Отчёт по лабораторной работе «IP-маршрутизация»

Гребенюк Александр Андреевич

8 декабря 2015 г.

Содержание

1. Топология сети

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

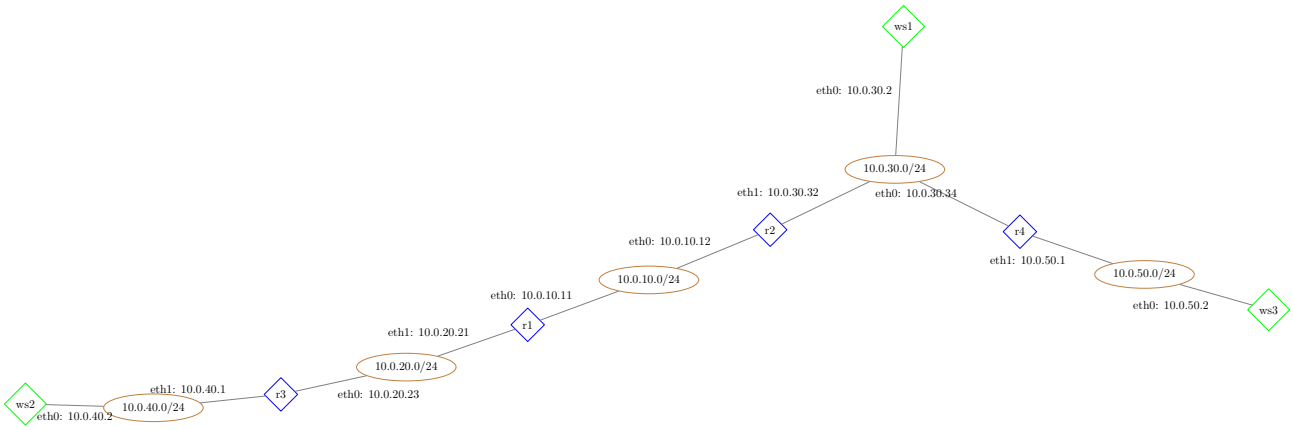

Рис. 1: Топология сети

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 10.0.10.11
    netmask 255.255.255.0
    up ip r add 10.0.30.0/24 via 10.0.10.12 dev eth0
    up ip r add 10.0.50.0/24 via 10.0.10.12 dev eth0
    down ip r del 10.0.30.0/24
    down ip r del 10.0.50.0/24

auto eth1
iface eth1 inet static
    address 10.0.20.21
    netmask 255.255.255.0
    up ip r add 10.0.40.0/24 via 10.0.20.23 dev eth1
    down ip r del 10.0.40.0/24
```

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

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.0.10.12
```

```
    netmask 255.255.255.0
    up ip r add 10.0.20.0/24 via 10.0.10.11 dev eth0
    up ip r add 10.0.40.0/24 via 10.0.10.11 dev eth0
    down ip r del 10.0.20.0/24
    down ip r del 10.0.40.0/24

auto eth1
iface eth1 inet static
    address 10.0.30.32
    netmask 255.255.255.0
    up ip r add 10.0.50.0/24 via 10.0.30.34 dev eth1
    down ip r del 10.0.50.0/24
```

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

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.0.20.23
    netmask 255.255.255.0
    up ip r add 10.0.10.0/24 via 10.0.20.21 dev eth0
    up ip r add 10.0.30.0/24 via 10.0.20.21 dev eth0
    up ip r add 10.0.50.0/24 via 10.0.20.21 dev eth0
    down ip r del 10.0.10.0/24
    down ip r del 10.0.30.0/24
    down ip r del 10.0.50.0/24

auto eth1
iface eth1 inet static
    address 10.0.40.1
    netmask 255.255.255.0
```

– Ниже приведён файл настройки протокола IP маршрутизатора **r4**.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.0.30.34
    netmask 255.255.255.0
    up ip r add 10.0.10.0/24 via 10.0.30.32 dev eth0
    up ip r add 10.0.20.0/24 via 10.0.30.32 dev eth0
    up ip r add 10.0.40.0/24 via 10.0.30.32 dev eth0
    down ip r del 10.0.10.0/24
    down ip r del 10.0.20.0/24
    down ip r del 10.0.40.0/24

auto eth1
```

```
iface eth1 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 10.0.30.2
    netmask 255.255.255.0
    gateway 10.0.30.32
```

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

```
auto lo
iface lo inet loopback

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

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

```
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

– Таблица маршрутизации **r1**.

```
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.11
10.0.20.0/24 dev eth1  proto kernel  scope link  src 10.0.20.21
10.0.30.0/24 via 10.0.10.12 dev eth0
10.0.40.0/24 via 10.0.20.23 dev eth1
10.0.50.0/24 via 10.0.10.12 dev eth0
```

– Таблица маршрутизации **r2**.

```
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.12
10.0.20.0/24 via 10.0.10.11 dev eth0
10.0.30.0/24 dev eth1  proto kernel  scope link  src 10.0.30.32
10.0.40.0/24 via 10.0.10.11 dev eth0
10.0.50.0/24 via 10.0.30.34 dev eth1
```

– Таблица маршрутизации **r3**.

```
10.0.10.0/24 via 10.0.20.21 dev eth0
10.0.20.0/24 dev eth0  proto kernel  scope link  src 10.0.20.23
10.0.30.0/24 via 10.0.20.21 dev eth0
10.0.40.0/24 dev eth1  proto kernel  scope link  src 10.0.40.1
10.0.50.0/24 via 10.0.20.21 dev eth0
```

– Таблица маршрутизации **r4**.

```
10.0.10.0/24 via 10.0.30.32 dev eth0
10.0.20.0/24 via 10.0.30.32 dev eth0
10.0.30.0/24 dev eth0  proto kernel  scope link  src 10.0.30.34
10.0.40.0/24 via 10.0.30.32 dev eth0
10.0.50.0/24 dev eth1  proto kernel  scope link  src 10.0.50.1
```

4. Проверка настройки сети

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

```
root@ws1:~# traceroute 10.0.40.2
traceroute to 10.0.40.2 (10.0.40.2), 30 hops max, 60 byte packets
 1  10.0.30.32 (10.0.30.32)  0.115 ms  0.083 ms  0.078 ms
 2  10.0.10.11 (10.0.10.11)  0.187 ms  0.101 ms  0.074 ms
 3  10.0.20.23 (10.0.20.23)  0.220 ms  0.132 ms  0.136 ms
 4  10.0.40.2 (10.0.40.2)  0.275 ms  0.203 ms  0.182 ms
```

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

```
root@ws1:~# traceroute 10.0.50.2
traceroute to 10.0.50.2 (10.0.50.2), 30 hops max, 60 byte packets
 1  10.0.30.32 (10.0.30.32)  0.371 ms  0.176 ms  0.134 ms
 2  10.0.30.34 (10.0.30.34)  0.219 ms  0.063 ms  0.060 ms
 3  10.0.50.2 (10.0.50.2)  0.246 ms  0.134 ms  0.107 ms
```

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

```
root@ws2:~# traceroute 10.0.30.2
traceroute to 10.0.30.2 (10.0.30.2), 30 hops max, 60 byte packets
 1  10.0.40.1 (10.0.40.1)  0.148 ms  0.143 ms  0.103 ms
 2  10.0.20.21 (10.0.20.21)  0.226 ms  0.119 ms  0.203 ms
 3  10.0.10.12 (10.0.10.12)  0.261 ms  0.182 ms  0.109 ms
 4  10.0.30.2 (10.0.30.2)  0.146 ms  0.130 ms  0.119 ms
```

5. Маршрутизация

Таблица 1: MAC-адреса

Host	Interface	MAC
ws1	eth0	7e:a6:65:cd:a1:ab
ws2	eth0	42:70:9a:0b:39:80
r1	eth0 eth1	be:14:9b:41:0f:21 3a:bf:72:36:29:d9
r2	eth0 eth1	ca:a0:5b:a0:0e:20 96:e8:1f:21:5e:44
r3	eth0 eth1	ee:93:78:86:e8:49 76:e5:80:0f:a1:02

– Таблица маршрутизации **r1**.

```
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.11
10.0.20.0/24 dev eth1  proto kernel  scope link  src 10.0.20.21
10.0.30.0/24 via 10.0.10.12 dev eth0
10.0.40.0/24 via 10.0.20.23 dev eth1
10.0.50.0/24 via 10.0.10.12 dev eth0
```

– Таблица маршрутизации **r2**.

```
10.0.10.0/24 dev eth0  proto kernel  scope link  src 10.0.10.12
10.0.20.0/24 via 10.0.10.11 dev eth0
10.0.30.0/24 dev eth1  proto kernel  scope link  src 10.0.30.32
10.0.40.0/24 via 10.0.10.11 dev eth0
10.0.50.0/24 via 10.0.30.34 dev eth1
```

– Таблица маршрутизации **r3**.

```
10.0.10.0/24 via 10.0.20.21 dev eth0
10.0.20.0/24 dev eth0  proto kernel  scope link  src 10.0.20.23
10.0.30.0/24 via 10.0.20.21 dev eth0
10.0.40.0/24 dev eth1  proto kernel  scope link  src 10.0.40.1
10.0.50.0/24 via 10.0.20.21 dev eth0
```

Показаны опыты после стирания кеша ARP.
ws1 выполняет команду

```
ping 10.0.40.2 -c 1
```

По таблице маршрутизации вычисляется, что **ws1** не имеет возможности непосредственно отправить ICMP-запрос в подсеть 10.0.40.0/24. Поэтому ICMP-запрос отправляется на маршрутизатор, IP-адрес которого известен из таблицы маршрутизации, но неизвестен MAC-адрес. Для определения MAC-адреса отправляется широковещательный ARP-запрос в интерфейс eth0, на который приходит ответ от **r2**.

```
root@ws1:~# cat eth0.log
7e:a6:65:cd:a1:ab > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has
    10.0.30.32 tell 10.0.30.2, length 28
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype ARP (0x0806), length 42: Reply 10.0.30.32 is-at
    96:e8:1f:21:5e:44, length 28
7e:a6:65:cd:a1:ab > 96:e8:1f:21:5e:44, ethertype IPv4 (0x0800), length 98: 10.0.30.2 > 10.0.40.2:
     ICMP echo request, id 1814, seq 1, length 64
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype IPv4 (0x0800), length 98: 10.0.40.2 > 10.0.30.2:
     ICMP echo reply, id 1814, seq 1, length 64
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype ARP (0x0806), length 42: Request who-has
    10.0.30.2 tell 10.0.30.32, length 28
7e:a6:65:cd:a1:ab > 96:e8:1f:21:5e:44, ethertype ARP (0x0806), length 42: Reply 10.0.30.2 is-at 7
    e:a6:65:cd:a1:ab, length 28
```

Аналогичным образом **r2** широковещательным ARP-запросом опрашивает интерфейс eth1(исходя из таблицы маршрутищации) и получает ответ от **r1** с его MAC-адресом, что позволяет направить ICMP-запрос (с TTL-1) далее.

```
root@r2:~# cat eth1.log
7e:a6:65:cd:a1:ab > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has
    10.0.30.32 tell 10.0.30.2, length 28
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype ARP (0x0806), length 42: Reply 10.0.30.32 is-at
    96:e8:1f:21:5e:44, length 28
7e:a6:65:cd:a1:ab > 96:e8:1f:21:5e:44, ethertype IPv4 (0x0800), length 98: 10.0.30.2 > 10.0.40.2:
     ICMP echo request, id 1814, seq 1, length 64
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype IPv4 (0x0800), length 98: 10.0.40.2 > 10.0.30.2:
     ICMP echo reply, id 1814, seq 1, length 64
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype ARP (0x0806), length 42: Request who-has
    10.0.30.2 tell 10.0.30.32, length 28
7e:a6:65:cd:a1:ab > 96:e8:1f:21:5e:44, ethertype ARP (0x0806), length 42: Reply 10.0.30.2 is-at 7
    e:a6:65:cd:a1:ab, length 28
```

Аналогичным образом **r1** отправляет ICMP-запрос на **r3**.

```
root@r2:~# cat eth0.log
ca:a0:5b:a0:0e:20 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has
    10.0.10.11 tell 10.0.10.12, length 28
be:14:9b:41:0f:21 > ca:a0:5b:a0:0e:20, ethertype ARP (0x0806), length 42: Reply 10.0.10.11 is-at
    be:14:9b:41:0f:21, length 28
ca:a0:5b:a0:0e:20 > be:14:9b:41:0f:21, ethertype IPv4 (0x0800), length 98: 10.0.30.2 > 10.0.40.2:
     ICMP echo request, id 1814, seq 1, length 64
be:14:9b:41:0f:21 > ca:a0:5b:a0:0e:20, ethertype IPv4 (0x0800), length 98: 10.0.40.2 > 10.0.30.2:
     ICMP echo reply, id 1814, seq 1, length 64
be:14:9b:41:0f:21 > ca:a0:5b:a0:0e:20, ethertype ARP (0x0806), length 42: Request who-has
    10.0.10.12 tell 10.0.10.11, length 28
ca:a0:5b:a0:0e:20 > be:14:9b:41:0f:21, ethertype ARP (0x0806), length 42: Reply 10.0.10.12 is-at
    ca:a0:5b:a0:0e:20, length 28

root@r1:~# cat eth0.log
ca:a0:5b:a0:0e:20 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has
    10.0.10.11 tell 10.0.10.12, length 28
be:14:9b:41:0f:21 > ca:a0:5b:a0:0e:20, ethertype ARP (0x0806), length 42: Reply 10.0.10.11 is-at
    be:14:9b:41:0f:21, length 28
```

```
ca:a0:5b:a0:0e:20 > be:14:9b:41:0f:21, ethertype IPv4 (0x0800), length 98: 10.0.30.2 > 10.0.40.2:
    ICMP echo request, id 1814, seq 1, length 64
be:14:9b:41:0f:21 > ca:a0:5b:a0:0e:20, ethertype IPv4 (0x0800), length 98: 10.0.40.2 > 10.0.30.2:
    ICMP echo reply, id 1814, seq 1, length 64
be:14:9b:41:0f:21 > ca:a0:5b:a0:0e:20, ethertype ARP (0x0806), length 42: Request who-has
    10.0.10.12 tell 10.0.10.11, length 28
ca:a0:5b:a0:0e:20 > be:14:9b:41:0f:21, ethertype ARP (0x0806), length 42: Reply 10.0.10.12 is-at
    ca:a0:5b:a0:0e:20, length 28
```

Аналогичным образом **r3** отпрявляет ICMP-запрос на **ws2**

```
root@r1:~# cat eth1.log
3a:bf:72:36:29:d9 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has
    10.0.20.23 tell 10.0.20.21, length 28
ee:93:78:86:e8:49 > 3a:bf:72:36:29:d9, ethertype ARP (0x0806), length 42: Reply 10.0.20.23 is-at
    ee:93:78:86:e8:49, length 28
3a:bf:72:36:29:d9 > ee:93:78:86:e8:49, ethertype IPv4 (0x0800), length 98: 10.0.30.2 > 10.0.40.2:
    ICMP echo request, id 1814, seq 1, length 64
ee:93:78:86:e8:49 > 3a:bf:72:36:29:d9, ethertype IPv4 (0x0800), length 98: 10.0.40.2 > 10.0.30.2:
    ICMP echo reply, id 1814, seq 1, length 64
ee:93:78:86:e8:49 > 3a:bf:72:36:29:d9, ethertype ARP (0x0806), length 42: Request who-has
    10.0.20.21 tell 10.0.20.23, length 28
3a:bf:72:36:29:d9 > ee:93:78:86:e8:49, ethertype ARP (0x0806), length 42: Reply 10.0.20.21 is-at
    3a:bf:72:36:29:d9, length 28

root@r3:~# cat eth0.log
3a:bf:72:36:29:d9 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has
    10.0.20.23 tell 10.0.20.21, length 28
ee:93:78:86:e8:49 > 3a:bf:72:36:29:d9, ethertype ARP (0x0806), length 42: Reply 10.0.20.23 is-at
    ee:93:78:86:e8:49, length 28
3a:bf:72:36:29:d9 > ee:93:78:86:e8:49, ethertype IPv4 (0x0800), length 98: 10.0.30.2 > 10.0.40.2:
    ICMP echo request, id 1814, seq 1, length 64
ee:93:78:86:e8:49 > 3a:bf:72:36:29:d9, ethertype IPv4 (0x0800), length 98: 10.0.40.2 > 10.0.30.2:
    ICMP echo reply, id 1814, seq 1, length 64
ee:93:78:86:e8:49 > 3a:bf:72:36:29:d9, ethertype ARP (0x0806), length 42: Request who-has
    10.0.20.21 tell 10.0.20.23, length 28
3a:bf:72:36:29:d9 > ee:93:78:86:e8:49, ethertype ARP (0x0806), length 42: Reply 10.0.20.21 is-at
    3a:bf:72:36:29:d9, length 28
```

ws2 получает ICMP-запрос и отправляет ICMP-ответ. На обратном пути следования ответа нет необходимости направлять широковещательные ARP-запросы для поиска MAC-адресов, т.к. ARP-кеш является заполненным.

```
root@ws2:~# cat eth0.log
76:e5:80:0f:a1:02 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has
    10.0.40.2 tell 10.0.40.1, length 28
42:70:9a:0b:39:80 > 76:e5:80:0f:a1:02, ethertype ARP (0x0806), length 42: Reply 10.0.40.2 is-at
    42:70:9a:0b:39:80, length 28
76:e5:80:0f:a1:02 > 42:70:9a:0b:39:80, ethertype IPv4 (0x0800), length 98: 10.0.30.2 > 10.0.40.2:
    ICMP echo request, id 1814, seq 1, length 64
42:70:9a:0b:39:80 > 76:e5:80:0f:a1:02, ethertype IPv4 (0x0800), length 98: 10.0.40.2 > 10.0.30.2:
    ICMP echo reply, id 1814, seq 1, length 64
```

```
42:70:9a:0b:39:80 > 76:e5:80:0f:a1:02, ethertype ARP (0x0806), length 42: Request who-has
    10.0.40.1 tell 10.0.40.2, length 28
76:e5:80:0f:a1:02 > 42:70:9a:0b:39:80, ethertype ARP (0x0806), length 42: Reply 10.0.40.1 is-at
    76:e5:80:0f:a1:02, length 28
```

6. Продолжительность жизни пакета

Добавим кольцо между **r1** и **r2**:

```
ip r del 10.0.40.0/24
ip r add 10.0.40.0/24 via 10.0.10.12 dev eth0
```

Получим таблицу маршрутизации **r1**:

```
10.0.10.0/24 dev eth0 proto kernel scope link src 10.0.10.11
10.0.20.0/24 dev eth1 proto kernel scope link src 10.0.20.21
10.0.30.0/24 via 10.0.10.12 dev eth0
10.0.40.0/24 via 10.0.10.12 dev eth0
10.0.50.0/24 via 10.0.10.12 dev eth0
```

Отправим ICMP-запрос с **ws1** на **ws2**:

```
root@ws1:~# ping 10.0.40.2 -c 1
PING 10.0.40.2 (10.0.40.2) 56(84) bytes of data.
From 10.0.10.11 icmp_seq=1 Time to live exceeded
```

Воспользуемся командой tcpdump: **ws1**:

```
7e:a6:65:cd:a1:ab > 96:e8:1f:21:5e:44, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype IPv4 (0x0800), length 126: (tos 0xc0, ttl 63, id
    20446, offset 0, flags [none], proto ICMP (1), length 112)
    10.0.10.11 > 10.0.30.2: ICMP time exceeded in-transit, length 92
        (tos 0x0, ttl 1, id 32160, offset 0, flags [DF], proto ICMP (1), length 84)
```

r2.eth1:

```
7e:a6:65:cd:a1:ab > 96:e8:1f:21:5e:44, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
96:e8:1f:21:5e:44 > 7e:a6:65:cd:a1:ab, ethertype IPv4 (0x0800), length 126: (tos 0xc0, ttl 63, id
    20446, offset 0, flags [none], proto ICMP (1), length 112)
    10.0.10.11 > 10.0.30.2: ICMP time exceeded in-transit, length 92
        (tos 0x0, ttl 1, id 32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
```

r2.eth0:

```
ca:a0:5b:a0:0e:20 > 3a:71:d5:2e:8b:c0, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
3a:71:d5:2e:8b:c0 > ca:a0:5b:a0:0e:20, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
```

```
ca:a0:5b:a0:0e:20 > 3a:71:d5:2e:8b:c0, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 61, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
3a:71:d5:2e:8b:c0 > ca:a0:5b:a0:0e:20, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 60, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
ca:a0:5b:a0:0e:20 > 3a:71:d5:2e:8b:c0, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 59, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
```

r1:

```
ca:a0:5b:a0:0e:20 > 3a:71:d5:2e:8b:c0, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
3a:71:d5:2e:8b:c0 > ca:a0:5b:a0:0e:20, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
ca:a0:5b:a0:0e:20 > 3a:71:d5:2e:8b:c0, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 61, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
3a:71:d5:2e:8b:c0 > ca:a0:5b:a0:0e:20, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 60, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
ca:a0:5b:a0:0e:20 > 3a:71:d5:2e:8b:c0, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 59, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
3a:71:d5:2e:8b:c0 > ca:a0:5b:a0:0e:20, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 58, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
ca:a0:5b:a0:0e:20 > 3a:71:d5:2e:8b:c0, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 57, id
    32160, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.30.2 > 10.0.40.2: ICMP echo request, id 1874, seq 1, length 64
```

r1 отправил сообщение о превышении времени жизни.

7. Изучение IP-фрагментации

ws1

```
ip l set mtu 500 dev eth0
```

r2

```
ip l set mtu 500 dev eth1
```

r1

```
ping 10.0.30.2 -c 1 -s 1000
```

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

```
root@r2:~# cat eth1.log
IP (tos 0x0, ttl 63, id 40191, offset 0, flags [+], proto ICMP (1), length 500)
    10.0.10.11 > 10.0.30.2: ICMP echo request, id 1865, seq 1, length 480
```

```
IP (tos 0x0, ttl 63, id 40191, offset 480, flags [+], proto ICMP (1), length 500)
    10.0.10.11 > 10.0.30.2: ip-proto-1
IP (tos 0x0, ttl 63, id 40191, offset 960, flags [none], proto ICMP (1), length 68)
    10.0.10.11 > 10.0.30.2: ip-proto-1
IP (tos 0x0, ttl 64, id 12026, offset 0, flags [+], proto ICMP (1), length 500)
    10.0.30.2 > 10.0.10.11: ICMP echo reply, id 1865, seq 1, length 480
IP (tos 0x0, ttl 64, id 12026, offset 480, flags [+], proto ICMP (1), length 500)
    10.0.30.2 > 10.0.10.11: ip-proto-1
IP (tos 0x0, ttl 64, id 12026, offset 960, flags [none], proto ICMP (1), length 68)
    10.0.30.2 > 10.0.10.11: ip-proto-1
```

Вывод **tcpdump** на маршрутизаторе **r2** после сети с уменьшенным MTU.

```
root@r2:~# cat eth0.log
IP (tos 0x0, ttl 64, id 40191, offset 0, flags [none], proto ICMP (1), length 1028)
    10.0.10.11 > 10.0.30.2: ICMP echo request, id 1865, seq 1, length 1008
IP (tos 0x0, ttl 63, id 12026, offset 0, flags [+], proto ICMP (1), length 500)
    10.0.30.2 > 10.0.10.11: ICMP echo reply, id 1865, seq 1, length 480
IP (tos 0x0, ttl 63, id 12026, offset 480, flags [+], proto ICMP (1), length 500)
    10.0.30.2 > 10.0.10.11: ip-proto-1
IP (tos 0x0, ttl 63, id 12026, offset 960, flags [none], proto ICMP (1), length 68)
    10.0.30.2 > 10.0.10.11: ip-proto-1
```

Вывод **tcpdump** на узле получателя **ws1**.

```
root@ws1:~# cat eth0.log
IP (tos 0x0, ttl 63, id 40191, offset 0, flags [+], proto ICMP (1), length 500)
    10.0.10.11 > 10.0.30.2: ICMP echo request, id 1865, seq 1, length 480
IP (tos 0x0, ttl 63, id 40191, offset 480, flags [+], proto ICMP (1), length 500)
    10.0.10.11 > 10.0.30.2: ip-proto-1
IP (tos 0x0, ttl 63, id 40191, offset 960, flags [none], proto ICMP (1), length 68)
    10.0.10.11 > 10.0.30.2: ip-proto-1
IP (tos 0x0, ttl 64, id 12026, offset 0, flags [+], proto ICMP (1), length 500)
    10.0.30.2 > 10.0.10.11: ICMP echo reply, id 1865, seq 1, length 480
IP (tos 0x0, ttl 64, id 12026, offset 480, flags [+], proto ICMP (1), length 500)
    10.0.30.2 > 10.0.10.11: ip-proto-1
IP (tos 0x0, ttl 64, id 12026, offset 960, flags [none], proto ICMP (1), length 68)
    10.0.30.2 > 10.0.10.11: ip-proto-1
```

8. Отсутствие сети

```
root@ws3:~# ping 10.0.70.2
PING 10.0.70.2 (10.0.70.2) 56(84) bytes of data.
From 10.0.50.1 icmp_seq=1 Destination Net Unreachable
From 10.0.50.1 icmp_seq=2 Destination Net Unreachable
From 10.0.50.1 icmp_seq=3 Destination Net Unreachable
```

9. Отсутствие IP-адреса в сети

```
root@ws3:~# ping 10.0.40.3
```

```
PING 10.0.40.3 (10.0.40.3) 56(84) bytes of data.
From 10.0.20.23 icmp_seq=1 Destination Host Unreachable
From 10.0.20.23 icmp_seq=2 Destination Host Unreachable
From 10.0.20.23 icmp_seq=3 Destination Host Unreachable
```