Усова Виктория Лабораторная работа № 1

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

Для PC 1 назначим адрес 192.168.12.5:

ip 192.168.12.5 255.255.255.0

save

```
PC1> ip 192.168.12.5 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.12.5 255.255.255.0

PC1> save
Saving startup configuration to startup.vpc
.  done
```

Для PC2 192.168.12.6:

ip 192.168.12.6 255.255.255.0

save

```
PC2> ip 192.168.12.6 255.255.255.0
Checking for duplicate address...
PC2 : 192.168.12.6 255.255.255.0

PC2> save
Saving startup configuration to startup.vpc
.  done
```

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

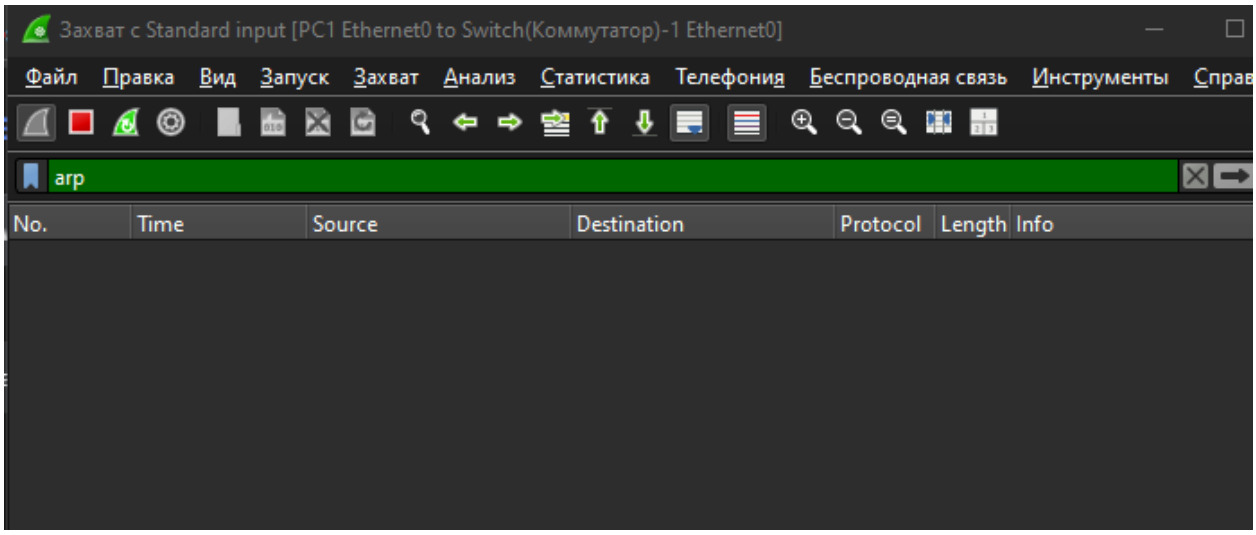
Проверим пакеты на PC1, введя команду ping 192.168.12.6

```
PC1> ping 192.168.12.6

84 bytes from 192.168.12.6 icmp_seq=1 ttl=64 time=0.471 ms
84 bytes from 192.168.12.6 icmp_seq=2 ttl=64 time=0.510 ms
84 bytes from 192.168.12.6 icmp_seq=3 ttl=64 time=4.521 ms
84 bytes from 192.168.12.6 icmp_seq=4 ttl=64 time=0.596 ms
84 bytes from 192.168.12.6 icmp_seq=5 ttl=64 time=5.634 ms
```

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

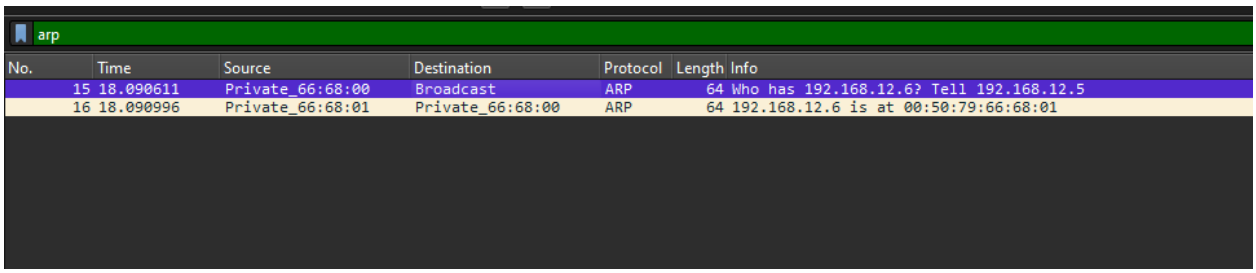
Нажимаем на связь PC1 и коммутатора и начинаем захват, применяем фильтр. То же самое со связью PC2 и коммутатора.

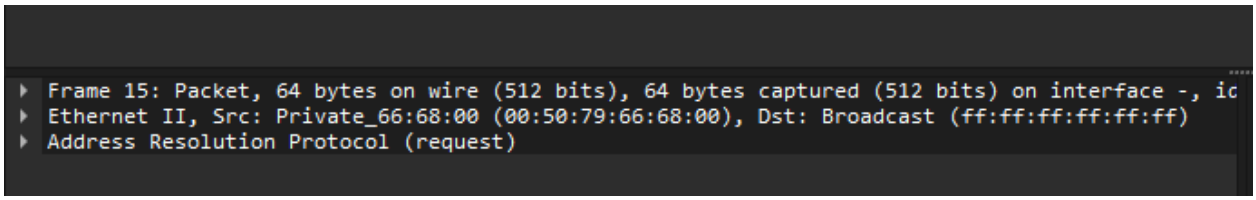


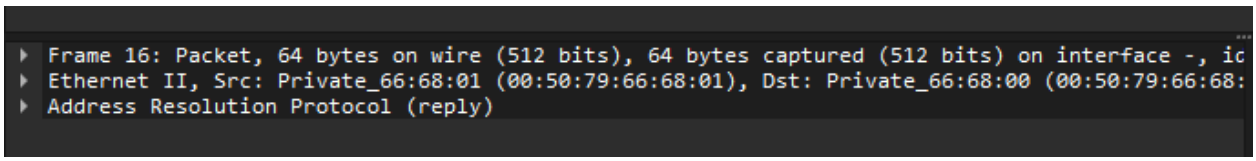
Снова ловим пинг с PC2 ping 192.168.12.6

Так сможем захватить нужные нам пакеты – всего получили по 2 пакета

Проанализируем связь PC1 и коммутатора:





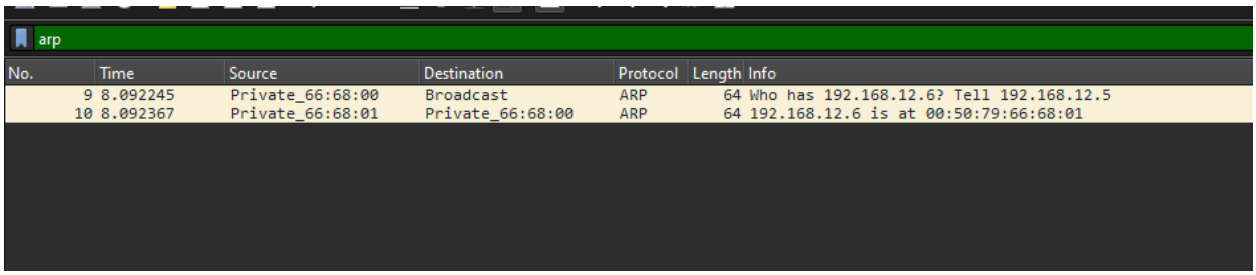


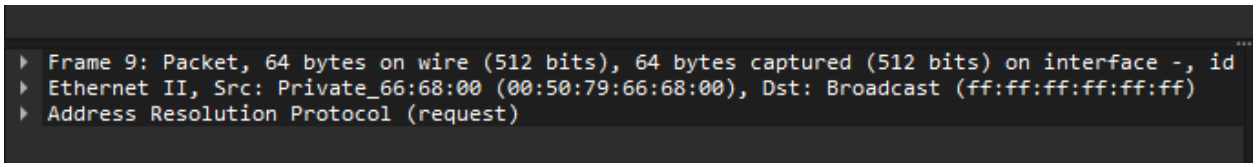
Захвачено 2 пакета протокола ARP:

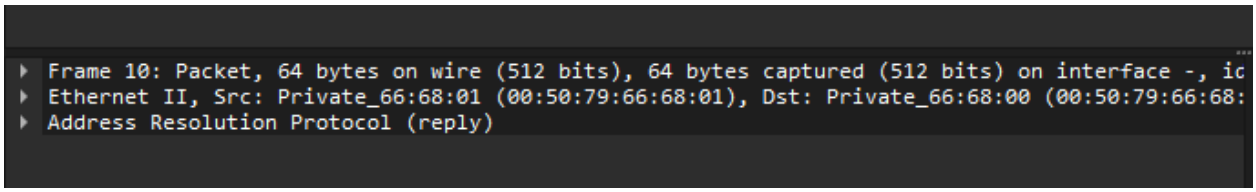
ПАКЕТ №15 (ARP Request): Время 18,091 секунд, источник 00:50:79:66:68:00 – MAC-адрес PC1. Цель – (всем) широковещательный домен, протокол ARP, длина 64 байта. Информация: «Какая машина имеет IP 192.168.12.6? Ответьте 192.168.12.5».

ПАКЕТ №16 (ARP Reply): Время 18,091 секунд (ответ был отправлен почти сразу после запроса), источник 00:50:79:66:68:01 – MAC-адрес PC2. Цель - 00:50:79:66:68:00 – MAC-адрес PC1 (Используется уже не Broadcast, так как MAC-адрес отправителя запроса был сообщён ранее). Протокол ARP. Длина 64 байта, Информация – машина сообщает что адресу IP 192.168.12.6 соответствует MAC 00:50:79:66:68:01.

Также проанализируем связь PC2 с коммутатором.

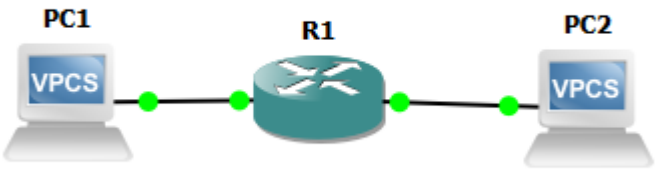






Как видим фреймы полностью соответствуют по содержанию фреймам, захваченных между PC1 и коммутатором. Отличается лишь время (время отсчитывается с момента начала захвата и номер фрейма).

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



Настроим роутер:

enable

configure terminal

interface FastEthernet0/0

ip address 192.168.1.1 255.255.255.0

no shutdown

exit

interface FastEthernet1/0

ip address 192.168.2.1 255.255.255.0

no shutdown

exit

Включаем маршрутизацию

ip routing

end

write memory

Настроим PC1

ip 192.168.1.10 192.168.1.1

save

Настроим PC2

ip 192.168.2.10 192.168.2.1

save

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

Запускаем на PC1

ping 192.168.2.10

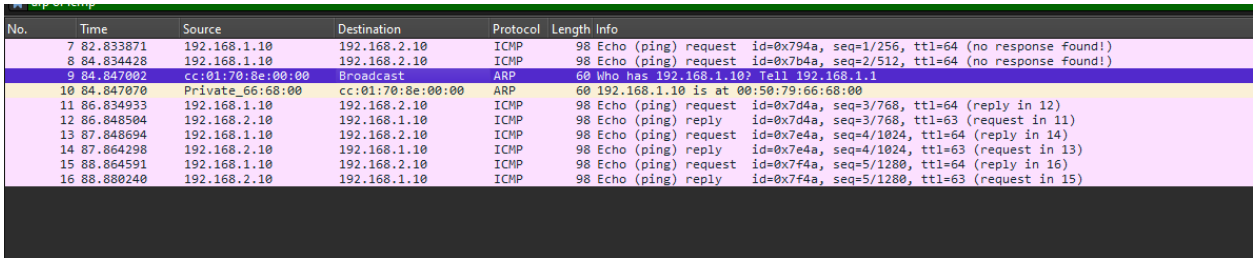
```
PC1> ping 192.168.2.10

192.168.2.10 icmp_seq=1 timeout
84 bytes from 192.168.2.10 icmp_seq=2 ttl=63 time=19.940 ms
84 bytes from 192.168.2.10 icmp_seq=3 ttl=63 time=15.884 ms
84 bytes from 192.168.2.10 icmp_seq=4 ttl=63 time=14.817 ms
84 bytes from 192.168.2.10 icmp_seq=5 ttl=63 time=15.962 ms
```

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

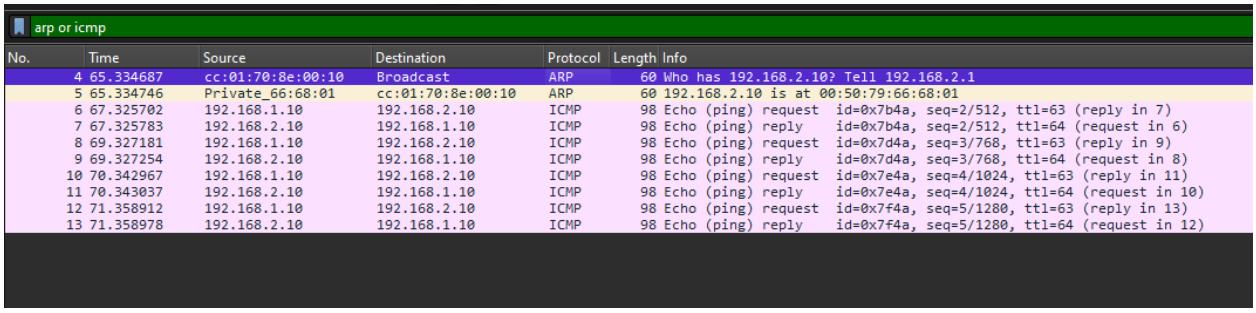
Снова начинаем захват пакетов, ставим фильтр arp or icmp и запускаем ping

С PC1 на порт маршрутизатора



No.	Time	Source	Destination	Protocol	Length	Info
7	82.833871	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x794a, seq=1/256, ttl=64 (no response found!)
8	84.834428	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7b4a, seq=2/512, ttl=64 (no response found!)
9	84.847002	cc:01:70:8e:00:00	Broadcast	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
10	84.847070	Private_66:68:00	cc:01:70:8e:00:00	ARP	60	192.168.1.10 is at 00:50:79:66:68:00
11	86.834933	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7d4a, seq=3/768, ttl=64 (reply in 12)
12	86.848504	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x7d4a, seq=3/768, ttl=63 (request in 11)
13	87.848694	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7e4a, seq=4/1024, ttl=64 (reply in 14)
14	87.864298	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x7e4a, seq=4/1024, ttl=63 (request in 13)
15	88.864591	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7f4a, seq=5/1280, ttl=64 (reply in 16)
16	88.880240	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x7f4a, seq=5/1280, ttl=63 (request in 15)

С маршрутизатора на PC2

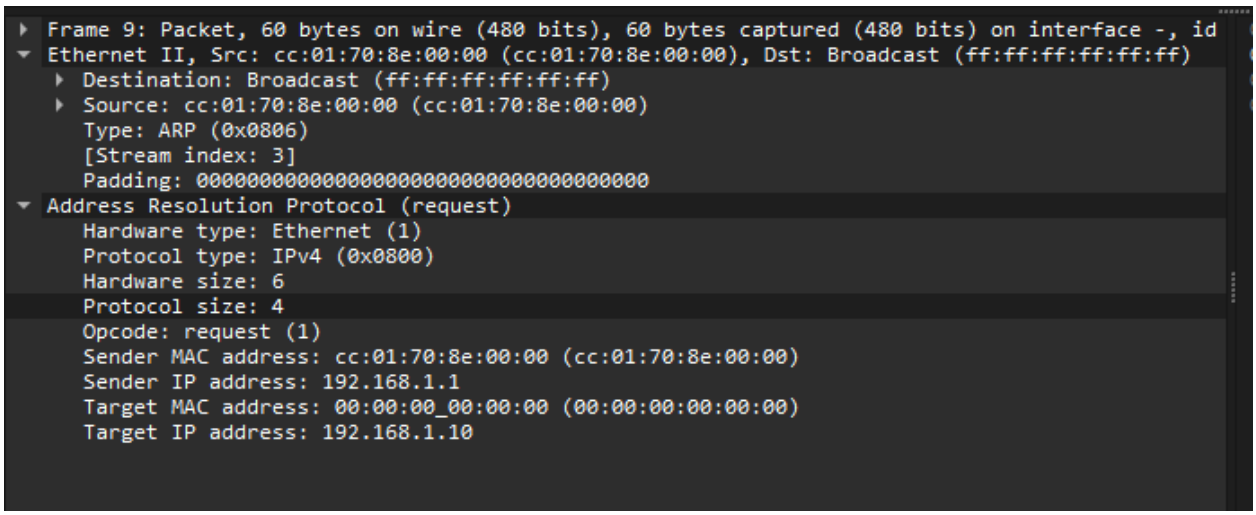


arp or icmp

No.	Time	Source	Destination	Protocol	Length	Info
4	65.334687	cc:01:70:8e:00:10	Broadcast	ARP	60	Who has 192.168.2.10? Tell 192.168.2.1
5	65.334746	Private_66:68:01	cc:01:70:8e:00:10	ARP	60	192.168.2.10 is at 00:50:79:66:68:01
6	67.325702	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7b4a, seq=2/512, ttl=63 (reply in 7)
7	67.325783	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x7b4a, seq=2/512, ttl=64 (request in 6)
8	69.327181	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7d4a, seq=3/768, ttl=63 (reply in 9)
9	69.327254	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x7d4a, seq=3/768, ttl=64 (request in 8)
10	70.342967	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7e4a, seq=4/1024, ttl=63 (reply in 11)
11	70.343037	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x7e4a, seq=4/1024, ttl=64 (request in 10)
12	71.358912	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x7f4a, seq=5/1280, ttl=63 (reply in 13)
13	71.358978	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x7f4a, seq=5/1280, ttl=64 (request in 12)

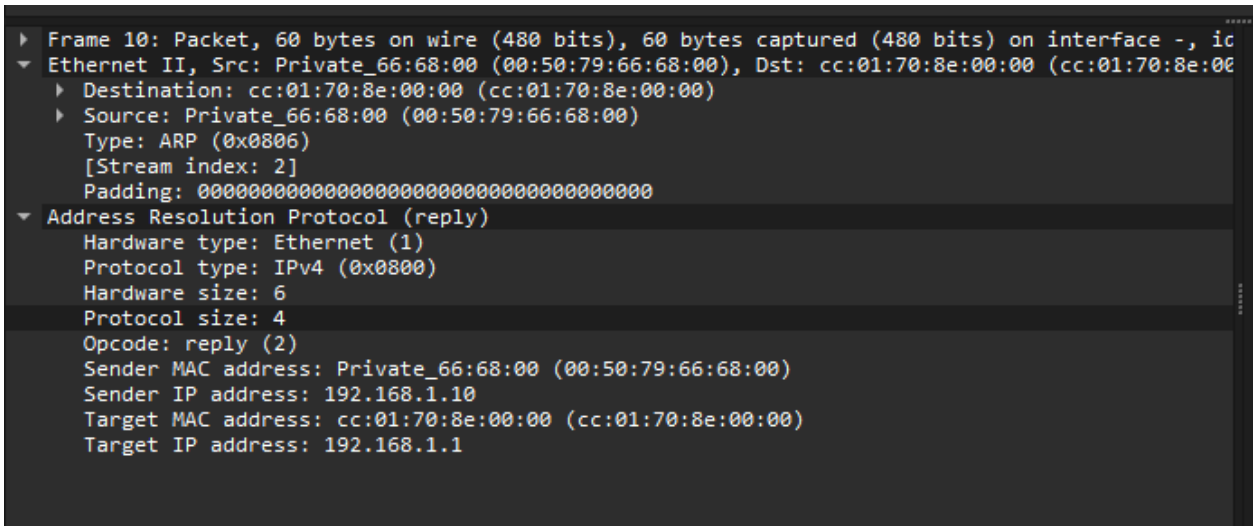
Сначала рассмотрим PC1. Есть 4 вида запроса:

ARP-запросы и ARP-ответы. Маршрутизатор начинает опрашивать кому принадлежат IP адреса, которых у него нет в кэше, и уже после управляет передачей пакетов по IP, зная MAC-адреса устройств. Но на линке между PC1 и маршрутизатором мы не получили запрос, кому принадлежит IP PC2 и наоборот, потому что маршрутизатор знает, что заданный IP подключён на заданном интерфейсе и отправляет запрос лишь в одну сеть.

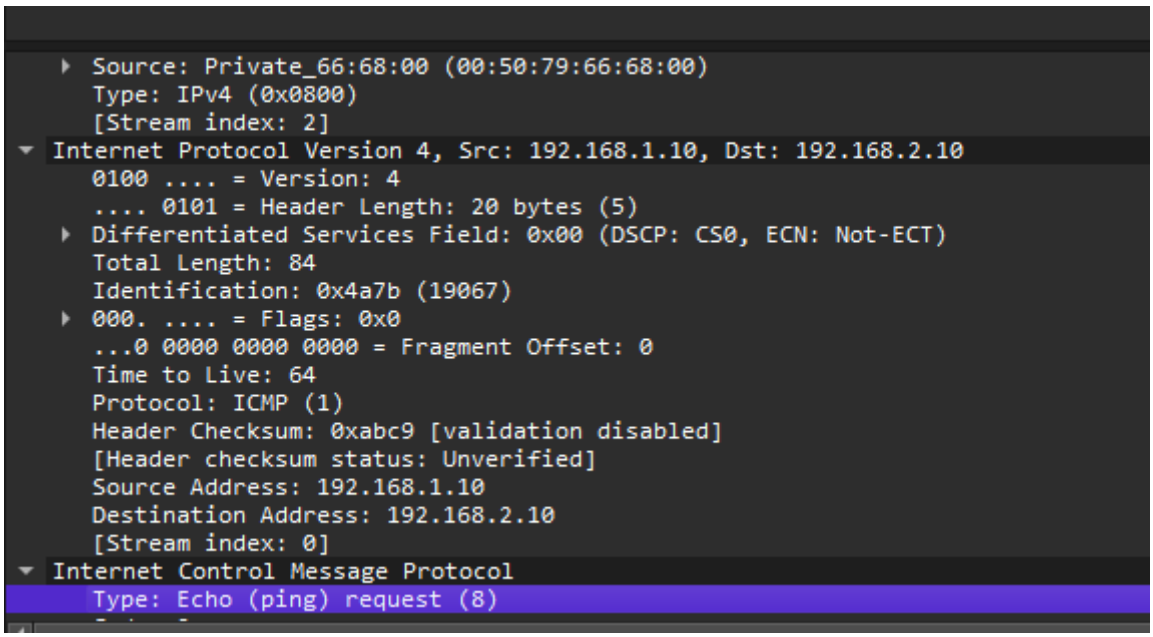


ARP-запрос. 60 байт с MAC-адреса 00:50:79:66:68:00 (PC1) на широковещательный домен. Младшие биты старшие байта единицы – запрос широковещательный и адрес назначен вручную, а не производителем. Источник, младшие биты старшего байта нули – запрос к конкретному уникальному адресу и адрес соответствует тому, что назначен производителем.

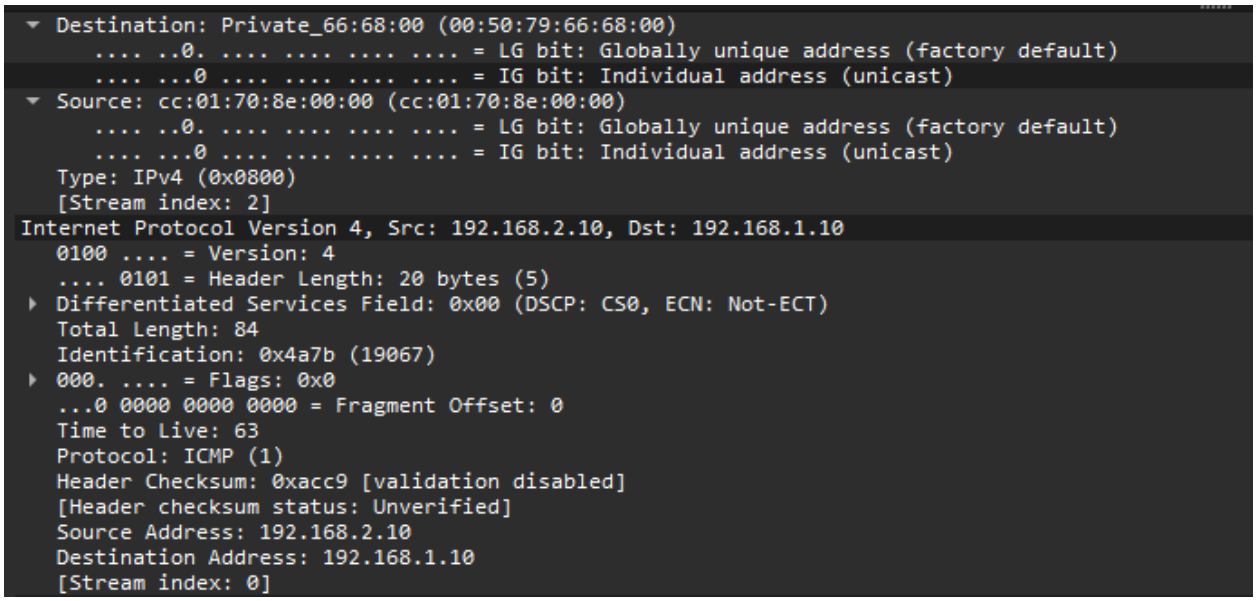
Протокол ARP, индекс потока 2. Padding – байты добавленные до минимального размера, контрольная сумма не проверялась.



ARP – ответ. 60 байт было передано. (Контрольная сумма не учтена). Источник MAC-адрес для сети с PC1. Цель – MAC-адрес PC1. В обоих случаях младшие биты старшего байта равны 0 – передача данных по одному конкретному адресу и адреса соответствуют адресам, заданным производителем.



ICMP запрос – передано 98. 20 IP-заголовок + 8 ICMP-заголовок + 14 байт Ethernet-заголовок – 56 байт полезной нагрузки. На Ethernet-уровне источник – MAC адрес PC1. Цель – интерфейс маршрутизатора в сети с PC1. Младшие биты – 0. Передача по протоколу IPv4. IPv4 протокол: Источник – IP PC1, цель – IP PC2. То есть на Ethernet направление пакета – к интерфейсу маршрутизатора, по IP-протоколу – до PC2. TTL 64 -время жизни 64 хопа.



ICMP ответ – 98 байтов. На Ethernet-протоколе источник – интерфейс маршрутизатора, а для IP-протокола – PC2. Цель для обоих случае PC1 (Так как это ответ PC2 узнал IP адрес, которому нужно направить пакет). Младшие биты 0 в Ethernet. Был сообщён протокол IP. Длина заголовка 20 байт. Нет флагов (пакет можно фрагментировать, это последний фрейм пакета), сдвиг для фрейма – 0 (дошёл целиком). TTL 63 – один хоп прохождение через маршрутизатор.

НА PC2 APR и ICMP запросы-ответы идентичны, только на Ethernet источник и цель меняются с MAC PC2 на MAC интерфейса маршрутизатора в сети с PC2.