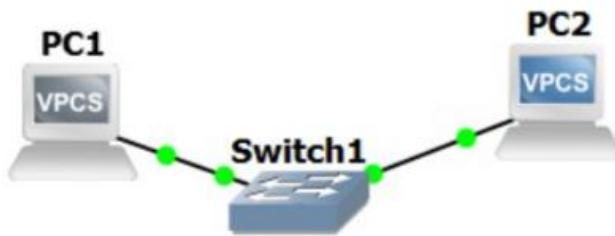


Лаба 1 Eltex

Уженцева Анна

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

Вот такая сеть получилась:



Назначим ip адреса

```
PC1> ip 192.168.1.10/24
Checking for duplicate address...
PC1 : 192.168.1.10 255.255.255.0
```

```
PC1> █
```

```
PC2> ip 192.168.1.20/24
Checking for duplicate address...
PC1 : 192.168.1.20 255.255.255.0
```

```
PC2> █
```

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

```
PC1> ping 192.168.1.20
84 bytes from 192.168.1.20 icmp_seq=1 ttl=64 time=1.551 ms
84 bytes from 192.168.1.20 icmp_seq=2 ttl=64 time=1.707 ms
84 bytes from 192.168.1.20 icmp_seq=3 ttl=64 time=1.523 ms
84 bytes from 192.168.1.20 icmp_seq=4 ttl=64 time=1.884 ms
84 bytes from 192.168.1.20 icmp_seq=5 ttl=64 time=1.314 ms
```

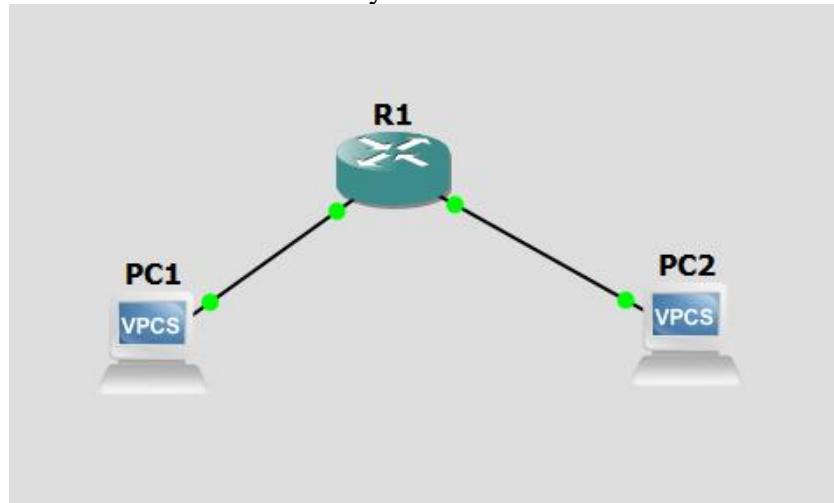
4. Перехватить трафик протокола arp на всех линках(nb!), задокументировать и проанализировать заголовки пакетов в программе Wireshark, для фильтрации трафика, относящегося к указанному протоколу использовать фильтры Wireshark

1 0.000000	Private_66:68:00	Broadcast	ARP	64 Who has 192.168.1.20? Tell 192.168.1.10
2 0.000150	Private 66:68:01	Private 66:68:00	ARP	64 192.168.1.20 is at 00:50:79:66:68:01

Здесь мы можем видеть два типа пакетов: запрос (who has...) поступает от PC1, спрашивает у кого адрес 192.168.1.10 и пишет свой адрес и ответ (... is at ..) от компьютера PC2 с ip и MAC адресами.

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

Вот такая сеть в итоге получилась:



Теперь настроим компьютеры и маршрутизатор:

```

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)#exit
R1(config)#
*Mar 1 00:02:23.739: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:02:24.739: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config)#interface FastEthernet0/1
^
% Invalid input detected at '^' marker.

R1(config)# interface FastEthernet0/1
^
% Invalid input detected at '^' marker.

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)#exit
*Mar 1 00:03:51.175: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
*Mar 1 00:03:52.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
R1(config-if)#exit
R1(config)#end
R1#
*Mar 1 00:03:56.459: %SYS-5-CONFIG_I: Configured from console by console
R1#write memory

```

```

PC1> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
PC1 : 192.168.1.10 255.255.255.0 gateway 192.168.1.1
1

```

```

PC2> ip 192.168.2.10/24 192.168.2.1
Checking for duplicate address...
PC2 : 192.168.2.10 255.255.255.0 gateway 192.168.2.1

```

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

```

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=20.548 ms
84 bytes from 192.168.2.10 icmp_seq=3 ttl=63 time=15.274 ms
84 bytes from 192.168.2.10 icmp_seq=4 ttl=63 time=15.338 ms
84 bytes from 192.168.2.10 icmp_seq=5 ttl=63 time=15.348 ms

PC1>

```

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

arp						
No.	Time	Source	Destination	Protocol	Length	Info
5	25.630864	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.10
6	25.641380	cc:01:57:ea:00:00	Private_66:68:00	ARP	60	192.168.1.1 is at cc:01:57:ea:00:00

PC1 спрашивает: "Кто имеет IP 192.168.1.1? Ответьте на мой MAC"

Роутер отвечает: "Я (192.168.1.1) имею MAC cc:01:1f:a4:00:20"

Второй компьютер:

arp						
No.	Time	Source	Destination	Protocol	Length	Info
4	25.651461	cc:01:57:ea:00:10	Broadcast	ARP	60	Who has 192.168.2.10? Tell 192.168.2.1
5	25.651530	Private_66:68:01	cc:01:57:ea:00:10	ARP	60	192.168.2.10 is at 00:50:79:66:68:01

Разберем icmp. Тут есть два типа пакетов: echo request и echo reply. Это запрос на ping и ответ на запрос.

PC1:

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
7	25.642505	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1504, seq=1/256, ttl=64 (no response found!)
8	27.643482	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1704, seq=2/512, ttl=64 (reply in 9)
9	27.663944	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1704, seq=2/512, ttl=63 (request in 8)
10	28.664511	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1804, seq=3/768, ttl=64 (reply in 11)
11	28.679692	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1804, seq=3/768, ttl=63 (request in 10)
12	29.689528	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1904, seq=4/1024, ttl=64 (reply in 13)
13	29.695790	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1904, seq=4/1024, ttl=63 (request in 12)
14	30.696573	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1a04, seq=5/1280, ttl=64 (reply in 15)
15	30.711868	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1a04, seq=5/1280, ttl=63 (request in 14)

PC2:

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
→ 6	27.653878	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1704, seq=2/512, ttl=63 (reply in 7)
← 7	27.653989	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1704, seq=2/512, ttl=64 (request in 6)
8	28.669662	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1804, seq=3/768, ttl=63 (reply in 9)
9	28.669838	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1804, seq=3/768, ttl=64 (request in 8)
10	29.685695	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1904, seq=4/1024, ttl=63 (reply in 11)
11	29.685760	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1904, seq=4/1024, ttl=64 (request in 10)
12	30.701784	192.168.1.10	192.168.2.10	ICMP	98	Echo (ping) request id=0x1a04, seq=5/1280, ttl=63 (reply in 13)
13	30.701824	192.168.2.10	192.168.1.10	ICMP	98	Echo (ping) reply id=0x1a04, seq=5/1280, ttl=64 (request in 12)