

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

ОТЧЕТ

по лабораторной работе №3

по дисциплине

Сети и телекоммуникации

РУКОВОДИТЕЛЬ:

(подпись)

Гай В.Е.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Береснева М.А.
(фамилия, и.,о.)

18-АС
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород 2020

Задание

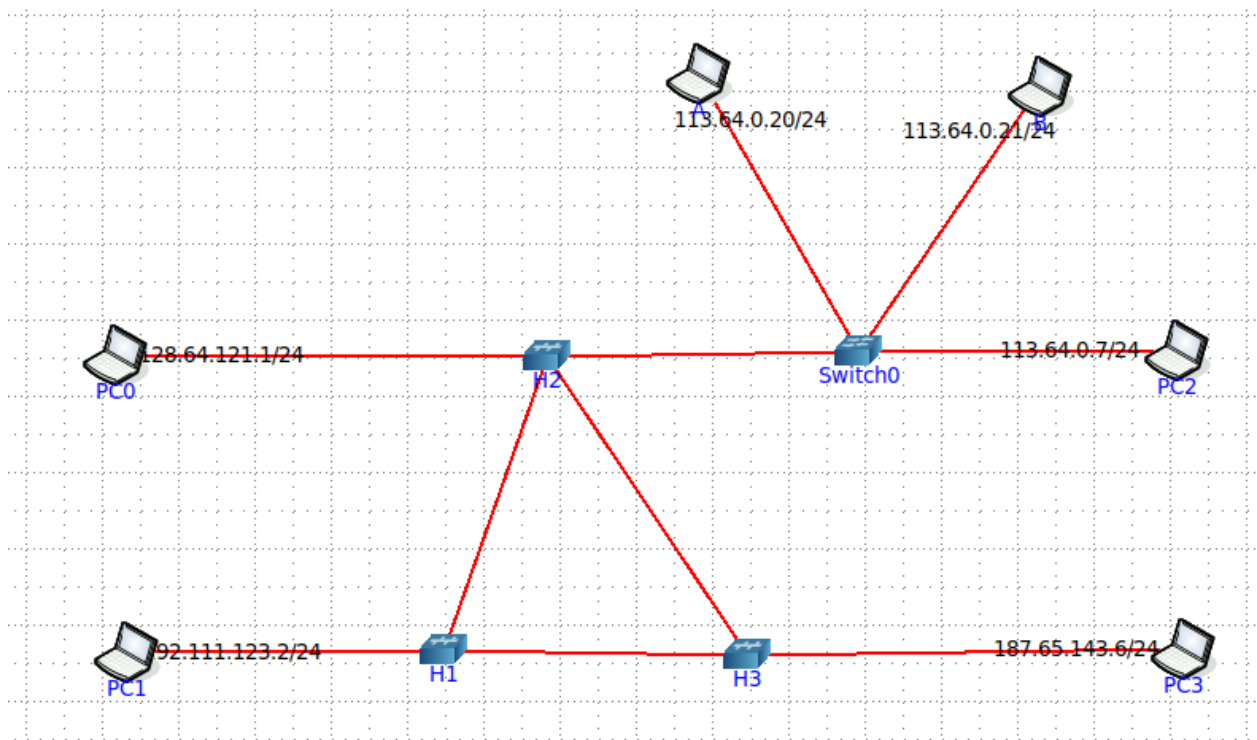
Для экспериментов использовать схему из первой лабораторной работы. Все ip-адреса (или маски) необходимо поменять так, чтобы адрес сети у всех компьютеров был один. Все действия должны быть выполнены в симуляторе сетей CORE.

Часть 1. Формирование запроса и получение ответа

1. Начать захват пакетов при помощи WireShark.
2. Сформировать кадр ARP-запроса с помощью утилиты PackETH и отправить его в сеть (компьютеры выбрать самостоятельно).
3. Убедиться, что был получен кадр ARP-ответа, соответствующий посланному запросу. Захваченные пакеты сохранить для отчета. Вывести arp таблицу (команда «arp»).
4. Прекратить захват пакетов.

Часть 2. ARP-спуфинг

1. Выделить на схеме и обозначить три компьютера: А, В, Сервер.
2. Подготовить кадр ARP-ответа, направляемый Сервером хосту А с помощью программы PackETH. Кадр должен быть составлен так, чтобы MAC-адресу Сервера соответствовал IP-адрес хоста В. Вывести arp таблицу на хосте А. Отправить сформированный пакет от Сервера хосту А.



К серверу могут подключаться любые компьютеры

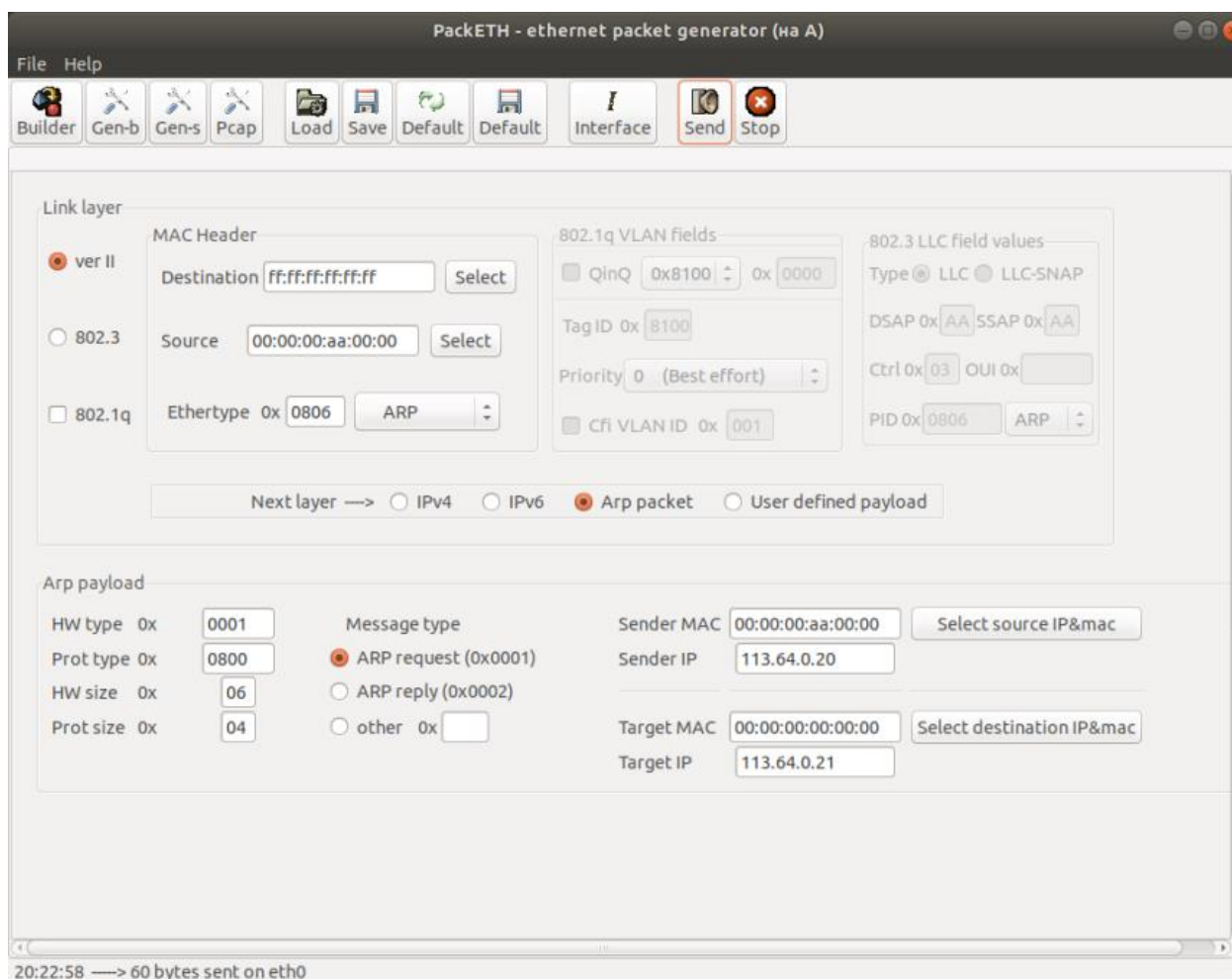
```

core@core-VirtualBox: ~
Файл Правка Вид Поиск Терминал Вкладки Справка
core@core-VirtualBox: ~
core@core-VirtualBox:~$ xhost +
access control disabled, clients can connect from any host
core@core-VirtualBox:~$
  
```

На компьютере A запускаем PackETH

```

Терминал
Файл Правка Вид Поиск Терминал Справка
root@A:/tmp/pycore.43593/A.conf# DISPLAY=:0 packeth
Gtk-Message: 13:50:26.172: Failed to load module "canberra-gtk-module"
g_thread NOT supported
  
```



Destination – широковещательный MAC-адрес

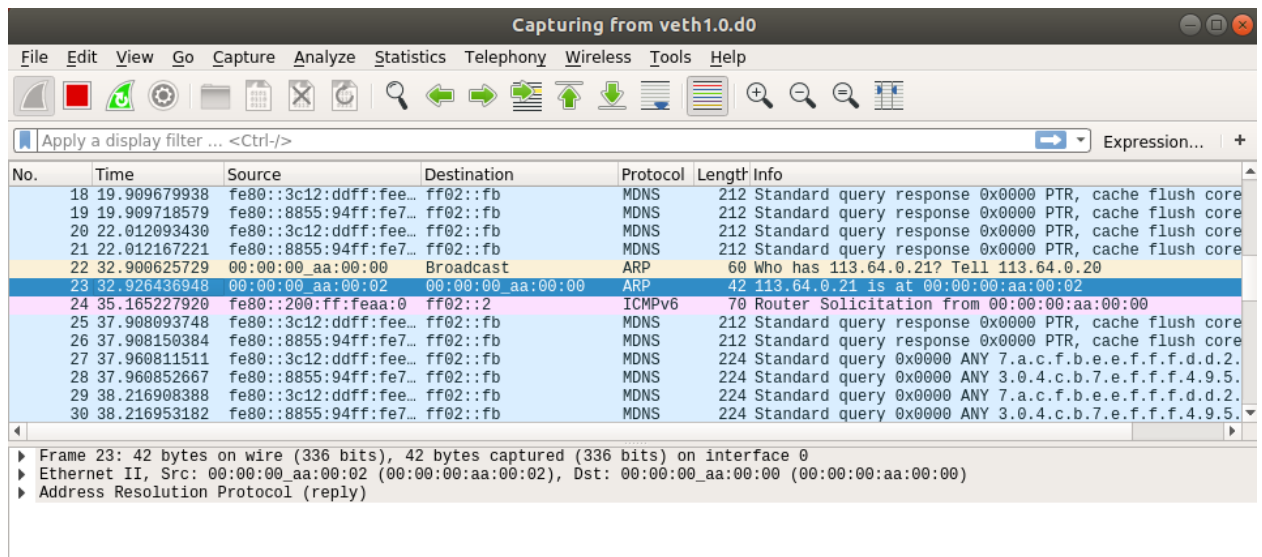
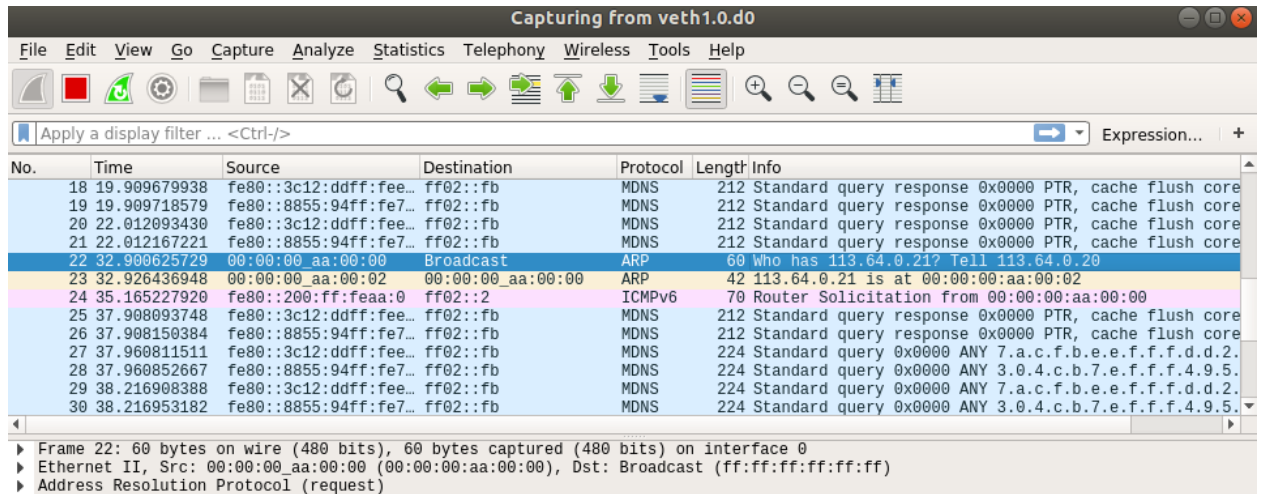
Source – MAC-адрес компьютера А (с помощью ifconfig)

Sender MAC – MAC-адрес компьютера А

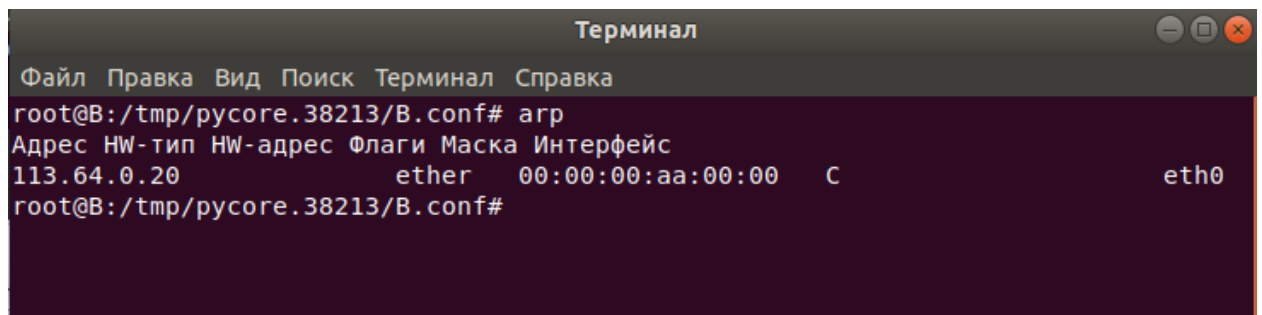
Sender IP – IP компьютера А

Target IP - IP компьютера В

Запускаем Wireshark и видим, что компьютер А посылает запрос и получает ответ.

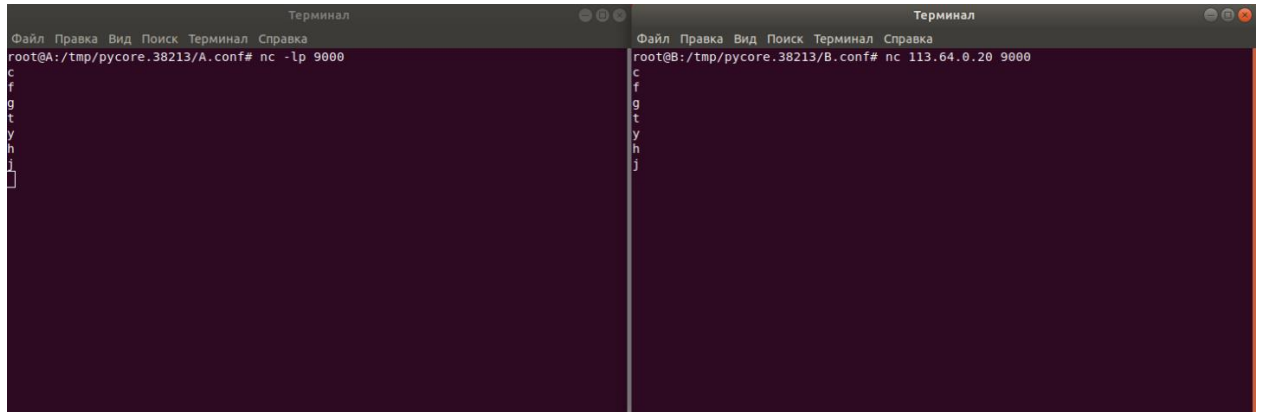


Смотрим arp-таблицу на В и видим, что в ней находится MAC-адрес компьютера А.



В консоли компьютера А "nc -lp 9000".

В консоли компьютера В "nc 113.64.0.20 9000"



The image shows a Wireshark packet capture window titled 'Capturing from veth1.0.d0'. The interface includes a menu bar, a toolbar, and a packet list table. The packet list table shows the following data:

No.	Time	Source	Destination	Protocol	Length	Info
267	323.122433528	113.64.0.21	113.64.0.20	TCP	74	36332 → 9000 [SYN, Seq=0 Win=64240 Len=0 MSS=1460 SA
268	323.124209499	113.64.0.20	113.64.0.21	TCP	74	9000 → 36332 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0
269	323.134143575	113.64.0.21	113.64.0.20	TCP	66	36332 → 9000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval
270	324.406341156	113.64.0.21	113.64.0.20	TCP	68	36332 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=2
271	324.407714030	113.64.0.20	113.64.0.21	TCP	66	9000 → 36332 [ACK] Seq=1 Ack=3 Win=65280 Len=0 TSval
272	328.286657283	00:00:00_aa:00:02	00:00:00_aa:00:00	ARP	42	Who has 113.64.0.20? Tell 113.64.0.21
273	328.288168843	00:00:00_aa:00:00	00:00:00_aa:00:02	ARP	42	113.64.0.20 is at 00:00:00_aa:00:00
274	329.023215205	113.64.0.21	113.64.0.20	TCP	68	36332 → 9000 [PSH, ACK] Seq=3 Ack=1 Win=64256 Len=2
275	329.024583697	113.64.0.20	113.64.0.21	TCP	66	9000 → 36332 [ACK] Seq=1 Ack=5 Win=65280 Len=0 TSval
276	329.295605406	113.64.0.21	113.64.0.20	TCP	68	36332 → 9000 [PSH, ACK] Seq=5 Ack=1 Win=64256 Len=2
277	329.296978158	113.64.0.20	113.64.0.21	TCP	66	9000 → 36332 [ACK] Seq=7 Ack=1 Win=65280 Len=0 TSval
278	329.546229446	113.64.0.21	113.64.0.20	TCP	68	36332 → 9000 [PSH, ACK] Seq=7 Ack=1 Win=64256 Len=2
279	329.547704179	113.64.0.20	113.64.0.21	TCP	66	9000 → 36332 [ACK] Seq=1 Ack=9 Win=65280 Len=0 TSval
280	329.809633325	113.64.0.21	113.64.0.20	TCP	68	36332 → 9000 [PSH, ACK] Seq=9 Ack=1 Win=64256 Len=2
281	329.811234817	113.64.0.20	113.64.0.21	TCP	66	9000 → 36332 [ACK] Seq=1 Ack=11 Win=65280 Len=0 TSval
282	330.040073077	113.64.0.21	113.64.0.20	TCP	68	36332 → 9000 [PSH, ACK] Seq=11 Ack=1 Win=64256 Len=2
283	330.043347480	113.64.0.20	113.64.0.21	TCP	66	9000 → 36332 [ACK] Seq=1 Ack=13 Win=65280 Len=0 TSval
284	332.924011559	113.64.0.21	113.64.0.20	TCP	68	36332 → 9000 [PSH, ACK] Seq=13 Ack=1 Win=64256 Len=2
285	332.925627936	113.64.0.20	113.64.0.21	TCP	66	9000 → 36332 [ACK] Seq=1 Ack=15 Win=65280 Len=0 TSval

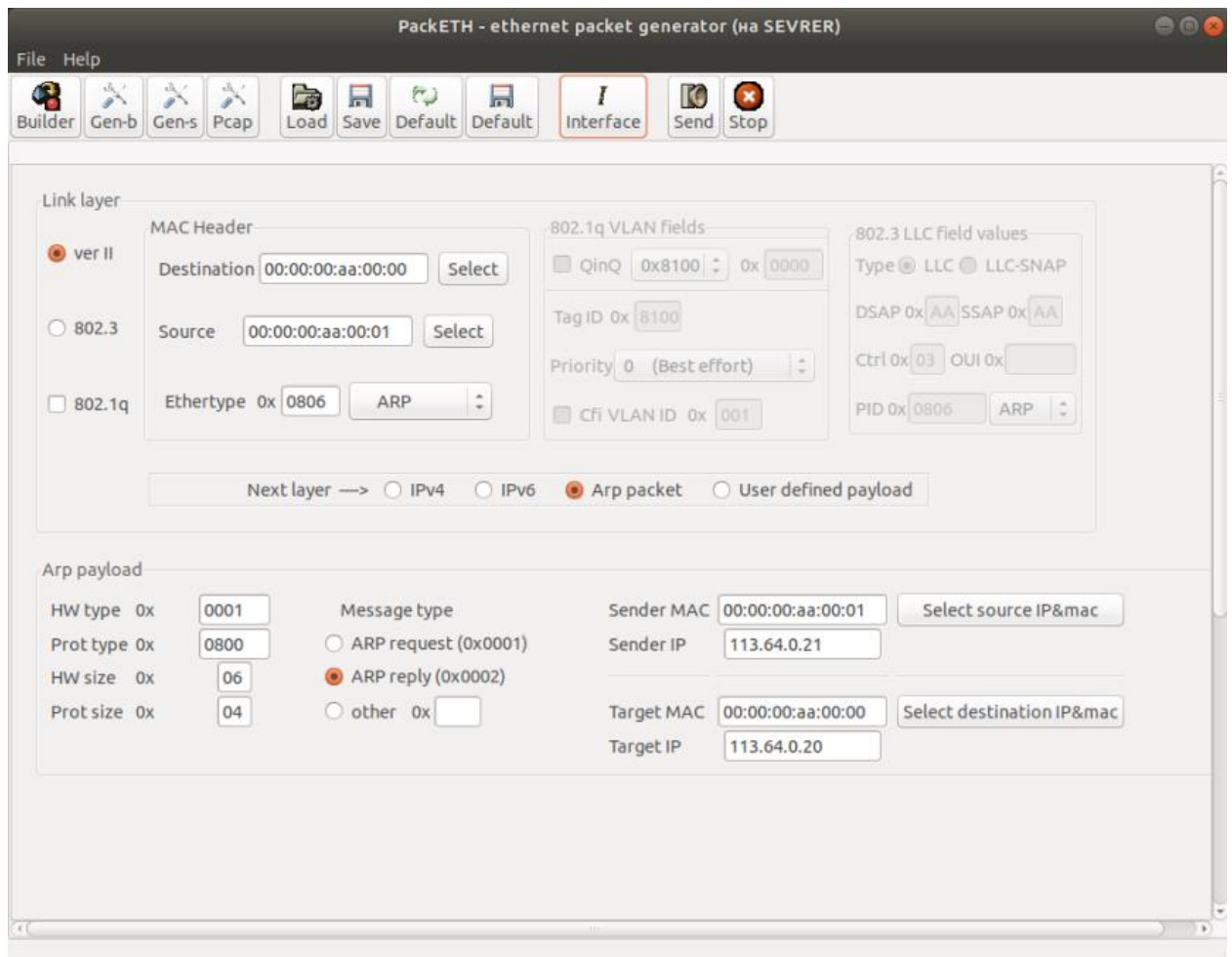
Below the table, there is a status bar showing details for Frame 23: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0. The details pane shows Ethernet II, Src: 00:00:00_aa:00:02 (00:00:00_aa:00:02), Dst: 00:00:00_aa:00:00 (00:00:00_aa:00:00), and Address Resolution Protocol (reply).

Сервер пакеты не видит.

The image shows a Wireshark packet capture window titled 'Capturing from veth3.0.d0'. The interface includes a menu bar, a toolbar, and a packet list table. The packet list table shows the following data:

No.	Time	Source	Destination	Protocol	Length	Info
20	36.81998536	fe80::8855:94ff:fe7::fb	ff02::fb	MDNS	224	Standard query 0x0000 ANY 3.0.4.c.b.7.e.f.f.f.4.9.5.
21	36.272560123	fe80::908c:ccff:fe7::fb	ff02::fb	MDNS	224	Standard query 0x0000 ANY 7.6.1.3.0.7.e.f.f.f.4.9.5.
22	36.272654717	fe80::8855:94ff:fe7::fb	ff02::fb	MDNS	224	Standard query 0x0000 ANY 3.0.4.c.b.7.e.f.f.f.4.9.5.
23	36.522296193	fe80::908c:ccff:fe7::fb	ff02::fb	MDNS	224	Standard query 0x0000 ANY 3.0.4.c.b.7.e.f.f.f.4.9.5.
24	36.522366598	fe80::8855:94ff:fe7::fb	ff02::fb	MDNS	224	Standard query response 0x0000 PTR, cache flush core
25	36.722367127	fe80::908c:ccff:fe7::fb	ff02::fb	MDNS	212	Standard query response 0x0000 PTR, cache flush core
26	36.722439841	fe80::8855:94ff:fe7::fb	ff02::fb	MDNS	212	Standard query response 0x0000 PTR, cache flush core
27	37.732445222	fe80::908c:ccff:fe7::fb	ff02::fb	MDNS	212	Standard query response 0x0000 PTR, cache flush core
28	37.732521549	fe80::8855:94ff:fe7::fb	ff02::fb	MDNS	212	Standard query response 0x0000 PTR, cache flush core
29	39.752077109	fe80::908c:ccff:fe7::fb	ff02::fb	MDNS	212	Standard query response 0x0000 PTR, cache flush core
30	39.752154797	fe80::8855:94ff:fe7::fb	ff02::fb	MDNS	212	Standard query response 0x0000 PTR, cache flush core

Below the table, there is a status bar showing details for Frame 1: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface 0. The details pane shows Ethernet II, Src: 92:8c:cc:70:31:07 (92:8c:cc:70:31:07), Dst: IPv6cast, Fb (33:33:00:00:00:fb), Internet Protocol Version 6, Src: fe80::908c:ccff:fe70:3167, Dst: ff02::fb, User Datagram Protocol, Src Port: 5353, Dst Port: 5353, and Multicast Domain Name System (response).



Destination – MAC-адрес компьютера A

Source – MAC-адрес сервера

Sender MAC – MAC-адрес сервера

Sender IP – IP компьютера B

Target MAC - MAC-адрес компьютера A

Target IP - IP компьютера A

Видим, что в arp-таблице компьютера А сохранился MAC-адрес сервера

```
Терминал
Файл Правка Вид Поиск Терминал Справка
root@A:/tmp/pycore.38213/A.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 113.64.0.20 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 fe80::200:ff:feaa:0 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:00:00 txqueuelen 1000 (Ethernet)
    RX packets 720 bytes 141463 (141.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 41 bytes 2638 (2.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@A:/tmp/pycore.38213/A.conf# arp
Адрес HW-тип HW-адрес Флаги Маска Интерфейс
113.64.0.21 ether 00:00:00:aa:00:01 C eth0
root@A:/tmp/pycore.38213/A.conf#
```

Отправляем пакет от А к В.

The screenshot shows two terminal windows and a packet capture window. The left terminal (Host A) shows the command `nc -lp 9000` being executed. The right terminal (Host B) shows the command `nc 113.64.0.20 9000`. The packet capture window, titled "Capturing from veth3.0.d0", displays a list of captured packets. The selected packet (No. 548) is a DNS Standard query response from 113.64.0.21 to 113.64.0.20. The packet details pane shows the following information:

- Frame 1: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface 0
- Ethernet II, Src: 92:8c:cc:70:31:67 (92:8c:cc:70:31:67), Dst: IPv6mcast_fb (33:33:00:00:00:fb)
- Internet Protocol Version 6, Src: fe80::908c:ccff:fe70:3167, Dst: ff02::fb
- User Datagram Protocol, Src Port: 5353, Dst Port: 5353


```
Терминал
Файл Правка Вид Поиск Терминал Справка
root@B:/tmp/pycore.38835/B.conf# nc 113.64.0.20 9000
a
[ ]

Терминал
Файл Правка Вид Поиск Терминал Справка
root@A:/tmp/pycore.38835/A.conf# nc -lp 9000
a
[ ]
```

Capturing from veth3.0.24

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter <Ctrl-F> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	113.64.0.20	113.64.0.21	TCP	68	9000 → 33494 [PSH, ACK] Seq=1 Ack=1 Win=510 Len=2 TSval=
2	0.207380818	113.64.0.20	113.64.0.21	TCP	68	TCP Retransmission 9000 → 33494 [PSH, ACK] Seq=1 Ack=
3	0.414841431	113.64.0.20	113.64.0.21	TCP	68	TCP Retransmission 9000 → 33494 [PSH, ACK] Seq=1 Ack=
4	0.827138075	113.64.0.20	113.64.0.21	TCP	68	TCP Retransmission 9000 → 33494 [PSH, ACK] Seq=1 Ack=
5	1.658794349	113.64.0.20	113.64.0.21	TCP	68	TCP Retransmission 9000 → 33494 [PSH, ACK] Seq=1 Ack=
6	3.323638844	113.64.0.20	113.64.0.21	TCP	68	TCP Retransmission 9000 → 33494 [PSH, ACK] Seq=1 Ack=
7	5.062852694	00:00:00_aa:00:00	00:00:00_aa:00:00	ARP	42	Who has 113.64.0.21? Tell 113.64.0.20
8	6.106908139	00:00:00_aa:00:00	00:00:00_aa:00:00	ARP	42	Who has 113.64.0.21? Tell 113.64.0.20
9	6.618929147	113.64.0.20	113.64.0.21	TCP	68	TCP Retransmission 9000 → 33494 [PSH, ACK] Seq=1 Ack=
10	7.131483334	00:00:00_aa:00:00	00:00:00_aa:00:00	ARP	42	Who has 113.64.0.21? Tell 113.64.0.20
11	13.275156356	00:00:00_aa:00:00	Broadcast	ARP	42	Who has 113.64.0.21? Tell 113.64.0.20