**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

*дисциплина:* ***Сетевые технологии***

Студент: Талебу Тенке франк Устон

Группа: НФИбд-02-23

**МОСКВА**

2025 г.

## Цель работы

Изучить посредством Wireshark кадры Ethernet, проанализировать PDU протоколы транспортного и прикладного уровней стека TCP/IP.

## Задание

1. Изучение возможностей команды ipconfig для ОС типа Windows (ifconfig для систем типа Linux).

2. Определение MAC-адреса устройства и его типа.

3. С помощью Wireshark захватить и проанализировать пакеты ARP и ICMP в части кадров канального уровня.

4. С помощью Wireshark захватить и проанализировать пакеты HTTP, DNS в части заголовков и информации протоколов TCP, UDP, QUIC.

5. С помощью Wireshark проанализировать handshake протокола TCP.

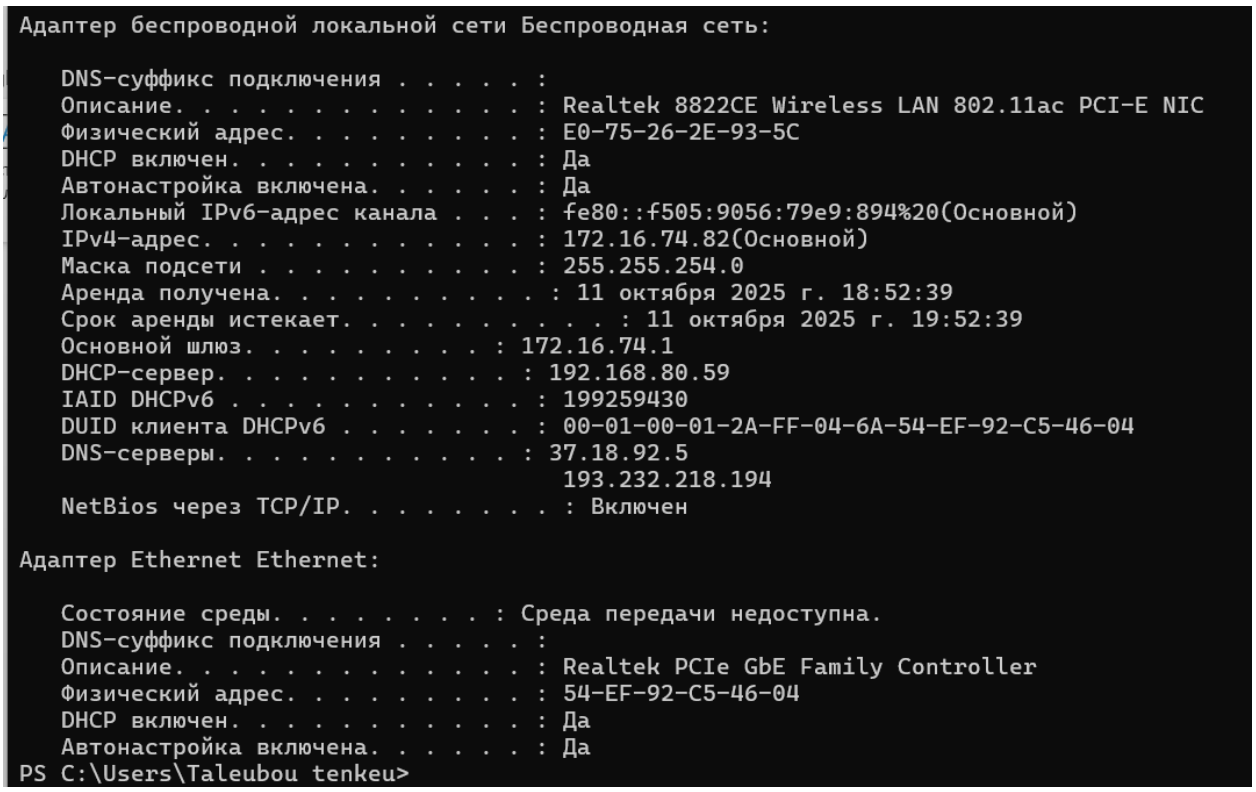
## Выполнение лабораторной работы

№1

```
Адаптер беспроводной локальной сети Подключение по локальной сети* 2:

   Состояние среды. . . . . . . . : Среда передачи недоступна.
   DNS-суффикс подключения . . . . . :

Адаптер Ethernet Ethernet 8:

   Состояние среды. . . . . . . . : Среда передачи недоступна.
   DNS-суффикс подключения . . . . . :

Адаптер беспроводной локальной сети Беспроводная сеть:

   DNS-суффикс подключения . . . . . : rudn.ru
   Локальный IPv6-адрес канала . . . : fe80::f505:9056:79e9:894%20
   IPv4-адрес. . . . . . . . . . . . : 192.168.205.147
   Маска подсети . . . . . . . . . . : 255.255.224.0
   Основной шлюз. . . . . . . . . : 192.168.192.1

Адаптер Ethernet Ethernet:
```

С помощью команды ipconfig для ОС типа Windows выведем информацию о текущем сетевом соединении.маску подсети (используется для определения сетевой и узловой частей IPv4-адреса) и шлюз.

*Рисунок 1. Команда ipconfig*




Используем также опцию /all для вывода более подробной информации.

Определим MAC-адреса сетевых интерфейсов на моем компьютере.

У меня есть помимо основной беспроводной сети WI-FI еще две локальные сети (я так понимаю они виртуальные). MAC-адрес для первого виртуального адаптера. MAC-адрес состоит из 6 октетов: первые 3 октета идентифицируют производителя, последние 3 октета идентифицируют сетевой интерфейс.Нас интересуют последние два бита (нулевой и первый биты). У меня оба нули => мой адрес индивидуальный и глобально администрируемый.

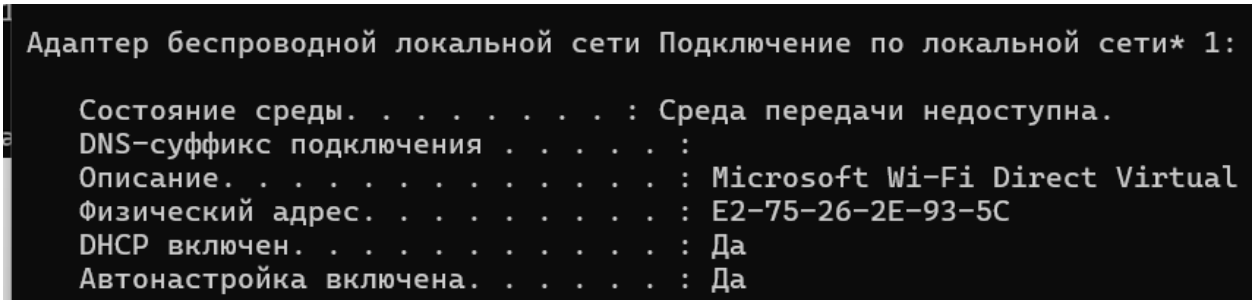



*Рисунок 3. MAC-адрес в случае виртуального адаптера 1*

*Рисунок 2. Команда ipconfig /all*

```
Адаптер беспроводной локальной сети Подключение по локальной сети* 2:

   Состояние среды. . . . . : Среда передачи недоступна.
   DNS-суффикс подключения . . . . . :
   Описание. . . . . : Microsoft Wi-Fi Direct Virtual
   Физический адрес. . . . . : F2-75-26-2E-93-5C
```

MAC-адрес для второго виртуального адаптераЭтот адрес является индивидуальным и локально администрируемым (поэтому по нему нельзя узнать производителя).

*Рисунок 4. MAC-адрес в случае виртуального адаптера 2*

```
Адаптер беспроводной локальной сети Беспроводная сеть:

   DNS-суффикс подключения . . . . . :
   Описание. . . . . : Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC
   Физический адрес. . . . . : E0-75-26-2E-93-5C
   DHCP включен. . . . . : Да
   Автонастройка включена. . . . . : Да
   Локальный IPv6-адрес канала . . . : fe80::f505:9056:79e9:894%20(Основной)
   IPv4-адрес. . . . . : 172.16.74.82(Основной)
```

MAC-адрес для беспроводной сети WI-FI Этот адрес является индивидуальным и глобально администрируемым

*Рисунок 5. MAC-адрес в случае WI-FI*

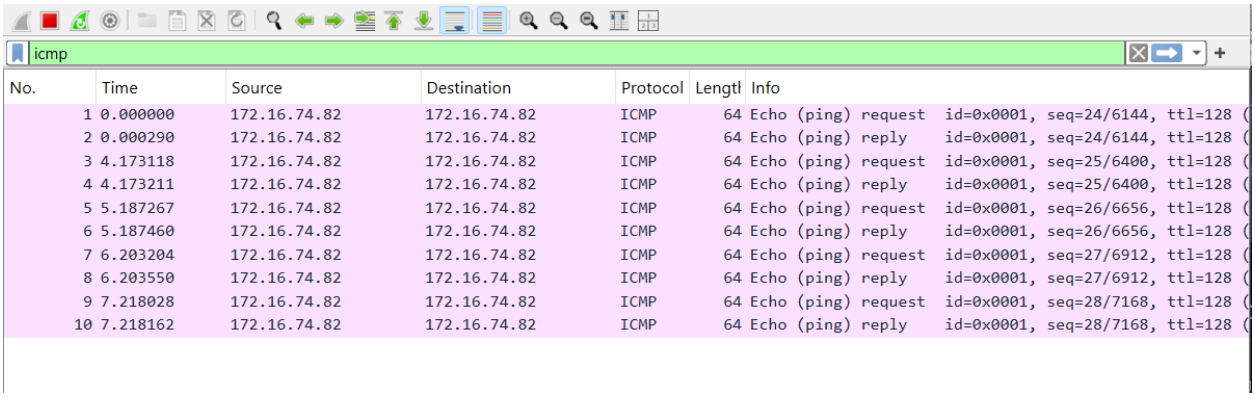
№2

Из предыдущего задания мы узнали адрес основного шлюза

```
PS C:\Users\Taleubou tenkeu> ping 172.16.74.82

Обмен пакетами с 172.16.74.82 по с 32 байтами данных:
Ответ от 172.16.74.82: число байт=32 время<1мс TTL=128
Ответ от 172.16.74.82: число байт=32 время<1мс TTL=128
Ответ от 172.16.74.82: число байт=32 время<1мс TTL=128
Ответ от 172.16.74.82: число байт=32 время<1мс TTL=128

Статистика Ping для 172.16.74.82:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
```

Теперь пропингуем его, предварительно запустив Wireshark и включив захват трафика. Посылаются 4 пакета, 4 пакета получено назад.

*Рисунок 6. Пингование шлюза*



icmp

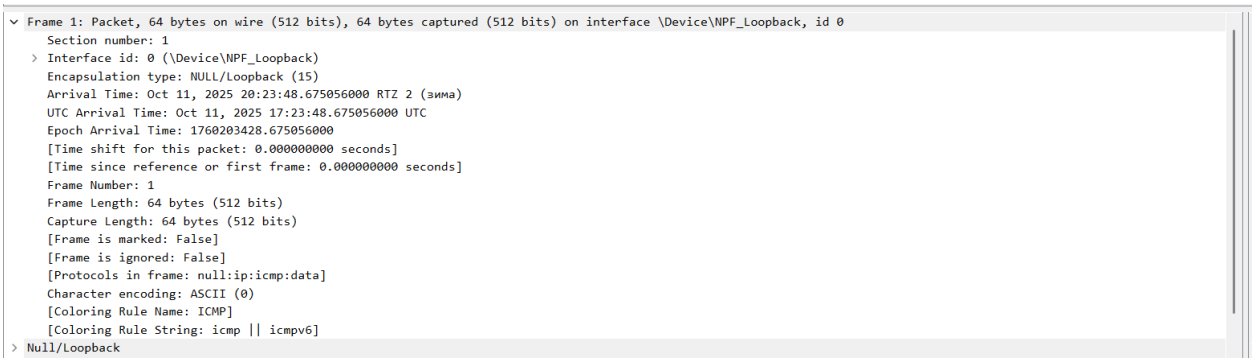
| No. | Time | Source | Destination | Protocol | Lengtl | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) request id=0x0001, seq=24/6144, ttl=128 ( |
| 2 | 0.000290 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) reply id=0x0001, seq=24/6144, ttl=128 ( |
| 3 | 4.173118 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) request id=0x0001, seq=25/6400, ttl=128 ( |
| 4 | 4.173211 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) reply id=0x0001, seq=25/6400, ttl=128 ( |
| 5 | 5.187267 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) request id=0x0001, seq=26/6656, ttl=128 ( |
| 6 | 5.187460 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) reply id=0x0001, seq=26/6656, ttl=128 ( |
| 7 | 6.203204 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) request id=0x0001, seq=27/6912, ttl=128 ( |
| 8 | 6.203550 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) reply id=0x0001, seq=27/6912, ttl=128 ( |
| 9 | 7.218028 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) request id=0x0001, seq=28/7168, ttl=128 ( |
| 10 | 7.218162 | 172.16.74.82 | 172.16.74.82 | ICMP | 64 | Echo (ping) reply id=0x0001, seq=28/7168, ttl=128 ( |

В строке фильтра пропишем фильтр icmp. Убедимся, что в списке пакетов отобразятся только пакеты ICMP, в частности пакеты, которые были

*Рисунок 7. Пакеты ICMP*

сгенерированы с помощью команды ping, отправленной с моего устройства на шлюз по умолчанию.

Изучим эхо-запрос и эхо-ответ ICMP в программе Wireshark:



```
Frame 1: Packet, 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_Loopback, id 0
    Section number: 1
  > Interface id: 0 (\Device\NPF_Loopback)
    Encapsulation type: NULL/Loopback (15)
    Arrival Time: Oct 11, 2025 20:23:48.675056000 RTZ 2 (зима)
    UTC Arrival Time: Oct 11, 2025 17:23:48.675056000 UTC
    Epoch Arrival Time: 1760203428.675056000
    [Time shift for this packet: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 64 bytes (512 bits)
    Capture Length: 64 bytes (512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: null:ip:icmp:data]
    Character encoding: ASCII (0)
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
> Null/Loopback
```

На панели списка пакетов (верхний раздел) выберем первый указанный кадр ICMP — эхо-запрос. Изучим информацию на панели сведений о пакете в средней части экрана. На вкладке физического уровня можно найти длину кадра (64 бита), тип Ethernet – Ethernet (1).

*Рисунок 8. Кадр физического уровня*

Чтобы узнать MAC-адрес источника и шлюза перейдем на канальный

уровень. Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства Адрес шлюза (destination, то куда отправлен запрос) Тип адреса тут указан (показаны нулевые и первые биты MAC-адресов). Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

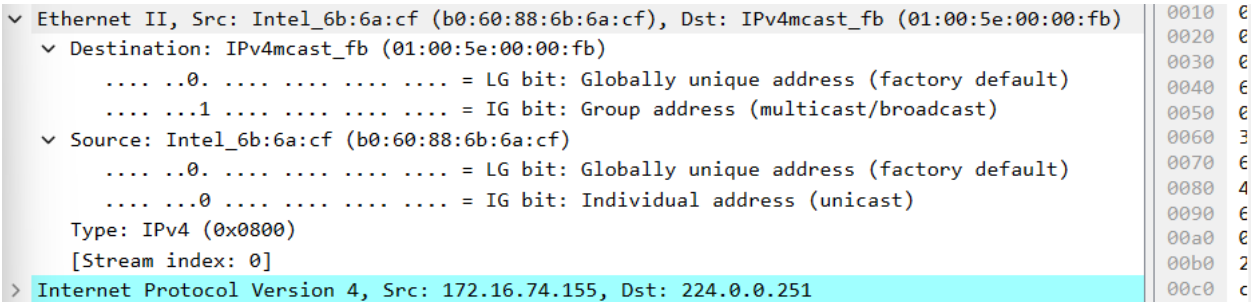



*Рисунок 9. Кадр канального уровня*

Далее посмотрим на полученный ответ. Тут все почти то же самое, что и в запросе (длина кадра 64 бита). Только теперь MAC-адрес источника – MAC-адрес шлюза а адрес назначения – адрес моего устройства

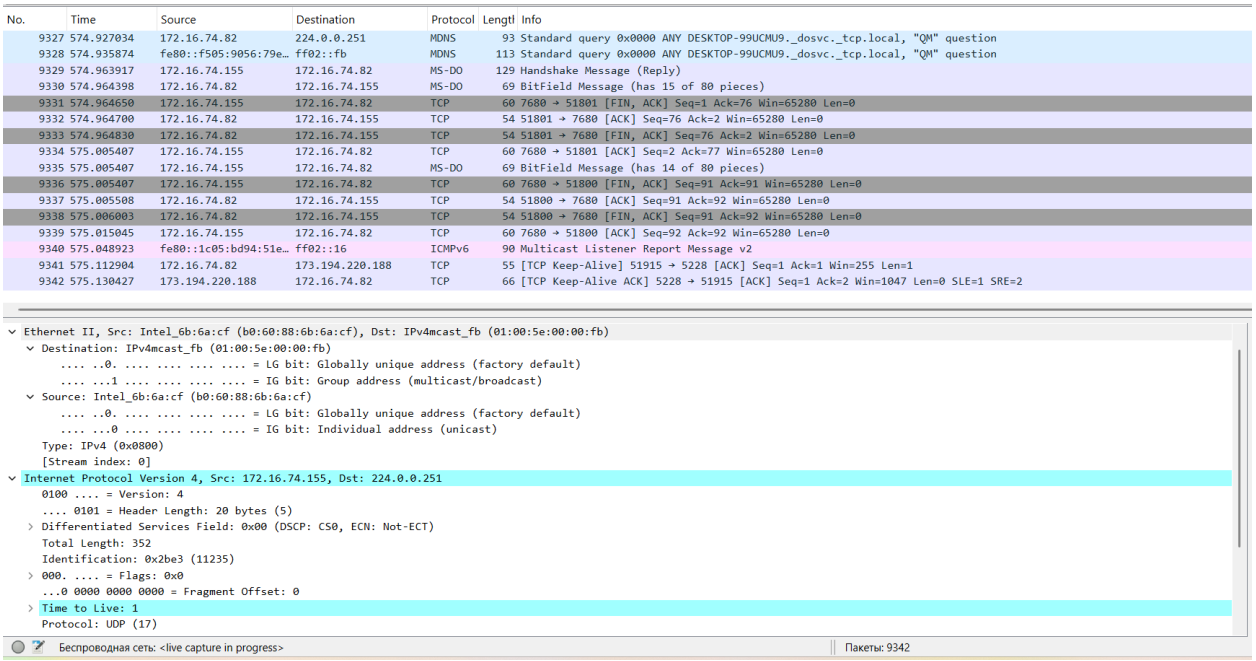



*Рисунок 10. Эхо-ответ*

Изучим кадры данных протокола ARP. Hardware type – это адрес канального уровня (Ethernet (1)), Protocol type – сетевой уровень (протокол IPv4), далее указаны размеры MAC-адреса (6 байт) и размер IPv4-адреса (4

байта). Код запроса – 1.

```
PS C:\Users\Taleubou tenkeu> ping www.yandex.ru

Обмен пакетами с www.YANDEX.ru [5.255.255.77] с 32 байтами данных:
Ответ от 5.255.255.77: число байт=32 время=26мс TTL=248
Ответ от 5.255.255.77: число байт=32 время=8мс TTL=248
Ответ от 5.255.255.77: число байт=32 время=5мс TTL=248
Ответ от 5.255.255.77: число байт=32 время=5мс TTL=248

Статистика Ping для 5.255.255.77:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 5мсек, Максимальное = 26 мсек, Среднее = 11 мсек
PS C:\Users\Taleubou tenkeu>
```

Изучим данные в полях заголовка Ethernet II.

Здесь указаны MAC-адреса источника и получателя. Получатель в нашем случае – широковещательный адрес (групповой и локально администрируемый). Источник – адрес нашего шлюза (индивидуальный и глобально администрируемый).

*Рисунок 11. Протокол ARP*

```
PS C:\Users\Taleubou tenkeu> ping www.yandex.ru

Обмен пакетами с www.YANDEX.ru [5.255.255.77] с 32 байтами данных:
Ответ от 5.255.255.77: число байт=32 время=5мс TTL=248
Ответ от 5.255.255.77: число байт=32 время=6мс TTL=248
Ответ от 5.255.255.77: число байт=32 время=5мс TTL=248
Ответ от 5.255.255.77: число байт=32 время=5мс TTL=248

Статистика Ping для 5.255.255.77:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 5мсек, Максимальное = 6 мсек, Среднее = 5 мсек
PS C:\Users\Taleubou tenkeu>
```

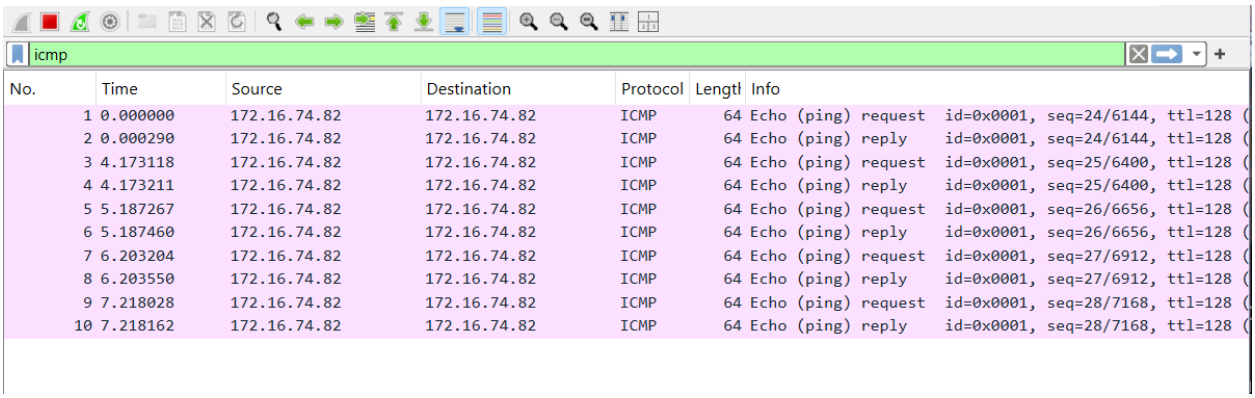
Начнем новый процесс захвата трафика в Wireshark. Пропингуем сайт яндекса.

Изучим запрос протокола ICMP. Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства Адрес получателя (destination, то куда

*Рисунок 12. Пингование сайта www.yandex.ru*

отправлен запрос) Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

*Рисунок 13. Запрос протокола ICMP*




Изучим запрос протокола ICMP. Адрес источника (Source, то куда откуда

*Рисунок 14. Ответ протокола ICMP*

запрос отправлен) - Адрес получателя (Destination, то куда отправлен запрос) – это адрес моего устройства Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

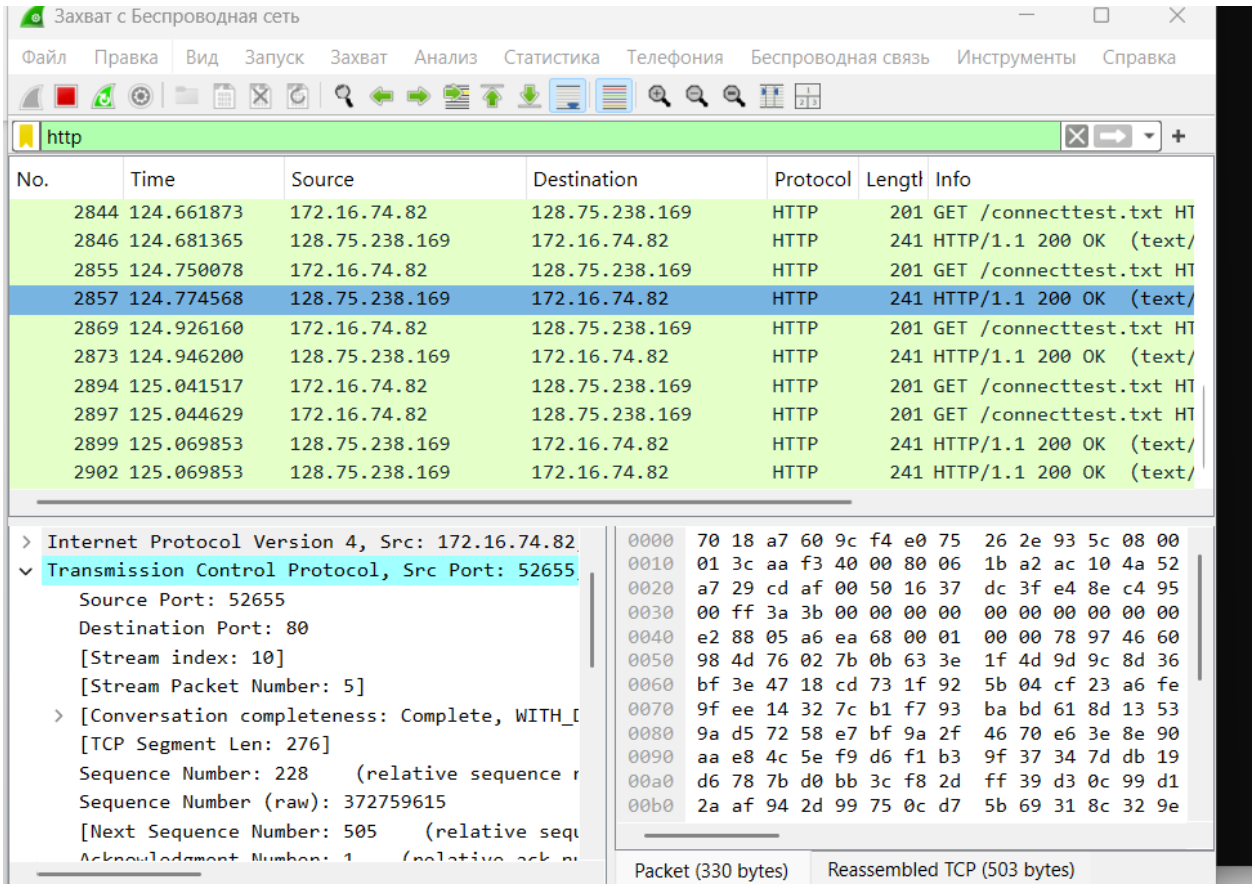
№3

В браузере перейдем на сайт, работающий по протоколу HTTP (например, на сайт CERN http://info.cern.ch/). Для получения большей информации для Wireshark поперемещались по ссылкам или разделам сайта в браузере.

В Wireshark в строке фильтра укажем http и проанализируем информацию по протоколу TCP в случае запросов и ответов.

Открываем раздел протокола TCP в случае запроса. Видим, что порт получателя (это стандартный порт для http). Порт источника(он определяется случайным образом из незанятых и непривилегированных портов). Также тут есть поле Порядковый номер (Sequence Number) и поле Номер подтверждения (Acknowledgment Number).

*Рисунок 15. Протокол TCP (случай запроса)*




Далее рассмотрим ответ. Здесь у нас поменялись местами порты источника и получателя.

*Рисунок 16. Протокол TCP (случай ответа)*

Теперь порт источника – порт сайта (80). А порт получателя (выбранный случайным образом).

В Wireshark в строке фильтра укажем dns и проанализируем информацию по протоколу UDP в случае запросов и ответов.

В случае запроса: порт источника (выбранный случайным образом из незанятых и непривилегированных портов). Порт получателя

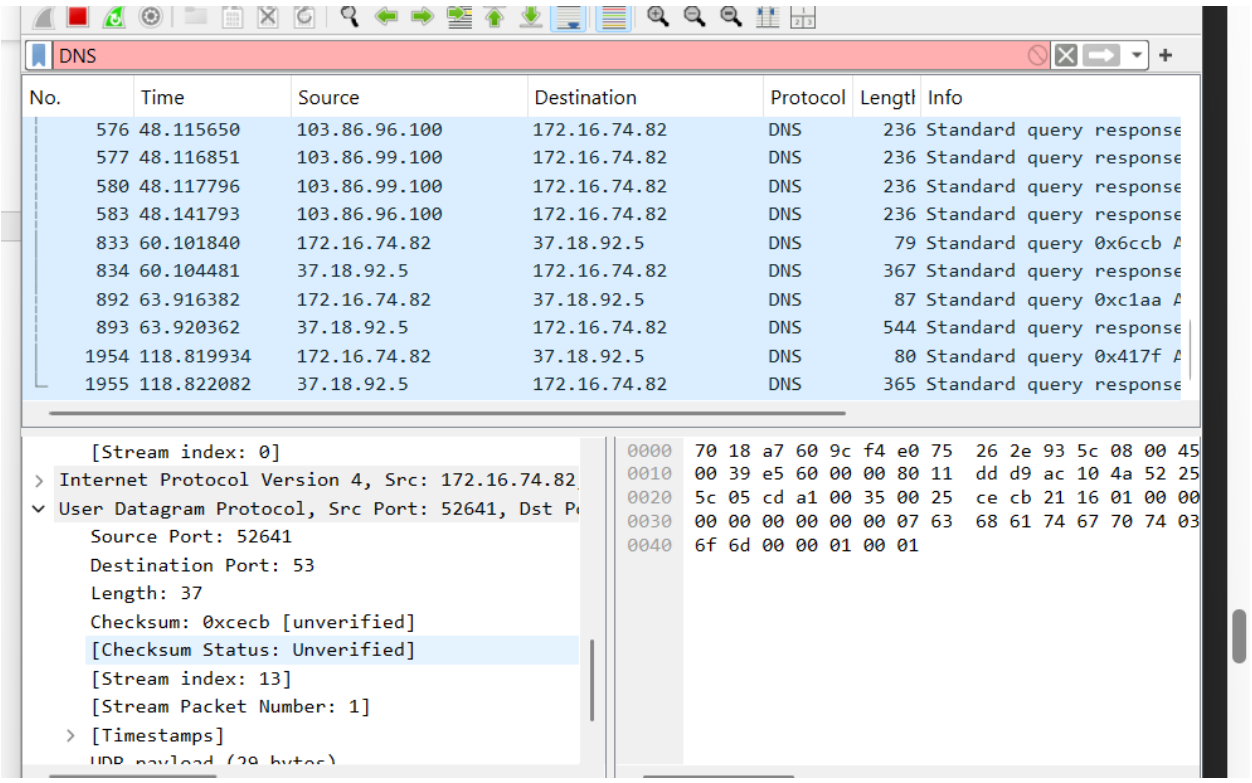



*Рисунок 17. Протокол UDP (случай запроса)*

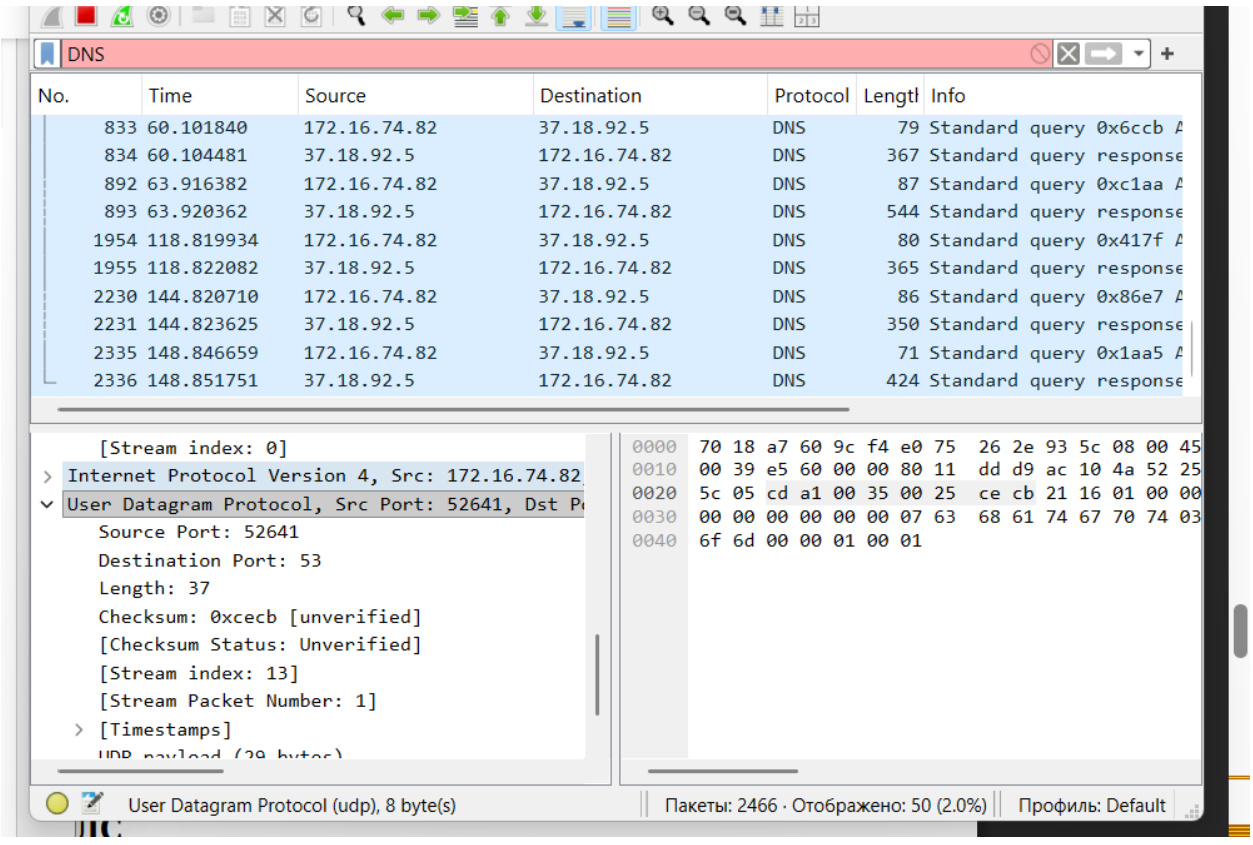


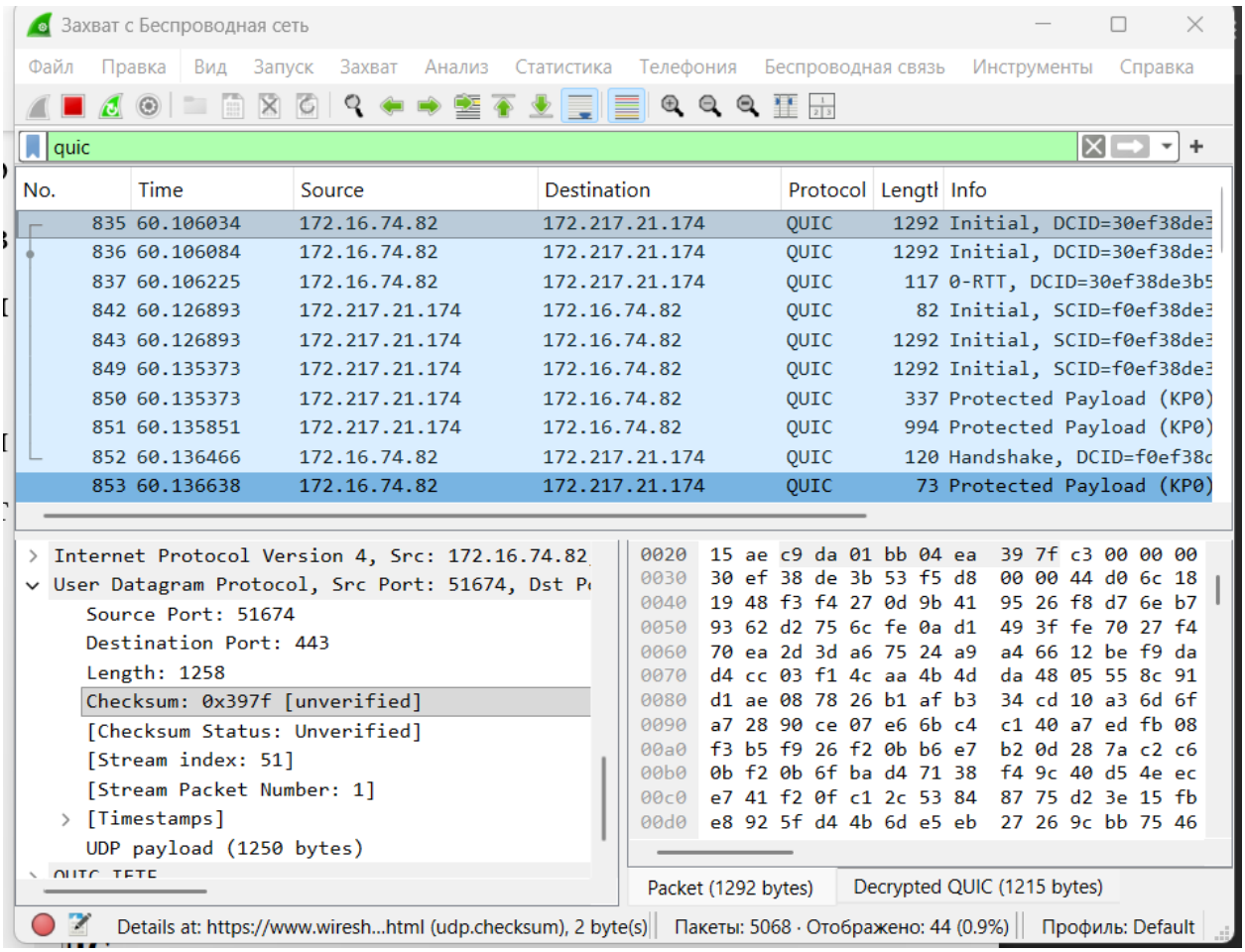

*Рисунок 18. Протокол UDP (случай ответа)*

В случае ответа порт источника  а порт получателя.

В Wireshark в строке фильтра укажем quic и проанализируем информацию по протоколу quic в случае запросов и ответов.

Как и в случае dns можем посмотреть информацию транспортного уровня по протоколу UDP. Порт источника задан случайно, выбором из непривелигированных и незанятых портов, и равен, порт получателя равен - это стандартный порт HTTPS, то есть quic сразу криптуется.

Для создания альтернативы TCP поверх UDP строятся протоколы прикладного уровня QUIC IETF, которые управляют трафиком, управляют качеством обслуживания




.

*Рисунок 19. Запрос quic*

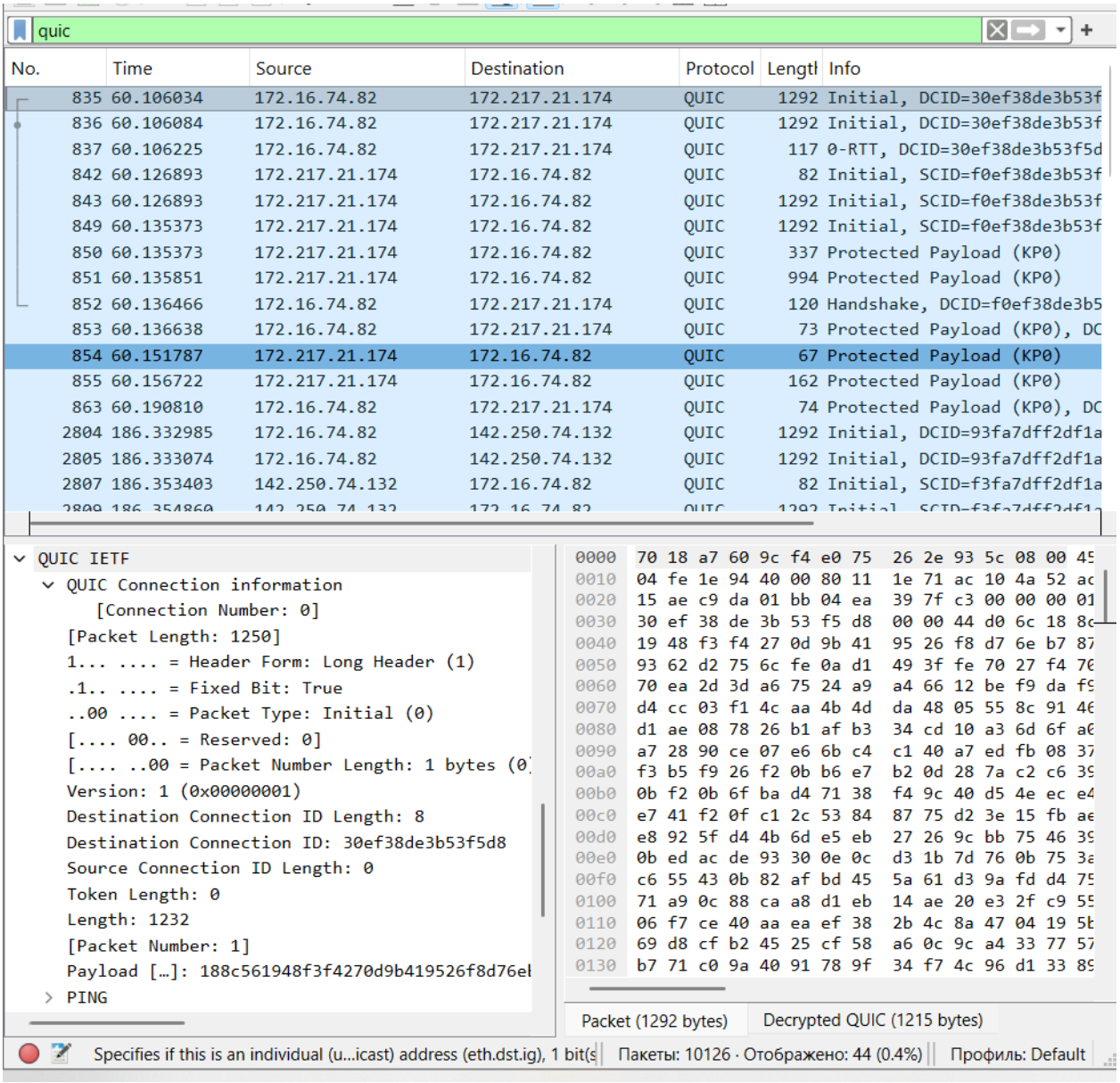
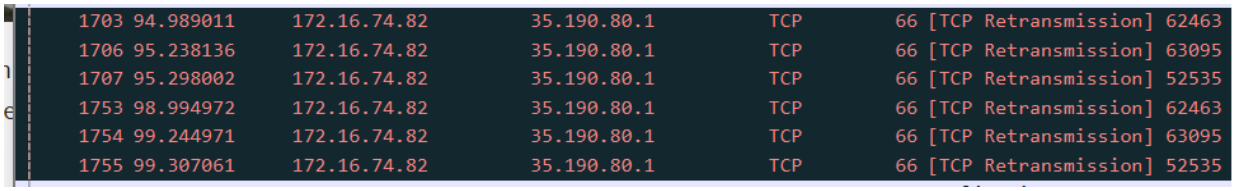



*Рисунок 20. Ответ quic*

В случае ответа порты заданы наоборот, то есть источник порт, получатель.

№4

в Wireshark пакетов TCP.



| | | | | | |
|---|---|---|---|---|---|
| 1703 | 94.989011 | 172.16.74.82 | 35.190.80.1 | TCP | 66 [TCP Retransmission] 62463 |
| 1706 | 95.238136 | 172.16.74.82 | 35.190.80.1 | TCP | 66 [TCP Retransmission] 63095 |
| 1707 | 95.298002 | 172.16.74.82 | 35.190.80.1 | TCP | 66 [TCP Retransmission] 52535 |
| 1753 | 98.994972 | 172.16.74.82 | 35.190.80.1 | TCP | 66 [TCP Retransmission] 62463 |
| 1754 | 99.244971 | 172.16.74.82 | 35.190.80.1 | TCP | 66 [TCP Retransmission] 63095 |
| 1755 | 99.307061 | 172.16.74.82 | 35.190.80.1 | TCP | 66 [TCP Retransmission] 52535 |

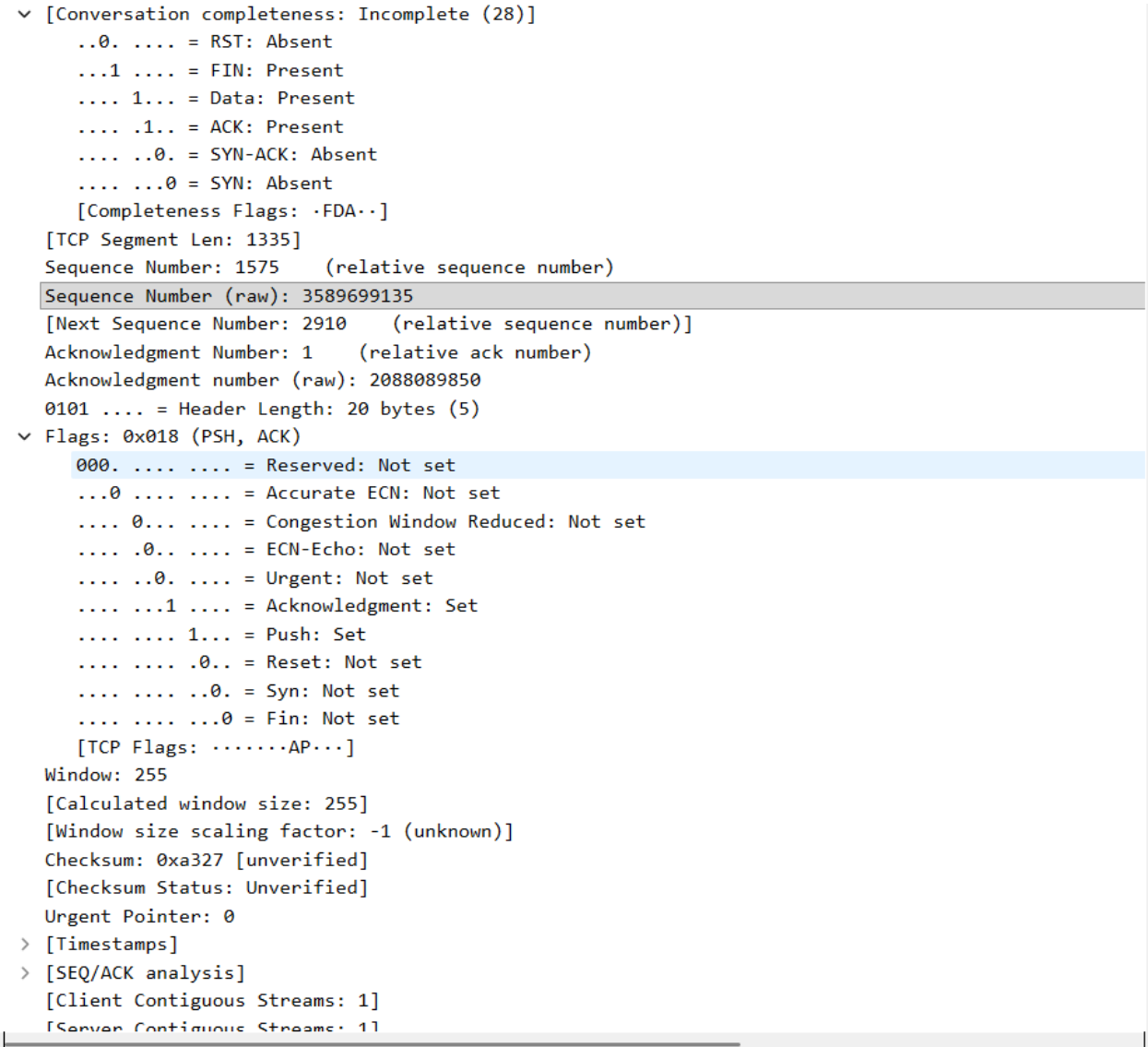
Проанализируем handshake протокола TCP.

*Рисунок 21. Handshake TCP*

Режим активного доступа (Active Open). Клиент посылает сообщение SYN, ISSa, т.е. в передаваемом сообщении установлен бит SYN (Synchronize Sequence Number), а в поле Порядковый номер (Sequence Number) — начальное 32-битное значение ISSa (Initial Sequence Number).

Значение Sequence Number равно (ISSa), значение Acknowledgment Number равно 0. Также видим, что установлен флаг SYN.

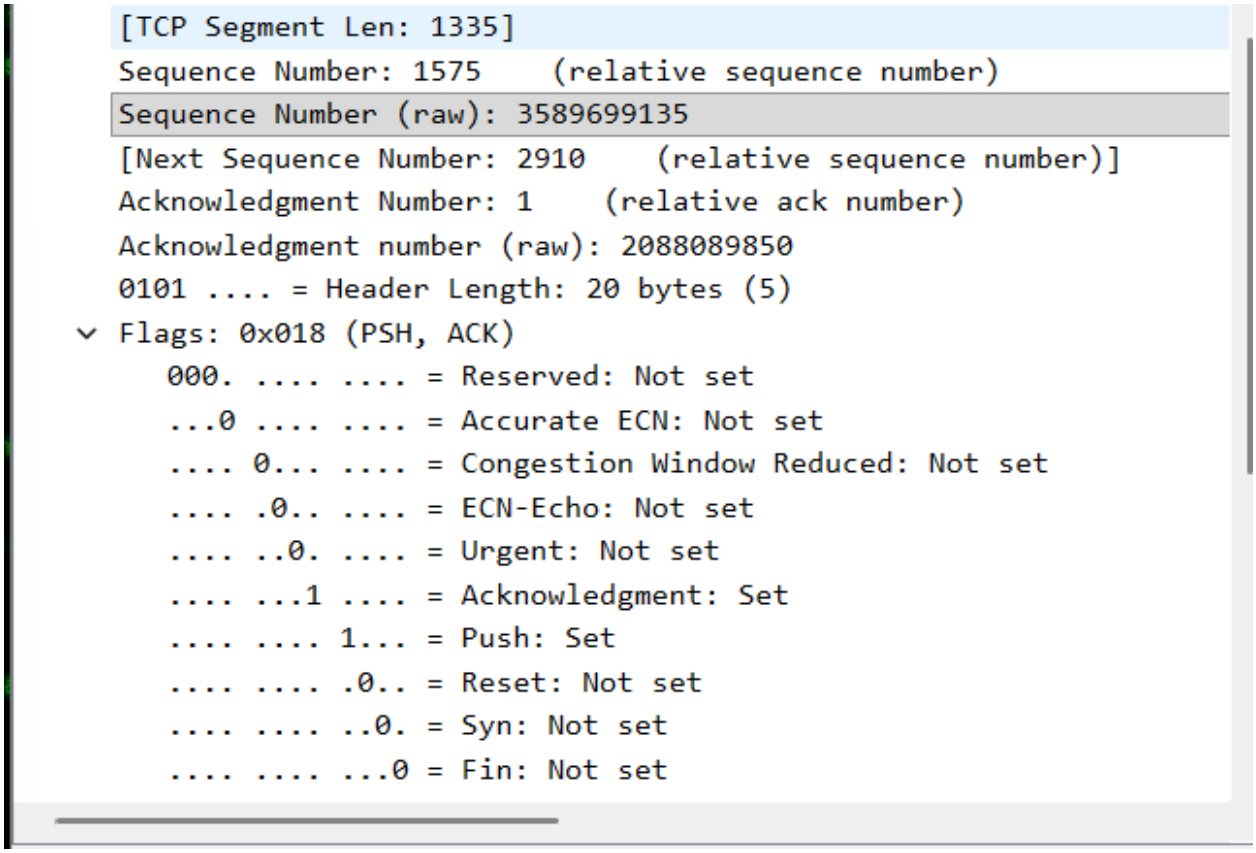
*Рисунок 22. Протокол TCP для первой ступени handshake*



Режим пассивного доступа (Passive Open). Сервер откликается, посылая сообщение SYN, ACK, ISSb, ACK(ISSa+1), т.е. установлены биты SYN и ACK; в поле Порядковый номер (Sequence Number) хостом B устанавливается начальное значение счётчика — ISSb; поле Номер подтверждения (Acknowledgment Number) содержит значение ISSa, полученное в первом пакете от хоста A и увеличенное на единицу.

Sequence Number равен 20880889850 (начальное значение счётчика — ISSb). Установлены флаги SYN, ACK.

*Рисунок 23. Протокол TCP для второй ступени handshake*




Завершение рукопожатия. Клиент отправляет подтверждение получения SYN

сегмента от сервера с идентификатором, равным ISN (сервера)+1: ACK, ISSa+1, ACK(ISSb+1). В этом пакете установлен битACK, поле Порядковый номер (Sequence Number) содержит ISSa+1, поле Номер подтверждения (Acknowledgment Number) содержит значение ISSb+1. Посылкой этого пакета заканчивается трёхступенчатый handshake, и TCP-соединение считается установленным.

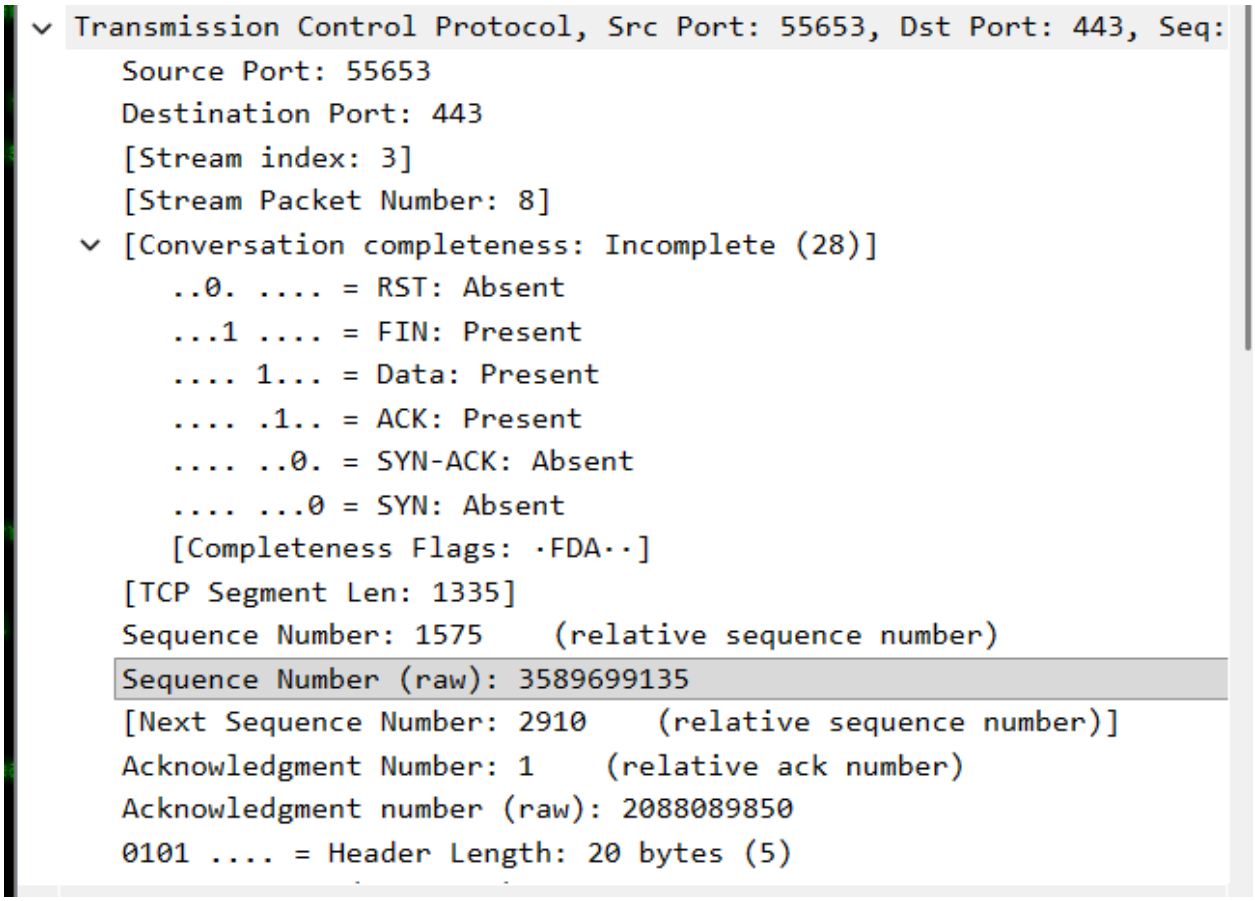



*Рисунок 24. Протокол TCP для третьей ступени handshake*

Далее посмотрим график потока. Здесь в принципе все то же самое, что мы уже разобрали, только на графике. Клиент посылает запрос серверу (установлен бит SYN), Seq = 0. Далее сервер отвечает клиенту (установлены биты SYN, ACK), Seq = 0, Ack =1 (это относительные значения). Ну звершение рукопажаттия: клиент отправляет серверу подтверждение получения SYN сегмента, Seq = 1, Ack = 1.

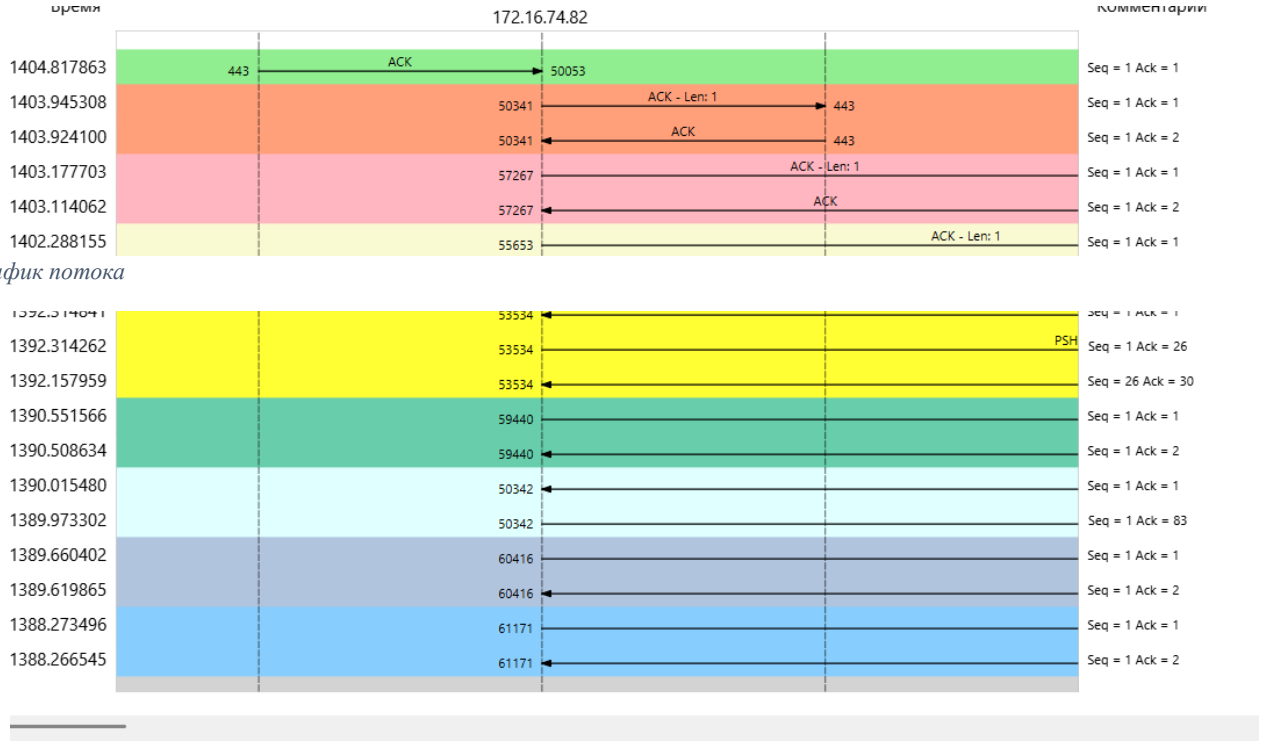



*Рисунок 25. График потока*

## Выводы

В процессе выполнения данной лабораторной работы я изучил посредством Wireshark кадры Ethernet, проанализировал PDU протоколы транспортного и прикладного уровней стека TCP/IP.