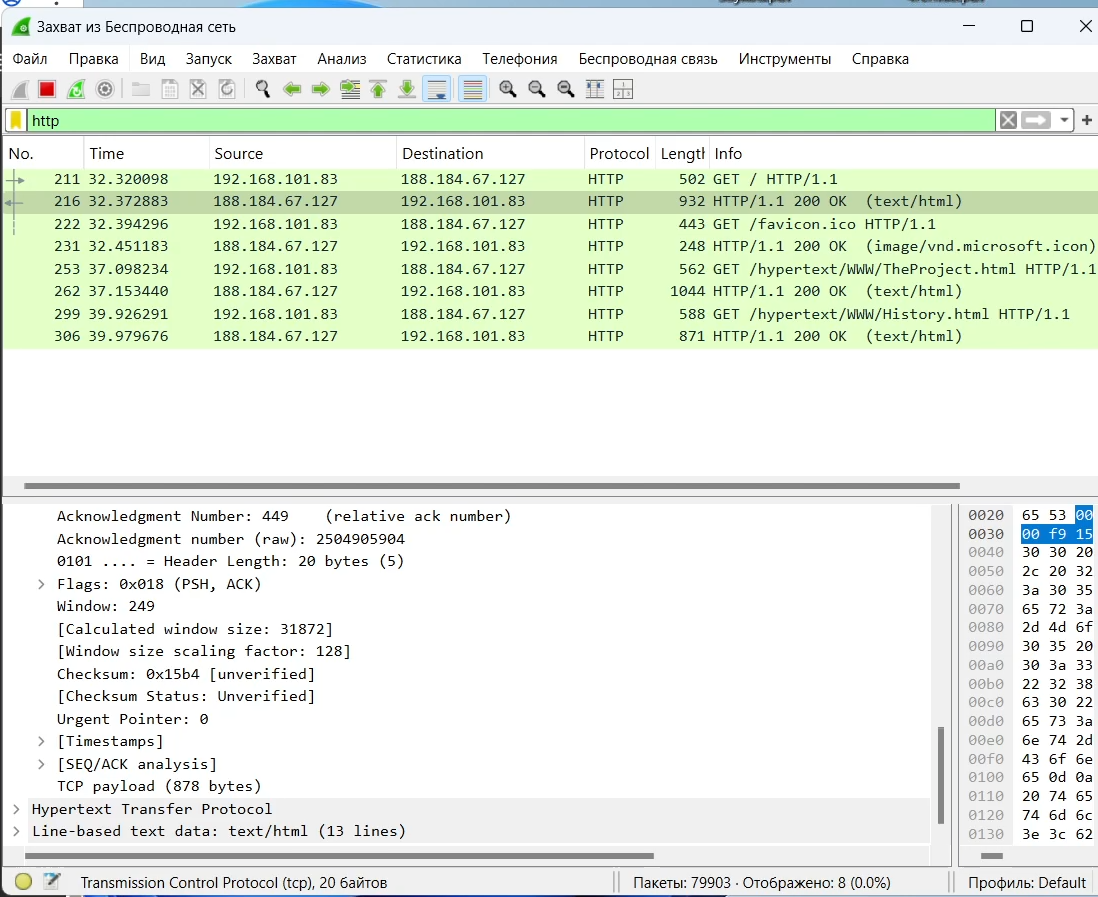
## 2.3 Анализ протоколов транспортного уровня в Wireshark

### 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.

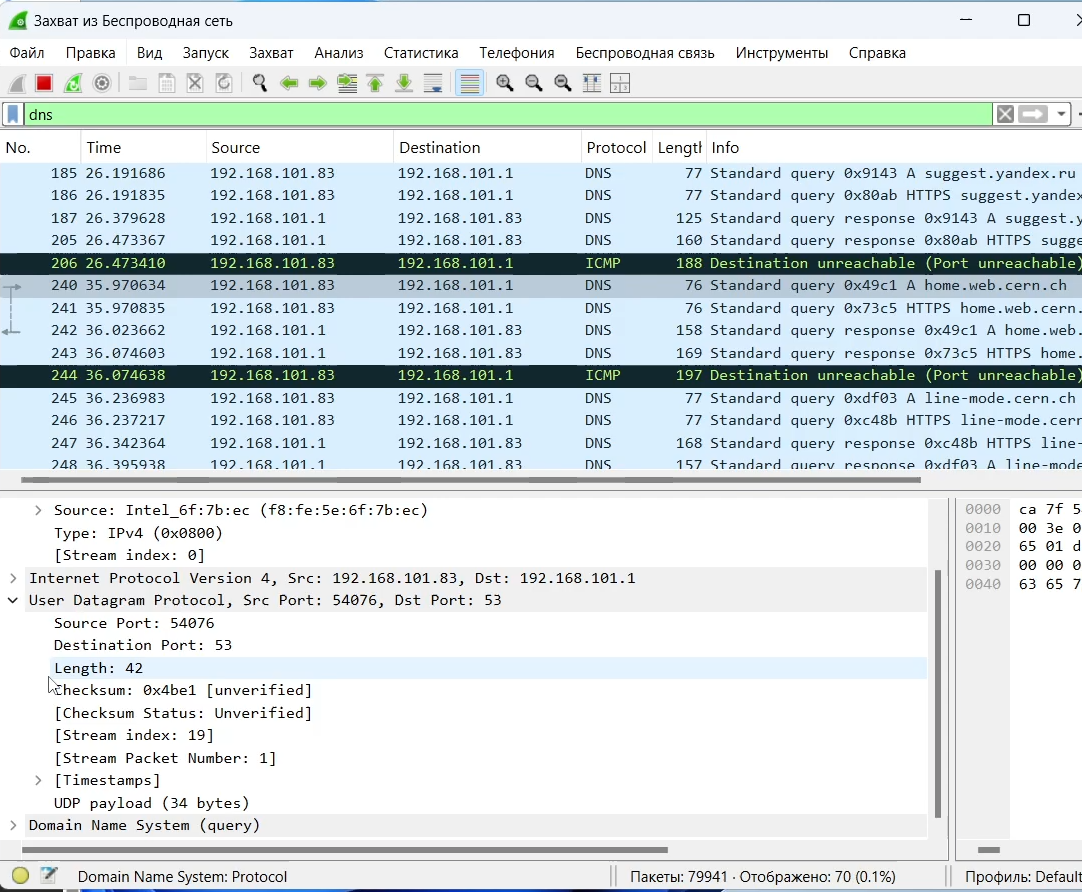
* 
* Рис. 6: HTTP-трафик

1. 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. 7: Детализация TCP-пакета

### 2.3.2 Анализ DNS (UDP)

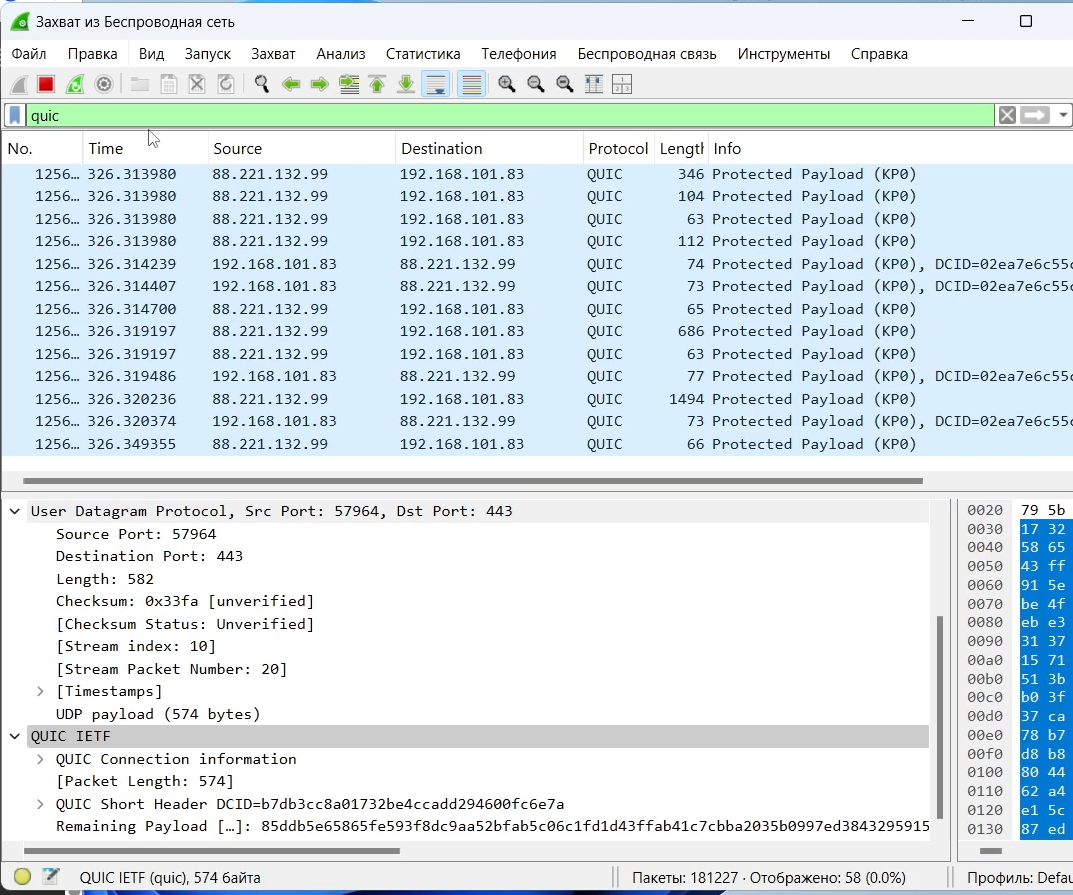
1. Для анализа DNS-запросов был применён фильтр **dns**.
2. В захваченных пакетах видно:
   * Клиент (адрес 192.168.101.83) отправляет DNS-запрос на сервер (192.168.101.1) по **UDP-порту 53**.
   * Запрос имеет тип Standard query (например, на домены yandex.ru, cern.ch).
   * Сервер отвечает сообщением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

* 
* Рис. 8: DNS-запросы и ответы

1. 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)

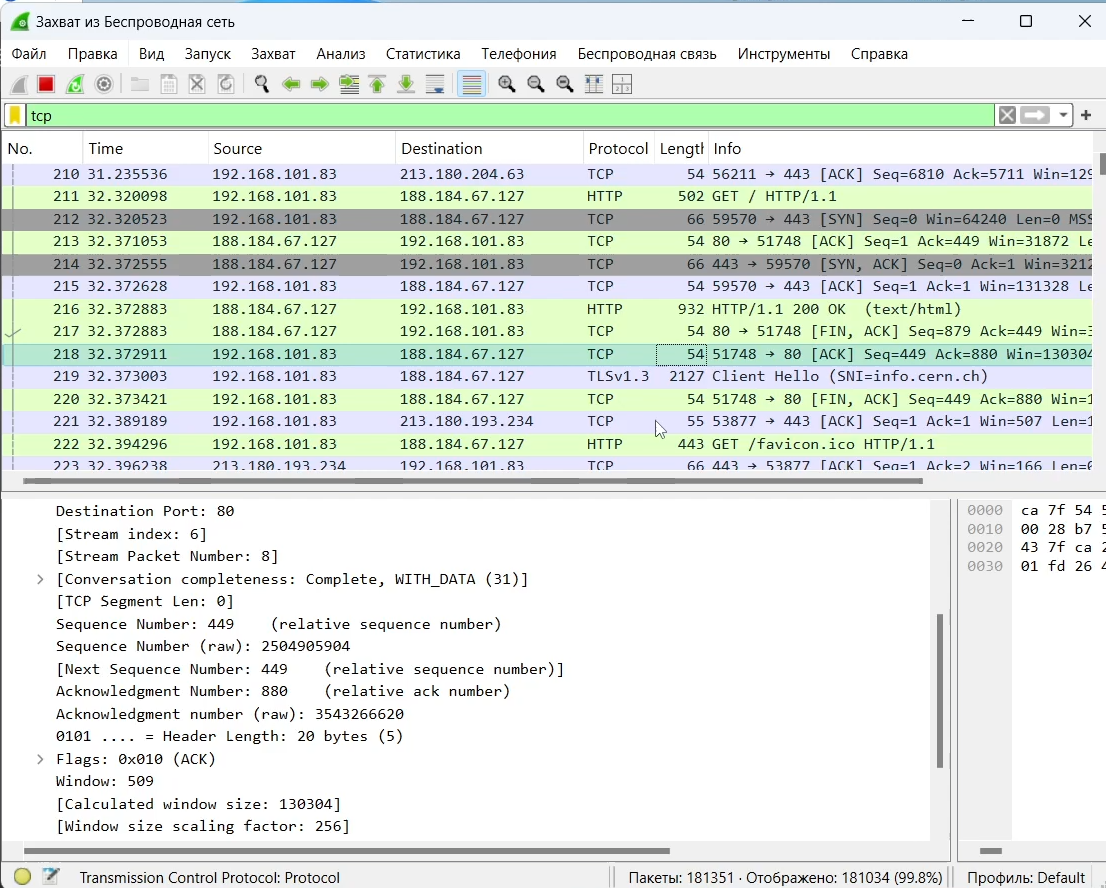
1. Для анализа работы протокола **QUIC** был применён фильтр **quic**.
2. В захваченных данных отображается взаимодействие с сервером по защищённому каналу через порт **443**.

* 
* Рис. 9: QUIC-трафик

1. 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

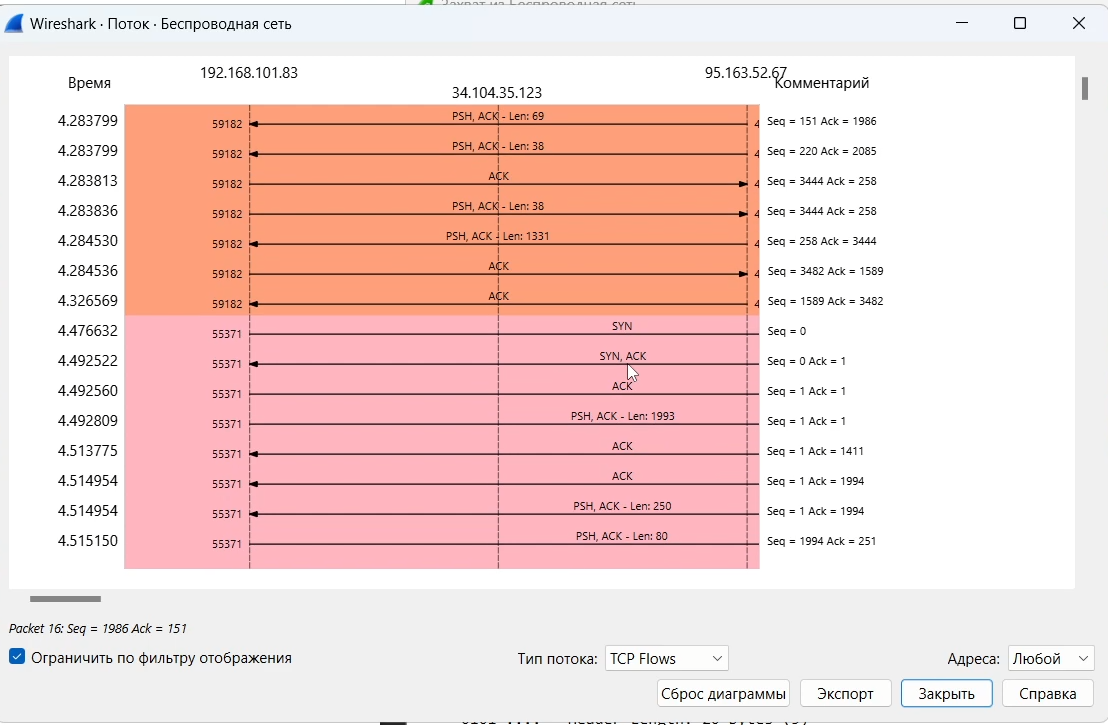
1. В захваченных данных с фильтром **tcp** отображаются пакеты установления соединения.

* 
* Рис. 10: TCP пакеты в Wireshark

1. 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 свой начальный номер последовательности.
   * **ACK** — клиент отправляет пакет с подтверждением (ACK) для завершения установки соединения. После этого соединение считается установленным, и начинается передача данных.
2. 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. 11: График TCP потока

1. 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 (SYN → SYN/ACK → ACK) и подтверждена надёжность протокола за счёт обмена номерами последовательностей и подтверждений. Построен график потока, наглядно отражающий этапы установления соединения и передачу данных.