

# Отчёт по лабораторной работе «IP-маршрутизация»

Фроловский Алексей Вадимович

4 октября 2013 г.

## Содержание

1. Топология сети	1
2. Назначение IP-адресов	1
3. Таблица маршрутизации	4
4. Проверка настройки сети	5
5. Маршрутизация	6
6. Продолжительность жизни пакета	8
7. Изучение IP-фрагментации	9
8. Отсутствие сети	10
9. Отсутствие IP-адреса в сети	11

## 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 192.168.11.1
netmask 255.255.255.0
```

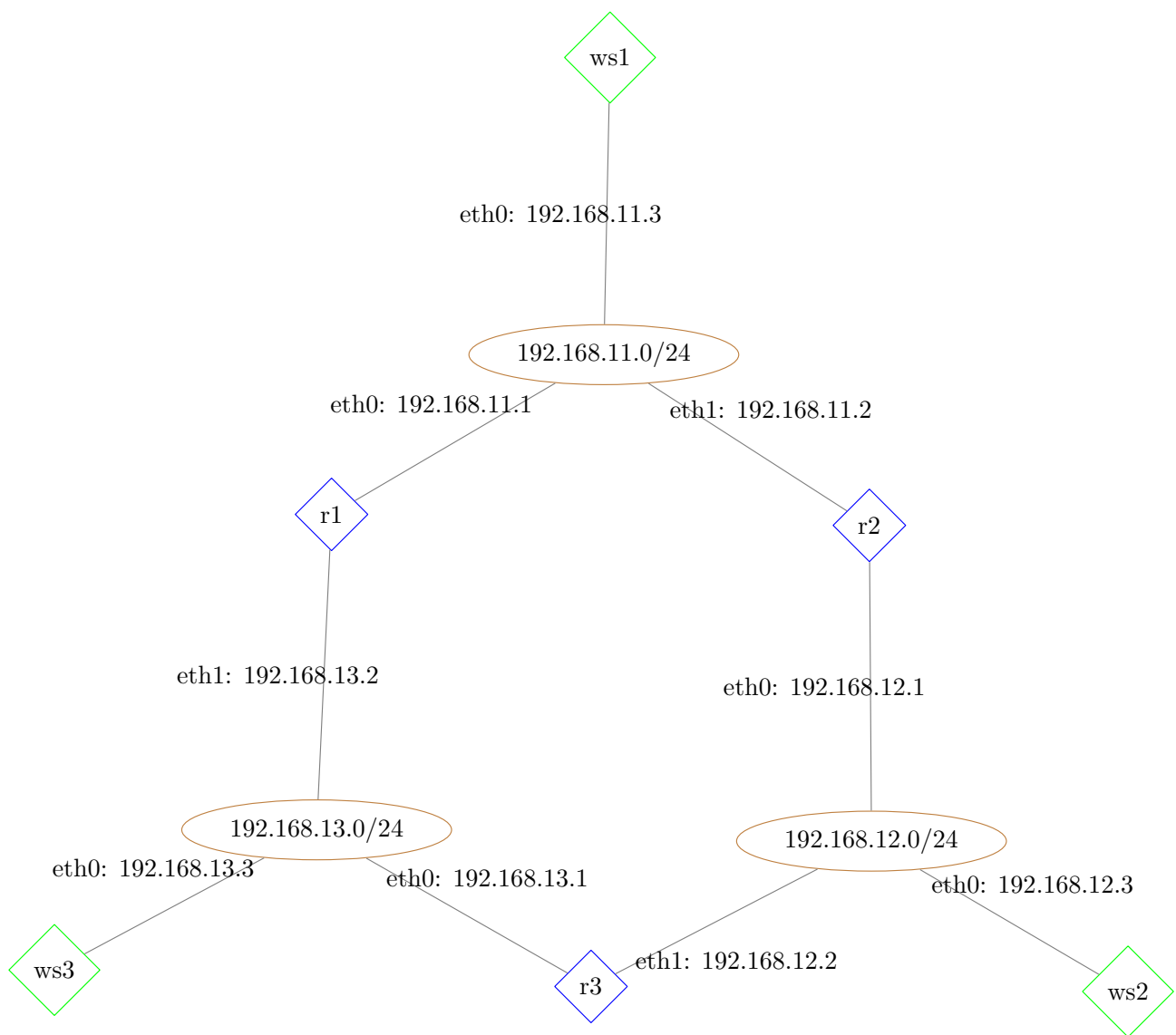


Рис. 1. Топология сети

```
up ip r add 192.168.12.0/24 via 192.168.11.2 dev eth0
down ip r del 192.168.12.0/24

auto eth1
iface eth1 inet static
address 192.168.13.2
netmask 255.255.255.0
```

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

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.12.1
netmask 255.255.255.0
up ip r add 192.168.13.0/24 via 192.168.12.2 dev eth0
down ip r del 192.168.13.0/24

auto eth1
iface eth1 inet static
address 192.168.11.2
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 192.168.13.1
netmask 255.255.255.0
up ip r add 192.168.11.0/24 via 192.168.13.2 dev eth0
down ip r del 192.168.11.0/24

auto eth1
iface eth1 inet static
address 192.168.12.2
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 192.168.11.3
netmask 255.255.255.0
```

```
up ip r add 192.168.12.0/24 via 192.168.11.2 dev eth0
down ip r del 192.168.12.0/24
gateway 192.168.11.1
```

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

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.12.3
netmask 255.255.255.0
up ip r add 192.168.13.0/24 via 192.168.12.2 dev eth0
down ip r del 192.168.13.0/24
gateway 192.168.12.1
```

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

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.13.3
netmask 255.255.255.0
up ip r add 192.168.11.0/24 via 192.168.13.2 dev eth0
down ip r del 192.168.11.0/24
gateway 192.168.13.1
```

### 3. Таблица маршрутизации

Ниже показана таблица маршрутизации для маршрутизатора **r1**:

```
192.168.13.0/24 dev eth1 proto kernel scope link src 192.168.13.2
192.168.12.0/24 via 192.168.11.2 dev eth0
192.168.11.0/24 dev eth0 proto kernel scope link src 192.168.11.1
```

Ниже показана таблица маршрутизации для маршрутизатора **r2**:

```
192.168.13.0/24 via 192.168.12.2 dev eth0
192.168.12.0/24 dev eth0 proto kernel scope link src 192.168.12.1
192.168.11.0/24 dev eth1 proto kernel scope link src 192.168.11.2
```

Ниже показана таблица маршрутизации для маршрутизатора **r3**:

```
192.168.13.0/24 dev eth0 proto kernel scope link src 192.168.13.1
192.168.12.0/24 dev eth1 proto kernel scope link src 192.168.12.2
192.168.11.0/24 via 192.168.13.2 dev eth0
```

Ниже показана таблица маршрутизации для рабочей станции **ws1**:

```
192.168.12.0/24 via 192.168.11.2 dev eth0
192.168.11.0/24 dev eth0 proto kernel scope link src 192.168.11.3
default via 192.168.11.1 dev eth0
```

Ниже показана таблица маршрутизации для рабочей станции **ws2**:

```
192.168.13.0/24 via 192.168.12.2 dev eth0
192.168.12.0/24 dev eth0 proto kernel scope link src 192.168.12.3
default via 192.168.12.1 dev eth0
```

Ниже показана таблица маршрутизации для рабочей станции **ws3**:

```
192.168.13.0/24 dev eth0 proto kernel scope link src 192.168.13.3
192.168.11.0/24 via 192.168.13.2 dev eth0
default via 192.168.13.1 dev eth0
```

## 4. Проверка настройки сети

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

```
traceroute to 192.168.12.3 (192.168.12.3), 64 hops max, 40 byte packets
 1  192.168.11.2  6 ms  1 ms  1 ms
 2  192.168.12.3 13 ms  1 ms  1 ms
```

Вывод **traceroute** от маршрутизатора **r1** до узла **ws2** при нормальной работе сети:

```
traceroute to 192.168.12.3 (192.168.12.3), 64 hops max, 40 byte packets
 1  192.168.11.2  7 ms  0 ms  0 ms
 2  192.168.12.3 11 ms  1 ms  0 ms
```

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

```
traceroute to 192.168.13.3 (192.168.13.3), 64 hops max, 40 byte packets
 1  192.168.11.1  6 ms  1 ms  1 ms
 2  192.168.13.3 12 ms  1 ms  1 ms
```

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

```
traceroute to 192.168.11.3 (192.168.11.3), 64 hops max, 40 byte packets
 1  192.168.12.1  1 ms  1 ms  1 ms
 2  192.168.11.3  2 ms  2 ms  1 ms
```

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

```
traceroute to 192.168.13.3 (192.168.13.3), 64 hops max, 40 byte packets
 1  192.168.12.2 13 ms  1 ms  1 ms
 2  192.168.13.3 13 ms  1 ms  1 ms
```

Вывод **traceroute** от маршрутизатора **r2** до узла **ws3** при нормальной работе сети:

```
tracert to 192.168.13.3 (192.168.13.3), 64 hops max, 40 byte packets
 1  192.168.12.2  3 ms  0 ms  0 ms
 2  192.168.13.3 11 ms  0 ms  0 ms
```

Вывод **tracert** от узла **ws3** до узла **ws1** при нормальной работе сети:

```
tracert to 192.168.11.3 (192.168.11.3), 64 hops max, 40 byte packets
 1  192.168.13.2  1 ms  1 ms  1 ms
 2  192.168.11.3  2 ms  2 ms  1 ms
```

Вывод **tracert** от маршрутизатора **r3** до узла **ws1** при нормальной работе сети:

```
tracert to 192.168.11.3 (192.168.11.3), 64 hops max, 40 byte packets
 1  192.168.13.2  1 ms  1 ms  0 ms
 2  192.168.11.3  1 ms  1 ms  1 ms
```

Вывод **tracert** от узла **ws3** до узла **ws2** при нормальной работе сети:

```
tracert to 192.168.12.3 (192.168.12.3), 64 hops max, 40 byte packets
 1  192.168.13.1  1 ms  1 ms  1 ms
 2  192.168.12.3  2 ms  1 ms  2 ms
```

## 5. Маршрутизация

Маршрутизатор **r1** имеет следующие параметры сетевых интерфейсов:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: teql0: <NOARP> mtu 1500 qdisc noop qlen 100
    link/void
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 0e:ab:f8:0c:10:4b brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.1/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::cab:f8ff:fe0c:104b/64 scope link
        valid_lft forever preferred_lft forever
4: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether fa:de:dc:30:96:57 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.2/24 brd 192.168.13.255 scope global eth1
    inet6 fe80::f8de:dcff:fe30:9657/64 scope link
        valid_lft forever preferred_lft forever
```

Таблица маршрутизации для **r1** имеет следующий вид:

```
192.168.13.0/24 dev eth1 proto kernel scope link src 192.168.13.2
192.168.12.0/24 via 192.168.11.2 dev eth0
192.168.11.0/24 dev eth0 proto kernel scope link src 192.168.11.1
```

Маршрутизатор **r2** имеет следующие параметры сетевых интерфейсов:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: teql0: <NOARP> mtu 1500 qdisc noop qlen 100
    link/void
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 3a:40:ee:31:9e:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.12.1/24 brd 192.168.12.255 scope global eth0
    inet6 fe80::3840:eeff:fe31:9ecd/64 scope link
        valid_lft forever preferred_lft forever
4: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 12:3e:e2:7d:e3:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.2/24 brd 192.168.11.255 scope global eth1
    inet6 fe80::103e:e2ff:fe7d:e387/64 scope link
        valid_lft forever preferred_lft forever
```

Таблица маршрутизации для **r2** имеет следующий вид:

```
192.168.13.0/24 via 192.168.12.2 dev eth0
192.168.12.0/24 dev eth0 proto kernel scope link src 192.168.12.1
192.168.11.0/24 dev eth1 proto kernel scope link src 192.168.11.2
```

Рабочая станция **ws1** имеет следующие параметры сетевых интерфейсов:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: teql0: <NOARP> mtu 1500 qdisc noop qlen 100
    link/void
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether a6:f9:52:b6:1e:69 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.3/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a4f9:52ff:feb6:1e69/64 scope link
        valid_lft forever preferred_lft forever
```

Таблица маршрутизации для **ws1** имеет следующий вид:

```
192.168.12.0/24 via 192.168.11.2 dev eth0
192.168.11.0/24 dev eth0 proto kernel scope link src 192.168.11.3
default via 192.168.11.1 dev eth0
```

Рабочая станция **ws2** имеет следующие параметры сетевых интерфейсов:

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: tegl0: <NOARP> mtu 1500 qdisc noop qlen 100
    link/void
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether da:53:12:09:ea:4e brd ff:ff:ff:ff:ff:ff
    inet 192.168.12.3/24 brd 192.168.12.255 scope global eth0
    inet6 fe80::d853:12ff:fe09:ea4e/64 scope link
        valid_lft forever preferred_lft forever

```

Далее показана отправка пакета с рабочей станции **ws1** на рабочую станцию **ws2** (после стирания кеша ARP) для демонстрации косвенной маршрутизации:

```
| ping 192.168.12.3 -c 1
```

На первом шаге рабочая станция **ws1** определяет по своей маршрутной таблице, что пакет нужно отправить на адрес 192.168.11.2. Затем рабочая станция **ws1** пытается узнать MAC-адрес с помощью протокола ARP и отправляет сам пакет на маршрутизатор **r2**:

```

a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42:
    arp who-has 192.168.11.2 tell 192.168.11.3
12:3e:e2:7d:e3:87 > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42:
    arp reply 192.168.11.2 is-at 12:3e:e2:7d:e3:87
a6:f9:52:b6:1e:69 > 12:3e:e2:7d:e3:87, ethertype IPv4 (0x0800), length 98:
    192.168.11.3 > 192.168.12.3: ICMP echo request, id 24322, seq 1, length 64

```

После чего маршрутизатор **r2**, находящийся в одном сегменте сети с рабочей станцией **ws2**, определяет MAC-адрес получателя и отправляет непосредственно сам пакет получателю:

```

3a:40:ee:31:9e:cd > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42:
    arp who-has 192.168.12.3 tell 192.168.12.1
da:53:12:09:ea:4e > 3a:40:ee:31:9e:cd, ethertype ARP (0x0806), length 42:
    arp reply 192.168.12.3 is-at da:53:12:09:ea:4e
3a:40:ee:31:9e:cd > da:53:12:09:ea:4e, ethertype IPv4 (0x0800), length 98:
    192.168.11.3 > 192.168.12.3: ICMP echo request, id 24322, seq 1, length 64

```

## 6. Продолжительность жизни пакета

Создадим маршрутную петлю между маршрутизаторами **r1** и **r2**. Для этого отключим на маршрутизаторе **r2** интерфейс eth0 и добавим неверный маршрут до сети 192.168.12.0:

```

ip link set eth0 down
ip route add 192.168.12.0/24 via 192.168.11.1 dev eth1

```

После чего таблица маршрутизации для маршрутизатора **r2** будет иметь следующий вид:



```
| 192.168.12.0/24 via 192.168.11.1 dev eth1
| 192.168.11.0/24 dev eth1 proto kernel scope link src 192.168.11.2
```

С рабочей станции **ws1** пошлем эхо-запрос на рабочую станцию **ws2**:

```
| ping 192.168.12.3 -c 1
```

Ниже приведены перемещения пакета с эхо-запросом по маршрутной петле, из которых видно, что сообщение об истечении жизни пакета было отправлено с маршрутизатора **r1**:

```
| a6:f9:52:b6:1e:69 > 12:3e:e2:7d:e3:87, ethertype IPv4 (0x0800), length 98:
|   (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
|   192.168.11.3 > 192.168.12.3: ICMP echo request, id 8450, seq 1, length 64
| 12:3e:e2:7d:e3:87 > 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)
|   192.168.11.3 > 192.168.12.3: ICMP echo request, id 8450, seq 1, length 64
| :
| 0e:ab:f8:0c:10:4b > 12:3e:e2:7d:e3:87, ethertype IPv4 (0x0800), length 98:
|   (tos 0x0, ttl 2, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
|   192.168.11.3 > 192.168.12.3: ICMP echo request, id 8450, seq 1, length 64
| 12:3e:e2:7d:e3:87 > 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)
|   192.168.11.3 > 192.168.12.3: ICMP echo request, id 8450, seq 1, length 64
| 0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype IPv4 (0x0800), length 126:
|   (tos 0xc0, ttl 64, id 15633, offset 0, flags [none], proto ICMP (1), length 112)
|   192.168.11.1 > 192.168.11.3: ICMP time exceeded in-transit, length 92
|     (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
|   192.168.11.3 > 192.168.12.3: ICMP echo request, id 8450, seq 1, length 64
```

## 7. Изучение IP-фрагментации

Для изучения фрагментации IP-пакетов выполним следующие действия:

- Изменим маршрут следования пакетов между узлами **ws1** и **ws2** сети таким образом, чтобы пакеты, между этими сетями проходили через маршрутизаторы **r1** и **r3**. Для этого выполним следующие действия:

- Удалим маршрут следования пакетов из сети 192.168.11.0/24 в сеть 192.168.12.0/24 через маршрутизатор **r2** из маршрутной таблицы рабочей станции **ws1** и маршрутизатора **r1**:

```
| ip route del 192.168.12.0/24 via 192.168.11.2 dev eth0
```

- Добавим маршрут следования пакетов из сети 192.168.11.0/24 в сеть 192.168.12.0/24 через маршрутизатор **r3** в маршрутную таблицу маршрутизатора **r1**:

```
| ip route add 192.168.12.0/24 via 192.168.13.1 dev eth1
```

- Уменьшим величину MTU в сети между маршрутизаторами **r1** и **r3**. Поскольку все подключенные к одному сегменту сетевые интерфейсы должны иметь одинаковое представление о величине MTU сегмента, то выполним следующие действия:

- Уменьшим величину MTU на интерфейсе eth1 маршрутизатора **r1**:

```
|ip link set dev eth1 mtu 576
```

- Уменьшим величину MTU на интерфейсе eth0 маршрутизатора **r3** и рабочей станции **ws3**:

```
|ip link set dev eth0 mtu 576
```

- Отключим механизм борьбы с фрагментацией на рабочей станции **ws1**:

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

Для тестирования отправим эхо-запрос с рабочей станции **ws1** на рабочую станцию **ws2**:

```
|ping -c 1 -s 1000 192.168.12.3
```

Вывод **tcpdump** на маршрутизаторе **r1** (перед сетью с уменьшенным MTU):

```
|IP (tos 0x0, ttl 64, id 16587, offset 0, flags [none], proto ICMP (1), length 1028)
  192.168.11.3 > 192.168.12.3: ICMP echo request, id 11266, seq 1, length 1008
```

Вывод **tcpdump** на маршрутизаторе **r3** (после сети с уменьшенным MTU):

```
|IP (tos 0x0, ttl 63, id 16587, offset 0, flags [+], proto ICMP (1), length 572)
  192.168.11.3 > 192.168.12.3: ICMP echo request, id 11266, seq 1, length 552
|IP (tos 0x0, ttl 63, id 16587, offset 552, flags [none], proto ICMP (1), length 476)
  192.168.11.3 > 192.168.12.3: icmp
```

Вывод **tcpdump** на узле получателя:

```
|IP (tos 0x0, ttl 62, id 16587, offset 0, flags [none], proto ICMP (1), length 1028)
  192.168.11.3 > 192.168.12.3: ICMP echo request, id 11266, seq 1, length 1008
```

## 8. Отсутствие сети

Для проведения опыта отключим на маршрутизаторе **r2** интерфейс eth0:

```
|ip link set eth0 down
```

Затем отправим пакет с эхо-запросом с рабочей станции **ws1** на рабочую станцию **ws2**:

```
|ping 192.168.12.3 -c 1
```

Вывод **ping** на рабочей станции **ws1**:

```

PING 192.168.12.3 (192.168.12.3) 56(84) bytes of data.
From 192.168.11.2 icmp_seq=1 Destination Net Unreachable

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

```

Вывод **tcpdump** на маршрутизаторе **r2**:

```

IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
  192.168.11.3 > 192.168.12.3: ICMP echo request, id 5378, seq 1, length 64
IP (tos 0xc0, ttl 64, id 60546, offset 0, flags [none], proto ICMP (1), length 112)
  192.168.11.2 > 192.168.11.3: ICMP net 192.168.12.3 unreachable, length 92
    IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
      192.168.11.3 > 192.168.12.3: ICMP echo request, id 5378, seq 1, length 64

```

## 9. Отсутствие IP-адреса в сети

Для проведения опыта отправим пакет с эхо-запросом с рабочей станции **ws1** в сеть 192.168.12.0/24 на несуществующий узел 192.168.12.4:

```

ping 192.168.12.4 -c 1

```

Вывод **ping** на рабочей станции **ws1**:

```

PING 192.168.12.4 (192.168.12.4) 56(84) bytes of data.
From 192.168.11.2 icmp_seq=1 Destination Host Unreachable

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

```

Вывод **tcpdump** на маршрутизаторе **r2**:

```

IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
  192.168.11.3 > 192.168.12.4: ICMP echo request, id 5634, seq 1, length 64
IP (tos 0xc0, ttl 64, id 36124, offset 0, flags [none], proto ICMP (1), length 112)
  192.168.11.2 > 192.168.11.3: ICMP host 192.168.12.4 unreachable, length 92
    IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
      192.168.11.3 > 192.168.12.4: ICMP echo request, id 5634, seq 1, length 64

```