

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Направление подготовки 11.03.02

Практическая работа №5

Изучение работы протоколов стека TCP/IP  
с помощью Wireshark

Выполнил:

Швалов Даниил Андреевич

Группа: К33211

Проверил:

Харитонов Антон

Санкт-Петербург

2023

## 1. Введение

**Цель работы:** разобраться со стеком TCP/IP, анализируя пакеты, которые отправляются и принимаются с помощью данного стека, научиться собирать сетевой трафик с помощью программы Wireshark, научиться фильтровать собранный трафик, находить и просматривать соединения.

## 2. Ход работы

### 2.1. Начало работы с Wireshark

Для того, чтобы настроить перехват трафика на интерфейсе так, чтобы он завершился после сбора 5 Мб, необходимо на панели нажать «Capture», перейти в «Options» и во вкладке «Options» в разделе «Stop capture automatically after» указать «5 megabytes» (рис. 1).

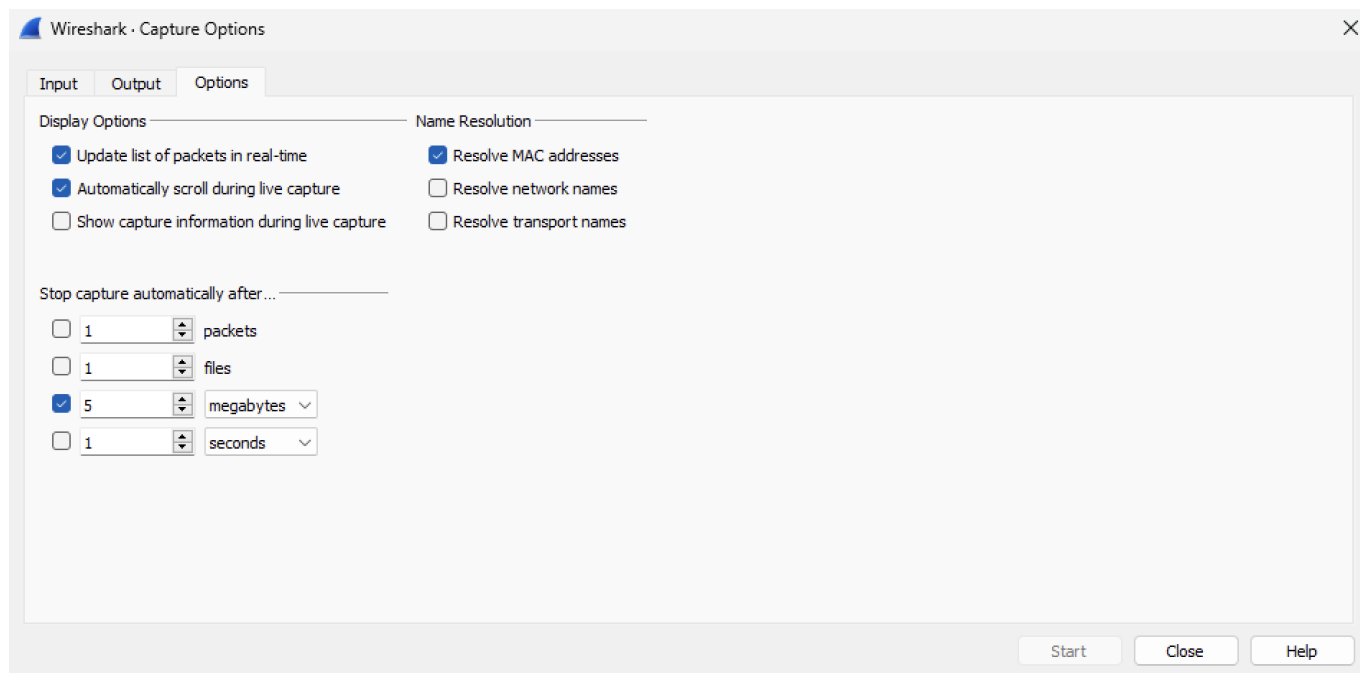


Рисунок 1 – Настройка автоматического завершения перехвата трафика

Для того, чтобы настроить сохранение информации о трафике в файл, необходимо в том же окне во вкладке «Output» прописать путь до директории, в котором будет сохранен файл, а также его название (рис. 2).

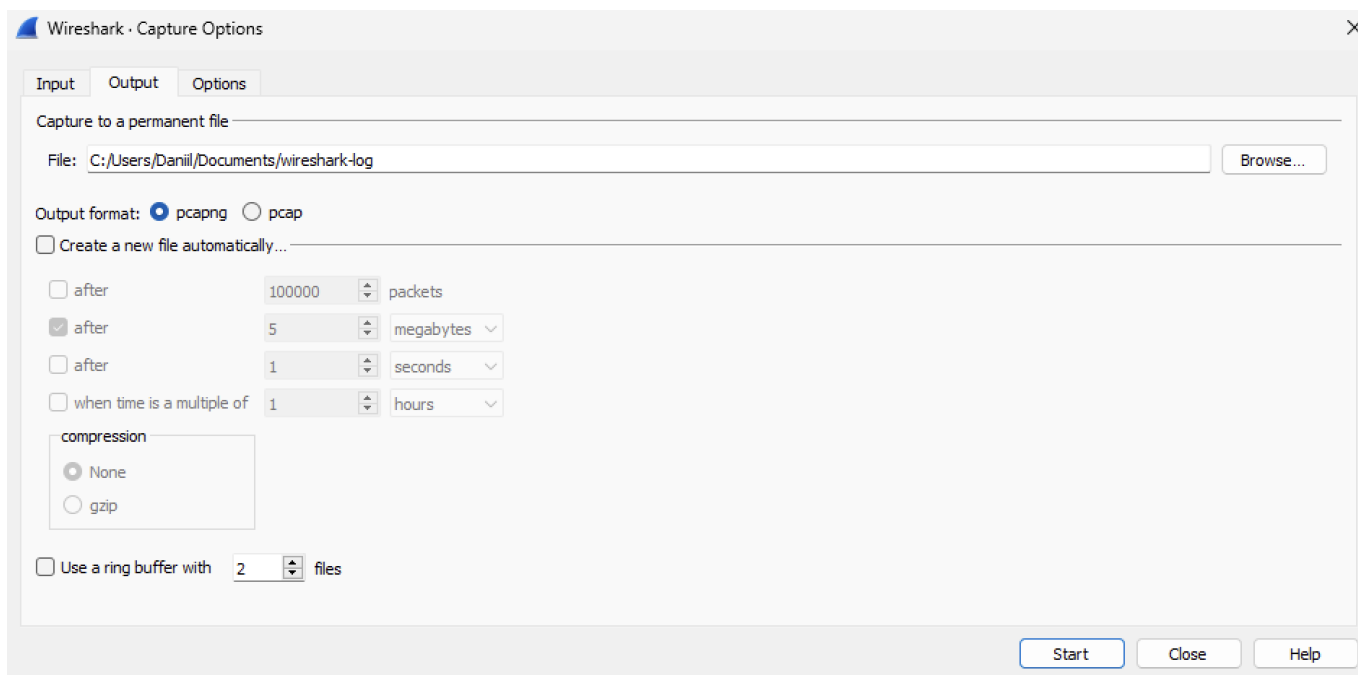


Рисунок 2 – Настройка сохранения данных в файл

Для определения узла с максимальной активностью по объему переданных данных необходимо на панели нажать «Statistics», «Endpoints». В открывшемся окне перейти на вкладку «IPv4» и, поскольку активность оценивается по объему **переданных** данных, необходимо отсортировать данный список по количеству «Tx Bytes» (рис. 3). В данном случае это узел с IPv4 адресом 172.67.186.132.

The image shows the 'Wireshark · Endpoints · wireshark-log.pcapng' window. It displays a table of network endpoints. The table has columns for Address, Packets, Bytes, Tx Packets, Tx Bytes, Rx Packets, Rx Bytes, Country, City, AS Number, and AS Organization. The first row is highlighted in blue.

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number	AS Organization
172.67.186.132	946	910 k	760	882 k	186	27 k	—	—	—	—
192.168.64.2	6,908	4765 k	2,326	659 k	4,582	4105 k	—	—	—	—
87.245.209.215	1,166	665 k	760	606 k	406	59 k	—	—	—	—
13.107.21.200	1,563	824 k	976	570 k	587	254 k	—	—	—	—
151.101.246.219	415	439 k	334	431 k	81	8029	—	—	—	—
104.18.202.232	369	373 k	263	362 k	106	10 k	—	—	—	—
63.250.38.192	277	269 k	209	261 k	68	8258	—	—	—	—
64.233.165.97	198	187 k	149	180 k	49	6633	—	—	—	—
213.246.63.45	153	136 k	118	131 k	35	4550	—	—	—	—
204.79.197.203	174	138 k	137	128 k	37	9415	—	—	—	—
64.233.161.94	103	114 k	85	111 k	18	2347	—	—	—	—
108.156.20.98	102	102 k	75	99 k	27	2811	—	—	—	—
192.0.77.2	65	53 k	44	48 k	21	5261	—	—	—	—
64.233.162.139	68	42 k	46	37 k	22	5136	—	—	—	—
80.239.137.154	42	29 k	29	27 k	13	1868	—	—	—	—
20.103.180.120	63	38 k	33	27 k	30	10 k	—	—	—	—
192.168.64.1	278	35 k	138	25 k	140	10 k	—	—	—	—
2.23.144.241	33	24 k	24	22 k	9	1728	—	—	—	—
204.79.197.200	50	22 k	29	17 k	21	4945	—	—	—	—
104.208.16.90	209	137 k	91	15 k	118	121 k	—	—	—	—
204.79.197.219	43	17 k	26	14 k	17	2916	—	—	—	—
192.0.76.3	38	14 k	21	11 k	17	2634	—	—	—	—
23.72.139.65	62	14 k	38	9419	24	5504	—	—	—	—
108.177.14.95	28	10 k	16	8761	12	1882	—	—	—	—

At the bottom, there are checkboxes for 'Name resolution' and 'Limit to display filter'. On the right, there are buttons for 'Copy', 'Map', 'Close', and 'Help', along with an 'Endpoint Types' dropdown menu.

Рисунок 3 – Узлы с максимальной активностью по объему переданных данных

Для определения самого активного TCP-порта на хосте по количеству переданных пакетов необходимо в поле ввода ввести следующее условие фильтрации:

`ip.src == 192.168.64.2`

Тем самым все пакеты будут отфильтрованы по адресу отправителя. В данном случае адрес хоста равен 192.168.64.2. Затем с помощью окна «Endpoints», с включенной опцией «Limit to display filter». Во вкладке «TCP» необходимо отсортировать список по «Tx Bytes», поскольку активность определяется по количеству переданных пакетов. В данном случае самым активным TCP-портом является 50162 (рис. 4).

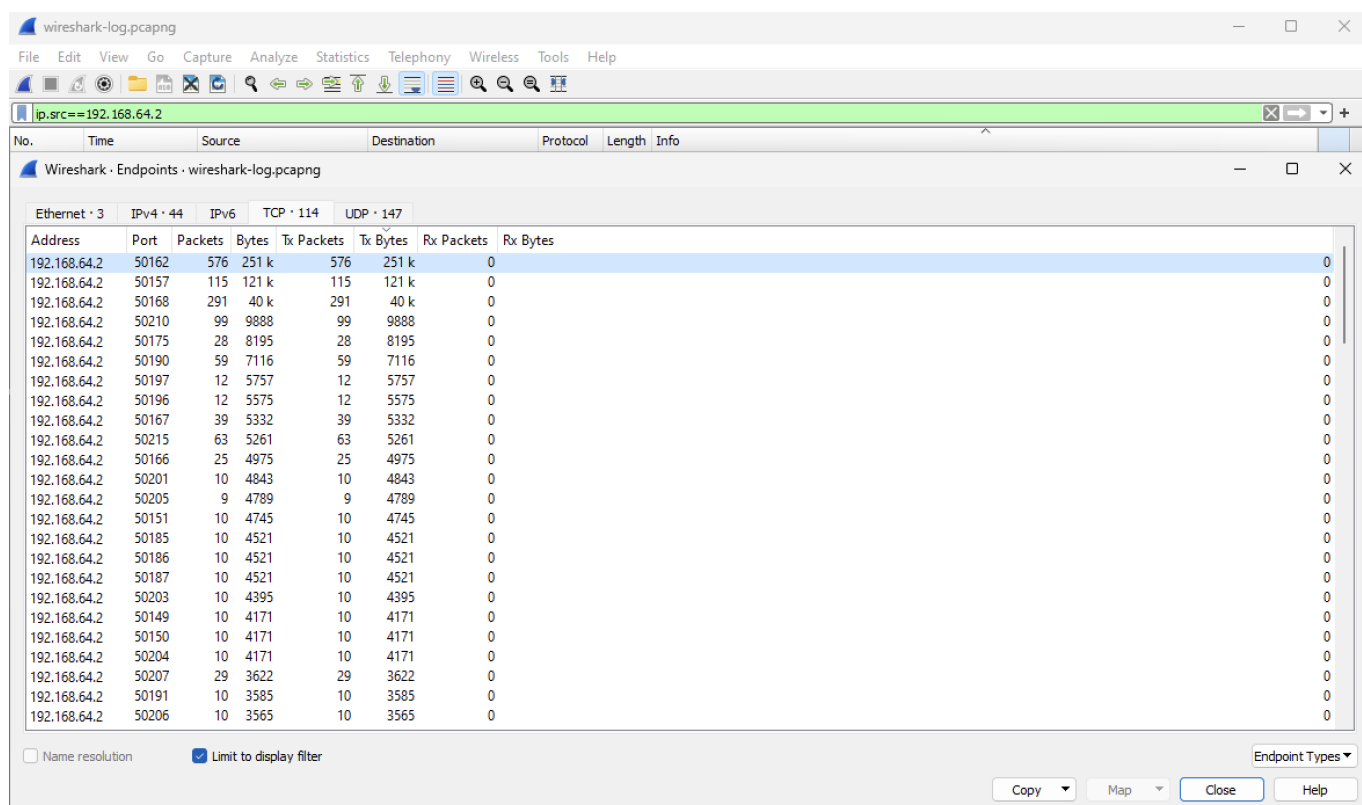


Рисунок 4 – Самый активный TCP-порт на хосте

Для построения графа интенсивности TCP и UDP трафика необходимо на панели в разделе «Statistics» открыть окно «I/O Graphs». Там в качестве графиков необходимо добавить два источника данных: первый с «Display Filter» равный «tcp», второй с «Display Filter» равный «udp» (рис. 5).

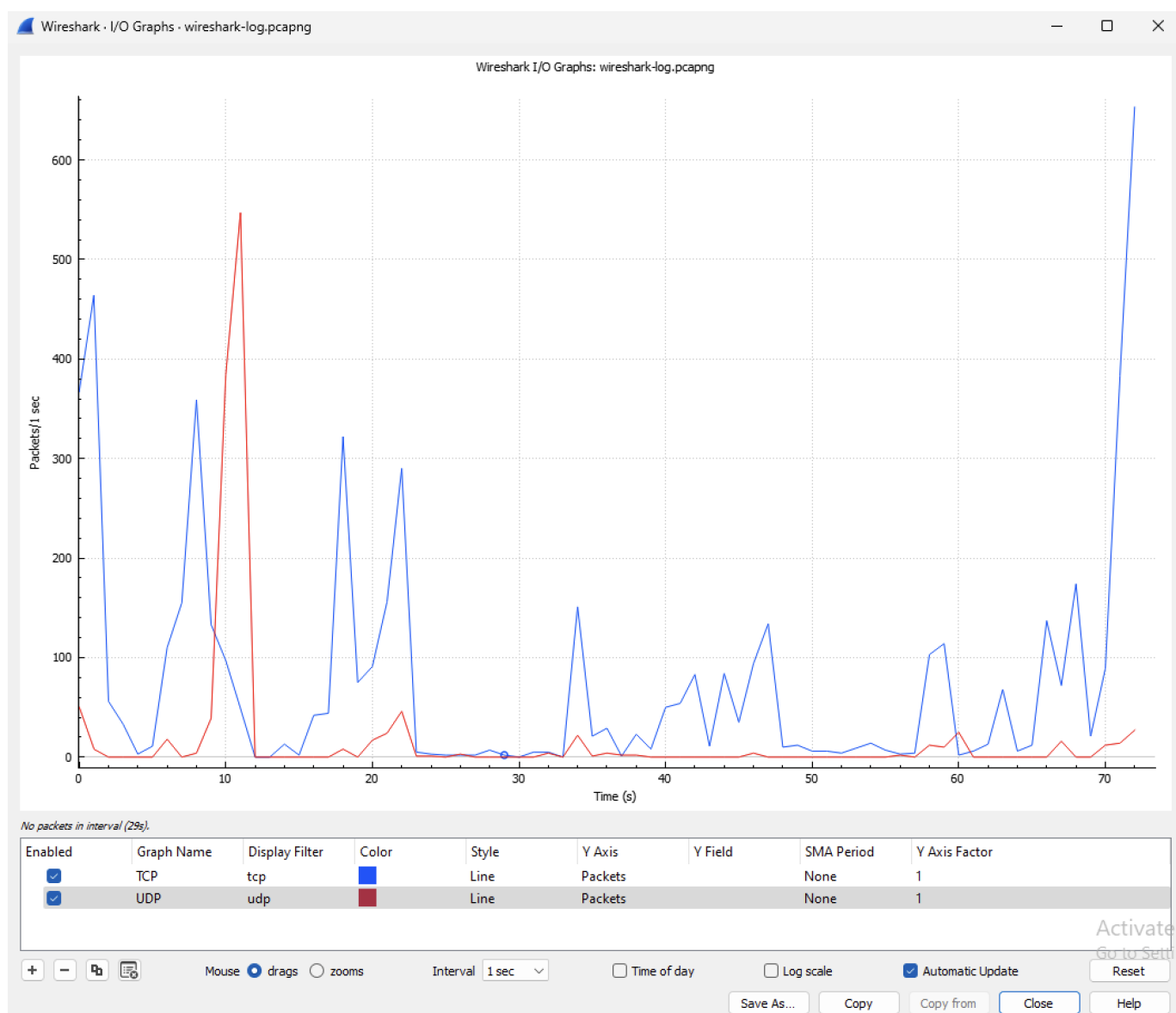


Рисунок 5 – График интенсивности TCP и UDP трафика

Для построения диаграммы связей пакетов HTTPS необходимо в поле фильтрации ввести следующее условие:

`tcp.port==443`

Таким образом, поскольку HTTPS работает поверх протокола TCP на порту 443, весь трафик будет отфильтрован по протоколу HTTPS. Затем на панели в разделе «Statistics» необходимо открыть «Flow graph». В открывшемся окне необходимо включить опцию «Limit to display filter», а также в выпадающем поле «Flow type» выбрать «TCP Flows». Тем самым будет построена диаграмма связей для HTTPS пакетов (рис. 6).

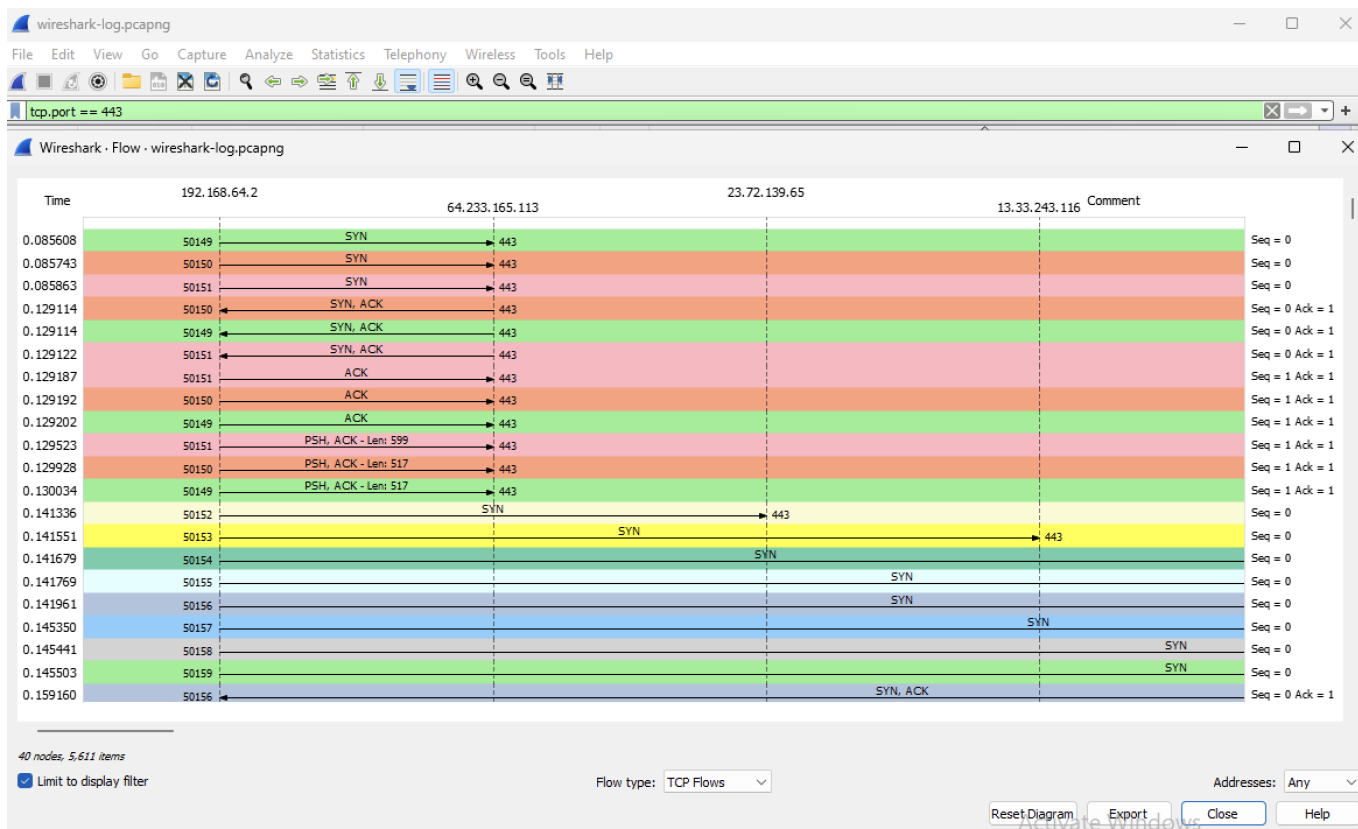


Рисунок 6 – Диаграмма связей HTTPS пакетов

В моем случае не было зафиксировано HTTP трафика, поэтому для фильтрации использовался протокол HTTPS. Для фильтрации HTTP трафика достаточно изменить TCP-порт с 443 на 80. Для того, чтобы отфильтровать HTTPS трафик только для клиента, исключив пакеты к серверу, воспользуемся следующим фильтром:

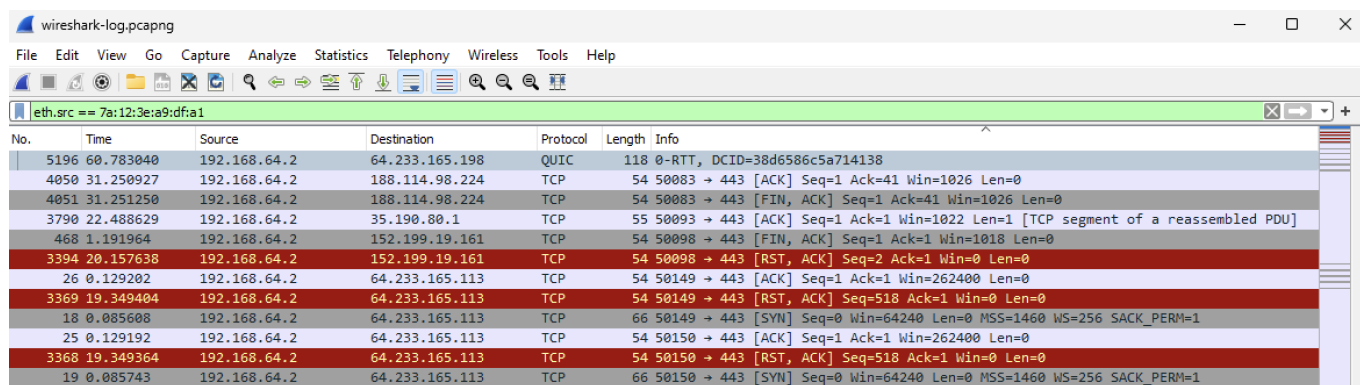
```
(tcp.dstport == 443 and ip.src == 192.168.64.2) or
(tcp.srcport == 443 and ip.dst == 192.168.64.2)
```

В данном фильтре выбирается весь трафик, который либо отправляется хостом на внешний сервер на порт 443, либо отправляется внешним сервером с порта 443 на адрес хоста. В данный фильтр не попадет трафик сервера, поскольку сервер либо отправляет с 443 порта на порт, отличный от 443, либо принимает на 443 порт с порта, отличного от 443.

Для фильтрации всех кадров Ethernet, отправленных с сетевого интерфейса хоста, необходимо воспользоваться следующим фильтром:

```
eth.src == 7a:12:3e:a9:df:a1
```

Тем самым, поскольку в данном случае MAC-адрес хоста равен 7a:12:3e:a9:df:a1, будут показаны только пакеты, отправляемые с сетевого интерфейса хоста (рис. 7).



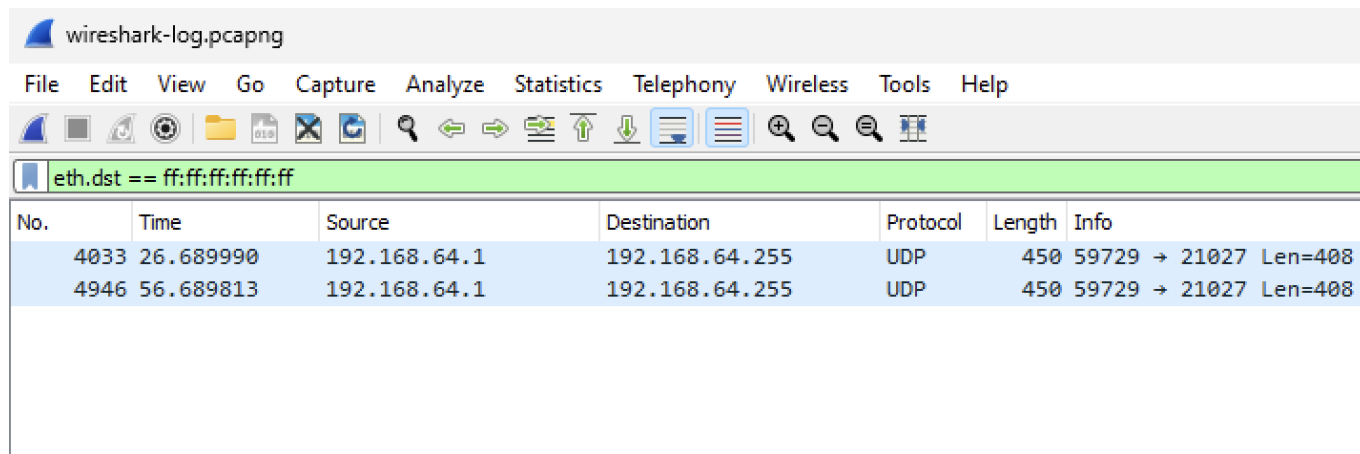
No.	Time	Source	Destination	Protocol	Length	Info
5196	60.783040	192.168.64.2	64.233.165.198	QUIC	118	0-RTT, DCID=38d6586c5a714138
4050	31.250927	192.168.64.2	188.114.98.224	TCP	54	50083 → 443 [ACK] Seq=1 Ack=41 Win=1026 Len=0
4051	31.251250	192.168.64.2	188.114.98.224	TCP	54	50083 → 443 [FIN, ACK] Seq=1 Ack=41 Win=1026 Len=0
3790	22.488629	192.168.64.2	35.190.80.1	TCP	55	50093 → 443 [ACK] Seq=1 Ack=1 Win=1022 Len=1 [TCP segment of a reassembled PDU]
468	1.191964	192.168.64.2	152.199.19.161	TCP	54	50098 → 443 [FIN, ACK] Seq=1 Ack=1 Win=1018 Len=0
3394	20.157638	192.168.64.2	152.199.19.161	TCP	54	50098 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
26	0.129202	192.168.64.2	64.233.165.113	TCP	54	50149 → 443 [ACK] Seq=1 Ack=1 Win=262400 Len=0
3369	19.349404	192.168.64.2	64.233.165.113	TCP	54	50149 → 443 [RST, ACK] Seq=518 Ack=1 Win=0 Len=0
18	0.085608	192.168.64.2	64.233.165.113	TCP	66	50149 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
25	0.129192	192.168.64.2	64.233.165.113	TCP	54	50150 → 443 [ACK] Seq=1 Ack=1 Win=262400 Len=0
3368	19.349364	192.168.64.2	64.233.165.113	TCP	54	50150 → 443 [RST, ACK] Seq=518 Ack=1 Win=0 Len=0
19	0.085743	192.168.64.2	64.233.165.113	TCP	66	50150 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

Рисунок 7 – Кадры Ethernet, отправляемые с сетевого интерфейса хоста

Для того, чтобы отфильтровать только широковещательные сообщения, необходимо использовать следующий фильтр:

`eth.dst == ff:ff:ff:ff:ff:ff`

Таким образом будут показаны только те пакеты, которые были отправлены на широковещательный адрес. Как видно на рис. 8, среди протоколов, есть только UDP.



No.	Time	Source	Destination	Protocol	Length	Info
4033	26.689990	192.168.64.1	192.168.64.255	UDP	450	59729 → 21027 Len=408
4946	56.689813	192.168.64.1	192.168.64.255	UDP	450	59729 → 21027 Len=408

Рисунок 8 – Список широковещательных сообщений

Исходя из того, что широковещательных рассылок было достаточно мало, можно сказать, что устройство было подключено к маршрутизатору.

## 2.2. Сбор и анализ данных протокола ICMP

Для разрешения ICMP-запросов в Windows необходимо настроить брандмауэр. Для этого нужно создать новое Inbound правило (рис. 9), которое будет пропускать все входящие пакеты по протоколу ICMPv4 (рис. 10 и 11).

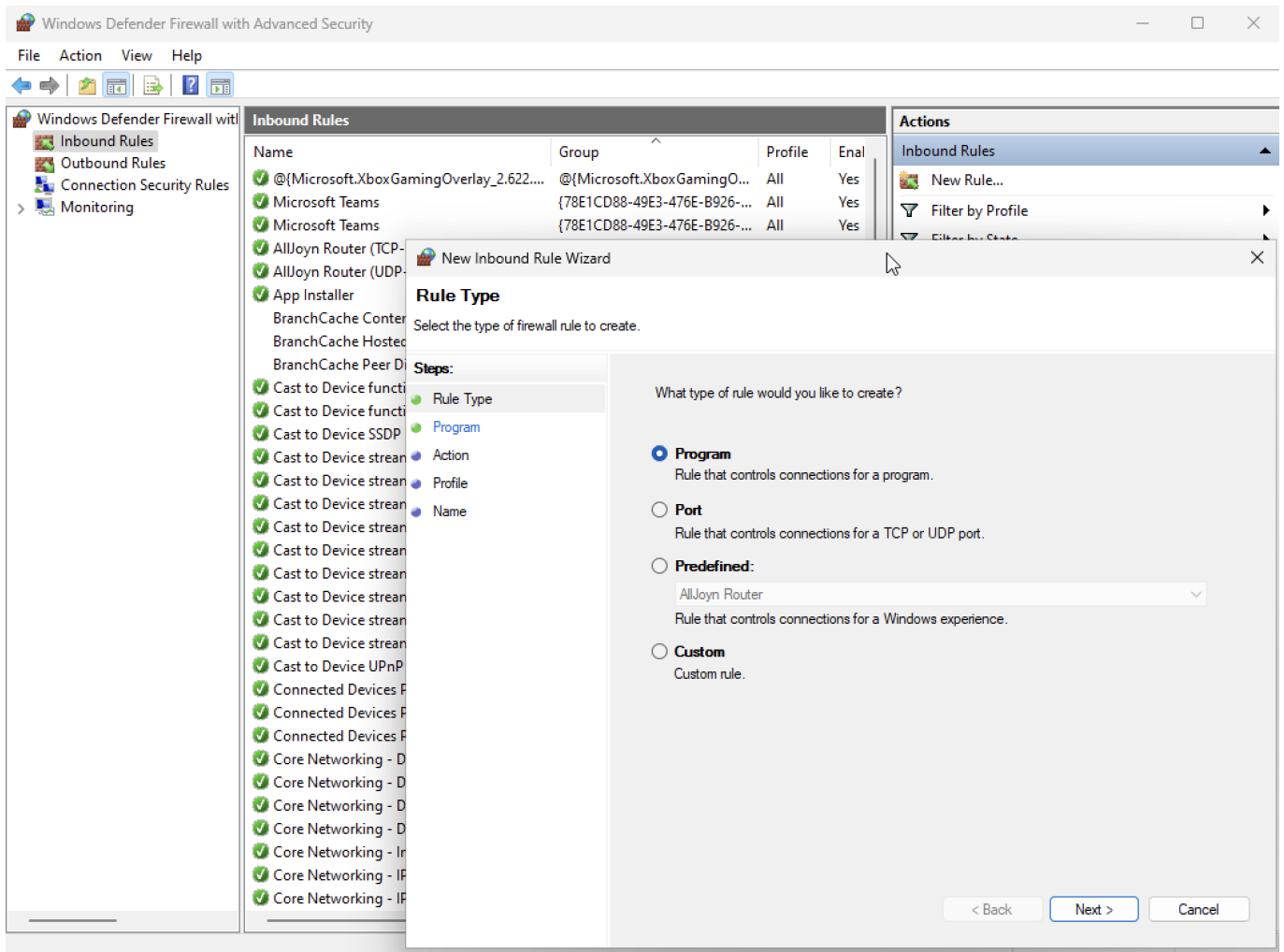


Рисунок 9 – Создание нового правила в брандмауэре Windows



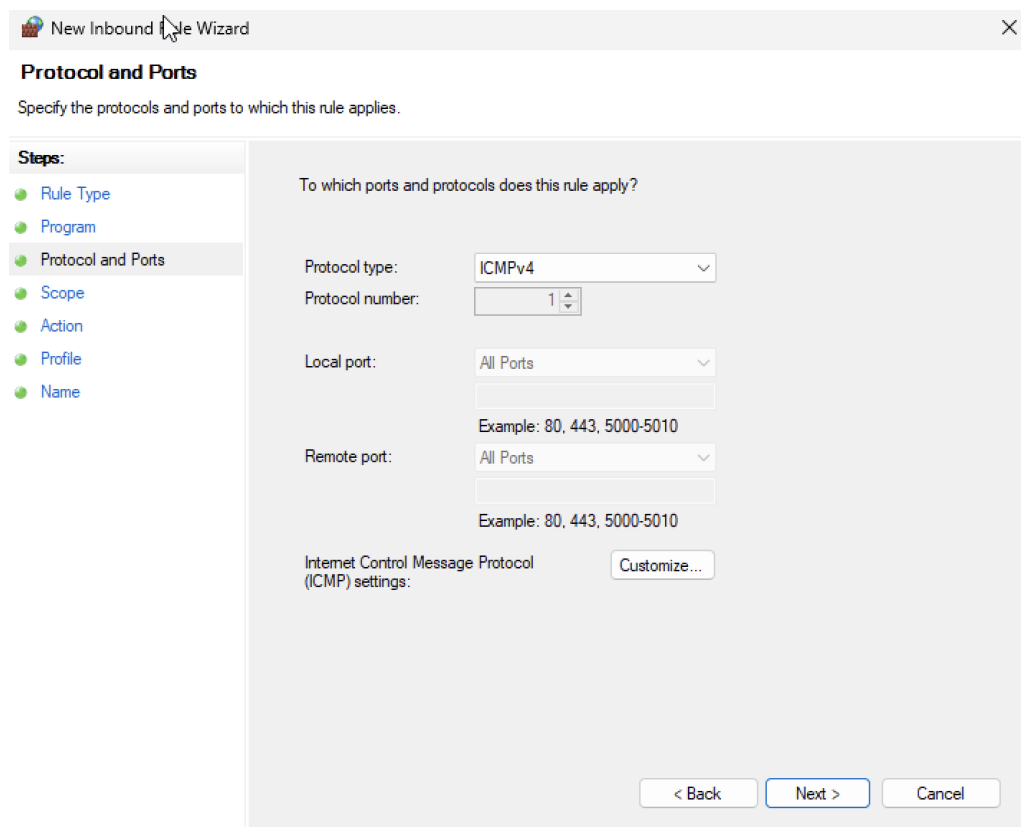


Рисунок 10 – Выбор протокола при создании нового правила

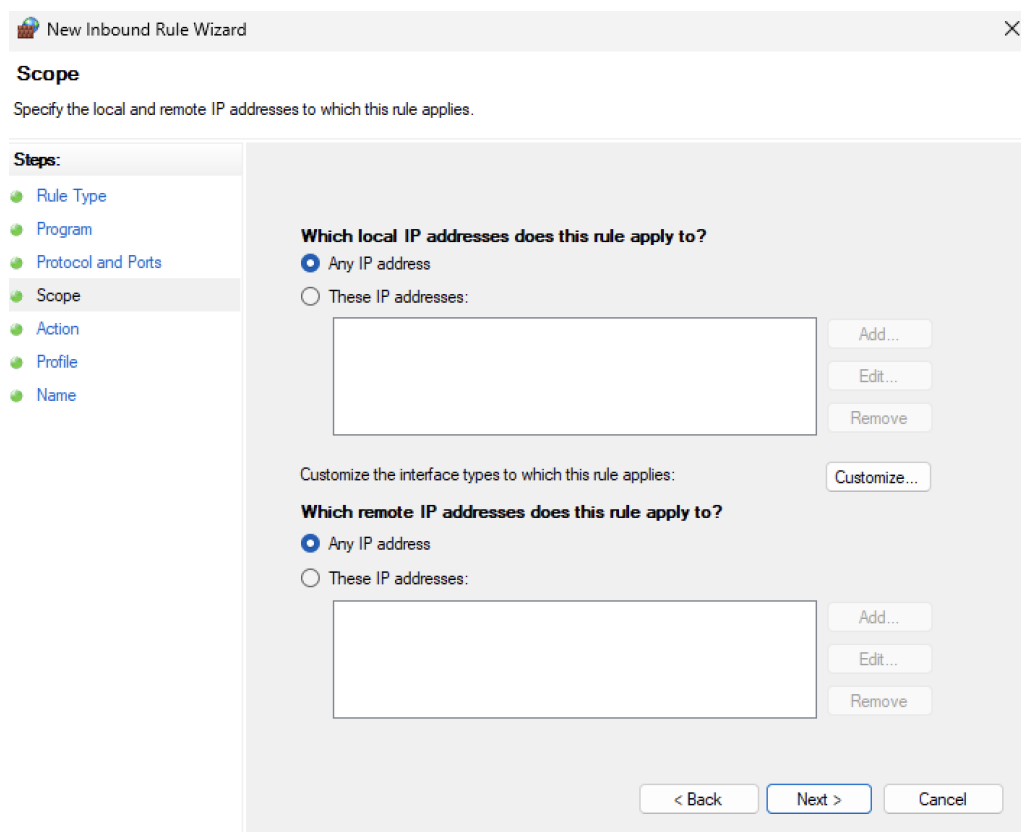


Рисунок 11 – Разрешение входящих запросов с любого IP-адреса

В качестве локального устройства, которому будут отправляться ICMP запросы, был выбран компьютер с IP-адресом 192.168.1.254. Хост, в свою очередь, имеет адрес IP-адрес 192.168.1.101.

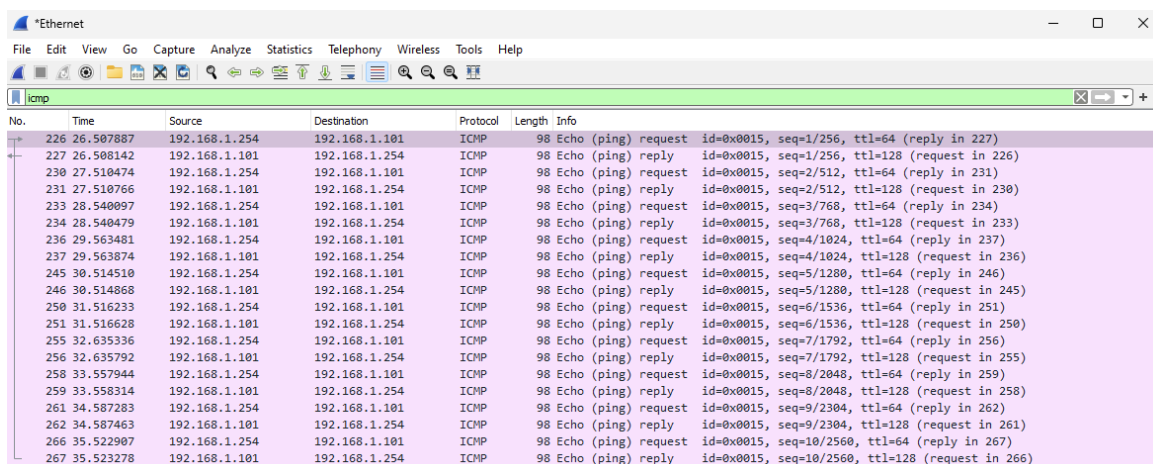
Для фильтрации ICMP-пакетов, отправляемых при ping-запросе, необходимо использовать следующий фильтр:

```
icmp
```

После запуска перехвата пакетов в Wireshark, а также после запуска утилиты ping на другом локальном компьютере (рис. 12), через некоторое время в окне Wireshark появится информация о перехваченных ICMP-пакетах (рис. 13). Как можно видеть, IP-адреса отправителя и получателя соответствуют адресам компьютеров отправителя и получателя.

```
→ ~ ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=128 time=3.83 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=128 time=3.97 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=128 time=33.0 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=128 time=55.2 ms
64 bytes from 192.168.1.101: icmp_seq=5 ttl=128 time=4.17 ms
64 bytes from 192.168.1.101: icmp_seq=6 ttl=128 time=4.58 ms
64 bytes from 192.168.1.101: icmp_seq=7 ttl=128 time=122 ms
64 bytes from 192.168.1.101: icmp_seq=8 ttl=128 time=43.0 ms
64 bytes from 192.168.1.101: icmp_seq=9 ttl=128 time=70.6 ms
64 bytes from 192.168.1.101: icmp_seq=10 ttl=128 time=4.43 ms
^C
--- 192.168.1.101 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 3.834/34.438/121.566/37.390 ms
```

Рисунок 12 – Создание ICMP-запроса с помощью утилиты ping



No.	Time	Source	Destination	Protocol	Length	Info
226	26.507887	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=1/256, ttl=64 (reply in 227)
227	26.508142	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=1/256, ttl=128 (request in 226)
230	27.510474	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=2/512, ttl=64 (reply in 231)
231	27.510766	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=2/512, ttl=128 (request in 230)
233	28.540097	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=3/768, ttl=64 (reply in 234)
234	28.540479	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=3/768, ttl=128 (request in 233)
236	29.563481	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=4/1024, ttl=64 (reply in 237)
237	29.563874	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=4/1024, ttl=128 (request in 236)
245	30.514510	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=5/1280, ttl=64 (reply in 246)
246	30.514868	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=5/1280, ttl=128 (request in 245)
250	31.516233	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=6/1536, ttl=64 (reply in 251)
251	31.516628	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=6/1536, ttl=128 (request in 250)
255	32.635336	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=7/1792, ttl=64 (reply in 256)
256	32.635792	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=7/1792, ttl=128 (request in 255)
258	33.557944	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=8/2048, ttl=64 (reply in 259)
259	33.558314	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=8/2048, ttl=128 (request in 258)
261	34.587283	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=9/2304, ttl=64 (reply in 262)
262	34.587463	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=9/2304, ttl=128 (request in 261)
266	35.522907	192.168.1.254	192.168.1.101	ICMP	98	Echo (ping) request id=0x0015, seq=10/2560, ttl=64 (reply in 267)
267	35.523278	192.168.1.101	192.168.1.254	ICMP	98	Echo (ping) reply id=0x0015, seq=10/2560, ttl=128 (request in 266)

Рисунок 13 – Перехваченные ICMP-пакеты

Убедимся, что MAC-адреса отправителя и получателя совпадают с адресами на компьютерах. Так как запрос осуществлялся с компьютера на операционной системе Linux, воспользуемся утилитой `ifconfig` для получения MAC-адреса (рис. 14). Для получения MAC-адреса на компьютере получателя получим с помощью утилиты `ipconfig`, так как лабораторная работа выполняется на операционной системе Windows (рис. 15).

```
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.254 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::6245:cbff:fe9a:b780 prefixlen 64 scopeid 0x20<link>
    ether 60:45:cb:9a:b7:80 txqueuelen 1000 (Ethernet)
    RX packets 8045408 bytes 1407233052 (1.4 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3373552 bytes 539995831 (539.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device memory 0xf6500000-f651ffff
```

Рисунок 14 – Информация о MAC-адресе отправителя

```
Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : 
Description . . . . . : Red Hat VirtIO Ethernet Adapter
Physical Address. . . . . : 7A-12-3E-A9-DF-A1
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::894a:cc2e:76da:4fa8%9(Preferred)
IPv4 Address. . . . . : 192.168.1.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Wednesday, November 15, 2023 9:09:21 PM
Lease Expires . . . . . : Wednesday, November 15, 2023 11:09:20 PM
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 242881086
DHCPv6 Client DUID. . . . . : 00-01-00-01-2C-E0-F6-6F-7A-12-3E-A9-DF-A1
DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled
```

Рисунок 15 – Информация о MAC-адресе получателя

Теперь проверим содержимое полученных пакетов. Откроем вкладку «Ethernet II» (рис. 16). Как видно, адреса совпадают с теми, что были получены ранее. Несмотря на то, что ICMP-запрос выполнялся по IP адресу, компьютеры смогли определить MAC-адреса друг друга. Это произошло с помощью протокола ARP, который позволяет устройствам узнать MAC-адрес получателя по IP-адресу.

```

> Frame 226: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
✓ Ethernet II, Src: ASUSTekC_9a:b7:80 (60:45:cb:9a:b7:80), Dst: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
  > Destination: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
  > Source: ASUSTekC_9a:b7:80 (60:45:cb:9a:b7:80)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.254, Dst: 192.168.1.101
> Internet Control Message Protocol

```

Рисунок 16 – Информация о MAC-адресах в ICMP-пакете

В качестве удаленных узлов (сайтов зарубежных СМИ) выберем следующие URL-адреса: insider.com, theguardian.com и nytimes.com. Выполним ICMP-запросы с помощью утилиты ping с включенным перехватом пакетов в Wireshark (рис. 17).

```

C:\Users\Daniil>ping insider.com

Pinging insider.com [151.101.2.217] with 32 bytes of data:
Reply from 151.101.2.217: bytes=32 time=23ms TTL=56
Reply from 151.101.2.217: bytes=32 time=29ms TTL=56
Reply from 151.101.2.217: bytes=32 time=29ms TTL=56
Reply from 151.101.2.217: bytes=32 time=24ms TTL=56

Ping statistics for 151.101.2.217:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 23ms, Maximum = 29ms, Average = 26ms

C:\Users\Daniil>ping theguardian.com

Pinging theguardian.com [151.101.193.111] with 32 bytes of data:
Reply from 151.101.193.111: bytes=32 time=24ms TTL=56
Reply from 151.101.193.111: bytes=32 time=24ms TTL=56
Reply from 151.101.193.111: bytes=32 time=24ms TTL=56
Reply from 151.101.193.111: bytes=32 time=29ms TTL=56

Ping statistics for 151.101.193.111:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 29ms, Average = 25ms

C:\Users\Daniil>ping nytimes.com

Pinging nytimes.com [151.101.193.164] with 32 bytes of data:
Reply from 151.101.193.164: bytes=32 time=23ms TTL=56
Reply from 151.101.193.164: bytes=32 time=24ms TTL=56
Reply from 151.101.193.164: bytes=32 time=29ms TTL=56
Reply from 151.101.193.164: bytes=32 time=24ms TTL=56

Ping statistics for 151.101.193.164:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 23ms, Maximum = 29ms, Average = 25ms

```

Рисунок 17 – ICMP-запросы на удаленные узлы

Как видно на рис. 17, удаленные узлы имеют следующие IP-адреса:

- **insider.com**: 151.101.2.217;
- **theguardian.com**: 151.101.193.111;
- **nytimes.com**: 151.101.193.164.

Проанализируем пакеты в Wireshark. Как видно на рис. 18-20, IP-адреса отправителя и получателя ICMP-запросов соответствуют ожидаемым.

```
▼ Ethernet II, Src: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1), Dst: TP-Linl
  > Destination: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
  > Source: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.1.101, Dst: 151.101.2.217
```

Рисунок 18 – Информация о ICMP-запросе на первый удаленный узел

```
▼ Ethernet II, Src: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1), Dst: TP-Link_
  > Destination: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
  > Source: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.1.101, Dst: 151.101.193.111
```

Рисунок 19 – Информация о ICMP-запросе на второй удаленный узел

```
▼ Ethernet II, Src: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1), Dst: TP-Link_
  > Destination: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
  > Source: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.1.101, Dst: 151.101.193.164
```

Рисунок 20 – Информация о ICMP-запросе на третий удаленный узел

Заметим, что MAC-адрес получателя всегда остается одним и тем же, несмотря на изменение адреса удаленного узла. Это происходит потому, что все отправляемые пакеты проходят через маршрутизатор, поскольку удаленные узлы находятся за пределами локальной сети. Поэтому все MAC-адреса получателя совпадают с MAC-адресом маршрутизатора (рис. 21).

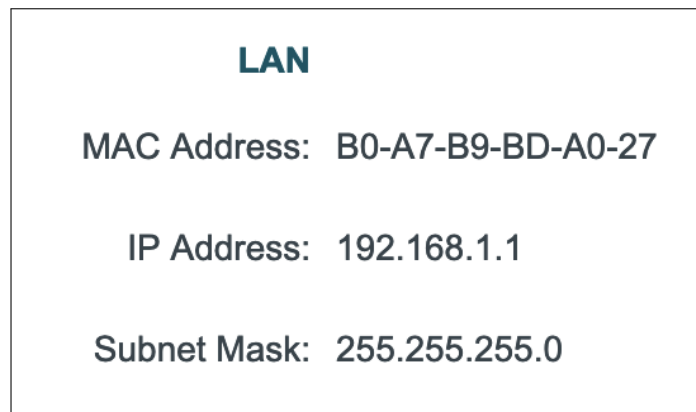


Рисунок 21 – Информация о MAC-адресе маршрутизатора

### 2.3. Анализ полей TCP

В качестве FTP-сервера, к которому будет произведено подключение, был выбран FTP-сервер Яндекса с адресом `ftp.yandex.ru`. Для того, чтобы лишние TCP-пакеты не попадали при перехвате пакетов в Wireshark, необходимо было указать адрес FTP-сервера. Для того, чтобы узнать адрес, использовалась утилита `nslookup` (рис. 22).

```
C:\Users\Daniil>nslookup ftp.yandex.ru
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:     mirror.yandex.ru
Addresses: 2a02:6b8::183
           213.180.204.183
Aliases:  ftp.yandex.ru
```

Рисунок 22 – Нахождение IP-адреса FTP-сервера с помощью утилиты `nslookup`

Для фильтрации перехватываемых пакетов использовался следующий фильтр:  
`tcp and ip.addr == 213.180.204.183`

В итоге были перехвачены пакеты, представленные на рис. 23.

No.	Time	Source	Destination	Protocol	Length	Info
10	2.649417	192.168.1.101	213.180.204.183	TCP	66	50421 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
11	2.677167	213.180.204.183	192.168.1.101	TCP	66	21 → 50421 [SYN, ACK] Seq=0 Ack=1 Win=42300 Len=0 MSS=1410 SACK_PERM=1 WS=4096
12	2.677394	192.168.1.101	213.180.204.183	TCP	54	50421 → 21 [ACK] Seq=1 Ack=1 Win=262144 Len=0
13	2.702755	213.180.204.183	192.168.1.101	FTP	140	Response: 220-Welcome to Yandex Mirror FTP service. Your served by: mirror01sas.mds.ya...
14	2.702755	213.180.204.183	192.168.1.101	FTP	60	Response: 220
15	2.702823	192.168.1.101	213.180.204.183	TCP	54	50421 → 21 [ACK] Seq=1 Ack=93 Win=262144 Len=0
16	2.703426	192.168.1.101	213.180.204.183	FTP	64	Request: AUTH TLS
17	2.724122	213.180.204.183	192.168.1.101	TCP	60	21 → 50421 [ACK] Seq=93 Ack=11 Win=45056 Len=0
18	2.724122	213.180.204.183	192.168.1.101	FTP	92	Response: 530 Please login with USER and PASS.
19	2.724327	192.168.1.101	213.180.204.183	FTP	64	Request: AUTH SSL
20	2.744851	213.180.204.183	192.168.1.101	TCP	60	21 → 50421 [ACK] Seq=131 Ack=21 Win=45056 Len=0
21	2.744851	213.180.204.183	192.168.1.101	FTP	92	Response: 530 Please login with USER and PASS.
22	2.749074	192.168.1.101	213.180.204.183	FTP	70	Request: USER anonymous
24	2.768978	213.180.204.183	192.168.1.101	TCP	60	21 → 50421 [ACK] Seq=169 Ack=37 Win=45056 Len=0
25	2.768978	213.180.204.183	192.168.1.101	FTP	88	Response: 331 Please specify the password.
26	2.769230	192.168.1.101	213.180.204.183	FTP	82	Request: PASS anonymous@example.com
27	2.795844	213.180.204.183	192.168.1.101	FTP	77	Response: 230 Login successful.
28	2.798169	192.168.1.101	213.180.204.183	FTP	59	Request: PWD
29	2.819220	213.180.204.183	192.168.1.101	FTP	88	Response: 257 "/" is the current directory
30	2.889283	192.168.1.101	213.180.204.183	TCP	54	50421 → 21 [ACK] Seq=70 Ack=260 Win=261888 Len=0

Рисунок 23 – Перехваченные пакеты при подключении к FTP-серверу

Рассмотрим TCP-пакеты поочередно. Первый перехваченный TCP-пакет (рис. 24), имеет поля, которые находятся в таблице 1.

```

> Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{784C0136-CC1E-4F8B-878A-EF797F1F2CAC}, id 0
▼ Ethernet II, Src: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1), Dst: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
  > Destination: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
  > Source: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.101, Dst: 213.180.204.183
▼ Transmission Control Protocol, Src Port: 50421, Dst Port: 21, Seq: 0, Len: 0
  Source Port: 50421
  Destination Port: 21
  [Stream index: 0]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 3519252155
  [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 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): 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
  .... .... ..0 = Fin: Not set
  [TCP Flags: .....S.]
  Window: 64240
  [Calculated window size: 64240]
  Checksum: 0xf21b [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]

```

Рисунок 24 – Первый перехваченный TCP-пакет

Таблица 1 – Поля первого ТСП-пакета

Название поля	Значение поля
IP-адрес источника	192.168.1.101
IP-адрес назначения	213.180.204.183
Номер порта источника	50421
Номер порта назначения	21
Порядковый номер	0
Номер подтверждения	0
Длина заголовка	32
Размер окна	64240

Поля второго ТСП-пакета (рис. 25) находятся в таблице 2.

```
> Frame 11: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{784C0136-CC1E-4F8B-878A-EF797F1F2CAC}, id 0
▼ Ethernet II, Src: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27), Dst: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
  > Destination: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
  > Source: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 213.180.204.183, Dst: 192.168.1.101
▼ Transmission Control Protocol, Src Port: 21, Dst Port: 50421, Seq: 0, Ack: 1, Len: 0
  Source Port: 21
  Destination Port: 50421
  [Stream index: 0]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 3378510284
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 3519252156
  1000 .... = Header Length: 32 bytes (8)
▼ Flags: 0x012 (SYN, ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  ....0... = Congestion Window Reduced (CWR): Not set
  ....0... = ECN-Echo: Not set
  ....0... = Urgent: Not set
  ....1... = Acknowledgment: Set
  ....0... = Push: Not set
  ....0... = Reset: Not set
  > ....1... = Syn: Set
  ....0... = Fin: Not set
  [TCP Flags: .....A..S.]
  Window: 42300
  [Calculated window size: 42300]
  Checksum: 0x84c0 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  > [Timestamps]
```

Рисунок 25 – Второй перехваченный ТСП-пакет



Таблица 2 – Поля второго ТСР-пакета

Название поля	Значение поля
IP-адрес источника	213.180.204.183
IP-адрес назначения	192.168.1.101
Номер порта источника	21
Номер порта назначения	50421
Порядковый номер	0
Номер подтверждения	1
Длина заголовка	32
Размер окна	42300

Поля третьего ТСР-пакета (рис. 26) находятся в таблице 3.

```
> Frame 12: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{784C0136-CC1E-4F8B-878A-EF797F1F2CAC}, id 0
▼ Ethernet II, Src: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1), Dst: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
  > Destination: TP-Link_bd:a0:27 (b0:a7:b9:bd:a0:27)
  > Source: 7a:12:3e:a9:df:a1 (7a:12:3e:a9:df:a1)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.101, Dst: 213.180.204.183
▼ Transmission Control Protocol, Src Port: 50421, Dst Port: 21, Seq: 1, Ack: 1, Len: 0
  Source Port: 50421
  Destination Port: 21
  [Stream index: 0]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3519252156
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 3378510285
  0101 .... = Header Length: 20 bytes (5)
▼ Flags: 0x010 (ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  ....0... = Congestion Window Reduced (CWR): Not set
  ....0... = ECN-Echo: Not set
  ....0... = Urgent: Not set
  ....1... = Acknowledgment: Set
  ....0... = Push: Not set
  ....0... = Reset: Not set
  ....0... = Syn: Not set
  ....0... = Fin: Not set
  [TCP Flags: .....A....]
  Window: 1024
  [Calculated window size: 262144]
  [Window size scaling factor: 256]
  Checksum: 0x66a2 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
> [Timestamps]
```

Рисунок 26 – Третий перехваченный ТСР-пакет

Таблица 3 – Поля третьего ТСР-пакета

Название поля	Значение поля
IP-адрес источника	192.168.1.101
IP-адрес назначения	213.180.204.183
Номер порта источника	50421
Номер порта назначения	21
Порядковый номер	1
Номер подтверждения	1
Длина заголовка	20
Размер окна	1024

С помощью следующего фильтра

ftp and ip.addr == 213.180.204.183

можно получить только те пакеты, что относятся к протоколу FTP (рис. 27).

No.	Time	Source	Destination	Protocol	Length	Info
8	3.368625	213.180.204.183	192.168.1.101	FTP	140	Response: 220-Welcome to Yandex Mirror FTP service. Your served by: mirror01vla.mds.ya...
9	3.368625	213.180.204.183	192.168.1.101	FTP	60	Response: 220
11	3.369123	192.168.1.101	213.180.204.183	FTP	64	Request: AUTH TLS
13	3.387330	213.180.204.183	192.168.1.101	FTP	92	Response: 530 Please login with USER and PASS.
14	3.387605	192.168.1.101	213.180.204.183	FTP	64	Request: AUTH SSL
16	3.405094	213.180.204.183	192.168.1.101	FTP	92	Response: 530 Please login with USER and PASS.
17	3.423440	192.168.1.101	213.180.204.183	FTP	70	Request: USER anonymous
19	3.441874	213.180.204.183	192.168.1.101	FTP	88	Response: 331 Please specify the password.
20	3.442095	192.168.1.101	213.180.204.183	FTP	82	Request: PASS anonymous@example.com
21	3.464585	213.180.204.183	192.168.1.101	FTP	77	Response: 230 Login successful.
22	3.466972	192.168.1.101	213.180.204.183	FTP	59	Request: PWD
23	3.484451	213.180.204.183	192.168.1.101	FTP	88	Response: 257 "/" is the current directory
24	3.486066	192.168.1.101	213.180.204.183	FTP	62	Request: TYPE I
25	3.503653	213.180.204.183	192.168.1.101	FTP	85	Response: 200 Switching to Binary mode.
26	3.503921	192.168.1.101	213.180.204.183	FTP	60	Request: PASV
27	3.521721	213.180.204.183	192.168.1.101	FTP	108	Response: 227 Entering Passive Mode (213,180,204,183,210,250).
28	3.522053	192.168.1.101	213.180.204.183	FTP	60	Request: LIST
32	3.560862	213.180.204.183	192.168.1.101	FTP	93	Response: 150 Here comes the directory listing.
40	3.578670	213.180.204.183	192.168.1.101	FTP	78	Response: 226 Directory send OK.

Рисунок 27 – Перехваченные FTP-пакеты

### **3. Заключение**

В ходе выполнения данной лабораторной работы я разобрался со стеком TCP/IP, анализируя пакеты, которые отправляются и принимаются с помощью данного стека, научился собирать сетевой трафик с помощью программы Wireshark, научился фильтровать собранный трафик, находить и просматривать соединения.