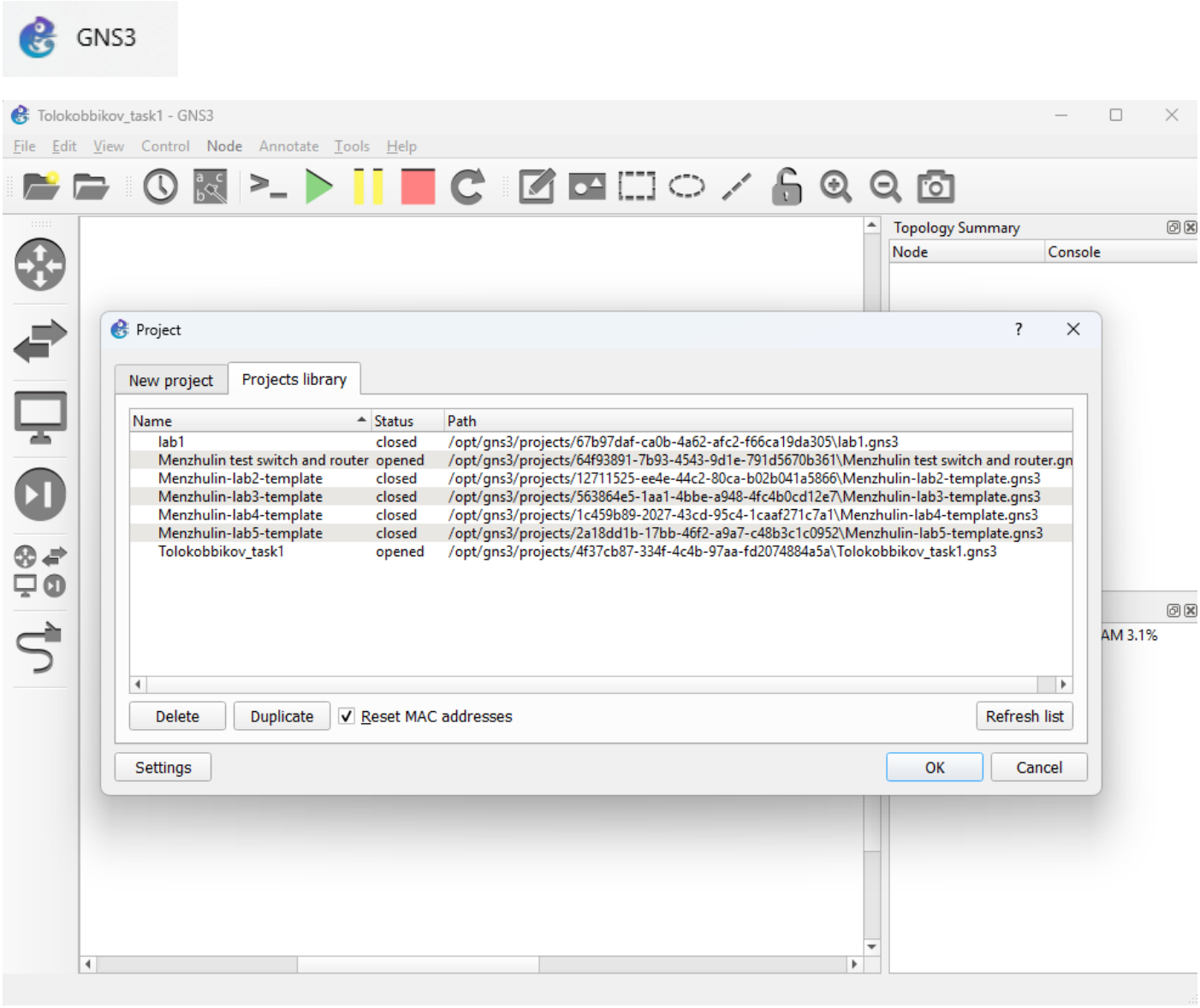
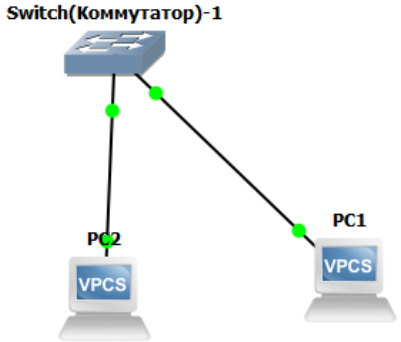
1) Установить и настроить эмулятор GNS3



2) Создать простейшую сеть, состоящую из 1 коммутатора и 2 компьютеров, назначить им произвольные ip адреса из одной сети

Ставим коммутатор, 2 VPSC, подключаем их между собой по Ethernet.

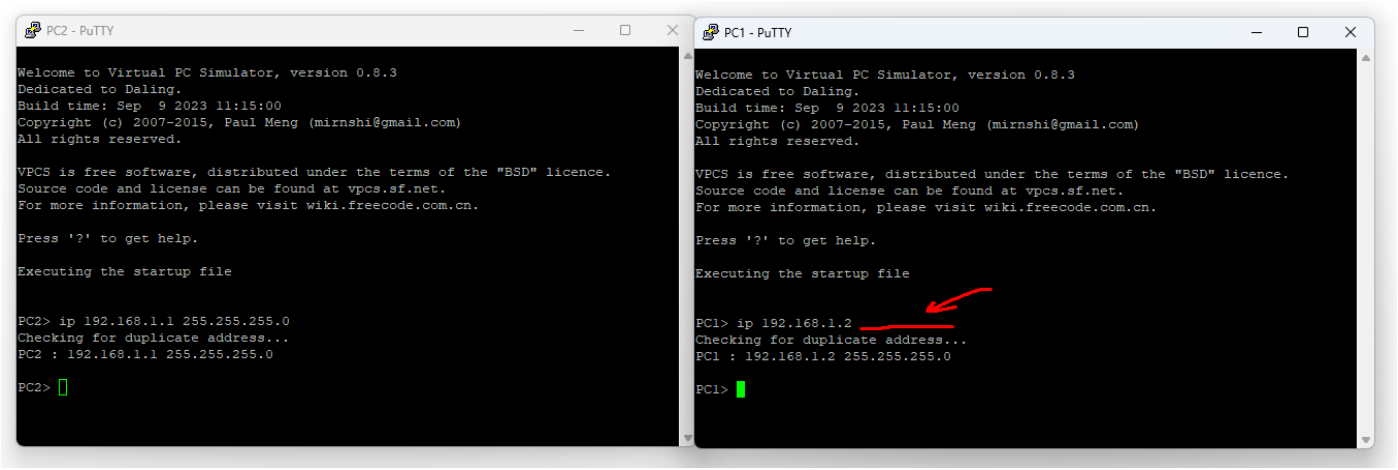


Жмём зелёную кнопку, начинаем симуляцию.

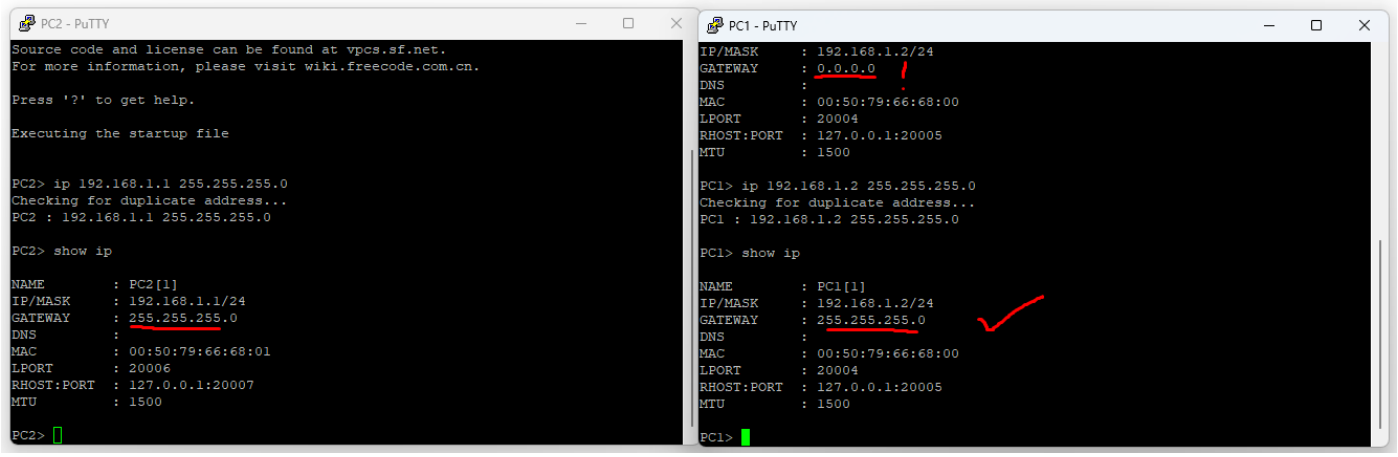
Присваиваем IP. Я перепутал компьютеры местами, поэтому получилось так:
PC1 – 192.168.1.2

PC2 – 192.168.1.1

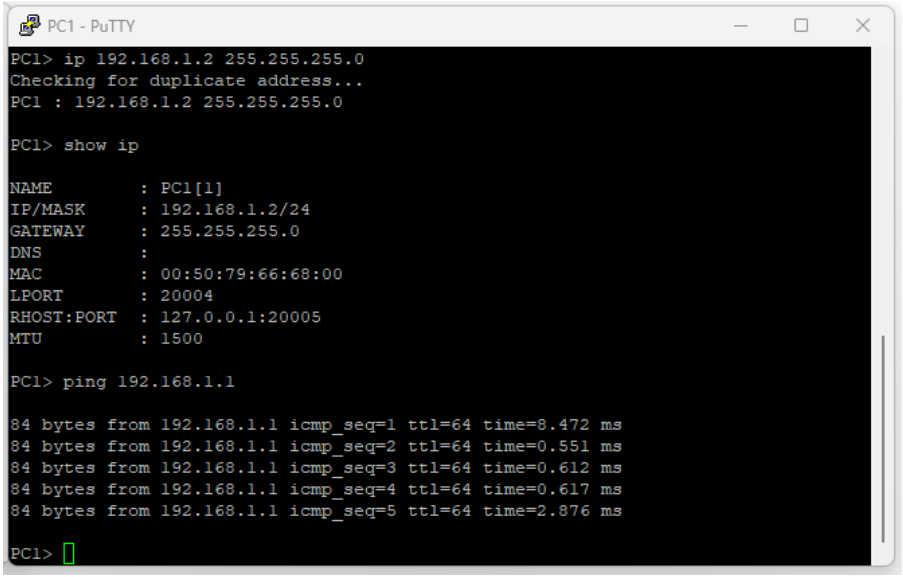
Маска – 255.255.255.0



Причём я забыл маску вписать второму, так что по новой.



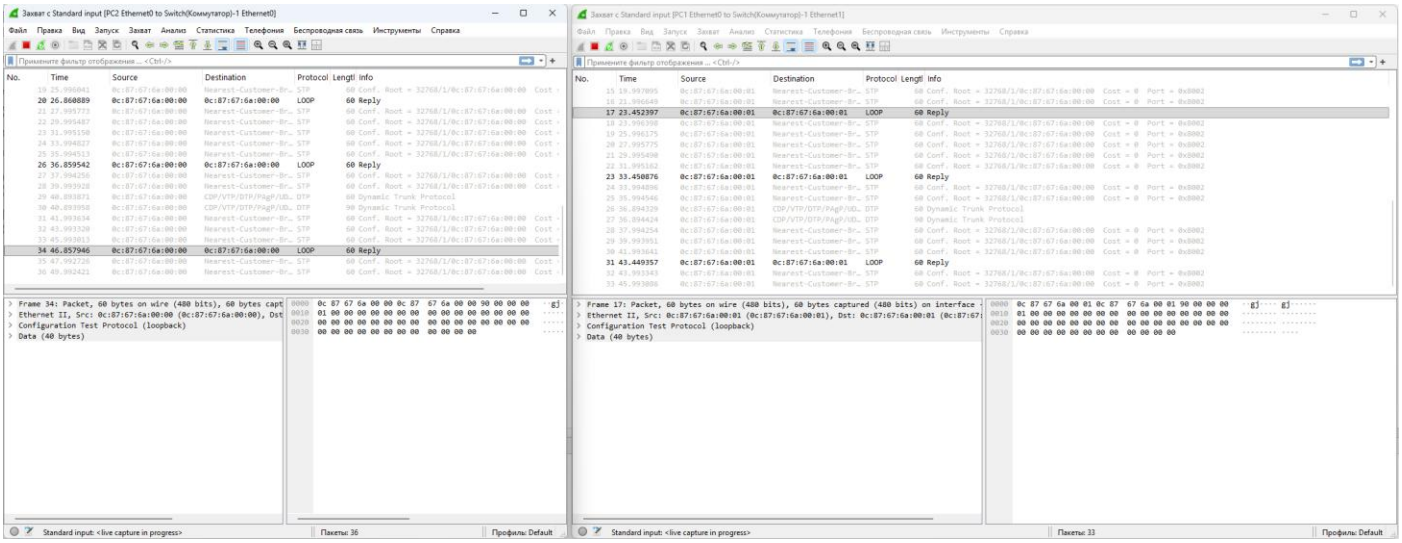
3) Запустить симуляцию, выполнить команду ping с одного из компьютеров, используя ip адрес второго компьютера



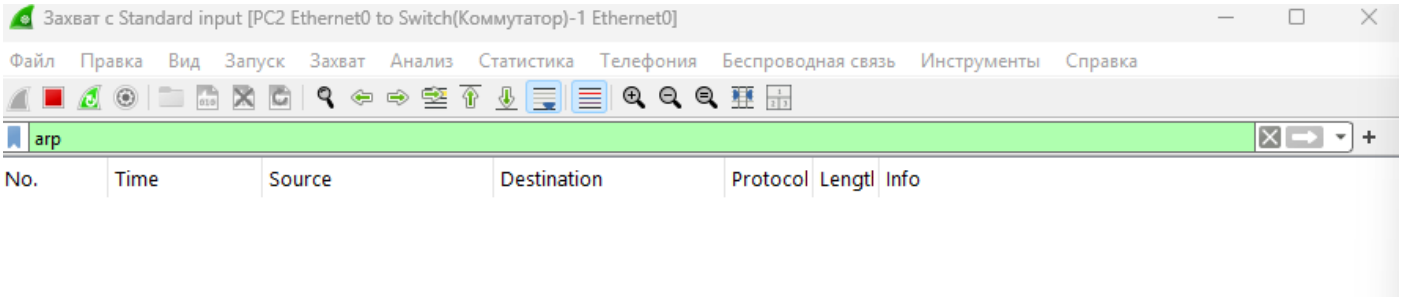
С первого компьютера с ip 192.168.1.2 пинганули второй с ip 192.168.1.1

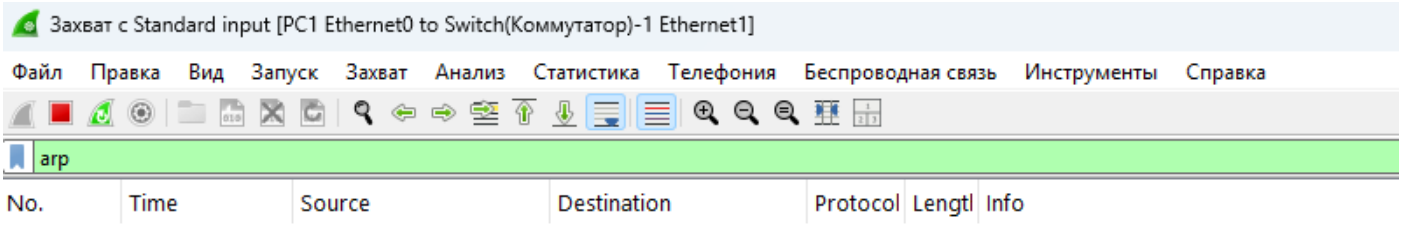
4) Перехватить трафик протокола arp на всех линках(nb!), задокументировать и проанализировать заголовки пакетов в программе Wireshark, для фильтрации трафика, относящегося к указанному протоколу использовать фильтры Wireshark

ПКМ на линк, “start capture”, смотрим на пакеты

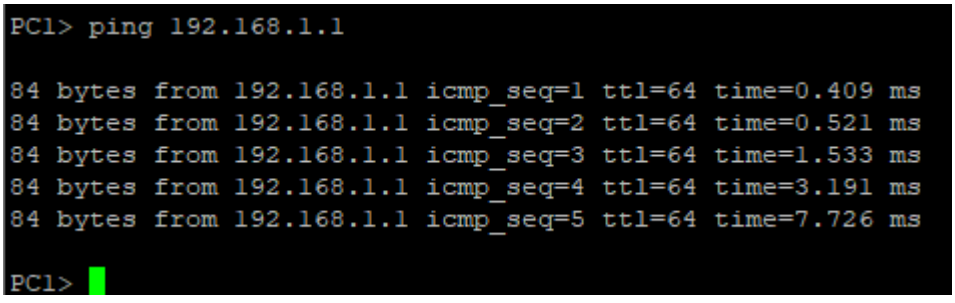


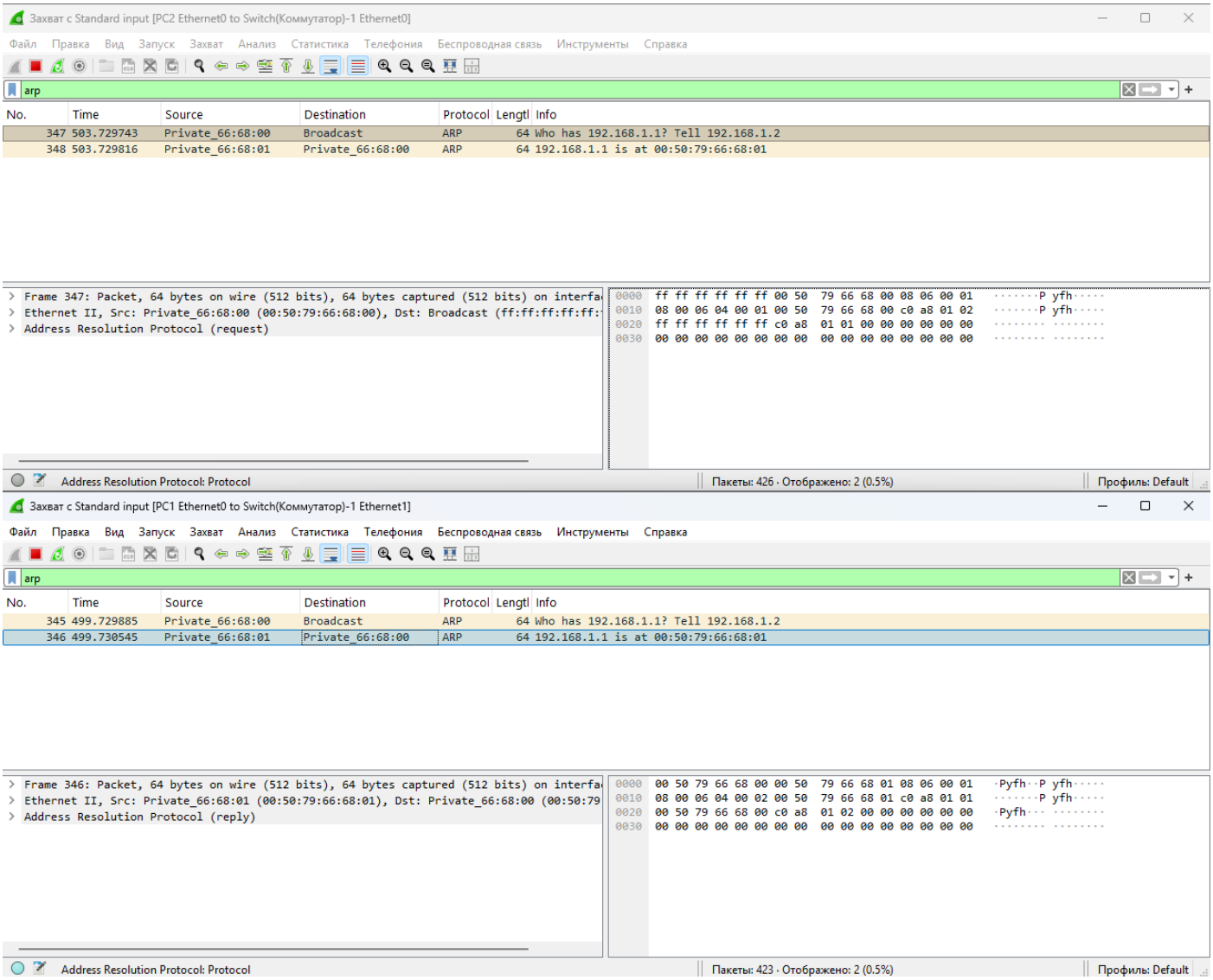
Применим фильтр arp





Снова делаем ping





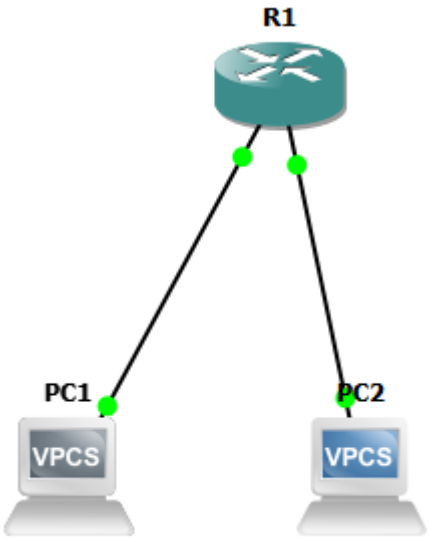
Теперь анализируем пакеты arp после ping:

На каждом линке два пакета. Первый - broadcast, то есть для всех. В инфо сказано "Who has 192.168.1.1"? Tell 192.168.1.2. Что прямым текстом говорит, что мы по всей сети ищем обладателя такого ip и просим нам ответить назад, оставив свой ip.

Так как 192.168.1.1 есть в сети, то он пакет получил и отправил уже приватный ответ для 192.168.1.2, не за чем бродкастом опять кричать. В ответ причём он сообщил свой MAK адрес. Мол «Я тут».

Заголовки рассмотрим, проанализируем и сравним в конце отчёта, чтобы было, что сравнивать

5) Создать простейшую сеть, состоящую из 1 маршрутизатора и 2 компьютеров, назначить им произвольные ip адреса из разных сетей



Настроим маршрутизатор

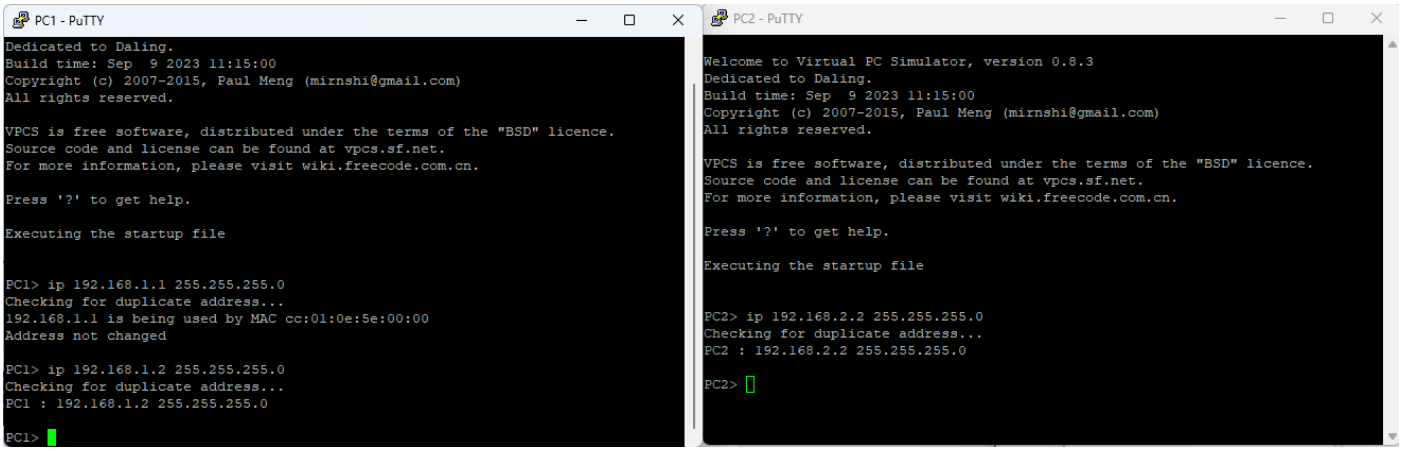
```
R1#configure
Configuring from terminal, memory, or network [terminal]? terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#des
*Mar  1 00:04:01.967: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:04:02.967: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#description "To PC1"
R1(config-if)#exit
R1(config)#interface FastEthernet1/0
R1(config-if)#ip address 192.168.2.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:05:08.247: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
*Mar  1 00:05:09.247: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
R1(config-if)#description "To PC2"
R1(config-if)#exit
R1(config)#exit
R1#
*Mar  1 00:05:55.131: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip
% Incomplete command.

R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.1.1     YES manual up                    up
FastEthernet1/0            192.168.2.1     YES manual up                    up
Ethernet2/0                unassigned      YES unset  administratively down down
Ethernet2/1                unassigned      YES unset  administratively down down
Ethernet2/2                unassigned      YES unset  administratively down down
Ethernet2/3                unassigned      YES unset  administratively down down
Serial3/0                  unassigned      YES unset  administratively down down
Serial3/1                  unassigned      YES unset  administratively down down
Serial3/2                  unassigned      YES unset  administratively down down
Serial3/3                  unassigned      YES unset  administratively down down
R1#
```

Ещё запишем конфигурацию

```
R1#write memory
Building configuration...
[OK]
R1#
```

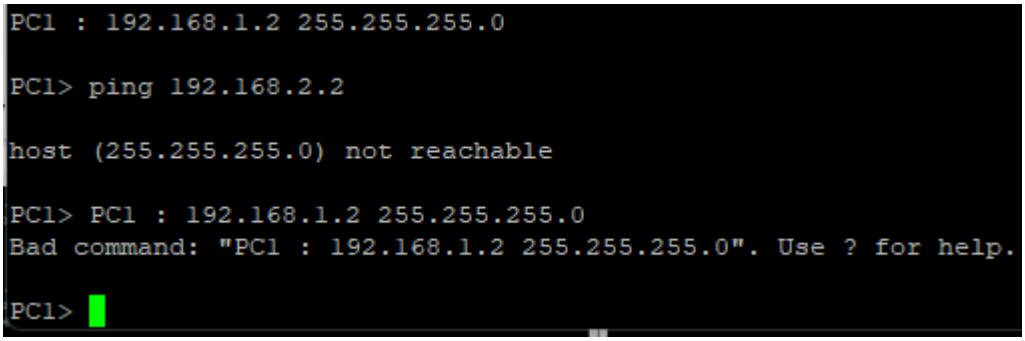
На всякий

Теперь настроим компьютеры

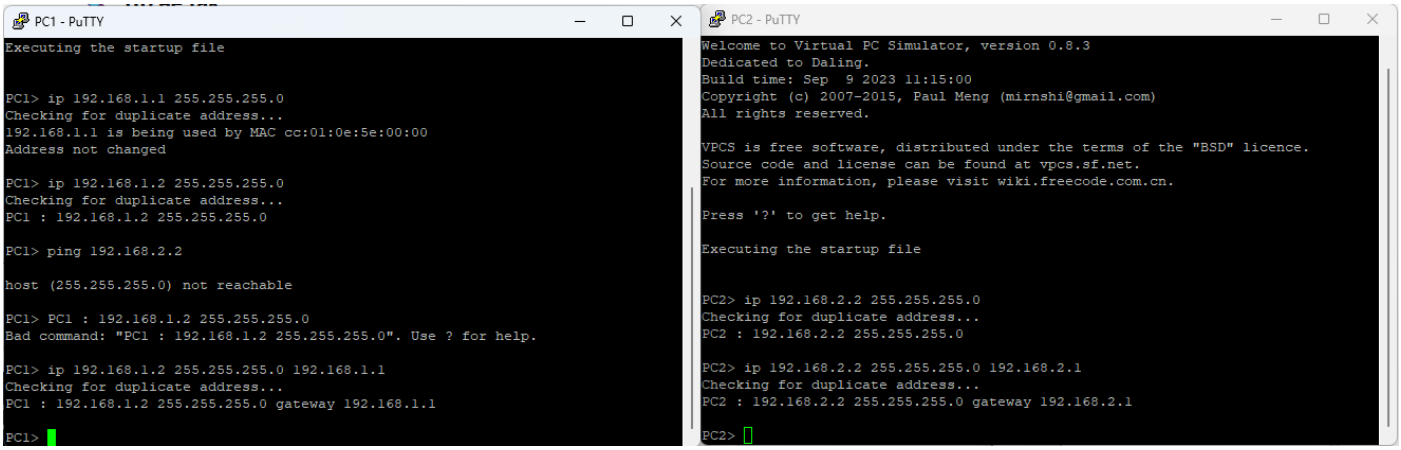


PC1 теперь с логичным ему ip – 192.168.1.2

PC2 – 192.168.2.2



Оказалось, забыл шлюз указать. Исправляю:



6) Запустить симуляцию, выполнить команду ping с одного из компьютеров, используя ip адрес второго компьютера

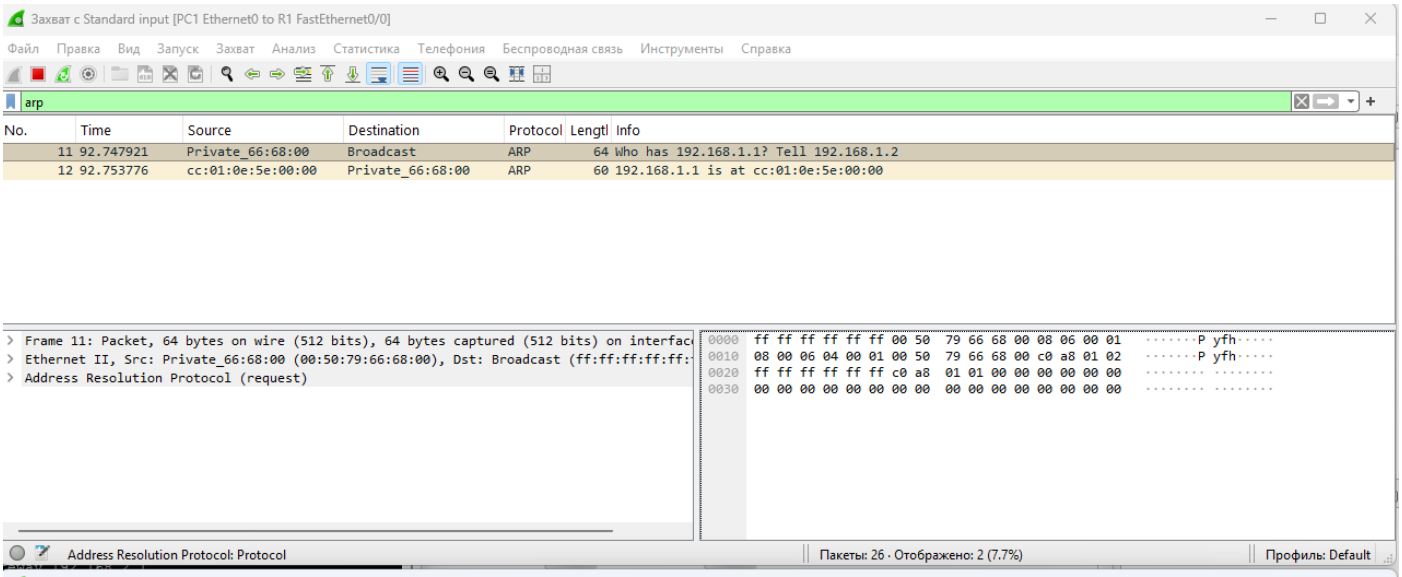
```
PC1> ping 192.168.2.2

192.168.2.2 icmp_seq=1 timeout
84 bytes from 192.168.2.2 icmp_seq=2 ttl=63 time=13.299 ms
84 bytes from 192.168.2.2 icmp_seq=3 ttl=63 time=16.335 ms
84 bytes from 192.168.2.2 icmp_seq=4 ttl=63 time=16.338 ms
84 bytes from 192.168.2.2 icmp_seq=5 ttl=63 time=16.039 ms

PC1> ping 192.168.2.2

84 bytes from 192.168.2.2 icmp_seq=1 ttl=63 time=19.489 ms
84 bytes from 192.168.2.2 icmp_seq=2 ttl=63 time=16.238 ms
84 bytes from 192.168.2.2 icmp_seq=3 ttl=63 time=15.655 ms
84 bytes from 192.168.2.2 icmp_seq=4 ttl=63 time=16.068 ms
84 bytes from 192.168.2.2 icmp_seq=5 ttl=63 time=15.347 ms

PC1>
```

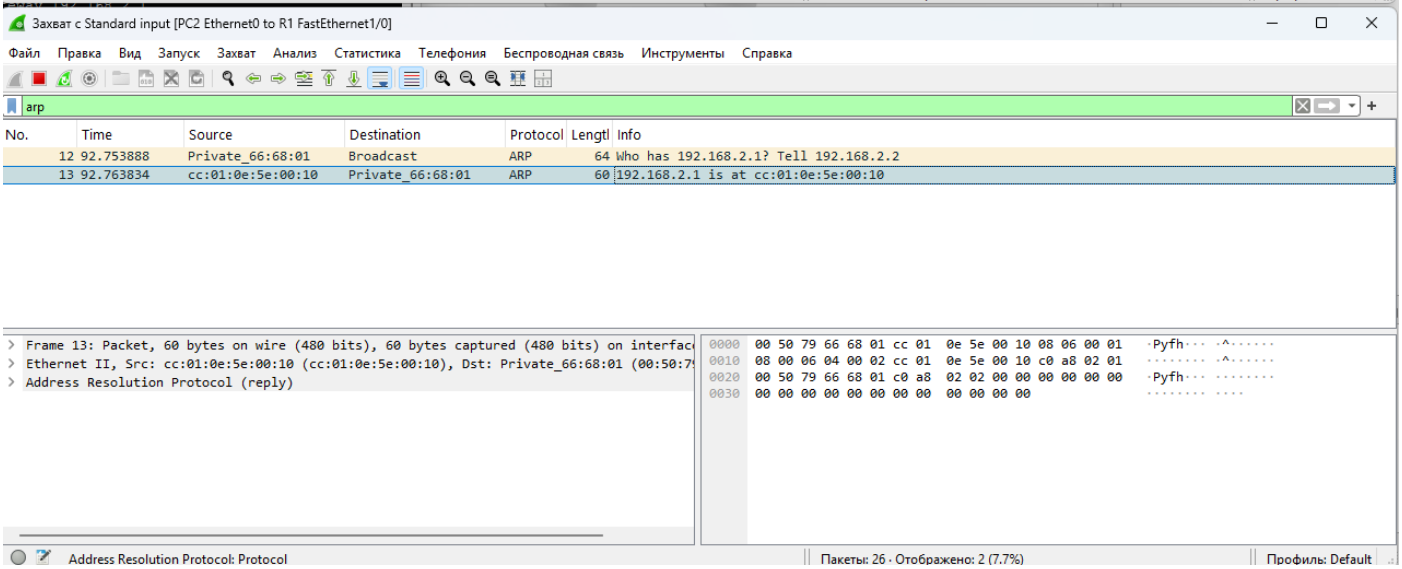
```
PC2> ping 192.168.1.2

84 bytes from 192.168.1.2 icmp_seq=1 ttl=63 time=10.476 ms
84 bytes from 192.168.1.2 icmp_seq=2 ttl=63 time=17.236 ms
84 bytes from 192.168.1.2 icmp_seq=3 ttl=63 time=16.506 ms
84 bytes from 192.168.1.2 icmp_seq=4 ttl=63 time=16.841 ms
84 bytes from 192.168.1.2 icmp_seq=5 ttl=63 time=16.921 ms

PC2>
```

Куда первый подевался, не очень понятно. Но работает теперь

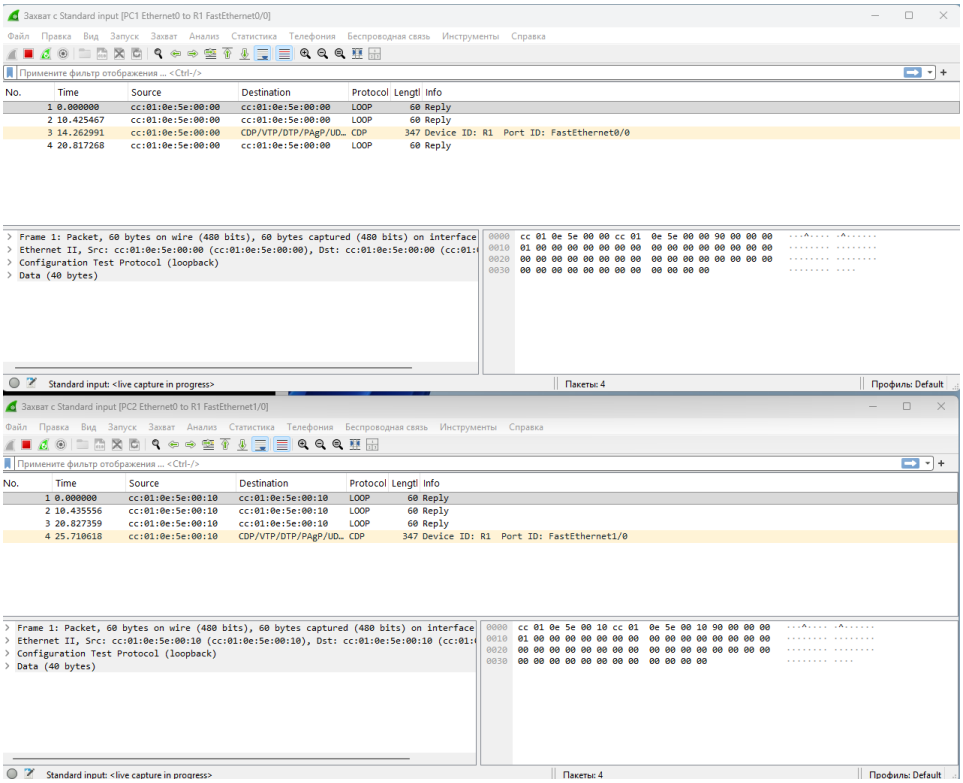
7) Перехватить трафик протокола arp и icmp на всех линках(nb!), задокументировать и проанализировать заголовки пакетов в программе Wireshark, для фильтрации трафика, относящегося к указанному протоколу использовать фильтры Wireshark

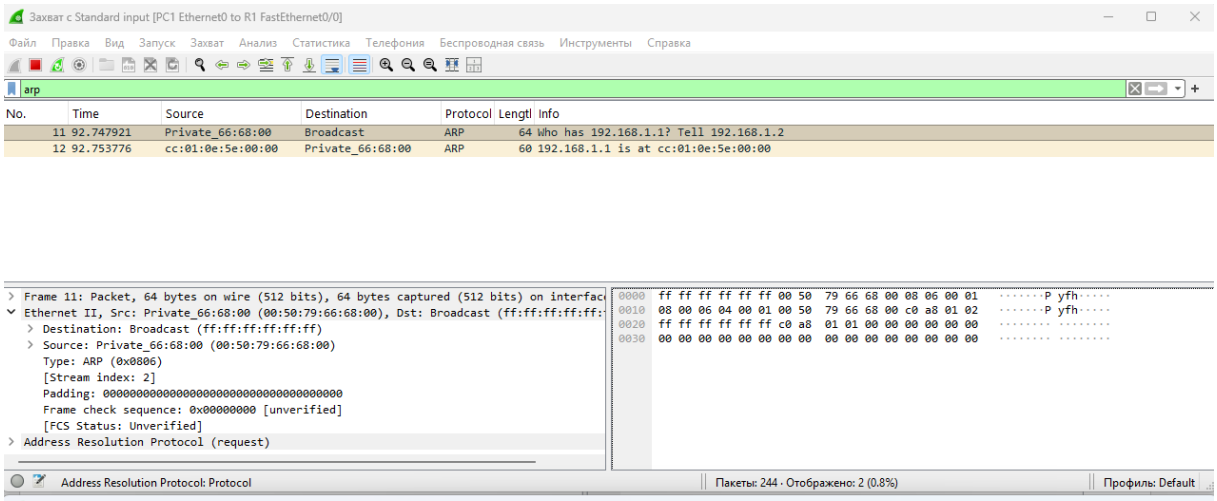
```
PC1> ping 192.168.2.2

84 bytes from 192.168.2.2 icmp_seq=1 ttl=63 time=29.769 ms
84 bytes from 192.168.2.2 icmp_seq=2 ttl=63 time=17.100 ms
84 bytes from 192.168.2.2 icmp_seq=3 ttl=63 time=15.973 ms
84 bytes from 192.168.2.2 icmp_seq=4 ttl=63 time=16.700 ms
84 bytes from 192.168.2.2 icmp_seq=5 ttl=63 time=16.690 ms

PC1>
```

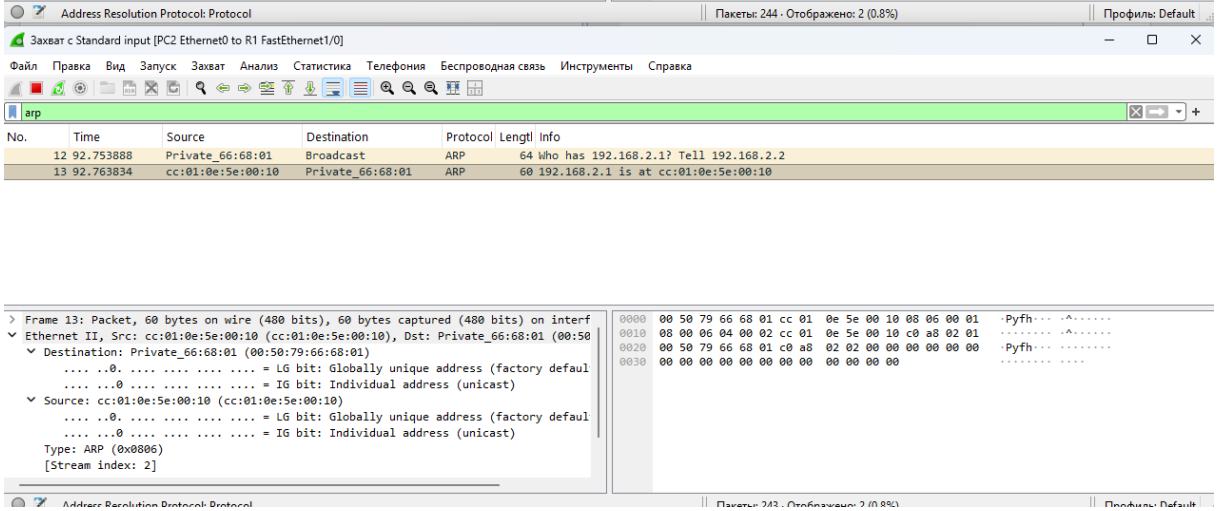




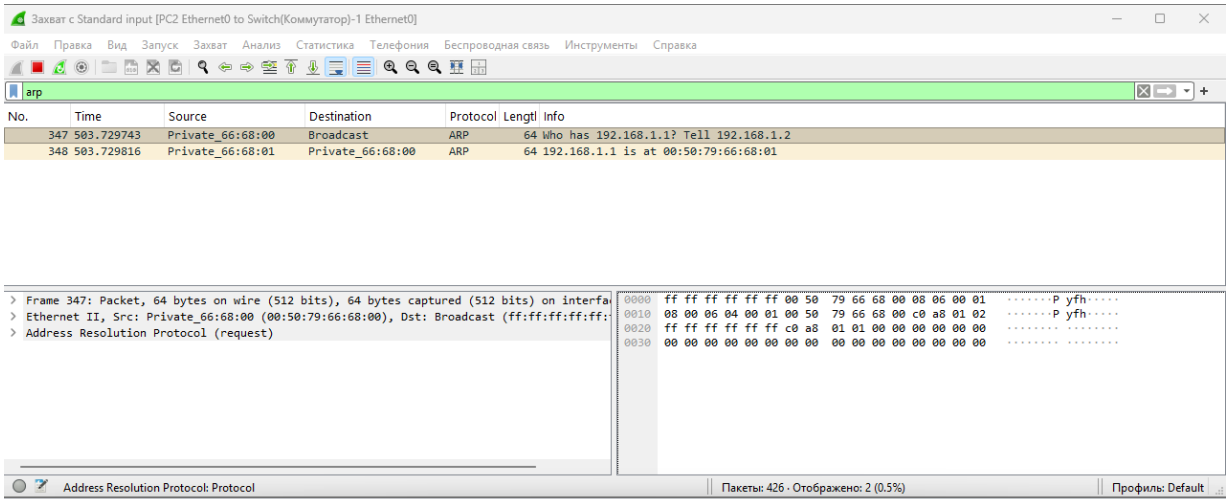
Ну, кстати, не так часто пакетами обмениваются на фоне прошлой структуры

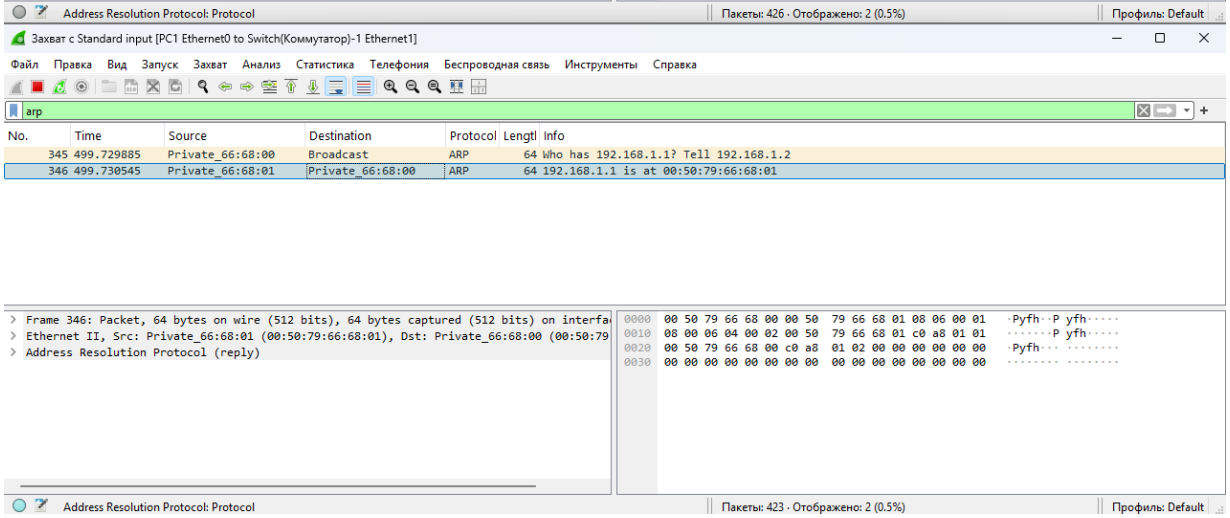






Из прошлого сетапа:





Отличия в заголовках в том, что маршрутизатор при пересылке меняет Source пакета, а коммутатор этого не делает.

С коммутатором:

Пакет шёл с source: private_66:68:00, через коммутатор так и прошёл, не изменился source.

Ответный пакет шёл с source: private_66:68:01, так и остался private_66:68:01

С маршрутизатором:

Пакет шёл с source: private_66:68:00, через маршрутизатор прошёл и стал source: private_66:68:01. Мол, теперь это пакет, высланный маршрутизатором. Поменялся мак адрес отправителя.

И ответ теперь тоже маршрутизатору, destination: cc:01:0e:5e:00:10, но маршрутизатор меняет destination на private_66:68:00. PC1 успешно получает свой пакет через маршрутизатор.