Отчёт по лабораторной работе «IP-маршрутизация»

Жарова Н. А.

4 января 2021 г.

Содержание

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 10.0.10.1
netmask 255.255.255.0
```

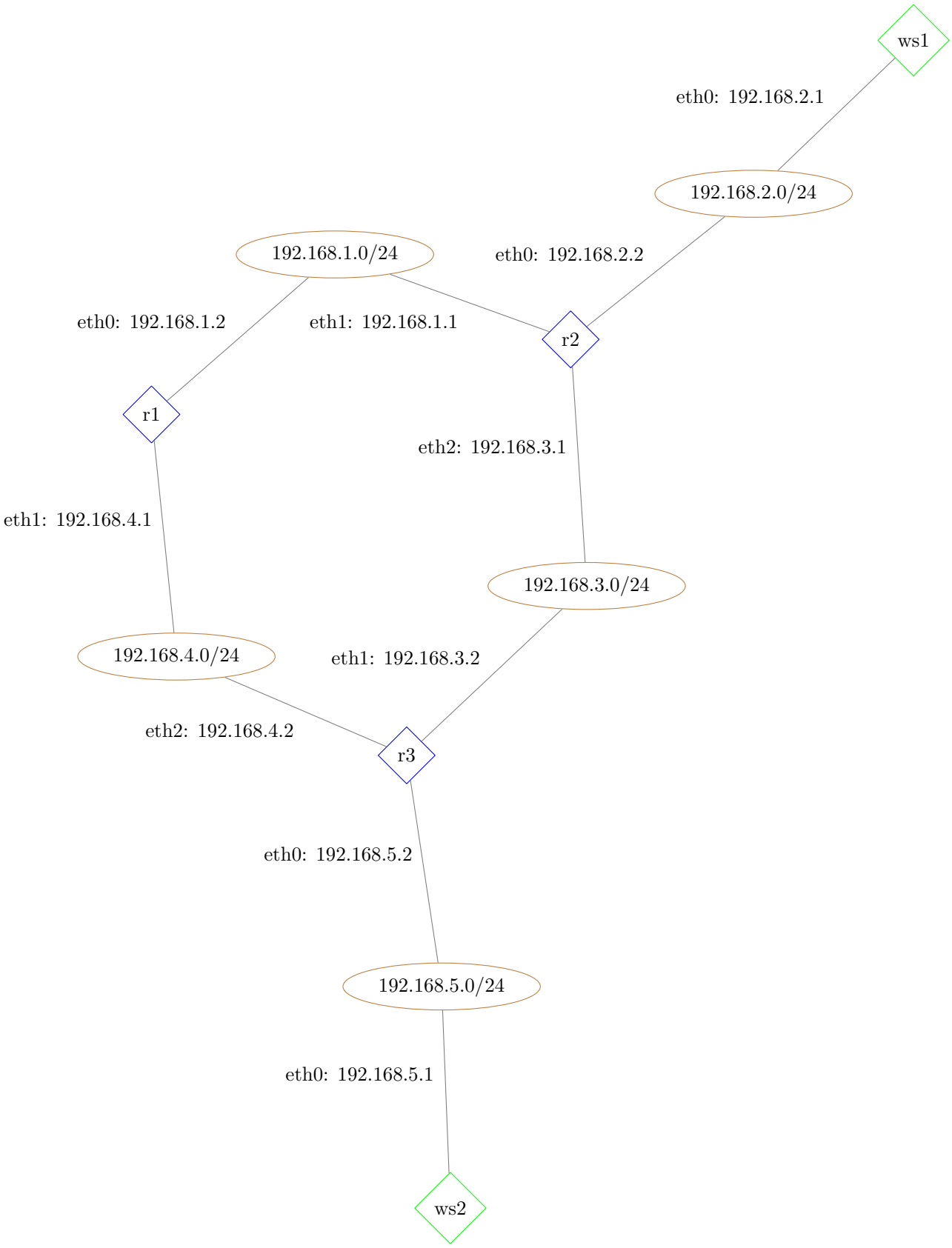

Рис. 1. Топология сети

```
up ip r add 10.0.20.0/24 via 10.0.10.2 dev eth0; ip r add 10.0.30.0/24 via 10.0.10.2 dev eth0
down ip r del 10.0.20.0/24; ip r del 10.0.30.0/24

auto eth1
iface eth1 inet static
address 10.0.40.1
netmask 255.255.255.0
up ip r add 10.0.50.0/24 via 10.0.40.2 dev eth1
down ip r del 10.0.50.0/24
```

Ниже приведён файл настройки протокола IP маршрутизатора **r2**.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.0.10.2
netmask 255.255.255.0
up ip r add 10.0.40.0/24 via 10.0.10.1 dev eth0
down ip r del 10.0.40.0/24

auto eth1
iface eth1 inet static
address 10.0.30.1
netmask 255.255.255.0
up ip r add 10.0.50.0/24 via 10.0.30.2 dev eth1
down ip r del 10.0.50.0/24

auto eth2
iface eth2 inet static
address 10.0.20.1
netmask 255.255.255.0
```

Ниже приведён файл настройки протокола IP маршрутизатора **r3**.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.0.30.2
netmask 255.255.255.0
up ip r add 10.0.20.0/24 via 10.0.30.1 dev eth0
down ip r del 10.0.20.0/24

auto eth1
iface eth1 inet static
address 10.0.40.2
netmask 255.255.255.0
```

```
up ip r add 10.0.10.0/24 via 10.0.40.1 dev eth1
down ip r del 10.0.10.0/24

auto eth2
iface eth2 inet static
address 10.0.50.1
netmask 255.255.255.0
```

Ниже приведён файл настройки протокола IP рабочей станции **ws1**.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.0.20.2
netmask 255.255.255.0
gateway 10.0.20.1
```

Ниже приведён файл настройки протокола IP рабочей станции **ws2**.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.0.50.2
netmask 255.255.255.0
gateway 10.0.50.1
```

3. Таблица маршрутизации

Вывод (*ip r*) таблицы маршрутизации для **r1**.

```
10.0.20.0/24 via 10.0.10.2 dev eth0
10.0.50.0/24 via 10.0.40.2 dev eth1
10.0.30.0/24 via 10.0.10.2 dev eth0
10.0.40.0/24 dev eth1  proto kernel  scope link  src 10.0.40.1
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.1
```

Вывод (*ip r*) таблицы маршрутизации для **r2**.

```
10.0.20.0/24 dev eth2  proto kernel  scope link  src 10.0.20.1
10.0.50.0/24 via 10.0.30.2 dev eth1
10.0.30.0/24 dev eth1  proto kernel  scope link  src 10.0.30.1
10.0.40.0/24 via 10.0.10.1 dev eth0
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.2
```

Вывод (*ip r*) таблицы маршрутизации для **r3**.

```
10.0.20.0/24 via 10.0.30.1 dev eth0
10.0.50.0/24 dev eth2  proto kernel  scope link  src 10.0.50.1
10.0.30.0/24 dev eth0  proto kernel  scope link  src 10.0.30.2
10.0.40.0/24 dev eth1  proto kernel  scope link  src 10.0.40.2
10.0.10.0/24 via 10.0.40.1 dev eth1
```

Вывод (*ip r*) таблицы маршрутизации для **ws1**.

```
10.0.20.0/24 dev eth0  proto kernel  scope link  src 10.0.20.2
default via 10.0.20.1 dev eth0
```

Вывод (*ip r*) таблицы маршрутизации для **ws2**.

```
10.0.50.0/24 dev eth0  proto kernel  scope link  src 10.0.50.2
default via 10.0.50.1 dev eth0
```

4. Проверка настройки сети

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

```
traceroute to 10.0.20.2 (10.0.20.2), 64 hops max, 40 byte packets
1  10.0.50.1 (10.0.50.1)  2 ms  1 ms  1 ms
2  10.0.30.1 (10.0.30.1)  5 ms  1 ms  1 ms
3  10.0.20.2 (10.0.20.2)  12 ms  1 ms  1 ms
```

Вывод **traceroute** от узла ws1 до r1 (eth1) при нормальной работе сети.

```
traceroute to 10.0.40.1 (10.0.40.1), 64 hops max, 40 byte packets
1  10.0.20.1 (10.0.20.1)  1 ms  1 ms  0 ms
2  10.0.40.1 (10.0.40.1)  7 ms  1 ms  1 ms
```

Вывод **traceroute** от узла ws2 до r1 (eth0) при нормальной работе сети.

```
traceroute to 10.0.10.1 (10.0.10.1), 64 hops max, 40 byte packets
1  10.0.50.1 (10.0.50.1)  1 ms  1 ms  0 ms
2  10.0.10.1 (10.0.10.1)  11 ms  1 ms  1 ms
```

5. Маршрутизация

Пример косвенной маршрутизации при передаче пакета
от ws1 (eth0 - a6:f9:52:b6:1e:69 - 10.0.20.2/24) до r1 (eth0 - 0e:ab:f8:0c:10:4b - 10.0.10.1/24) через r2 (eth2 - 4a:6c:31:ed:d1:db - 10.0.20.1/24 и eth0 - 3a:40:ee:31:9e:cd - 10.0.10.2/24).

Маршрутная таблица маршрутизатора r2 (вывод команды ip r):

```
10.0.20.0/24 dev eth2  proto kernel  scope link  src 10.0.20.1
10.0.50.0/24 via 10.0.30.2 dev eth1
10.0.30.0/24 dev eth1  proto kernel  scope link  src 10.0.30.1
10.0.40.0/24 via 10.0.10.1 dev eth0
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.2
```

Показаны опыты после стирания кеша ARP.

На ws1 будет вызвана команда:

```
ping 10.0.10.1
```

Далее показана отправка пакета на маршрутизатор r2 (косвенная маршрутизация).

```
tcpdump -tne -i eth2

a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806),
     length 42: arp who-has 10.0.20.1 tell 10.0.20.2
4a:6c:31:ed:d1:db > a6:f9:52:b6:1e:69, ethertype ARP (0x0806),
    length 42: arp reply 10.0.20.1 is-at 4a:6c:31:ed:d1:db
a6:f9:52:b6:1e:69 > 4a:6c:31:ed:d1:db, ethertype IPv4 (0x0800),
    length 98: 10.0.20.2 > 10.0.10.1: ICMP echo request, id 23554, seq 1, length 64
4a:6c:31:ed:d1:db > a6:f9:52:b6:1e:69, ethertype IPv4 (0x0800),
    length 98: 10.0.10.1 > 10.0.20.2: ICMP echo reply, id 23554, seq 1, length 64
```

Затем маршрутизатор отправил его далее на маршрутизатор r1.

```
tcpdump -tne -i eth0

3a:40:ee:31:9e:cd > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806),
    length 42: arp who-has 10.0.10.1 tell 10.0.10.2
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype ARP (0x0806),
    length 42: arp reply 10.0.10.1 is-at 0e:ab:f8:0c:10:4b
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800),
    length 98: 10.0.20.2 > 10.0.10.1: ICMP echo request, id 23554, seq 1, length 64
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800),
    length 98: 10.0.10.1 > 10.0.20.2: ICMP echo reply, id 23554, seq 1, length 64
```

6. Продолжительность жизни пакета

Для создания маршрутной петли (сеть 10.0.40.0/24 будет завёрнута между r2 и r1) на маршрутизаторе r1 были запущены следующие команды:

```
ip l set eth1 down
ip r add 10.0.40.0/24 via 10.0.10.2 dev eth0
```

Теперь на r1 таблица маршрутизации выглядит следующим образом:

```
10.0.20.0/24 via 10.0.10.2 dev eth0
10.0.30.0/24 via 10.0.10.2 dev eth0
10.0.40.0/24 via 10.0.10.2 dev eth0
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.1
```

С ws1 отправим ping в завёрнутую сеть:

```
ping 10.0.40.2 -c 1
PING 10.0.40.2 (10.0.40.2) 56(84) bytes of data.
From 10.0.10.1 icmp_seq=1 Time to live exceeded
```

На r2 будем перехватывать трафик на интерфейсе, подключенном к завёрнутой сети:

```
tcpdump -tnve -i eth0
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)
    10.0.20.2 > 10.0.40.2: ICMP echo request, id 22018, seq 1, length 64
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)
    10.0.20.2 > 10.0.40.2: ICMP echo request, id 22018, seq 1, length 64
...
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98:
    (tos 0x0, ttl 2, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.20.2 > 10.0.40.2: ICMP echo request, id 22018, seq 1, length 64
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98:
    (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.20.2 > 10.0.40.2: ICMP echo request, id 22018, seq 1, length 64
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 126:
    (tos 0xc0, ttl 64, id 42765, offset 0, flags [none], proto ICMP (1), length 112)
    10.0.10.1 > 10.0.20.2: ICMP time exceeded in-transit, length 92
        (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
        10.0.20.2 > 10.0.40.2: ICMP echo request, id 22018, seq 1, length 64
```

В итоге, когда r1 должен был в очередной раз отправить пакет на r2, TTL достигло значения 0, и r1 отправил ICMP-сообщение с информацией о том, что время жизни пакета истекло.

7. Изучение IP-фрагментации

Уменьшим MTU сети 10.0.30.0/24. Для этого на r2 введём команду:

```
ip l set dev eth1 mtu 576
```

А на r3:

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

На ws1 отключим PMTU и запустим ping с размером пакета 1000 на ws2:

```
echo 1 > /proc/sys/net/ipv4/ip_no_pmtu_disc
ping 10.0.50.2 -c 1 -s 1000
```

Вывод **tcpdump** на маршрутизаторе r2 перед сетью с уменьшенным MTU.

```
tcpdump -tnv -i eth2
IP (tos 0x0, ttl 64, id 61706, offset 0, flags [none], proto ICMP (1), length 1028)
    10.0.20.2 > 10.0.50.2: ICMP echo request, id 22786, seq 1, length 1008
IP (tos 0x0, ttl 62, id 2471, offset 0, flags [none], proto ICMP (1), length 1028)
    10.0.50.2 > 10.0.20.2: ICMP echo reply, id 22786, seq 1, length 1008
```

Вывод **tcpdump** на маршрутизаторе r3 после сети с уменьшенным MTU.

```
tcpdump -tnv -i eth0
IP (tos 0x0, ttl 63, id 61706, offset 0, flags [+], proto ICMP (1), length 572)
    10.0.20.2 > 10.0.50.2: ICMP echo request, id 22786, seq 1, length 552
IP (tos 0x0, ttl 63, id 61706, offset 552, flags [none], proto ICMP (1), length 476)
    10.0.20.2 > 10.0.50.2: icmp
IP (tos 0x0, ttl 63, id 2471, offset 0, flags [+], proto ICMP (1), length 572)
    10.0.50.2 > 10.0.20.2: ICMP echo reply, id 22786, seq 1, length 552
IP (tos 0x0, ttl 63, id 2471, offset 552, flags [none], proto ICMP (1), length 476)
    10.0.50.2 > 10.0.20.2: icmp
```

Вывод **tcpdump** на ws2.

```
tcpdump -tnv -i eth0
IP (tos 0x0, ttl 62, id 61706, offset 0, flags [none], proto ICMP (1), length 1028)
    10.0.20.2 > 10.0.50.2: ICMP echo request, id 22786, seq 1, length 1008
IP (tos 0x0, ttl 64, id 2471, offset 0, flags [none], proto ICMP (1), length 1028)
    10.0.50.2 > 10.0.20.2: ICMP echo reply, id 22786, seq 1, length 1008
```

8. Отсутствие сети

С ws1 была запущена команда:

```
ping 20.0.20.1 -c 1
PING 20.0.20.1 (20.0.20.1) 56(84) bytes of data.
From 10.0.20.1 icmp_seq=1 Destination Net Unreachable
```

На маршрутизаторе r2 запущен перехват трафика:

```
tcpdump -n -i eth2
11:20:56.315890 IP 10.0.20.2 > 20.0.20.1: ICMP echo request, id 19714, seq 1, length 64
11:20:56.315927 IP 10.0.20.1 > 10.0.20.2: ICMP net 20.0.20.1 unreachable, length 92
```

Как видно, производится только один ARP-запрос, на который тут же генерируется ICMP-сообщение с информацией о том, что искомая сеть не найдена.

9. Отсутствие IP-адреса в сети

С ws1 была запущена команда:

```
ping 10.0.30.3 -c 1
PING 10.0.30.3 (10.0.30.3) 56(84) bytes of data.
From 10.0.20.1 icmp_seq=1 Destination Host Unreachable
```

На маршрутизаторе r2 запущен перехват трафика на интерфейсе, подключённом к той же сети, что и ws1:

```
tcpdump -n -i eth2
11:27:53.487495 IP 10.0.20.2 > 10.0.30.3: ICMP echo request, id 20226, seq 1, length 64
11:27:56.493708 IP 10.0.20.1 > 10.0.20.2: ICMP host 10.0.30.3 unreachable, length 92
```

На маршрутизаторе r2 запущен перехват трафика на интерфейсе, подключённом к сети, в котором должен был находится целевой ip-адрес:

```
tcpdump -n -i eth1
11:31:13.746943 arp who-has 10.0.30.3 tell 10.0.30.1
11:31:14.743571 arp who-has 10.0.30.3 tell 10.0.30.1
11:31:15.743563 arp who-has 10.0.30.3 tell 10.0.30.1
```

Как видно, производится 3 ARP-запроса с таймаутом в 1 секунду. После того, как на последний запрос в течении секунды не пришёл ответ, генерируется ICMP-сообщение с информацией о том, что целевой IP-адрес не найден.