

# Маршрутизация. Фрагментация

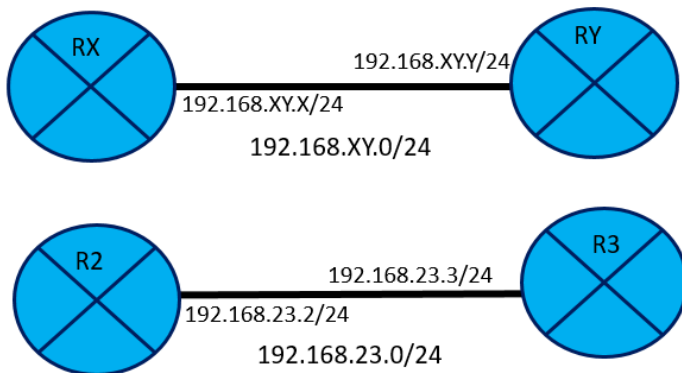
*Как определить, что устройство принадлежит подсети?*

Берём адрес устройства, адрес подсети и накладываем на них маску. Если сетевые части совпадают, то устройство принадлежит этой подсети.

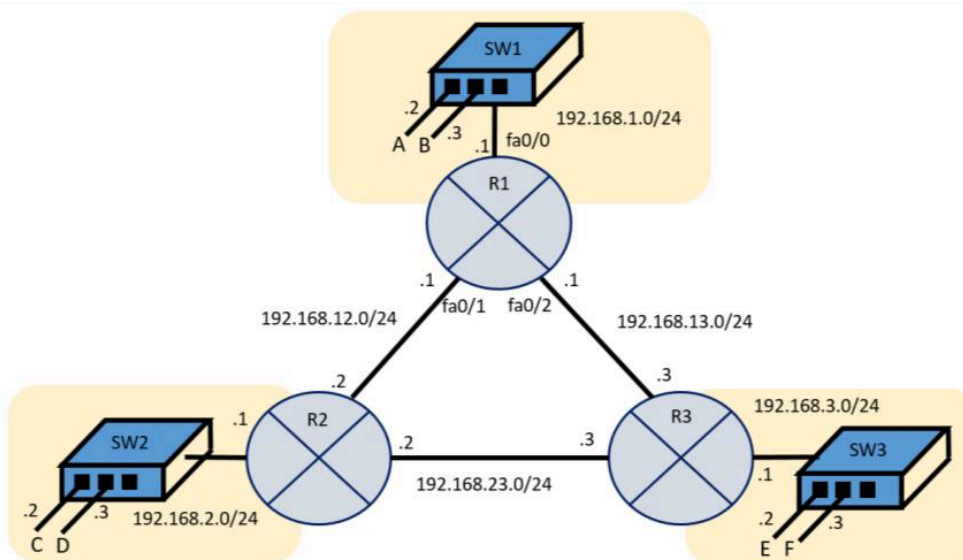
*Как определить, что два узла принадлежат одной подсети?*

Берём их адреса, накладываем на них маску подсети и если сетевая часть совпадает, то они принадлежат одной подсети.

*Как формируются адреса у роутеров?*



Для соседних роутеров  $R_x$  и  $R_y$  формируют адреса по следующему правилу: 192.168.XY.X|Y, где  $X < Y$ .



Пусть A хочет переслать данные C. A берёт свой IP-адрес, свою маску и IP-адрес C. Накладывая свою маску на свой адрес и адрес C понимает, что C в другой подсети. Тогда он отправляет данные роутеру R1 согласно своей таблице маршрутизации (там есть два

поля -- запись о подсети, куда мы подключены, и маршрут по умолчанию). R1 отправляет данные R2. R2, видя, что C в подсети, подключённой к одному из его интерфейсов, пересылает данные C.

*Таблица маршрутизации роутера R1*

C	192.168.1.0/24	fa0/0	Это те сети, в которые подключены интерфейсы на R1: в сторону пользователей, в сторону R2 и в сторону R3. Эти сети попадают в таблицу маршрутизации автоматически после того, как мы их включили, настроили IP-адреса и указали маски.
C	192.168.12.0/24	fa0/1	
C	192.168.13.0/24	fa0/2	
S	192.168.2.0/24	192.168.12.2	<u>Эти записи администратор добавляет вручную:</u> он записывает маршруты до сетей, расположенных за роутерами R2 и R3. Здесь нельзя указать интерфейс в третьей колонке: необходимо указать IP-адрес устройства, которому нужно передать данные, чтобы достичь нужной подсети <sup>41</sup> .
S	192.168.3.0/24	192.168.13.3	

В первой колонке -- откуда появилась информация в таблице:

- **C (connected)** -- сети, в которые подключены интерфейсы роутера
- **S (static)** -- статические маршруты, записи добавлены вручную

Когда R1 получает пакет от A, он не знает масок A и C. Поэтому:

1. Выделяем сетевую часть у каждой записи в таблице
2. На IP-адрес получателя накладываем маску сети из каждой записи и сравниваем

1	C	192.168.1.0/24	11000000	10101000	00000001	00000000
2	C	192.168.12.0/24	11000000	10101000	00001100	00000000
3	C	192.168.13.0/24	11000000	10101000	00001101	00000000
4	S	192.168.2.0/24	11000000	10101000	00000010	00000000
5	S	192.168.3.0/24	11000000	10101000	00000011	00000000

Понимаем, что нужно отправить данные роутеру R2 -- ему соответствует вторая строка таблицы маршрутизации, значит, R1 отправит данные через fa0/1.

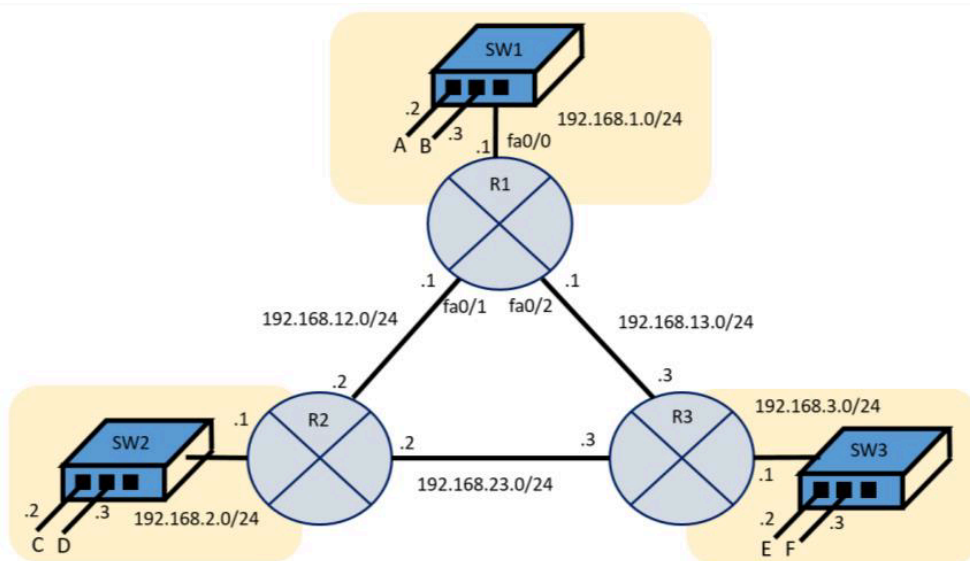
✍ Если в таблице маршрутизации подходит несколько записей, выбираем запись с наиболее длинной маской

Разные источники могут попытаться впихнуть в таблицу одинаковые данные. На этот случай вводят следующее понятие:

**AD** -- число, определяющее степень доверия источнику.

- C -- AD = 0 (самые доверенные)
- S -- AD = 1
- OSPF -- AD = 110
- RIP -- AD = 120

Допустим, администратор ошибся: на R1 написал маршрут на сеть 192.168.2.0/24, на R2 -- маршрут на эту же подсеть в сторону R3, на R3 -- маршрут на эту же подсеть в сторону R1.



Благодаря AD проблемы не возникает.

Но если у R2 упадёт левый интерфейс -- всё сломается. Пакет из A в C начнёт бегать по кругу R1 >>> R2 >>> R3 >>> R1. Для защиты от этого используют поле TTL.

## Фрагментация

**MTU** -- максимальный объем данных, который может быть отправлен по сети без дальнейшего фрагментирования (одним пакетом).

**Фрагментация** -- деление исходного IP-пакета на два или более фрагмента, так, чтобы в итоге эти фрагменты были меньше, чем MTU на канале.

Каждый фрагмент -- полноценный IP-пакет.

ID у каждого родительского пакета свой. У дочерних пакетов -- тот же ID. Благодаря Fragment Offset можем найти местоположение фрагмента в исходном пакете. Если один фрагмент пропал -- всё отбрасывается, пакет не собирается.

Сборка -- только на получателе.

*Фрагментация не предполагает одинакового разделения.*

2400 байт		
400 байт	1000 байт	1000 байт