

# Отчёт по лабораторной работе «IP-маршрутизация»

Руднев Дмитрий Николаевич

13 ноября 2016 г.

## Содержание

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

## 1. Топология сети

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

## 2. Назначение IP-адресов

Ниже приведён файл настройки протокола IP маршрутизатора r1.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
```

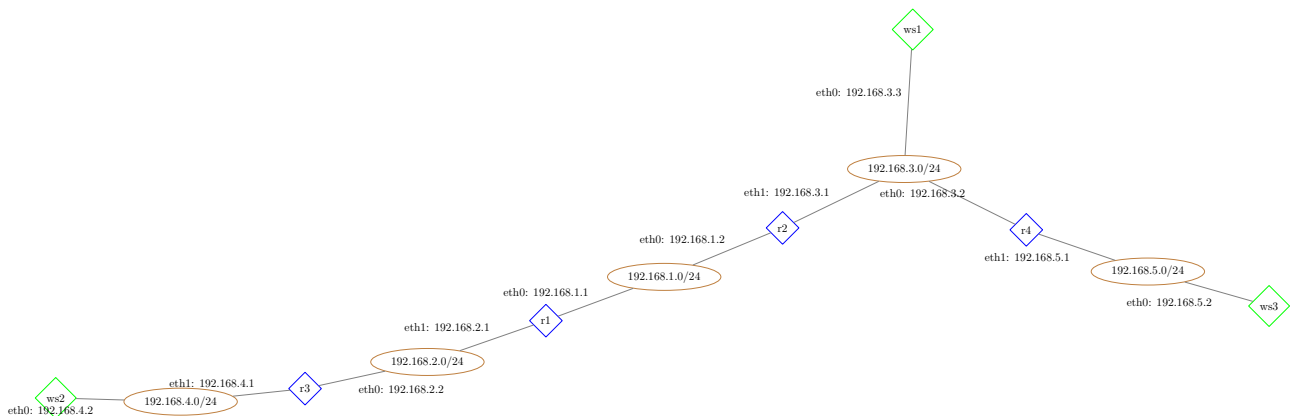


Рис. 1. Топология сети

```
up ip r add 192.168.3.0/24 via 192.168.1.2 dev eth0
up ip r add 192.168.5.0/24 via 192.168.1.2 dev eth0

down ip r del 192.168.3.0/24
down ip r del 192.168.5.0/24

auto eth1
iface eth1 inet static
address 192.168.2.1
netmask 255.255.255.0

up ip r add 192.168.4.0/24 via 192.168.2.2 dev eth1
down ip r del 192.168.4.0/24
```

Ниже приведён файл настройки протокола IP рабочей станции ws1.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.3.3
netmask 255.255.255.0
gateway 192.168.3.1
```

### 3. Таблица маршрутизации

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

```
192.168.5.0/24 via 192.168.1.2 dev eth0
192.168.4.0/24 via 192.168.2.2 dev eth1
192.168.3.0/24 via 192.168.1.2 dev eth0
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 192.168.1.1
```

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 192.168.1.1 dev eth0
192.168.3.0/24 dev eth1 proto kernel scope link src 192.168.3.1
192.168.2.0/24 via 192.168.1.1 dev eth0
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.2
```

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

```
192.168.5.0/24 via 192.168.2.1 dev eth0
192.168.4.0/24 dev eth1 proto kernel scope link src 192.168.4.1
192.168.3.0/24 via 192.168.2.1 dev eth0
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 dev eth1 proto kernel scope link src 192.168.5.1
192.168.4.0/24 via 192.168.3.1 dev eth0
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.2
192.168.2.0/24 via 192.168.3.1 dev eth0
192.168.1.0/24 via 192.168.3.1 dev eth0
```

## 4. Проверка настройки сети

Вывод **traceroute** от узла `ws1` до `ws3` при нормальной работе сети.

```
traceroute to 192.168.5.2 (192.168.5.2), 64 hops max, 40 byte packets
 1  192.168.3.1 (192.168.3.1)  2 ms  0 ms  8 ms
 2  192.168.3.2 (192.168.3.2)  19 ms  0 ms  0 ms
 3  192.168.5.2 (192.168.5.2)  6 ms  0 ms  0 ms
```

Вывод **traceroute** от узла `ws1` до `ws2` при нормальной работе сети.

```
traceroute to 192.168.4.2 (192.168.4.2), 64 hops max, 40 byte packets
 1  192.168.3.1 (192.168.3.1)  0 ms  0 ms  0 ms
 2  192.168.1.1 (192.168.1.1)  12 ms  1 ms  0 ms
 3  192.168.2.2 (192.168.2.2)  11 ms  1 ms  1 ms
 4  192.168.4.2 (192.168.4.2)  12 ms  1 ms  1 ms
```

Вывод **traceroute** от узла `ws2` до `ws3` при нормальной работе сети.

```
traceroute to 192.168.5.2 (192.168.5.2), 64 hops max, 40 byte packets
 1  192.168.4.1 (192.168.4.1)  1 ms  0 ms  0 ms
 2  192.168.2.1 (192.168.2.1)  0 ms  0 ms  0 ms
 3  192.168.1.2 (192.168.1.2)  0 ms  0 ms  0 ms
 4  192.168.3.2 (192.168.3.2)  11 ms  1 ms  1 ms
 5  192.168.5.2 (192.168.5.2)  1 ms  3 ms  4 ms
```

## 5. Маршрутизация

Вначале стоит написать, какие MAC-адреса интерфейсов в опыте были у каких машин. Затем вывести маршрутную таблицу маршрутизатора (вывод команды `ip r!`)

```
ws1 eth0 MAC: a6:f9:52:b6:1e:69
ws3 eth0 MAC: b2:0b:69:d6:a7:1e

r2 eth0 MAC: 3a:40:ee:31:9e:cd
r2 eth1 MAC: 12:3e:e2:7d:e3:87
r4 eth0 MAC: 4a:e4:d9:3b:f2:04
r4 eth1 MAC: 42:9b:97:db:b0:a6

маршрутная таблица маршрутизатора r2:
192.168.5.0/24 via 192.168.3.2 dev eth1
192.168.4.0/24 via 192.168.1.1 dev eth0
192.168.3.0/24 dev eth1 proto kernel scope link src 192.168.3.1
192.168.2.0/24 via 192.168.1.1 dev eth0
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.2

маршрутная таблица маршрутизатора r4:
192.168.5.0/24 dev eth1 proto kernel scope link src 192.168.5.1
192.168.4.0/24 via 192.168.3.1 dev eth0
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.2
192.168.2.0/24 via 192.168.3.1 dev eth0
192.168.1.0/24 via 192.168.3.1 dev eth0
```

Показаны опыты после стирания кеша ARP.

Далее показана отправка пакета на маршрутизатор (косвенная маршрутизация).

```
ws1:~# ping -c 1 192.168.5.2
r2 tcpdump:
a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.3.1 tell a6:f9:52:b6:1e:69
12:3e:e2:7d:e3:87 > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42: arp reply 192.168.3.1 from a6:f9:52:b6:1e:69
a6:f9:52:b6:1e:69 > 12:3e:e2:7d:e3:87, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 12345, len 84) 192.168.5.2 > 192.168.3.1
12:3e:e2:7d:e3:87 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.3.2 tell 12:3e:e2:7d:e3:87
4a:e4:d9:3b:f2:04 > 12:3e:e2:7d:e3:87, ethertype ARP (0x0806), length 42: arp reply 192.168.3.2 from 4a:e4:d9:3b:f2:04
12:3e:e2:7d:e3:87 > 4a:e4:d9:3b:f2:04, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 12345, len 84) 192.168.5.2 > 192.168.3.2
```

Затем маршрутизатор отправил его далее.

```
r2 tcpdump:
12:3e:e2:7d:e3:87 > 4a:e4:d9:3b:f2:04, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 12345, len 84) 192.168.5.2 > 192.168.3.2

r4 tcpdump:
a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.3.1 tell a6:f9:52:b6:1e:69
12:3e:e2:7d:e3:87 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.3.2 tell 12:3e:e2:7d:e3:87
4a:e4:d9:3b:f2:04 > 12:3e:e2:7d:e3:87, ethertype ARP (0x0806), length 42: arp reply 192.168.3.2 from 4a:e4:d9:3b:f2:04
12:3e:e2:7d:e3:87 > 4a:e4:d9:3b:f2:04, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 12345, len 84) 192.168.5.2 > 192.168.3.2
```

```
ws3 tcpdump:
42:9b:97:db:b0:a6 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.5.2
b2:0b:69:d6:a7:1e > 42:9b:97:db:b0:a6, ethertype ARP (0x0806), length 42: arp reply 192.168.5.2
42:9b:97:db:b0:a6 > b2:0b:69:d6:a7:1e, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.5.2 > 192.168.4.2
b2:0b:69:d6:a7:1e > 42:9b:97:db:b0:a6, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.4.2 > 192.168.5.2
```

## 6. Продолжительность жизни пакета

Сначала написать как и на чём ломали.

```
r1 router:
ip l set eth1 down
ip r add 192.168.4.0/24 via 192.168.1.2 dev eth0
```

Потом какая-то таблица вышла.

```
192.168.5.0/24 via 192.168.1.2 dev eth0
192.168.4.0/24 via 192.168.1.2 dev eth0
192.168.3.0/24 via 192.168.1.2 dev eth0
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.1
```

Потом что слали.

```
ws1:~# ping -c 1 192.168.4.2
```

И что в итоге получилось.

```
r1 tcpdump:
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.5.2 > 192.168.4.2
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.4.2 > 192.168.5.2
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 61, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.5.2 > 192.168.4.2
...
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 126: (tos 0xc0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.4.2 > 192.168.5.2
(tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.3.3 > 192.168.4.2
```

И r1 в итоге отправил сообщение о завершении жизни.

## 7. Изучение IP-фрагментации

Написать, на каких узлах и как изменяли MTU.

```
r1:
ip l set dev eth0 mtu 576

r2:
ip l set dev eth0 mtu 576
```

Какие команды давали для тестирования и где.

```
ws1:
ping -c 1 -s 1000 192.168.4.2
```

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

```
r2 eth0:
IP (tos 0x0, ttl 63, id 55859, offset 0, flags [+], proto ICMP (1), length 572) 192.168.3.3 > 1
IP (tos 0x0, ttl 63, id 55859, offset 552, flags [none], proto ICMP (1), length 476) 192.168.3.3
```

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

```
r1 eth0:
IP (tos 0x0, ttl 63, id 55859, offset 0, flags [+], proto ICMP (1), length 572) 192.168.3.3 > 1
IP (tos 0x0, ttl 63, id 55859, offset 552, flags [none], proto ICMP (1), length 476) 192.168.3.3
r1 eth1:
IP (tos 0x0, ttl 62, id 55861, offset 0, flags [none], proto ICMP (1), length 1028) 192.168.3.3
```

Вывод **tcpdump** на узле получателя.

```
ws2:
IP (tos 0x0, ttl 61, id 55860, offset 0, flags [none], proto ICMP (1), length 1028) 192.168.3.3
```

## 8. Отсутствие сети

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

```
ws1:
ping -c 1 -s 1000 10.10.4.2
r2:
IP (...) 192.168.3.1 > 192.168.3.3: ICMP net 10.10.4.2 unreachable, length 556
IP (...) 192.168.3.3 > 10.10.4.2: ICMP echo request, id 28418, seq 1, length 1008[|icmp
```

## 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 -c 1 -s 1000 192.168.4.100
r3:
IP (...) 192.168.2.2 > 192.168.3.3: ICMP host 192.168.4.100 unreachable, length 556
IP (...) 192.168.3.3 > 192.168.4.100: ICMP echo request, id 28674, seq 1, length 1008[|icmp
```