

# **Отчёт по лабораторной работе №3**

**Сетевые технологии**

Ищенко Ирина НПИбд-02-22

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение лабораторной работы</b>	<b>6</b>
<b>3 Выводы</b>	<b>22</b>

# Список иллюстраций

2.1 Команда ipconfig . . . . .	6
2.2 Команда ipconfig /all . . . . .	7
2.3 Пингование шлюза . . . . .	8
2.4 Пакеты ICMP. Кадр физического уровня . . . . .	8
2.5 Эхо-ответ. Кадр канального уровня . . . . .	9
2.6 Протокол ARP . . . . .	10
2.7 Пингование сайта www.yandex.ru . . . . .	11
2.8 Запрос протокола ICMP . . . . .	11
2.9 Ответ протокола ICMP . . . . .	12
2.10 Протокол TCP . . . . .	13
2.11 Протокол UDP . . . . .	14
2.12 Протокол TCP . . . . .	15
2.13 Протокол UDP (случай ответа) . . . . .	16
2.14 Запрос quic . . . . .	17
2.15 Ответ quic . . . . .	18
2.16 Протокол TCP для первой ступени handshake . . . . .	19
2.17 Протокол TCP для второй ступени handshake . . . . .	20
2.18 Протокол TCP для третьей ступени handshake . . . . .	21
2.19 График потока . . . . .	21

# **Список таблиц**

# **1 Цель работы**

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

## 2 Выполнение лабораторной работы

С помощью команды ipconfig для ОС типа Windows выведем информацию о текущем сетевом соединении. Отсюда мы можем узнать IPv6-адрес, IPv4-адрес (уникальный IPv4-адрес узла), маску подсети (используется для определения сетевой и узловой частей IPv4-адреса) и шлюз (рис. 2.1).

```
Administrator: Командная строка
Настройка протокола IP для Windows

Адаптер Ethernet VirtualBox Host-Only Network:
DNS-суффикс подключения . . . . . : 
Локальный IPv6-адрес канала . . . . . : fe80::1fb:20fd:6e13:e512%17
IPv4-адрес . . . . . : 192.168.56.1
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . :

Адаптер беспроводной локальной сети Подключение по локальной сети* 1:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :

Адаптер беспроводной локальной сети Подключение по локальной сети* 2:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :

Адаптер беспроводной локальной сети Беспроводная сеть:
DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . . . : fe80::1411:3b7:b2eb:fc0a%19
IPv4-адрес. . . . . : 192.168.170.67
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . : 192.168.170.5

C:\WINDOWS\system32> 3.3.2.1. Постановка задачи
```

Рис. 2.1: Команда ipconfig

Используем также опцию /all для вывода более подробной информации. Определим MAC-адреса сетевых интерфейсов на моем компьютере (рис. 2.2). У меня есть помимо основной беспроводной сети WI-FI еще две локальные сети - виртуальные. MAC-адрес для первого виртуального адаптера: 94-08-53-47-36-D7. MAC-адрес состоит из 6 октетов: первые 3 октета идентифицируют производителя, последние 3 октета идентифицируют сетевой интерфейс. Разберем первый

байт MAC-адреса (94), переведем в двоичный код  $94 = 10010100$ . Нас интересуют последние два бита (нулевой и первый биты). У меня оба нули  $\Rightarrow$  мой адрес индивидуальный и глобально администрируемый.

Адаптер беспроводной локальной сети Подключение по локальной сети\* 1:

Состояние среды . . . . . : Среда передачи недоступна.  
DNS-суффикс подключения . . . . . :  
Описание . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #3  
Физический адрес . . . . . : 94-08-53-47-36-D7  
DHCP включен . . . . . : Да  
Автонастройка включена . . . . . : Да

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

Состояние среды . . . . . : Среда передачи недоступна.  
DNS-суффикс подключения . . . . . :  
Описание . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #4  
Физический адрес . . . . . : D6-08-53-47-36-D7  
DHCP включен . . . . . : Нет  
Автонастройка включена . . . . . : Да

Адаптер беспроводной локальной сети Беспроводная сеть:

DNS-суффикс подключения . . . . . :  
Описание . . . . . : Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC  
Физический адрес . . . . . : 94-08-53-47-36-D7  
DHCP включен . . . . . : Да  
Автонастройка включена . . . . . : Да  
Локальный IPv6-адрес канала . . . . . : fe80::1411:3b7:b2eb:fc0a%19(Основной)  
IPv4-адрес . . . . . : 192.168.170.67(Основной)  
Маска подсети . . . . . : 255.255.255.0  
Аренда получена . . . . . : 9 октября 2024 г. 17:12:22  
Срок аренды истекает . . . . . : 9 октября 2024 г. 19:07:19  
Основной шлюз . . . . . : 192.168.170.5  
DHCP-сервер . . . . . : 192.168.170.5  
IAID DHCPv6 . . . . . : 177473619  
DUID клиента DHCPv6 . . . . . : 00-01-00-01-2A-93-94-9C-94-08-53-47-36-D7  
DNS-серверы . . . . . : 192.168.170.5  
NetBIOS через TCP/IP . . . . . : Включен

Рис. 2.2: Команда ipconfig /all

MAC-адрес для второго виртуального адаптера: D6-08-53-47-36-D7. Переводим первый байт в двоичный код D6 = 11010110. Этот адрес является индивидуальным и локально администрируемым (поэтому по нему нельзя узнать производителя).

MAC-адрес для беспроводной сети WI-FI: 94-08-53-43-36-D7. Переводим первый байт в двоичный код  $94 = 10010100$ . Этот адрес является индивидуальным и глобально администрируемым (производитель Liteon Technology Corporation).

Из предыдущего задания мы узнали адрес основного шлюза: 192.168.170.5. Теперь пропингуем его (рис. 2.3), предварительно запустив Wireshark и включив захват трафика. Посылаются 4 пакета, 4 пакета получено назад.

```
C:\WINDOWS\system32>ping 192.168.170.5

Обмен пакетами с 192.168.170.5 по с 32 байтами данных:
Ответ от 192.168.170.5: число байт=32 время=92мс TTL=64
Ответ от 192.168.170.5: число байт=32 время=11мс TTL=64
Ответ от 192.168.170.5: число байт=32 время=73мс TTL=64
Ответ от 192.168.170.5: число байт=32 время=6мс TTL=64

Статистика Ping для 192.168.170.5:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
        (0% потеря)
Приблизительное время приема-передачи в мс:
    Минимальное = 6мсек, Максимальное = 92 мсек, Среднее = 45 мсек
```

Рис. 2.3: Пингование шлюза

В строке фильтра пропишем фильтр icmp. Убедимся, что в списке пакетов отобразятся только пакеты ICMP, в частности пакеты, которые были сгенерированы с помощью команды ping, отправленной с моего устройства на шлюз по умолчанию. Изучим эхо-запрос (рис. 2.4) и эхо-ответ ICMP (рис. 2.5) в программе Wireshark: 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 (74 бита), тип Ethernet – Ethernet (1).

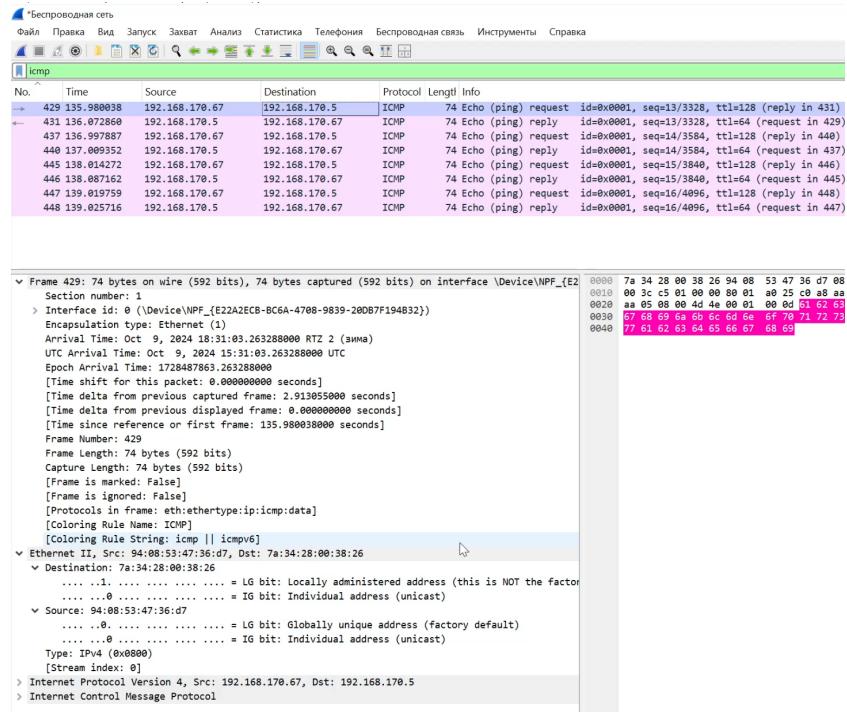


Рис. 2.4: Пакеты ICMP. Кадр физического уровня

Чтобы узнать MAC-адрес источника и шлюза перейдем на канальный уровень.

Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства (94-08-53-43-36-D7). Адрес шлюза (destination, то куда отправлен запрос) - 7A-34-28-00-38-26. Тип адреса тут указан (показаны нулевые и первые биты MAC-адресов). Адрес источника индивидуальный и глобально администрируемый, адрес шлюза индивидуальный и локально администрируемый. Далее посмотрим на полученный ответ. Тут все почти то же самое, что и в запросе (длина кадра 74 бита). Только теперь MAC-адрес источника - MAC-адрес шлюза (7A-34-28-00-38-26), а адрес назначения – адрес моего устройства (94-08-53-43-36-D7).

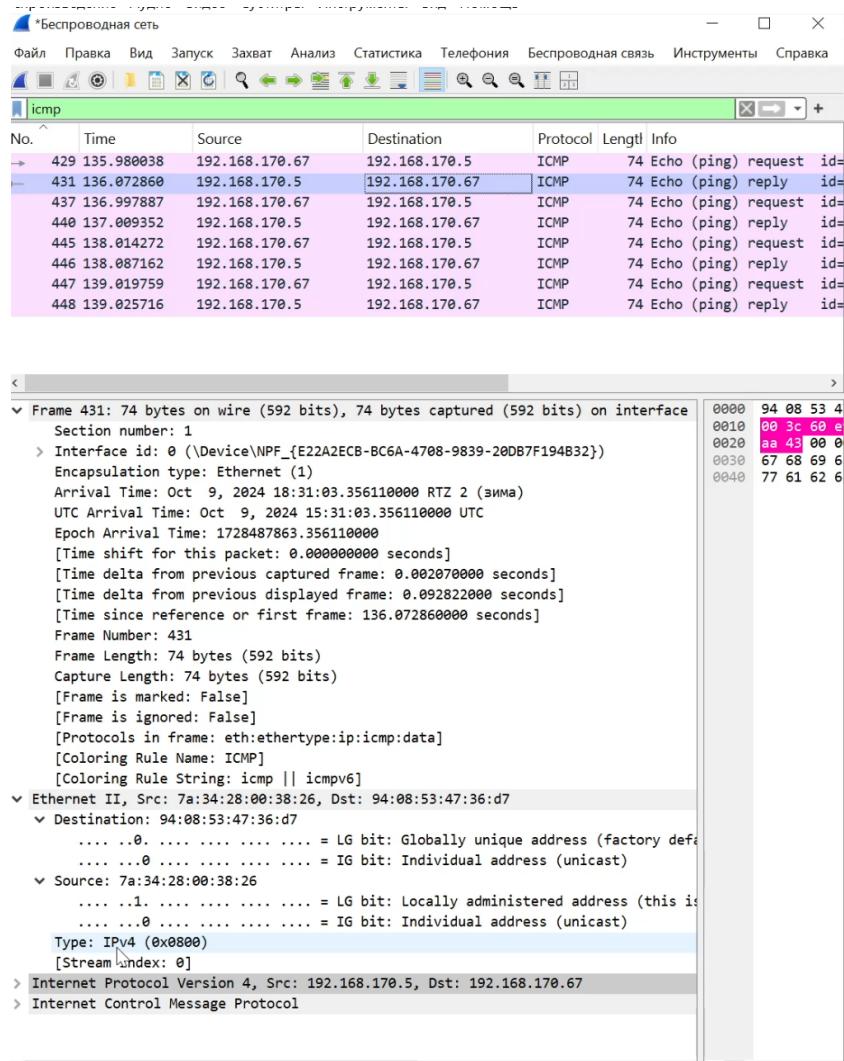


Рис. 2.5: Эхо-ответ. Кадр канального уровня

Изучим кадры данных протокола ARP (рис. 2.6). 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. Изучим данные в полях заголовка Ethernet II. Здесь указаны MAC-адреса источника и получателя. Получатель в нашем случае – индивидуальный и локально администрируемый адрес. Источник – адрес нашего шлюза (индивидуальный и глобально администрируемый).

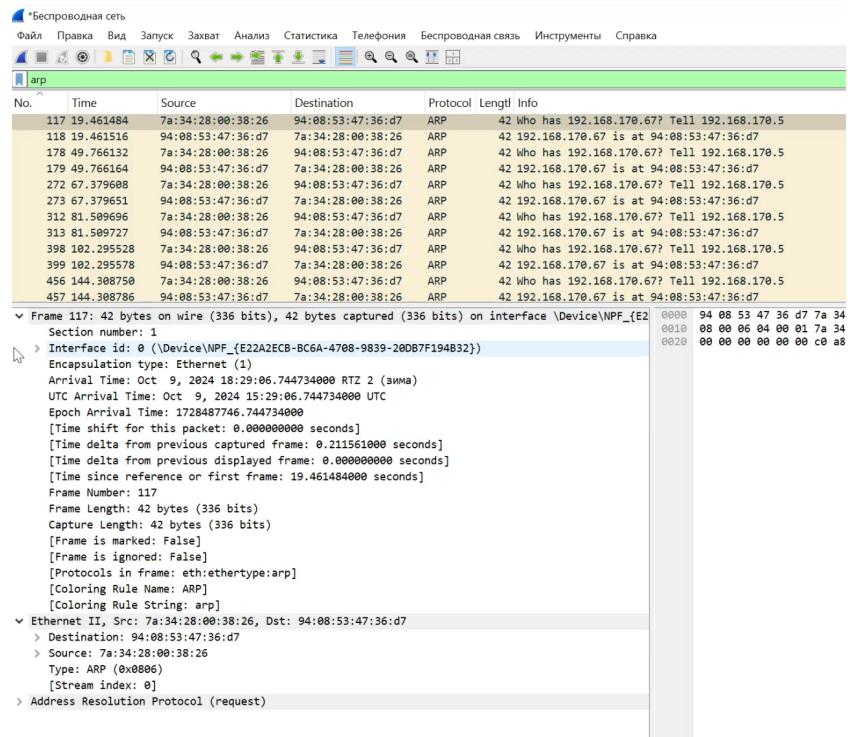


Рис. 2.6: Протокол ARP

Начнем новый процесс захвата трафика в Wireshark. Пропингуем сайт яндекса (рис. 2.7). Изучим запрос протокола ICMP (рис. 2.8). Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства (94-08-53-43-36-D7) - индивидуальный и глобально администрируемый. Адрес получателя (destination, то куда отправлен запрос) - 7A-34-28-00-38-26 - индивидуальный и локальный администрируемый.

```
C:\WINDOWS\system32>ping www.yandex.ru

Обмен пакетами с www.yandex.ru [77.88.44.5] с 32 байтами данных:
Ответ от 77.88.44.55: число байт=32 время=93мс TTL=52
Ответ от 77.88.44.55: число байт=32 время=85мс TTL=52
Ответ от 77.88.44.55: число байт=32 время=209мс TTL=52
Ответ от 77.88.44.55: число байт=32 время=228мс TTL=52

Статистика Ping для 77.88.44.55:
Пакетов: отправлено = 4, получено = 4, потеряно = 0
(0% потеря)
Приблизительное время приема-передачи в мс:
Минимальное = 85мсек, Максимальное = 228 мсек, Среднее = 153 мсек

C:\WINDOWS\system32>
```

Рис. 2.7: Пингование сайта www.yandex.ru

```
*Беспроводная сеть
Файл Правка Вид Запуск Захват Анализ Статистика Телефония Беспроводная связь Инструменты Справка
icmp

No. Time Source Destination Protocol Length Info
→ 28 15.310692 192.168.170.67 77.88.44.55 ICMP 74 Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 29)
← 29 15.403616 77.88.44.55 192.168.170.67 ICMP 74 Echo (ping) reply id=0x0001, seq=20/5120, ttl=52 (request in 28)
30 16.331555 192.168.170.67 77.88.44.55 ICMP 74 Echo (ping) request id=0x0001, seq=21/5376, ttl=128 (reply in 31)
31 16.416408 77.88.44.55 192.168.170.67 ICMP 74 Echo (ping) reply id=0x0001, seq=21/5376, ttl=52 (request in 30)
32 17.345647 192.168.170.67 77.88.44.55 ICMP 74 Echo (ping) request id=0x0001, seq=22/5632, ttl=128 (reply in 33)
33 17.554624 77.88.44.55 192.168.170.67 ICMP 74 Echo (ping) reply id=0x0001, seq=22/5632, ttl=52 (request in 32)
34 18.349021 192.168.170.67 77.88.44.55 ICMP 74 Echo (ping) request id=0x0001, seq=23/5888, ttl=128 (reply in 35)
35 18.577386 77.88.44.55 192.168.170.67 ICMP 74 Echo (ping) reply id=0x0001, seq=23/5888, ttl=52 (request in 34)

Frame 28: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{E22A2ECB-BC6A-4708-9839-20087F194B32}, id 0
Section number: 1
> Interface id: 0 (\Device\NPF_{E22A2ECB-BC6A-4708-9839-20087F194B32})
Encapsulation type: Ethernet (3)
Arrival Time: Oct 9, 2024 18:46:40.532165000 RTZ 2 (зима)
UTC Arrival Time: Oct 9, 2024 15:46:40.532166000 UTC
Epoch Arrival Time: 1738488000.532166000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 15.310692000 seconds]
Frame Number: 28
Frame Length: 74 bytes (592 bits)
Capture Length: 74 bytes (592 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocol in frame: ethertype:ip:icmp:data]
[Coloring Rule Name: ICMP]
[Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: 08:00:53:47:36:D7, Dst: 7A:34:28:00:38:26
Destination: 7A:34:28:00:38:26
    ....1.... .... .... .... = 16 bit: Locally administered address (this is NOT the factory default)
    ....0.... .... .... .... = 16 bit: Individual address (unicast)
Source: 94:08:53:47:36:D7
    ....0.... .... .... .... = 16 bit: Globally unique address (factory default)
    ....0.... .... .... .... = 16 bit: Individual address (unicast)
Type: IPv4 (0x0800)
[Stream index: 0]
> Internet Protocol Version 4, Src: 192.168.170.67, Dst: 77.88.44.55
> Internet Control Message Protocol
```

Рис. 2.8: Запрос протокола ICMP

Изучим запрос протокола ICMP (рис. 2.9). Адрес источника (Source, то куда откуда запрос отправлен) - 7A-34-28-00-38-26. Адрес получателя (Destination, то куда отправлен запрос) – это адрес моего устройства (94-08-53-43-36-D7).

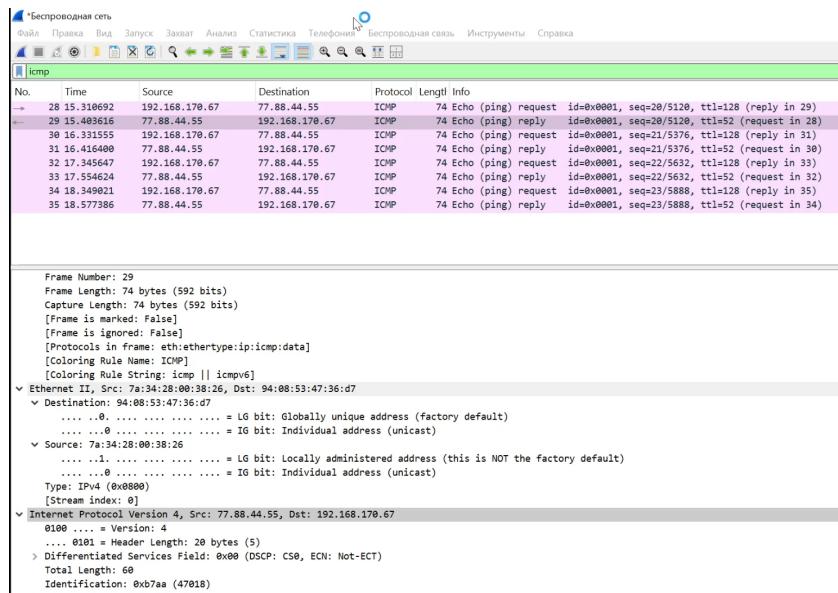


Рис. 2.9: Ответ протокола ICMP

В браузере перейдем на сайт, работающий по протоколу HTTP (например, на сайт CERN <http://info.cern.ch/>). Для получения большей информации для Wireshark попремещались по ссылкам или разделам сайта в браузере. В Wireshark в строке фильтра укажем http и проанализируем информацию по протоколу TCP в случае запросов (рис. 2.10) и ответов (рис. 2.12). Открываем раздел протокола TCP в случае запроса. Видим, что порт получателя – 80 (это стандартный порт для http). Порт источника - 26392 (он определяется случайнным образом из незанятых и непривилегированных портов). Также тут есть поле Порядковый номер (Sequence Number) и поле Номер подтверждения (Acknowledgment Number).

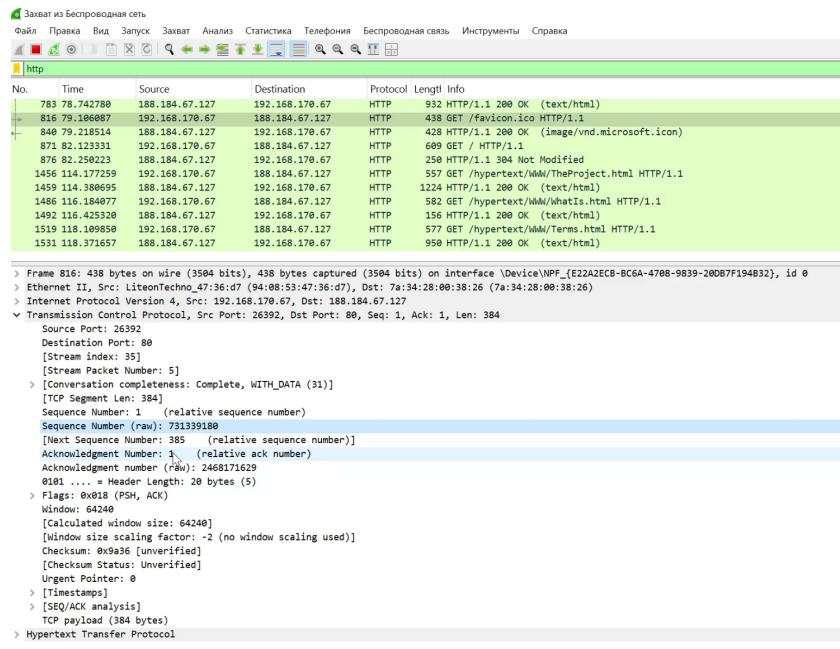


Рис. 2.10: Протокол TCP

Далее рассмотрим ответ. Здесь у нас поменялись местами порты источника и получателя. Теперь порт источника – порт сайта (80). А порт получателя - 26392 (выбранный случайным образом). В Wireshark в строке фильтра укажем dns и проанализируем информацию по протоколу UDP в случае запросов и ответов. В случае запроса (рис. 2.11): порт источника – 59667 (выбранный случайным образом из незанятых и непrivилегированных портов). Порт получателя - 53.

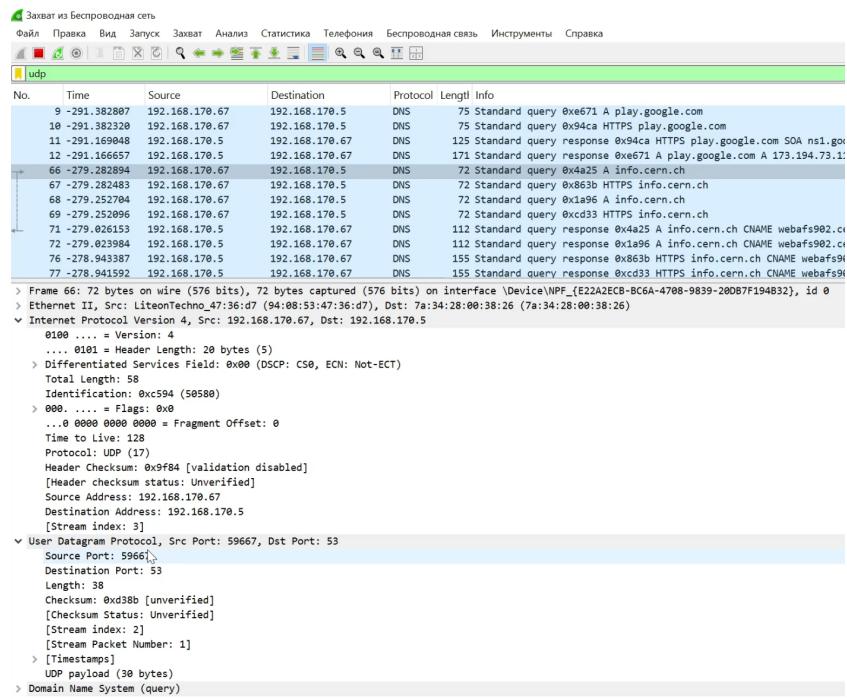


Рис. 2.11: Протокол UDP

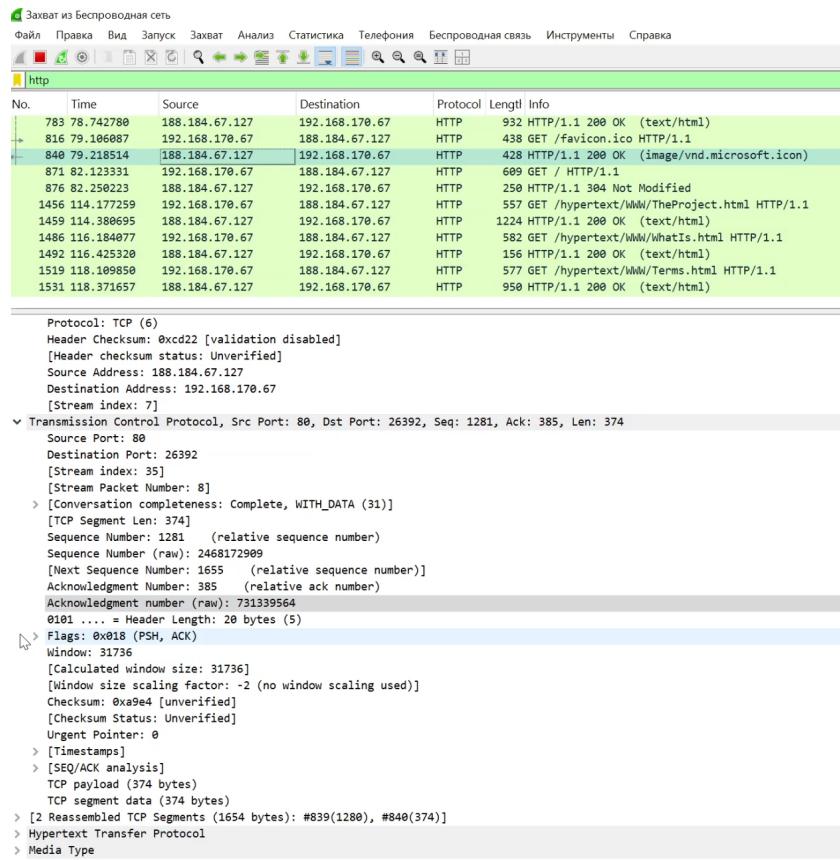


Рис. 2.12: Протокол TCP

В случае ответа (рис. 2.13) порт источника - 53, а порт получателя - 59667.

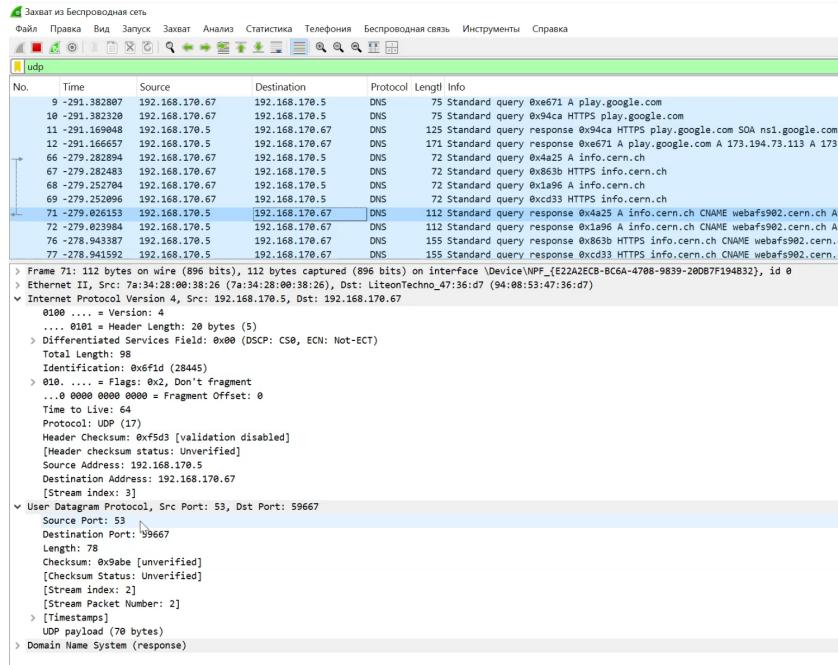


Рис. 2.13: Протокол UDP (случай ответа)

В Wireshark в строке фильтра укажем `quic` и проанализируем информацию по протоколу `quic` в случае запросов (рис. 2.14) и ответов (рис. 2.15). Как и в случае `dns` можем посмотреть информацию транспортного уровня по протоколу UDP. Порт источника задан случайно, выбором из непривилегированных и незанятых портов, и равен 54133, порт получателя равен 443 - 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.

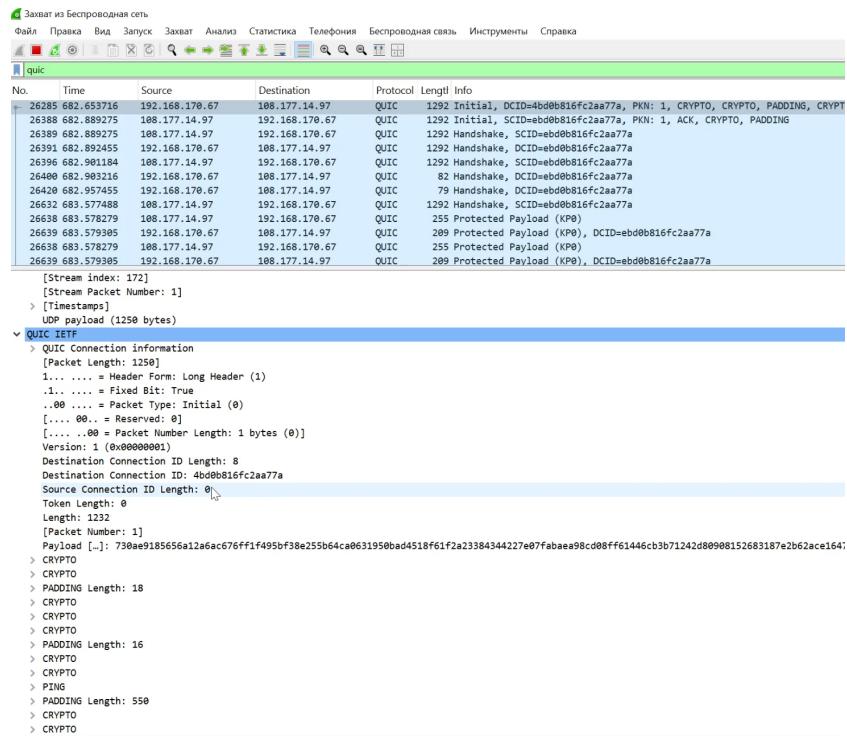


Рис. 2.14: Запрос quic

В случае ответа порты заданы наоборот, то есть источник - 443 порт, получатель – 54133.

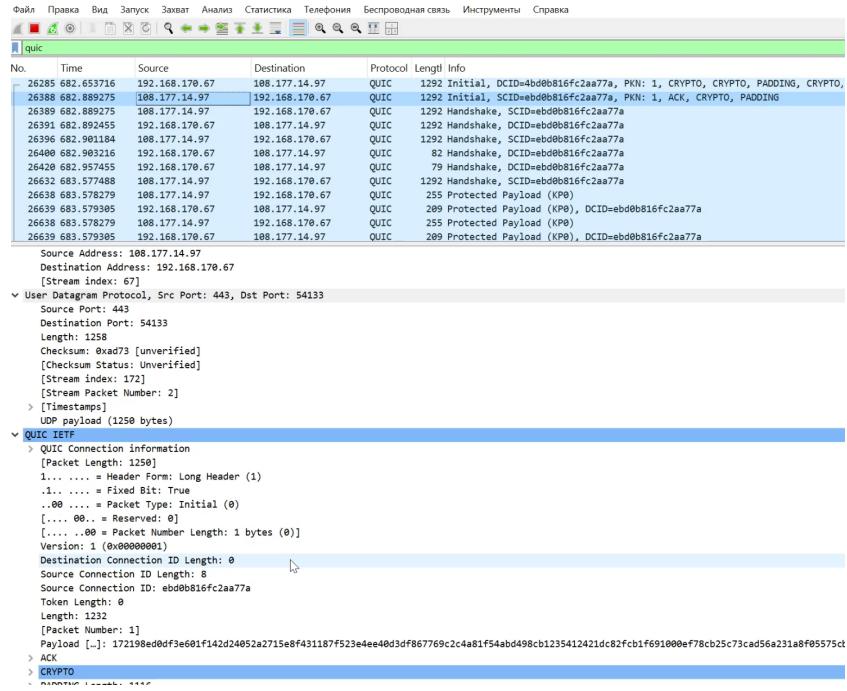


Рис. 2.15: Ответ quic

На моем устройстве используем соединение по HTTP с каким-то сайтом для захвата в Wireshark пакетов TCP. Проанализируем handshake протокола TCP. Режим активного доступа (Active Open) (рис. 2.16). Клиент посыпает 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 1820833647 (ISSa), значение Acknowledgment Number равно 0. Также видим, что установлен флаг SYN.

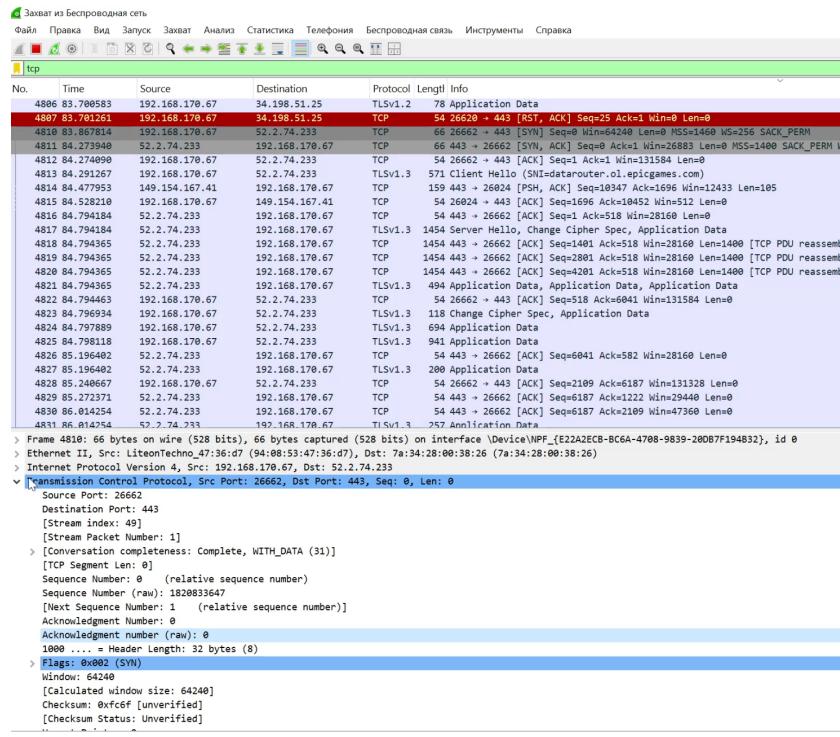


Рис. 2.16: Протокол TCP для первой ступени handshake

Режим пассивного доступа (Passive Open) (рис. 2.17). Сервер откликается, посыпая сообщение SYN, ACK, ISSb, ACK(ISSa+1), т.е. установлены биты SYN и ACK; в поле Порядковый номер (Sequence Number) хостом В устанавливается начальное значение счётчика – ISSb; поле Номер подтверждения (Acknowledgment Number) содержит значение ISSa, полученное в первом пакете от хоста А и увеличенное на единицу. Действительно, Acknowledgment Number равно 1820833647 (значение ISSa) + 1 = 1820833648. Sequence Number равен 492681311 (начальное значение счётчика – ISSb). Установлены флаги SYN, ACK.

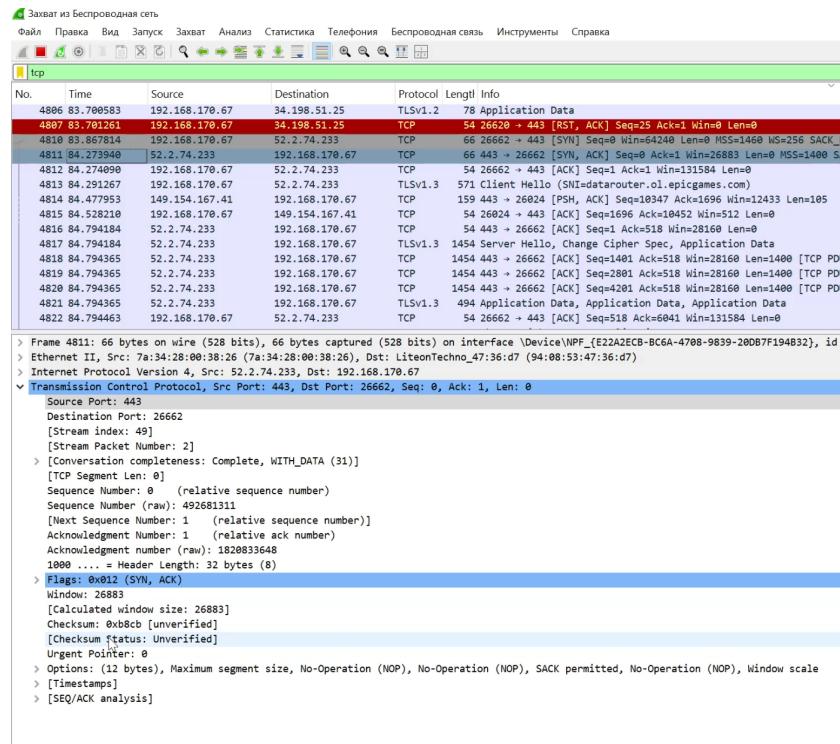


Рис. 2.17: Протокол TCP для второй ступени handshake

Завершение рукопожатия (рис. 2.18). Клиент отправляет подтверждение получения SYN сегмента от сервера с идентификатором, равным ISN (сервера)+1: ACK, ISS<sub>a</sub>+1, ACK(ISS<sub>b</sub>+1). В этом пакете установлен бит ACK, поле Порядковый номер (Sequence Number) содержит значение ISS<sub>a</sub>+1, поле Номер подтверждения (Acknowledgment Number) содержит значение ISS<sub>b</sub>+1. Посылкой этого пакета заканчивается трёхступенчатый handshake, и TCP соединение считается установленным. Действительно, Acknowledgment Number равно 492681311 (значение ISS<sub>b</sub>) + 1 = 492681312. Sequence Number равен 1820833647 (значение ISS<sub>a</sub>) + 1 = 1820833648. Установлен флаг ACK.

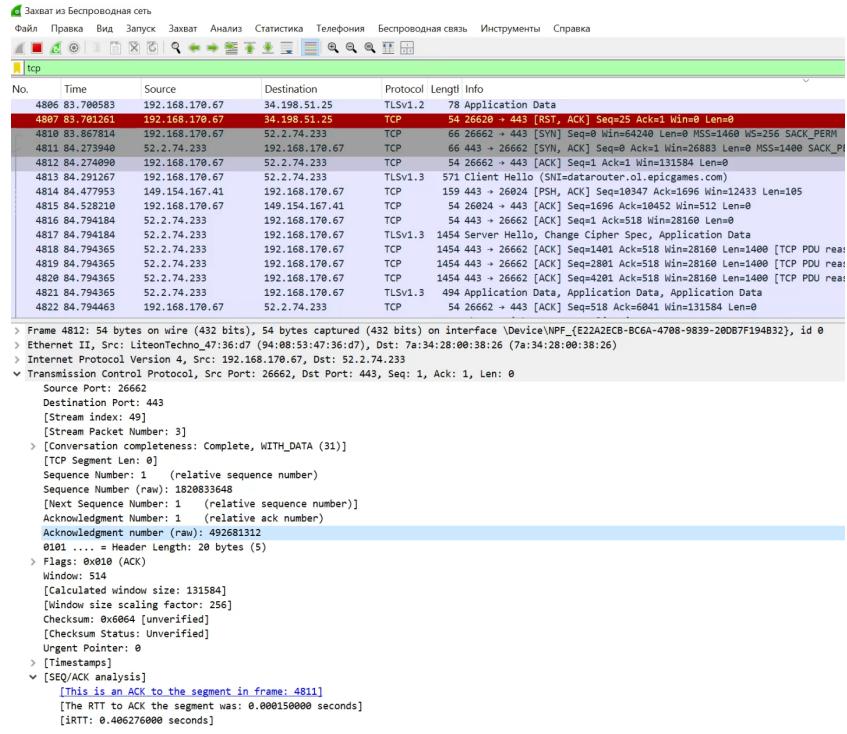


Рис. 2.18: Протокол TCP для третьей ступени handshake

Далее посмотрим график потока (рис. 2.19). Здесь в принципе все то же самое, что мы уже разобрали, только на графике. Клиент посыпает запрос серверу (установлен бит SYN), Seq = 0. Далее сервер отвечает клиенту (установлены биты SYN, ACK), Seq = 0, Ack = 1 (это относительные значения). Завершение рукопожатия: клиент отправляет серверу подтверждение получения SYN сегмента, Seq = 1, Ack = 1.

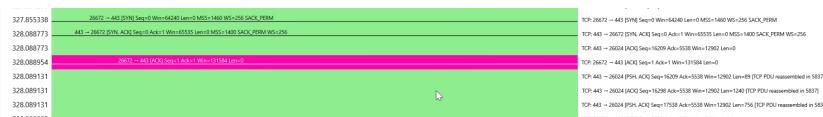


Рис. 2.19: График потока

## **3 Выводы**

В ходе лабораторной работы я изучила посредством Wireshark кадры Ethernet, анализ PDU протоколов транспортного и прикладного уровней стека TCP/IP.