

# Отчёт по лабораторной работе «IP-маршрутизация»

Trung Luong

4 октября 2020 г.

## Содержание

1. Топология сети	1
2. Назначение IP-адресов	1
3. Таблица маршрутизации	1
4. Проверка настройки сети	3
5. Маршрутизация	4
6. Продолжительность жизни пакета	5
7. Изучение IP-фрагментации	8
8. Отсутствие сети	9
9. Отсутствие IP-адреса в сети	9

## 1. Топология сети

Топология сети и используемые IP-адреса показаны на рис. 1.

## 2. Назначение IP-адресов

Ниже приведён файл настройки протокола IP маршрутизатора (указать, какого).

| Сюда нужно поместить характерный `/etc/network/interfaces` маршрутизатора

Ниже приведён файл настройки протокола IP рабочей станции (указать, какой).

| Сюда нужно поместить характерный `/etc/network/interfaces` рабочей станции

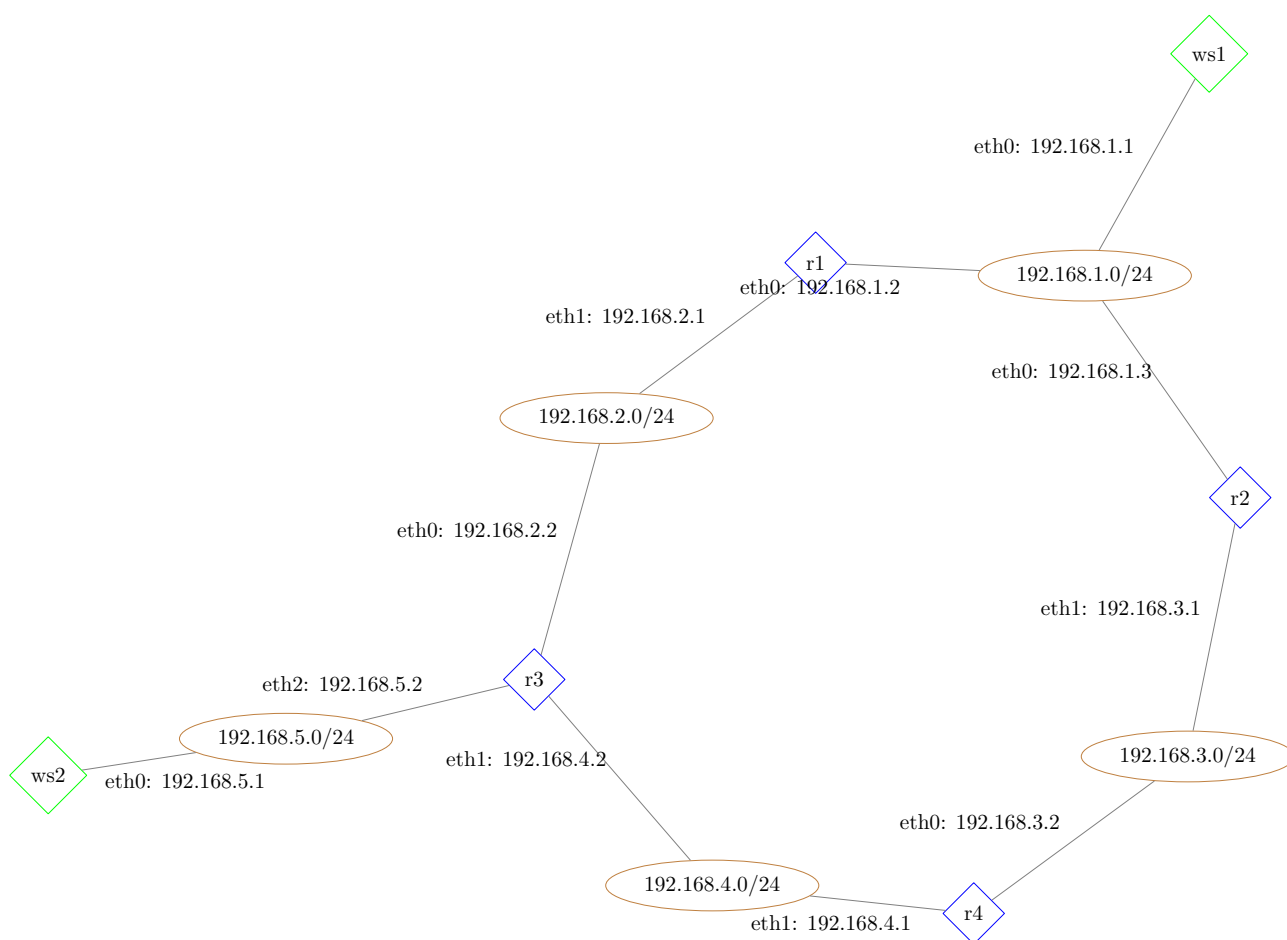


Рис. 1. Топология сети

### 3. Таблица маршрутизации

Вывести (командой `ip r`) таблицу маршрутизации для **r1**.

```
192.168.5.0/24 via 192.168.2.2 dev eth1
192.168.4.0/24 via 192.168.2.2 dev eth1
192.168.3.0/24 via 192.168.2.2 dev eth1
192.168.2.0/24 dev eth1 proto kernel scope link src 192.168.2.1
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.2
```

Вывести (командой `ip r`) таблицу маршрутизации для **r2**.

```
192.168.5.0/24 via 192.168.3.2 dev eth1
192.168.4.0/24 via 192.168.3.2 dev eth1
192.168.3.0/24 dev eth1 proto kernel scope link src 192.168.3.1
192.168.2.0/24 via 192.168.3.2 dev eth1
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.3
```

Вывести (командой `ip r`) таблицу маршрутизации для **r3**.

```
192.168.5.0/24 dev eth2 proto kernel scope link src 192.168.5.2
192.168.4.0/24 dev eth1 proto kernel scope link src 192.168.4.2
192.168.3.0/24 via 192.168.4.1 dev eth1
192.168.2.0/24 dev eth0 proto kernel scope link src 192.168.2.2
192.168.1.0/24 via 192.168.2.1 dev eth0
```

Вывести (командой `ip r`) таблицу маршрутизации для **r4**.

```
192.168.5.0/24 via 192.168.4.2 dev eth1
192.168.4.0/24 dev eth1 proto kernel scope link src 192.168.4.1
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.2
192.168.2.0/24 via 192.168.4.2 dev eth1
192.168.1.0/24 via 192.168.4.2 dev eth1
```

Вывести (командой `ip r`) таблицу маршрутизации для **ws1**.

```
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.1
default via 192.168.1.2 dev eth0
```

Вывести (командой `ip r`) таблицу маршрутизации для **ws2**.

```
192.168.5.0/24 dev eth0 proto kernel scope link src 192.168.5.1
default via 192.168.5.2 dev eth0
```

## 4. Проверка настройки сети

Вывод **tracert** от узла **ws1** такого-то до такого-то при нормальной работе сети.

```
ws1:~# tracert -n 192.168.5.1
tracert to 192.168.5.1 (192.168.5.1), 64 hops max, 40 byte packets
 1  192.168.1.2  1 ms  1 ms  1 ms
 2  192.168.2.2  12 ms  1 ms  1 ms
 3  192.168.5.1  12 ms  1 ms  1 ms
```

Вывод **tracert** от узла **ws2** такого-то до такого-то при нормальной работе сети.

```
ws2:~# tracert -n 192.168.1.1
tracert to 192.168.1.1 (192.168.1.1), 64 hops max, 40 byte packets
 1  192.168.5.2  1 ms  0 ms  1 ms
 2  192.168.2.1  1 ms  1 ms  1 ms
 3  192.168.1.1  1 ms  1 ms  1 ms
```

## 5. Маршрутизация

Тестирование на MAC-адреса интерфейс **r1**:

```
192.168.5.0/24 via 192.168.2.2 dev eth1
192.168.4.0/24 via 192.168.2.2 dev eth1
192.168.3.0/24 via 192.168.2.2 dev eth1
192.168.2.0/24 dev eth1  proto kernel  scope link  src 192.168.2.1
192.168.1.0/24 dev eth0  proto kernel  scope link  src 192.168.1.2
```

Тестирование на MAC-адреса интерфейс **r3**:

```
192.168.5.0/24 dev eth2  proto kernel  scope link  src 192.168.5.2
192.168.4.0/24 dev eth1  proto kernel  scope link  src 192.168.4.2
192.168.3.0/24 via 192.168.4.1 dev eth1
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.2
192.168.1.0/24 via 192.168.2.1 dev eth0
```

Для стирания кеша ARP, выполняется следующая команда:

```
ip n flush all
```

Далее показана отправка пакета на маршрутизатор (косвенная маршрутизация).

```
a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.1.2
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42: arp reply 192.168.1.2
a6:f9:52:b6:1e:69 > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: 192.168.1.1 > 192.168.5.1
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype IPv4 (0x0800), length 98: 192.168.5.1 > 192.168.1.1
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42: arp who-has 192.168.1.1
a6:f9:52:b6:1e:69 > 0e:ab:f8:0c:10:4b, ethertype ARP (0x0806), length 42: arp reply 192.168.1.1
```

```
fa:de:dc:30:96:57 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.2.2: ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype ARP (0x0806), length 42: arp reply 192.168.2.2: fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: 192.168.1.1 > 192.168.5.1: ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: 192.168.5.1 > 192.168.1.1: ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype ARP (0x0806), length 42: arp who-has 192.168.2.2: fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype ARP (0x0806), length 42: arp reply 192.168.2.2:
```

Затем маршрутизатор отправил его далее.

```
c2:57:e2:f3:3f:00 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.5.1: da:53:12:09:ea:4e > c2:57:e2:f3:3f:00, ethertype ARP (0x0806), length 42: arp reply 192.168.5.1: c2:57:e2:f3:3f:00 > da:53:12:09:ea:4e, ethertype IPv4 (0x0800), length 98: 192.168.1.1 > 192.168.5.1: da:53:12:09:ea:4e > c2:57:e2:f3:3f:00, ethertype IPv4 (0x0800), length 98: 192.168.5.1 > 192.168.1.1: da:53:12:09:ea:4e > c2:57:e2:f3:3f:00, ethertype ARP (0x0806), length 42: arp who-has 192.168.5.1: c2:57:e2:f3:3f:00 > da:53:12:09:ea:4e, ethertype ARP (0x0806), length 42: arp reply 192.168.5.1:
```

## 6. Продолжительность жизни пакета

Сначала написать как и на чём ломали.

```
r3:~# ip link set eth2 down
r3:~# ip route add 192.168.5.0/24 via 192.168.2.1 dev eth0
```

Потом какая-то таблица вышла.

```
192.168.5.0/24 via 192.168.2.1 dev eth0
192.168.4.0/24 dev eth1 proto kernel scope link src 192.168.4.2
192.168.3.0/24 via 192.168.4.1 dev eth1
192.168.2.0/24 dev eth0 proto kernel scope link src 192.168.2.2
192.168.1.0/24 via 192.168.2.1 dev eth0
```

Потом что слали.

```
ws1:~# ping 192.168.5.1 -c 1
PING 192.168.5.1 (192.168.5.1) 56(84) bytes of data.
From 192.168.2.2 icmp_seq=1 Time to live exceeded

--- 192.168.5.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

И что в итоге получилось.

```
r3:~# tcpdump -tnve -i eth0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
fa:de:dc:30:96:57 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.2.2: ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype ARP (0x0806), length 42: arp reply 192.168.2.2: fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 65535, len 98, interface 0, flags 0x0, window 0, checksum 0, options 0x00000000, ttl 63, len 98) ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 65535, len 98, interface 0, flags 0x0, window 0, checksum 0, options 0x00000000, ttl 62, len 98) fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 61, id 65535, len 98, interface 0, flags 0x0, window 0, checksum 0, options 0x00000000, ttl 61, len 98) ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 60, id 65535, len 98, interface 0, flags 0x0, window 0, checksum 0, options 0x00000000, ttl 60, len 98)
```



```

fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 9, id
ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 8, id
fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 7, id
ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 6, id
fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 5, id
ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 4, id
fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 3, id
ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 2, id
fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 1, id
ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 126: (tos 0xc0, ttl 64,
(tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.1.1 > 1
ee:97:f2:ab:47:0c > fa:de:dc:30:96:57, ethertype ARP (0x0806), length 42: arp who-has 192.168.2
fa:de:dc:30:96:57 > ee:97:f2:ab:47:0c, ethertype ARP (0x0806), length 42: arp reply 192.168.2.1

```

И кто в итоге отправил сообщение о завершении жизни.

```

ws1:~# traceroute -n 192.168.5.1
traceroute to 192.168.5.1 (192.168.5.1), 64 hops max, 40 byte packets
 1  192.168.1.2  6 ms  0 ms  0 ms
 2  192.168.2.2 11 ms  0 ms  0 ms
 3  192.168.1.2  0 ms  0 ms  0 ms
 4  192.168.2.2  0 ms  0 ms  0 ms
 5  * 192.168.1.2  1 ms  0 ms
 6  192.168.2.2  0 ms  0 ms  0 ms
 7  192.168.1.2  0 ms  *  2 ms
 8  192.168.2.2  2 ms  1 ms  1 ms
 9  192.168.1.2  1 ms  1 ms  *
10  192.168.2.2  1 ms  1 ms  0 ms
11  192.168.1.2  1 ms  0 ms  0 ms
12  * 192.168.2.2  2 ms  1 ms
13  192.168.1.2  1 ms  1 ms  1 ms
14  192.168.2.2  1 ms  *  1 ms
15  192.168.1.2  1 ms  1 ms  1 ms
16  192.168.2.2  1 ms  1 ms  *
17  192.168.1.2  1 ms  1 ms  1 ms
18  192.168.2.2  1 ms  1 ms  1 ms
19  * 192.168.1.2  2 ms  2 ms
20  192.168.2.2  2 ms  2 ms  2 ms
21  192.168.1.2  1 ms  *  1 ms
22  192.168.2.2  1 ms  1 ms  1 ms
23  192.168.1.2  1 ms  1 ms  *
24  192.168.2.2  1 ms  1 ms  1 ms
25  192.168.1.2  1 ms  1 ms  1 ms
26  * 192.168.2.2  1 ms  1 ms
27  192.168.1.2  1 ms  1 ms  1 ms
28  192.168.2.2  1 ms  *  2 ms
29  192.168.1.2  1 ms  1 ms  1 ms
30  192.168.2.2  1 ms  1 ms  *
31  192.168.1.2  5 ms  2 ms  1 ms
32  192.168.2.2  1 ms  1 ms  1 ms

```

```

33 * 192.168.1.2 2 ms 2 ms
34 192.168.2.2 2 ms 2 ms 2 ms
35 192.168.1.2 2 ms * 2 ms
36 192.168.2.2 1 ms 1 ms 1 ms
37 192.168.1.2 1 ms 1 ms *
38 192.168.2.2 2 ms 1 ms 1 ms
39 192.168.1.2 1 ms 2 ms 1 ms
40 * 192.168.2.2 2 ms 1 ms
41 192.168.1.2 1 ms 1 ms 1 ms
42 192.168.2.2 2 ms * 2 ms
43 192.168.1.2 1 ms 2 ms 1 ms
44 192.168.2.2 1 ms 1 ms *
45 192.168.1.2 2 ms 2 ms 2 ms
46 192.168.2.2 2 ms 2 ms 2 ms
47 * 192.168.1.2 2 ms 2 ms
48 192.168.2.2 2 ms 2 ms 2 ms
49 192.168.1.2 2 ms * 2 ms
50 192.168.2.2 2 ms 2 ms 2 ms
51 192.168.1.2 2 ms 2 ms *
52 192.168.2.2 2 ms 2 ms 2 ms
53 192.168.1.2 2 ms 2 ms 2 ms
54 * 192.168.2.2 4 ms 3 ms
55 192.168.1.2 3 ms 3 ms 3 ms
56 192.168.2.2 3 ms * 8 ms
57 192.168.1.2 4 ms 2 ms 2 ms
58 192.168.2.2 2 ms 2 ms *
59 192.168.1.2 8 ms 7 ms 4 ms
60 192.168.2.2 3 ms 3 ms 2 ms
61 * 192.168.1.2 7 ms 4 ms
62 192.168.2.2 4 ms 3 ms 3 ms
63 192.168.1.2 2 ms * 3 ms
64 192.168.2.2 2 ms 2 ms 2 ms

```

## 7. Изучение IP-фрагментации

Написать, на каких узлах и как изменяли MTU.

Изменение MTU

Изменение MTU

Какие команды давали для тестирования и где.

```

r1:~# ip link set dev eth1 mtu 600
r3:~# ip link set dev eth0 mtu 600

echo 1 > /proc/sys/net/ipv4/ip_no_pmtu_disc

ws1:~# ping 192.168.5.1 -c 1 -s 1000

```



Вывод **tcpdump** на маршрутизаторе перед сетью с уменьшенным MTU.

```
r1:~# tcpdump -tnv -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP (tos 0x0, ttl 64, id 29245, offset 0, flags [none], proto ICMP (1), length 1028) 192.168.1.1 > 192.168.5.1
IP (tos 0x0, ttl 62, id 39740, offset 0, flags [none], proto ICMP (1), length 1028) 192.168.5.1 > 192.168.1.1
```

Вывод **tcpdump** на маршрутизаторе после сети с уменьшенным MTU.

```
r3:~# tcpdump -tnv -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP (tos 0x0, ttl 63, id 29245, offset 0, flags [+], proto ICMP (1), length 596) 192.168.1.1 > 192.168.5.1
IP (tos 0x0, ttl 63, id 29245, offset 576, flags [none], proto ICMP (1), length 452) 192.168.1.1 > 192.168.5.1
IP (tos 0x0, ttl 63, id 39740, offset 0, flags [+], proto ICMP (1), length 596) 192.168.5.1 > 192.168.1.1
IP (tos 0x0, ttl 63, id 39740, offset 576, flags [none], proto ICMP (1), length 452) 192.168.5.1 > 192.168.1.1
```

Вывод **tcpdump** на узле получателя.

```
ws2:~# tcpdump -tnv -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP (tos 0x0, ttl 62, id 29245, offset 0, flags [none], proto ICMP (1), length 1028) 192.168.1.1 > 192.168.5.1
IP (tos 0x0, ttl 64, id 39740, offset 0, flags [none], proto ICMP (1), length 1028) 192.168.5.1 > 192.168.1.1
```

## 8. Отсутствие сети

Аналогично опишите опыт, когда маршрутизатор отправляет сообщение об отсутствии с сети. С командами и выводом, мак адреса не нужны.

```
ws1:~# ping 192.168.6.1 -c 1
PING 192.168.6.1 (192.168.6.1) 56(84) bytes of data.
From 192.168.1.2 icmp_seq=1 Destination Net Unreachable

--- 192.168.6.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

r1:~# tcpdump -n -i eth0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
22:07:14.304125 IP 192.168.1.1 > 192.168.6.1: ICMP echo request, id 11010, seq 1, length 64
22:07:14.304143 IP 192.168.1.2 > 192.168.1.1: ICMP net 192.168.6.1 unreachable, length 92
```

## 9. Отсутствие IP-адреса в сети

Аналогично опишите опыт, когда маршрутизатор отправляет сообщение об отсутствии требуемого IP-адреса в сети. С командами и выводом, мак адреса не нужны.

```
ws1:~# ping 192.168.5.4 -c 1
PING 192.168.5.4 (192.168.5.4) 56(84) bytes of data.
From 192.168.2.2 icmp_seq=1 Destination Host Unreachable
```

```
--- 192.168.5.4 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

r3:~# tcpdump -n -i eth2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 96 bytes
22:11:10.233182 arp who-has 192.168.5.4 tell 192.168.5.2
22:11:11.235214 arp who-has 192.168.5.4 tell 192.168.5.2
22:11:12.235217 arp who-has 192.168.5.4 tell 192.168.5.2
```