

МИНОБРАЗОВАНИЯ 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

ОТЧЕТ

По лабораторной работе №1

РУКОВОДИТЕЛЬ:

(подпись)

Гай В.Е.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Расторопов Д.С.
(фамилия, и.,о.)

19-В-2
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород 2022

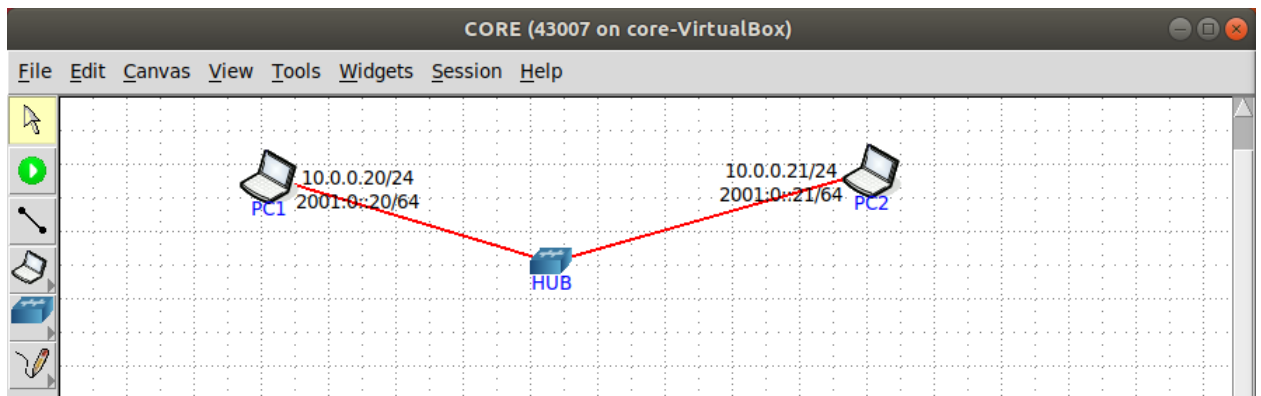
Задание:

Работа с анализатором протоколов tcpdump

1. Запустить tcpdump в режиме захвата всех пакетов, проходящих по сети. Количество захватываемых пакетов ограничить 10. Результаты протоколировать в файл.
2. Запустить tcpdump в режиме перехвата широковещательного трафика (фильтр по MAC-адресу). Количество захватываемых пакетов ограничить 5. Включить распечатку пакета в шестнадцатеричной системе (включая заголовок канального уровня).
3. Запустить tcpdump так, чтобы он перехватывал только пакеты протокола ICMP, отправленные на определенный IP-адрес. При этом включить распечатку пакета в шестнадцатеричной системе и ASCII-формате (включая заголовок канального уровня). Количество захватываемых пакетов ограничить 3. Для генерирования пакетов воспользоваться утилитой ping.
4. Запустить tcpdump в режиме сохранения данных в двоичном режиме так, чтобы он перехватывал пакеты, созданные утилитой traceroute для определения маршрута к заданному в варианте узлу. Включить распечатку пакета в шестнадцатеричной системе и ASCII-формате (включая заголовок канального уровня). Количество захватываемых пакетов ограничить 7. Результат работы программы писать в файл.
5. Прочитать программой tcpdump созданный в предыдущем пункте файл.
6. Придумать три задания для фильтрации пакетов на основе протоколов ARP, TCP, UDP, ICMP

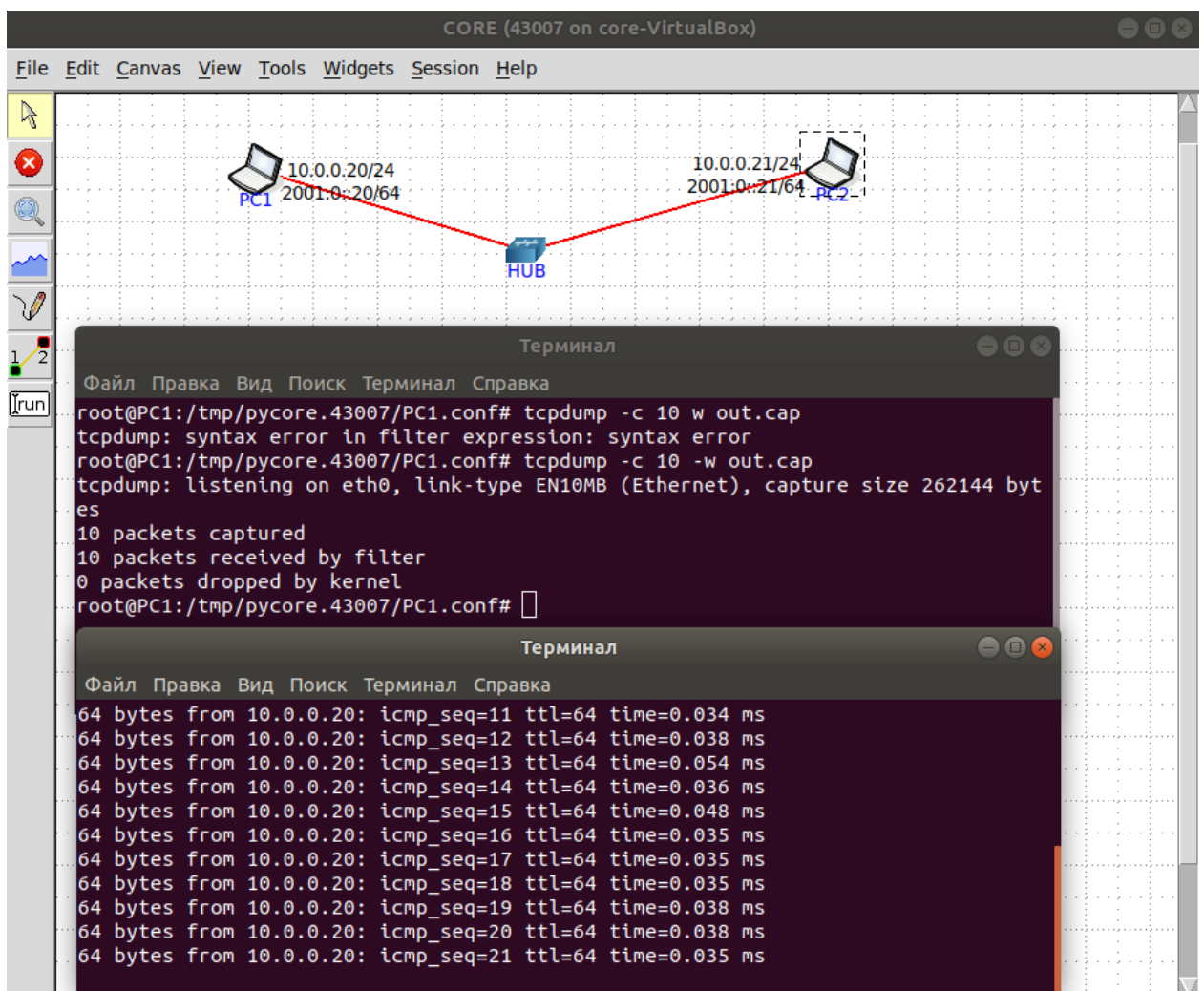
Работа с анализатором протоколов wireshark

1. Захватить 5-7 пакетов широковещательного трафика (фильтр по IP-адресу). Результат сохранить в текстовый файл.
2. Захватить 3-4 пакета ICMP, полученных от определенного узла. Для генерирования пакетов воспользоваться утилитой ping. Результат сохранить в текстовый файл.
3. Перехватить пакеты, созданные утилитой traceroute для определения маршрута к заданному в варианте узлу. По результатам построить диаграмму Flow Graph. Диаграмму сохранить либо в виде текстового файла либо в виде изображения.
4. Прочитать файл, созданный программой tcpdump. Сравнить с тем, что было получено утилитой wireshark.



Работа с анализатором протоколов tcpdump

1. Запустить tcpdump в режиме захвата всех пакетов, проходящих по сети. Количество захватываемых пакетов ограничить 10. Результаты протоколировать в файл.



2. Запустить tcpdump в режиме перехвата широковещательного трафика (фильтр по MAC-адресу). Количество захватываемых пакетов ограничить 5. Включить распечатку пакета в шестнадцатеричной системе (включая заголовок канального уровня).

CORE (43007 on core-VirtualBox)

File Edit Canvas View Tools Widgets Session Help

10.0.0.20/24
2001:0::20/64 PC1

10.0.0.21/24
2001:0::21/64 PC2

HUB

Терминал

Файл Правка Вид Поиск Терминал Справка

```
root@PC2:/tmp/pycore.43007/PC2.conf# tcpdump -c 5 -x 'ether dst ff:ff:ff:ff:ff:ff'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:49:24.465432 ARP, Request who-has PC2 (Broadcast) tell 10.0.0.20, length 46
  0x0000: 0001 0800 0604 0001 0000 00aa 0000 0a00
  0x0010: 0014 ffff ffff ffff 0a00 0015 0000 0000
  0x0020: 0000 0000 0000 0000 0000 0000 0000
17:50:07.329409 ARP, Request who-has PC2 tell 10.0.0.20, length 46
  0x0000: 0001 0800 0604 0001 ffff ffff ffff 0a00
  0x0010: 0014 0000 0000 0000 0a00 0015 0000 0000
  0x0020: 0000 0000 0000 0000 0000 0000 0000
17:50:07.329425 ARP, Reply PC2 is-at 00:00:00:aa:00:01 (oui Ethernet), length 28
  0x0000: 0001 0800 0604 0002 0000 00aa 0001 0a00
  0x0010: 0015 ffff ffff ffff 0a00 0014
17:50:53.695944 ARP, Request who-has PC2 tell 10.0.0.20, length 46
  0x0000: 0001 0800 0604 0001 0000 00aa 0000 0a00
  0x0010: 0014 0000 0000 0000 0a00 0015 0000 0000
  0x0020: 0000 0000 0000 0000 0000 0000 0000
17:51:12.745839 ARP, Request who-has PC2 tell 10.0.0.20, length 46
  0x0000: 0001 0800 0604 0001 0000 00aa 0000 0a00
  0x0010: 0014 0000 0000 0000 0a00 0015 0000 0000
  0x0020: 0000 0000 0000 0000 0000 0000 0000
5 packets captured
5 packets received by filter
0 packets dropped by kernel
root@PC2:/tmp/pycore.43007/PC2.conf#
```

Canvas1

PackETH - ethernet packet generator (на PC1)

File Help

Builder Gen-b Gen-s Pcap Load Save Default Default Interface Send Stop

3. Запустить tcpdump так, чтобы он перехватывал только пакеты протокола ICMP, отправленные на определенный IP-адрес. При этом включить распечатку пакета в шестнадцатеричной системе и ASCII-формате (включая заголовок канального уровня). Количество захватываемых пакетов ограничить 3. Для генерирования пакетов воспользоваться утилитой ping.

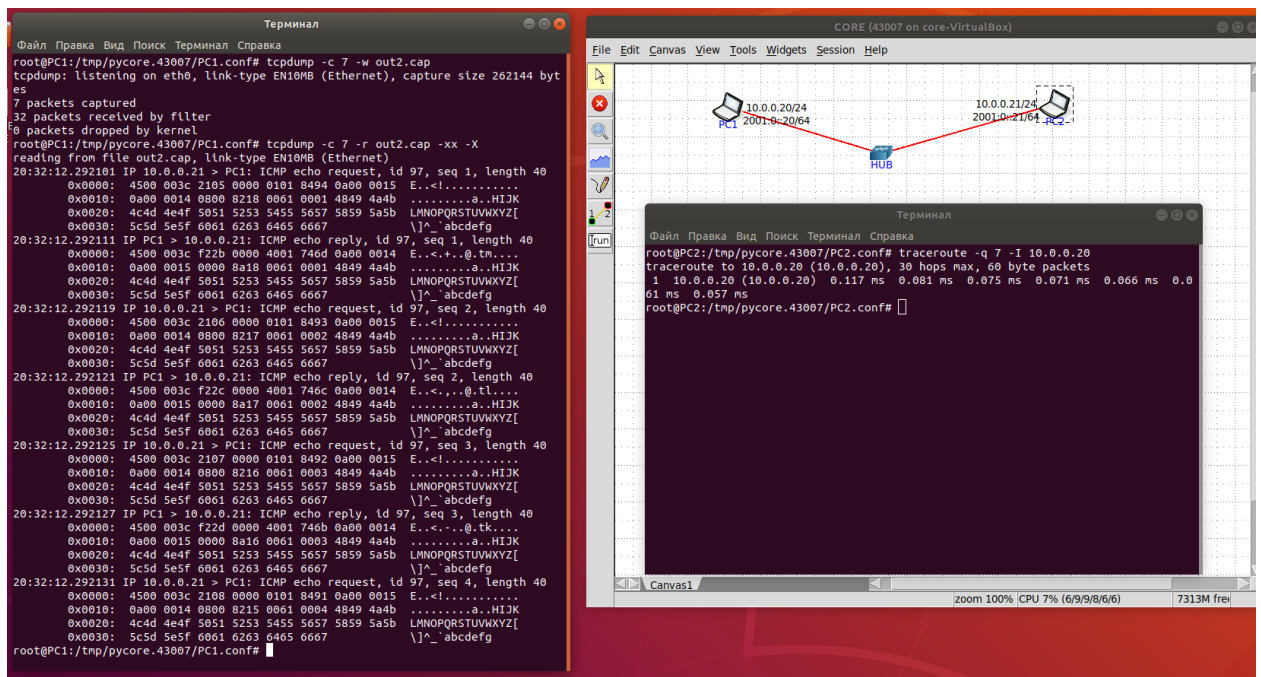
```

CORE (43007 on core-VirtualBox)
File Edit Canvas View Tools Widgets Session Help

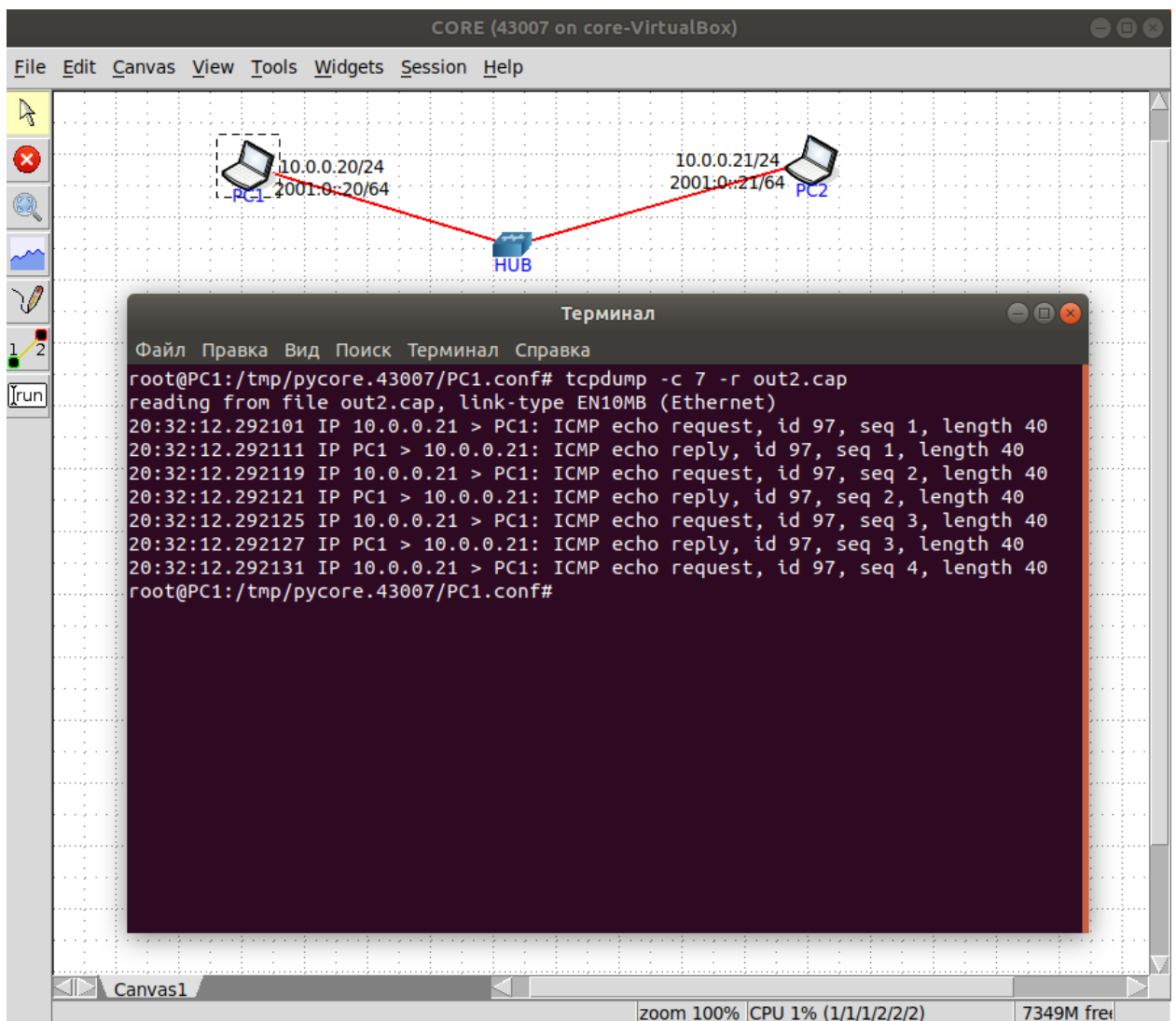
Терминал
Файл Правка Вид Поиск Терминал Справка
root@PC2:/tmp/pycore.43007/PC2.conf# ping 10.0.0.20
PING 10.0.0.20 (10.0.0.20) 56(84) bytes of data.
64 bytes from 10.0.0.20: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 10.0.0.20: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 10.0.0.20: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 10.0.0.20: icmp_seq=4 ttl=64 time=0.033 ms
64 bytes from 10.0.0.20: icmp_seq=5 ttl=64 time=0.038 ms
64 bytes from 10.0.0.20: icmp_seq=6 ttl=64 time=0.043 ms
64 bytes from 10.0.0.20: icmp_seq=7 ttl=64 time=0.048 ms
64 bytes from 10.0.0.20: icmp_seq=8 ttl=64 time=0.039 ms
64 bytes from 10.0.0.20: icmp_seq=9 ttl=64 time=0.032 ms

Терминал
Файл Правка Вид Поиск Терминал Справка
<PC1.conf# tcpdump -c 3 -XX 'dst host 10.0.0.21 and ip proto \icmp'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:54:20.435292 IP PC1 > 10.0.0.21: ICMP echo reply, id 62, seq 1, length 64
  0x0000:  0000 00aa 0001 0000 00aa 0000 0800 4500  .....E.
  0x0010:  0054 c70a 0000 4001 9f76 0a00 0014 0a00  .T....@.v.....
  0x0020:  0015 0000 43a0 003e 0001 1c49 9960 0000  ....C.>...I.`..
  0x0030:  0000 41a4 0600 0000 0000 1011 1213 1415  ..A.....
  0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  .....! "#$%
  0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
  0x0060:  3637                                     67
17:54:21.482967 IP PC1 > 10.0.0.21: ICMP echo reply, id 62, seq 2, length 64
  0x0000:  0000 00aa 0001 0000 00aa 0000 0800 4500  .....E.
  0x0010:  0054 c754 0000 4001 9f2c 0a00 0014 0a00  .T.T..@.,.....
  0x0020:  0015 0000 23e5 003e 0002 1d49 9960 0000  ....#..>...I.`..
  0x0030:  0000 5f5e 0700 0000 0000 1011 1213 1415  .._.....
  0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  .....! "#$%
  0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
  0x0060:  3637                                     67
17:54:22.486566 IP PC1 > 10.0.0.21: ICMP echo reply, id 62, seq 3, length 64
  0x0000:  0000 00aa 0001 0000 00aa 0000 0800 4500  .....E.
  0x0010:  0054 c7c0 0000 4001 9ec0 0a00 0014 0a00  .T....@.....
  0x0020:  0015 0000 05d6 003e 0003 1e49 9960 0000  .....>...I.`..
  0x0030:  0000 7c6c 0700 0000 0000 1011 1213 1415  ..|l.....
  0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  .....! "#$%
  0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
  0x0060:  3637                                     67
3 packets captured
3 packets received by filter
  
```

4. Запустить tcpdump в режиме сохранения данных в двоичном режиме так, чтобы он перехватывал пакеты, созданные утилитой traceroute для определения маршрута к заданному в варианте узлу. Включить распечатку пакета в шестнадцатеричной системе и ASCII-формате (включая заголовок канального уровня). Количество захватываемых пакетов ограничить 7. Результат работы программы писать в файл.

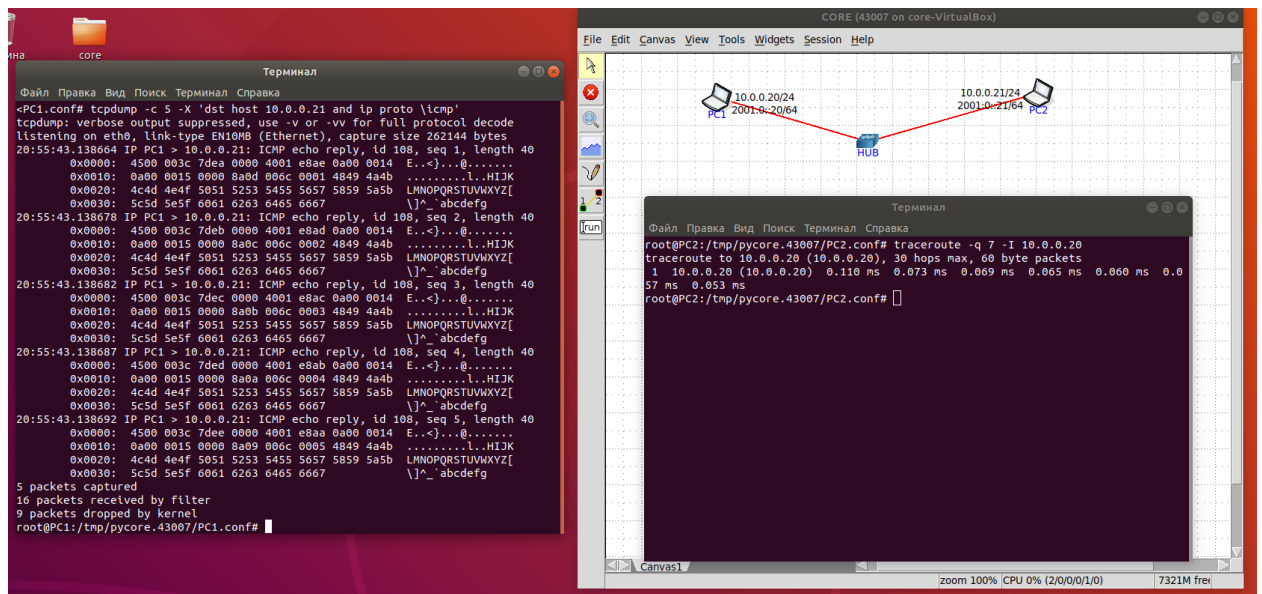


5. Прочсть программой tcpdump созданный в предыдущем пункте файл.

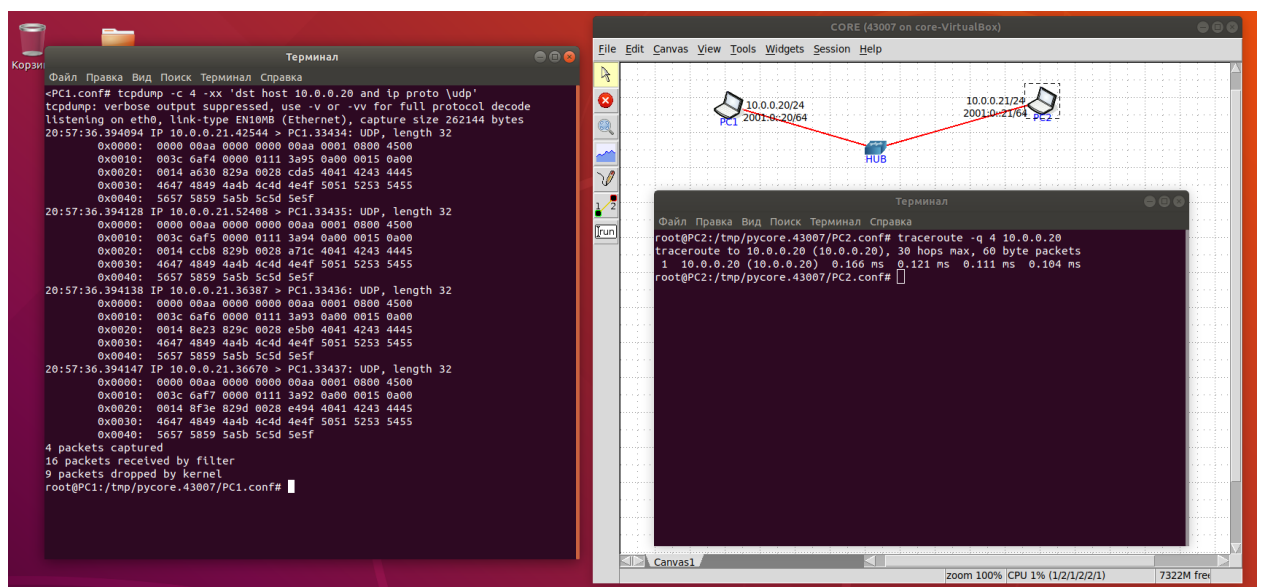


6. Придумать три задания для фильтрации пакетов на основе протоколов ARP, TCP, UDP, ICMP

1) Запустить `tcpdump` так, чтобы он перехватывал только пакеты протоколов ICMP, отправленные на определенный IP-адрес, чтобы он перехватывал пакеты, созданные утилитой `tracert`. Включить распечатку пакета в шестнадцатеричной системе и ASCII-формате (не включая заголовок канального уровня). Количество захватываемых пакетов ограничить 5.



2) Запустить `tcpdump` так, чтобы он перехватывал только пакеты протокола UDP, отправленные на определенный IP-адрес, чтобы он перехватывал пакеты, созданные утилитой `tracert`. Включить распечатку пакета в шестнадцатеричной системе (включая заголовок канального уровня). Количество захватываемых пакетов ограничить 4.



3) Запустить `tcpdump` так, чтобы он перехватывал только пакеты протокола ARP, отправленные на определенный IP-адрес, чтобы он перехватывал пакеты, созданные утилитой `ping`. Включить распечатку пакета в шестнадцатеричной системе (включая заголовок канального уровня). Количество захватываемых пакетов ограничить 4.

```
root@PC1:/tmp/pycore.43007/PC1.conf# tcpdump -c 4 -xx 'ether proto arp'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:59:13.088948 ARP, Request who-has 10.0.0.21 tell PC1, length 28
  0x0000: 0000 00aa 0001 0000 00aa 0000 0806 0001
  0x0010: 0000 0004 0001 0000 00aa 0000 0a00 0014
  0x0020: 0000 0000 0000 0a00 0015
  20:59:13.088985 ARP, Request who-has PC1 tell 10.0.0.21, length 28
  0x0000: 0000 00aa 0000 0000 00aa 0001 0806 0001
  0x0010: 0000 0004 0001 0000 00aa 0001 0a00 0015
  0x0020: 0000 0000 0000 0a00 0014
  20:59:13.088994 ARP, Reply PC1 is-at 00:00:00:aa:00:00 (out Ethernet), length 28
  0x0000: 0000 00aa 0001 0000 00aa 0000 0806 0001
  0x0010: 0000 0004 0002 0000 00aa 0000 0a00 0014
  0x0020: 0000 00aa 0001 0a00 0015
  20:59:13.088995 ARP, Reply 10.0.0.21 is-at 00:00:00:aa:00:01 (out Ethernet), length 28
  0x0000: 0000 00aa 0000 0000 00aa 0001 0806 0001
  0x0010: 0000 0004 0002 0000 00aa 0001 0a00 0015
  0x0020: 0000 00aa 0000 0a00 0014
  4 packets captured
  4 packets received by filter
  0 packets dropped by kernel
root@PC1:/tmp/pycore.43007/PC1.conf#
```

```
root@PC2:/tmp/pycore.43007/PC2.conf# ping 10.0.0.20
PING 10.0.0.20 (10.0.0.20) 56(84) bytes of data:
 64 bytes from 10.0.0.20: icmp_seq=1 ttl=64 time=0.033 ms
 64 bytes from 10.0.0.20: icmp_seq=2 ttl=64 time=0.033 ms
 64 bytes from 10.0.0.20: icmp_seq=3 ttl=64 time=0.037 ms
 64 bytes from 10.0.0.20: icmp_seq=4 ttl=64 time=0.040 ms
 64 bytes from 10.0.0.20: icmp_seq=5 ttl=64 time=0.045 ms
 64 bytes from 10.0.0.20: icmp_seq=6 ttl=64 time=0.062 ms
 64 bytes from 10.0.0.20: icmp_seq=7 ttl=64 time=0.041 ms
 64 bytes from 10.0.0.20: icmp_seq=8 ttl=64 time=0.052 ms
 64 bytes from 10.0.0.20: icmp_seq=9 ttl=64 time=0.038 ms
 64 bytes from 10.0.0.20: icmp_seq=10 ttl=64 time=0.067 ms
 64 bytes from 10.0.0.20: icmp_seq=11 ttl=64 time=0.047 ms
^C
--- 10.0.0.20 ping statistics ---
 11 packets transmitted, 11 received, 0% packet loss, time 10221ms
 rtt min/avg/max/ndev = 0.033/0.045/0.067/0.010 ms
root@PC2:/tmp/pycore.43007/PC2.conf#
```

Работа с анализатором протоколов wireshark

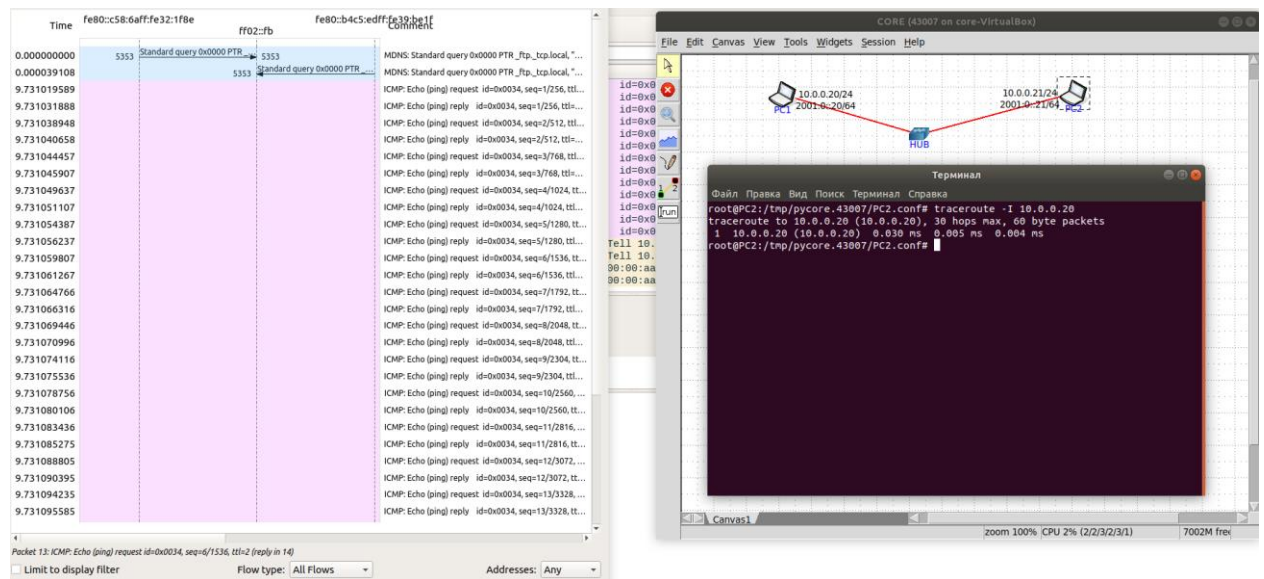
1. Захватить 5-7 пакетов широковещательного трафика (фильтр по IP-адресу). Результат сохранить в текстовый файл.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.21	10.0.0.20	ICMP	98	Echo (ping) request id=0x0029, seq=1/256, ttl=64 (reply in 2)
2	0.000011070	10.0.0.20	10.0.0.21	ICMP	98	Echo (ping) reply id=0x0029, seq=1/256, ttl=64 (request in 1)
3	1.011834024	10.0.0.21	10.0.0.20	ICMP	98	Echo (ping) request id=0x0029, seq=2/512, ttl=64 (reply in 4)
4	1.011848823	10.0.0.20	10.0.0.21	ICMP	98	Echo (ping) reply id=0x0029, seq=2/512, ttl=64 (request in 3)
5	2.036482604	10.0.0.21	10.0.0.20	ICMP	98	Echo (ping) request id=0x0029, seq=3/768, ttl=64 (reply in 6)
6	2.036498983	10.0.0.20	10.0.0.21	ICMP	98	Echo (ping) reply id=0x0029, seq=3/768, ttl=64 (request in 5)

2. Захватить 3-4 пакета ICMP, полученных от определенного узла. Для генерирования пакетов воспользоваться утилитой `ping`. Результат сохранить в текстовый файл.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.21	10.0.0.20	ICMP	98	Echo (ping) request id=0x0029, seq=1/256, ttl=64 (reply in 2)
2	0.000011070	10.0.0.20	10.0.0.21	ICMP	98	Echo (ping) reply id=0x0029, seq=1/256, ttl=64 (request in 1)
3	1.011834024	10.0.0.21	10.0.0.20	ICMP	98	Echo (ping) request id=0x0029, seq=2/512, ttl=64 (reply in 4)

3. Перехватить пакеты, созданные утилитой traceroute для определения маршрута к заданному в варианте узлу. По результатам построить диаграмму Flow Graph. Диаграмму сохранить либо в виде текстового файла либо в виде изображения.



4. Прочитать файл, созданный программой tcpdump. Сравнить с тем, что было получено утилитой wireshark.

