

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

дисциплина: Сетевые технологии

Студент: Бакулин Никита 1032201747

Группа: НПИбд-01-20

МОСКВА

2022 г.

Постановка задачи

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

Выполнение работы

1.

1.1. С помощью команды `ifconfig` для систем типа Linux выведите информацию о текущем сетевом соединении. Используйте опции `-l` для списка интерфейсов и `-v` для более подробной информации.

```
nbakulin@mbp-nbakulin ~ % ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM, TXCSUM, TXSTATUS, SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
anpi0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether 76:07:e2:76:64:b2
    inet6 fe80::7407:e2ff:fe76:64b2%anpi0 prefixlen 64 scopeid 0x4
    nd6 options=201<PERFORMNUD,DAD>
    media: none
    status: inactive
anpi1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether 76:07:e2:76:64:b3
    inet6 fe80::7407:e2ff:fe76:64b3%anpi1 prefixlen 64 scopeid 0x5
    nd6 options=201<PERFORMNUD,DAD>
    media: none
    status: inactive
```

Рисунок 1

```
nbakulin@mbp-nbakulin ~ % ifconfig -l
lo0 gif0 stf0 anpi0 anpi1 anpi2 ap1 en4 en5 en6 en1 en2 en0 en3 awd10 bridge0 llw0 utun1 utun2 utun3 vmenet
0 bridge100 vmenet1 bridge101
```

Рисунок 2

```
nbakulin@mbp-nbakulin ~ % ifconfig -v
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384 index 1
    eflags=12000000<ECN_DISABLE, SENDLIST>
    xflags=4<NOAUTONX>
    options=1203<RXCSUM, TXCSUM, TXSTATUS, SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
    link quality: 100 (good)
    state availability: 0 (true)
    timestamp: disabled
    qosmarking enabled: no mode: none
    low power mode: disabled
    multi layer packet logging (mpklog): disabled
    routermode4: disabled
    routermode6: disabled
```

Рисунок 3

1.2. Определите MAC-адреса сетевых интерфейсов на вашем компьютере. Подтвердите свой ответ скриншотом.

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=6463<RXCSUM, TXCSUM, TSO4, TSO6, CHANNEL_IO, PARTIAL_CSUM, ZEROINVERT_CSUM>
    ether f8:4d:89:66:87:71
```

1.3. Опишите структуру MAC-адресов вашего устройства. Первые три байта идентифицируют производителя. Адрес универсально администрируемый.

MAC address details		Vendor details	
Is valid	True	OUI	F8:4D:89 ⓘ
Virtual Machine	Not detected ⓘ	Is private	False
Transmission type	Unicast ⓘ	Company name	Apple, Inc
Administration type	UAA ⓘ	Company address	1 Infinite Loop Cupertino CA 95014 US
Applications ⓘ	Not detected	Country code	US
Wireshark notes ⓘ	No details		

Рисунок 4

Рисунок 5

2.

- 2.1. Установите на вашем устройстве Wireshark.
- 2.2. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика.
- 2.3. На вашем устройстве в консоли определите с помощью команды `ipconfig` для ОС типа Windows или `ifconfig` для систем типа Linux IP-адрес вашего устройства и шлюз по умолчанию (default gateway).

```
nbakulin@mbp-nbakulin ~ % ifconfig en0
en0: flags=8963<UP, BROADCAST, SMART, RUNNING, PROMISC, SIMPLEX, MULTICAST> mtu 1500
    options=6463<RXCSUM, TXCSUM, TSO4, TSO6, CHANNEL_IO, PARTIAL_CSUM, ZEROINVERT_CSUM>
    ether f8:4d:89:66:87:71
    inet 192.168.1.72 netmask 0xfffff00 broadcast 192.168.1.255
```

Рисунок 6

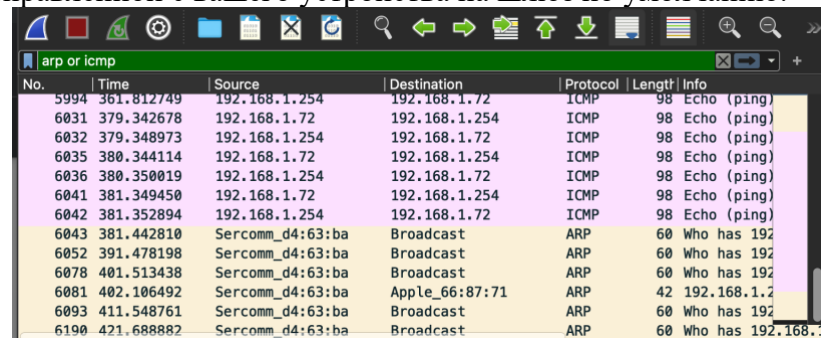
- 2.4. На вашем устройстве в консоли с помощью команды `ping` адрес_шлюза пропингуйте шлюз по умолчанию. Для остановки процесса используйте комбинацию клавиш `Ctrl + c` или изначально при помощи параметров команды `ping` задайте число сообщений эхо-запроса.

```
nbakulin@mbp-nbakulin ~ % ping 192.168.1.254 -c 3
PING 192.168.1.254 (192.168.1.254): 56 data bytes
64 bytes from 192.168.1.254: icmp_seq=0 ttl=64 time=6.388 ms
64 bytes from 192.168.1.254: icmp_seq=1 ttl=64 time=6.121 ms
64 bytes from 192.168.1.254: icmp_seq=2 ttl=64 time=3.636 ms

--- 192.168.1.254 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.636/5.382/6.388/1.239 ms
```

Рисунок 7

- 2.5. В Wireshark остановите захват трафика. В строке фильтра пропишите фильтр `arp or icmp`. Убедитесь, что в списке пакетов отобразятся только пакеты ARP или ICMP, в частности пакеты, которые были сгенерированы с помощью команды `ping`, отправленной с вашего устройства на шлюз по умолчанию.



No.	Time	Source	Destination	Protocol	Length	Info
5994	361.812749	192.168.1.254	192.168.1.72	ICMP	98	Echo (ping)
6031	379.342678	192.168.1.72	192.168.1.254	ICMP	98	Echo (ping)
6032	379.348973	192.168.1.254	192.168.1.72	ICMP	98	Echo (ping)
6035	380.344114	192.168.1.72	192.168.1.254	ICMP	98	Echo (ping)
6036	380.350019	192.168.1.254	192.168.1.72	ICMP	98	Echo (ping)
6041	381.349450	192.168.1.72	192.168.1.254	ICMP	98	Echo (ping)
6042	381.352894	192.168.1.254	192.168.1.72	ICMP	98	Echo (ping)
6043	381.442810	Sercomm_d4:63:ba	Broadcast	ARP	60	Who has 192
6052	391.478198	Sercomm_d4:63:ba	Broadcast	ARP	60	Who has 192
6078	401.513438	Sercomm_d4:63:ba	Broadcast	ARP	60	Who has 192
6081	402.106492	Sercomm_d4:63:ba	Apple_66:87:71	ARP	42	192.168.1.72
6093	411.548761	Sercomm_d4:63:ba	Broadcast	ARP	60	Who has 192
6190	421.688882	Sercomm_d4:63:ba	Broadcast	ARP	60	Who has 192.168.1

Рисунок 8

- 2.6. Изучите эхо-запрос и эхо-ответ ICMP в программе Wireshark:
 - На панели списка пакетов (верхний раздел) выберите первый указанный кадр ICMP — эхо-запрос. Изучите информацию на панели сведений о пакете в средней части экрана. Длина кадра 98 байт, IPv4, мас адреса представлены на картинке (destination и source).
 - На панели списка пакетов (верхний раздел) выберите второй указанный кадр ICMP — эхо-ответ. Изучите информацию на панели сведений о пакете в средней части экрана. Длина кадра 98 байт, IPv4, мас адреса представлены на картинке (destination и source).

```
Frame 5933: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0,
Ethernet II, Src: Apple_66:87:71 (f8:4d:89:66:87:71), Dst: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
> Destination: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
> Source: Apple_66:87:71 (f8:4d:89:66:87:71)
Type: IPv4 (0x0800)
```

Рисунок 9

```

> Frame 5934: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0
✓ Ethernet II, Src: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba), Dst: Apple_66:87:71 (f8:4d:89:66:87:71)
  > Destination: Apple_66:87:71 (f8:4d:89:66:87:71)
  > Source: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
  Type: IPv4 (0x0800)

```

Рисунок 10

2.7. Изучите кадры данных протокола ARP. Type: ARP

```

> Frame 146: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0
✓ Ethernet II, Src: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba), Dst: Apple_66:87:71 (f8:4d:89:66:87:71)
  > Destination: Apple_66:87:71 (f8:4d:89:66:87:71)
  > Source: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
  Type: ARP (0x0806)

```

Рисунок 11

2.8. Начните новый процесс захвата трафика в Wireshark. На вашем устройстве в консоли пропингуйте по имени какой-нибудь известный вам адрес, например ping ya.ru.

```

[nbakulin@mbp-nbakulin ~ % ping ya.ru -c 3
PING ya.ru (87.250.250.242): 56 data bytes
64 bytes from 87.250.250.242: icmp_seq=0 ttl=246 time=17.146 ms
64 bytes from 87.250.250.242: icmp_seq=1 ttl=246 time=17.064 ms
64 bytes from 87.250.250.242: icmp_seq=2 ttl=246 time=17.316 ms

--- ya.ru ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.064/17.175/17.316/0.105 ms

```

Рисунок 12

2.9. В Wireshark остановите захват трафика. Изучите запросы и ответы протоколов ARP и ICMP. Определите MAC-адреса источника и получателя, определите тип MAC-адресов.

```

> Frame 7: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0,
✓ Ethernet II, Src: Apple_66:87:71 (f8:4d:89:66:87:71), Dst: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
  > Destination: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
  > Source: Apple_66:87:71 (f8:4d:89:66:87:71)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.72, Dst: 87.250.250.242

```

Рисунок 13

```

> Frame 8: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0
✓ Ethernet II, Src: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba), Dst: Apple_66:87:71 (f8:4d:89:66:87:71)
  > Destination: Apple_66:87:71 (f8:4d:89:66:87:71)
  > Source: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 87.250.250.242, Dst: 192.168.1.72

```

Рисунок 14

3.

- 3.1. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика
- 3.2. На вашем устройстве в браузере перейдите на сайт, работающий по протоколу HTTP (например, на сайт CERN <http://info.cern.ch/>). При необходимости получения большей информации для Wireshark перемещайтесь по ссылкам или разделам сайта в браузере.
- 3.3. В Wireshark в строке фильтра укажите http и проанализируйте информацию по протоколу TCP в случае запросов и ответов. IPv4 -> TCP -> HTTP. HTTP исходящий GET, в ответ статус OK.

```

> Frame 74: 600 bytes on wire (4800 bits), 600 bytes captured (4800 bits) on interface en0, id 1
Ethernet II, Src: Apple_66:87:71 (f8:4d:89:66:87:71), Dst: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
  > Destination: Sercomm_d4:63:ba (94:4a:0c:d4:63:ba)
  > Source: Apple_66:87:71 (f8:4d:89:66:87:71)
  Type: IPv6 (0x86dd)
  > Internet Protocol Version 6, Src: 2a00:1370:8188:4a99:195f:7e51:e98d:9f7e, Dst: 2001:1458:d00
  > Transmission Control Protocol, Src Port: 55692, Dst Port: 80, Seq: 1, Ack: 1, Len: 514
    Source Port: 55692
    Destination Port: 80
    [Stream index: 1]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 514]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 1957077007
    [Next Sequence Number: 515 (relative sequence number)]
    Acknowledgment Number: 1 (relative ack number)
    Acknowledgment number (raw): 545362199
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
  Window: 2052
  [Calculated window size: 131328]
  [Window size scaling factor: 64]
  Checksum: 0x9612 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]

```

Рисунок 15

```

Hypertext Transfer Protocol
  > GET /hypertext/WWW/TheProject.html HTTP/1.1\r\n
    Host: info.cern.ch\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.164 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/svg+xml,*/*;q=0.8\r\n
    Referer: http://info.cern.ch/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7\r\n
  \r\n
  [Full request URI: http://info.cern.ch/hypertext/WWW/TheProject.html]
  [HTTP request 1/1]
  [Response in frame: 86]

```

Рисунок 16

```

Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Sat, 24 Sep 2022 19:15:58 GMT\r\n
    Server: Apache\r\n
    Last-Modified: Thu, 03 Dec 1992 08:37:20 GMT\r\n
    ETag: "8a9-291e721905000"\r\n
    Accept-Ranges: bytes\r\n
  > Content-Length: 2217\r\n
  Connection: close\r\n
  Content-Type: text/html\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.080250000 seconds]
  [Request in frame: 74]
  [Request URI: http://info.cern.ch/hypertext/WWW/TheProject.html]
  File Data: 2217 bytes

```

Рисунок 17

3.4. Wireshark в строке фильтра укажите dns и проанализируйте информацию по протоколу UDP в случае запросов и ответов. IPv4 -> UDP -> DNS

```

Frame 219: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface en0,
Ethernet II, Src: Apple_66:87:71 (f8:4d:89:66:87:71), Dst: Sercomm_d4:63:ba (94:4a:0c:
Internet Protocol Version 4, Src: 192.168.1.72, Dst: 192.168.1.254
User Datagram Protocol, Src Port: 1797, Dst Port: 53
  Source Port: 1797
  Destination Port: 53
  Length: 41
  Checksum: 0x368e [unverified]
  [Checksum Status: Unverified]
  [Stream index: 5]
  > [Timestamps]
  UDP payload (33 bytes)
  Domain Name System (query)

```

Рисунок 18

3.5. Wireshark в строке фильтра укажите quic и проанализируйте информацию по протоколу quic в случае запросов и ответов. IPv6 -> UDP -> QUIC. Handshake.

```

> Frame 72: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface en0,
> Ethernet II, Src: Apple_66:87:71 (f8:4d:89:66:87:71), Dst: Sercomm_d4:63:ba (94:4a:0c:d4:63:b
> Internet Protocol Version 6, Src: 2a00:1370:8188:4a99:195f:7e51:e98d:9f7e, Dst: 2a00:1450:401
> User Datagram Protocol, Src Port: 52052, Dst Port: 443
< QUIC IETF
  > QUIC Connection information
    [Packet Length: 1230]
    1... .... = Header Form: Long Header (1)
    .1.. .... = Fixed Bit: True
    ..00 .... = Packet Type: Initial (0)
    .... 00.. = Reserved: 0
    .... ..00 = Packet Number Length: 1 bytes (0)
    Version: 1 (0x00000001)
    Destination Connection ID Length: 8
    Destination Connection ID: d9ae9e3efe277c9b
    Source Connection ID Length: 0
    Token Length: 70
    Token: 009b66ea308ed8a67eb555a20f1b2350d0e84ca4dd9ef11ecce3dcd84bb999985ba89a9a...
    Length: 1141
    Packet Number: 1
    Payload: 95ee5063c1343b3b46034482ee8a5a9958fb2f32c2e552ade75934551ca2309a225bc73a...
  > CRYPTO
  > PADDING Length: 284
  > PING
  > PADDING Length: 87

```

Рисунок 19

Source	Destination	Protocol	Length	Info
2a00:1370:818...	2a00:1450:4010:c0...	QUIC	1292	Initial, DCID=d9ae9e3efe277c9b, PKN: 1, CRYPTO, PADDING, PING, PADDING
2a00:1370:818...	2a00:1450:4010:c0...	QUIC	141	0-RTT, DCID=d9ae9e3efe277c9b
2a00:1450:401...	2a00:1370:8188:4a...	QUIC	1292	Protected Payload (KP0)
2a00:1370:818...	2a00:1450:4010:c0...	QUIC	140	Handshake, DCID=d9ae9e3efe277c9b

Рисунок 20

3.6. Остановите захват трафика в Wireshark.

4.

- 4.1. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика.
- 4.2. На вашем устройстве или используйте подсоединение по telnet или ssh к вашему маршрутизатору (например с помощью PUTTY или соответствующих команд в консоли), или соединение по HTTP с каким-то сайтом для захвата в Wireshark пакетов TCP.
- 4.3. В Wireshark проанализируйте handshake протокола TCP. SYN -> SYN, ACK -> ACK. Sequence Number в Acknowledgment Number (+1)

Source	Destination	Protocol	Length	Info
192.168.1.72	185.189.255.78	TCP	78	55897 → 22 [SYN] Seq=0 Win=65535
185.189.255.78	192.168.1.72	TCP	74	22 → 55897 [SYN, ACK] Seq=0 Ack=1
192.168.1.72	185.189.255.78	TCP	66	55897 → 22 [ACK] Seq=1 Ack=1 Win=

Рисунок 21

```

Transmission Control Protocol, Src Port: 55897, Dst Port: 22, Seq: 0, Len: 0
  Source Port: 55897
  Destination Port: 22
  [Stream index: 23]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 1170874376
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1011 .... = Header Length: 44 bytes (11)
  > Flags: 0x002 (SYN)

```

Рисунок 22

```

Transmission Control Protocol, Src Port: 22, Dst Port: 55897, Seq: 0, Ack: 1, Len: 0
Source Port: 22
Destination Port: 55897
[Stream index: 23]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 1967593994
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1170874377
1010 .... = Header Length: 40 bytes (10)
> Flags: 0x012 (SYN, ACK)

```

Рисунок 23

```

Transmission Control Protocol, Src Port: 55897, Dst Port: 22, Seq: 1, Ack: 1, Len: 0
Source Port: 55897
Destination Port: 22
[Stream index: 23]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1170874377
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1967593995
1000 .... = Header Length: 32 bytes (8)
> Flags: 0x010 (ACK)

```

Рисунок 24

4.4. В Wireshark в меню «Статистика» выберите «График Потока». В отчёте приведите пояснения по изменениям значений соответствующих сообщений при установлении соединения по TCP.

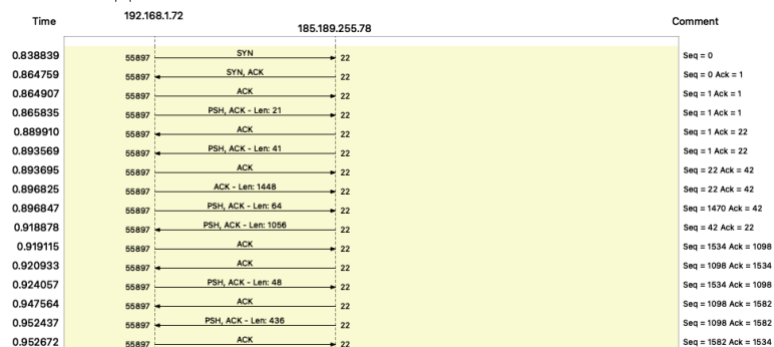


Рисунок 25

4.5. Остановите захват трафика в Wireshark.