

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное государственное учреждение высшего образования

Санкт-петербургский национальный исследовательский институт информационных
технологий, механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа №3

**Мониторинг сетевого трафика на хосте на примере работы с утилитами
диагностики и мониторинга сетевых соединений в Linux**
По дисциплине «Телекоммуникационные системы и технологии»

Выполнил студент группы № М33091:

Максимов Лев

Сидорцов Владимир

Мирзабеков Ренат

Цыденов Алексей

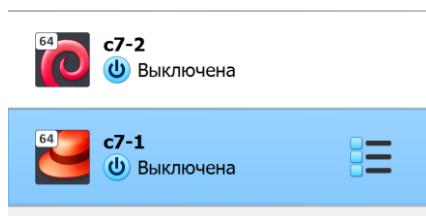
Проверил:

Шараева Кристина Витальевна

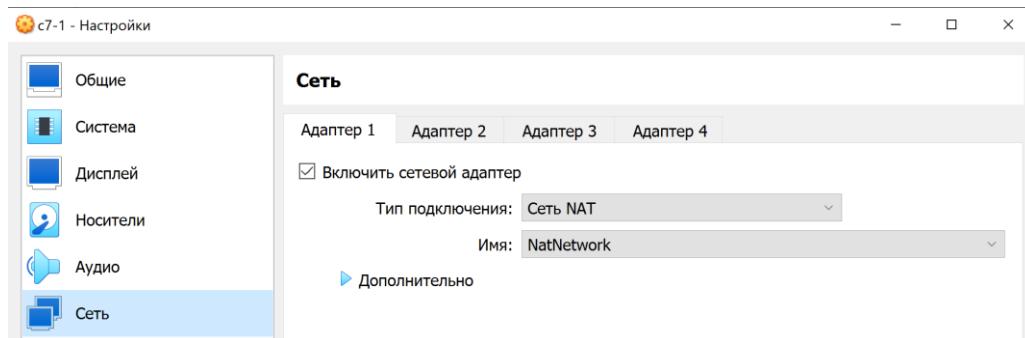
САНКТ-ПЕТЕРБУРГ

Часть 1. Настройка инфраструктуры

- Подготовьте две виртуальные машины.
- Одну машину назовите c7-1, другой c7-2. Одна из машин должна работать на CentOS7, вторая на Debian 11.



- На обоих машинах сетевые интерфейсы настройте в режим Сеть NAT, а внутри машин получение адресов - автоматически.



- Определите полученные адреса для машин c7-1 и c7-2.

c7-1:

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
[root@localhost lab1]# ifconfig
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.100.0.3 netmask 255.255.255.0 broadcast 10.100.0.255
                inet6 fe80::5481:64%0 netmask 64 scopeid 0x20<link>
                    ether 08:00:27:6d:c7:8b txqueuelen 1000 (Ethernet)
                    RX packets 0 bytes 0 (0.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 13 bytes 938 (938.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.0.0.0 broadcast 10.255.255.255
                ether 08:00:27:6d:c7:8b txqueuelen 1000 (Ethernet)
                    RX packets 18 bytes 3284 (3.2 KIB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 24 bytes 2216 (2.1 KIB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.0.4 netmask 255.0.0.0 broadcast 10.255.255.255
                inet6 fe80::5760:700%cb42:3072 netmask 64 scopeid 0x20<link>
                    ether 08:00:27:38:a9:72 txqueuelen 1000 (Ethernet)
                    RX packets 78 bytes 9556 (9.3 KIB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 65 bytes 6212 (6.0 KiB)
                    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 (Local Loopback)
                    RX packets 5 bytes 736 (736.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 5 bytes 736 (736.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost lab1]#
```

с7-2:

```
[root@debian11: "# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:79:40:cd brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.5/8 brd 10.255.255.255 scope global dynamic enp0s3
        valid_lft 569sec preferred_lft 569sec
    inet 10.0.0.6/8 brd 10.255.255.255 scope global secondary dynamic enp0s3
        valid_lft 576sec preferred_lft 576sec
    inet6 fe80::a00:27ff:fe79:40cd/64 scope link
        valid_lft forever preferred_lft forever
root@debian11: "# _
```

5. Установите на реальном хосте программу Wireshark (<https://www.wireshark.org>).

Если вы используете WiFi при инсталляции прсар включите поддержку IEEE 802.11 .

6. На хосте с7-1 с помощью утилиты ping проверьте доступность внешней сети, послав 5 эхо-запросов на сервер 8.8.8.8 или 1.1.1.1 (!)

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
-- 8.8.8 ping statistics --
38 packets transmitted, 38 received, 0% packet loss, time 37085ms
rtt min/avg/max/mdev = 19.757/38.162/96.563/21.310 ms
[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=108 time=50.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=100 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=108 time=40.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=108 time=30.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=108 time=40.7 ms
^C
-- 8.8.8 ping statistics --
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 30.069/52.423/100.121/24.700 ms
[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=108 time=42.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=25.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=108 time=181 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=108 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=108 time=28.8 ms
^C
-- 8.8.8 ping statistics --
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 25.436/64.462/181.924/59.173 ms
[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=108 time=46.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=29.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=108 time=66.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=108 time=306 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=108 time=1389 ms
^C
-- 8.8.8 ping statistics --
7 packets transmitted, 5 received, 28% packet loss, time 6009ms
rtt min/avg/max/mdev = 29.019/367.638/1389.517/520.859 ms, pipe 2
[root@localhost ~]#
```

```
[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=108 time=44.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=20.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=108 time=18.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=108 time=20.3 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=108 time=25.9 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=108 time=21.4 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=108 time=43.4 ms
^C
-- 8.8.8 ping statistics --
7 packets transmitted, 7 received, 0% packet loss, time 6013ms
rtt min/avg/max/mdev = 18.533/27.818/44.270/10.354 ms
[root@localhost ~]# _
```

7. Проверьте на с7-1 наличие перечисленных утилит. В случае, если утилиты, упомянутые в работе отсутствуют на хосте, их следует установить.

a. nload

b. iftop

c. bmon

d. nethogs

e. mtr

f. traceroute

g. vnstat

h. nc

Часть 2. Диагностика соединения

1. Познакомитесь с ключами утилиты ping.

2. На машине с7-2 напишите команды ping, которые (!):

a. отправляют 10 пакетов на с7-1

```
root@debian11:~/labs/lab3# ping -c 10 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=2.21 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.991 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.703 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=1.10 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=1.10 ms
64 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=1.25 ms
64 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=0.986 ms
64 bytes from 10.0.2.15: icmp_seq=8 ttl=64 time=1.19 ms
64 bytes from 10.0.2.15: icmp_seq=9 ttl=64 time=0.989 ms
64 bytes from 10.0.2.15: icmp_seq=10 ttl=64 time=1.26 ms

--- 10.0.2.15 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9017ms
rtt min/avg/max/mdev = 0.703/1.177/2.207/0.376 ms
root@debian11:~/labs/lab3# _
```

b. отправляют 10 пакетов с интервалом 10 секунд на машину с7-1

```
root@debian11:~/labs/lab3# ping -c 10 -i 10 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=1.33 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=1.44 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=1.33 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.977 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=1.31 ms
64 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=1.18 ms
64 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=1.19 ms
64 bytes from 10.0.2.15: icmp_seq=8 ttl=64 time=1.37 ms
64 bytes from 10.0.2.15: icmp_seq=9 ttl=64 time=0.972 ms
64 bytes from 10.0.2.15: icmp_seq=10 ttl=64 time=1.09 ms

--- 10.0.2.15 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 90071ms
rtt min/avg/max/mdev = 0.972/1.219/1.439/0.155 ms
root@debian11:~/labs/lab3# _
```

c. отправляет 5 пакетов размером 1500 байт на машину с7-1

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
root@debian11:~/labs/lab3# ping -c 5 -l 1500 10.0.2.15
ping: WARNING: probably, rcvbuf is not enough to hold preload
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=1.90 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=2.26 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=1.36 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.922 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=0.435 ms

--- 10.0.2.15 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 0.435/1.375/2.262/0.655 ms, pipe 5
root@debian11:~/labs/lab3#
```

3. Выясните что означают использование ключа **-f** (используйте его только при использовании утилиты ping между хостами c7-1 и c7-2)

-f - Установка флага, запрещающего фрагментацию пакета. Используется для заполнения сети, отправляя сто или более пакетов в секунду.

4. Познакомитесь с ключами утилиты mtr.

```
[root@localhost ~]# mtr --help
usage: mtr [-BfhvwrctglxspQomniuT46] [--help] [--version] [--report]
           [--report-wide] [--report-cycles=COUNT] [--curses] [--gtk]
           [--csv|-C] [--raw] [--xml] [--split] [--mpls] [--no-dns] [--show-ips]
           [--address interface] [--filename=FILE|-F]
           [--ipinfo=item_no|-y item_no]
           [--aslookup|-z]
           [--psize=bytes/-s bytes] [--order fields]
           [--report-wide|-w] [--inet] [--inet6] [--max-ttl=NUM] [--first-ttl=NUM]
           [--bitpattern=NUM] [--tos=NUM] [--udp] [--tcp] [--port=PORT] [--timeout=SECONDS]
           [--interval=SECONDS] HOSTNAME
[root@localhost ~]#
```

5. С хоста c7-1 соберите статистику соединения с хостом www.itmo.ru

My traceroute [v0.85]								Thu Mar 23 02:20:50 2023		
localhost.localdomain (0.0.0.0)										
Keys: Help Display mode Restart statistics Order of fields quit										
Host	Packets				Pings				Wrst	StDev
	Loss%	Snt	Last	Avg	Best	Wrst	StDev			
1. 10.0.0.1	0.0%	9	1.3	3.1	0.8	18.5	5.8			
2. www.huaweimobilewifi.com	0.0%	9	4.6	5.1	3.8	7.6	1.1			
3. ???										
4. 10.17.140.62	11.1%	9	19.6	29.3	19.6	51.8	12.9			
5. 10.17.136.54	0.0%	9	40.8	37.8	23.5	55.9	11.5			
6. oct-cr03-be79.100.spb.mts-internet.net	0.0%	8	36.0	33.5	23.2	53.4	10.4			
7. oct-cr01-be2.78.spb.mts-internet.net	37.5%	8	34.2	28.3	22.3	34.3	5.6			
8. mag9-cr02-be1.78.msk.mts-internet.net	62.5%	8	46.4	48.3	46.4	49.5	1.6			
9. m9-cr04-be8.77.msk.mts-internet.net	0.0%	8	76.6	57.1	38.8	76.6	12.3			
10. m9-cr03-ae13.77.msk.mts-internet.net	0.0%	8	55.9	45.3	33.7	55.9	8.6			
11. m9-cr03-as200350.msk.mts-internet.net	0.0%	8	66.5	51.5	34.3	75.9	13.8			
12. ???										

6. Определите значение всех параметров, выводимых утилитой mtr.

host — имя хоста;

Loss% — процент потерь пакетов;

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

Snt — количество отправленных пакетов;

Last — время задержки последнего отправленного пакета в миллисекундах;

Avg — среднее время задержки;

Best — минимальное время задержки;

Wrst — максимальное время задержки;

StDev — среднеквадратичное отклонение времени задержки.

7. Напишите команду, которая сохранит в файл расширенную статистику работы mtr при отправке 40 пакетов (!).

```
[root@localhost labs]# mtr -c 40 -r www.itmo.ru > logs2.txt
[root@localhost labs]# _
```

Start: Thu Mar 23 02:24:14 2023								
HOST:	localhost.localdomain	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.--- gateway		0.0%	40	1.0	0.8	0.6	1.5	0.0
2.--- www.huaweinomobilewifi.com		0.0%	40	7.7	5.8	3.8	13.8	2.1
3.--- ???		100.0	40	0.0	0.0	0.0	0.0	0.0
4.--- 10.17.140.62		2.5%	40	28.9	38.8	20.0	100.5	16.3
5.--- 10.17.136.54		2.5%	40	88.1	37.9	20.6	88.1	14.4
6.--- oct-cr03-be79.100.spb.mts		0.0%	40	38.7	43.7	20.0	149.0	26.7
7.--- oct-cr01-be2.78.spb.mts-i		62.5%	40	44.7	42.1	22.6	88.9	20.1
8.--- mag9-cr02-be1.78.msk.mts-		75.0%	40	33.1	55.4	33.1	68.7	12.7
9.--- m9-cr04-be8.77.msk.mts-in		0.0%	40	42.3	58.5	31.1	149.5	27.2
10.--- m9-cr03-ae13.77.msk.mts-i		0.0%	40	40.2	52.3	32.1	237.7	34.6
11.--- m9-cr03-as200350.msk.mts-		0.0%	40	40.1	47.5	32.0	177.2	22.7
12.--- ???		100.0	40	0.0	0.0	0.0	0.0	0.0

Часть 3. Работа с Wireshark

1. Настройте перехват трафика на реальном интерфейсе, так чтобы он завершился после сбора 5 Мб (для увеличения интенсивности генерации кадров открыть любой сайт в браузере).

2. Используя инструментарий статистики, определите (!):

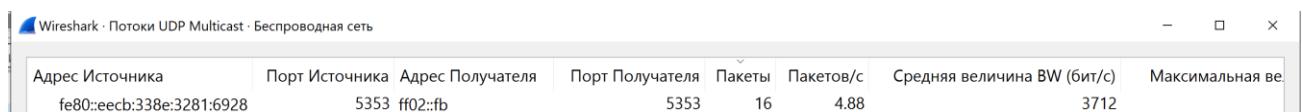
а. Узел с максимальной активностью (по объему переданных данных),

Статистика->Конечные узлы + сортировка по байтам:



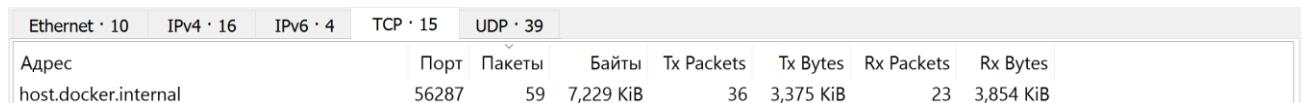
б. Узел, осуществивший наибольшее количество широковещательных

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux
рассылок,

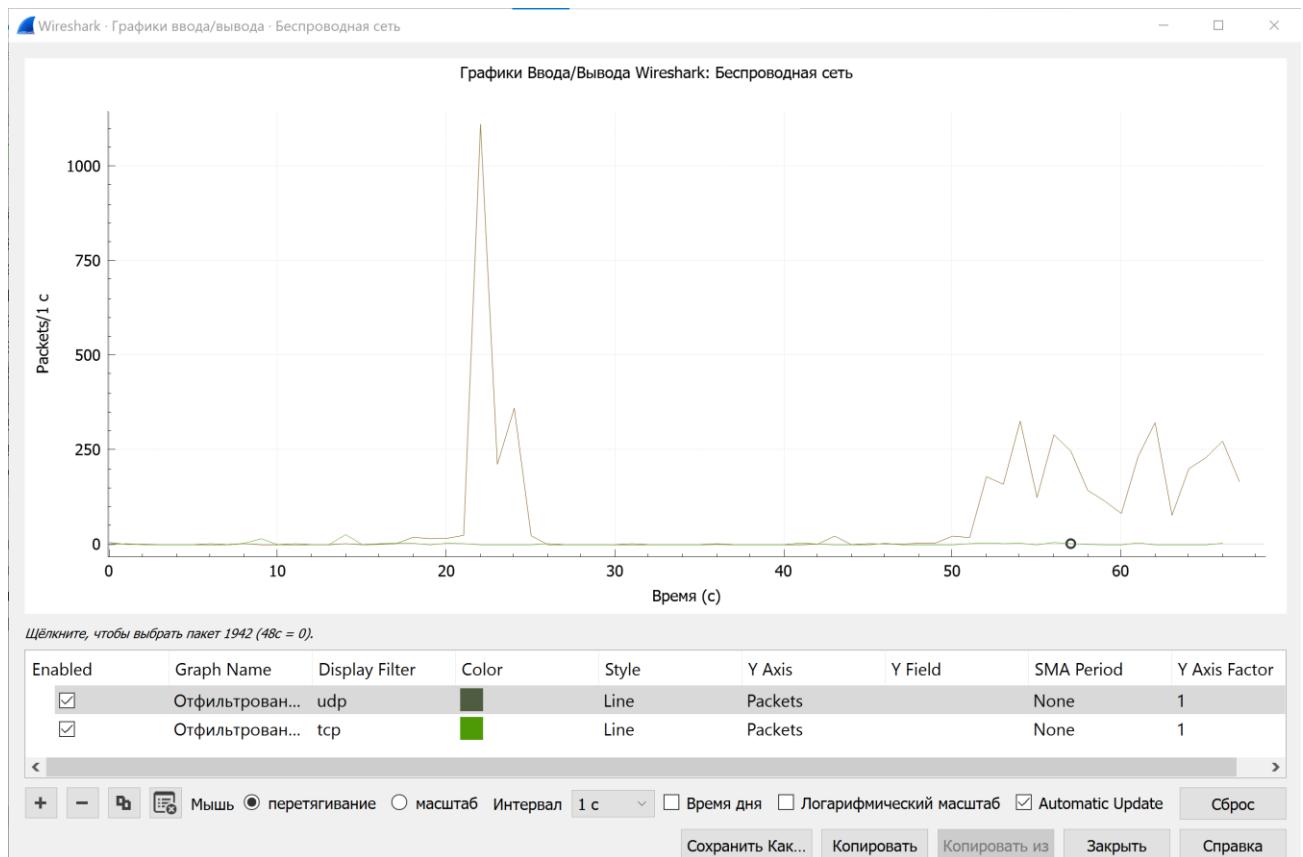


с. Самый активный TCP-порт на хосте (по количеству переданных пакетов)

По протоколу TCP:



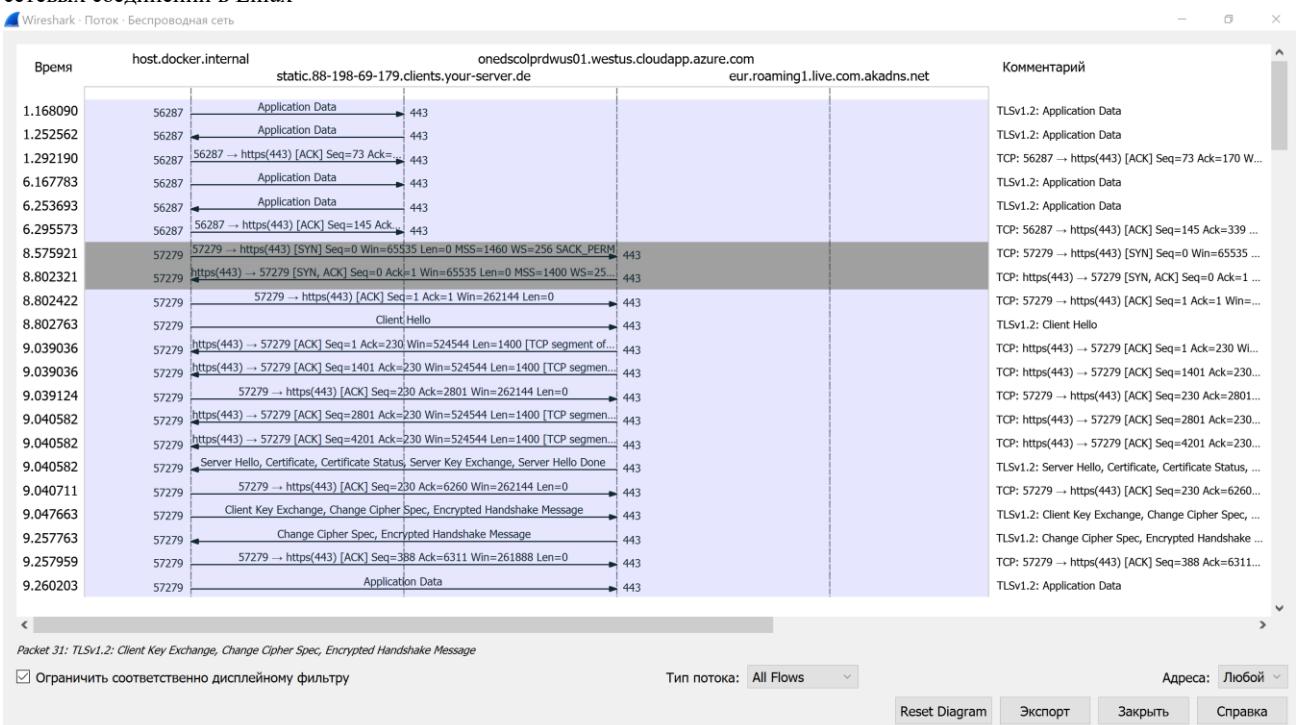
d. Постройте на одной координатной сетке постройте графики интенсивности TCP и UDP трафика (пункт Io Graphs).



(Последовательно применить фильтры и на каждом шаге открывать Графики Ввода/вывода, тогда указанные данные отобразятся на одном графике)

е. Постройте диаграмму связей только для пакетов, содержащих сообщения протокола HTTPS (пункт Flow Graph)

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux



Для этого нужно применить фильтр: `tcp.port == 443`

3. Напишите фильтры, которые выделяют из общего числа пакеты (!):

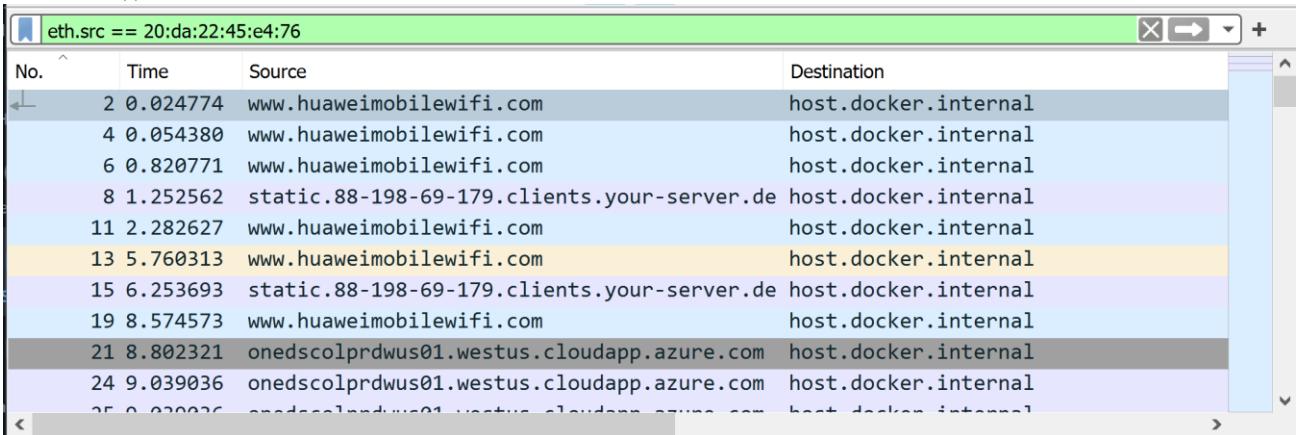
a. Отбирающие сообщения протоколов HTTP и FTP и относящиеся только к взаимодействию локальных клиентов и внешнего сервера. То есть в случае, если на вашем компьютере запущен и Web-браузер и Web-сервер, фильтр должен отбирать только трафик от и к Web-браузеру, игнорируя трафик от и к Web-серверу.

(ip.src == 192.168.8.100 ip.dst == 192.168.8.100) && (http.request http.response ftp.request ftp.response)			
No.	Time	Source	Destination
1930	45.996460	host.docker.internal	239.255.255.250
1936	46.998629	host.docker.internal	239.255.255.250
1938	47.999051	host.docker.internal	239.255.255.250
1942	48.999144	host.docker.internal	239.255.255.250

((ip.src == 192.168.8.100 && ip.dst == 192.168.8.1) (ip.dst == 192.168.8.100 && ip.src == 192.168.8.1)) && (http.request http.response)

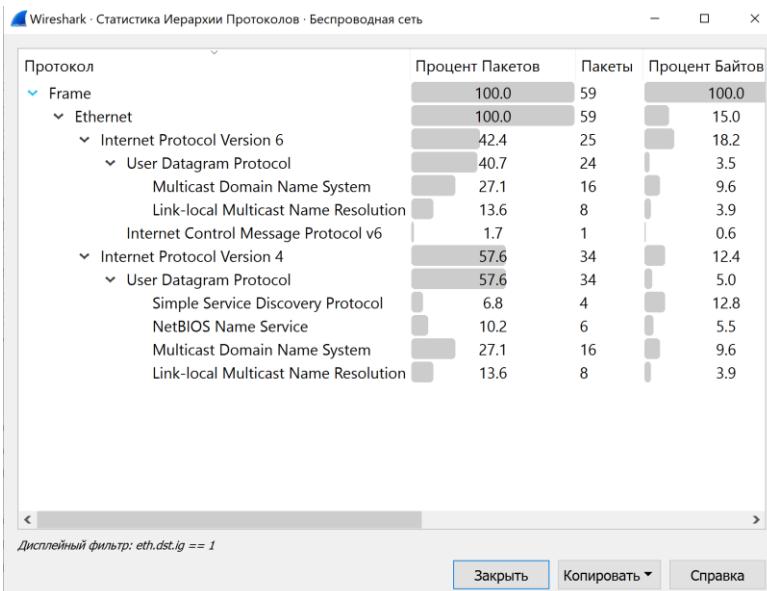
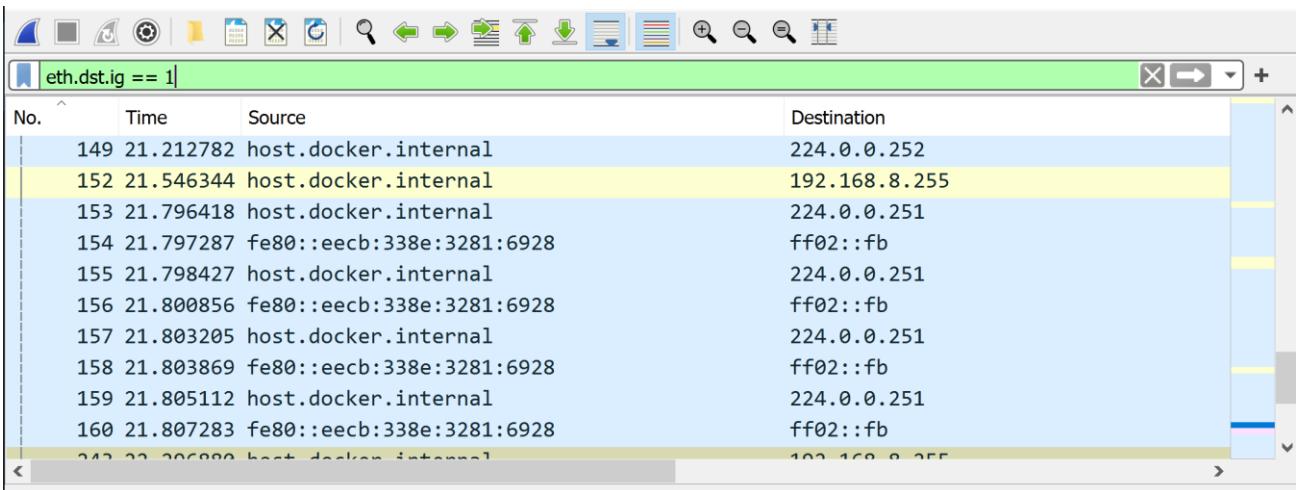
b. Все кадры Ethernet, отправленные с сетевого интерфейса хоста.

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux



с. Напишите фильтр, отбирающий только широковещательные сообщения.

Определите назначение 3-х широковещательных рассылок разных протоколов (или тех, которые удалось обнаружить).



Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

Протоколы: upd, icmpv6

- d. Определить адреса, на которые поступают данные кадры и пакеты для канального и сетевого уровня.

Endpoint Settings

- Разрешение имён
- Ограничить соответственно д/р

Копировать

Карта

Ethernet · 8	IPv4 · 5	IPv6 · 4	TCP	UDP · 18		
Адрес	Пакеты	Байты	Total Packets	Percent Filtered	Tx Packets	Tx Byte
01:00:5e:00:00:fb	16	1,172 KiB	16	100.00%	0	0 байты
01:00:5e:00:00:fc	8	552 байты	8	100.00%	0	0 байты
01:00:5e:7f:ff:fa	4	868 байты	4	100.00%	0	0 байты
28:c6:3f:af:1d:c9	59	5,361 KiB	5 208	1.13%	59	5,361 KiB
33:33:00:00:00:fb	16	1,484 KiB	16	100.00%	0	0 байты
33:33:00:01:00:03	8	712 байты	8	100.00%	0	0 байты
33:33:ff:76:02:e5	1	86 байты	1	100.00%	0	0 байты
ff:ff:ff:ff:ff:ff	6	552 байты	6	100.00%	0	0 байты

Protokol

- Bluetooth
- DCCP
- Ethernet
- FC
- FDDI
- IEEE 802.11
- IEEE 802.15.4
- IPv4

Filter list for specific type

Закрыть Справка

Ethernet · 8 IPv4 · 5 IPv6 · 4 TCP UDP · 18

Адрес	Порт	Пакеты	Байты	Total Packets	Percent Filtered	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
192.168.8.100	137	6	552 байты	6	100.00%	6	552 байты	0	0 байты
192.168.8.100	5353	16	1,172 KiB	16	100.00%	16	1,172 KiB	0	0 байты
192.168.8.100	56565	4	868 байты	4	100.00%	4	868 байты	0	0 байты
192.168.8.100	57198	2	138 байты	2	100.00%	2	138 байты	0	0 байты
192.168.8.100	57923	2	138 байты	2	100.00%	2	138 байты	0	0 байты
192.168.8.100	58589	2	138 байты	2	100.00%	2	138 байты	0	0 байты
192.168.8.100	61899	2	138 байты	2	100.00%	2	138 байты	0	0 байты
192.168.8.255	137	6	552 байты	6	100.00%	0	0 байты	6	552 байты
224.0.0.251	5353	16	1,172 KiB	19	84.21%	0	0 байты	16	1,172 KiB
224.0.0.252	5355	8	552 байты	8	100.00%	0	0 байты	8	552 байты
239.255.255.250	1900	4	868 байты	4	100.00%	0	0 байты	4	868 байты
fe80::eecb:338e:3281:6928	5353	16	1,484 KiB	16	100.00%	16	1,484 KiB	0	0 байты
fe80::eecb:338e:3281:6928	57198	2	178 байты	2	100.00%	2	178 байты	0	0 байты
fe80::eecb:338e:3281:6928	57923	2	178 байты	2	100.00%	2	178 байты	0	0 байты
fe80::eecb:338e:3281:6928	58589	2	178 байты	2	100.00%	2	178 байты	0	0 байты
fe80::eecb:338e:3281:6928	61899	2	178 байты	2	100.00%	2	178 байты	0	0 байты
ff02::1:3	5355	8	712 байты	8	100.00%	0	0 байты	8	712 байты
ff02::fb	5353	16	1,484 KiB	16	100.00%	0	0 байты	16	1,484 KiB

- e. Напишите фильтры для каждой из трех широковещательных рассылок, выбранных в пункте 3-с.

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

The top Wireshark window displays a list of UDP packets captured on interface eth0. The columns shown are No., Time, Source (S), Destination, and a detailed view of the packet content. The destination IP address for most packets is 192.168.8.255, while others are 224.0.0.251 or ff02::fb. The source IP address is fe80::eecb:3281:6928. The bottom Wireshark window displays a single ICMPv6 packet captured on interface eth0, with the source IP address fe80::eecb:3281:6928 and the destination IP address ff02::1:ff76:2e5.

No.	Time	S	Destination
152	21.546344	udp udpcap udpencap udplite fe80::eecb:3281:6928	192.168.8.255 224.0.0.251 ff02::fb 224.0.0.251 ff02::fb 224.0.0.251 ff02::fb 224.0.0.251 ff02::fb 192.168.8.255 224.0.0.251
153	21.796418	r.internal	
154	21.797287	r.internal	
155	21.798427	host.docker.internal	
156	21.800856	fe80::eecb:3281:6928	
157	21.803205	host.docker.internal	
158	21.803869	fe80::eecb:3281:6928	
159	21.805112	host.docker.internal	
160	21.807283	fe80::eecb:3281:6928	
243	22.296880	host.docker.internal	
1020	45.006460	host.docker.internal	

No.	Time	S	Destination
1498	23.258318	fe80::eecb:3281:6928	ff02::1:ff76:2e5

f. На основании собранной статистики и анализа адресов определить, к какому типу коммутационного оборудования подключен используемый компьютер (концентратор, коммутатор или маршрутизатор).

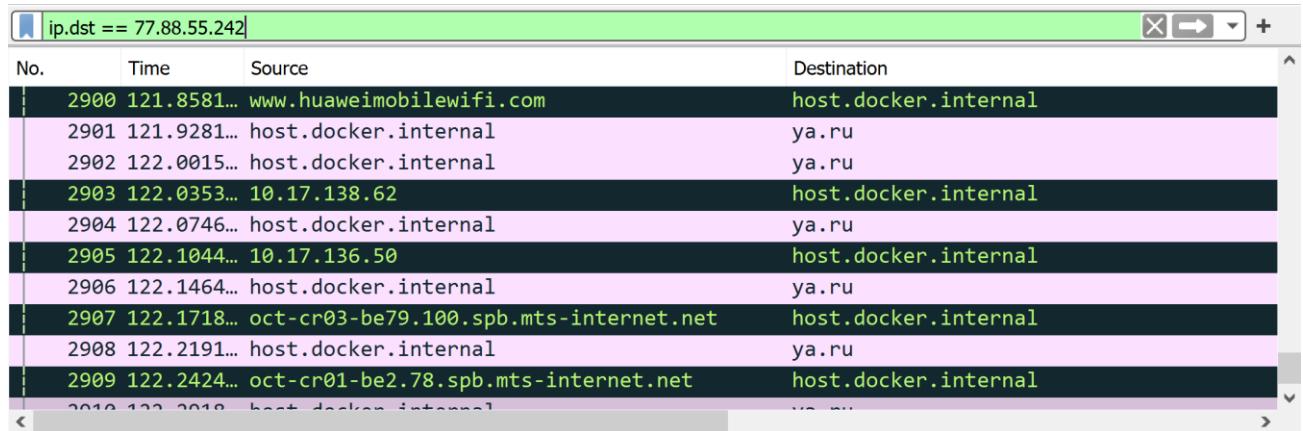
Скорее всего коммутатор.

4. В виртуальной машине с помощью утилиты mtr выведите статистику передачи трафика до хоста ya.ru, отправив 111 запросов и выводя на экран, как имена, так и ip адреса промежуточных устройств (!).

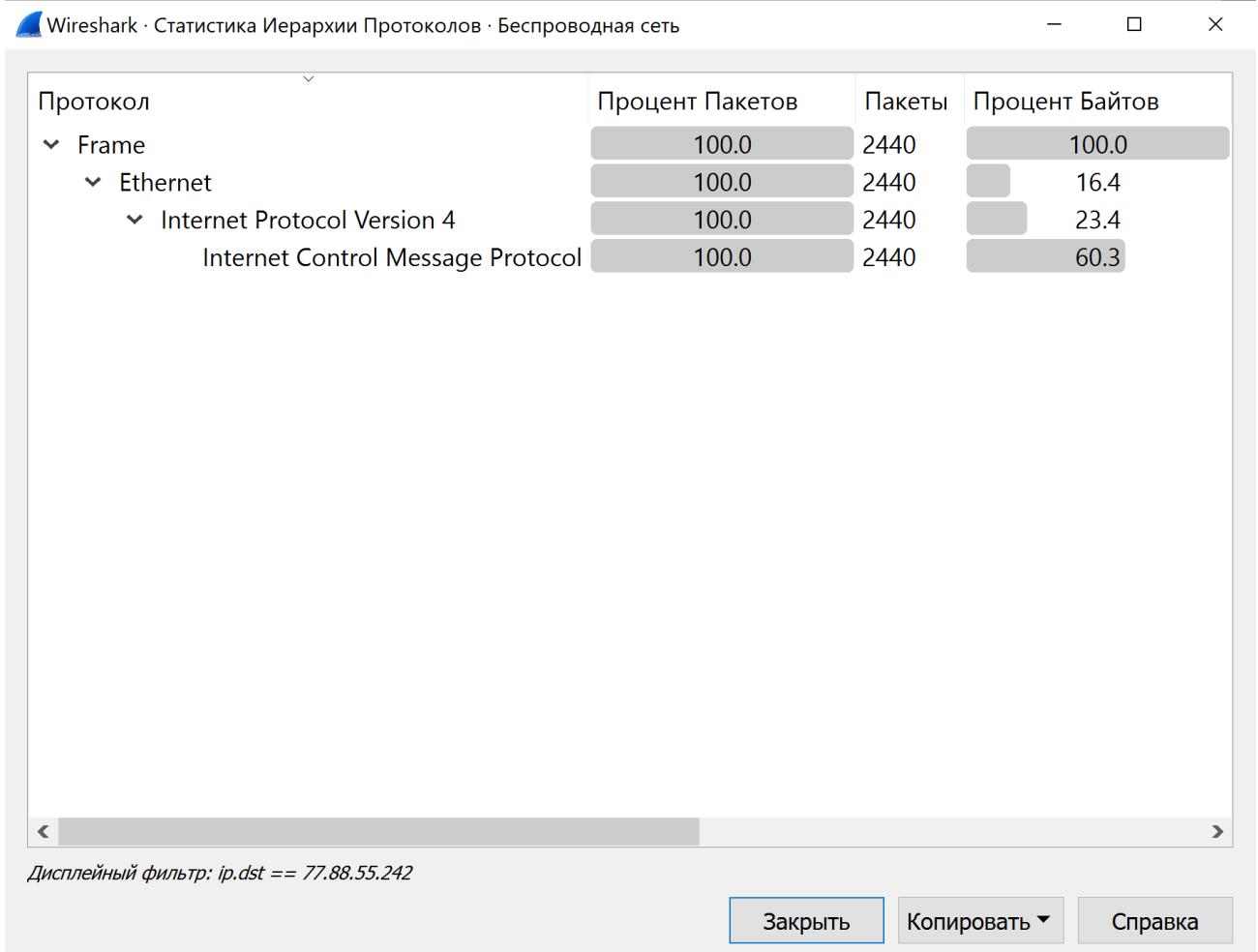
Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
Start: Thu Mar 23 19:47:27 2023
HOST: localhost.localdomain      Loss%  Snt   Last    Avg   Best  Wrst StDev
1.-- gateway (10.0.0.1)          0.0%  111    0.6    1.0   0.4   6.7   0.9
2.-- www.huaweimobilewifi.com   0.0%  111    4.0    5.4   3.8   18.4  2.2
3.-- ???                         100.0  111    0.0    0.0   0.0   0.0   0.0
4.-- 10.17.138.62               0.9%  111   37.7   25.2  18.4  40.0  4.5
5.-- 10.17.136.50               8.9%  111   35.7   25.8  17.8  64.4  7.4
6.-- oct-cr03-be79.100.spb.mts  0.0%  111   36.5   26.2  18.6  40.0  5.3
7.-- oct-cr01-be2.78.spb.mts-i  52.3% 111   43.7   26.9  19.4  43.7  5.4
8.-- mag9-cr02-be1.78.msk.mts-  87.4% 111   60.4   39.6  32.0  60.4  7.0
9.-- a197-cr01-ae10.77.msk.mts 28.8%  111   36.4   41.1  31.4  64.5  6.2
10.-- a197-cr04-be31.77.msk.mts 1.8%  111   32.5   36.3  29.9  78.4  5.6
11.-- a197-cr04-as13238.msk.mts 0.9%  111   32.7   37.5  29.5  60.5  4.8
12.-- sas-32z1-ae2.yndx.net (87  0.9%  111   58.9   50.5  37.5  79.0  7.4
13.-- 10.4.1.1                  2.7%  111   41.6   44.6  37.2  77.3  6.5
14.-- ya.ru (77.88.55.242)       0.9%  111   40.1   43.0  37.0  59.4  4.2
```

5. В Wireshark напишите фильтр, отбирающий сетевые сообщения из п. 4. Определите, с помощью какого протокола осуществляется проверка доступности (!).



Протокол ICMP (как и указано по умолчанию в mtr):



Часть 4. Определение маршрута прохождения пакета

1. Познакомитесь с ключами утилиты traceroute.
2. На машине c7-1 напишите команды traceroute, которые (!):
 - a. определяют маршрут до хоста 8.8.8.8 с помощью ICMP

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
[root@localhost lab3]# traceroute -I 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 gateway (10.0.0.1)  0.930 ms  0.542 ms  0.513 ms
 2 www.huaweimobilewifi.com (192.168.8.1)  13.967 ms  12.960 ms  11.998 ms
 3 * * *
 4 10.17.138.62 (10.17.138.62)  43.229 ms  47.161 ms  55.059 ms
 5 10.17.136.50 (10.17.136.50)  55.568 ms  56.090 ms  57.716 ms
 6 oct-cr03-be79.100.spb.mts-internet.net (212.188.18.189)  60.605 ms  28.499 ms  36.021 ms
 7 72.14.197.244 (72.14.197.244)  39.472 ms  34.250 ms  34.716 ms
 8 74.125.244.129 (74.125.244.129)  34.640 ms  32.813 ms *
 9 74.125.244.133 (74.125.244.133)  33.740 ms  48.051 ms  56.580 ms
10 72.14.232.84 (72.14.232.84)  56.312 ms  55.572 ms  53.645 ms
11 142.251.61.219 (142.251.61.219)  38.028 ms  45.884 ms  48.590 ms
12 142.250.56.15 (142.250.56.15)  47.733 ms  46.625 ms  44.841 ms
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 dns.google (8.8.8.8)  42.607 ms  31.141 ms  39.782 ms
[root@localhost lab3]# _
```

б. определяют маршрут до хоста 8.8.8.8 с помощью UDP

```
[root@localhost lab3]# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 gateway (10.0.0.1)  1.154 ms  0.845 ms  0.680 ms
 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
[root@localhost lab3]# _
```

(по умолчанию)

с. определяют маршрут до хоста 8.8.8.8 с помощью TCP

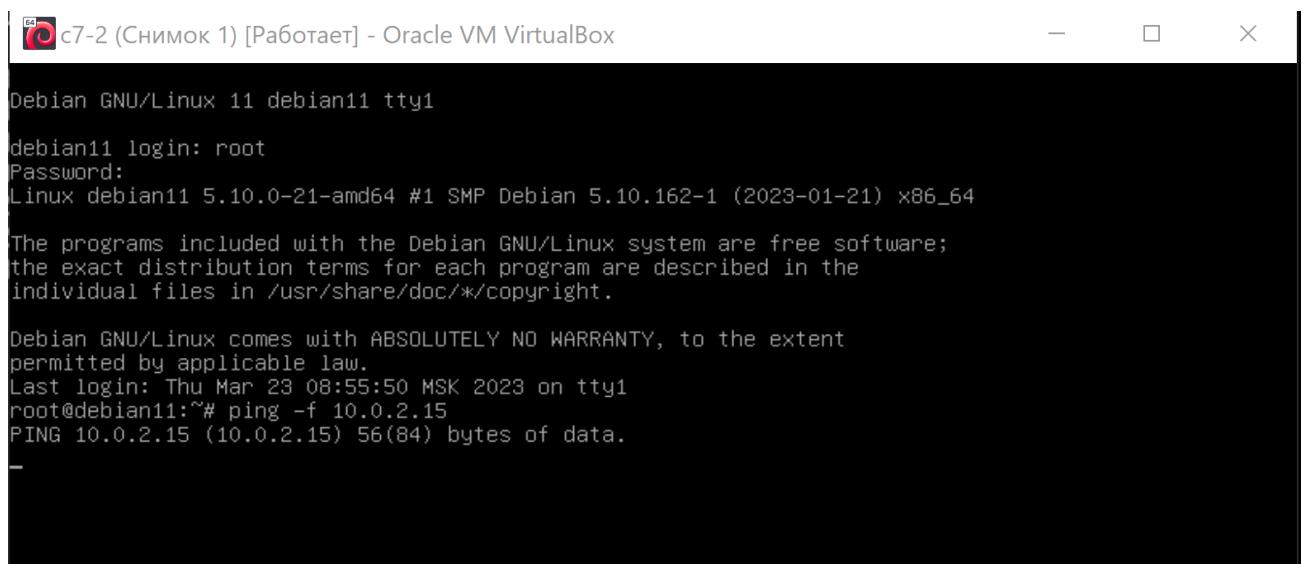
Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
[root@localhost lab3]# traceroute -T 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
1 * * *
2 * * *
3 * * *
4 * * *
5 * * *
6 * * *
7 * * *
8 * * *
9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *

[root@localhost lab3]#
```

Часть 5. Текущий мониторинг сетевых интерфейсов

1. С хоста c7-2 запустите отправку запросов утилитой ping в режиме flood на внутренний интерфейс c7-1.



The screenshot shows a terminal window titled "c7-2 (Снимок 1) [Работает] - Oracle VM VirtualBox". The terminal displays the following Debian boot messages:

```
Debian GNU/Linux 11 debian11 tty1
debian11 login: root
Password:
Linux debian11 5.10.0-21-amd64 #1 SMP Debian 5.10.162-1 (2023-01-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Mar 23 08:55:50 MSK 2023 on tty1
root@debian11:~# ping -f 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
```

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

2. На хосте c7-1 последовательно с помощью любой из утилит nload, iftop или bmon получите данные о загрузке интерфейса, на который отправляет трафик хост c7-2 (!).

```
Device enp0s3 [10.0.2.15] (1/2):
=====
Incoming:
[REDACTED]
          Curr: 399.19 kBit/s
          Avg: 336.90 kBit/s
          Min: 0.00 Bit/s
          Max: 501.52 kBit/s
          Ttl: 7.59 MByte
Outgoing:
[REDACTED]
          Curr: 399.19 kBit/s
          Avg: 336.88 kBit/s
          Min: 0.00 Bit/s
          Max: 501.52 kBit/s
          Ttl: 7.81 MByte
```

3. Изменяйте размер пакета, передаваемой утилитой ping пакета от 100 до 60100 с шагом 10000. Определите, как меняется загрузка на сетевом интерфейсе (!).

```
root@debian11:~# ping -f -s 100 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 100(128) bytes of data.
```

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

Device enp0s3 [10.0.2.15] (1/2):

=====
Incoming:

```
Curr: 529.50 kBit/s  
Avg: 12.06 MBit/s  
Min: 0.00 Bit/s  
Max: 23.20 MBit/s  
Ttl: 746.57 MByte
```

Outgoing:

```
Curr: 529.50 kBit/s  
Avg: 12.06 MBit/s  
Min: 0.00 Bit/s  
Max: 23.06 MBit/s  
Ttl: 746.74 MByte
```

```
root@debian11:~# ping -f -s 10100 10.0.2.15  
PING 10.0.2.15 (10.0.2.15) 10100(10128) bytes of data.  
-
```

Device enp0s3 [10.0.2.15] (1/2):

=====
Incoming:

```
Curr: 11.82 MBit/s  
Avg: 14.13 MBit/s  
Min: 0.00 Bit/s  
Max: 23.20 MBit/s  
Ttl: 670.36 MByte
```

Outgoing:

```
Curr: 11.76 MBit/s  
Avg: 14.13 MBit/s  
Min: 0.00 Bit/s  
Max: 23.06 MBit/s  
Ttl: 670.52 MByte
```

```
root@debian11:~# ping -f -s 20100 10.0.2.15  
PING 10.0.2.15 (10.0.2.15) 20100(20128) bytes of data.  
-
```

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

Device enp0s3 [10.0.2.15] (1/2):

Device Express Technologies, Inc.

Incoming:

Outgoing: Curr: 18.59 Mbit/s
Avg: 15.65 MBit/s
Min: 0.00 Bit/s
Max: 23.20 MBit/s
Ttl: 483.84 MByte

Outgoing:

```
root@debian11:~# ping -f -s 30100 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 30100(30128) bytes of data.
```

Device enp0s3 [10.0.2.15] (1/2):

Incoming:

Outgoing:

Curr: 20.77 MBit/s
Avg: 10.36 MBit/s
Min: 0.00 Bit/s
Max: 24.14 MBit/s
Ttl: 863.90 MByte

```
root@debian11:~# ping -f -s 40100 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 40100(40128) bytes of data.
```

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

Device enp0s3 [10.0.2.15] (1/2):

Incoming:

Outgoing:

```
root@debian11:~# ping -f -s 50100 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 50100(50128) bytes of data.
```

Device enp0s3 [10.0.2.15] (1/2):

Journal of Health Politics, Policy and Law, Vol. 35, No. 4, December 2010
DOI 10.1215/03616878-35-4 © 2010 by The University of Chicago

Incoming:

Outgoing:

```
root@debian11:~# ping -f -s 60100 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 60100(60128) bytes of data.
```

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

Часть 6. Сбор статистики о загрузки сетевого интерфейса

1. На хосте c7-1 запустите демон vnstat.
 2. Поставите на мониторинг интерфейс, через который машина c7-1 подключена к с7-1

```
vnstat --create -i enp0s3
```

vnstat -i enp0s3 -l

3. С хоста c7-2 запустите отправку запросов утилитой ping в режиме flood, так чтобы работа утилиты прекратилась после отправки 500 пакетов.

```
root@debian11:~# ping -f -c 500 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.

--- 10.0.2.15 ping statistics ---
500 packets transmitted, 500 received, 0% packet loss, time 635ms
rtt min/avg/max/mdev = 0.289/0.967/9.565/0.828 ms, pipe 2, ipg/ewma 1.272/1.306 ms
root@debian11:~#
```

4. Выведите статистику собранного трафика (!).

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
[root@localhost lab3]# vnstat -i enp0s3 -l
Monitoring enp0s3... (press CTRL-C to stop)

rx:      0 kbit/s  0 p/s          tx:      0 kbit/s  0 p/s^C

enp0s3  /  traffic statistics

          rx          |          tx
-----+-----+
bytes      49 KiB |      48 KiB
-----+-----+
max        146 kbit/s |     145 kbit/s
average    3.89 kbit/s |     3.87 kbit/s
min        0 kbit/s  |     0 kbit/s
-----+-----+
packets    504 |      504
-----+-----+
max        190 p/s  |     190 p/s
average    5 p/s   |     5 p/s
min        0 p/s   |     0 p/s
-----+-----+
time       1.67 minutes

[root@localhost lab3]#
```

Часть 7. Диагностика работы приложений через сеть

1. Установите несколько соединений с SSH сервером на хосте c7-1 с хоста c7-2. Для простоты можно открыть несколько физических консолей или запускать ssh клиент в скрипте, передавая пароль в явном виде с помощью утилиты sshpass (sshpass -p MyPlainPassword_DontBeatMeSecurityMamager ssh username@host_address). Никогда не поступайте так в реальной жизни! Если нужно используйте аутентификацию по ключам.

ssh root@10.0.2.15 + пароль и также с другой консоли

2. Используя утилиту netstat или lsof на c7-1 вывести все активные (прослушиваемые) порты. (!)

```
[root@localhost lab3]# netstat -pnltu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp     0      0 0.0.0.0:22                0.0.0.0:*            LISTEN    872/sshd
tcp     0      0 127.0.0.1:25              0.0.0.0:*            LISTEN    1028/master
tcp6    0      0 ::1:22                   ::*:*               LISTEN    872/sshd
tcp6    0      0 ::1:25                   ::*:*               LISTEN    1028/master
udp     0      0 127.0.0.1:323             0.0.0.0:*            LISTEN    649/chrony
udp     0      0 0.0.0.0:68                0.0.0.0:*            LISTEN    1254/dhcclient
udp6    0      0 ::1:323                  ::*:*               LISTEN    649/chrony
[root@localhost lab3]# _
```

3. Используя утилиту netstat или ss все установленные соединения (!).

```
[root@localhost lab3]# netstat -lnpptux > logs7.txt_
```

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

Active Internet connections (only servers)						
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	872/sshd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	1028/master
tcp6	0	0	:::22	:::*	LISTEN	872/sshd
tcp6	0	0	:::1:25	:::*	LISTEN	1028/master
udp	0	0	127.0.0.1:323	0.0.0.0:*		649/chronyrd
udp	0	0	0.0.0.0:68	0.0.0.0:*		1254/dhcclient
udp6	0	0	:::1:323	:::*		649/chronyrd
Active UNIX domain sockets (only servers)						
Proto	RefCnt	Flags	Type	State	I-Node	PID/Program name
unix	2	[ACC]	STREAM	LISTENING	17922	1028/master
unix	2	[ACC]	STREAM	LISTENING	17925	1028/master
unix	2	[ACC]	STREAM	LISTENING	17934	1028/master
unix	2	[ACC]	STREAM	LISTENING	17937	1028/master
unix	2	[ACC]	STREAM	LISTENING	17940	1028/master
unix	2	[ACC]	STREAM	LISTENING	17943	1028/master
unix	2	[ACC]	STREAM	LISTENING	12300	1/systemd
unix	2	[ACC]	STREAM	LISTENING	17946	1028/master
unix	2	[ACC]	STREAM	LISTENING	12046	1/systemd
unix	2	[ACC]	STREAM	LISTENING	17949	1028/master
unix	2	[ACC]	STREAM	LISTENING	17952	1028/master
unix	2	[ACC]	STREAM	LISTENING	17955	1028/master
unix	2	[ACC]	STREAM	LISTENING	17928	1028/master
unix	2	[ACC]	STREAM	LISTENING	17887	1028/master
unix	2	[ACC]	STREAM	LISTENING	17891	1028/master
unix	2	[ACC]	STREAM	LISTENING	17894	1028/master
unix	2	[ACC]	STREAM	LISTENING	17916	1028/master
unix	2	[ACC]	SEQPACKET	LISTENING	12063	1/systemd
unix	2	[ACC]	STREAM	LISTENING	14379	1/systemd
unix	2	[ACC]	STREAM	LISTENING	7477	1/systemd
unix	2	[ACC]	STREAM	LISTENING	18501	681/NetworkManager
						Path
						/private/proxywrite
						/private/smtp
						/private/error
						/private/retry
						/private/discard
						/private/local
						/run/lvm/lvmetad.so\$
						/private/virtual
						/run/systemd/private
						/private/lmtp
						/private/anvil
						/private/scache
						/private/relay
						/public/pickup
						/public/cleanup
						/public/qmgr
						/public/fflush
						/run/udev/control
						/run/dbus/system_bu\$
						/run/systemd/journ\$
						/var/run/NetworkMan\$
unix	2	[ACC]	STREAM	LISTENING	18501	681/NetworkManager
unix	2	[ACC]	STREAM	LISTENING	17919	1028/master
unix	2	[ACC]	STREAM	LISTENING	12402	1/systemd
unix	2	[ACC]	STREAM	LISTENING	17907	1028/master
unix	2	[ACC]	STREAM	LISTENING	17910	1028/master
unix	2	[ACC]	STREAM	LISTENING	17931	1028/master
unix	2	[ACC]	STREAM	LISTENING	17898	1028/master
unix	2	[ACC]	STREAM	LISTENING	17901	1028/master
unix	2	[ACC]	STREAM	LISTENING	17904	1028/master
unix	2	[ACC]	STREAM	LISTENING	17913	1028/master

4. Напишите скрипт, который выводит список IP-адресов и количество подключений с ними к нашему хосту через порт, задаваемый параметрами скрипта (значение по умолчанию 22). Список упорядочить по количеству соединений с IP адреса. Ради большей наглядности результатов вы можете дополнительно подключиться по SSH к c7-1 с основного хоста или с дополнительных виртуальных машин. Для выполнения задания вам могут понадобиться утилиты grep, awk, cut, sort и uniq, но в выборе инструментов вы не ограничены. (!)

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

```
#!/bin/bash
port1=":"
if [[ $1 == "" ]]
then
    port1=$port1"22"
else
    port1=$port1$1
fi
netstat -lnp tu | awk '{if ($4 ~ "'"$port1'"") {print $4, $5}}' | sort -k2 | uniq -c
```

```
[root@localhost lab3]# ./script?
      1 ::::22 ::::*
      1 0.0.0.0:22 0.0.0.0:*
[root@localhost lab3]#
```

5. Закройте все соединения по ssh с хостом с7-1.

exit

6. Познакомитесь с ключами утилиты nethogs.

```
[root@localhost lab3]# nethogs --help
nethogs: invalid option -- '_'
usage: nethogs [-V] [-h] [-b] [-d seconds] [-v mode] [-c count] [-t] [-p] [-s] [device [device [device ...]]]
              -V : prints version.
              -h : prints this help.
              -b : bughunt mode - implies tracemode.
              -d : delay for update refresh rate in seconds. default is 1.
              -v : view mode (0 = KB/s, 1 = total KB, 2 = total B, 3 = total MB). default is 0.
              -c : number of updates. default is 0 (unlimited).
              -t : tracemode.
              -p : sniff in promiscuous mode (not recommended).
              -s : sort output by sent column.
              -a : monitor all devices, even loopback/stopped ones.
                    device : device(s) to monitor. default is all interfaces up and running excluding loopback
When nethogs is running, press:
  q: quit
  s: sort by SENT traffic
  r: sort by RECEIVE traffic
  m: switch between total (KB, B, MB) and KB/s mode
[root@localhost lab3]# _
```

7. С хоста с7-2 подключитесь по ssh к машине с7-1. В терминале ssh запустите утилиту top.

8. На хосте с7-1 с помощью утилиты nethogs определите (!)

- Среднюю скорость передачи данных до sshd.
- PID процесса sshd.

Лабораторная работа 3. Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux

NetHogs version 0.8.5					
PID	USER	PROGRAM	DEV	SENT	RECEIVED
7854	root	sshd: root@pts/0	enp0s3	1.699	0.039 KB/sec
	? root	unknown TCP		0.000	0.000 KB/sec
TOTAL				1.699	0.039 KB/sec

Часть 8. Работа с утилитой nc (NetCat)

1. На машине c7-1 на отдельной консоли запустите tcpdump для сбора всего трафика с портов 9999 и 4444, так, чтобы на консоль выводилось содержимое сообщения, а не только информация из служебных заголовков (!).

```
tcpdump -vv -i enp0s3 port 4444 or port 9999 -w file_name
```

2. Используя утилиту nc на обоих машинах передайте текстовый файл с произвольным текстовым содержимым (не менее 20 слов) принимая файл на порту tcp 9999 (!).

Открыть файл для получения:

```
[root@localhost lab3]# nc -l -p 9999 > newfile
```

Команда для передачи файла:

```
[root@debian11:~/labs/lab3# cat file2.txt | nc 10.0.2.15 9999]
```

3. Используя утилиту nc на обоих машинах организовать текстовый чат между машинами через порт udp 4444.

a. Hi! How are you?

b. Fine! And You?

c. So am i!

```
[root@localhost lab3]# nc -lp 4444
Hi! How are you?
Fine! And You?
So am i!
root@debian11:~/labs/lab3# nc 10.0.2.15 4444
Hi! How are you?
Fine! And You?
So am i!
```

Завершите сессию (Ctrl+C) (!).

Примечание: учитите, что, если у вас работает firewall, нужно будет его выключить (что плохо) или добавить разрешения на порты (что хорошо). Так, для FirewallD это можно сделать так: firewall-cmd --permanent --add-port=80/tcp.

4. Остановите работу `tcpdump`, проанализируйте перехваченные сообщения.

Какие выводы можно сделать?

Примечание: вывод `tcpdump` можно направить в файл с помощью ключа `-w`. Это будет файл стандарта `pcap`, который можно открыть в `Wireshark` для удобного анализа.

5. Этот пункт выполняется по желанию. С помощью nc можно организовать reverse shell. На машине с Linux Centos 7 с помощью ключа -e запустите команду /bin/bash с перенаправлением вывода-ввода на порт tcp 4445, так же как вы делали для организации чата. Со второй Linux машине подключитесь к порту 4445 и позадавайте команды bash, например получите версию ядра, адрес или hostname.