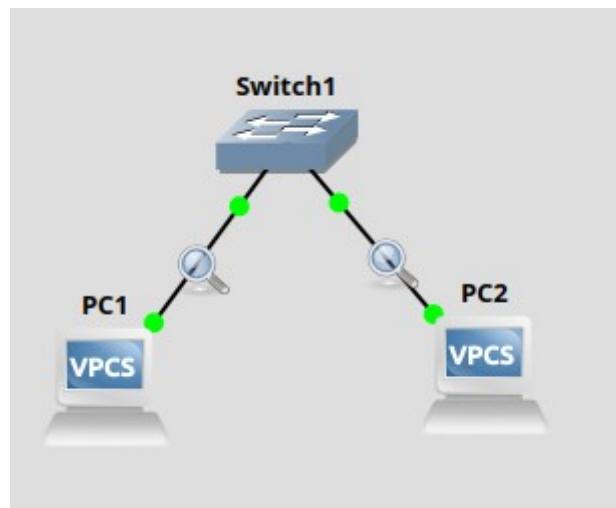


# Практика 1

## Коммутатор и 2ПК:

- Создал простую схему состоящую из коммутатора и двух ПК.



- Назначил произвольные адреса из одной сети

- У первого ПК ip 192.168.1.1 255.255.255.0

```
PC1> ip 192.168.1.1 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.1.1 255.255.255.0
```

- У второго ПК ip 192.168.1.2 255.255.255.0

```
PC2> ip 192.168.1.2 255.255.255.0
Checking for duplicate address...
PC2 : 192.168.1.2 255.255.255.0
```

- Выполнил команду ping с первого ПК, используя ip адрес второго ПК (ping 192.168.1.2)

```
PC1> ping 192.168.1.2
```

```
84 bytes from 192.168.1.2 icmp_seq=1 ttl=64 time=0.152 ms
84 bytes from 192.168.1.2 icmp_seq=2 ttl=64 time=0.267 ms
84 bytes from 192.168.1.2 icmp_seq=3 ttl=64 time=0.339 ms
84 bytes from 192.168.1.2 icmp_seq=4 ttl=64 time=0.226 ms
84 bytes from 192.168.1.2 icmp_seq=5 ttl=64 time=0.263 ms
```

4. Перехватил трафик протокола arp на всех линках с помощью Wireshark

Трафик от первого ПК до коммутатора

arp						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:50:79:66:68:00	Broadcast	ARP	64	Who has 192.168.1.2? Tell 192.168.1.1
2	0.000160	00:50:79:66:68:01	00:50:79:66:68:00	ARP	64	192.168.1.2 is at 00:50:79:66:68:01

Трафик от коммутатора до второго ПК

arp						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:50:79:66:68:00	Broadcast	ARP	64	Who has 192.168.1.2? Tell 192.168.1.1
2	0.000132	00:50:79:66:68:01	00:50:79:66:68:00	ARP	64	192.168.1.2 is at 00:50:79:66:68:01

## Анализ трафика

### 1. Исходное состояние

- ПК1 (IP: 192.168.1.1, MAC: 00:50:79:66:68:00) хочет отправить данные на ПК2 (IP: 192.168.1.2)
- ПК1 проверяет свой ARP-кеш, но не находит MAC-адрес для 192.168.1.2
- Для установления соединения необходим MAC-адрес получателя

### 2. Формирование ARP-запроса:

ПК1 создает ARP Request пакет:

- Отправитель: IP 192.168.1.1, MAC 00:50:79:66:68:00
- Получатель: IP 192.168.1.2, MAC: неизвестен (00:00:00:00:00:00)
- Тип операции: request (1)

На канальном уровне пакет помещается в Ethernet-фрейм:

- Source MAC: 00:50:79:66:68:00
- Destination MAC: ff:ff:ff:ff:ff:ff (широковещательный адрес)

```

▼ Ethernet II, Src: 00:50:79:66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: 00:50:79:66:68:00 (00:50:79:66:68:00)
  Type: ARP (0x0806)
  Padding: 00000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]

```

### 3. Работа коммутатора с ARP-запросом:

- Коммутатор получает фрейм на порту, подключенном к ПК1
- Так как destination MAC = ff:ff:ff:ff:ff:ff, коммутатор:
  - ✓ Заносит в свою MAC-таблицу запись: Порт 1 → MAC 00:50:79:66:68:00
  - ✓ Отправляет фрейм на все активные порты, кроме порта-источника
  - ARP Request достигает всех устройств в сети

```

▼ Address Resolution Protocol (request)                                Broadcast (ff:ff:ff:ff:ff:ff)
  Hardware type: Ethernet (1)                                         | 
  Protocol type: IPv4 (0x0800)                                         | 
  Hardware size: 6                                                 | 
  Protocol size: 4                                                 | 
  Opcode: request (1)                                              | 
  Sender MAC address: 00:50:79:66:68:00 (00:50:79:66:68:00)    | 
  Sender IP address: 192.168.1.1                                     | 
  Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)                  | 
  Target IP address: 192.168.1.2                                     | 

```

### 4. Получение ARP-запроса 2ПК:

- 2ПК получает широковещательную посылку
- Узнает свой ip-адрес

```

▼ Ethernet II, Src: 00:50:79:66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: 00:50:79:66:68:00 (00:50:79:66:68:00)
  Type: ARP (0x0806)
  Padding: 00000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]

▼ Address Resolution Protocol (request)                                Broadcast (ff:ff:ff:ff:ff:ff)
  Hardware type: Ethernet (1)                                         | 
  Protocol type: IPv4 (0x0800)                                         | 
  Hardware size: 6                                                 | 
  Protocol size: 4                                                 | 
  Opcode: request (1)                                              | 
  Sender MAC address: 00:50:79:66:68:00 (00:50:79:66:68:00)    | 
  Sender IP address: 192.168.1.1                                     | 
  Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)                  | 
  Target IP address: 192.168.1.2                                     | 

```

### 5. Формирование ответа от 2ПК:

ПК2 (MAC: 00:50:79:66:68:01) создает ARP Reply:

- Отправитель: IP 192.168.1.2, MAC 00:50:79:66:68:01
- Получатель: IP 192.168.1.1, MAC 00:50:79:66:68:00

- Тип операции: reply

Ethernet-фрейм:

- Source MAC: 00:50:79:66:68:01
- Destination MAC: 00:50:79:66:68:00

```

▼ Ethernet II, Src: 00:50:79:66:68:01 (00:50:79:66:68:01), Dst: 00:50:79:66:68:00 (00:50:79:66:68:00)
  ▶ Destination: 00:50:79:66:68:00 (00:50:79:66:68:00)
  ▶ Source: 00:50:79:66:68:01 (00:50:79:66:68:01)
    Type: ARP (0x0806)
    Padding: 00000000000000000000000000000000
    Frame check sequence: 0x00000000 [unverified]
    [FCS Status: Unverified]
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 00:50:79:66:68:01 (00:50:79:66:68:01)
  Sender IP address: 192.168.1.2
  Target MAC address: 00:50:79:66:68:00 (00:50:79:66:68:00)
  Target IP address: 192.168.1.1

```

## 6. Работа коммутатора с ARP-ответом:

- Коммутатор получает фрейм от ПК2
- Заносит в MAC-таблицу: Порт 2 → MAC 00:50:79:66:68:01
- Проверяет destination MAC 00:50:79:66:68:00
- Находит в таблице: MAC 00:50:79:66:68:00 → Порт 1
- Пересыпает фрейм только на порт 1 (к ПК1)

## 7. Завершение процесса:

- ПК1 получает ARP Reply
- Обновляет ARP-кеш: 192.168.1.2 → 00:50:79:66:68:01
- Теперь может отправлять данные напрямую ПК2

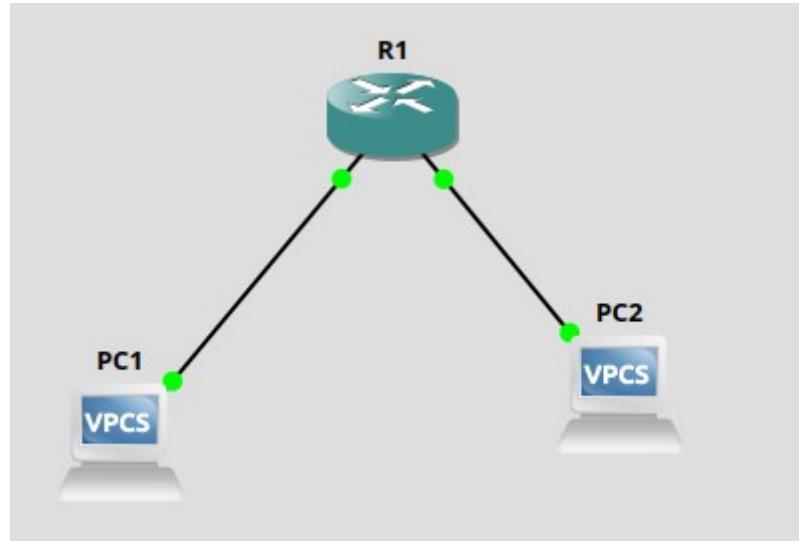
```

▼ Ethernet II, Src: 00:50:79:66:68:01 (00:50:79:66:68:01), Dst: 00:50:79:66:68:00 (00:50:79:66:68:00)
  ▶ Destination: 00:50:79:66:68:00 (00:50:79:66:68:00)
  ▶ Source: 00:50:79:66:68:01 (00:50:79:66:68:01)
    Type: ARP (0x0806)
    Padding: 00000000000000000000000000000000
    Frame check sequence: 0x00000000 [unverified]
    [FCS Status: Unverified]
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 00:50:79:66:68:01 (00:50:79:66:68:01)
  Sender IP address: 192.168.1.2
  Target MAC address: 00:50:79:66:68:00 (00:50:79:66:68:00)
  Target IP address: 192.168.1.1

```

## Маршрутизатор и 2ПК:

1. Создал схему, состоящую из маршрутизатора и 2ПК. ПК подключены в порты FastEthernet0/0 и FastEthernet1/0.



2. Настроил интерфейсы портов на маршрутизаторе:

configure terminal

для ПК1:

- interface FastEthernet0/0
- ip address 192.168.1.1 255.255.255.0
- no shutdown
- description Link to PC1
- exit

для ПК2:

- interface FastEthernet1/0
- ip address 192.168.2.1 255.255.255.0
- no shutdown
- description Link to PC2
- exit

do show ip interface brief

R1(config)#do 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

end

write memory

3. Настроил ip-адреса на ПК:

для ПК1 (ip 192.168.1.10, маска 255.255.255.0, шлюз 192.168.1.1):

- ip 192.168.1.10 255.255.255.0 192.168.1.1
- show ip

```
PC1> show ip

NAME      : PC1[1]
IP/MASK   : 192.168.1.10/24
GATEWAY   : 192.168.1.1
DNS       :
MAC       : 00:50:79:66:68:01
LPORT     : 20028
RHOST:PORT : 127.0.0.1:20029
MTU       : 1500
```

для ПК2 (ip 192.168.2.20, маска 255.255.255.0, шлюз 192.168.2.1):

- ip 192.168.2.20 255.255.255.0 192.168.2.1
- show ip

4. Перехватил трафик протокола arp и icmp на всех линках с помощью Wireshark

```
PC2> show ip

NAME      : PC2[1]
IP/MASK   : 192.168.2.20/24
GATEWAY   : 192.168.2.1
DNS       :
MAC       : 00:50:79:66:68:00
LPORT     : 20030
RHOST:PORT : 127.0.0.1:20031
MTU       : 1500
```

ping 192.168.2.20

```

PC1> ping 192.168.2.20

192.168.2.20 icmp_seq=1 timeout
84 bytes from 192.168.2.20 icmp_seq=2 ttl=63 time=11.179 ms
84 bytes from 192.168.2.20 icmp_seq=3 ttl=63 time=15.743 ms
84 bytes from 192.168.2.20 icmp_seq=4 ttl=63 time=15.998 ms
84 bytes from 192.168.2.20 icmp_seq=5 ttl=63 time=15.047 ms

```

На ПК1:

No.	Time	Source	Destination	Protocol	Length	Info
2	21.837563	00:50:79:66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.10
3	21.844216	cc:01:16:34:00:00	00:50:79:66:68:01	ARP	60	192.168.1.1 is at cc:01:16:34:00:00
7	57.337565	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0x488c, seq=1/256, ttl=64 (no response...)
8	59.338070	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0x488c, seq=2/512, ttl=64 (reply in 9)
9	59.355960	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0x488c, seq=2/512, ttl=63 (request in 8)
10	60.357210	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0x488c, seq=3/768, ttl=64 (reply in 11)
11	60.372500	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0x498c, seq=3/768, ttl=63 (request in ...)
12	61.373132	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0x4a8c, seq=4/1024, ttl=64 (reply in 13)
13	61.385959	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0x4a8c, seq=4/1024, ttl=63 (request in ...)
14	62.386989	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0xb8c, seq=5/1280, ttl=64 (reply in 15)
15	62.402011	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0xb8c, seq=5/1280, ttl=63 (request in ...)

На ПК2:

No.	Time	Source	Destination	Protocol	Length	Info
5	57.301995	cc:01:16:34:00:10	Broadcast	ARP	60	Who has 192.168.2.20? Tell 192.168.2.1
6	57.302130	00:50:79:66:68:00	cc:01:16:34:00:10	ARP	60	192.168.2.20 is at 00:50:79:66:68:00
7	59.305562	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0x488c, seq=2/512, ttl=63 (reply in 8)
8	59.305765	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0x488c, seq=2/512, ttl=64 (request in 7)
9	60.322096	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0x498c, seq=3/768, ttl=63 (reply in 10)
10	60.322282	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0x498c, seq=3/768, ttl=64 (request in 9)
11	61.335578	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0xa8c, seq=4/1024, ttl=63 (reply in 12)
12	61.335762	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa8c, seq=4/1024, ttl=64 (request in ...)
13	62.351628	192.168.1.10	192.168.2.20	ICMP	98	Echo (ping) request id=0xb8c, seq=5/1280, ttl=63 (reply in 14)
14	62.351792	192.168.2.20	192.168.1.10	ICMP	98	Echo (ping) reply id=0xb8c, seq=5/1280, ttl=64 (request in ...)

### Шаг 1: ПК1 определяет получателя

- ПК1 (192.168.1.10) хочет отправить пакет на ПК2 (192.168.2.20)
- ПК1 проверяет находится ли ПК2 в одной сети.
- ПК1 отправляя на шлюз 192.168.1.1

### Шаг 2: ПК1 ищет MAC шлюза (ARP)

- ПК1 отправляет ARP Request кто имеет 192.168.1.1
- Маршрутизатор отвечает, MAC: cc:01:16:34:00:00

▼ Ethernet II, Src: 00:50:79:66:68:01 (00:50:79:66:68:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Source: 00:50:79:66:68:01 (00:50:79:66:68:01)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
Frame check sequence: 0x00000000 [unverified]
[FCS Status: Unverified]
▼ Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: 00:50:79:66:68:01 (00:50:79:66:68:01)
Sender IP address: 192.168.1.10
Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
Target IP address: 192.168.1.1

### Шаг 3: ПК1 отправляет ICMP на маршрутизатор

- Ethernet II, Src: 00:50:79:66:68:01 (00:50:79:66:68:01), Dst: cc:01:16:34:00:00 (cc:01:16:34:00:00)

▼ Ethernet II, Src: 00:50:79:66:68:01 (00:50:79:66:68:01), Dst: cc:01:16:34:00:00 (cc:01:16:34:00:00)  
  ↳ Destination: cc:01:16:34:00:00 (cc:01:16:34:00:00)  
  ↳ Source: 00:50:79:66:68:01 (00:50:79:66:68:01)  
    Type: IPv4 (0x0800)

- Первый пакет теряется из-за того, что маршрутизатор не знает MAC адрес ПК2.

```
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x188a [correct]
    [Checksum Status: Good]
    Identifier (BE): 1921 (0x0781)
    Identifier (LE): 33031 (0x8107)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
[No response seen]
  [Expert Info (Warning/Sequence): No response seen to ICMP request]
Data (56 bytes)
  Data: 08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334
  [Length: 56]
```

#### Шаг 4: Маршрутизатор ищет MAC ПК2 (ARP)

- Маршрутизатор высылает широковещательный адресс ищет 192.168.2.20
  - ПК2 отправляет ответ с MAC адрессом

```
‐ Ethernet II, Src: cc:01:16:34:00:10 (cc:01:16:34:00:10), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ↳ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ↳ Source: cc:01:16:34:00:10 (cc:01:16:34:00:10)
    Type: ARP (0x0806)
    Padding: 0000000000000000000000000000000000000000000000000000000000000000
‐ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: cc:01:16:34:00:10 (cc:01:16:34:00:10)
  Sender IP address: 192.168.2.1
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.2.20
Ищет 2 ПК
```

```
‐ Ethernet II, Src: 00:50:79:66:68:00 (00:50:79:66:68:00), Dst: cc:01:16:34:00:10 (cc:01:16:34:00:10)
  ▶ Destination: cc:01:16:34:00:10 (cc:01:16:34:00:10)
  ▶ Source: 00:50:79:66:68:00 (00:50:79:66:68:00)
  ▶ Type: ARP (0x0806)
  ▶ Padding: 00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
‐ Address Resolution Protocol (reply)
  ▶ Hardware type: Ethernet (1)
  ▶ Protocol type: IPv4 (0x0800)
  ▶ Hardware size: 6
  ▶ Protocol size: 4
  ▶ Opcode: reply (2)
  ▶ Sender MAC address: 00:50:79:66:68:00 (00:50:79:66:68:00)
  ▶ Sender IP address: 192.168.2.20
  ▶ Target MAC address: cc:01:16:34:00:10 (cc:01:16:34:00:10)
  ▶ Target IP address: 192.168.2.1
```

### Шаг 5: Маршрутизатор пересыпает пакет

- Меняет MAC-адреса:
  - Ethernet: Src=Router\_MAC\_Fa1/0, Dst=PC2\_MAC
  - Сохраняет IP-адреса
  - Отправляет на ПК2

```
▶ Ethernet II, Src: 00:50:79:66:68:01 (00:50:79:66:68:01), Dst: cc:01:16:34:00:00 (cc:01:16:34:00:00)
└ Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.2.20
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x8108 (33032)
  ▶ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x7532 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.10
    Destination Address: 192.168.2.20
```

## **Вывод:**

## **Сеть с коммутатором:**

- Устройства в одной сети общаются напрямую
  - ARP разрешает MAC-адреса в пределах подсети
  - Коммутатор пересыпает фреймы на основе MAC-таблицы
  - Ping работает мгновенно

### **Сеть с маршрутизатором:**

- Маршрутизатор разделяет сети на разные подсети
  - Для межсетевой связи нужны шлюзы по умолчанию
  - Требуется отдельное ARP разрешение в каждой подсети

- Первый ping-пакет теряется на ARP-разрешение

### Итог:

Коммутатор работает на L2 (MAC-адреса), маршрутизатор на L3 (IP-адреса). ARP важен для обеих конфигураций, но работает по-разному.