

# Лабораторная работа №3

## Анализ трафика в Wireshark

---

Демидова Екатерина Алексеевна

23 сентября 2023

Российский университет дружбы народов

## Вводная часть

---

## Цель работы

---

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

# Задание

---

1. MAC-адресация
  - 1.1 Изучение возможностей команды ipconfig для ОС типа Windows (ifconfig для систем типа Linux).
  - 1.2 Определение MAC-адреса устройства и его типа.
2. Анализ кадров канального уровня в Wireshark
  - 2.1 Установить на домашнем устройстве Wireshark.
  - 2.2 С помощью Wireshark захватить и проанализировать пакеты ARP и ICMP в части кадров канального уровня
3. Анализ протоколов транспортного уровня в Wireshark. С помощью Wireshark захватить и проанализировать пакеты HTTP, DNS в части заголовков и информации протоколов TCP, UDP, QUIC.
4. Анализ handshake протокола TCP в Wireshark. С помощью Wireshark проанализировать handshake протокола TCP.

## Выполнение лабораторной работы

---

## MAC-адресация

---

## MAC-адресация

С помощью команды `ifconfig` выведем информацию о текущем сетевом соединении.

```
eademidova@Demidrol:~$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Локальная петля (Loopback))
            RX packets 34787 bytes 2798391 (2.7 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 34787 bytes 2798391 (2.7 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.147 netmask 255.255.255.0 broadcast 192.168.2.255
        inet6 fe80::17ff:4909:39e4:9973 prefixlen 64 scopeid 0x20<link>
            ether e8:f4:08:c4:e3:ca txqueuelen 1000 (Ethernet)
            RX packets 824722 bytes 1017809698 (1.0 GB)
            RX errors 0 dropped 80 overruns 0 frame 0
            TX packets 253712 bytes 56982702 (56.9 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 1: ВЫВОД КОМАНДЫ `ifconfig`

## MAC-адресация

Так же добавив название сетевого интерфейса можно вывести информацию только о нём.

```
eademidova@Demidrol:~$ ifconfig wlp0s20f3
wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.2.147  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::17ff:4909:39e4:9973  prefixlen 64  scopeid 0x20<link>
            ether e8:f4:08:c4:e3:ca  txqueuelen 1000  (Ethernet)
            RX packets 825254  bytes 1017952582 (1.0 GB)
            RX errors 0  dropped 82  overruns 0  frame 0
            TX packets 254436  bytes 57254208 (57.2 MB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eademidova@Demidrol:~$
```

Рис. 2: Вывод информации о конкретном сетевом интерфейсе'

## MAC-адресация

Теперь выключим сетевой интерфейс добавив команду `down`(с помощью команды `ip` его можно включить).

```
eademidova@Demidrol:~$ sudo ifconfig wlp0s20f3 down
eademidova@Demidrol:~$ ifconfig
    lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
            loop txqueuelen 1000  (Локальная петля (Loopback))
            RX packets 43853  bytes 3500431 (3.5 MB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 43853  bytes 3500431 (3.5 MB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eademidova@Demidrol:~$ ifconfig -a
    lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
            loop txqueuelen 1000  (Локальная петля (Loopback))
            RX packets 44477  bytes 3547647 (3.5 MB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 44477  bytes 3547647 (3.5 MB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

    wlp0s20f3: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.1.105  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::7bbf:cc5e:7d67:1bib  prefixlen 64  scopeid 0x20<link>
            ether e8:f4:08:c4:e3:ca  txqueuelen 1000  (Ethernet)
            RX packets 836617  bytes 1026176106 (1.0 GB)
            RX errors 0  dropped 86  overruns 0  frame 0
            TX packets 263105  bytes 59331535 (59.3 MB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eademidova@Demidrol:~$
```

Рис. 3: Выключение сетевого интерфейса и демонстрация опции `-a`

## MAC-адресация

Также можно использовать опцию **-s**, с помощью которой выводится краткий список интерфейсов.

```
eademidova@Demidrol:~$ ifconfig -s
Iface      MTU     RX-OK RX-ERR RX-DRP RX-OVR     TX-OK TX-ERR TX-DRP TX-OVR Flg
lo       65536    58233     0     0 0      58233     0     0     0 LRU
wlp0s20f  1500   851363     0    86 0     278785     0     0     0 BMRU
eademidova@Demidrol:~$
```

Рис. 4: Использование опции **-s**

## MAC-адресация

```
link/ether 00:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
eademidova@Demidrol:~$ ip -brief link
lo          UNKNOWN      00:00:00:00:00:00 <LOOPBACK,UP,LOWER_UP>
wlp0s20f3    DOWN        e8:f4:08:c4:e3:ca <BROADCAST,MULTICAST>
eademidova@Demidrol:~$
```

Рис. 5: Демонстрация MAC-адреса

## MAC-адресация

Первые три байта e8:f4:08 в этом адресе соответствуют индентификатору производителя. Последние три байта c4:e3:ca идентифицируют сетевой интерфейс.

MAC Address Lookup

Find the vendor name of a device by entering an OUI or a MAC address

MAC EBF408

Check an OUIs or a MAC address and display details like vendor name, location, MAC details, and more ...

Search by Vendor Name?

**Intel Corporate**

Vendor Details

OUI: EBF4:08

Vendor name: Intel Corporate ↗

Address

Lot 8  
Jalan Hi-Tech 2/3  
Kulim Kedah 09000  
MY.

Assignment Type MA-L

Mac Address Block Large (previously named OUI). Number of address  $2^{24}$  (~16 Million)

Initial registration: 23 October 2020



Рис. 6: Демонстрация производителя сетевого оборудования

Возьмем первый байт e8 и переведём его в двоичную систему. Получим 11101000. Так как последний бит ноль, то адрес является индивидуальным. А так как предпоследний бит ноль, то адрес глобально администрируемый.

## Анализ кадров канального уровня в Wireshark

---

# Анализ кадров канального уровня в Wireshark

Запустим wireshark и начнем захват трафика.

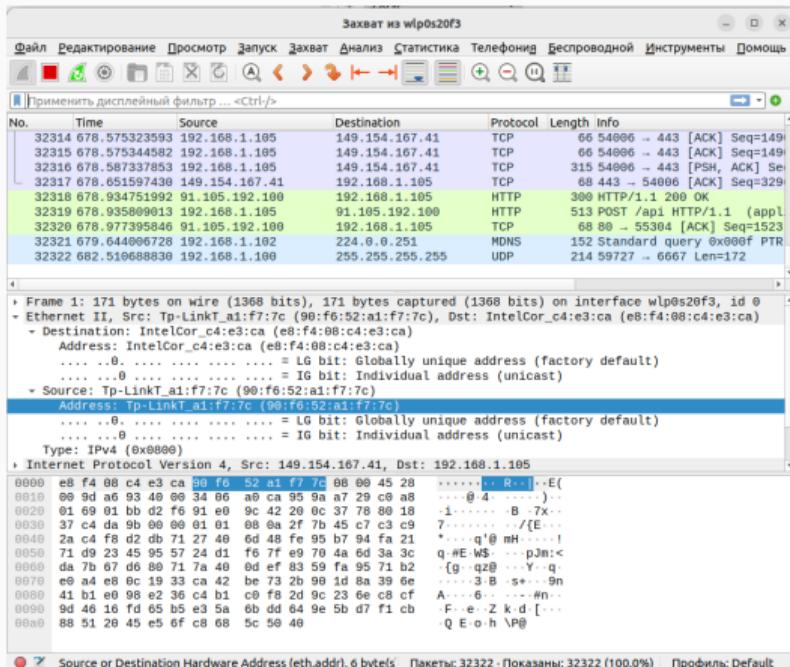


Рис. 7: Захват трафика в wireshark

# Анализ кадров канального уровня в Wireshark

```
wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.105 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::7bbfc:cc5e:7d67:1b1b/64 scopeid 0x20<link>
            ether e8:f4:08:c4:e3:ca txqueuelen 1000 (Ethernet)
            RX packets 958063 bytes 1118510172 (1.1 GB)
            RX errors 0 dropped 88 overruns 0 frame 0
            TX packets 367820 bytes 93223431 (93.2 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eademidova@Dentdrol:~$ ip route
default via 192.168.1.1 dev wlp0s20f3 proto dhcp metric 600
192.168.1.0/24 dev wlp0s20f3 proto kernel scope link src 192.168.1.105 metric 600
eademidova@Dentdrol:~$
```

Рис. 8: Определение IP-адреса устройства и шлюза по умолчанию

## Анализ кадров канального уровня в Wireshark

С помощью команды `ping 192.168.1.1` пропингуем шлюз по умолчанию.

```
eademidova@Demidrol:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=3.57 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=8.40 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=3.54 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=3.34 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 3.335/4.712/8.403/2.132 ms
eademidova@Demidrol:~$
```

Рис. 9: Пингование шлюза по умолчанию

# Анализ кадров канального уровня в Wireshark

The screenshot shows a Wireshark capture window titled "arp or icmp". The packet list pane displays several ICMP Echo (ping) requests and responses between two hosts. The details pane shows the structure of a selected ICMP frame, including fields like Type: IPv4 (0x0800), Src: 192.168.1.105, and Dst: 192.168.1.1. The bytes pane shows the raw hex and ASCII data of the selected frame.

No.	Time	Source	Destination	Protocol	Length	Info
629	102.299702396	Tp-LinkT_a1:f7:7c	IntelCor_c4:e3:ca	ARP	42	Who has 192.168.1.105? Tell 192.168.1.1
630	102.299829410	IntelCor_c4:e3:ca	Tp-LinkT_a1:f7:7c	ARP	42	192.168.1.105 is at e8:f4:08:c4:e3:ca
1422	160.989796289	192.168.1.105	192.168.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=64
1423	160.993349548	192.168.1.1	192.168.1.105	ICMP	98	Echo (ping) reply id=0x0002, seq=1/256, ttl=64
1429	161.991432645	192.168.1.105	192.168.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=64
1430	161.999818442	192.168.1.1	192.168.1.105	ICMP	98	Echo (ping) reply id=0x0002, seq=2/512, ttl=64
1431	162.992890734	192.168.1.105	192.168.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=64
1432	162.996356488	192.168.1.1	192.168.1.105	ICMP	98	Echo (ping) reply id=0x0002, seq=3/768, ttl=64
1438	163.994441119	192.168.1.105	192.168.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, ttl=64
1439	163.997764552	192.168.1.105	192.168.1.105	ICMP	98	Echo (ping) reply id=0x0002, seq=4/1024, ttl=64

Frame 1422: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface wlp0s20f3, id 0  
Ethernet II, Src: IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca), Dst: Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c)  
    Address: Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c)  
        ....0..... = LG bit: Globally unique address (factory default)  
        ....0..... = IG bit: Individual address (unicast)  
    Source: IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca)  
        Address: IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca)  
        ....0..... = LG bit: Globally unique address (factory default)  
        ....0..... = IG bit: Individual address (unicast)  
    Type: IPv4 (0x0800)  
    Internet Protocol Version 4, Src: 192.168.1.105, Dst: 192.168.1.1  
    0000 90 f6 52 a1 f7 7c e8 f4 08 c4 e3 ca 08 00 45 00 .R...|.....E.  
    0010 00 54 a9 14 40 00 40 01 0d da c0 a8 01 69 c0 a8 -T-@.....1.  
    0020 01 01 08 00 37 5f 00 02 00 01 0d b5 08 65 00 00 ....7\_.....  
    0030 00 00 e7 b0 04 00 00 00 00 00 10 11 12 13 14 15 .....!#\$%  
    0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 .....&'()\*,-. ./012345  
    0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 67  
    0060 36 37

Ethernet (eth), 14 byte(s)

Пакеты: 2292 - Показаны: 10 (0.4%) - Потеряно: 0 (0.0%)    Профиль: Default

Рис. 10: Кадр ICMP - эхо-запрос

# Анализ кадров канального уровня в Wireshark

The screenshot shows the Wireshark interface with the following details:

- Search Bar:** Contains the filter "arp or icmp".
- Table Headers:** No., Time, Source, Destination, Protocol, Length, Info.
- List of Packets:** A list of captured packets, mostly ICMP Echo requests and replies between various hosts on the network.
- Selected Packet:** The 1423 packet is selected, showing its details:
  - Frame 1423:** 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface wlp0s20f3, id 0.
  - Ethernet II:** Src: Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c), Dst: IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca)
  - Destination:** IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca)
    - Address: IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca)
    - .... .0. .... .... .... = LG bit: Globally unique address (factory default)
    - .... .0. .... .... .... = IG bit: Individual address (unicast)
  - Source:** Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c)
    - Address: Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c)
    - .... .0. .... .... .... = LG bit: Globally unique address (factory default)
    - .... .0. .... .... .... = IG bit: Individual address (unicast)
  - Type:** IPv4 (0x0800)
  - Internet Protocol Version 4:** Src: 192.168.1.1, Dst: 192.168.1.105
- Hex View:** Shows the raw byte sequence of the selected packet.
- Text View:** Shows the ASCII representation of the selected packet.
- Bottom Status Bar:** Shows "Ethernet (eth), 14 byte(s)", "Пакеты: 2292 - Показаны: 10 (0.4%) - Потеряно: 0 (0.0%)", and "Профиль: Default".

Рис. 11: Кадр ICMP - эхо-ответ

# Анализ кадров канального уровня в Wireshark

The screenshot shows the Wireshark interface with the following details:

- Frame 629:** An ARP request frame (Type: ARP (0x0806)) sent from Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c) to IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca).
  - Destination:** IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca)
    - Address: IntelCor\_c4:e3:ca (e8:f4:08:c4:e3:ca)
      - ... .0. .... .... .... = LG bit: Globally unique address (factory default)
      - ... .0. .... .... .... = IG bit: Individual address (unicast)
  - Source:** Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c)
    - Address: Tp-LinkT\_a1:f7:7c (90:f6:52:a1:f7:7c)
      - ... .0. .... .... .... = LG bit: Globally unique address (factory default)
      - ... .0. .... .... .... = IG bit: Individual address (unicast)
  - Type:** ARP (0x0806)
- Address Resolution Protocol (request):** The raw hex dump of the ARP frame.

At the bottom, it says: Пакеты: 2292 - Показаны: 10 (0.4%) - Потеряно: 0 (0.0%) - Профиль: Default

Рис. 12: Кадр ARP

# Анализ кадров канального уровня в Wireshark

Теперь начнём новый процесс захвата трафика и пропингуем сайт wikipedia.org.

```
[root@demidova ~]# ping wikipedia.org
eademidova@Demidrol:~$ sudo ping wikipedia.org
PING wikipedia.org (185.15.59.224) 56(84) bytes of data.
64 bytes from text-lb.esams.wikimedia.org (185.15.59.224): icmp_seq=1 ttl=52 time=49.8 ms
64 bytes from text-lb.esams.wikimedia.org (185.15.59.224): icmp_seq=2 ttl=52 time=49.6 ms
64 bytes from text-lb.esams.wikimedia.org (185.15.59.224): icmp_seq=3 ttl=52 time=49.4 ms
64 bytes from text-lb.esams.wikimedia.org (185.15.59.224): icmp_seq=4 ttl=52 time=50.7 ms
^C
--- wikipedia.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 49.442/49.867/50.695/0.490 ms
eademidova@Demidrol:~$
```

Рис. 13: Пингование сайта wikipedia.org

## Анализ кадров канального уровня в Wireshark

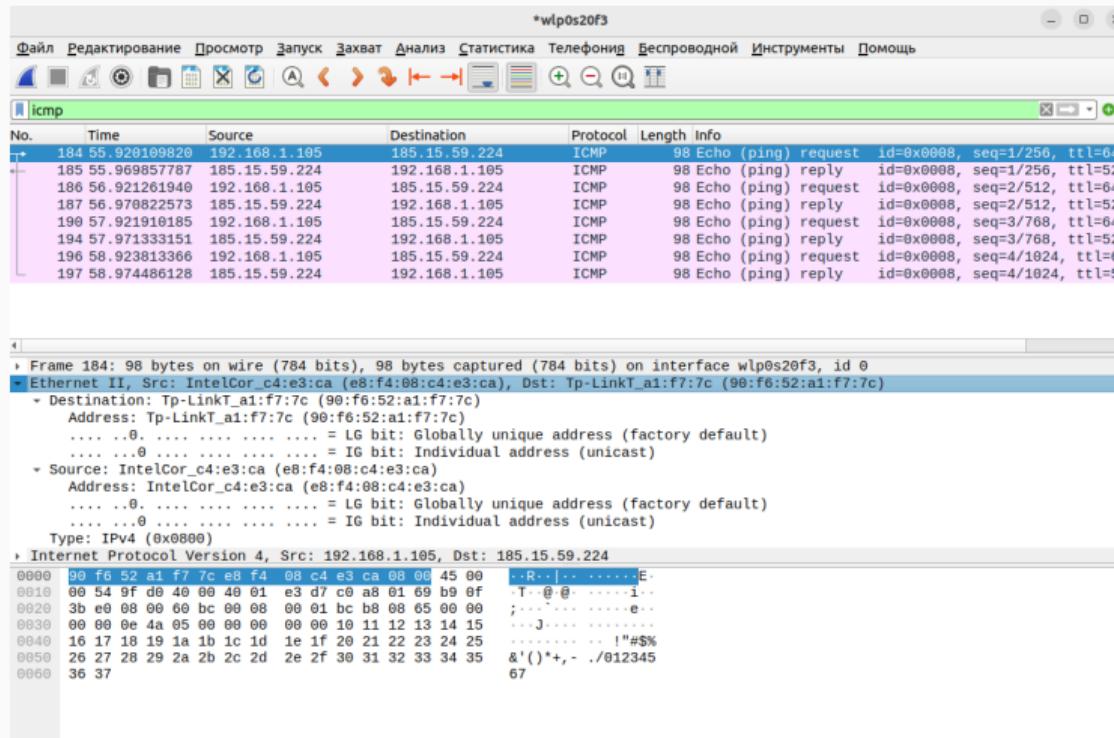


Рис. 14: Пингование сайта wikipedia.org. Кадр ICMP - эхо-запрос

# Анализ кадров канального уровня в Wireshark

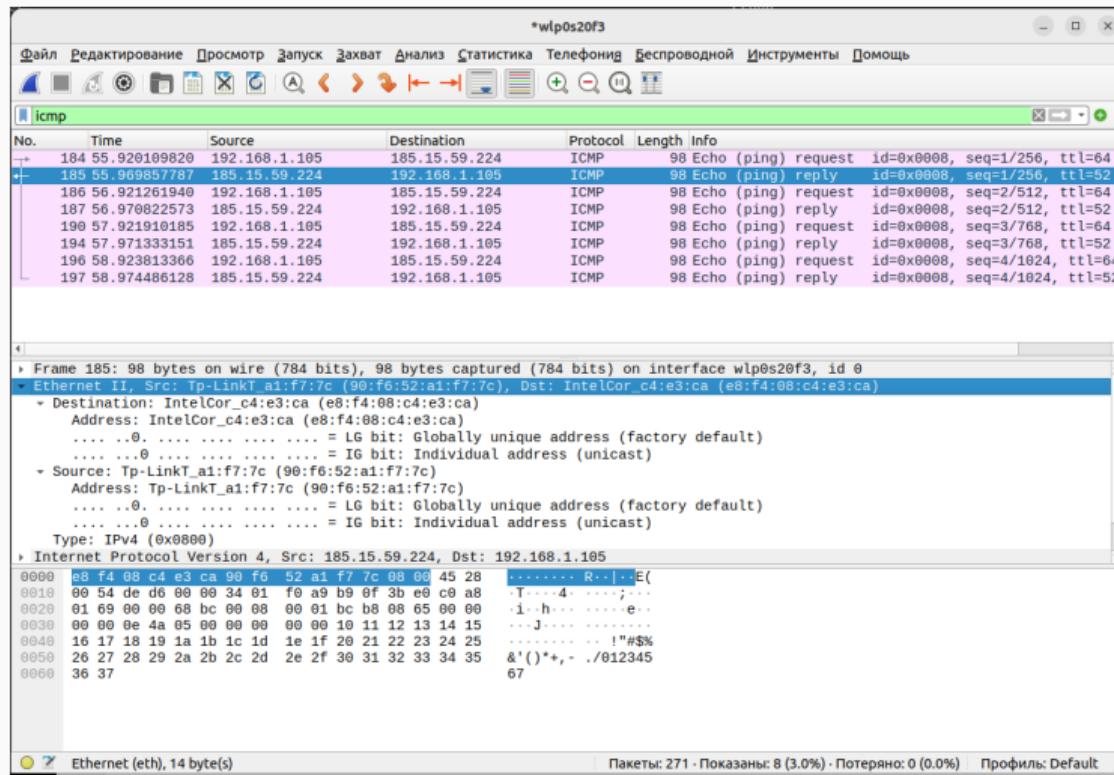


Рис. 15: Пингование сайта wikipedia.org. Кадр ICMP - эхо-ответ

# Анализ кадров канального уровня в Wireshark

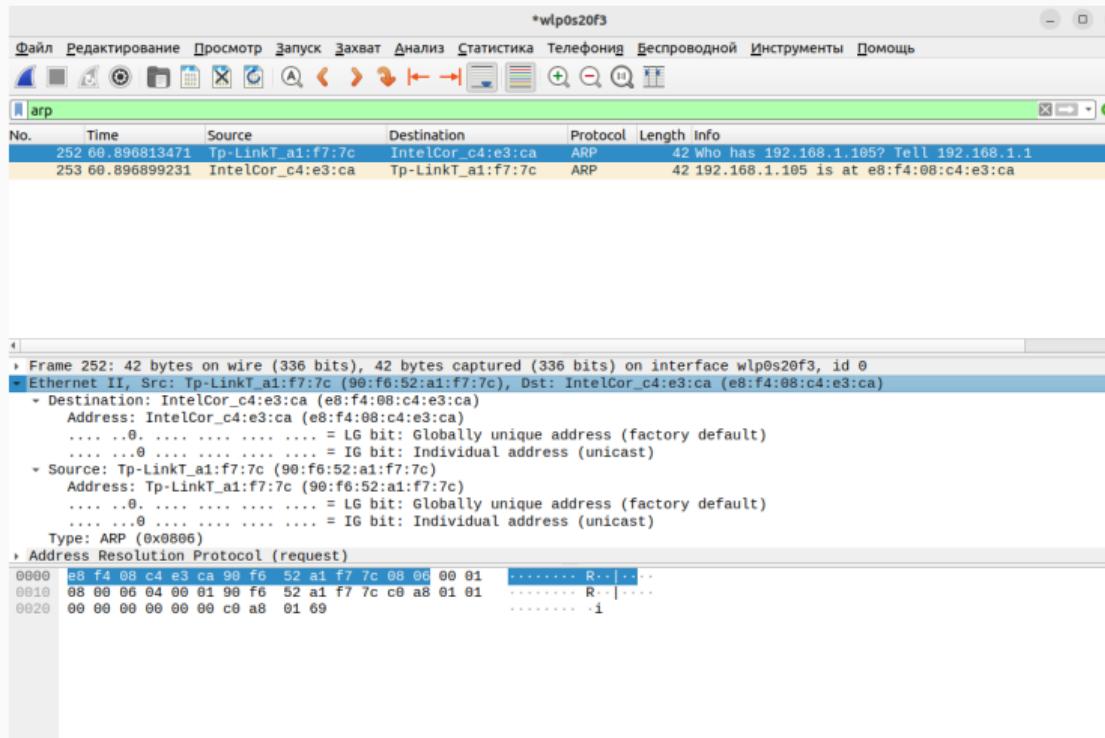


Рис. 16: Пингование сайта wikipedia.org. Кадр ARP - запрос

# Анализ кадров канального уровня в Wireshark

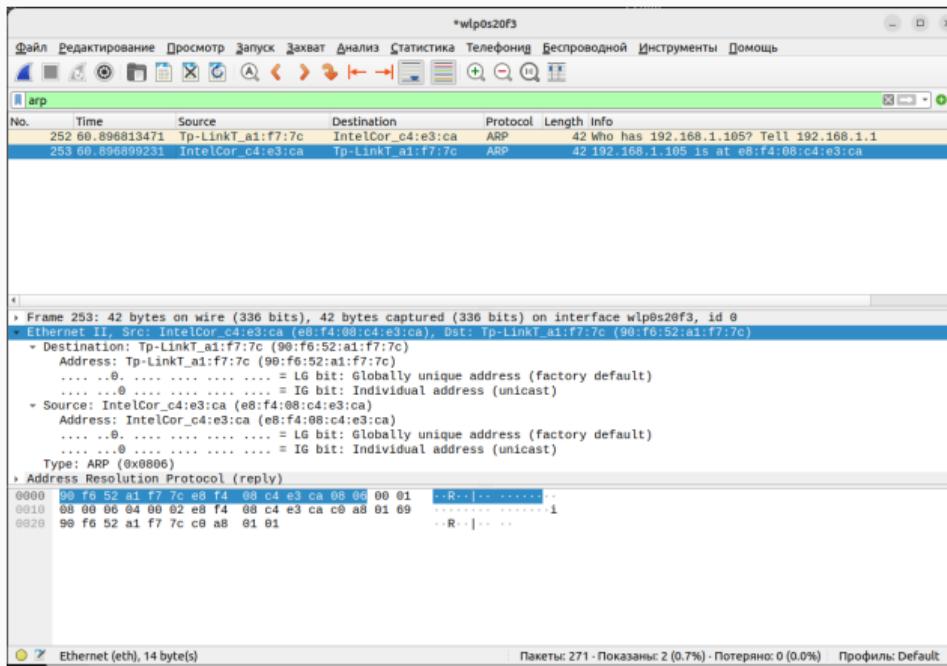


Рис. 17: Пингование сайта wikipedia.org. Кадр ARP - ответ

## Анализ протоколов транспортного уровня в Wireshark

---

# Анализ протоколов транспортного уровня в Wireshark

Порт источника задан случайно, выбором из непривелигированных и незанятых портов, и равен 555882, порт назначения равен 80 - это стандартный порт HTTP.

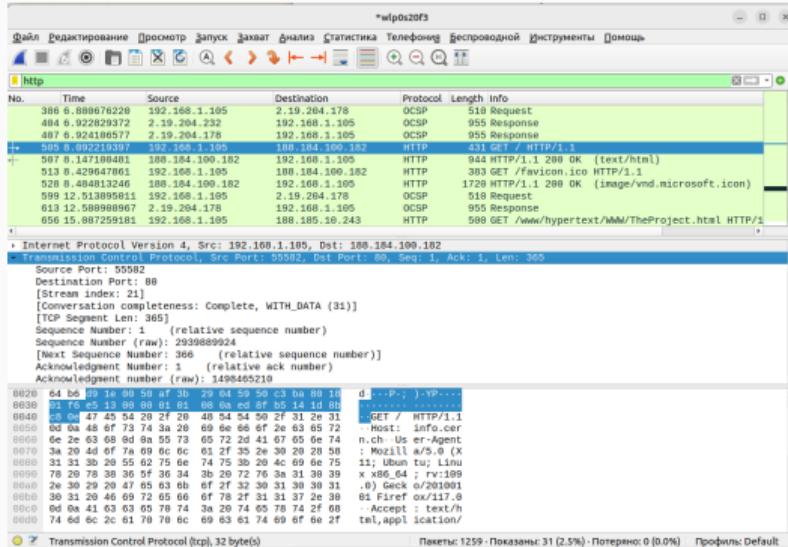


Рис. 18: Запрос HTTP по протоколу TCP

# Анализ протоколов транспортного уровня в Wireshark

В случае ответа порты заданы наоборот, то есть источник - 80 порт, назначение - 55582.

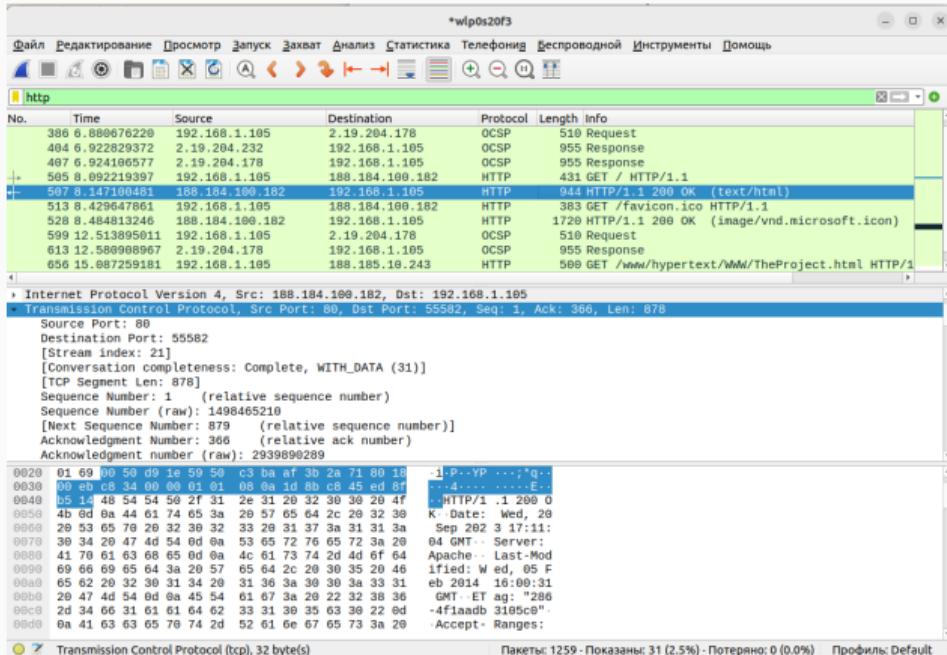


Рис. 19: Ответ HTTP по протоколу TCP

# Анализ протоколов транспортного уровня в Wireshark

В разделе HTP можно увидеть ссылку на html-страницу.

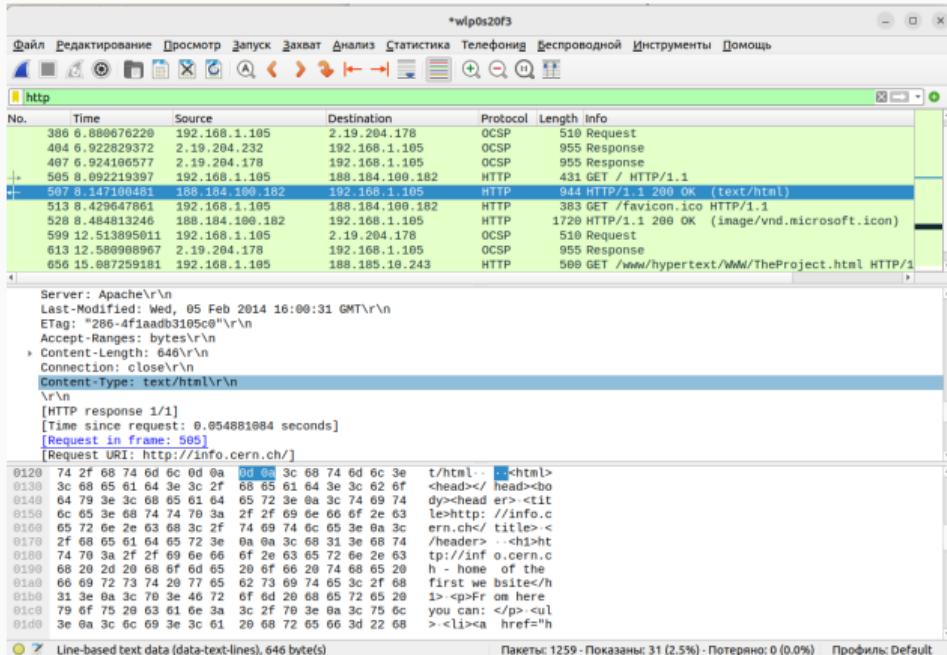


Рис. 20: Раздел HTP

# Анализ протоколов транспортного уровня в Wireshark

The screenshot shows the Wireshark interface with the following details:

- Toolbar:** File, Редактирование (Edit), Просмотр (View), Запуск (Capture), Захват (Capture), Анализ (Analysis), Статистика (Statistics), Телефония (Telephony), Беспроводной (Wireless), Инструменты (Tools), Домошь (Help).
- Menu Bar:** http
- Table View:** Shows network traffic with columns: No., Time, Source, Destination, Protocol, Length, Info. The table includes rows for OCSP requests, an HTTP GET request for "TheProject.html", and various other HTTP responses.
- Text Data View:** Shows the raw line-based text data for the selected HTTP response (row 507). It displays the HTML content of the CERN homepage, including the title, header, and body text.
- Hex View:** Shows the hex dump of the selected packet (HTTP response from info.cern.ch).
- Status Bar:** Line-based text data (data-text-lines), 646 byte(s) | Пакеты: 1259 · Показаны: 31 (2.5%) · Потеряно: 0 (0.0%) | Профиль: Default

Рис. 21: Раздел Line-based text data

# Анализ протоколов транспортного уровня в Wireshark

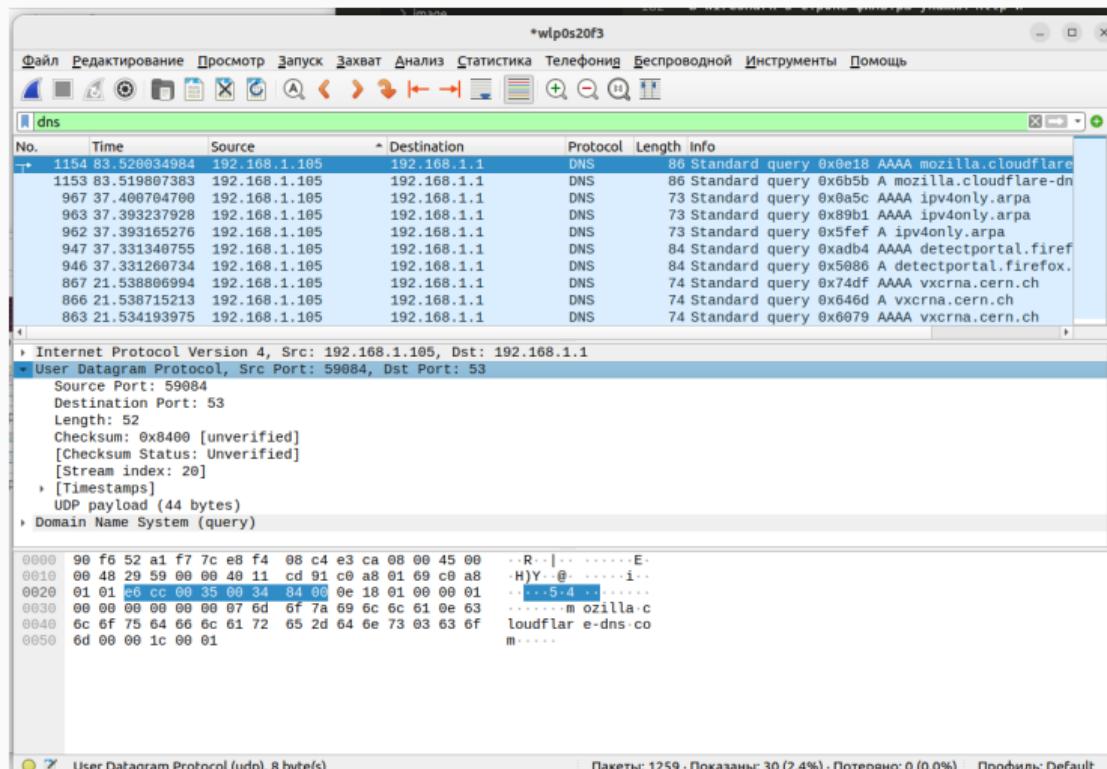


Рис. 22: Запрос dns по протоколу UDP

# Анализ протоколов транспортного уровня в Wireshark

В случае ответа порты заданы наоборот, то есть источник - 53 порт, назначение - 59084.

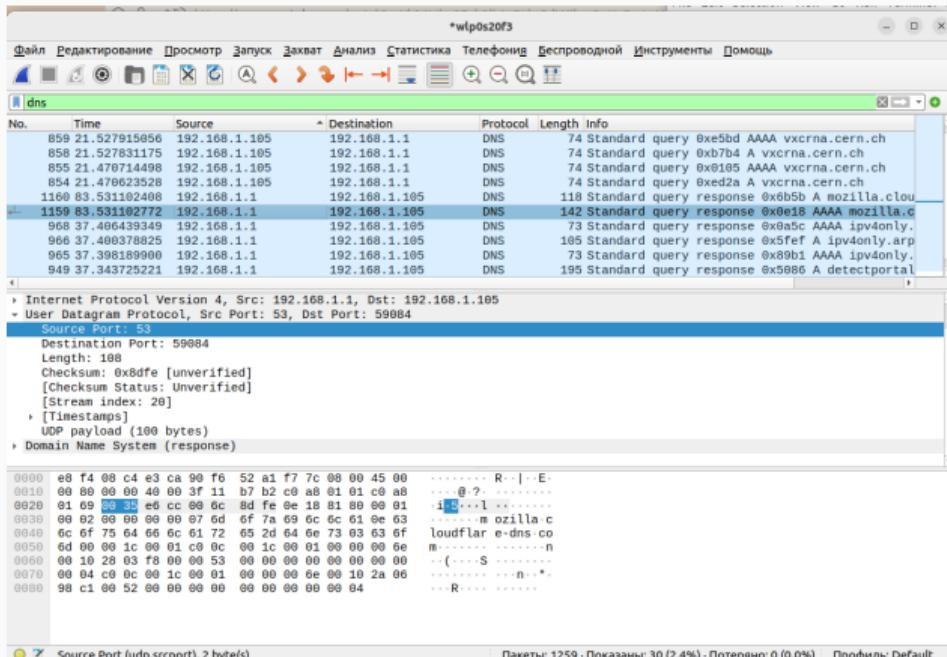


Рис. 23: Ответ DNS по протоколу UDP

## Анализ протоколов транспортного уровня в Wireshark

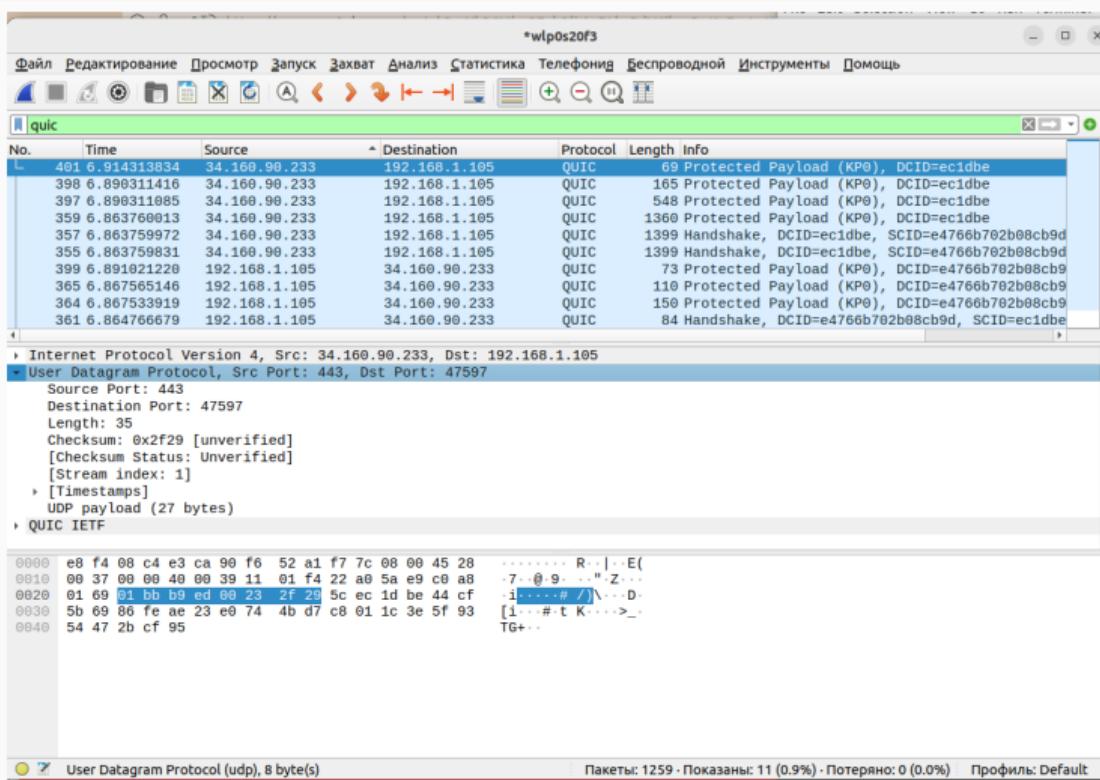


Рис. 24: Запрос QUIC по протоколу UDP

# Анализ протоколов транспортного уровня в Wireshark

В случае ответа порты заданы наоборот, то есть источник - 443 порт, назначение - 47597.

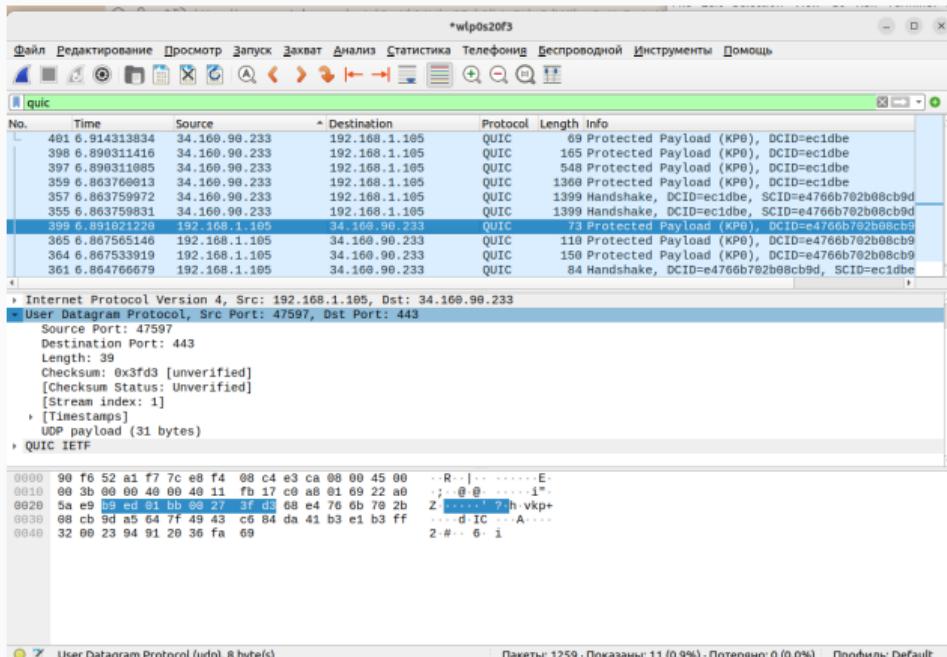


Рис. 25: Ответ QUIC по протоколу UDP

# Анализ протоколов транспортного уровня в Wireshark

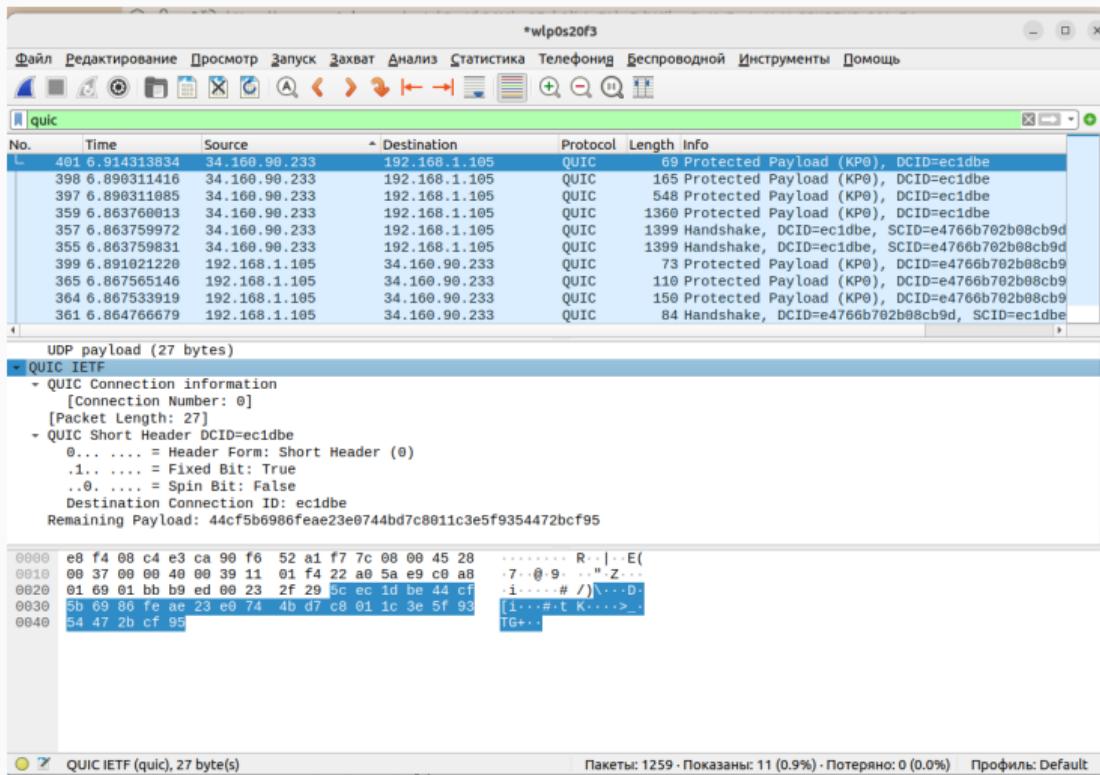


Рис. 26: Вкладка QUIC IETF в случае запроса quik

# Анализ протоколов транспортного уровня в Wireshark

The screenshot shows the Wireshark interface with the following details:

- File menu:** Файл, Редактирование, Просмотр, Запуск, Захват, Анализ, Статистика, Телефония, Беспроводной, Инструменты, Домашь.
- Toolbar:** Selection, Cut, Copy, Paste, Find, Replace, Select All, Zoom In, Zoom Out, Stop Capturing, Stop Analysis, Stop Statistics, Stop Telephone, Stop Wireless, Stop Tools, Stop Home.
- Search Bar:** quic
- Table View:** Shows a list of network packets. The selected packet (row 399) is highlighted in blue. The columns are: No., Time, Source, Destination, Protocol, Length, Info. The Info column for the selected packet shows: 73 Protected Payload (KPO), DCID=e4766b702b08cb9.
- Packet Details:** UDP payload (31 bytes) is expanded. It shows the QUIC IETF structure:
  - QUIC Connection information [Connection Number: 0]
  - [Packet Length: 31]
  - QUIC Short Header DCID=e4766b702b08cb9d
    - 0... .... = Header Form: Short Header (0)
    - .1... .... = Fixed Bit: True
    - .1... .... = Spin Bit: True
  - Destination Connection ID: e4766b702b08cb9d
  - Remaining Payload: a5647f4943c684da41b3e1b3ff32002394912036fa69
- Hex View:** Shows the raw hex data of the selected packet. The bytes are: 0000 90 f6 52 a1 f7 7c e8 f4 08 c4 e3 ca 08 00 45 00 .R..|... .E. 0010 00 3b 00 00 40 00 40 11 fb 17 c0 a8 01 69 22 a0 ;:..@..i.. 0020 5a e9 b9 ed 01 bb 00 27 3f d3 68 e4 76 6b 78 2b Z....?h.vkp+ 0030 08 cb 9d a5 64 7f 49 43 c6 84 da 41 b3 e1 b3 ff ..d..IC ...A... 0040 32 00 23 94 91 20 36 fa 69 2#.. 6 ..
- Status Bar:** Пакеты: 1259 · Показаны: 11 (0.9%) · Потеряно: 0 (0.0%) · Профиль: Default

Рис. 27: Вкладка QUIC IETF в случае ответа quik

## Анализ handshake протокола TCP в Wireshark

---

# Анализ handshake протокола TCP в Wireshark

Режим активного доступа. Во вкладке с флагами видно, что установлен бит SYN(Syn: set).

Порядковый номер равен 3980370530.

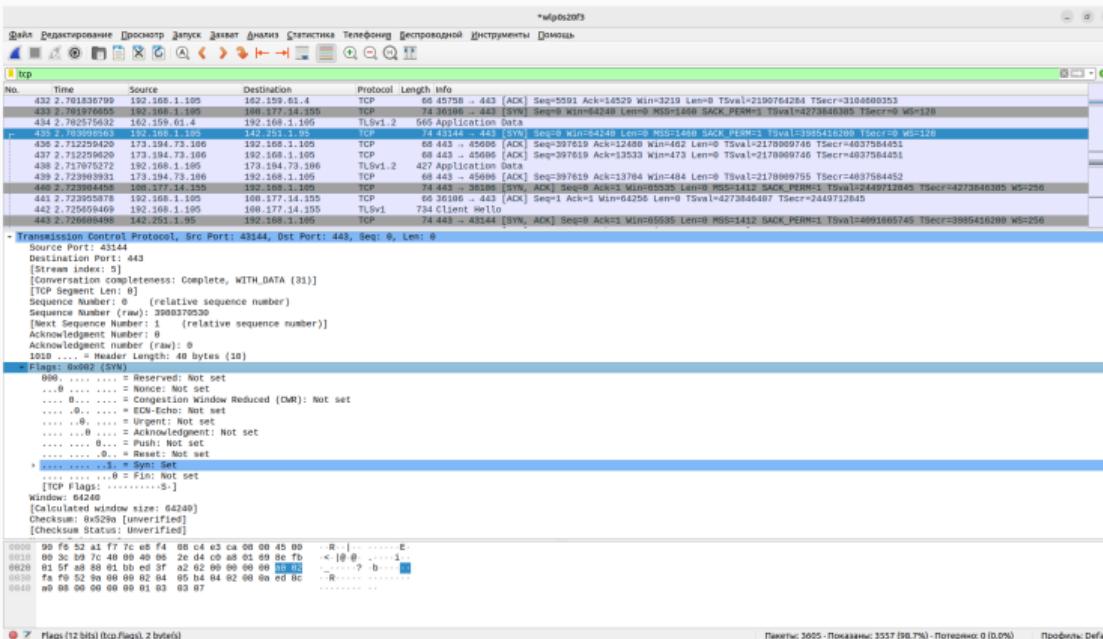


Рис. 28: Первая ступень handshake протокола TCP

# Анализ handshake протокола TCP в Wireshark

Режим пассивного доступа. Во вкладке с флагами видно, что установлены биты SYN и ACK(Syn: set, Acknowldgment: set). Порядковый номер содержит значение ISSb и равен 452625189. Поле номер подтверждения равно значению ISSa, которое получил в предыдущем пакете, плюс 1, то есть 3980370531.

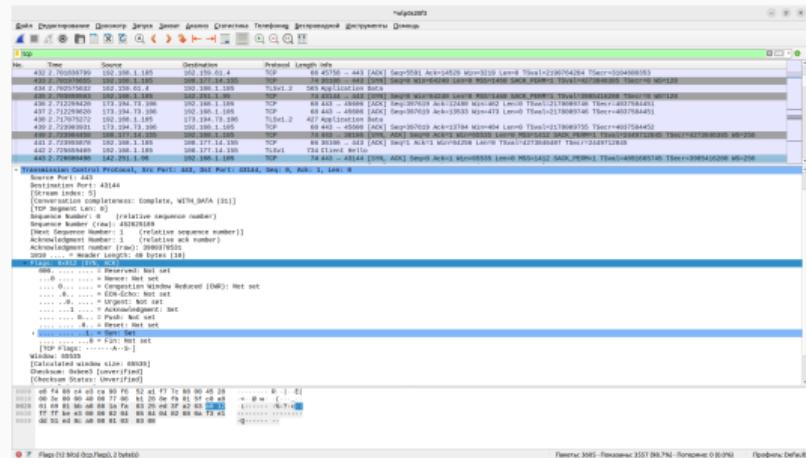


Рис. 29: Вторая ступень handshake протокола TCP

## Анализ handshake протокола TCP в Wireshark

Завершение рукопожатия. Во вкладке с флагами видно, что установлен бит ACK(Acknowledgment: set). Порядковый номер равен ISSa+1 и равен 3980370531. Поле номер подтверждения равно значению ISSb, которое получил в предыдущем пакете, плюс 1, то есть 452625190.

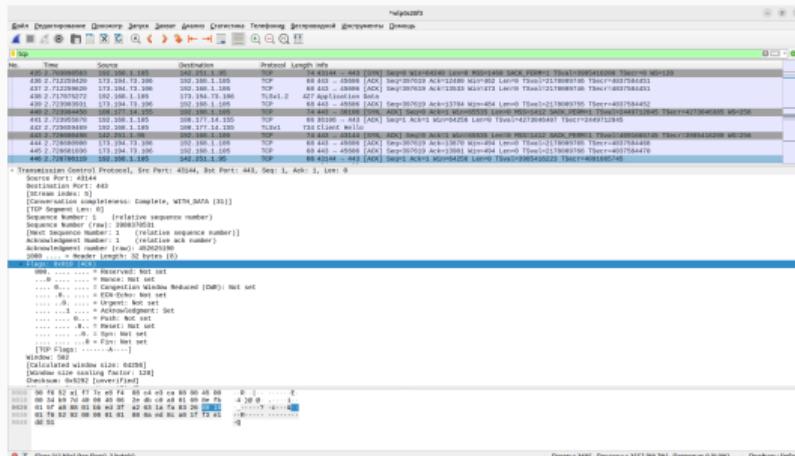


Рис. 30: Третья ступень handshake протокола TCP

## Анализ handshake протокола TCP в Wireshark

На графике видно, что сначала клиент послал сообщение на сервер(стрелка вправо), а значение seq = 0. Затем сервер откликнулся(стрелка влево), значение seq = 0, а значение ack = 1. И в третьем пакете клиент оправил подтверждение получения SYN-сегмента(стрелка влево), оба значения syn и ack стали равны 1.



Рис. 31: График потока

## Заключение

---

## Выводы

---

В результате выполнения лабораторной работы посредством Wireshark были изучены кадры Ethernet, а также проанализированы PDU протоколы транспортного и прикладного уровней стека TCP/IP.