

Отчёт по лабораторной работе №3

Дисциплина: Сетевые технологии

Ибрахим Мухсейн Алькамаль

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
2.1 MAC-адресация	6
2.2 Анализ кадров канального уровня в Wireshark	8
2.3 Анализ протоколов транспортного уровня в Wireshark	15
2.4 Анализ handshake протокола TCP в Wireshark	18
3 Выводы	22

Список иллюстраций

2.1	Команда ipconfig	7
2.2	Команда ipconfig /all	8
2.3	Запуск захвата трафика	9
2.4	Пинг шлюза по умолчанию	9
2.5	Кадр ICMP - эхо-запрос: информация о длине кадра, типе Ethernet и MAC-адресах	10
2.6	Кадр ICMP - эхо-ответ: информация о длине кадра, типе Ethernet, MAC-адресах	11
2.7	Кадр ARP: информация о длине кадра, типе Ethernet, MAC-адресах .	12
2.8	Пинг сайта ping rudn.ru	13
2.9	Запрос протокола ICMP	14
2.10	протокола ICMP	15
2.11	Кадр http - запрос	16
2.12	Кадр http - ответ	16
2.13	Кадр dns - запрос	17
2.14	Кадр quic - запрос	18
2.15	Первая ступень handshake TCP	19
2.16	Вторая ступень handshake TCP	20
2.17	Третья ступень handshake TCP	21
2.18	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

2.1 МАС-адресация

С помощью команды ipconfig для ОС типа Windows выводим информацию о текущем сетевом соединении. Просматриваем информацию о сетевых адаптерах и конкретно о беспроводном соединении. Отсюда можно узнать IPv6-адрес, IPv4-адрес (уникальный IPv4-адрес узла), маску подсети (используется для определения сетевой и узловой частей IPv4-адреса) и шлюз (рис. [fig:1]).

```
C:\Users\Ebrahim Alkamal>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Ethernet adapter vEthernet (Default Switch):
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::de9e:9f41:b2c5:3599%41
  IPv4 Address. . . . . : 172.29.0.1
  Subnet Mask . . . . . : 255.255.240.0
  Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 9:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 10:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:
  Connection-specific DNS Suffix . . . : rudn.ru
  IPv4 Address. . . . . : 172.16.36.21
  Subnet Mask . . . . . : 255.255.254.0
  Default Gateway . . . . . : 172.16.36.1

Ethernet adapter Bluetooth Network Connection:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :
```

Рисунок 2.1: Команда ipconfig

Вводим ipconfig /all для вывода более подробной информации. Просматриваем данные о беспроводном соединении. Видим описание устройства (производитель Intel, MAC-адрес - DC-45-46-63-D4-E5). MAC-адрес состоит из 6 октетов: первые 3 октета идентифицируют производителя, последние 3 октета идентифицируют сетевой интерфейс (рис. [fig:2]).

```

Ethernet adapter vEthernet (Default Switch):
  Connection-specific DNS Suffix . :
  Description . . . . . : Hyper-V Virtual Ethernet Adapter
  Physical Address . . . . . : 00-15-5D-46-67-91
  DHCP Enabled. . . . . : No
  Autoconfiguration Enabled . . . . . : Yes
  Link-local IPv6 Address . . . . . : fe80::de9e:9f41:b2c5:3599%41(PREFERRED)
  IPv4 Address. . . . . : 172.29.0.1(PREFERRED)
  Subnet Mask . . . . . : 255.255.240.0
  Default Gateway . . . . . :
  DHCPv6 IAID. . . . . : 687871325
  DHCPv6 Client DUID. . . . . : 00-01-00-01-30-A7-B7-80-08-BF-B8-C3-AD-EE
  NetBIOS over Tcpip. . . . . : Enabled

Wireless LAN adapter Local Area Connection* 9:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :
  Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
  Physical Address. . . . . : DC-46-28-6C-92-D4
  DHCP Enabled. . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes

Wireless LAN adapter Local Area Connection* 10:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :
  Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
  Physical Address. . . . . : DE-46-28-6C-92-D3
  DHCP Enabled. . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes

Wireless LAN adapter Wi-Fi:
  Connection-specific DNS Suffix . : ruds.ru
  Description . . . . . : Intel(R) Wi-Fi 6E AX211 160MHz
  Physical Address. . . . . : DC-46-28-6C-92-D3
  DHCP Enabled. . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  IPv4 Address. . . . . : 172.16.36.21(PREFERRED)
  Subnet Mask . . . . . : 255.255.254.0
  Lease Obtained. . . . . : Wednesday, December 17, 2025 12:53:34 AM
  Lease Expires . . . . . : Wednesday, December 17, 2025 4:53:14 AM
  Default Gateway . . . . . : 172.16.36.1
  DHCP Server . . . . . : 192.168.80.59
  DNS Servers . . . . . : 8.8.8.8
                                8.8.4.4
  NetBIOS over Tcpip. . . . . : Enabled

```

Рисунок 2.2: Команда ipconfig /all

Проверив на специальном сайте производителя устройства по первым 3 октетам выясняем, что устройство выпущено компанией Intel Corporate. Взяв первый байт (DC) и переведя в двоичную систему счисления, получаем 11011100. Так как последний бит = 0, адрес является индивидуальным. Предпоследний бит = 0, следовательно, адрес глобально администрируемый.

2.2 Анализ кадров канального уровня в Wireshark

Запускаем Wireshark и выбираем беспроводное соединение. Запускаем захват трафика (рис. [fig:3]).

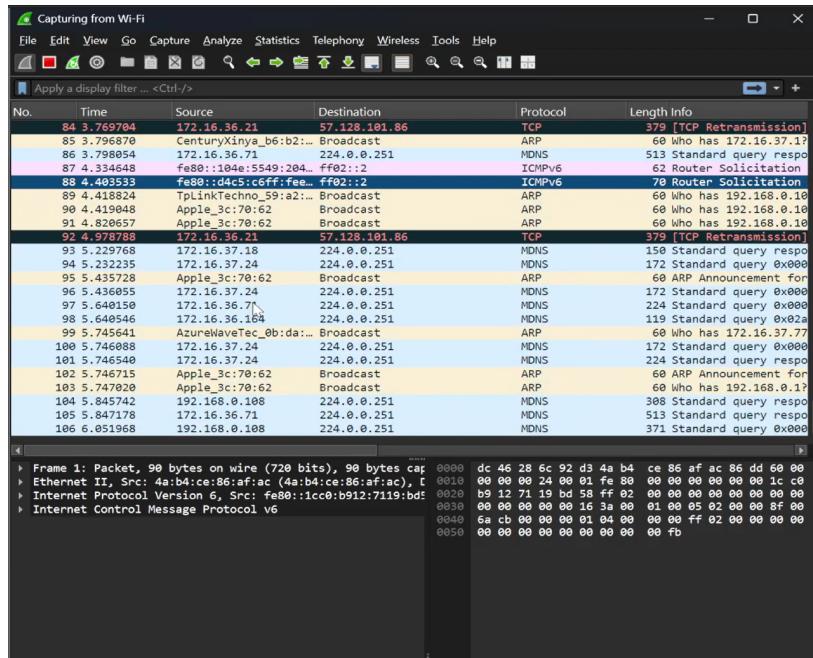


Рисунок 2.3: Запуск захвата трафика

Шлюз по умолчанию для моего устройства - 172.16.36.1 (было определено в предыдущем задании). С помощью команды ping 172.16.36.1 пингуем шлюз по умолчанию (рис. [fig:4]).

```
C:\Users\Ebrahim Alkamal>ping 172.16.36.1

Pinging 172.16.36.1 with 32 bytes of data:
Reply from 172.16.36.1: bytes=32 time=10ms TTL=254
Reply from 172.16.36.1: bytes=32 time=4ms TTL=254
Reply from 172.16.36.1: bytes=32 time=1ms TTL=254
Reply from 172.16.36.1: bytes=32 time=1ms TTL=254

Ping statistics for 172.16.36.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 10ms, Average = 4ms
```

Рисунок 2.4: Пинг шлюза по умолчанию

Останавливаем захват трафика. В строке фильтра указываем 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. Видим 4 пакета-запроса и 4 пакета-ответа. Выбираем

запрос и просматриваем в нижней части экрана информацию о нем. На вкладке физического уровня можно найти длину кадра (74 бита). Чтобы узнать MAC-адрес источника и шлюза перейдем на канальный уровень. Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства (DC-46-28-6C-92-D3). Адрес шлюза (destination, то куда отправлен запрос) - 70-18-A7-60-9C-D2. Тип адреса тут указан (показаны нулевые и первые биты MAC-адресов). Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые (рис. [fig:5])

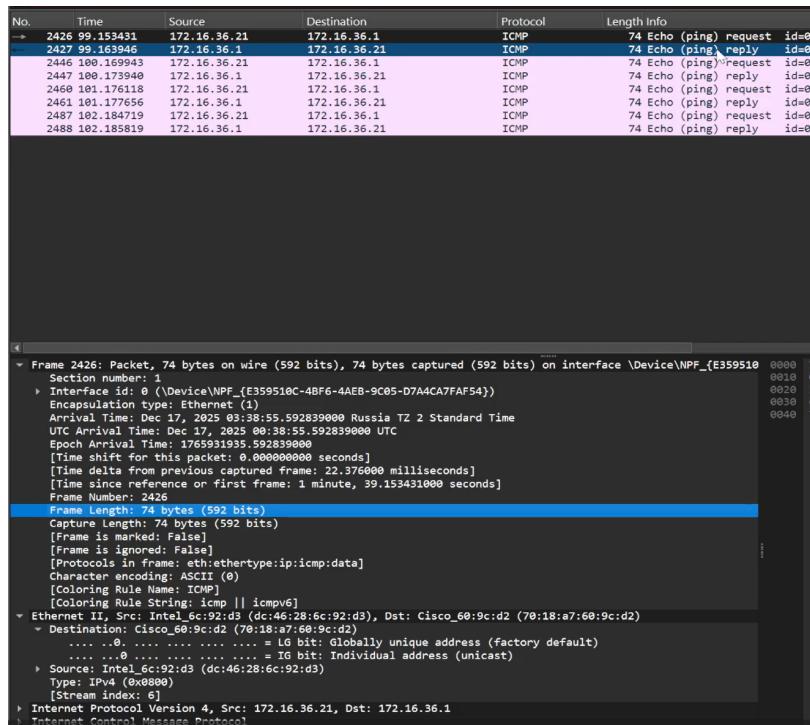


Рисунок 2.5: Кадр ICMP - эхо-запрос: информация о длине кадра, типе Ethernet и MAC-адресах

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 70-18-A7-60-9C-D2, а адрес назначения – адрес моего устройства (DC-46-28-6C-92-D3) (рис. [fig:6]).

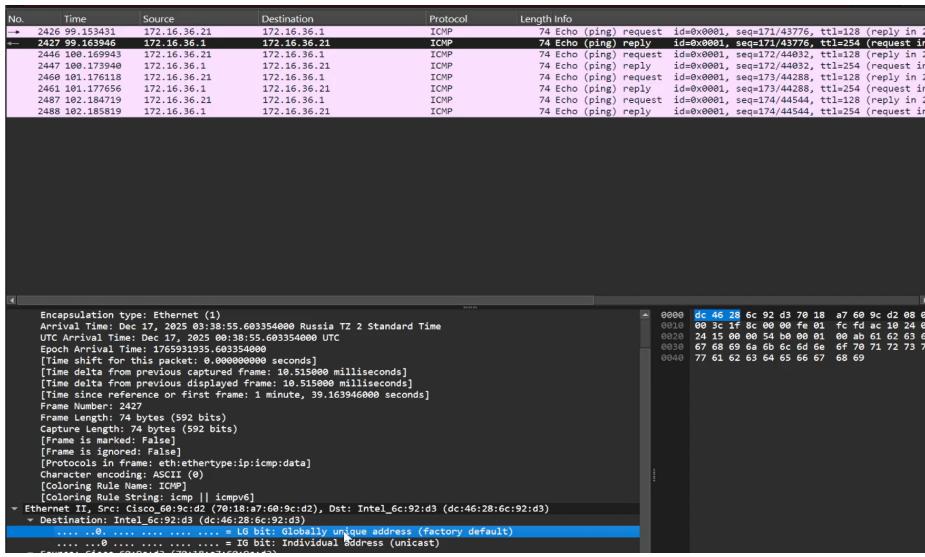


Рисунок 2.6: Кадр ICMP - эхо-ответ: информация о длине кадра, типе Ethernet, MAC-адресах

Изучим кадры данных протокола ARP. ARP-сообщение передаётся в кадре Ethernet II, общая длина кадра составляет 60 байт, включая служебное дополнение (padding). Поле Hardware type указывает на тип канального протокола — Ethernet (1), Protocol type — на протокол сетевого уровня IPv4. Размер MAC-адреса равен 6 байт, размер IPv4-адреса — 4 байта. Код операции равен 1, что соответствует ARP-запросу. В заголовке Ethernet II указаны MAC-адреса источника и получателя. Адрес получателя — широковещательный (ff:ff:ff:ff:ff:ff). Адрес источника — индивидуальный, глобально администрируемый MAC-адрес сетевого интерфейса отправителя. (индивидуальный и глобально администрируемый) (рис. [fig:7]).

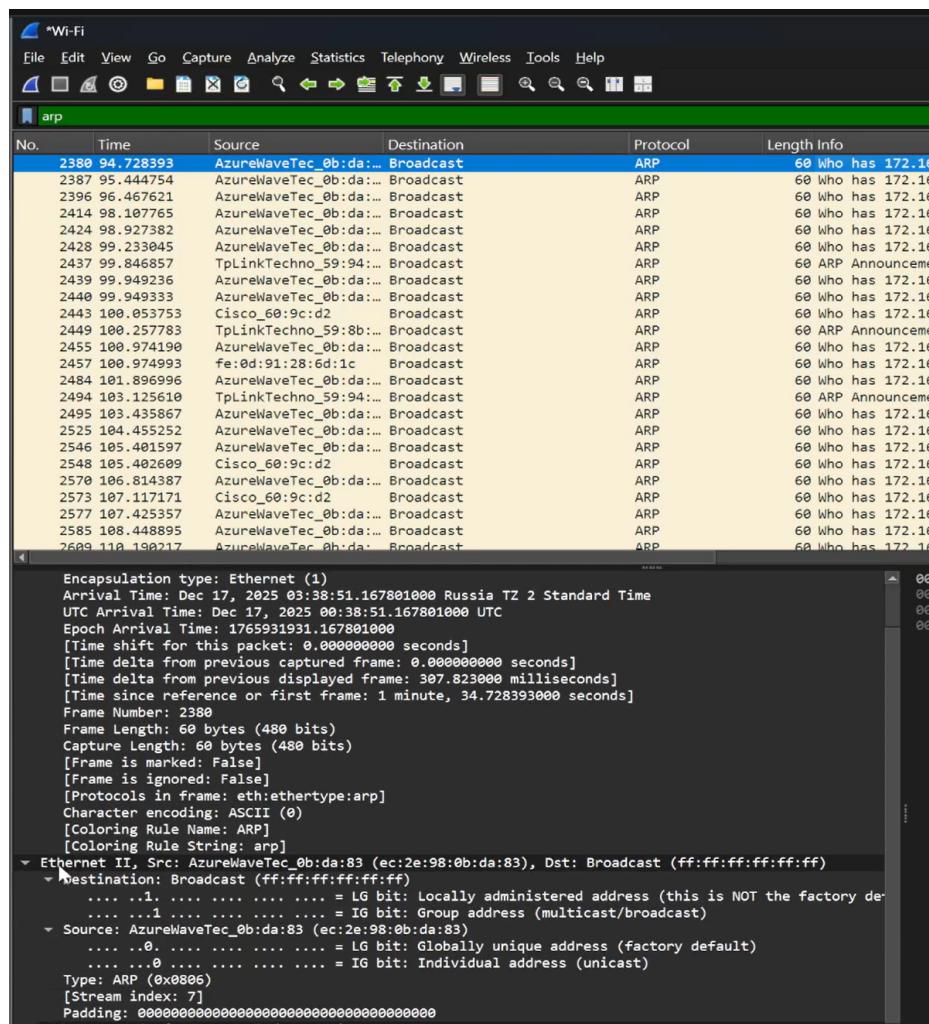


Рисунок 2.7: Кадр ARP: информация о длине кадра, типе Ethernet, MAC-адресах

Начинаем процесс захвата трафика в Wireshark. В командной строке выполняем команду ping rudn.ru. В результате все ICMP-запросы завершаются по тайм-ауту: отправлено 4 пакета, получено 0, потери составляют 100%. Это указывает на отсутствие ICMP-ответов от узла rudn.ru при данном сетевом подключении (рис. [fig:8]).

```
C:\Users\Ebrahim Alkamal>ping rudn.ru
Pinging rudn.ru [37.18.93.135] with 32 bytes of data:
Request timed out.

Ping statistics for 37.18.93.135:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Рисунок 2.8: Пинг сайта ping rudn.ru

Изучим запрос протокола ICMP (Echo Request). В заголовке Ethernet II адрес источника соответствует MAC-адресу сетевого интерфейса моего устройства — DC-46-28-6C-92-D3. Адрес назначения — MAC-адрес шлюза (следующего узла) — 70-18-A7-60-9C-D2. Оба MAC-адреса являются индивидуальными (unicast) и глобально администрируемыми (factory default) (рис. [fig:9]).

No.	Time	Source	Destination	Protocol	Length Info
120	9.625003	172.16.36.21	37.18.93.135	ICMP	74 Echo (ping) request
169	14.480260	172.16.36.21	37.18.93.135	ICMP	74 Echo (ping) request
294	19.487026	172.16.36.21	37.18.93.135	ICMP	74 Echo (ping) request
360	24.393133	172.16.36.21	37.18.93.135	ICMP	74 Echo (ping) request


```

Frame 120: Packet, 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{E359510C-4BF6-4AEB-9C05-D7A4CA7FAF54}
  Section number: 1
  ▶ Interface id: 0 (\Device\NPF_{E359510C-4BF6-4AEB-9C05-D7A4CA7FAF54})
    Encapsulation type: Ethernet (1)
    Arrival Time: Dec 17, 2025 03:47:54.599981000 Russia TZ 2 Standard Time
    UTC Arrival Time: Dec 17, 2025 00:47:54.599981000 UTC
    Epoch Arrival Time: 1765932474.599981000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 4.841000 milliseconds]
    [Time since reference or first frame: 9.625003000 seconds]
  Frame Number: 120
  Frame Length: 74 bytes (592 bits)
  Capture Length: 74 bytes (592 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:icmp:data]
    Character encoding: ASCII (0)
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
  ▶ Ethernet II, Src: Intel_6c:92:d3 (dc:46:28:6c:92:d3), Dst: Cisco_60:9c:d2 (70:18:a7:60:9c:d2)
    ▶ Destination: Cisco_60:9c:d2 (70:18:a7:60:9c:d2)
      .... ..0. .... .... .... = LG bit: Globally unique address (factory default)
      .... ..0. .... .... .... = IG bit: Individual address (unicast)
    ▶ Source: Intel_6c:92:d3 (dc:46:28:6c:92:d3)
      .... ..0. .... .... .... = LG bit: Globally unique address (factory default)
      .... ..0. .... .... .... = IG bit: Individual address (unicast)

```

Рисунок 2.9: Запрос протокола ICMP

Также был проанализирован трафик протокола ICMP. В ходе захвата были зафиксированы только ICMP-запросы типа Echo Request. ICMP-ответы типа Echo Reply в захваченном трафике отсутствуют. Это означает, что удалённый узел не отвечает на ICMP-запросы, либо ICMP-ответы блокируются на одном из сетевых устройств (маршрутизатором, межсетевым экраном или политиками безопасности сети). В результате обмен ICMP-сообщениями является односторонним (рис. [fig:10]).

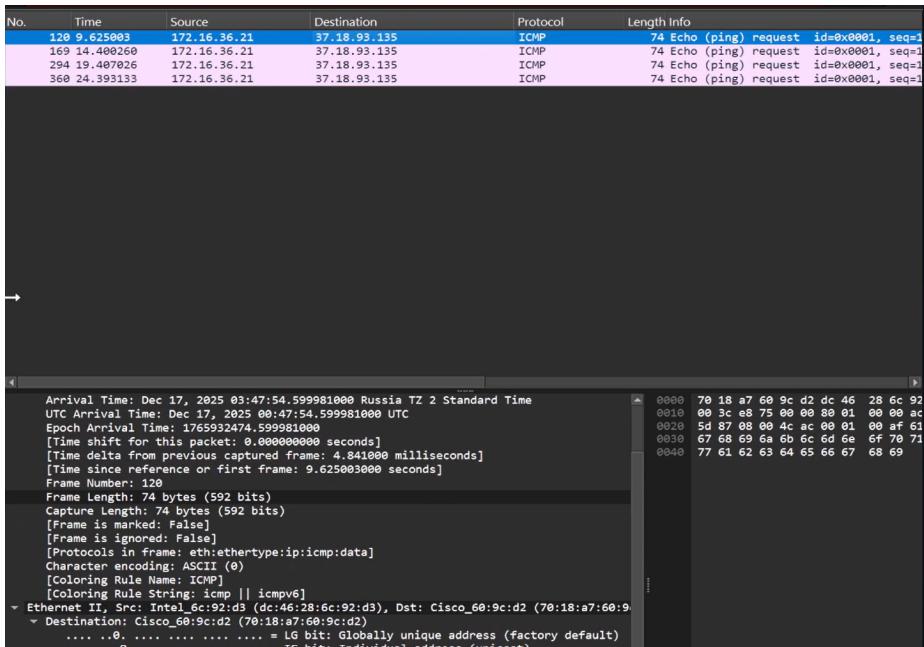


Рисунок 2.10: протокола ICMP

2.3 Анализ протоколов транспортного уровня в Wireshark

Запустив Wireshark, начинаем захват трафика. В браузере открываем сайт, работающий по протоколу HTTP (<http://info.cern.ch/>), и переходим по страницам сайта. В строке фильтра указываем http и анализируем HTTP-запросы, передаваемые поверх протокола TCP. В TCP-заголовке видно, что порт источника является динамическим и равен 64048, а порт назначения равен 80 — стандартному порту протокола HTTP. Также в заголовке TCP присутствуют поля порядкового номера (Sequence Number) и номера подтверждения (Acknowledgment Number), которые используются для обеспечения надёжной доставки данных(Acknowledgment Number) (рис. [fig:11])

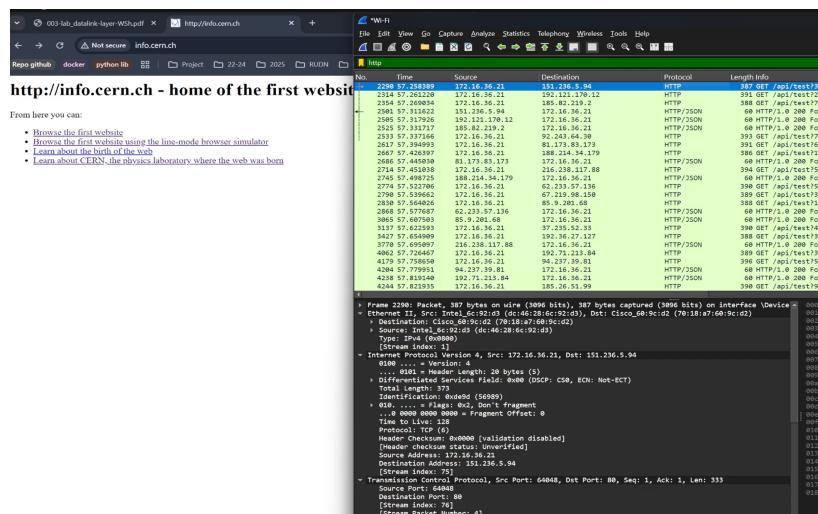


Рисунок 2.11: Кадр http - запрос

В случае ответа порты заданы наоборот, то есть источник - 80 порт, назначение

- 64048 (рис. [fig:12])

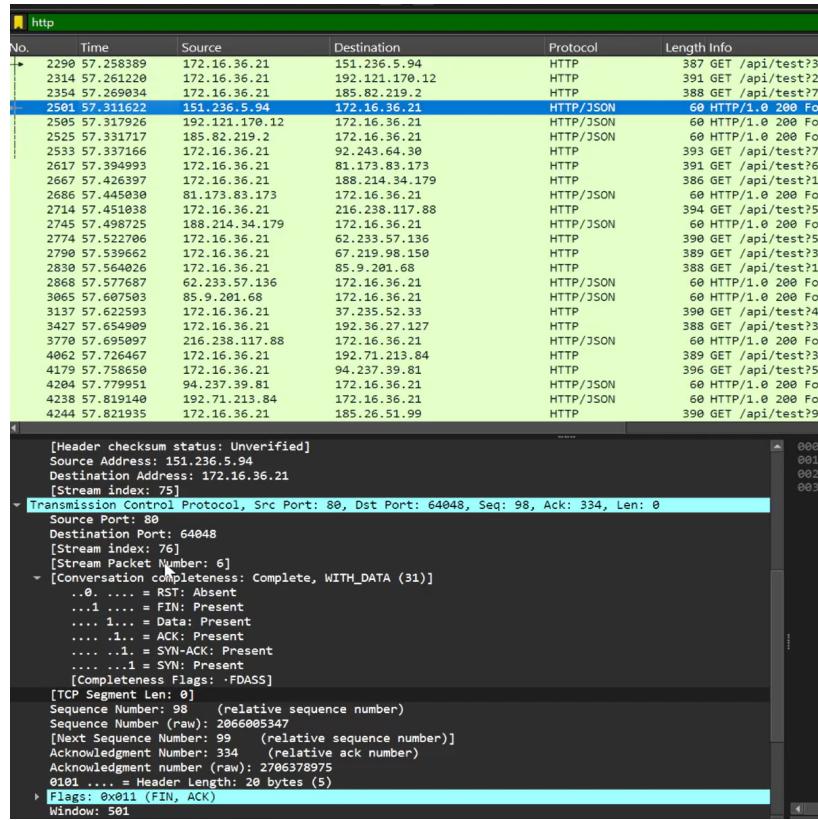


Рисунок 2.12: Кадр http - ответ

В Wireshark в строке фильтра укажем dns и проанализируем обмен данными по протоколу UDP при выполнении DNS-запросов и ответов. При отправке DNS-запроса порт источника является динамическим (случайным), а порт назначения равен 53 – стандартному порту службы DNS. В ответном DNS-сообщении порт источника равен 53, а порт назначения – динамическому порту клиента 64317 рис. ([fig:13]).

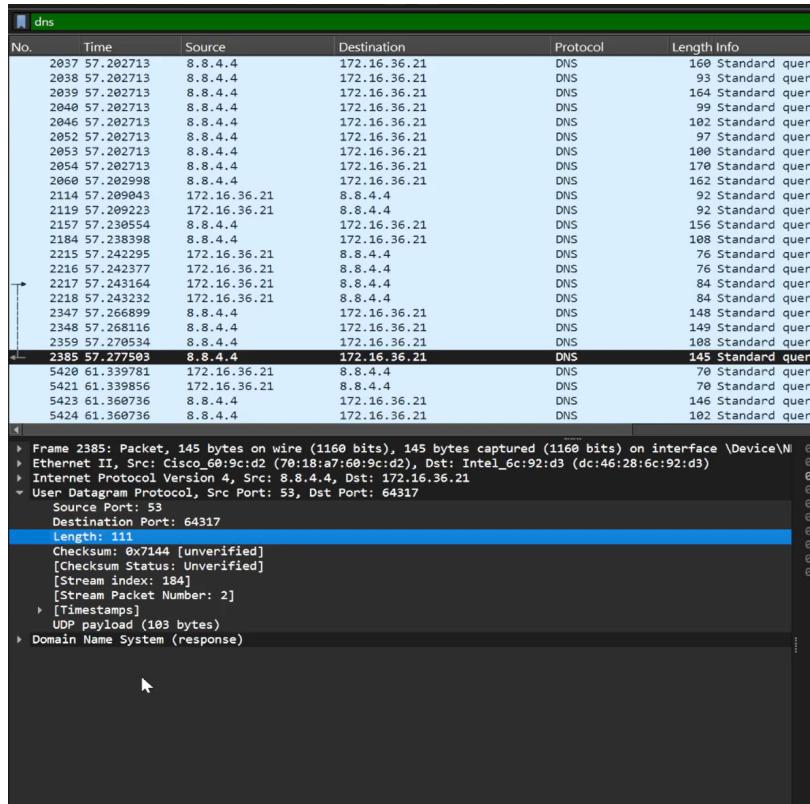


Рисунок 2.13: Кадр dns - запрос

В строке фильтра Wireshark укажем quic. Изначально QUIC-трафик не фиксировался, поэтому был использован браузер Google Chrome, после чего соответствующие пакеты появились. Проанализируем работу протокола QUIC. Как и в случае с DNS, можно рассмотреть информацию транспортного уровня, так как QUIC использует протокол UDP. Порт источника является динамическим (непrivилегированным) и равен 49768, порт назначения – 443, стандартный порт

HTTPS. Протокол QUIC изначально использует шифрование (TLS 1.3), встроенное в сам протокол, что обеспечивает защищённую передачу данных. В ответных пакетах номера портов меняются местами(рис. [fig:15]).

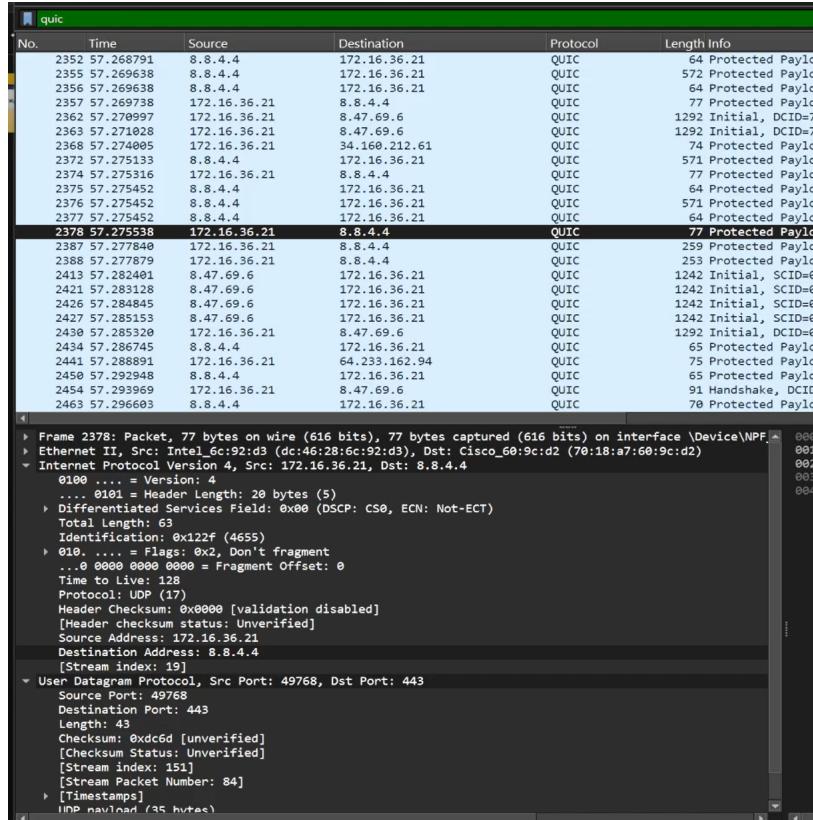


Рисунок 2.14: Кадр quic - запрос

2.4 Анализ handshake протокола TCP в Wireshark

Начав захват трафика, запускаем в браузере сайт, работающий по протоколу HTTP (<http://info.cern.ch/>), однако анализировать будем тот handshake, который нашли для примера. Установление связи клиент-сервер в TCP осуществляется в три этапа (трёхступенчатый handshake).

1. Режим активного доступа (Active Open). Клиент посыпает сообщение SYN, ISSa, т.е. в передаваемом сообщении установлен бит SYN (Synchronize Se-

quence Number), а в поле Порядковый номер (Sequence Number) — начальное 32-битное значение ISSa (Initial Sequence Number)

Находим кадр с флагом SYN. Sequence Number = 0 (рис. [fig:16]).

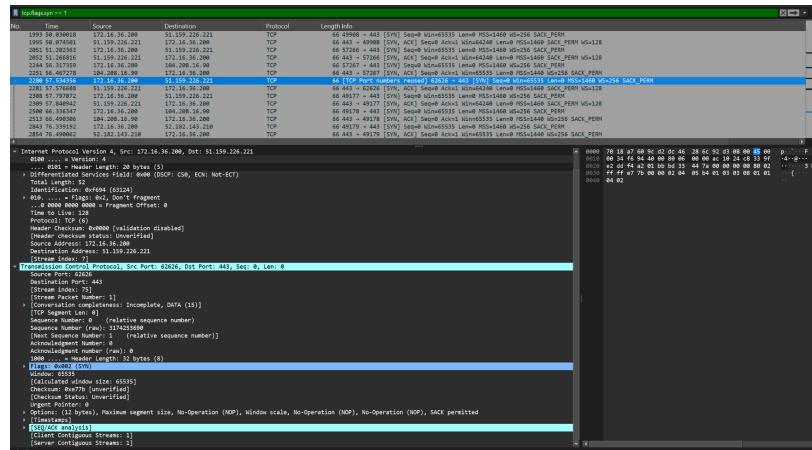


Рисунок 2.15: Первая ступень handshake TCP

2. 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 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.

Кадр с флагами SYN и ACK, где ACK равен Sequence Number из предыдущего шага, увеличенный на 1 ($0 + 1 = 1$) (рис. [fig:17]).

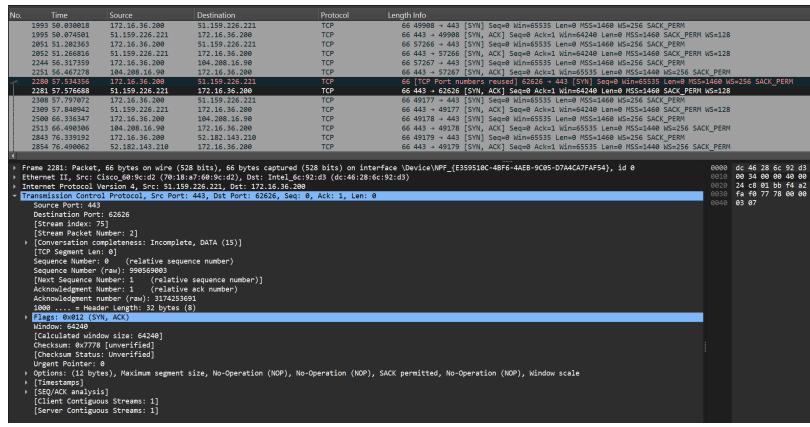


Рисунок 2.16: Вторая ступень handshake TCP

3. Завершение рукопожатия. Клиент отправляет подтверждение получения SYNсегмента от сервера с идентификатором, равным ISN (сервера)+1: ACK, ISS_a+1, ACK(ISS_b+1). В этом пакете установлен бит ACK, поле Порядковый номер (Sequence Number) содержит ISS_a+1, поле Номер подтверждения (Acknowledgment Number) содержит значение ISS_b+1. Посылкой этого пакета заканчивается трёхступенчатый handshake, и TCP-соединение считается установленным.

Теперь клиент может посыпать пакеты с данными на сервер по только что созданному виртуальному TCP-каналу: ACK, ISS_a+1, ACK(ISS_b+1); DATA.

Кадр с флагом ACK, где Sequence Number равен 6231, Acknowledgment Number равен 3590 (рис. [fig:18]).

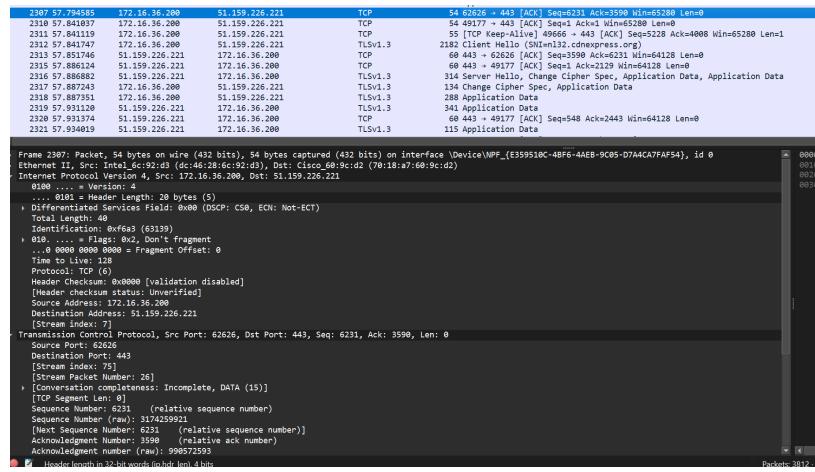


Рисунок 2.17: Третья ступень handshake TCP

В Wireshark в меню «Статистика» выбираем пункт «График потока». На графике видно, что сначала клиент отправляет серверу TCP-сегмент с флагом SYN и относительным номером последовательности Seq = 0. Затем сервер отвечает сегментом с флагами SYN и ACK, при этом Seq = 0, а Ack = 1, что подтверждает получение SYN от клиента. В третьем пакете клиент отправляет сегмент с установленным флагом ACK, где значения Seq = 1 и Ack = 1. Отправка этого пакета завершает процедуру трёхступенчатого TCP-рукопожатия (рис. [fig:19]).

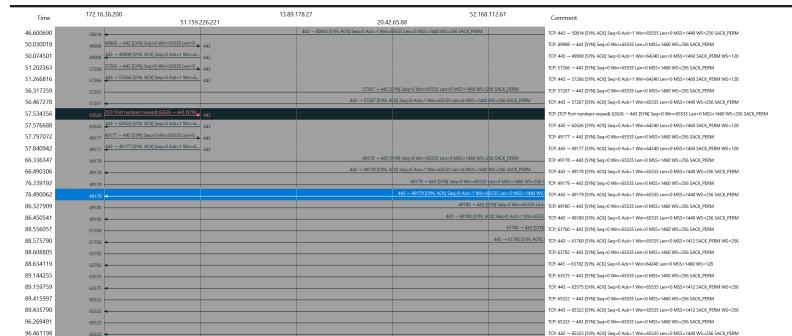


Рисунок 2.18: График потока

В Wireshark останавливаем захват трафика.

3 Выводы

В результате выполнения работы были изучены посредством Wireshark кадры Ethernet, произведен анализ PDU протоколов транспортного и прикладного уровней стека TCP/IP.