

Минобрнауки России
Федеральное государственное бюджетное образовательное
учреждение высшего образования
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. Р.Е. АЛЕКСЕЕВА
ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

Курс “Сети и телекоммуникации”
Отчет по лабораторной работе №4

Выполнил: студент группы 18 В-1

Грачев Д.С.

Проверил: Гай В.Е.

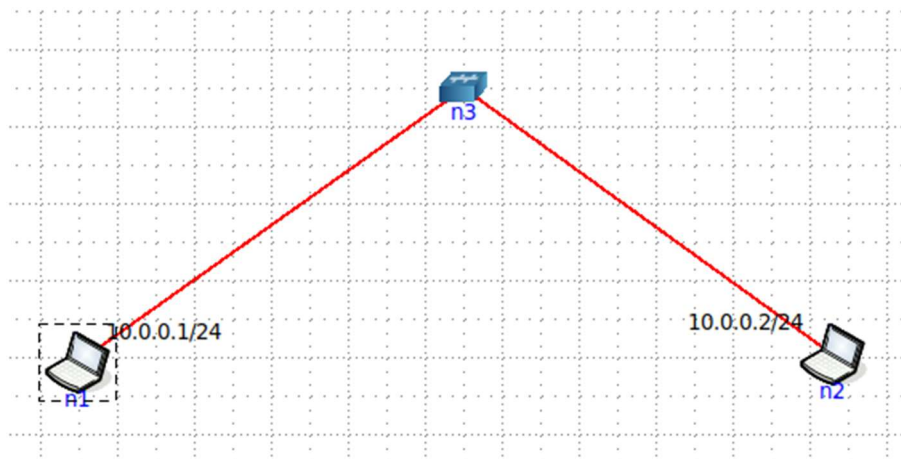
Нижний Новгород 2020

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:

Построили следующую сеть:



Запускаем UDP-сервер и пытаемся подключиться с помощью TCP-клиента:

The screenshot displays a network simulation environment with two terminal windows and a Wireshark packet capture interface.

Terminal 1 (Left): Shows the configuration of a UDP server on node n1. The command executed is `nc -u -l 90`.

Terminal 2 (Right): Shows the configuration of a TCP client on node n2. The command executed is `nc 10.0.0.1 90`.

Wireshark (Right): Shows a packet capture from interface 0. The packet list displays several DNS queries and a TCP SYN packet from 10.0.0.2 to 10.0.0.1. The packet details pane shows the structure of the captured packets, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.

Соединение не установлено, т.к. используются разные протоколы и клиент не может получить ответ от сервера.

Запускаем TCP сервер и клиент, разрываем соединение со стороны сервера

The image displays three sequential screenshots of a terminal window and Wireshark packet capture, illustrating a TCP connection test where the server initiates a reset.

Terminal Session:

```
core
Терминал
Файл Правка Вид Поиск Терминал Справка
root@n1:/tmp/pycore.37455/n1.conf# nc -u -l 90
^C
root@n1:/tmp/pycore.37455/n1.conf# nc -l 90

Терминал
Файл Правка Вид Поиск Терминал Справка
root@n1:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90

Терминал
Файл Правка Вид Поиск Терминал Справка
root@n1:/tmp/pycore.37455/n1.conf# nc -u -l 90
^C
root@n1:/tmp/pycore.37455/n1.conf# nc -l 90
Hi!
Hi!

Терминал
Файл Правка Вид Поиск Терминал Справка
root@n1:/tmp/pycore.37455/n1.conf# nc -u -l 90
^C
root@n1:/tmp/pycore.37455/n1.conf# nc -l 90
Hi!
Hi!

Терминал
Файл Правка Вид Поиск Терминал Справка
root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
Hi!
Hi!

Терминал
Файл Правка Вид Поиск Терминал Справка
root@n1:/tmp/pycore.37455/n1.conf# nc -u -l 90
^C
root@n1:/tmp/pycore.37455/n1.conf# nc -l 90
Hi!
Hi!

Терминал
Файл Правка Вид Поиск Терминал Справка
root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
Hi!
Hi!
```

Wireshark Packet Capture:

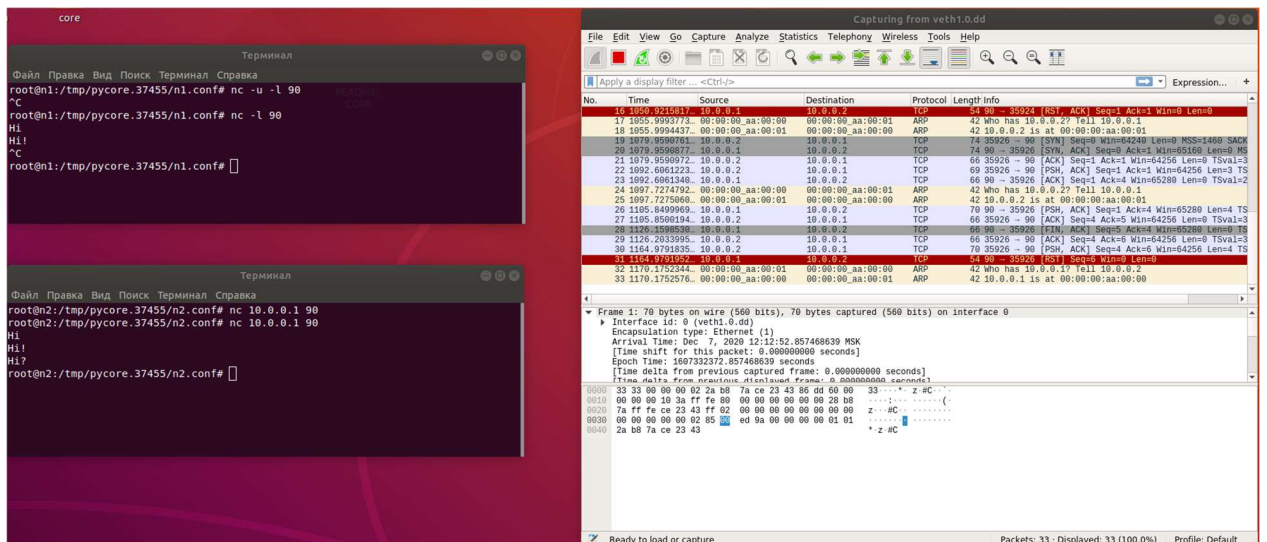
The Wireshark interface shows the capture of network traffic on interface 0 (veth1.0.0d). The packet list displays the following packets:

No.	Time	Source	Destination	Protocol	Length	Info
10	10.576.168612689	fe80::cfff:fe5:	ff02::fb	RDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" que
11	10.576.638213290	fe80::2088:7a7f:fc:	ff02::fb	RDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" que
12	10.688.128109927	fe80::2088:7a7f:fc:	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:01
13	10.959.9215572	00:00:00:aa:00:01	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
14	10.959.9215714	00:00:00:aa:00:00	00:00:00:aa:00:01	ARP	42	10.0.0.1 is at 00:00:00:aa:00:00
15	10.959.9215763	10.0.0.2	10.0.0.1	TCP	74	35924 → 35924 [SYN] Seq=8 Min=64240 Len=0 MSS=1460 SACK
16	10.959.9215811	10.0.0.1	10.0.0.2	TCP	74	35926 → 35926 [ACK] Seq=8 Min=64240 Len=0 MSS=1460 SACK
17	10.955.9993773	00:00:00:aa:00:00	00:00:00:aa:00:01	ARP	42	Who has 10.0.0.2? Tell 10.0.0.1
18	10.955.9994437	00:00:00:aa:00:01	00:00:00:aa:00:00	ARP	42	10.0.0.2 is at 00:00:00:aa:00:01
19	10.979.9598761	10.0.0.2	10.0.0.1	TCP	74	35926 → 35926 [SYN] Seq=8 Min=64240 Len=0 MSS=1460 SACK
20	10.979.9598777	10.0.0.1	10.0.0.2	TCP	74	35926 → 35926 [SYN] Seq=8 Min=64240 Len=0 MSS=1460 SACK
21	10.979.9598772	10.0.0.2	10.0.0.1	TCP	60	35926 → 35926 [ACK] Seq=1 Ack=1 Min=64256 Len=0 TSval=3
22	10.992.6861223	10.0.0.2	10.0.0.1	TCP	60	35926 → 35926 [ACK] Seq=1 Ack=1 Min=64256 Len=0 TSval=3
23	10.992.6861340	10.0.0.2	10.0.0.1	TCP	60	35926 → 35926 [ACK] Seq=1 Ack=1 Min=64256 Len=0 TSval=3
24	10.997.7274792	00:00:00:aa:00:00	00:00:00:aa:00:01	ARP	42	Who has 10.0.0.2? Tell 10.0.0.1
25	10.997.7275060	00:00:00:aa:00:01	00:00:00:aa:00:00	ARP	42	10.0.0.2 is at 00:00:00:aa:00:01
26	11.005.8499669	10.0.0.1	10.0.0.2	TCP	70	35926 → 35926 [PSH, ACK] Seq=1 Ack=4 Min=65280 Len=0 TSval=3
27	11.005.8500194	10.0.0.2	10.0.0.1	TCP	60	35926 → 35926 [ACK] Seq=4 Ack=5 Min=64256 Len=0 TSval=3
28	11.005.8500194	10.0.0.2	10.0.0.1	TCP	60	35926 → 35926 [ACK] Seq=4 Ack=5 Min=64256 Len=0 TSval=3
29	11.005.8500194	10.0.0.2	10.0.0.1	TCP	60	35926 → 35926 [ACK] Seq=4 Ack=5 Min=64256 Len=0 TSval=3

The packet details pane shows the following information for the selected packet (No. 29):

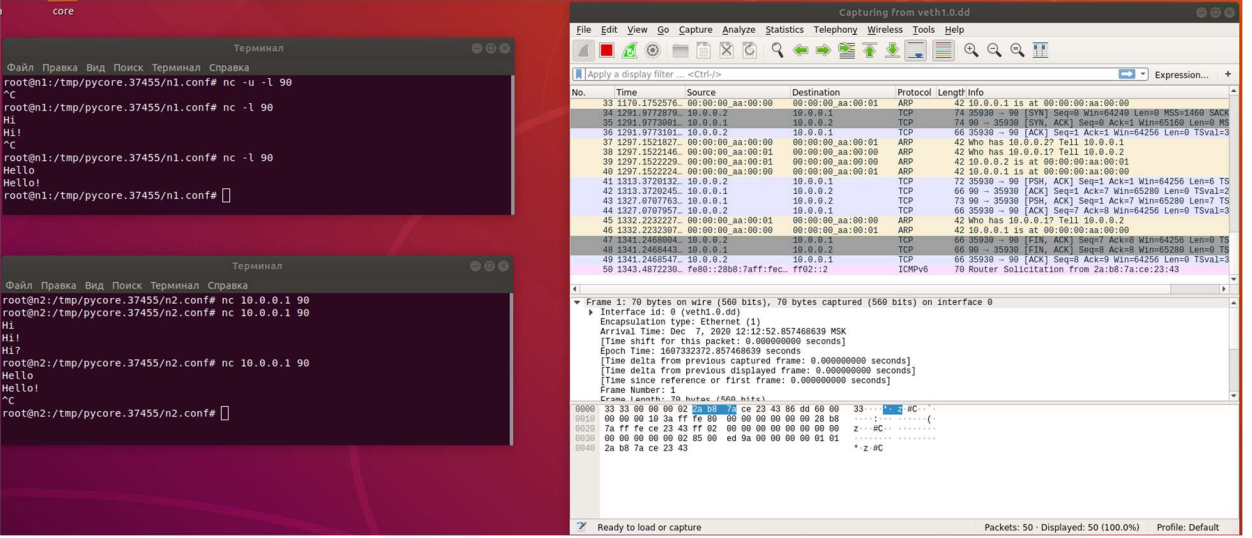
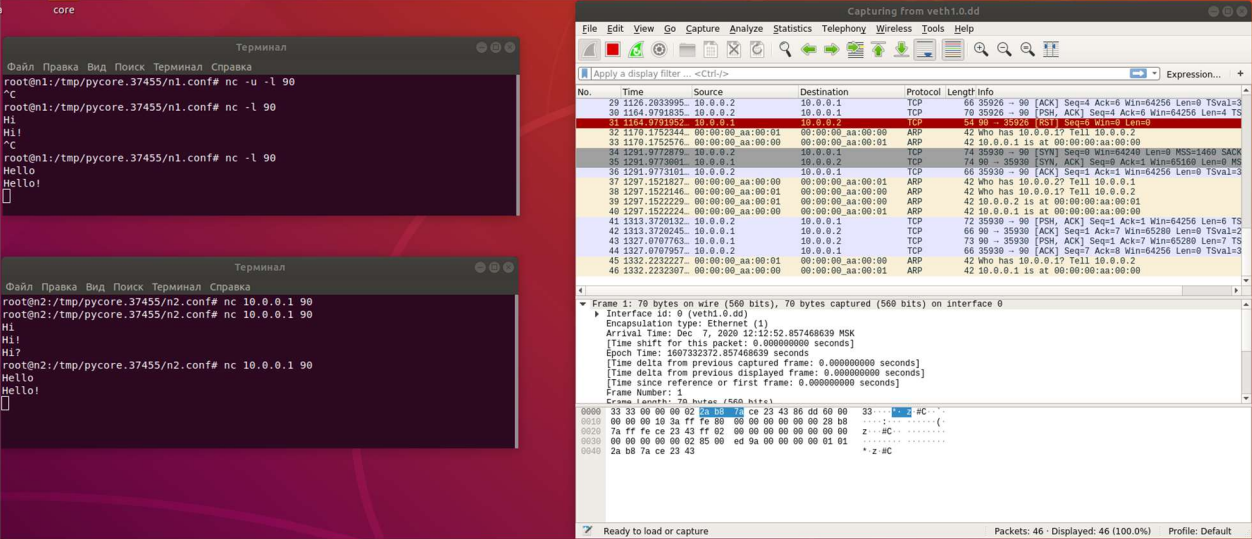
- Interface id: 0 (veth1.0.0d)
- Encapsulation type: Ethernet (1)
- Arrival Time: Dec 7, 2020 12:12:52.857468639 MSK
- Time shift for this packet: 0.000000000 seconds
- Epoch Time: 1607332372.857468639 seconds
- Time delta from previous captured frame: 0.000000000 seconds
- Time delta from previous displayed frame: 0.000000000 seconds
- Time since reference or first frame: 0.000000000 seconds
- Frame Number: 1
- Frame Length: 70 bytes (560 bits)
- Capture Length: 70 bytes (560 bits)
- [Frame is marked: False]
- [Frame is ignored: False]

The packet bytes pane shows the raw data of the packet, including the Ethernet II header, Internet Protocol Version 4 header, and the TCP segment.



Соединение устанавливается в три пакета. Для каждого сообщения отправляются два пакета: само сообщение и подтверждение получения. Разрыв соединения со стороны сервера провоцирует ошибку на стороне клиента (он не закрывает соединение).

Запускаем TCP сервер и клиент, разрываем соединение со стороны клиента



Соединение устанавливается в три пакета. Для каждого сообщения отправляются два пакета: само сообщение и подтверждение получения. Разрыв соединения со стороны клиента не провоцирует ошибок.


```
core

Терминал

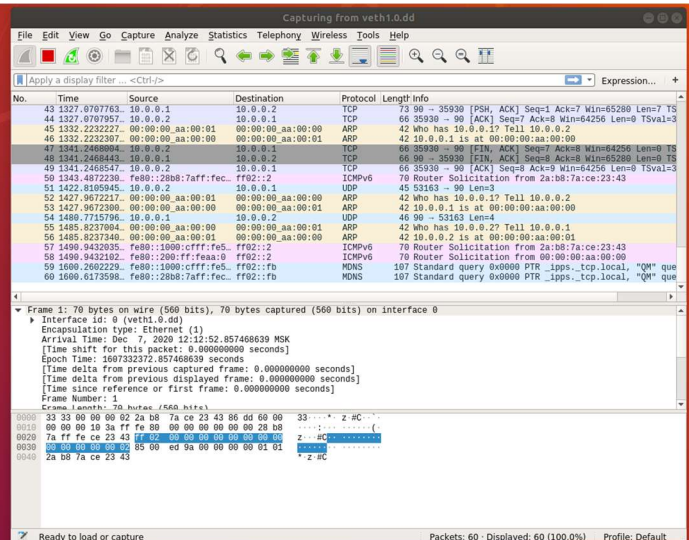
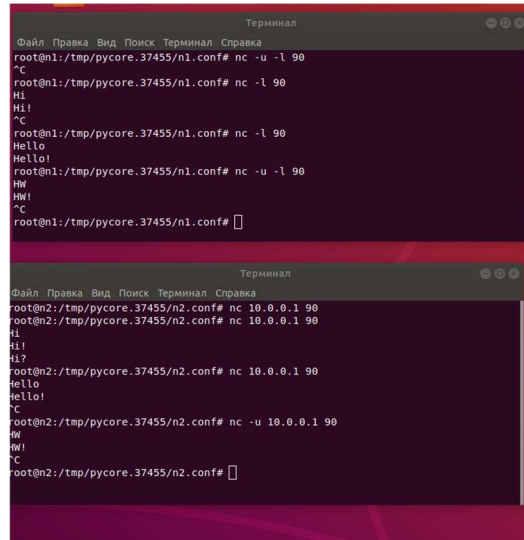
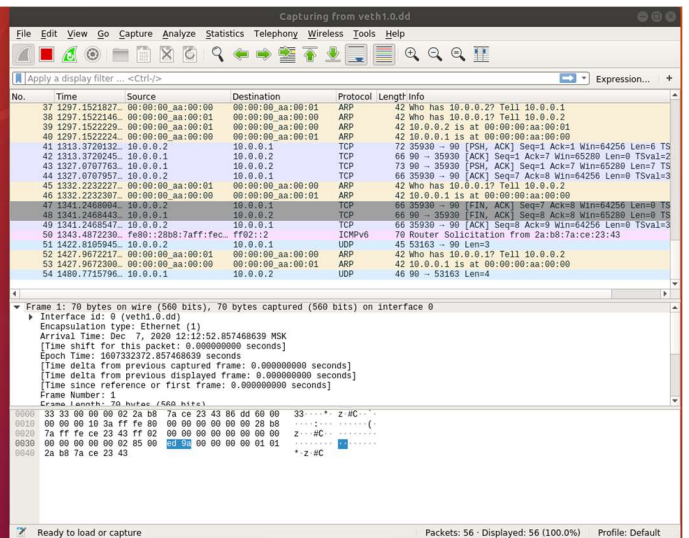
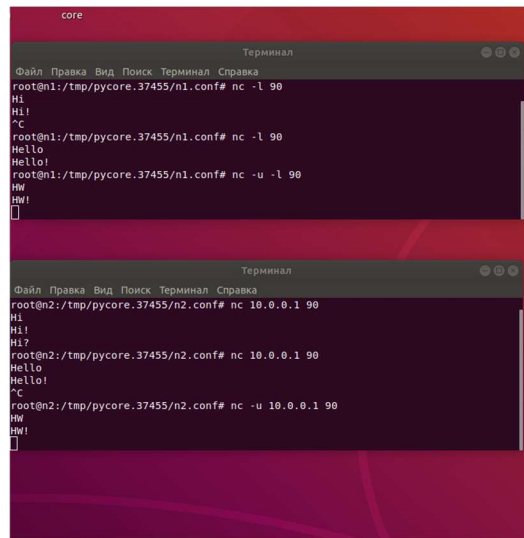
Файл Правка Вид Поиск Терминал Справка

^C
root@n1:/tmp/pycore.37455/n1.conf# nc -l 90
Hi!
Hi!!
^C
root@n1:/tmp/pycore.37455/n1.conf# nc -l 90
Hello
Hello!
root@n1:/tmp/pycore.37455/n1.conf# nc -u -l 90
HW
[]

Терминал

Файл Правка Вид Поиск Терминал Справка

root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
Hi!
Hi!!
root@n2:/tmp/pycore.37455/n2.conf# nc 10.0.0.1 90
Hello
Hello!
^C
root@n2:/tmp/pycore.37455/n2.conf# nc -u 10.0.0.1 90
HW
[]
```



Разрыв соединения происходит независимо на стороне сервера и клиента.