Отчёт по лабораторной работе «IP-маршрутизация»

Дун Юйхань

6 ноября 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 10.0.10.2
        netmask 255.255.255.0
```

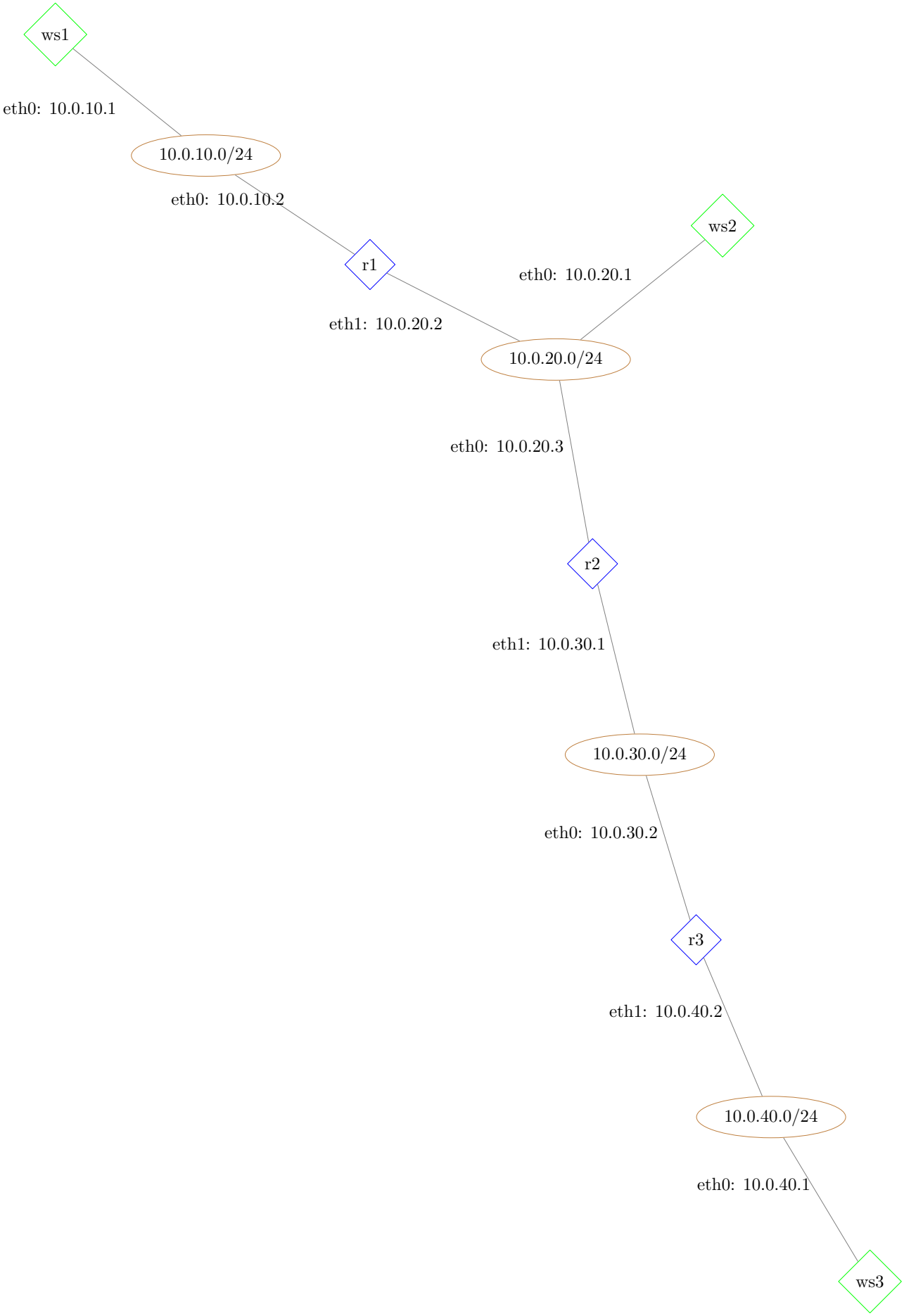

Рис. 1. Топология сети

```
auto eth1
iface eth1 inet static
        address 10.0.20.2
        netmask 255.255.255.0

        up ip r add 10.0.30.0/24 via 10.0.20.3 dev eth1
        down ip r del 10.0.30.0/24

        up ip r add 10.0.40.0/24 via 10.0.20.3 dev eth1
        down ip r del 10.0.40.0/24
```

Ниже приведён файл настройки протокола IP рабочей станции **r2**.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 10.0.20.3
        netmask 255.255.255.0

        up ip r add 10.0.10.0/24 via 10.0.20.2 dev eth0
        down ip r del 10.0.10.0/24

auto eth1
iface eth1 inet static
        address 10.0.30.1
        netmask 255.255.255.0

        up ip r add 10.0.40.0/24 via 10.0.30.2 dev eth1
        down ip r del 10.0.40.0/24
```

Ниже приведён файл настройки протокола IP рабочей станции **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 10.0.10.0/24 via 10.0.30.1 dev eth0
        down ip r del 10.0.10.0/24

        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
```

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

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 10.0.10.1
        netmask 255.255.255.0
        gateway 10.0.10.2
```

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

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 10.0.20.1
        netmask 255.255.255.0
        gateway 10.0.20.3
```

Ниже приведён файл настройки протокола IP рабочей станции **ws3**.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 10.0.40.1
        netmask 255.255.255.0
        gateway 10.0.40.2
```

3. Таблица маршрутизации

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

```
10.0.20.0/24 dev eth1  proto kernel  scope link  src 10.0.20.2
10.0.30.0/24 via 10.0.20.3 dev eth1
10.0.40.0/24 via 10.0.20.3 dev eth1
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.2
```

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

```
10.0.20.0/24 dev eth0  proto kernel  scope link  src 10.0.20.3
10.0.30.0/24 dev eth1  proto kernel  scope link  src 10.0.30.1
10.0.40.0/24 via 10.0.30.2 dev eth1
10.0.10.0/24 via 10.0.20.2 dev eth0
```

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

```
10.0.20.0/24 via 10.0.30.1 dev eth0
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 10.0.30.1 dev eth0
```

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

```
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.1
default via 10.0.10.2 dev eth0
```

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

```
10.0.20.0/24 dev eth0  proto kernel  scope link  src 10.0.20.1
default via 10.0.20.3 dev eth0
```

Вывести (командой ip r) таблицу маршрутизации для **ws3**.

```
10.0.40.0/24 dev eth0  proto kernel  scope link  src 10.0.40.1
default via 10.0.40.2 dev eth0
```

4. Проверка настройки сети

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

```
traceroute to 10.0.20.1 (10.0.20.1), 64 hops max, 40 byte packets
 1  10.0.40.2 (10.0.40.2)  5 ms  0 ms  0 ms
 2  10.0.30.1 (10.0.30.1)  13 ms  0 ms  1 ms
 3  10.0.20.1 (10.0.20.1)  15 ms  2 ms  1 ms
```

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

```
traceroute to 10.0.10.2 (10.0.10.2), 64 hops max, 40 byte packets
 1  10.0.20.3 (10.0.20.3)  0 ms  0 ms  0 ms
 2  10.0.10.2 (10.0.10.2)  23 ms  1 ms  1 ms
```

Вывод **traceroute** от ws1 до r2(eth1) при нормальной работе сети.

```
traceroute to 10.0.30.1 (10.0.30.1), 64 hops max, 40 byte packets
 1  10.0.10.2 (10.0.10.2)  4 ms  1 ms  0 ms
 2  10.0.30.1 (10.0.30.1)  12 ms  1 ms  1 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!)

от ws1(eth0 - 10.0.10.1/24) до r2(eth0 - 10.0.30.1/24) через r1(eth0 - 10.0.10.2/24 и eth1 - 10.0.20.2/24)

маршрутная таблица маршрутизатора r1

```
10.0.20.0/24 dev eth1  proto kernel  scope link  src 10.0.20.2
10.0.30.0/24 via 10.0.20.3 dev eth1
10.0.40.0/24 via 10.0.20.3 dev eth1
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 10.0.30.1
```

Далее показана отправка пакета на маршрутизатор r1(косвенная маршрутизация).

```
tcpdump -tne -i eth0

a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 10.0.10.2
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42: arp reply 10.0.10.2 i
a6:f9:52:b6:1e:69 > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: 10.0.10.1 > 10.0.30.
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype IPv4 (0x0800), length 98: 10.0.30.1 > 10.0.10.
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42: arp who-has 10.0.10.1
a6:f9:52:b6:1e:69 > 0e:ab:f8:0c:10:4b, ethertype ARP (0x0806), length 42: arp reply 10.0.10.1 i
```

Затем маршрутизатор отправил его далее на маршрутизатор r2.

```
tcpdump -tne -i eth0

fa:de:dc:30:96:57 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 10.0.20.3
3a:40:ee:31:9e:cd > fa:de:dc:30:96:57, ethertype ARP (0x0806), length 42: arp reply 10.0.20.3 i
fa:de:dc:30:96:57 > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: 10.0.10.1 > 10.0.30.
3a:40:ee:31:9e:cd > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: 10.0.30.1 > 10.0.10.
3a:40:ee:31:9e:cd > fa:de:dc:30:96:57, ethertype ARP (0x0806), length 42: arp who-has 10.0.20.2
fa:de:dc:30:96:57 > 3a:40:ee:31:9e:cd, ethertype ARP (0x0806), length 42: arp reply 10.0.20.2 i
```

6. Продолжительность жизни пакета

Для создания маршрутной петли (сеть 10.0.30.0/24 будет завёрнута между r2 и r1) на маршрутизаторе r2 были запущены следующие команды:

```
ip l set eth1 down
ip r add 10.0.30.0/24 via 10.0.20.2 dev eth0
```

Теперь на r2 таблица маршрутизации выглядит следующим образом:

```
10.0.20.0/24 dev eth0  proto kernel  scope link  src 10.0.20.3
10.0.30.0/24 via 10.0.20.2 dev eth0
10.0.10.0/24 via 10.0.20.2 dev eth0
```

С ws1 отправим ping в завёрнутую сеть:

```
ping 10.0.30.2 -c 1
PING 10.0.30.2 (10.0.30.2) 56(84) bytes of data.
From 10.0.20.3 icmp_seq=1 Time to live exceeded
```

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

```
fa:de:dc:30:96:57 > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id
3a:40:ee:31:9e:cd > fa:de:dc:30:96:57, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id
...
3a:40:ee:31:9e:cd > fa:de:dc:30:96:57, ethertype ARP (0x0806), length 42: arp who-has 10.0.20.2
fa:de:dc:30:96:57 > 3a:40:ee:31:9e:cd, ethertype ARP (0x0806), length 42: arp reply 10.0.20.2 i
```

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 10.0.30.1/24. Для этого на r2 введём команду:

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

А на r3 eth0:

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

На ws2 отключим PMTU и запустим ping с размером пакета 1000 на ws3:

```
echo 1 >/proc/sys/net/ipv4/ip_no_pmtu_disc
ping 10.0.40.1 -c 1 -s 1000
```

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

```
IP (tos 0x0, ttl 64, id 57496, offset 0, flags [none], proto ICMP (1), length 1028) 10.0.20.1 >
IP (tos 0x0, ttl 62, id 54250, offset 0, flags [none], proto ICMP (1), length 1028) 10.0.40.1 >
```

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

```
IP (tos 0x0, ttl 62, id 57496, offset 0, flags [none], proto ICMP (1), length 1028) 10.0.20.1 >
IP (tos 0x0, ttl 64, id 54250, offset 0, flags [none], proto ICMP (1), length 1028) 10.0.40.1 >
```

Вывод **tcpdump** на узле получателя ws3.

```
IP (tos 0x0, ttl 62, id 57496, offset 0, flags [none], proto ICMP (1), length 1028) 10.0.20.1 >
IP (tos 0x0, ttl 64, id 54250, offset 0, flags [none], proto ICMP (1), length 1028) 10.0.40.1 >
```

8. Отсутствие сети

С ws1 была запущена команда:

```
ping 10.0.100.1 -c 1
PING 10.0.100.1 (10.0.100.1) 56(84) bytes of data.
From 10.0.10.2 icmp_seq=1 Destination Net Unreachable
```

На маршрутизаторе r1 запущен перехват трафика:

```
tcpdump -n -i eth0

14:07:58.214405 IP 10.0.10.1 > 10.0.100.1: ICMP echo request, id 19202, seq 1, length 64
14:07:58.214423 IP 10.0.10.2 > 10.0.10.1: ICMP net 10.0.100.1 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 10.0.30.11 -c 1
PING 10.0.30.11 (10.0.30.11) 56(84) bytes of data.
From 10.0.20.3 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 eth0

09:20:03.950262 IP 10.0.10.1 > 10.0.30.11: ICMP echo request, id 17922, seq 1, length 64
09:20:06.954618 IP 10.0.20.3 > 10.0.10.1: ICMP host 10.0.30.11 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

09:22:52.750690 arp who-has 10.0.30.11 tell 10.0.30.1
09:22:53.744466 arp who-has 10.0.30.11 tell 10.0.30.1
09:22:54.744442 arp who-has 10.0.30.11 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.