

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего  
образования



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

Сети и телекоммуникации

Отчет

по лабораторной работе №4

ARP протокол

ПРОВЕРИЛ:

\_\_\_\_\_

Гай В.Е.

СТУДЕНТ:

\_\_\_\_\_

Козменкова Е.П.  
18 В-2

Нижний Новгород

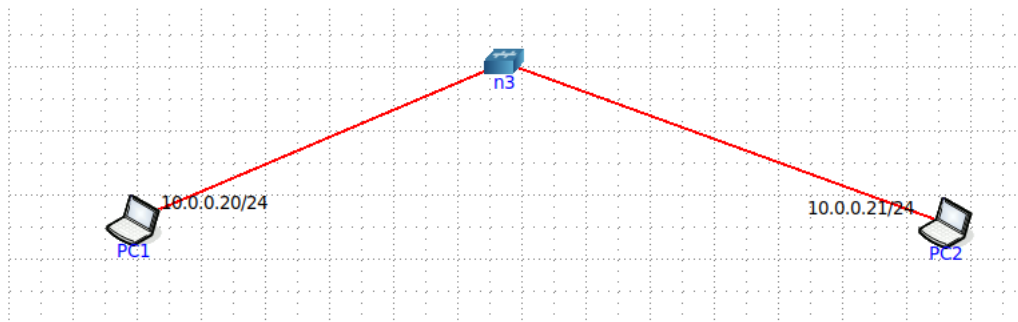
2020 г.

### Задание:

1. Создать сеть в Core, состоящую из двух компьютеров.
2. Запустить UDP сервер на одном из компьютеров. Подключиться к UDP серверу с помощью TCP клиента. Объяснить полученные пакеты в WireShark
3. Запустить TCP клиент, сервер, передать данные, затем прервать соединение (Ctrl + C) на стороне сервера. Объяснить полученные пакеты в WireShark
4. Запустить TCP клиент, сервер, передать данные, затем прервать соединение (Ctrl + C) на стороне клиента. Объяснить полученные пакеты в WireShark
5. Запустить UDP сервер на одном из компьютеров. Подключиться к UDP серверу с помощью UDP клиента. Передать данные. Объяснить полученные пакеты в WireShark

### Ход работы:

1. Создать сеть в Core, состоящую из двух компьютеров.



2. Запустить UDP сервер на одном из компьютеров. Подключиться к UDP серверу с помощью TCP клиента. Объяснить полученные пакеты в WireShark

### UDP сервер (10.0.0.20)

```
nc -u -l 2399
```

### TCP клиент

```
nc 10.0.0.20 2399
```

### Результат:

```
Терминал
Файл Правка Вид Поиск Терминал Справка
root@PC1:/tmp/pycore.45613/PC1.conf# nc -u -l 2399
[ ]

Терминал
Файл Правка Вид Поиск Терминал Справка
root@PC2:/tmp/pycore.45613/PC2.conf# nc 10.0.0.20 2399
root@PC2:/tmp/pycore.45613/PC2.conf#
root@PC2:/tmp/pycore.45613/PC2.conf# nc 10.0.0.20 2399
root@PC2:/tmp/pycore.45613/PC2.conf# [ ]
```

Ничего не произошло. Проверю, что будет в WireShark.

PC1:

1	0.00000000	10.0.0.21	10.0.0.20	TCP	74	47626 → 2399 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
2	0.000010786	10.0.0.20	10.0.0.21	TCP	54	2399 → 47626 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	5.125334304	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	Who has 10.0.0.21? Tell 10.0.0.20
4	5.125355379	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	Who has 10.0.0.20? Tell 10.0.0.21
5	5.125364873	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	10.0.0.21 is at 00:00:00_aa:00:01
6	5.125363901	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	10.0.0.20 is at 00:00:00_aa:00:00

Transmission Control Protocol, Src Port: 47626, Dst Port: 2399, Seq: 0, Len: 0

Source Port: 47626

Destination Port: 2399

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

[Next sequence number: 0 (relative sequence number)]

Acknowledgment number: 0

1010 .... = Header Length: 40 bytes (10)

▶ Flags: 0x002 (SYN)

Window size value: 64240

[Calculated window size: 64240]

Checksum: 0xbf10 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

▼ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

▶ TCP Option - Maximum segment size: 1460 bytes

▶ TCP Option - SACK permitted

▶ TCP Option - Timestamps: TSval 2037749371, TSecr 0

▶ TCP Option - No-Operation (NOP)

▶ TCP Option - Window scale: 7 (multiply by 128)

▼ [Timestamps]

[Time since first frame in this TCP stream: 0.000000000 seconds]

[Time since previous frame in this TCP stream: 0.000000000 seconds]

Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.0.21

Transmission Control Protocol, Src Port: 2399, Dst Port: 47626, Seq: 1, Ack: 1, Len: 0

Source Port: 2399

Destination Port: 47626

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)

[Next sequence number: 1 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

0101 .... = Header Length: 20 bytes (5)

▶ Flags: 0x014 (RST, ACK)

Window size value: 0

[Calculated window size: 0]

[Window size scaling factor: -1 (unknown)]

Checksum: 0x31c2 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

▼ [SEQ/ACK analysis]

[This is an ACK to the segment in frame: 1]

[The RTT to ACK the segment was: 0.000010786 seconds]

[rRTT: 0.000010786 seconds]

▼ [Timestamps]

[Time since first frame in this TCP stream: 0.000010786 seconds]

[Time since previous frame in this TCP stream: 0.000010786 seconds]

PC2:

2	27.898125419	10.0.0.21	10.0.0.20	TCP	74	47626 → 2399 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_P
3	27.898145942	10.0.0.20	10.0.0.21	TCP	54	2399 → 47626 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	33.023486619	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	Who has 10.0.0.21? Tell 10.0.0.20
5	33.023477304	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	Who has 10.0.0.20? Tell 10.0.0.21
6	33.023494876	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	10.0.0.21 is at 00:00:00_aa:00:01
7	33.023499500	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	10.0.0.20 is at 00:00:00_aa:00:00

Transmission Control Protocol, Src Port: 47626, Dst Port: 2399, Seq: 0, Len: 0

Source Port: 47626

Destination Port: 2399

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

[Next sequence number: 0 (relative sequence number)]

Acknowledgment number: 0

1010 .... = Header Length: 40 bytes (10)

▶ Flags: 0x002 (SYN)

Window size value: 64240

[Calculated window size: 64240]

Checksum: 0xbf10 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

▶ [Timestamps]

```
Transmission Control Protocol, Src Port: 2399, Dst Port: 47626, Seq: 1, Ack: 1, Len: 0
  Source Port: 2399
  Destination Port: 47626
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 1 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x014 (RST, ACK)
    Window size value: 0
    [Calculated window size: 0]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x31c2 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
```

В ответе сервера виден флаг RST:

```
0101 .... = Header Length: 20
  ▸ Flags: 0x014 (RST, ACK)
    Window size value: 0
```

Это значит, сервер не смог создать сокет для обслуживания нового клиента (связано это с тем, что мы подключаемся по TCP протоколу к UDP серверу).

3. Запустить TCP клиент, сервер, передать данные, затем прервать соединение (Ctrl + C) на стороне сервера. Объяснить полученные пакеты в WireShark

**TCP сервер (10.0.0.20)**

nc -l 2399

**TCP клиент**

nc 10.0.0.20 2399

Результат:

```
root@PC1:/tmp/pycore.45613/PC1.conf# nc -l 2399
123
[Ctrl+C]

root@PC2:/tmp/pycore.45613/PC2.conf# nc 10.0.0.20 2399
123
[Ctrl+C]
```

PC1:

9	37.402400357	10.0.0.200.11.1.1	10.0.0.20	ICMPv6	70 Router Solicitation from 00:00:00:aa:00:01
10	41.276474575	00:00:00_aa:00:01	Broadcast	ARP	42 Who has 10.0.0.20? Tell 10.0.0.21
11	41.276489481	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42 10.0.0.20 is at 00:00:00:aa:00:00
12	41.276506674	10.0.0.21	10.0.0.20	TCP	74 47720 → 2399 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
13	41.276516181	10.0.0.20	10.0.0.21	TCP	74 2399 → 47720 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460
14	41.276527215	10.0.0.21	10.0.0.20	TCP	66 47720 → 2399 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=203924
15	46.362439358	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42 Who has 10.0.0.21? Tell 10.0.0.20
16	46.362482460	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42 10.0.0.21 is at 00:00:00:aa:00:01
17	51.137084535	10.0.0.21	10.0.0.20	TCP	70 47720 → 2399 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=4 TSval=203924
18	51.137098526	10.0.0.20	10.0.0.21	TCP	66 2399 → 47720 [ACK] Seq=1 Ack=5 Win=65280 Len=0 TSval=286572
19	63.362747743	10.0.0.20	10.0.0.21	TCP	70 2399 → 47720 [PSH, ACK] Seq=1 Ack=5 Win=65280 Len=4 TSval=286572
20	63.362772071	10.0.0.21	10.0.0.20	TCP	66 47720 → 2399 [ACK] Seq=5 Ack=5 Win=64256 Len=0 TSval=203924

Первые две строки – ARP-запрос клиента и ARP-ответ сервера. Далее идет запрос с флагом SYN от клиента к серверу:

```

Transmission Control Protocol, Src Port: 47720, Dst Port: 2399, Seq: 0, Len: 0
  Source Port: 47720
  Destination Port: 2399
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  [Next sequence number: 0 (relative sequence number)]
  Acknowledgment number: 0
  1010 .... = Header Length: 40 bytes (10)
  ▸ Flags: 0x002 (SYN)
  Window size value: 64240
  [Calculated window size: 64240]
  Checksum: 0x1b47 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▸ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  ▸ [Timestamps]

```

После этого – ответ сервера с флагами SYN, ACK:

```

Transmission Control Protocol, Src Port: 2399, Dst Port: 47720, Seq: 0, Ack: 1, Len: 0
  Source Port: 2399
  Destination Port: 47720
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  [Next sequence number: 0 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  1010 .... = Header Length: 40 bytes (10)
  ▸ Flags: 0x012 (SYN, ACK)
  Window size value: 65160
  [Calculated window size: 65160]
  Checksum: 0x386b [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▸ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]

```

И, наконец, ответ клиента с флагом ACK – соединение установлено.

```

Transmission Control Protocol, Src Port: 47720, Dst Port: 2399, Seq: 1, Ack: 1, Len: 0
  Source Port: 47720
  Destination Port: 2399
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 1 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
  Window size value: 502
  [Calculated window size: 64256]
  [Window size scaling factor: 128]
  Checksum: 0x63ca [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]

```

После этого начинается передача данных. От клиента к серверу:

```

Transmission Control Protocol, Src Port: 47720, Dst Port: 2399, Seq: 1, Ack: 1, Len: 4
  Source Port: 47720
  Destination Port: 2399
  [Stream index: 0]
  [TCP Segment Len: 4]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 5 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x018 (PSH, ACK)
  Window size value: 502
  [Calculated window size: 64256]
  [Window size scaling factor: 128]
  Checksum: 0xd8fd [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
  TCP payload (4 bytes)
  ▾ Data (4 bytes)
    Data: 3132330a
    [Length: 4]

```

Флаги – ACK и PSH, в разделе Data видно сами передаваемые данные (последовательность символов ‘123’). И ответ сервера о полученных данных:

```

Transmission Control Protocol, Src Port: 2399, Dst Port: 47720, Seq: 1, Ack: 5, Len: 0
Source Port: 2399
Destination Port: 47720
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 5 (relative ack number)
1000 ... = Header Length: 32 bytes (8)
► Flags: 0x010 (ACK)
Window size value: 510
[Calculated window size: 65280]
[Window size scaling factor: 128]
Checksum: 0x16b6 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
► Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
► [SEQ/ACK analysis]
► [Timestamps]

```

Аналогично работает передача данных от сервера клиенту.

Первым соединением на стороне сервера.

29.875840144	10.0.0.20	10.0.0.21	TCP	66 2399 → 47732 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
29.876870521	10.0.0.21	10.0.0.20	TCP	66 47732 → 2399 [ACK] Seq=1 Ack=2 Win=64256 Len=0 TSval=

```

Transmission Control Protocol, Src Port: 2399, Dst Port: 47720, Seq: 5, Ack: 5, Len: 0
Source Port: 2399
Destination Port: 47720
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 5 (relative sequence number)
[Next sequence number: 5 (relative sequence number)]
Acknowledgment number: 5 (relative ack number)
1000 ... = Header Length: 32 bytes (8)
► Flags: 0x011 (FIN, ACK)
Window size value: 510
[Calculated window size: 65280]
[Window size scaling factor: 128]
Checksum: 0x97fe [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
► Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
► [Timestamps]

```

Сервер отправляет клиенту сегмент с флагами FIN и ACK. Далее от клиента приходит сегмент с флагом ACK.

```

Transmission Control Protocol, Src Port: 47720, Dst Port: 2399, Seq: 5, Ack: 6, Len: 0
Source Port: 47720
Destination Port: 2399
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 5 (relative sequence number)
[Next sequence number: 5 (relative sequence number)]
Acknowledgment number: 6 (relative ack number)
1000 ... = Header Length: 32 bytes (8)
► Flags: 0x010 (ACK)
Window size value: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x78ae [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
► Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
► [SEQ/ACK analysis]
► [Timestamps]

```

Но сервер продолжает чтение и ждет от клиента сегмент с флагом FIN. Первую соединение и со стороны клиента:

12 37.430753748	10.0.0.21	10.0.0.20	TCP	66 47732 → 2399 [FIN, ACK] Seq=1 Ack=2 Win=64256 Len=0
13 37.430765851	10.0.0.20	10.0.0.21	TCP	66 2399 → 47732 [ACK] Seq=2 Ack=2 Win=65280 Len=0 TSval=

Аналогичная ситуация.

```

Transmission Control Protocol, Src Port: 47732, Dst Port: 2399, Seq: 1, Ack: 2, Len: 0
  Source Port: 47732
  Destination Port: 2399
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 1 (relative sequence number)]
  Acknowledgment number: 2 (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x011 (FIN, ACK)
  Window size value: 502
  [Calculated window size: 64256]
  [Window size scaling factor: 128]
  Checksum: 0xd804 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [Timestamps]

Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.0.21
Transmission Control Protocol, Src Port: 2399, Dst Port: 47732, Seq: 2, Ack: 2, Len: 0
  Source Port: 2399
  Destination Port: 47732
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 2 (relative sequence number)
  [Next sequence number: 2 (relative sequence number)]
  Acknowledgment number: 2 (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x010 (ACK)
  Window size value: 510
  [Calculated window size: 65280]
  [Window size scaling factor: 128]
  Checksum: 0xba79 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]

```

4. Запустить UDP сервер на одном из компьютеров. Подключиться к UDP серверу с помощью UDP клиента. Передать данные. Объяснить полученные пакеты в Wireshark

### UDP сервер (10.0.0.20)

nc -u -l 2399

### UDP клиент

nc -u 10.0.0.20 2399

Если мы установим соединение и попробуем сразу передать данные с сервера на клиент, ничего не произойдет:

```

Файл Правка Вид Поиск Терминал Справка
root@PC1:/tmp/pycore.45613/PC1.conf# nc -u -l 2399
111
□

Терминал
Файл Правка Вид Поиск Терминал Справка
root@PC2:/tmp/pycore.45613/PC2.conf# nc -u 10.0.0.20 2399

```

Однако, если передадим данные с клиента, они дойдут (и дойдут те самые данные с сервера):

```

Файл Правка Вид Поиск Терминал Справка
root@PC1:/tmp/pycore.45613/PC1.conf# nc -u -l 2399
111
222
□

Терминал
Файл Правка Вид Поиск Терминал Справка
root@PC2:/tmp/pycore.45613/PC2.conf# nc -u 10.0.0.20 2399
222
111

```

Посмотрим, что происходит в Wireshark:



13	98.996925170	00:00:00_aa:00:01	Broadcast	ARP	42	Who has 10.0.0.20? Tell 10.0.0.21
14	98.996952464	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	10.0.0.20 is at 00:00:00_aa:00:00
15	98.996955166	10.0.0.21	10.0.0.20	UDP	46	41928 → 2399 Len=4
16	98.997015011	10.0.0.20	10.0.0.21	UDP	46	2399 → 41928 Len=4
17	104.191489997	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	Who has 10.0.0.21? Tell 10.0.0.20
18	104.191511019	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	10.0.0.21 is at 00:00:00_aa:00:01

### Пакеты, переданные по протоколу UDP:

▶ Internet Protocol Version 4, Src: 10.0.0.21, Dst: 10.0.0.20 ▼ User Datagram Protocol, Src Port: 41928, Dst Port: 2399 Source Port: 41928 Destination Port: 2399 Length: 12 Checksum: 0xda49 [unverified] [Checksum Status: Unverified] [Stream index: 2] ▼ Data (4 bytes) Data: 3232320a [Length: 4]	▶ Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.0.21 ▼ User Datagram Protocol, Src Port: 2399, Dst Port: 41928 Source Port: 2399 Destination Port: 41928 Length: 12 Checksum: 0xdc4a [unverified] [Checksum Status: Unverified] [Stream index: 2] ▼ Data (4 bytes) Data: 3131310a [Length: 4]
--	--

Если прервать соединение, никаких изменений в Wireshark мы не увидим:

19	188.415539242	fe80::200:ff:feaa:0	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00_aa:00:00
20	190.464003059	fe80::44a:d9ff:fe60...	ff02::2	ICMPv6	70	Router Solicitation from 06:4a:d9:60:07:f4
21	198.348322064	fe80::40ed:cdff:fe9...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM"
22	199.308454692	fe80::44a:d9ff:fe60...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM"
23	231.423520743	fe80::40ed:cdff:fe9...	ff02::2	ICMPv6	70	Router Solicitation from 06:4a:d9:60:07:f4
24	231.423498838	fe80::200:ff:feaa:1	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00_aa:00:01

Если же запустить netcat заново и попробовать сразу передавать данные с клиента, никаких задержек и подвисаний не будет, все пройдет правильно и в задуманном порядке:

```

root@PC1:/tmp/pycore.45613/PC1.conf# nc -u -l 2399
111
222
^C
root@PC1:/tmp/pycore.45613/PC1.conf# nc -u -l 2399
123
111
222
[ ]

Терминал
Файл Правка Вид Поиск Терминал Справка
root@PC2:/tmp/pycore.45613/PC2.conf# nc -u 10.0.0.20 2399
222
111
^C
root@PC2:/tmp/pycore.45613/PC2.conf# nc -u 10.0.0.20 2399
123
111
222
[ ]

```

25	424.258955954	10.0.0.21	10.0.0.20	UDP	46	42956 → 2399 Len=4
26	429.312419148	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	Who has 10.0.0.20? Tell 10.0.0.21
27	429.312470103	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	10.0.0.20 is at 00:00:00_aa:00:00
28	431.540600813	10.0.0.21	10.0.0.20	UDP	46	42956 → 2399 Len=4
29	434.054302383	10.0.0.20	10.0.0.21	UDP	46	2399 → 42956 Len=4
30	439.296435920	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	Who has 10.0.0.21? Tell 10.0.0.20
31	439.296459700	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	10.0.0.21 is at 00:00:00_aa:00:01