Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

Лабораторная Работа №3

по дисциплине “Компьютерные сети”

Выполнил студенты:
Алексеев Павел Алексеевич
Смирнов Тимур Олегович
Группа №K33421
Проверил:
Харитонов Антон

Санкт-Петербург
2022

Цель работы:

Получить практические навыки по работе с анализаторами сетевого трафика. На практике ознакомиться с различиями в принципах работы активного сетевого оборудования. Уяснить особенности взаимодействия сетевого и канального уровней на примере стека TCP/IP. Выяснить отличия форматов кадров Ethernet. Познакомиться с консольными утилитами диагностики и анализа сетевых соединений.

Ход работы:

Задание 1.8 На хосте c7-1 с помощью утилиты ping проверили доступность внешней сети, послав 5 эхо-запросов на сервер 8.8.8.8; также послали запросы на сервер 1.1.1.1

```
vboxuser@UBUNTU:~$ ping -c 5 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=63 time=20.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=63 time=17.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=63 time=16.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=63 time=10.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=63 time=12.0 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 10.587/15.270/20.031/3.492 ms
vboxuser@UBUNTU:~$ ping -c 5 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=63 time=25.6 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=63 time=20.3 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=63 time=19.0 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=63 time=18.5 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=63 time=19.0 ms

--- 1.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 18.470/20.483/25.619/2.634 ms
```

Задание 2.2a Написали команды, которые отправляют 10 пакетов на c7-1

```
vboxuser@c7-2:~$ ping -c 10 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=97.8 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.76 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=3.62 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.856 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=3.67 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=1.14 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=13.7 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.735 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=1.66 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=13.1 ms

--- 10.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9065ms
rtt min/avg/max/mdev = 0.735/13.803/97.753/28.365 ms
```

Задание 2.2b Написали команды, которые отправляют 10 пакетов с интервалом 10 секунд на машину c7-1

```
vboxuser@c7-2:~$ ping -c 10 -i 10  10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=159 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=114 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=47.1 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=736 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=178 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=341 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=16.2 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=106 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=350 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=33.4 ms

--- 10.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 90037ms
rtt min/avg/max/mdev = 16.205/208.063/736.078/207.973 ms
```

Задание 2.2c Написали команды, которые отправляет 5 пакетов размером 1500 байт на машину c7-1:

```
vboxuser@c7-2:~$ ping -c 5 -s 1500  10.0.0.1
PING 10.0.0.1 (10.0.0.1) 1500(1528) bytes of data.
1508 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=18.4 ms
1508 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=25.2 ms
1508 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=363 ms
1508 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=23.6 ms
1508 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=99.5 ms

--- 10.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4021ms
rtt min/avg/max/mdev = 18.378/106.008/363.402/132.133 ms
```

Задание 2.7 Написали команду, которая сохранит в файл расширенную статистику работы mtr при отправке 40 пакетов**.**

```
vboxuser@UBUNTU:~$ sudo mtr -r -c 40 itmo.ru > mtr-report-itmo
```

```
1 Start: 2022-11-11T23:45:47+0300
2 HOST: UBUNTU                     Loss%   Snt   Last   Avg  Best  Wrst StDev
3   1.|-- _gateway                  0.0%    40    0.7   0.6   0.2   1.0   0.2
4   2.|-- RT-GM-3                   0.0%    40    3.6   5.5   3.1  28.6   4.8
5   3.|-- spbr-bras34.sz.ip.rostele  0.0%    40   38.3  16.7   4.7 278.2  43.4
6   4.|-- 212.48.194.196            0.0%    40   10.7  26.4   4.9 267.7  60.6
7   5.|-- 87.226.183.89             2.5%    40   18.7  50.7  18.3 260.3  72.7
8   6.|-- broadband-90-154-109-162.  0.0%    40   20.4  40.9  16.9 200.4  49.7
9   7.|-- ???                      100.0    40    0.0   0.0   0.0   0.0   0.0
10  8.|-- ???                      100.0    40    0.0   0.0   0.0   0.0   0.0
11  9.|-- ???                      100.0    40    0.0   0.0   0.0   0.0   0.0
12 10.|-- ???                      100.0    40    0.0   0.0   0.0   0.0   0.0
13 11.|-- ???                      100.0    40    0.0   0.0   0.0   0.0   0.0
14 12.|-- ???                      100.0    40    0.0   0.0   0.0   0.0   0.0
15 13.|-- 51.250.54.78              0.0%    40   23.8  43.1  22.5 187.3  43.3
```

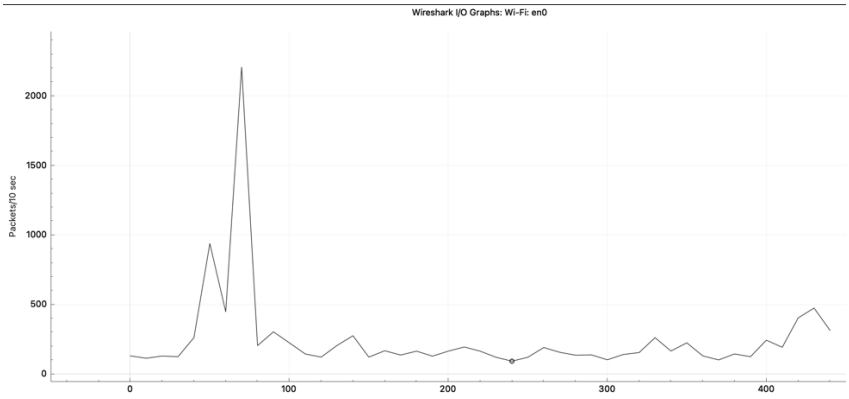
Задание 3.2a Используя инструментарий статистики, определили узел с максимальной активностью (по объему переданных данных):

IP Protocol Types	5693	0.0389	100%	8.8400	77.172
UDP	2042	0.0140	35.87%	7.5500	71.444
TCP	3647	0.0249	64.06%	8.8300	77.172

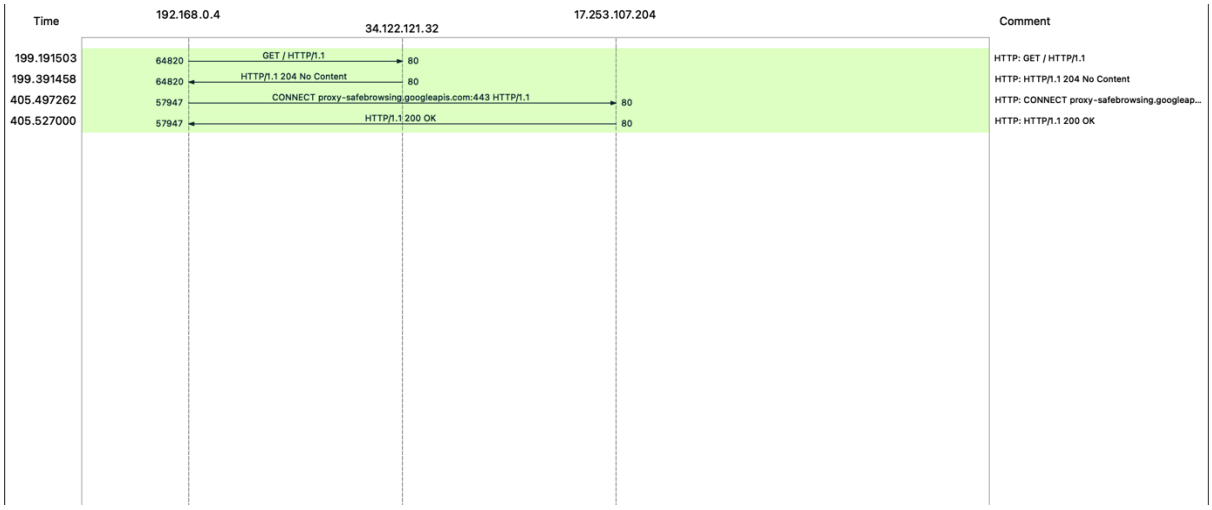
Задание 3.2c Используя инструментарий статистики, определили самый активный TCP-порт на хосте (по количеству переданных пакетов):

IP Protocol Types	5693	0.0389	100%	8.8400	77.172
UDP	2042	0.0140	35.87%	7.5500	71.444
TCP	3647	0.0249	64.06%	8.8300	77.172

Задание 3.2d Используя инструментарий статистики, построили на одной координатной сетке рафики интенсивности TCP и UDP трафика (пункт Io Graphs):

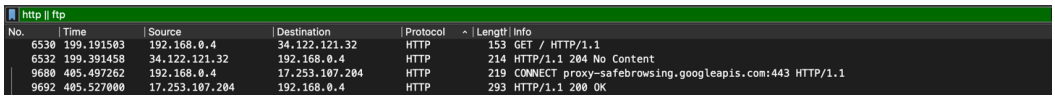


Задание 3.2e Используя инструментарий статистики, построили диаграмму связей только для пакетов, содержащих сообщения протокола HTTPS (пункт Flow Graph):



Задание 3.3a Написали фильтры, которые выделяют из общего числа пакеты, отбирающие сообщения протоколов HTTP и FTP и относящиеся

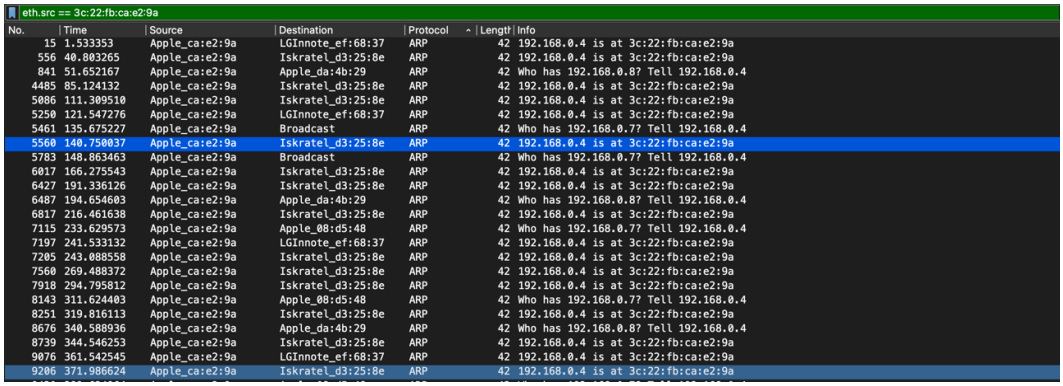
только к взаимодействию локальных клиентов и внешнего сервера. То есть в случае, если на вашем компьютере запущен и Web-броузер и Web-сервер, фильтр должен отбирать только трафик от и к Web-браузеру, игнорируя трафик от и к Web-серверу



http || ftp

No.	Time	Source	Destination	Protocol	Length	Info
6530	199.191503	192.168.0.4	34.122.121.32	HTTP	153	GET / HTTP/1.1
6532	199.391458	34.122.121.32	192.168.0.4	HTTP	214	HTTP/1.1 204 No Content
9680	405.497262	192.168.0.4	17.253.107.204	HTTP	219	CONNECT proxy-safebrowsing.googleapis.com:443 HTTP/1.1
9692	405.527000	17.253.107.204	192.168.0.4	HTTP	293	HTTP/1.1 200 OK

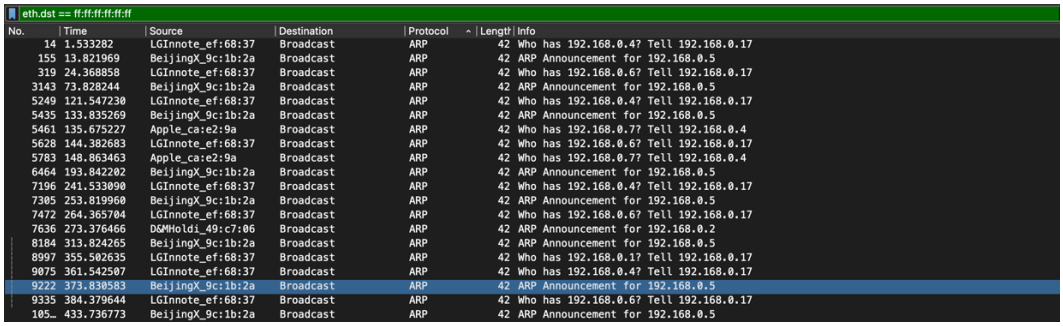
Задание 3.3b Написали фильтры, которые выделяют из общего числа все кадры Ethernet, отправленные с сетевого интерфейса хоста:



eth.src == 3c:22:fb:ca:e2:9a

No.	Time	Source	Destination	Protocol	Length	Info
15	1.533353	Apple_ca:e2:9a	LGInnote_ef:68:37	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
556	40.803265	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
841	51.652167	Apple_ca:e2:9a	Apple_da:4b:29	ARP	42	Who has 192.168.0.8? Tell 192.168.0.4
4485	85.124132	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
5086	111.309510	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
5250	121.547276	Apple_ca:e2:9a	LGInnote_ef:68:37	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
5461	135.675227	Apple_ca:e2:9a	Broadcast	ARP	42	Who has 192.168.0.7? Tell 192.168.0.4
5560	140.750037	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
5783	148.863463	Apple_ca:e2:9a	Broadcast	ARP	42	Who has 192.168.0.7? Tell 192.168.0.4
6017	166.275543	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
6427	191.336126	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
6487	194.654603	Apple_ca:e2:9a	Apple_da:4b:29	ARP	42	Who has 192.168.0.8? Tell 192.168.0.4
6817	216.461638	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
7115	233.629573	Apple_ca:e2:9a	Apple_08:d5:48	ARP	42	Who has 192.168.0.7? Tell 192.168.0.4
7197	241.533132	Apple_ca:e2:9a	LGInnote_ef:68:37	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
7205	243.088558	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
7560	269.488372	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
7918	294.795812	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
8143	311.624403	Apple_ca:e2:9a	Apple_08:d5:48	ARP	42	Who has 192.168.0.7? Tell 192.168.0.4
8251	319.816113	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
8676	340.588936	Apple_ca:e2:9a	Apple_da:4b:29	ARP	42	Who has 192.168.0.8? Tell 192.168.0.4
8739	344.546253	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
9076	361.542545	Apple_ca:e2:9a	LGInnote_ef:68:37	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a
9206	371.986624	Apple_ca:e2:9a	Iskratel_d3:25:8e	ARP	42	192.168.0.4 is at 3c:22:fb:ca:e2:9a

Задание 3.3c Написали фильтр, отбирающий только широковещательные сообщения. Определили назначение широковещательных рассылок протокола ARP, т.к. других протоколов не обнаружено. Цель широковещательных рассылок протокола ARP - определить MAC-адрес компьютера, которому нужно послать данные, зная его IP-адрес



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

No.	Time	Source	Destination	Protocol	Length	Info
14	1.533282	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.4? Tell 192.168.0.17
155	13.821969	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5
319	24.368858	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.6? Tell 192.168.0.17
3143	73.828244	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5
5249	121.547230	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.4? Tell 192.168.0.17
5435	133.835269	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5
5461	135.675227	Apple_ca:e2:9a	Broadcast	ARP	42	Who has 192.168.0.7? Tell 192.168.0.4
5628	144.382683	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.6? Tell 192.168.0.17
5783	148.863463	Apple_ca:e2:9a	Broadcast	ARP	42	Who has 192.168.0.7? Tell 192.168.0.4
6464	193.842202	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5
7196	241.533090	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.4? Tell 192.168.0.17
7305	253.819960	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5
7472	264.365704	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.6? Tell 192.168.0.17
7636	273.376466	D&MHoldi_49:c7:06	Broadcast	ARP	42	ARP Announcement for 192.168.0.2
8184	313.824265	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5
8997	355.502635	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.1? Tell 192.168.0.17
9075	361.542507	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.4? Tell 192.168.0.17
9222	373.830583	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5
9335	384.379644	LGInnote_ef:68:37	Broadcast	ARP	42	Who has 192.168.0.6? Tell 192.168.0.17
105…	433.736773	BeijingX_9c:1b:2a	Broadcast	ARP	42	ARP Announcement for 192.168.0.5

Задание 3.3d Определили адреса, на которые поступают данные кадры и пакеты для канального и сетевого уровня

Ответ: ff:ff:ff:ff:ff:ff

Задание 3.3e Т.к. все широковещательные рассылки у нас имеют одинаковый протокол - ARP - мы написали один общий фильтр: eth.dst == ff:ff:ff:ff:ff:ff && arp

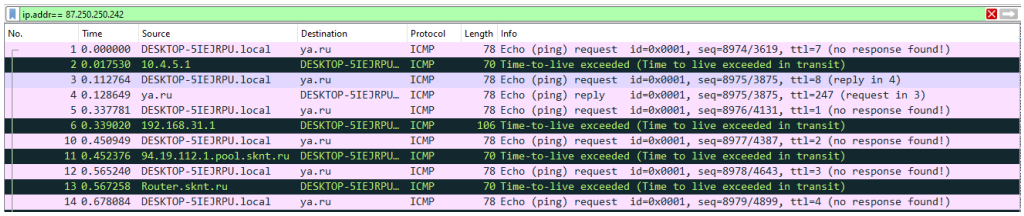
Задание 3.3f Маршрутизатор

Задание 3.4 В виртуальной машине с помощью утилиты mtr вывели статистику передачи трафика до хоста ya.ru, отправив 111 запросов и выводя на экран, как имена, так и ip адреса промежуточных устройств:

```
vboxuser@UBUNTU:~$ sudo mtr -r -c 111 ya.ru > mtr-ya.ru
```

```
Start: 2022-11-14T13:59:35+0300
HOST: UBUNTU                      Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- _gateway                   0.0%   111    1.2   1.0   0.2   3.7   0.4
  2.|-- RT-GM-3                    0.0%   111    4.5  14.5   3.3 324.1  48.5
  3.|-- spbr-bras34.sz.ip.rostele  0.0%   111    6.8  13.3   5.6 214.5  27.7
  4.|-- 212.48.194.196              0.0%   111    7.8  11.6   6.4 129.0  15.6
  5.|-- 188.254.2.0                 0.0%   111   10.1   9.6   6.7  18.8   1.4
  6.|-- 85.175.225.78               0.0%   111    9.8  16.5   6.4 366.4  45.2
  7.|-- sas-32z5-ae1.yndx.net       0.0%   111   25.7  35.3  23.4 269.7  36.9
  8.|-- ???                       100.0   111    0.0   0.0   0.0   0.0   0.0
  9.|-- ya.ru                       0.0%   111   24.0  24.4  20.6  46.0   2.6
```

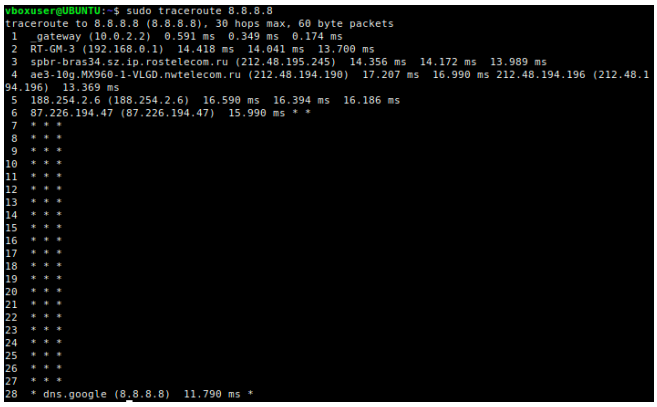
Задание 3.5 В Wireshark написали фильтр, отбирающий сетевые сообщения из п. 4. Определили, что проверка доступности осуществляется с помощью протокола ICMP.



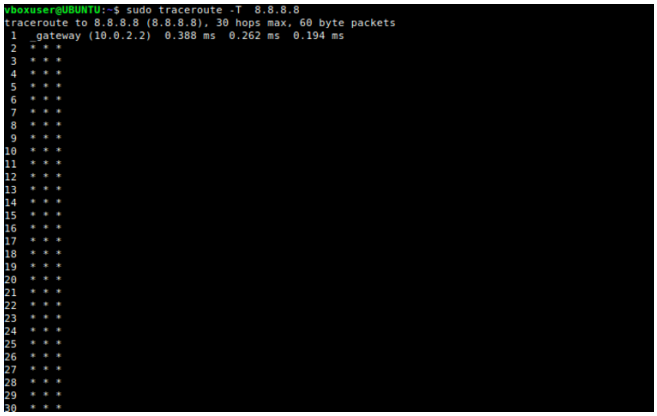
Задание 4.2a На машине c7-1 написали команды traceroute, которые определяют маршрут до хоста 8.8.8.8 с помощью ICMP:

```
vboxuser@UBUNTU:~$ sudo traceroute -I 8.8.8.8
[sudo] password for vboxuser:
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.891 ms  0.824 ms  0.794 ms
 2  RT-GM-3 (192.168.0.1)  13.946 ms  13.913 ms  13.866 ms
 3  spbr-bras34.sz.ip.rostelecom.ru (212.48.195.245)  14.139 ms  14.443
 ms  14.110 ms
 4  ae3-10g.MX960-1-VLGD.nwtelecom.ru (212.48.194.190)  14.043 ms  14.4
11 ms  14.368 ms
 5  188.254.2.4 (188.254.2.4)  19.099 ms  18.985 ms  22.933 ms
 6  87.226.194.47 (87.226.194.47)  15.495 ms  6.564 ms  6.733 ms
 7  74.125.244.180 (74.125.244.180)  7.563 ms  8.140 ms  8.991 ms
 8  142.251.61.219 (142.251.61.219)  13.403 ms  13.362 ms  13.215 ms
 9  172.253.79.113 (172.253.79.113)  11.954 ms  11.994 ms  12.879 ms
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  dns.google (8.8.8.8)  12.491 ms  13.400 ms  15.761 ms
```

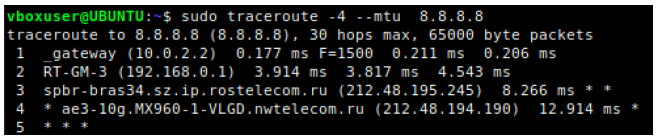
Задание 4.2b На машине c7-1 написали команды traceroute, которые определяют маршрут до хоста 8.8.8.8 с помощью UDP:



Задание 4.2c На машине c7-1 написали команды traceroute, которые определяют маршрут до хоста 8.8.8.8 с помощью TCP:

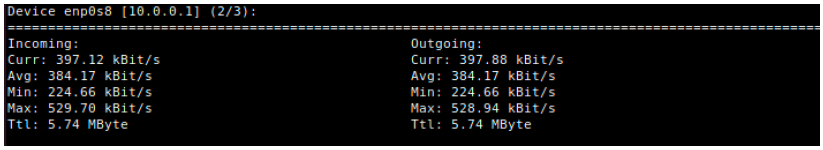


Задание 4.2d На машине c7-1 написали команды traceroute, которые позволяют определить используется ли по маршруту фрагментация IPv4:

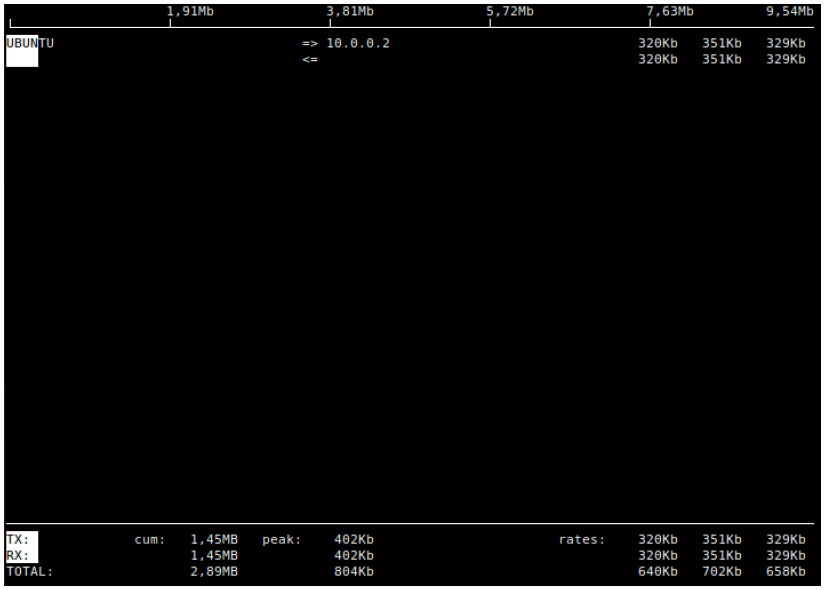


Задание 5 На хосте c7-1 с помощью утилиты **nload, iftop, bmon** получили данные о загрузке интерфейса, на который отправляет трафик хост c7-2:

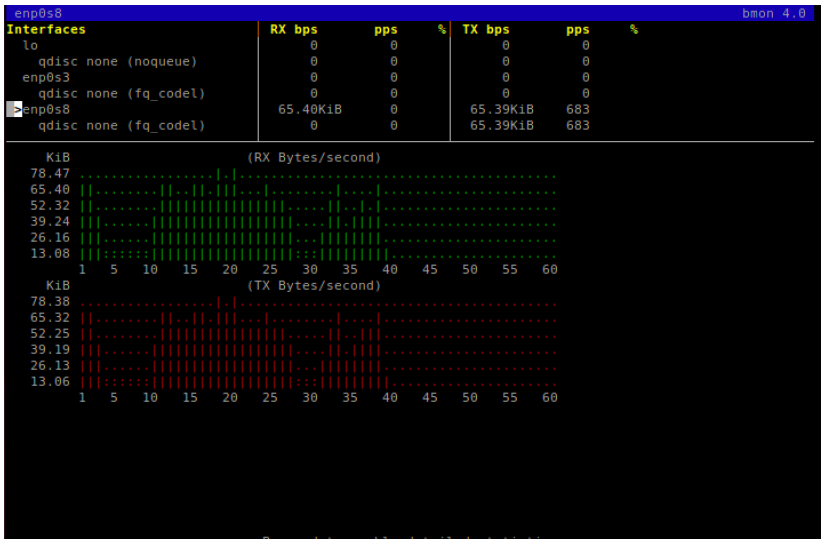
nload



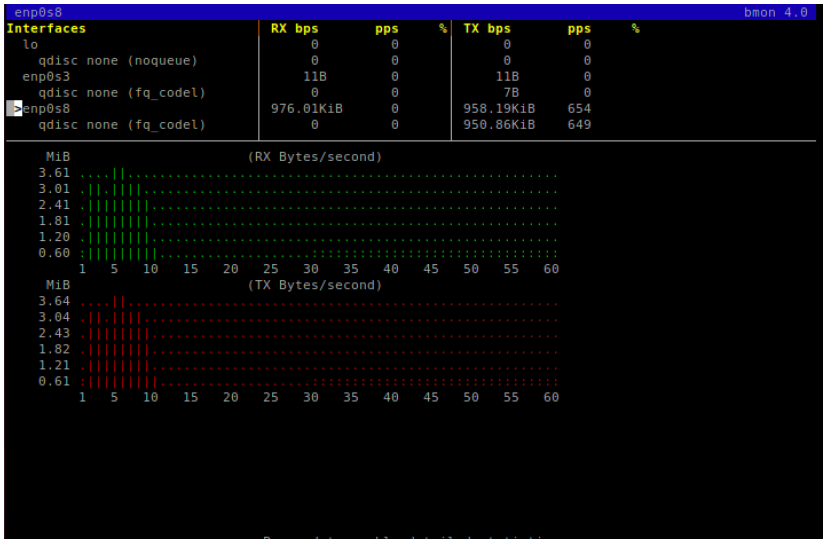
Iftop



bmon пакет 100



bmon пакет 60100



Задание 6

На хосте c7-1 запустили vnstat и поставили на мониторинг интерфейс enp0s8, через который машина c7-1 подключена к c7-2. С хоста c7-2 запустили отправку запросов утилитой ping в режиме flood, так чтобы работа утилиты прекратилась после отправки 500 пакетов.

Вывели статистику собранного трафика:

```
vboxuser@UBUNTU:~$ vnstat -l -i enp0s8
Monitoring enp0s8...    (press CTRL-C to stop)

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


 enp0s8  /  traffic statistics

                           rx         |       tx
--------------------------------------+------------------
  bytes                    47,97 KiB  |       47,97 KiB
--------------------------------------+------------------
          max           68,21 kbit/s  |    68,21 kbit/s
      average            8,36 kbit/s  |     8,36 kbit/s
          min               0 bit/s   |          0 bit/s
--------------------------------------+------------------
  packets                       502   |              502
--------------------------------------+------------------
          max                87 p/s   |           87 p/s
      average                10 p/s   |           10 p/s
          min                 0 p/s   |            0 p/s
--------------------------------------+------------------
  time                   47 seconds
```

Задание 7.2 Используя утилиту **lsof** на c7-1 вывели все активные порты:

```
vboxuser@UBUNTU:~$ sudo lsof -i -P
COMMAND     PID            USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r   540 systemd-resolve   13u  IPv4  17263      0t0  UDP localhost:53
systemd-r   540 systemd-resolve   14u  IPv4  17264      0t0  TCP localhost:53 (LISTEN)
avahi-dae   654           avahi   12u  IPv4  21007      0t0  UDP *:5353
avahi-dae   654           avahi   13u  IPv6  21008      0t0  UDP *:5353
avahi-dae   654           avahi   14u  IPv4  21009      0t0  UDP *:52770
avahi-dae   654           avahi   15u  IPv6  21010      0t0  UDP *:46084
NetworkMa   661            root   28u  IPv4  16155      0t0  UDP UBUNTU:68->_gateway:67
cupsd       772            root    6u  IPv6  16081      0t0  TCP ip6-localhost:631 (LISTEN)
cupsd       772            root    7u  IPv4  16082      0t0  TCP localhost:631 (LISTEN)
cups-brow   858            root    7u  IPv4  18122      0t0  UDP *:631
firefox    2300        vboxuser    3u  IPv4 133094      0t0  TCP UBUNTU:33702->ec2-54-148-69-31.us-west-2.compute.amaz
onaws.com:443 (ESTABLISHED)
firefox    2300        vboxuser   30u  IPv4  31299      0t0  TCP UBUNTU:59806->104.16.249.249:443 (ESTABLISHED)
sshd      29495            root    3u  IPv4 376620      0t0  TCP *:22 (LISTEN)
sshd      29495            root    4u  IPv6 376622      0t0  TCP *:22 (LISTEN)
sshd      29746            root    4u  IPv4 381802      0t0  TCP UBUNTU:22->10.0.0.2:58212 (ESTABLISHED)
sshd      29824        vboxuser    4u  IPv4 381802      0t0  TCP UBUNTU:22->10.0.0.2:58212 (ESTABLISHED)
```

Задание 7.3 Используя утилиту **netstat** вывели все установленные соединения:

```
vboxuser@UBUNTU:~$ netstat -a -t
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 UBUNTU:59806            104.16.249.249:https    ESTABLISHED
tcp        0      0 UBUNTU:46612            banjo.canonical.co:http TIME_WAIT
tcp        0      0 UBUNTU:57826            ec2-52-40-138-9.u:https ESTABLISHED
tcp        0      0 UBUNTU:33702            ec2-54-148-69-31.:https ESTABLISHED
tcp        0      0 UBUNTU:ssh              10.0.0.2:58212          ESTABLISHED
tcp        0      0 UBUNTU:45622            239.237.117.34.bc:https TIME_WAIT
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
```

Задание 7.4 С помощью утилиты netstat вывели список IP-адресов и количество подключений с них к c7-1 через порт 22, который по умолчанию используется SSH-протоколом. Также мы использовали утилиту grep:

```
vboxuser@UBUNTU:~$ netstat -n | grep ':22'
tcp        0      0 10.0.0.1:22             10.0.0.2:58212          ESTABLISHED
```

Задание 7.8 С помощью утилиты NetHogs определили PID процесса – 2300 и среднюю скорость передачи данных до sshd - 0.030 KB/sec

```
NetHogs version 0.8.6-3

    PID USER     PROGRAM                                DEV        SENT      RECEIVED
   2300 vboxus.. /snap/firefox/2067/usr/lib/firef..    enp0s3     0.029       0.030 KB/sec
  29824 vboxus.. sshd: vboxuser@pts/1                  enp0s8     0.000       0.000 KB/sec
      ? root     unknown TCP                                     0.000       0.000 KB/sec

  TOTAL                                                          0.029       0.030 KB/sec
```

Задание 7.9 С помощью команды tcpdump на c7-1 настроили вывод на экран содержимого пакетов от Windows-хоста по протоколу ssh:

```
vboxuser@UBUNTU:~$ sudo tcpdump tcp port 22 -A
```

Ответы на вопросы

1. mtr работает по протоколу ICMP, потому что mtr — это инструмент, сочетающий в себе traceroute и ping.

2. Поля mtr

- Имя хоста или IP-адрес конкретного прыжка в сети (Host).
- Процент пакетов, потерянных этим прыжком (Loss).
- Количество пакетов, отправленных во время прыжка (Snt).
- Время в пути туда и обратно (Last).
- Среднее, лучшее, худшее и стандартное отклонение времени с момента запуска mtr (Avg, Best, Wrst, StDev).

3. Типы кадров Ethernet:

- Кадр 802.3/LLC (или кадр Novell 802.2);
- Кадр Raw 802.3 (или кадр Novell 802.3);
- Кадр Ethernet DIX (или кадр Ethernet II);
- Кадр Ethernet SNAP.

У них отличаются назначение полей, значение MTU и формат.

4. В анализируемой сети используется тип Ethernet II. Он наиболее популярный.

5. Используя описание источников и адреса назначения, а также используемые при передаче протоколы.

6. На сетевом уровне используются широковещательные адреса, вид которых зависит от протокола.

7. ff:ff:ff:ff:ff:ff

8. Для того, чтобы целевой MAC-адрес по IP-адресу

9. **sudo traceroute I** использует icmp echo request, а **sudo traceroute T** отправляет tcp request

10. Утилита **bmon** понравилась больше всего, тк ее вывод наиболее понятен для нас.

11. Загрузка интерфейса в Части 5. п. 3 меняется линейно.
Чем больше размер пакет, тем больше загрузка.

12. на сетевом уровне OSI.

13. **ip neigh show, ip neigh delete**
Очистка arp-кэша может потребоваться в случае, если в работе сети появились такие проблемы, как ошибки при загрузке определенных сайтов или отсутствие пинга некоторых IP-адресов.

14. **sudo tcpdump host 192.168.0.254 ‘tcp port 80 or udp’**