

# Отчёт по лабораторной работе «IP-маршрутизация»

Binh D. Nguyen

28 сентября 2019 г.

## Содержание

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

## 1. Топология сети

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

## 2. Назначение IP-адресов

Ниже приведён файл настройки протокола IP маршрутизатора (указать, какого).

| Сюда нужно поместить характерный `/etc/network/interfaces` маршрутизатора

Ниже приведён файл настройки протокола IP рабочей станции (указать, какой).

| Сюда нужно поместить характерный `/etc/network/interfaces` рабочей станции

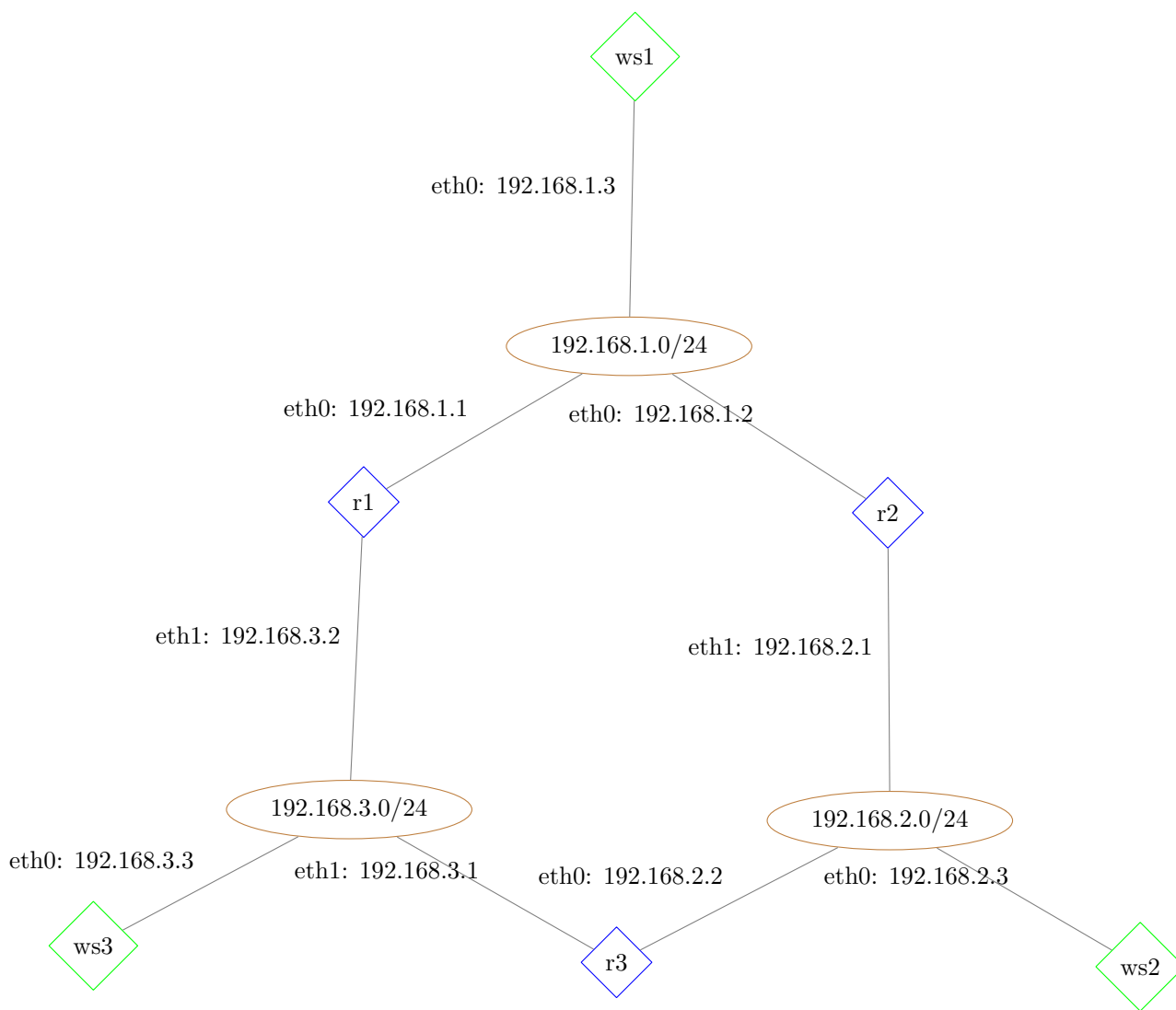


Рис. 1. Топология сети

### 3. Таблица маршрутизации

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

```
192.168.3.0/24 dev eth1 proto kernel scope link src 192.168.3.2
192.168.2.0/24 via 192.168.1.2 dev eth0
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.1
```

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

```
192.168.3.0/24 via 192.168.1.1 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 192.168.1.2
```

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

```
192.168.3.0/24 dev eth1 proto kernel scope link src 192.168.3.1
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 **ws1**.

```
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.3
default via 192.168.1.1 dev eth0
```

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

```
192.168.2.0/24 dev eth0 proto kernel scope link src 192.168.2.3
default via 192.168.2.1 dev eth0
```

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

```
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.3
default via 192.168.3.1 dev eth0
```

### 4. Проверка настройки сети

Вывод **tracert** от узла **ws1** такого-то до такого-то при нормальной работе сети.

```
ws1:~# tracert -n 192.168.2.3
tracert to 192.168.2.3 (192.168.2.3), 64 hops max, 40 byte packets
 1  192.168.1.1  5 ms  1 ms  1 ms
 2  192.168.1.2 21 ms  1 ms  1 ms
 3  192.168.2.3 13 ms  1 ms  1 ms
ws1:~# tracert -n 192.168.3.3
tracert to 192.168.3.3 (192.168.3.3), 64 hops max, 40 byte packets
 1  192.168.1.1  1 ms  0 ms  0 ms
 2  192.168.3.3 40 ms  3 ms  3 ms
```

Вывод **tracert** от узла **ws2** такого-то до такого-то при нормальной работе сети.

```
ws2:~# traceroute -n 192.168.1.3
traceroute to 192.168.1.3 (192.168.1.3), 64 hops max, 40 byte packets
 1  192.168.2.1  1 ms  1 ms  1 ms
 2  192.168.1.3  1 ms  1 ms  1 ms
ws2:~# traceroute -n 192.168.3.3
traceroute to 192.168.3.3 (192.168.3.3), 64 hops max, 40 byte packets
 1  192.168.2.1  1 ms  1 ms  9 ms
 2  192.168.1.1  11 ms  1 ms  1 ms
 3  192.168.3.3  13 ms  2 ms  1 ms
```

Вывод **traceroute** от узла **ws3** такого-то до такого-то при нормальной работе сети.

```
ws3:~# traceroute -n 192.168.1.3
traceroute to 192.168.1.3 (192.168.1.3), 64 hops max, 40 byte packets
 1  192.168.3.1  1 ms  1 ms  1 ms
 2  192.168.1.2  1 ms  1 ms  1 ms
 3  192.168.1.3  1 ms  1 ms  1 ms
ws3:~# traceroute -n 192.168.2.3
traceroute to 192.168.2.3 (192.168.2.3), 64 hops max, 40 byte packets
 1  192.168.3.1  1 ms  1 ms  0 ms
 2  192.168.2.3  1 ms  1 ms  1 ms
```

## 5. Маршрутизация

Тестирование на MAC-адреса интерфейс **r1**:

```
192.168.3.0/24 dev eth1  proto kernel  scope link  src 192.168.3.2
192.168.2.0/24 via 192.168.1.2 dev eth0
192.168.1.0/24 dev eth0  proto kernel  scope link  src 192.168.1.1
```

Для стирания кеша ARP, выполняется следующая команда:

```
ip n flush all
```

Далее показана отправка пакета на маршрутизатор (косвенная маршрутизация).

```
a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.1.3:
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42: arp reply 192.168.1.3:
a6:f9:52:b6:1e:69 > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: 192.168.1.3 > 192.168.1.2
```

Затем маршрутизатор отправил его далее.

```
0e:ab:f8:0c:10:4b > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.1.2:
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype ARP (0x0806), length 42: arp reply 192.168.1.2:
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: 192.168.1.3 > 192.168.1.2
3a:40:ee:31:9e:cd > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.1.2:
```

## 6. Продолжительность жизни пакета

Сначала написать как и на чём ломали.

```
r1:~# ip l set eth1 down
r1:~# ip r add 192.168.3.0/24 via 192.168.1.2
```

Потом какая-то таблица вышла.

```
192.168.3.0/24 via 192.168.1.2 dev eth0
192.168.2.0/24 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 1 192.168.3.3
PING 192.168.3.3 (192.168.3.3) 56(84) bytes of data.
From 192.168.1.2 icmp_seq=1 Time to live exceeded

--- 192.168.3.3 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

И что в итоге получилось.

```
r1:~# tcpdump -tnev
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
a6:f9:52:b6:1e:69 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.1.1 tell a6:f9:52:b6:1e:69
0e:ab:f8:0c:10:4b > a6:f9:52:b6:1e:69, ethertype ARP (0x0806), length 42: arp reply 192.168.1.1 from 0e:ab:f8:0c:10:4b
a6:f9:52:b6:1e:69 > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: arp who-has 192.168.1.2 tell 0e:ab:f8:0c:10:4b
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype ARP (0x0806), length 42: arp reply 192.168.1.2 from 3a:40:ee:31:9e:cd
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 61, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 60, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 59, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 58, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 57, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 56, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 55, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 54, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 53, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 52, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 51, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 50, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 49, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 48, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 47, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 46, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
0e:ab:f8:0c:10:4b > 3a:40:ee:31:9e:cd, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 45, id 65535, len 90, opt. 0) 192.168.1.1 > 192.168.3.3: ICMP: echo request seq=1
3a:40:ee:31:9e:cd > 0e:ab:f8:0c:10:4b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 44, id 65535, len 90, opt. 0) 192.168.1.2 > 192.168.3.3: ICMP: echo request seq=1
```

[illegible]

И кто в итоге отправил сообщение о завершении жизни.

```

ws1:~# traceroute -n 192.168.3.3
traceroute to 192.168.3.3 (192.168.3.3), 64 hops max, 40 byte packets
 1 192.168.1.1 1 ms 1 ms 0 ms
 2 192.168.1.2 1 ms 1 ms 1 ms
 3 192.168.1.1 1 ms 1 ms 1 ms
 4 192.168.1.2 1 ms 1 ms 1 ms
 5 * 192.168.1.1 8 ms 5 ms
 6 192.168.1.2 4 ms 7 ms 4 ms
 7 * 192.168.1.1 6 ms 6 ms
 8 192.168.1.2 6 ms 5 ms *
 9 192.168.1.1 7 ms 6 ms 7 ms
10 192.168.1.2 8 ms 8 ms 6 ms
11 192.168.1.1 3 ms * 6 ms
12 192.168.1.2 8 ms 8 ms 9 ms
13 192.168.1.1 13 ms 4 ms *
14 192.168.1.2 8 ms 8 ms 7 ms
15 192.168.1.1 13 ms 5 ms 5 ms
16 192.168.1.2 5 ms * 11 ms
17 192.168.1.1 18 ms 8 ms 5 ms
18 192.168.1.2 6 ms 5 ms *
19 192.168.1.1 11 ms 11 ms 13 ms
20 192.168.1.2 8 ms 8 ms 6 ms
21 192.168.1.1 7 ms * 12 ms
22 192.168.1.2 14 ms 15 ms 8 ms
23 192.168.1.1 7 ms 9 ms *
24 192.168.1.2 15 ms 10 ms 12 ms
25 192.168.1.1 10 ms 5 ms 3 ms
26 * 192.168.1.2 5 ms 4 ms
27 192.168.1.1 8 ms 3 ms 4 ms
28 192.168.1.2 9 ms * 7 ms
29 192.168.1.1 8 ms 4 ms 6 ms
30 192.168.1.2 21 ms 7 ms *
31 192.168.1.1 19 ms 17 ms 9 ms
32 192.168.1.2 8 ms 8 ms 6 ms
33 * 192.168.1.1 13 ms 12 ms
34 192.168.1.2 5 ms 5 ms 5 ms
35 192.168.1.1 12 ms * 10 ms
36 192.168.1.2 5 ms 9 ms 9 ms
37 192.168.1.1 14 ms 17 ms *
38 192.168.1.2 16 ms 15 ms 8 ms
39 192.168.1.1 6 ms 5 ms 5 ms
40 * 192.168.1.2 9 ms 6 ms
41 192.168.1.1 6 ms 12 ms 9 ms
42 192.168.1.2 5 ms * 23 ms
43 192.168.1.1 32 ms 14 ms 19 ms
44 192.168.1.2 22 ms 13 ms *
45 192.168.1.1 22 ms 18 ms 9 ms
46 192.168.1.2 4 ms 4 ms 5 ms
47 * 192.168.1.1 19 ms 20 ms
48 192.168.1.2 13 ms 14 ms 7 ms

```

```

49 192.168.1.1 6 ms * 32 ms
50 192.168.1.2 16 ms 11 ms 7 ms
51 192.168.1.1 9 ms 6 ms *
52 192.168.1.2 18 ms 22 ms 10 ms
53 192.168.1.1 10 ms 11 ms 11 ms
54 * 192.168.1.2 7 ms 5 ms
55 192.168.1.1 6 ms 5 ms 6 ms
56 192.168.1.2 6 ms * 22 ms
57 192.168.1.1 10 ms 7 ms 6 ms
58 192.168.1.2 6 ms 6 ms *
59 192.168.1.1 8 ms 6 ms 6 ms
60 192.168.1.2 6 ms 6 ms 6 ms
61 * 192.168.1.1 37 ms 18 ms
62 192.168.1.2 11 ms 13 ms 15 ms
63 192.168.1.1 6 ms * 31 ms
64 192.168.1.2 20 ms 22 ms 22 ms

```

## 7. Изучение IP-фрагментации

Написать, на каких узлах и как изменяли MTU.

Изменение MTU

Изменение MTU

Какие команды давали для тестирования и где.

```

r1:~# ip l set dev eth1 mtu 600
r3:~# ip l set dev eth1 mtu 600
ws3:~# echo 1 > /proc/sys/net/ipv4/ip_no_pmtu_disc

ws1:~# ping -c 1 192.168.3.3 -s 1000

```

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

Здесь вывод **tcpdump**, мак адреса не нужны.

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

```

r1:~# tcpdump -v -i eth1
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
11:23:59.968764 IP (tos 0x0, ttl 63, id 28941, offset 0, flags [+], proto ICMP (1), length 596) 192.168.1.3 > 192.168.3.3: icmp: 192.168.1.3 > 192.168.3.3
11:23:59.968789 IP (tos 0x0, ttl 63, id 28941, offset 576, flags [none], proto ICMP (1), length 452) 192.168.1.3 > 192.168.3.3: icmp: 192.168.1.3 > 192.168.3.3
11:24:04.963943 arp who-has 192.168.3.3 tell 192.168.3.2

```

Вывод **tcpdump** на узле получателя.

```

ws3:~# tcpdump -tnv -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP (tos 0x0, ttl 63, id 28941, offset 0, flags [+], proto ICMP (1), length 596) 192.168.1.3 > 192.168.3.3: icmp: 192.168.1.3 > 192.168.3.3
IP (tos 0x0, ttl 63, id 28941, offset 576, flags [none], proto ICMP (1), length 452) 192.168.1.3 > 192.168.3.3: icmp: 192.168.1.3 > 192.168.3.3
IP (tos 0x0, ttl 64, id 2077, offset 0, flags [+], proto ICMP (1), length 596) 192.168.3.3 > 192.168.1.3: icmp: 192.168.3.3 > 192.168.1.3
IP (tos 0x0, ttl 64, id 2077, offset 576, flags [none], proto ICMP (1), length 452) 192.168.3.3 > 192.168.1.3: icmp: 192.168.3.3 > 192.168.1.3

```



## 8. Отсутствие сети

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

```
ws1:~# ping 192.168.4.1 -c 1
PING 192.168.4.1 (192.168.4.1) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Destination Net Unreachable

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

r1:~# tcpdump icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
11:28:20.491868 IP 192.168.1.3 > 192.168.4.1: ICMP echo request, id 14594, seq 1, length 64
11:28:20.491922 IP 192.168.1.1 > 192.168.1.3: ICMP net 192.168.4.1 unreachable, length 92
```

## 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 192.168.2.5 -c 1
PING 192.168.2.5 (192.168.2.5) 56(84) bytes of data.
From 192.168.1.2 icmp_seq=1 Destination Host Unreachable

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

r2:~# tcpdump -v
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
11:34:00.430806 arp who-has 192.168.1.1 tell 192.168.1.3
11:34:00.440681 arp who-has 192.168.1.2 tell 192.168.1.1
11:34:00.440744 arp reply 192.168.1.2 is-at 3a:40:ee:31:9e:cd (oui Unknown)
11:34:00.440943 IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.1.2 > 192.168.1.3: ICMP echo request, id 14594, seq 1, length 64
11:34:03.493802 arp who-has 192.168.1.3 tell 192.168.1.2
11:34:03.494968 arp reply 192.168.1.3 is-at a6:f9:52:b6:1e:69 (oui Unknown)
11:34:03.494985 IP (tos 0xc0, ttl 64, id 22017, offset 0, flags [none], proto ICMP (1), length 84) 192.168.1.3 > 192.168.1.2: ICMP echo reply, id 14594, seq 1, length 64
11:34:03.494985 IP (tos 0xc0, ttl 62, id 0, offset 0, flags [DF], proto ICMP (1), length 84) 192.168.1.3 > 192.168.1.2: ICMP net 192.168.2.5 unreachable, length 92
```