РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Лабораторная работа №3

дисциплина: Реляционные базы данных

Студент: Сахно Никита

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

С помощью команды ipconfig для ОС типа Windows выведем информацию о текущем сетевом соединении. Мой провайдер – beeline. 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.

```
C:\WINDOWS\system32>ipconfig

Настройка протокола IP для Windows


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 . . . . . : beeline
   Локальный IPv6-адрес канала . . . : fe80::4c08:7b9d:d7e4:beba%16
   IPv4-адрес. . . . . . . . . . . . : 192.168.1.72
   Маска подсети . . . . . . . . . . : 255.255.255.0
   Основной шлюз. . . . . . . . . : 192.168.1.1
```

Рисунок 1. Команда ipconfig

Используем также опцию /all для вывода более подробной информации.

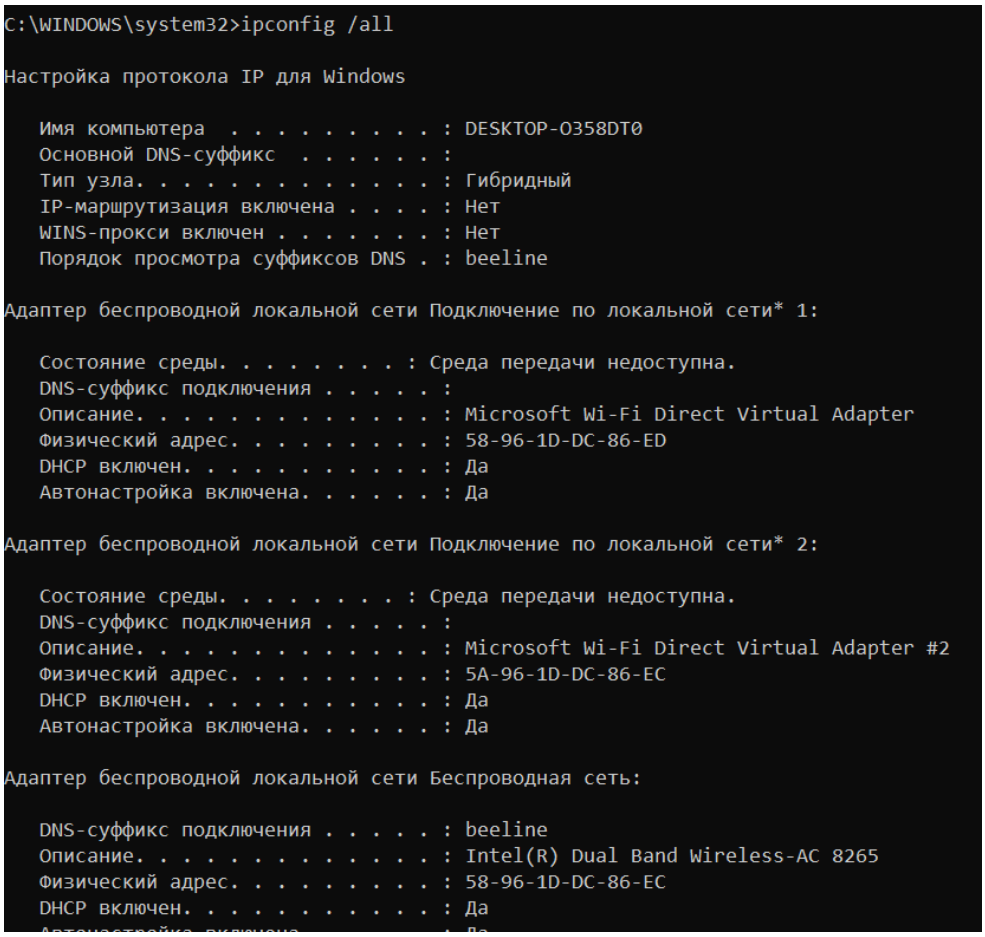
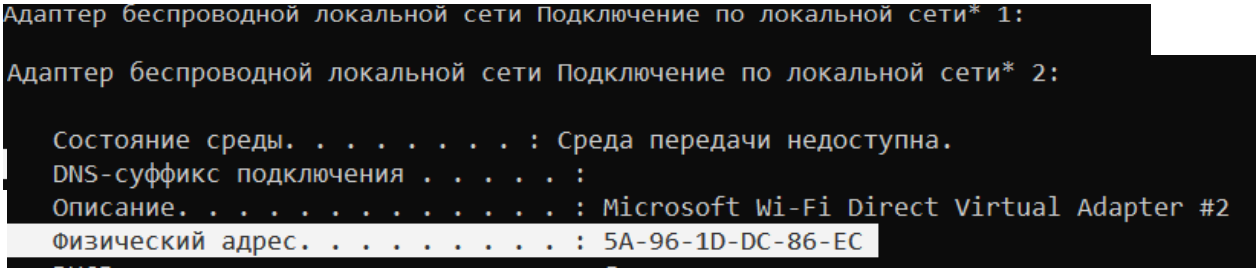


Рисунок 2. Команда ipconfig /all

Определим MAC-адреса сетевых интерфейсов на моем компьютере.

У меня есть помимо основной беспроводной сети WI-FI еще две локальные сети. MAC-адрес для первого виртуального адаптера: 58-96-1D-DC-86-ED. MAC-адрес состоит из 6 октетов: первые 3 октета идентифицируют производителя, последние 3 октета идентифицируют сетевой интерфейс. Я проверила на специальном сайте, что у меня производитель INTEL. Далее разберем первый байт MAC-адреса (58), переведем в двоичный код 58 = 01011000. Нас интересуют последние два бита (нулевой и первый биты). У меня оба нули => мой адрес индивидуальный и глобально администрируемый.

MAC-адрес для второго виртуального адаптера: 5A-96-1D-DC-86-EC.



Переводим первый байт в двоичный код 5A = 01011010. Этот адрес является индивидуальным и локально администрируемым.

MAC-адрес для беспроводной сети WI-FI: 58-96-1D-DC-86-EC. Переводим первый байт в двоичный код 58 = 01011000. Этот адрес является индивидуальным и глобально администрируемым (производитель INTEL).

```
Адаптер беспроводной локальной сети Беспроводная сеть:

   DNS-суффикс подключения . . . . . : beeline
   Описание. . . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
   Физический адрес. . . . . . . . . : 58-96-1D-DC-86-EC
```

Рисунок 4. MAC-адрес в случае WI-FI

№2

Из предыдущего задания мы узнали адрес основного шлюза: 192.168.1.1. Теперь пропингуем его, предварительно запустив Wireshark и включив захват трафика. Посылаются 4 пакета, 4 пакета получено назад.

```
C:\WINDOWS\system32>ping 192.168.1.1

Обмен пакетами с 192.168.1.1 по с 32 байтами данных:
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64

Статистика Ping для 192.168.1.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 1мсек, Максимальное = 1 мсек, Среднее = 1 мсек
```

Рисунок 5. Пингование шлюза

В строке фильтра пропишем фильтр icmp. Убедимся, что в списке

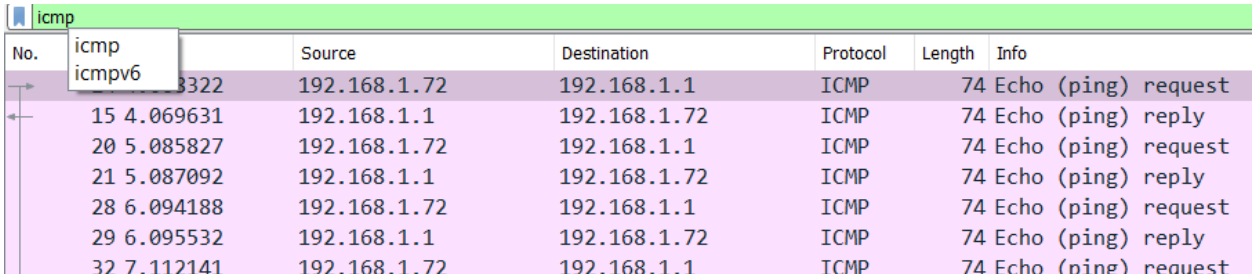


Рисунок 6. Пакеты ICMP

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 и эхо-ответ ICMP 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).

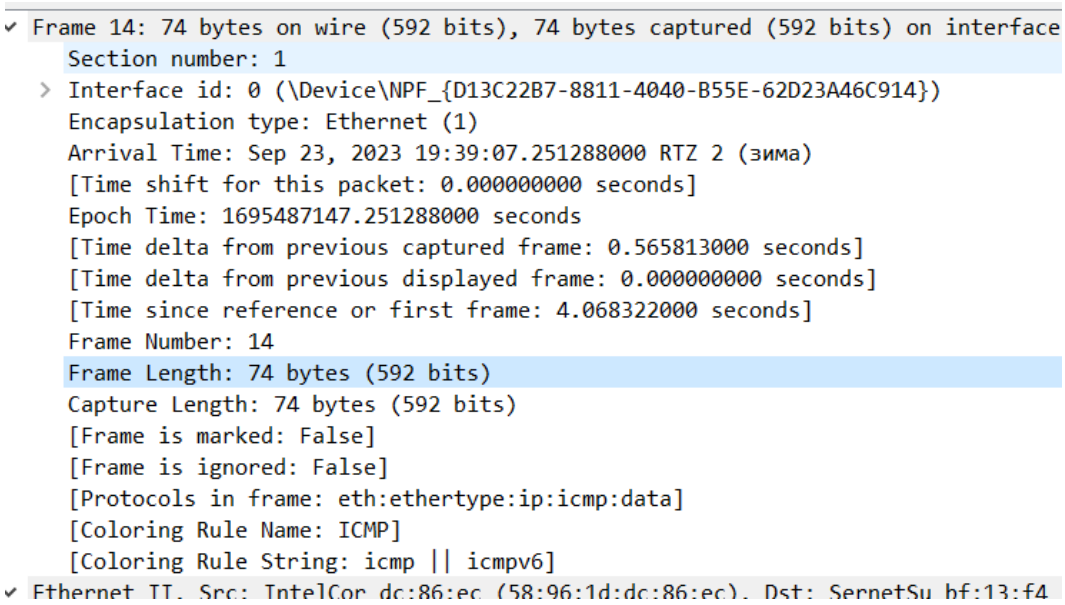


Рисунок 7. Кадр физического уровня

Чтобы узнать MAC-адрес источника и шлюза перейдем на канальный уровень. Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства (58-96-1D-DC-86-EC). Адрес шлюза (destination, то куда отправлен запрос) - 58-76-AC-BF-13-F4. Тип адреса тут указан (показаны нулевые и первые биты MAC-адресов). Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

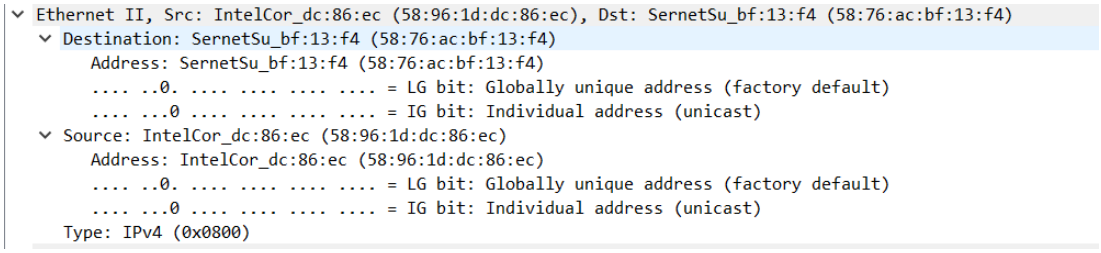


Рисунок 8. Кадр канального уровня

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 (58-76-AC-BF-13-F4), а адрес назначения – адрес моего устройства (58-96-1D-DC-86-EC).

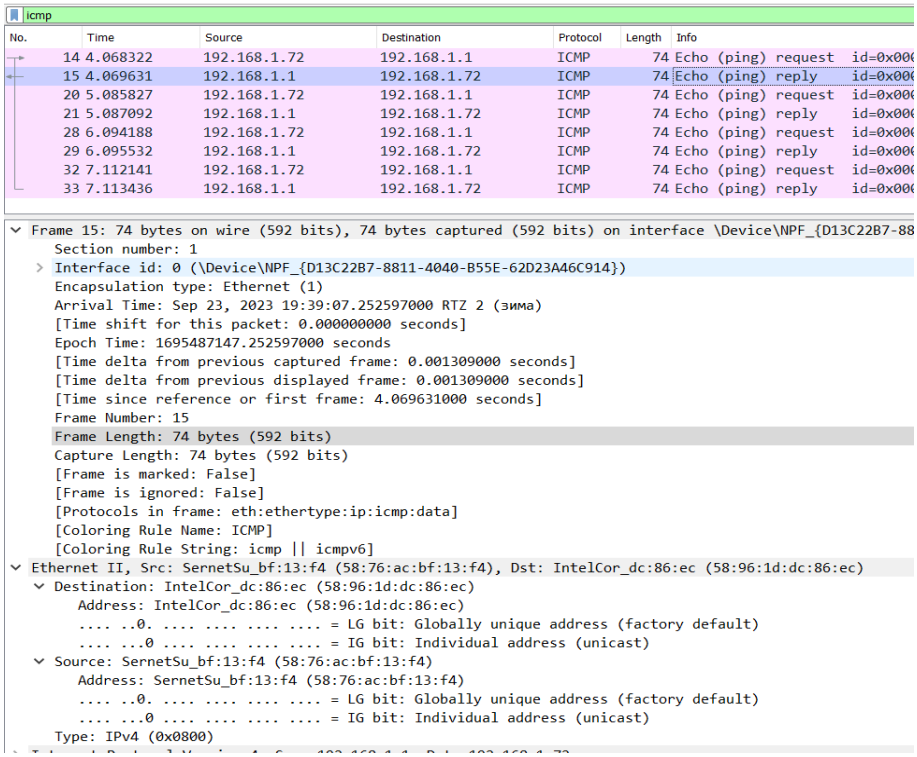


Рисунок 9. Эхо-ответ

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.

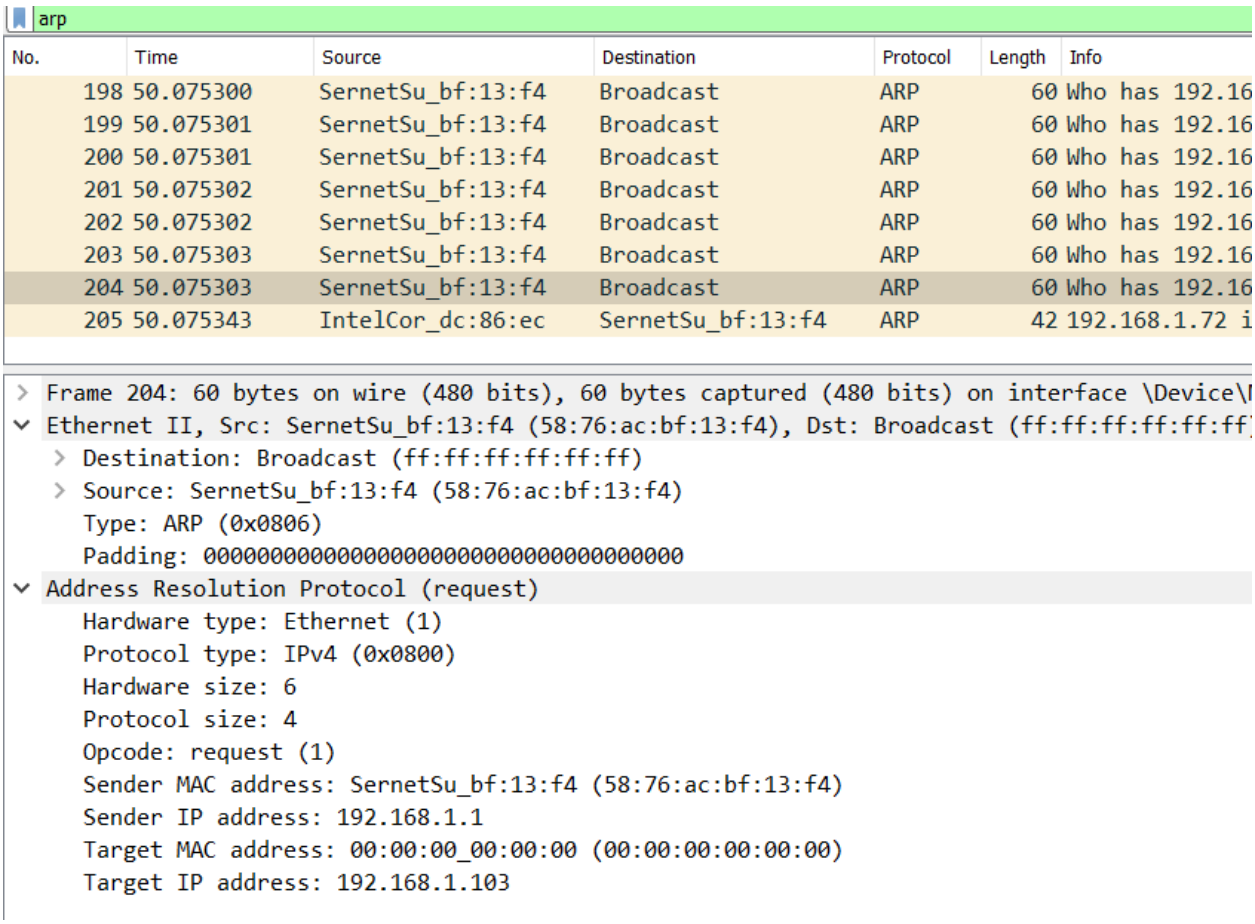


Рисунок 10. Протокол ARP

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).

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

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

Обмен пакетами с www.yandex.ru [77.88.55.88] с 32 байтами данных:
Ответ от 77.88.55.88: число байт=32 время=9мс TTL=55
Ответ от 77.88.55.88: число байт=32 время=9мс TTL=55
Ответ от 77.88.55.88: число байт=32 время=9мс TTL=55
Ответ от 77.88.55.88: число байт=32 время=9мс TTL=55

Статистика Ping для 77.88.55.88:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 9мсек, Максимальное = 9 мсек, Среднее = 9 мсек

C:\WINDOWS\system32>_
```

Рисунок 11. Пингование сайта www.yandex.ru

Изучим запрос протокола ICMP. Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства (58-96-1D-DC-86-EC). Адрес получателя (destination, то куда отправлен запрос) - 58-76-AC-BF-13-F4. Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

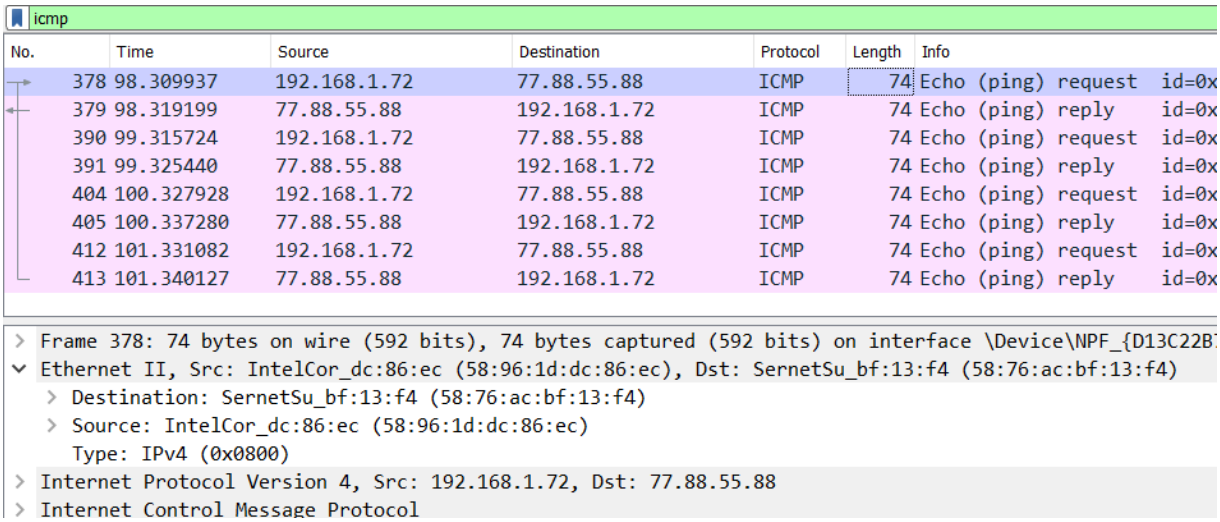


Рисунок 12. Запрос протокола ICMP

Изучим запрос протокола ICMP. Адрес источника (Source, то куда откуда запрос отправлен) - 58-76-AC-BF-13-F4. Адрес получателя (Destination, то куда отправлен запрос) – это адрес моего устройства (58-96-1D-DC-86-EC). Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

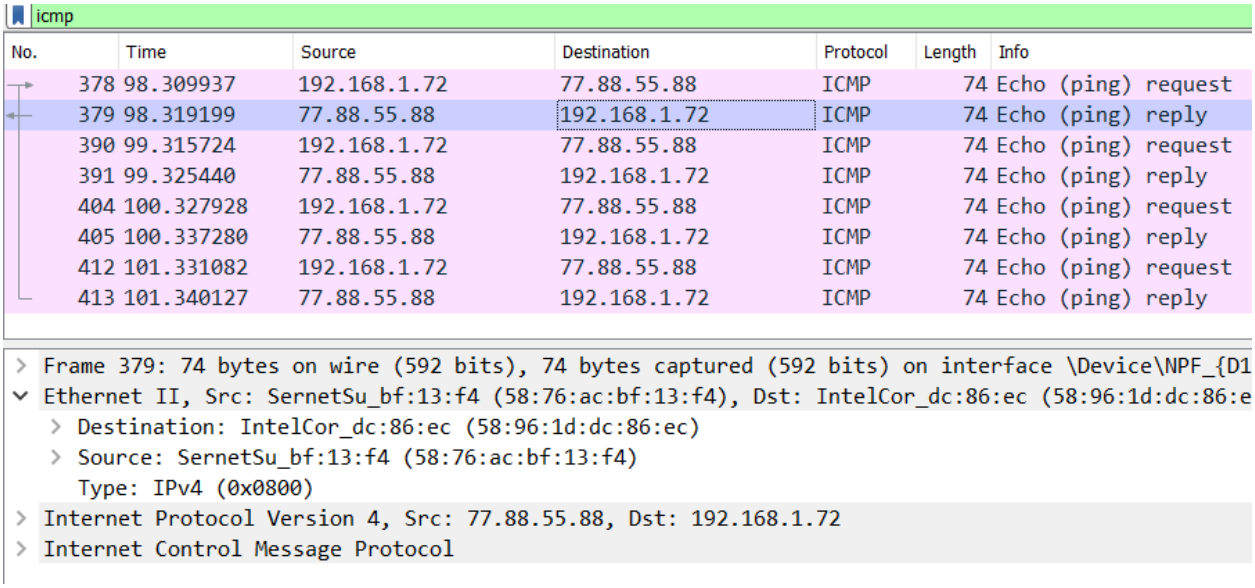


Рисунок 13. Ответ протокола ICMP

№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 – 80 (это стандартный порт для http). Порт источника – 50758 (он определяется случайным образом из незанятых и непривилегированных портов). Также тут есть поле Порядковый номер (Sequence Number) и поле Номер подтверждения (Acknowledgment Number).

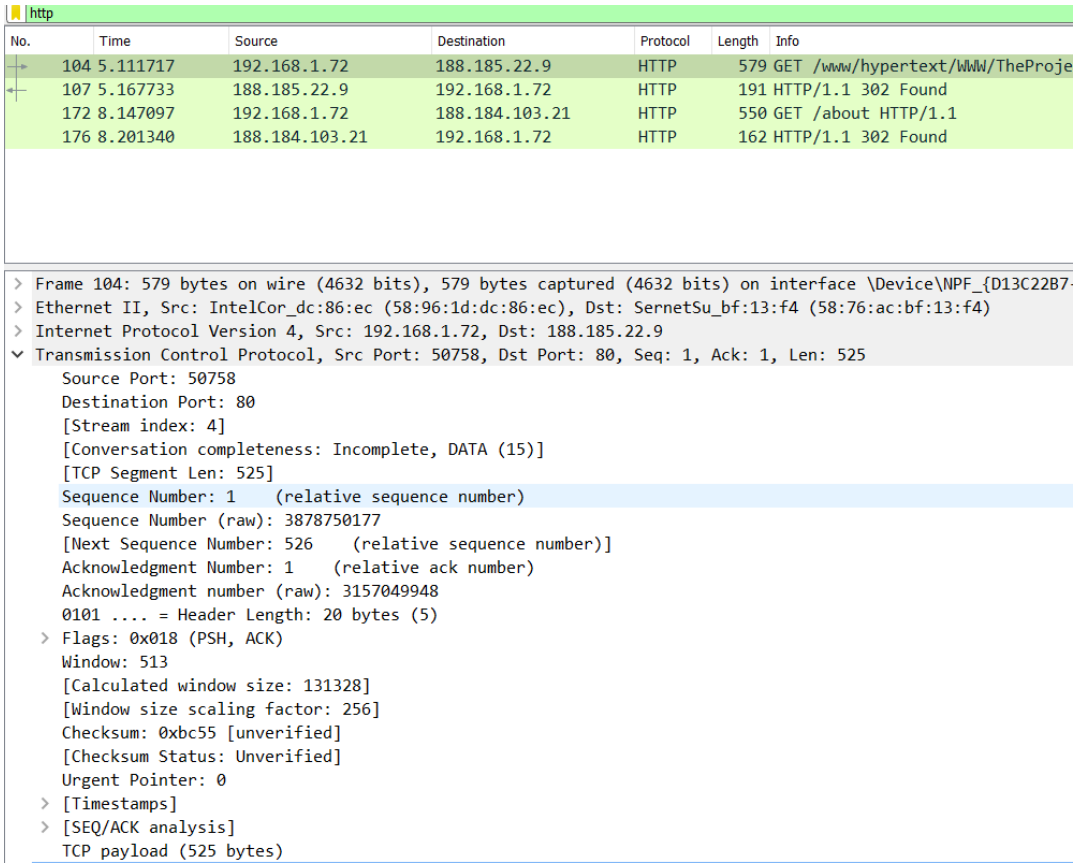


Рисунок 14. Протокол TCP (случай запроса)

Далее рассмотрим ответ. Здесь у нас поменялись местами порты источника и получателя. Теперь порт источника – порт сайта (80). А порт получателя – 50758 (выбранный случайным образом).

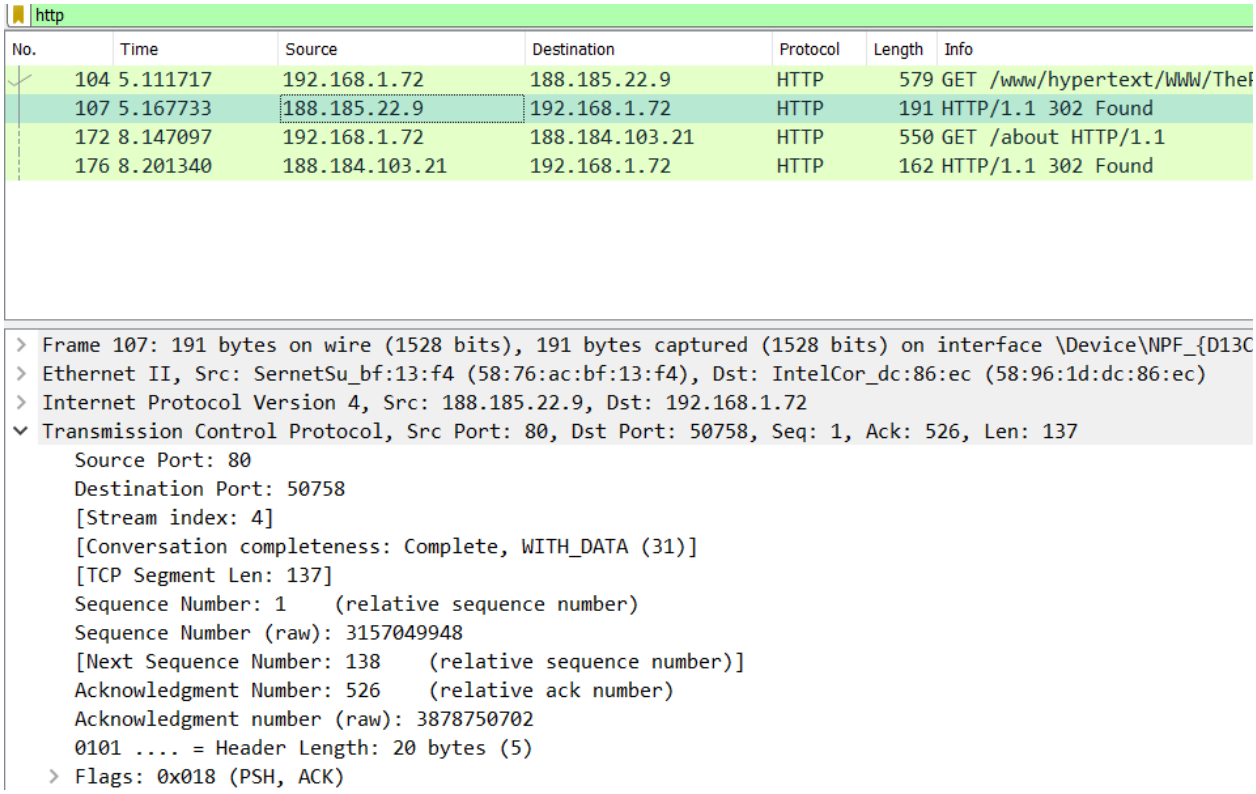


Рисунок 15. Протокол TCP (случай ответа)

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

В случае запроса: порт источника – 58034 (выбранный случайным образом из незанятых и непривилегированных портов). Порт получателя – 53.

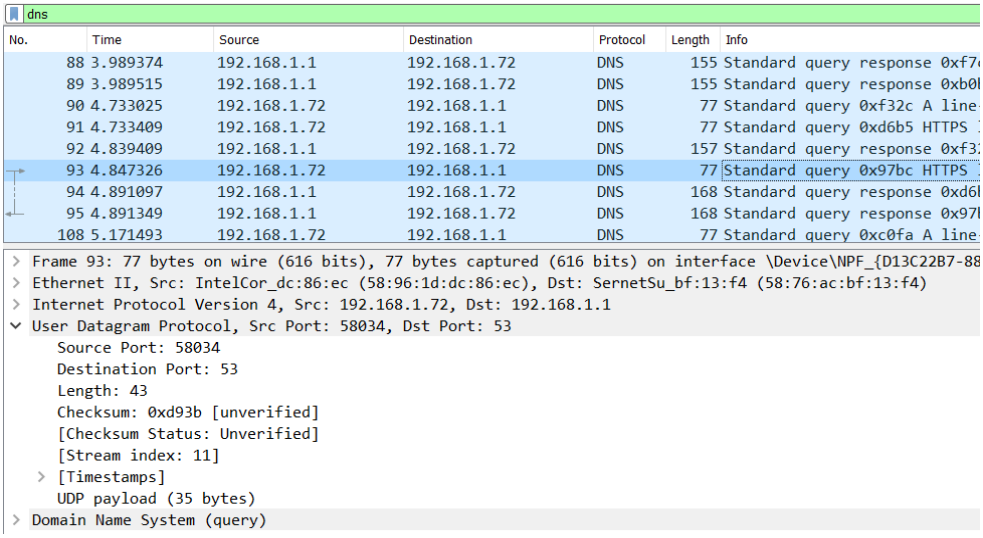


Рисунок 16. Протокол UDP (случай запроса)

В случае ответа порт источника – 53, а порт получателя – 58034.

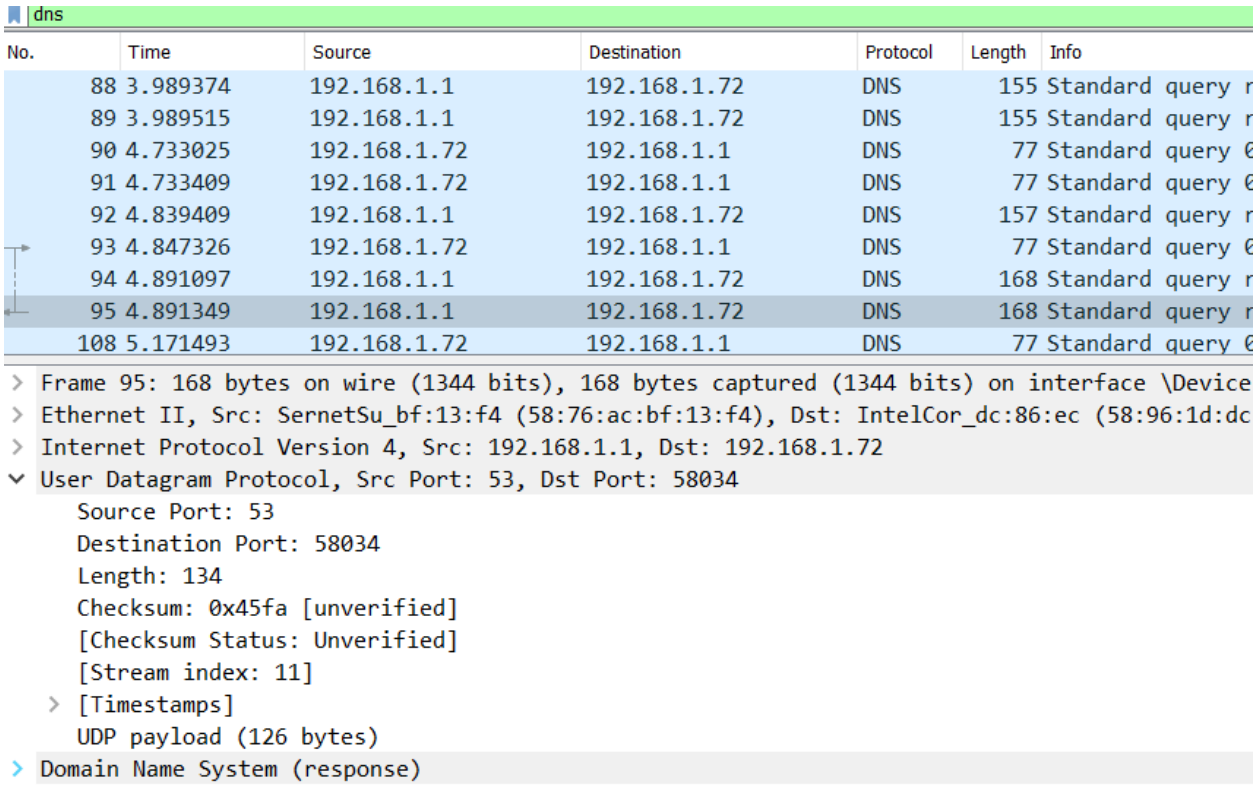


Рисунок 17. Протокол UDP (случай ответа)

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 64952, 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.

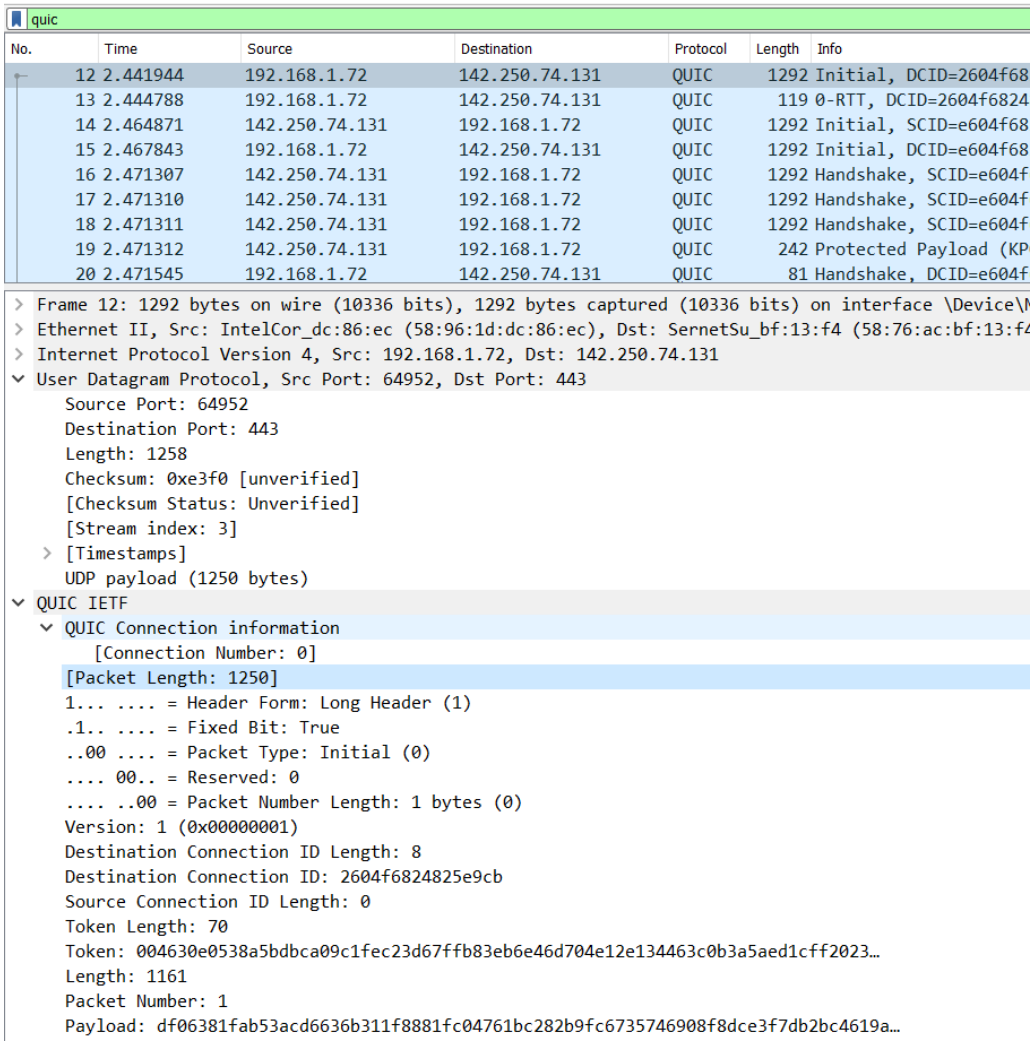


Рисунок 18. Запрос quic

В случае ответа порты заданы наоборот, то есть источник - 443 порт, получатель – 64952.

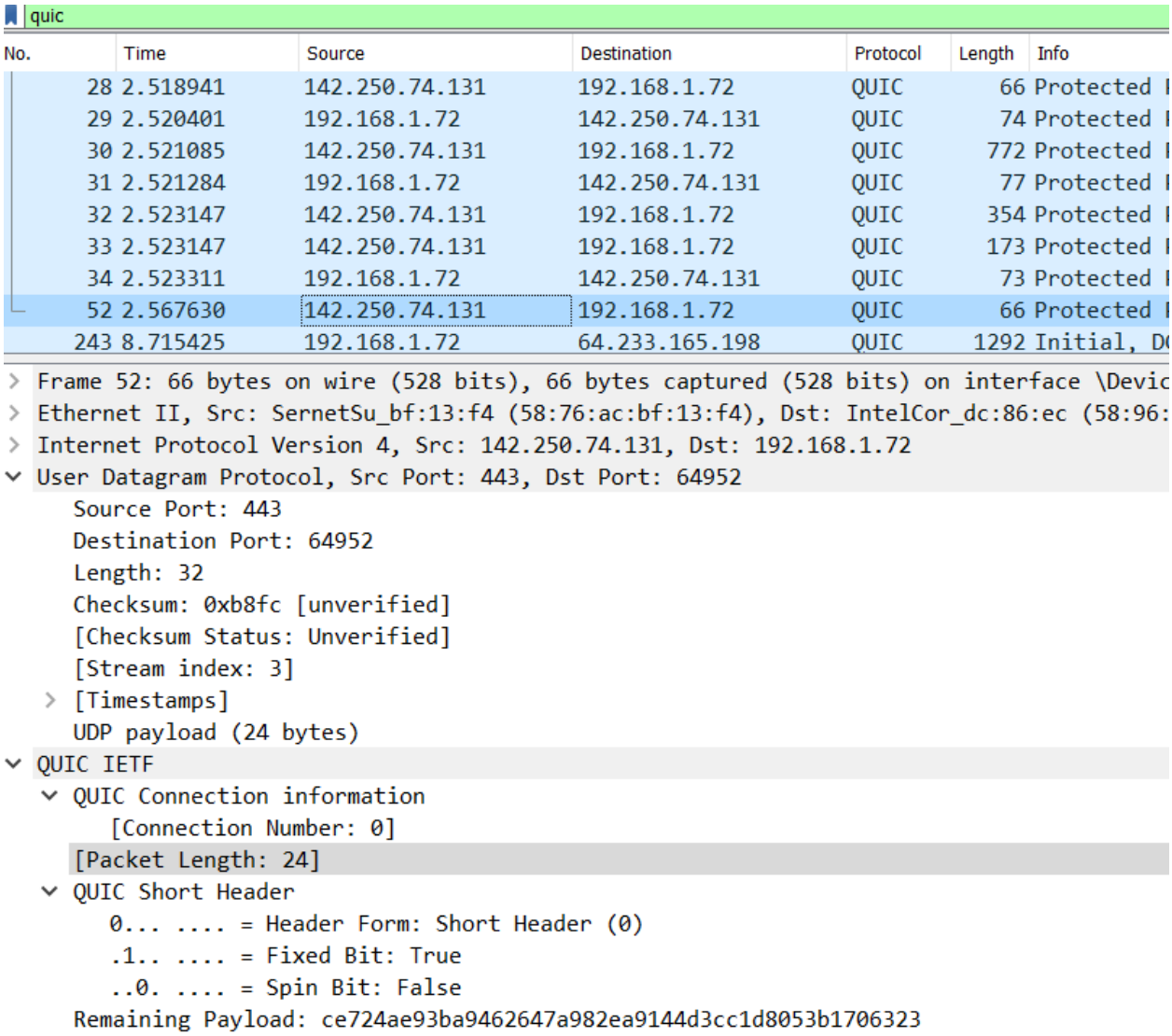
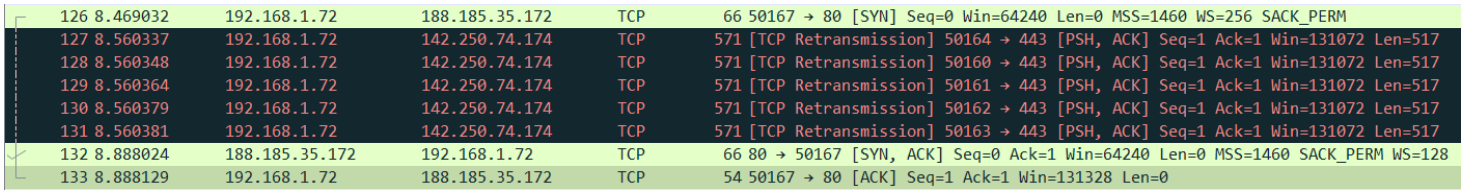


Рисунок 19. Ответ quic

№4

На моем устройстве используем соединение по HTTP с каким-то сайтом для захвата в Wireshark пакетов TCP.

Проанализируем handshake протокола TCP.

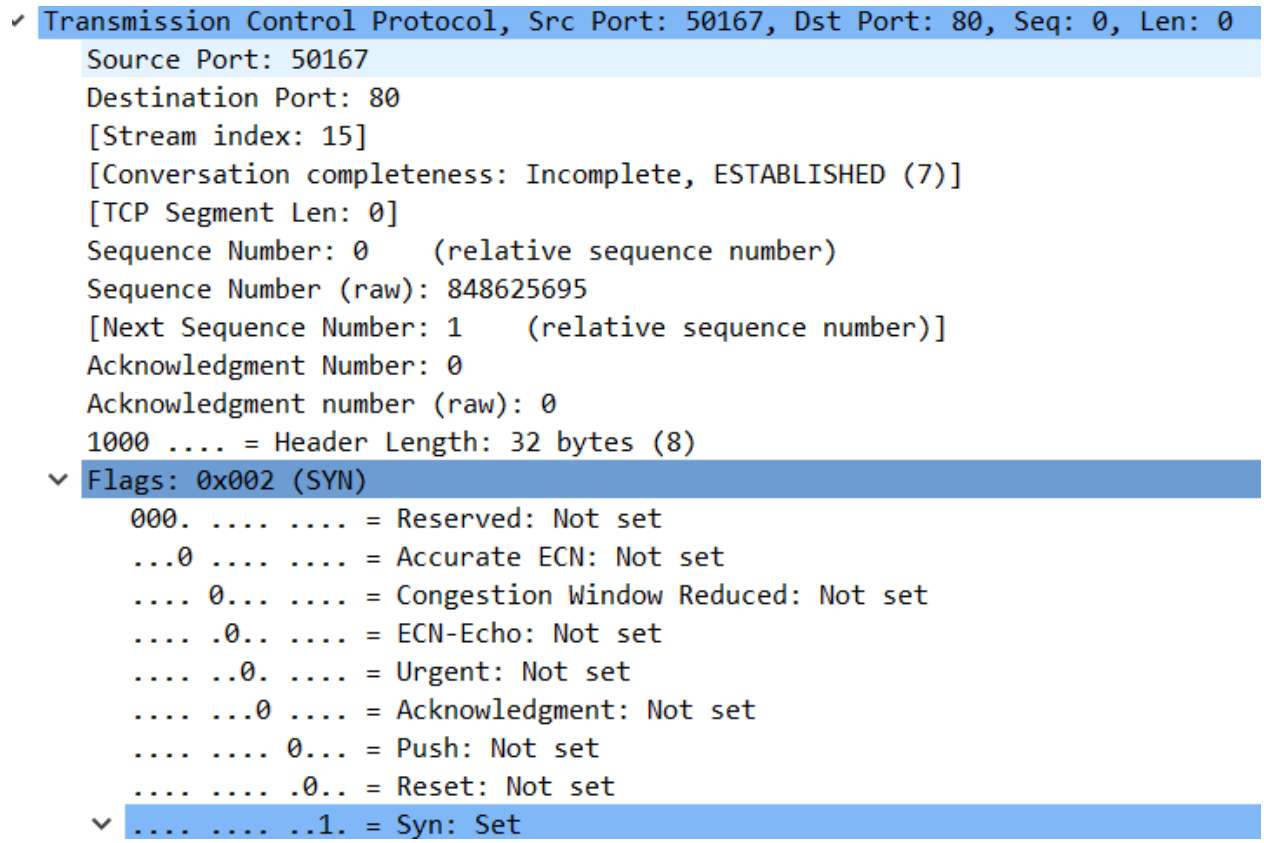


126	8.469032	192.168.1.72	188.185.35.172	TCP	66 50167 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
127	8.560337	192.168.1.72	142.250.74.174	TCP	571 [TCP Retransmission] 50164 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517
128	8.560348	192.168.1.72	142.250.74.174	TCP	571 [TCP Retransmission] 50160 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517
129	8.560364	192.168.1.72	142.250.74.174	TCP	571 [TCP Retransmission] 50161 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517
130	8.560379	192.168.1.72	142.250.74.174	TCP	571 [TCP Retransmission] 50162 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517
131	8.560381	192.168.1.72	142.250.74.174	TCP	571 [TCP Retransmission] 50163 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517
132	8.888024	188.185.35.172	192.168.1.72	TCP	66 80 → 50167 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
133	8.888129	192.168.1.72	188.185.35.172	TCP	54 50167 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0

Рисунок 20. 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 848625695 (ISSa), значение Acknowledgment Number равно 0. Также видим, что установлен флаг SYN.



```
Transmission Control Protocol, Src Port: 50167, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 50167
    Destination Port: 80
    [Stream index: 15]
    [Conversation completeness: Incomplete, ESTABLISHED (7)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 848625695
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    .... .... ..1. = Syn: Set
```

Рисунок 21. 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.

Действительно, Acknowledgment Number равно 848625695 (значение ISSa) + 1 = 848625696. Sequence Number равен 2149321476 (начальное значение счётчика — ISSb).

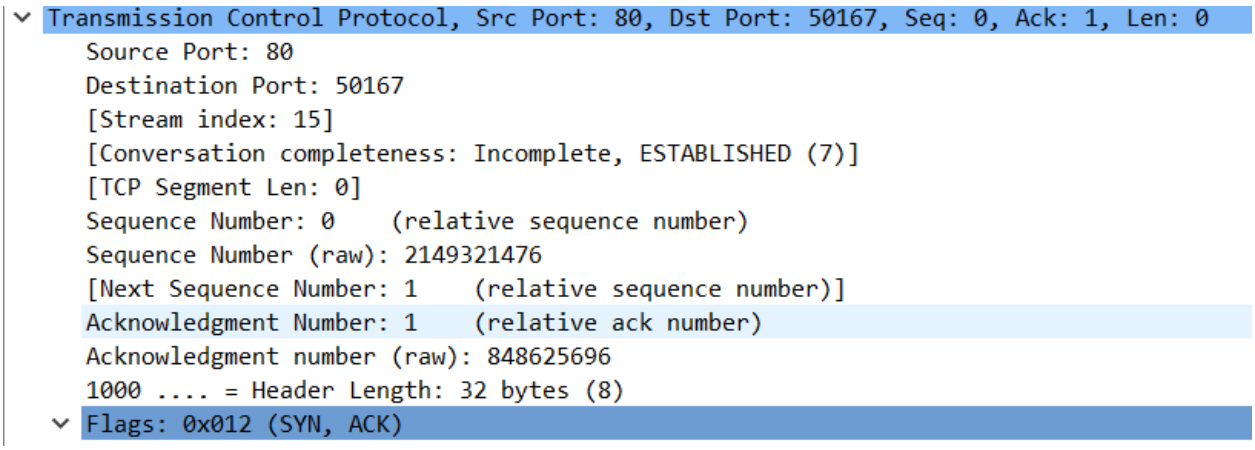


Рисунок 22. Протокол TCP для второй ступени handshake

Установлены флаги SYN, ACK.

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.

Действительно, Acknowledgment Number равно 2149321476 (значение ISSb) + 1 = 2149321477. Sequence Number равен 848625695 (ISSa) + 1 = 848625696. Установлен флаг ACK.

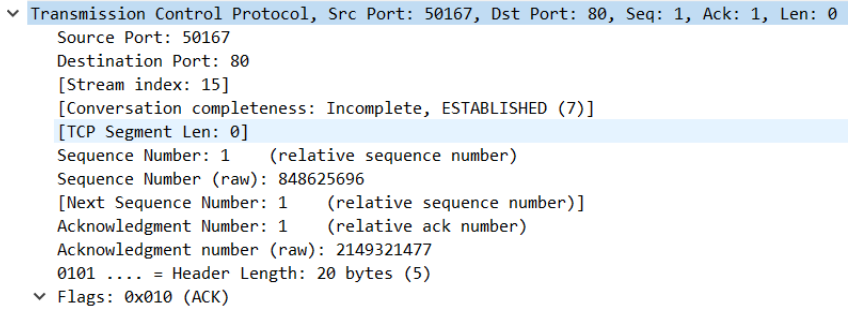


Рисунок 23. Протокол TCP для третьей ступени handshake

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 24. График потока

Выводы

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