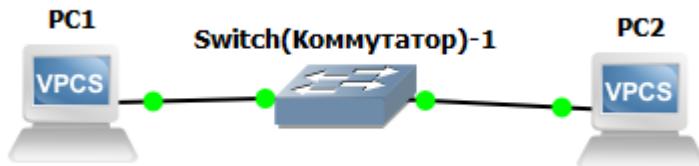


- 1) Установить и настроить эмулятор GNS3
- 2) Создать простейшую сеть, состоящую из 1 коммутатора и 2 компьютеров, назначить им произвольные ip адреса из одной сети



Для РС 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 РС2 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 РС1, введя команду ping 192.168.12.6

```

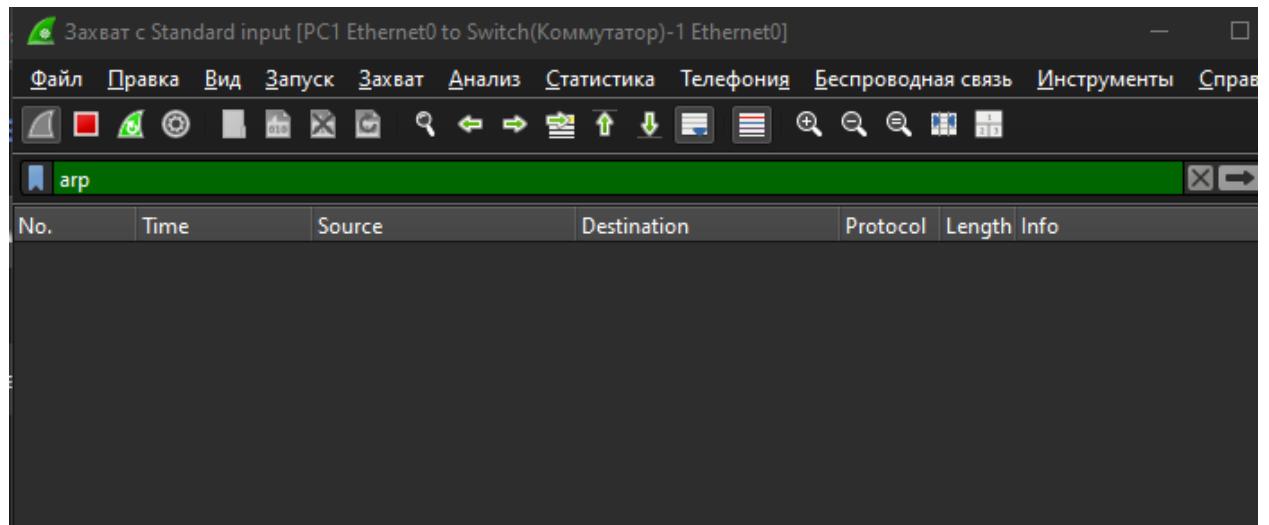
PC1> ping 192.168.12.6

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

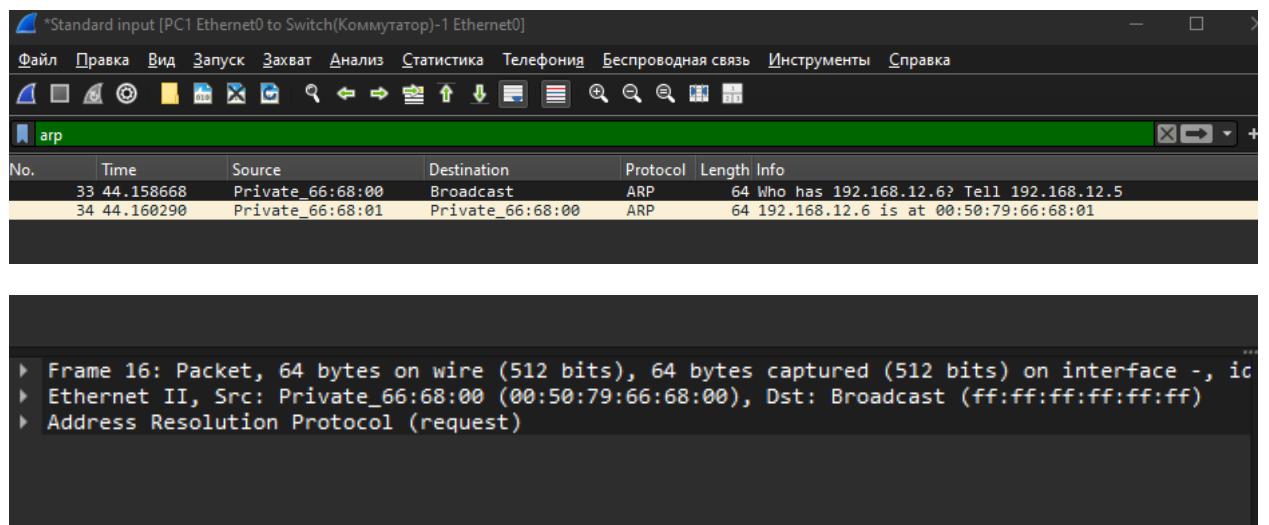
Нажимаем на связь PC1 и коммутатора и начинаем захват, применяем фильтр. То же самое со связью PC2 и коммутатора.



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:



```
▶ Frame 34: Packet, 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, interface index 1, link layer type 802.3 (Ethernet), duration 1.6029000000000002 ms, source Private_66:68:01 (00:50:79:66:68:01), destination Private_66:68:00 (00:50:79:66:68:00)
  Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: Private_66:68:00 (00:50:79:66:68:00) [oui-registered]
  Address Resolution Protocol (reply)
```

Захвачено 2 пакета протокола ARP:

ПАКЕТ №33 (ARP Request):

- Время: 44.158668 сек

- Ethernet заголовок:

Назначение: Broadcast (ff:ff:ff:ff:ff:ff)

Источник: 00:50:79:66:68:00 (PC1)

Тип: 0x0806 (ARP)

- ARP заголовок:

Opcode: 1 (request)

Sender: 192.168.12.5, Sender MAC: 00:50:79:66:68:00

Destination: Broadcast (ff:ff:ff:ff:ff:ff) - широковещательный адрес

ПАКЕТ №34 (ARP Reply):

- Время: 44.160290 сек

- Ethernet заголовок:

Назначение: 00:50:79:66:68:00 (на PC1)

Источник: 00:50:79:66:68:01 (PC2)

ARP заголовок:

Opcode: 2 (reply)

Sender : 192.168.12.6, Sender MAC: 00:50:79:66:68:01

Destination: 192.168.12.5, Target MAC: 00:50:79:66:68:00

Теперь сделаем то же самое, только будем анализировать связь PC2 с коммутатором.

| arp  |           |                  |                  |          |        |   |
|--|-----------|------------------|------------------|----------|--------|---|
| No.  | Time      | Source           | Destination      | Protocol | Length | Info                                    |
| 16   | 20.163005 | Private_66:68:00 | Broadcast        | ARP      | 64     | Who has 192.168.12.6? Tell 192.168.12.5 |
| 17   | 20.163077 | Private_66:68:01 | Private_66:68:00 | ARP      | 64     | 192.168.12.6 is at 00:50:79:66:68:01    |
| <pre>Frame 16: Packet, 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, interface Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff) Address Resolution Protocol (request)</pre>      |           |                  |                  |          |        |   |
| <pre>Frame 17: Packet, 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, interface Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: Private_66:68:00 (00:50:79:66:68:00) Address Resolution Protocol (reply)</pre> |           |                  |                  |          |        |   |

#### ПАКЕТ №16 (ARP Request):

- Время: 20.163005 сек

- Ethernet заголовок:

Назначение: Broadcast (ff:ff:ff:ff:ff:ff)

Источник: 00:50:79:66:68:00 (PC1)

Тип: 0x0806 (ARP)

- ARP заголовок:

Opcode: 1 (request)

Sender: 192.168.12.5, Sender MAC: 00:50:79:66:68:00

Destination: Broadcast (ff:ff:ff:ff:ff:ff) - широковещательный адрес

#### ПАКЕТ №17 (ARP Reply):

- Время: 20.163077 сек

- Ethernet заголовок:

Назначение: 00:50:79:66:68:00 (на PC1)

Источник: 00:50:79:66:68:01 (PC2)

ARP заголовок:

Opcode: 2 (reply)

Sender : 192.168.12.6, Sender MAC: 00:50:79:66:68:01

Destination: 192.168.12.5, Target MAC: 00:50:79:66:68:00

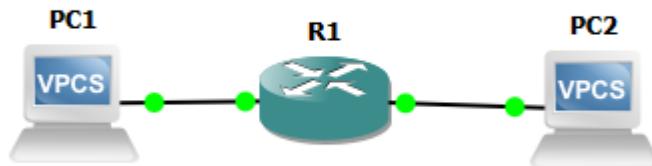
PC1 отправляет broadcast-запрос для поиска MAC-адреса PC2

PC2 отвечает unicast-сообщением с указанием своего MAC

Время отклика: ~0.0016 сек (1.6 мс)

Коммутатор трафик на все порты при неизвестных MAC (при использовании arp)

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

trace 192.168.2.10

ping 192.168.2.10

```
PC1> ping 192.168.2.10

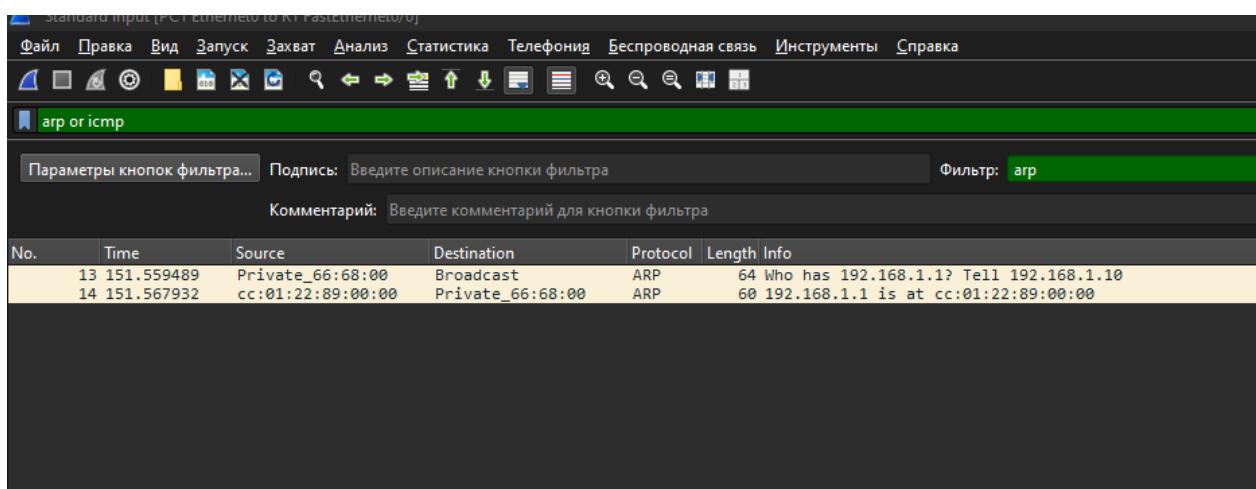
84 bytes from 192.168.2.10 icmp_seq=1 ttl=63 time=14.658 ms
84 bytes from 192.168.2.10 icmp_seq=2 ttl=63 time=16.046 ms
84 bytes from 192.168.2.10 icmp_seq=3 ttl=63 time=15.537 ms
192.168.2.10 icmp_seq=4 timeout
84 bytes from 192.168.2.10 icmp_seq=5 ttl=63 time=17.516 ms

PC1>
```

7) Перехватить трафик протокола arp и icmp на всех линках(нб!), 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

На PC1



Пакет 13: ARP-запрос от PC1

PC1 (192.168.1.10) ищет MAC-адрес своего шлюза (192.168.1.1).

Пакет 14: ARP-ответ от роутера

Роутер отвечает PC1, сообщая свой MAC-адрес.

На PC2

| No. | Time       | Source       | Destination  | Protocol | Length | Info   |
|-----|------------|--------------|--------------|----------|--------|--|
| 19  | 151.567927 | 192.168.1.10 | 192.168.2.10 | ICMP     | 98     | Echo (ping) request id=0xedaa7, seq=1/256, ttl=63 (reply in 22)  |
| 22  | 151.578606 | 192.168.2.10 | 192.168.1.10 | ICMP     | 98     | Echo (ping) reply id=0xedaa7, seq=1/256, ttl=64 (request in 19)  |
| 23  | 152.594349 | 192.168.1.10 | 192.168.2.10 | ICMP     | 98     | Echo (ping) request id=0xeea7, seq=2/512, ttl=63 (reply in 24)   |
| 24  | 152.594438 | 192.168.2.10 | 192.168.1.10 | ICMP     | 98     | Echo (ping) reply id=0xeea7, seq=2/512, ttl=64 (request in 23)   |
| 25  | 153.610516 | 192.168.1.10 | 192.168.2.10 | ICMP     | 98     | Echo (ping) request id=0xefa7, seq=3/768, ttl=63 (reply in 26)   |
| 26  | 153.610602 | 192.168.2.10 | 192.168.1.10 | ICMP     | 98     | Echo (ping) reply id=0xefa7, seq=3/768, ttl=64 (request in 25)   |
| 27  | 154.626895 | 192.168.1.10 | 192.168.2.10 | ICMP     | 98     | Echo (ping) request id=0xf0a7, seq=4/1024, ttl=63 (reply in 28)  |
| 28  | 154.626983 | 192.168.2.10 | 192.168.1.10 | ICMP     | 98     | Echo (ping) reply id=0xf0a7, seq=4/1024, ttl=64 (request in 27)  |
| 30  | 155.643396 | 192.168.1.10 | 192.168.2.10 | ICMP     | 98     | Echo (ping) request id=0xfa1a7, seq=5/1280, ttl=63 (reply in 31) |
| 31  | 155.643484 | 192.168.2.10 | 192.168.1.10 | ICMP     | 98     | Echo (ping) reply id=0xfa1a7, seq=5/1280, ttl=64 (request in 30) |

Пакет 19: ICMP Echo Request

PC1 отправляет ping на PC2. TTL=64

Пакет 22: ICMP Echo Reply

Ответ от PC2. TTL=63 (уменьшился на 1!) — это доказательство прохождения через роутер.

Destination: cc:01:22:89:00:00 (MAC роутера)

Source: 00:50:79:66:68:00 (MAC PC1)

Type: 0x0800 (IPv4)

Version: 4

Header Length: 20 bytes

TTL: 64

Protocol: 1 (ICMP)

Source: 192.168.1.10

Destination: 192.168.2.10

Type: 8 (Echo Request)

Code: 0

Checksum: 0xc627

Identifier: ...

Sequence: 1/256