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 №5

по дисциплине

Сети и телекоммуникации

РУКОВОДИТЕЛЬ:

_____ Гай В. Е.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

_____ Пигасин Д. А.

(подпись) (фамилия, и.,о.)

18-АС

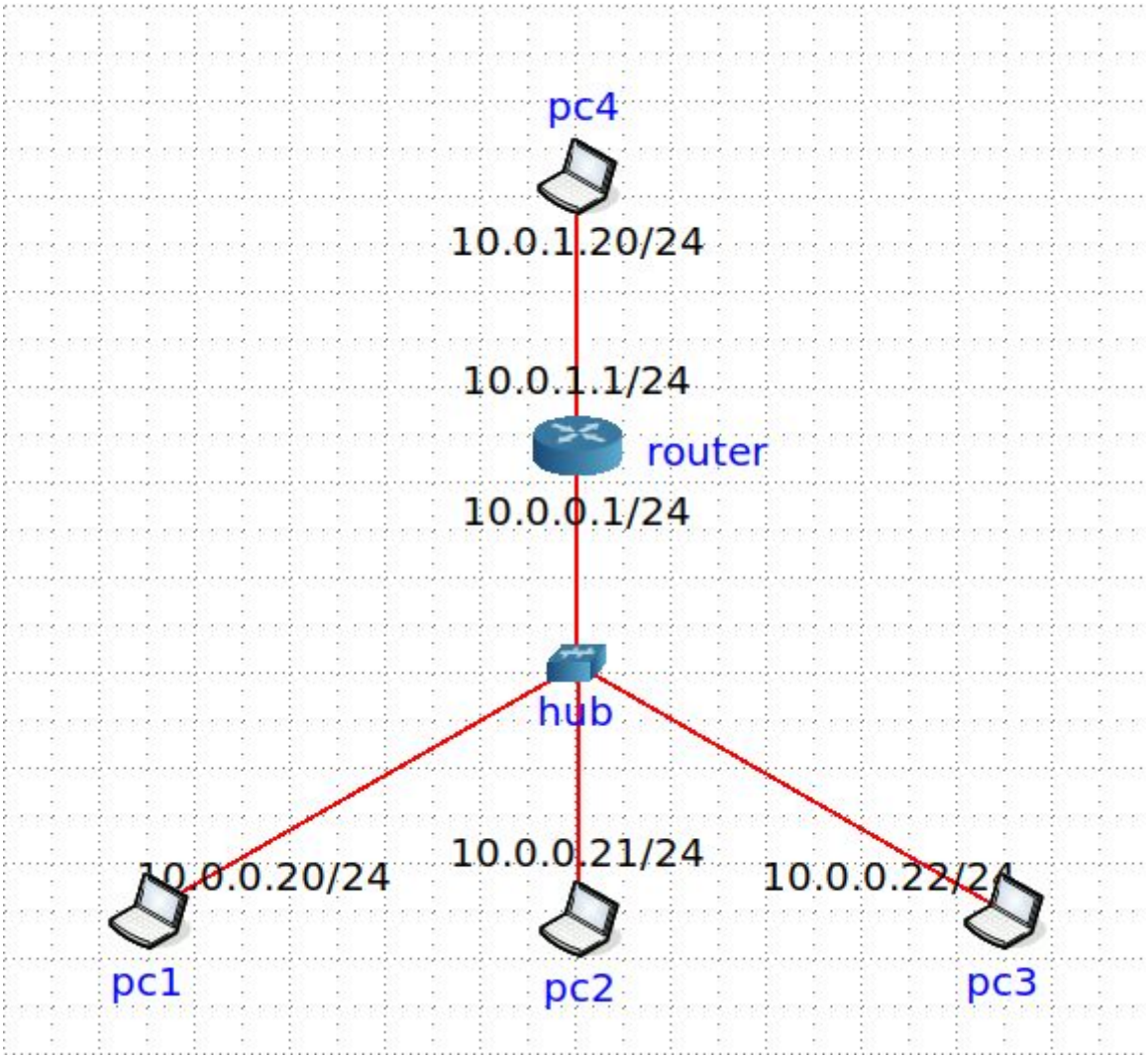
(шифр группы)

Работа защищена «___» _____

С оценкой _____

Нижний Новгород 2021

Схема сети



Работа с анализатором протоколов tcpdump

Для фильтрации “мусорных” пакетов, не имеющих отношения к лабораторной работе, также применялся фильтр “not dst host 224.0.0.5 and not ether proto \ip6”, который не указывается в заданиях, чтобы акцентировать внимание на выполнении поставленной задачи.

```
root@pc1:/tmp/pycore.46601/pc1.conf# tcpdump -l
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:57:14.512864 IP6 fe80::200:ff:feaa:4 > ff02::5: OSPFv3, Hello, length 36
17:57:14.526449 IP _gateway > 224.0.0.5: OSPFv2, Hello, length 44
17:57:16.527048 IP _gateway > 224.0.0.5: OSPFv2, Hello, length 44
17:57:16.891004 IP6 pc1 > ip6-allrouters: ICMP6, router solicitation, length 16
17:57:16.891112 IP6 fe80::58f7:f4ff:fe99:89f > ip6-allrouters: ICMP6, router solicitation, length 16
17:57:17.915012 IP6 fe80::789d:e5ff:fe6d:5d8d > ip6-allrouters: ICMP6, router solicitation, length 16
17:57:18.426845 IP6 fe80::200:ff:feaa:1 > ip6-allrouters: ICMP6, router solicitation, length 16
17:57:18.528340 IP _gateway > 224.0.0.5: OSPFv2, Hello, length 44
17:57:19.681629 IP6 fe80::58f7:f4ff:fe99:89f.mdns > ff02::fb.mdns: 0 [9q] PTR (QM)? _nfs._tcp.local. PTR (QM)? _ipp._tcp.local. PTR (QM)? _ipps._tcp.loca
l. PTR (QM)? _ftp._tcp.local. PTR (QM)? _webdav._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _smb._tcp.local. PTR
 (QM)? _afpovertcp._tcp.local. (141)
17:57:19.739870 IP6 fe80::789d:e5ff:fe6d:5d8d.mdns > ff02::fb.mdns: 0 [9q] PTR (QM)? _nfs._tcp.local. PTR (QM)? _ipp._tcp.local. PTR (QM)? _ipps._tcp.loc
al. PTR (QM)? _ftp._tcp.local. PTR (QM)? _webdav._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _smb._tcp.local. PT
R (QM)? _afpovertcp._tcp.local. (141)
17:57:20.529591 IP _gateway > 224.0.0.5: OSPFv2, Hello, length 44
17:57:22.530922 IP _gateway > 224.0.0.5: OSPFv2, Hello, length 44
^C
12 packets captured
12 packets received by filter
0 packets dropped by kernel
root@pc1:/tmp/pycore.46601/pc1.conf#
```

Для генерирования пакетов в большинстве случаев использовалась утилита ping.

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

`tcpdump -c 10 -l | tee /home/bytehater/Desktop/1.txt`

```
<her proto \ip6' | tee /home/bytehater/Desktop/1.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:08:25.835786 ARP, Request who-has _gateway tell 10.0.0.21, length 28
18:08:25.835810 ARP, Reply _gateway is-at 00:00:00:aa:00:04 (oui Ethernet), length 28
18:08:25.835830 IP 10.0.0.21 > 10.0.1.20: ICMP echo request, id 24, seq 1, length 64
18:08:25.835904 IP 10.0.1.20 > 10.0.0.21: ICMP echo reply, id 24, seq 1, length 64
18:08:26.843102 IP 10.0.0.21 > 10.0.1.20: ICMP echo request, id 24, seq 2, length 64
18:08:26.843165 IP 10.0.1.20 > 10.0.0.21: ICMP echo reply, id 24, seq 2, length 64
18:08:27.866883 IP 10.0.0.21 > 10.0.1.20: ICMP echo request, id 24, seq 3, length 64
18:08:27.866941 IP 10.0.1.20 > 10.0.0.21: ICMP echo reply, id 24, seq 3, length 64
18:08:28.891047 IP 10.0.0.21 > 10.0.1.20: ICMP echo request, id 24, seq 4, length 64
18:08:28.891111 IP 10.0.1.20 > 10.0.0.21: ICMP echo reply, id 24, seq 4, length 64
10 packets captured
10 packets received by filter
0 packets dropped by kernel
root@pc1:/tmp/pycore.46601/pc1.conf#
```

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).

`tcpdump -c 5 -l -xx 'ether dst ff:ff:ff:ff:ff:ff' | tee /home/bytehater/Desktop/2.txt`

```
<ff:ff:ff:ff:ff' | tee /home/bytehater/Desktop/2.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:09:59.371019 ARP, Request who-has 10.0.0.22 tell 10.0.0.21, length 28
        0x0000:  ffff ffff ffff 0000 00aa 0005 0806 0001
        0x0010:  0800 0604 0001 0000 00aa 0005 0a00 0015
        0x0020:  0000 0000 0000 0a00 0016
18:10:43.282070 ARP, Request who-has pc1 tell 10.0.0.22, length 28
        0x0000:  ffff ffff ffff 0000 00aa 0001 0806 0001
        0x0010:  0800 0604 0001 0000 00aa 0001 0a00 0016
        0x0020:  0000 0000 0000 0a00 0014
18:11:09.308254 ARP, Request who-has _gateway tell 10.0.0.22, length 28
        0x0000:  ffff ffff ffff 0000 00aa 0001 0806 0001
        0x0010:  0800 0604 0001 0000 00aa 0001 0a00 0016
        0x0020:  0000 0000 0000 0a00 0001
^C3 packets captured
3 packets received by filter
0 packets dropped by kernel

root@pc1:/tmp/pycore.46601/pc1.conf#
```

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.

```
tcpdump -c 3 -l -XX 'ip proto \icmp and dst host 10.0.1.20' |
tee /home/bytehater/Desktop/3.txt
```

```
<host 10.0.1.20' | tee /home/bytehater/Desktop/3.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:16:43.471646 IP 10.0.0.21 > 10.0.1.20: ICMP echo request, id 24, seq 1, length 64
        0x0000:  0000 00aa 0004 0000 00aa 0005 0800 4500  ..............E.
        0x0010:  0054 110f 4000 4001 1472 0a00 0015 0a00  .T..@.@..r......
        0x0020:  0114 0800 41ce 0018 0001 dbb3 3b60 0000  ....A.......;`..
        0x0030:  0000 d931 0700 0000 0000 1011 1213 1415  ...1............
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ...........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
18:16:44.475058 IP 10.0.0.21 > 10.0.1.20: ICMP echo request, id 24, seq 2, length 64
        0x0000:  0000 00aa 0004 0000 00aa 0005 0800 4500  ..............E.
        0x0010:  0054 112b 4000 4001 1456 0a00 0015 0a00  .T.+@.@..V......
        0x0020:  0114 0800 a5bf 0018 0002 dcb3 3b60 0000  ............;`..
        0x0030:  0000 743f 0700 0000 0000 1011 1213 1415  ..t?............
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ...........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
18:16:45.498773 IP 10.0.0.21 > 10.0.1.20: ICMP echo request, id 24, seq 3, length 64
        0x0000:  0000 00aa 0004 0000 00aa 0005 0800 4500  ..............E.
        0x0010:  0054 1210 4000 4001 1371 0a00 0015 0a00  .T..@.@..q......
        0x0020:  0114 0800 f161 0018 0003 ddb3 3b60 0000  .....a......;`..
        0x0030:  0000 279c 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
3 packets captured
3 packets received by filter
0 packets dropped by kernel
root@pc1:/tmp/pycore.46601/pc1.conf#
```

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.

```
tcpdump -c 7 -l -XX -w /home/bytehater/Desktop/4.b
```

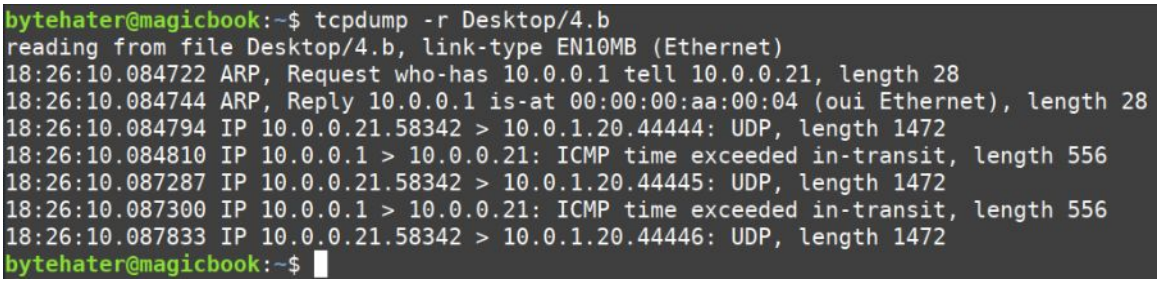
Вместо traceroute использовался аналог - tracepath

```
root@pc2:/tmp/pycore.46601/pc2.conf# tracepath 10.0.1.20
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              0.222ms
 1:  _gateway                                              0.091ms
 2:  10.0.1.20                                             0.134ms reached
     Resume: pmtu 1500 hops 2 back 2
root@pc2:/tmp/pycore.46601/pc2.conf#
```

```
<b 'not dst host 224.0.0.5 and not ether proto \ip6'
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144
 bytes
7 packets captured
8 packets received by filter
0 packets dropped by kernel
root@pc1:/tmp/pycore.46601/pc1.conf#
```

5. Прочесть программой tcpdump созданный в предыдущем пункте файл.

```
tcpdump -r /home/bytehater/Desktop/4.b
```



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

1) перехват только ARP пакетов

   ```
   tcpdump -l 'ether proto \arp'
   ```

2) перехват всех пакетов кроме ICMP

   ```
   tcpdump -l 'not ip proto \icmp'
   ```

3) перехват не широковещательных ARP пакетов

   ```
   tcpdump -l 'ether proto \arp and not ether dst
   ff:ff:ff:ff:ff:ff'
   ```

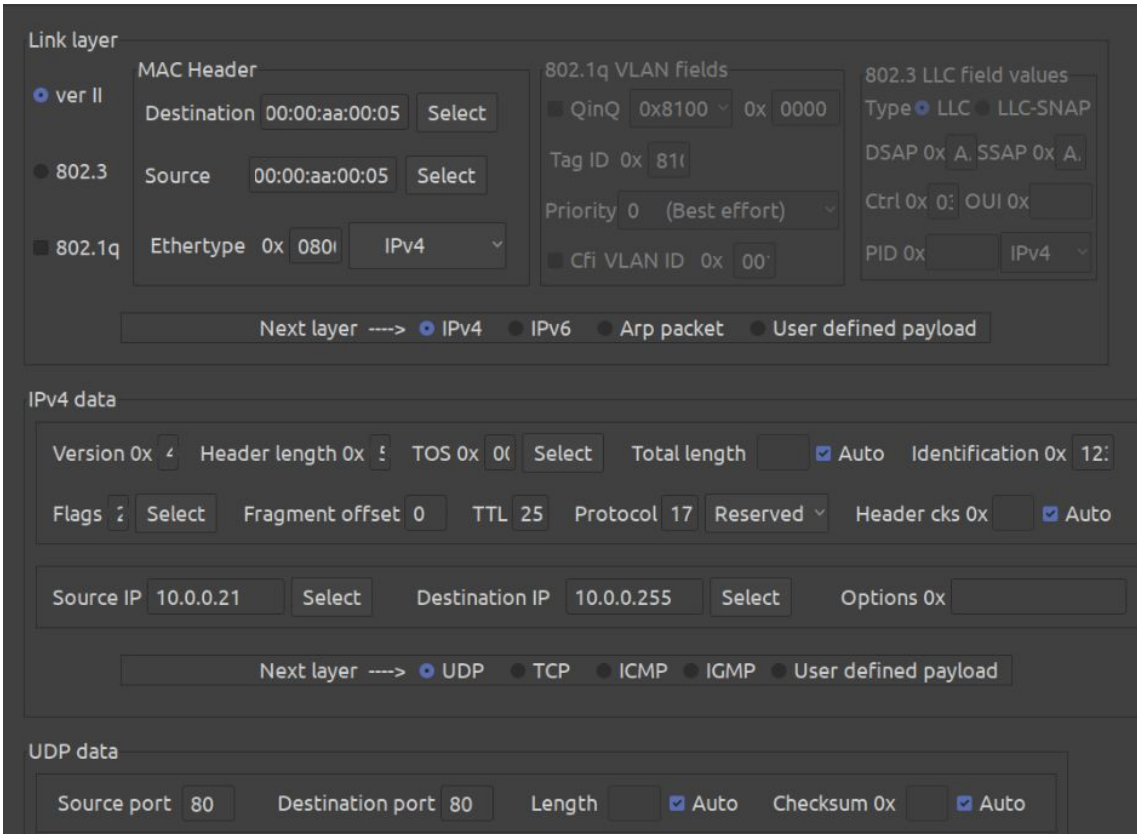
Работа с анализатором протоколов wireshark

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

not ospf and ip.dst==10.0.0.255

No.	Time	Source	Destination	Protocol	Length	Info
4	4.828223176	10.0.0.21	10.0.0.255	UDP	60	80 → 80 Len=0
10	13.158303050	10.0.0.21	10.0.0.255	UDP	60	80 → 80 Len=0
11	13.791550208	10.0.0.21	10.0.0.255	UDP	60	80 → 80 Len=0
12	13.992650178	10.0.0.21	10.0.0.255	UDP	60	80 → 80 Len=0
14	14.172875048	10.0.0.21	10.0.0.255	UDP	60	80 → 80 Len=0

Для генерации пакетов использовалась утилита packeth.



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

icmp and ip.src==10.0.0.21

No.	Time	Source	Destination	Protocol	Length	Info
82	91.137082122	10.0.0.21	10.0.1.20	ICMP	98	Echo (ping) request
84	92.160974755	10.0.0.21	10.0.1.20	ICMP	98	Echo (ping) request
88	93.184750563	10.0.0.21	10.0.1.20	ICMP	98	Echo (ping) request
90	94.208978720	10.0.0.21	10.0.1.20	ICMP	98	Echo (ping) request
93	95.232902550	10.0.0.21	10.0.1.20	ICMP	98	Echo (ping) request

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.

not ospf

No.	Time	Source	Destination	Protocol	Length	Info
4	4.551478912	00:00:00_aa:00:05	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.21
5	4.551535166	00:00:00_aa:00:04	00:00:00_aa:00:05	ARP	42	10.0.0.1 is at 00:00:00:aa:00:04
6	4.551553838	10.0.0.21	10.0.1.20	UDP	1514	50292 → 44444 Len=1472
7	4.551595803	10.0.0.1	10.0.0.21	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
8	4.553930111	10.0.0.21	10.0.1.20	UDP	1514	50292 → 44445 Len=1472
9	4.553977229	10.0.0.1	10.0.0.21	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
10	4.554636260	10.0.0.21	10.0.1.20	UDP	1514	50292 → 44446 Len=1472
11	4.554757241	10.0.1.20	10.0.0.21	ICMP	590	Destination unreachable (Port unreachable)
15	9.583278493	00:00:00_aa:00:04	00:00:00_aa:00:05	ARP	42	Who has 10.0.0.21? Tell 10.0.0.1
16	9.583320547	00:00:00_aa:00:05	00:00:00_aa:00:04	ARP	42	10.0.0.21 is at 00:00:00:aa:00:05

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

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:00:00_aa:00:05	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.21
2	0.000022	00:00:00_aa:00:04	00:00:00_aa:00:05	ARP	42	10.0.0.1 is at 00:00:00:aa:00:04
3	0.000072	10.0.0.21	10.0.1.20	UDP	1514	58342 → 44444 Len=1472
4	0.000088	10.0.0.1	10.0.0.21	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
5	0.002565	10.0.0.21	10.0.1.20	UDP	1514	58342 → 44445 Len=1472
6	0.002578	10.0.0.1	10.0.0.21	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
7	0.003111	10.0.0.21	10.0.1.20	UDP	1514	58342 → 44446 Len=1472

Пакеты перехваченные tcpdump совпадают с полученными с помощью wireshark (с учетом ограничение на количество пакетов в первом случае)