

Отчёт по лабораторной работе «Динамическая IP-маршрутизация»

Binh D. Nguyen

1 ноября 2019 г.

Содержание

1. Настройка сети	1
1.1. Топология сети	1
1.2. Назначение IP-адресов	1
1.3. Настройка протокола RIP	3
2. Проверка настройки протокола RIP	4
3. Расщепленный горизонт и испорченные обратные обновления	6
4. Имитация устранимой поломки в сети	7
5. Имитация неустранимой поломки в сети	8

1. Настройка сети

1.1. Топология сети

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

Перечень узлов, на которых используется динамическая IP-маршрутизация: все кроме ws1 и ws2

1.2. Назначение IP-адресов

Ниже приведён файл сетевой настройки маршрутизатора r1.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.3.31
netmask 255.255.255.0
```

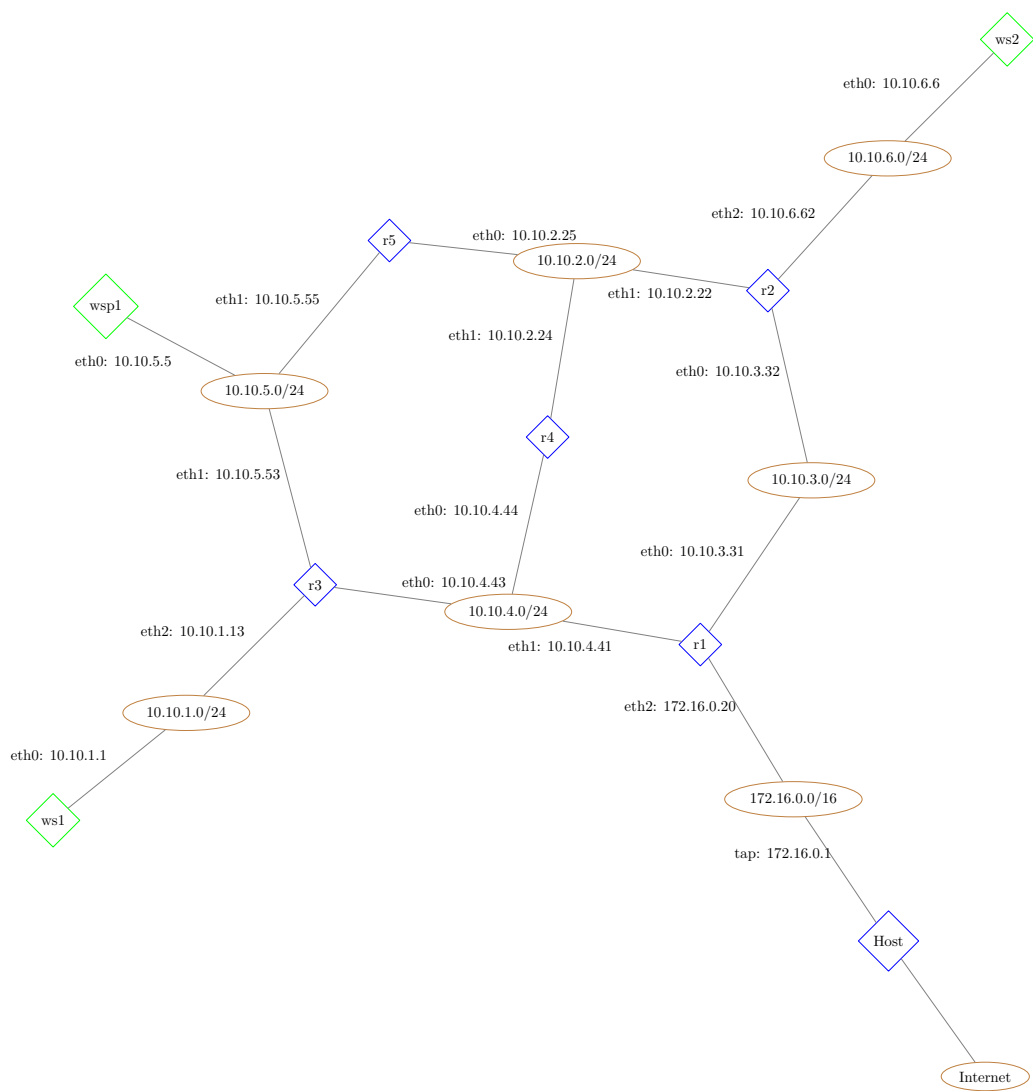


Рис. 1. Топология сети

```
auto eth1
iface eth1 inet static
address 10.10.4.41
netmask 255.255.255.0
```

Ниже приведён файл сетевой настройки рабочей станции ws1.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.1.1
netmask 255.255.255.0
gateway 10.10.1.13
```

1.3. Настройка протокола RIP

Ниже приведен файл /etc/quagga/ripd.conf маршрутизатора r1.

```
! Этот настройки, касающиеся протокола RIP.
router rip

! Раскомментируйте ниже все интерфейсы, подключённые
! к сетям с другими маршрутизаторами.
network eth0
network eth1
! network eth2

! Уменьшаем значения всех таймеров для ускорения опытов.
! Рассылка: 10 сек., устаревание: 60 сек., сборка мусора: 120 сек.
timers basic 10 60 120

! Следующие две строки заставляют маршрутизатор
! добавлять в сообщения протокола RIP все известные ему маршруты.
redistribute kernel
! redistribute connected

! Это имя файла журнала службы RIP.
! Его содержимое можно изучить в случае неполадок
log file /var/log/quagga/ripd.log
```

Ниже приведен файл /etc/quagga/ripd.conf рабочей станции, связанной с несколькими маршрутизаторами wsr1.

```
! Этот настройки, касающиеся протокола RIP.
router rip

! Раскомментируйте ниже все интерфейсы, подключённые
! к сетям с другими маршрутизаторами.
```

```

network eth0
! network eth1
! network eth2

! Уменьшаем значения всех таймеров для ускорения опытов.
! Рассылка: 10 сек., устаревание: 60 сек., сборка мусора: 120 сек.
timers basic 10 60 120

! Следующие две строчки заставляют маршрутизатор
! добавлять в сообщения протокола RIP все известные ему маршруты.
redistribute kernel
redistribute connected

! Это имя файла журнала службы RIP.
! Его содержимое можно изучить в случае неполадок
log file /var/log/quagga/ripd.log

```

2. Проверка настройки протокола RIP

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

```

ws2:~# traceroute 10.10.1.1
traceroute to 10.10.1.1 (10.10.1.1), 64 hops max, 40 byte packets
 1  10.10.6.62 (10.10.6.62)  8 ms  2 ms  1 ms
 2  10.10.2.24 (10.10.2.24)  12 ms  2 ms  2 ms
 3  10.10.5.53 (10.10.5.53)  35 ms  4 ms  5 ms
 4  10.10.1.1 (10.10.1.1)  14 ms  2 ms  3 ms

```

Вывод **traceroute** от узла ws2 до внешнего IP 8.8.8.8.

```

ws2:~# ping 8.8.8.8 -c 1
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=125 time=34.2 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 34.276/34.276/34.276/0.000 ms
ws2:~# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 64 hops max, 40 byte packets
 1  10.10.6.62 (10.10.6.62)  1 ms  2 ms  2 ms
 2  10.10.3.31 (10.10.3.31)  2 ms  1 ms  1 ms
 3  172.16.0.1 (172.16.0.1)  1 ms  3 ms  2 ms
 4  192.168.37.2 (192.168.37.2)  3 ms  2 ms  2 ms
 5  * * *
 6  * * *
 7  * 10.10.6.62 (10.10.6.62)  1 ms !N  1 ms !N

```

Вывод сообщения RIP.

```

r2:~#tcpdump -tnv udp
...
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 112) 10.10.3.31.520 > 22
    RIPv2, Response, length: 84, routes: 4
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4:      10.10.2.0/24, tag 0x0000, metric: 2, next-hop: self[|rip]
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 132) 10.10.3.32.520 > 22
    RIPv2, Response, length: 104, routes: 5
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 3, next-hop: self
        AFI: IPv4:      10.10.2.0/24, tag 0x0000, metric: 1, next-hop: self[|rip]
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 112) 10.10.3.31.520 > 22
    RIPv2, Response, length: 84, routes: 4
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4:      10.10.2.0/24, tag 0x0000, metric: 2, next-hop: self[|rip]
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 132) 10.10.3.32.520 > 22
    RIPv2, Response, length: 104, routes: 5
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 3, next-hop: self
        AFI: IPv4:      10.10.2.0/24, tag 0x0000, metric: 1, next-hop: self[|rip]
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 132) 10.10.3.32.520 > 22
    RIPv2, Response, length: 104, routes: 5
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 3, next-hop: self
        AFI: IPv4:      10.10.2.0/24, tag 0x0000, metric: 1, next-hop: self[|rip]
...

```

Вывод таблицы RIP.

```

r1:~# vtysh

Hello, this is Quagga (version 0.99.10).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r1# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network      Next Hop      Metric From      Tag Time
R(n) 10.10.1.0/24  10.10.4.43      2 10.10.4.43      0 00:50
R(n) 10.10.2.0/24  10.10.4.44      2 10.10.4.44      0 00:59
C(i) 10.10.3.0/24  0.0.0.0         1 self            0
C(i) 10.10.4.0/24  0.0.0.0         1 self            0
R(n) 10.10.5.0/24  10.10.4.43      2 10.10.4.43      0 00:50
R(n) 10.10.6.0/24  10.10.3.32      2 10.10.3.32      0 00:59
r1# exit

```

Вывод таблицы маршрутизации.

```

r1:~# ip r
10.10.6.0/24 via 10.10.3.32 dev eth0 proto zebra metric 2

```

```

10.10.4.0/24 dev eth1  proto kernel  scope link  src 10.10.4.41
10.10.5.0/24 via 10.10.4.43 dev eth1  proto zebra  metric 2
10.10.2.0/24 via 10.10.4.44 dev eth1  proto zebra  metric 2
10.10.3.0/24 dev eth0  proto kernel  scope link  src 10.10.3.31
10.10.1.0/24 via 10.10.4.43 dev eth1  proto zebra  metric 2
172.16.0.0/16 dev eth2  proto kernel  scope link  src 172.16.0.20
default via 172.16.0.1 dev eth2

```

3. Расщепленный горизонт и испорченные обратные обновления

Поместить сюда вывод сообщения одного и того же маршрутизатор с включенным расщ. горизонтом, с включенными испорченными обновлениями, с отключённым расщ. гор.

```

# r5/etc/quagga/ripd.conf
interface eth1
ip rip split-horizon poisoned-reverse

# bash
r5:~# tcpdump -nvt -i eth1
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 172) 10.10.5.55.520 > 22
    RIPv2, Response, length: 144, routes: 7
        AFI: IPv4:      0.0.0.0/0 , tag 0x0000, metric: 3, next-hop: self
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 16, next-hop: 10.10.5.53[|rip]
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 152) 10.10.5.53.520 > 22
    RIPv2, Response, length: 124, routes: 6
        AFI: IPv4:      0.0.0.0/0 , tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 1, next-hop: self[|rip]
...

# r5/etc/quagga/ripd.conf
interface eth1
no ip rip split-horizon

# bash
r5:~# tcpdump -nvt -i eth1
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 152) 10.10.5.53.520 > 22
    RIPv2, Response, length: 124, routes: 6
        AFI: IPv4:      0.0.0.0/0 , tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 1, next-hop: self[|rip]
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 152) 10.10.5.53.520 > 22
    RIPv2, Response, length: 124, routes: 6
        AFI: IPv4:      0.0.0.0/0 , tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4:      10.10.1.0/24, tag 0x0000, metric: 1, next-hop: self[|rip]
...

```

Вернуть настройки в исходное состояние (включенный без испорченных).

4. Имитация устранимой поломки в сети

Маршрутизатор r5 был выключен.

Вывод таблицы RIP непосредственно перед истечением таймера устаревания (на r2 - маршрутизаторе-соседе отключенного).

```
r2# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop          Metric From          Tag Time
R(n) 0.0.0.0/0         10.10.3.31          2 10.10.3.31         0 00:54
R(n) 10.10.1.0/24      10.10.2.24          3 10.10.2.24         0 00:56
C(i) 10.10.2.0/24      0.0.0.0             1 self               0
C(i) 10.10.3.0/24      0.0.0.0             1 self               0
R(n) 10.10.4.0/24      10.10.2.24          2 10.10.2.24         0 00:56
R(n) 10.10.5.0/24      10.10.2.24          3 10.10.2.24         0 00:56
C(i) 10.10.6.0/24      0.0.0.0             1 self               0

ws2:~# ping -c 1 10.10.5.5
PING 10.10.5.5 (10.10.5.5) 56(84) bytes of data.
64 bytes from 10.10.5.5: icmp_seq=1 ttl=62 time=1.03 ms

--- 10.10.5.5 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 21.620/21.620/21.620/0.000 ms

ws2:~# traceroute 10.10.5.5
traceroute to 10.10.5.5 (10.10.5.5), 64 hops max, 40 byte packets
 1  10.10.6.62 (10.10.6.62)  7 ms  2 ms  4 ms
 2  10.10.2.25 (10.10.2.25)  2 ms  4 ms  3 ms
 3  10.10.5.5 (10.10.5.5)  12 ms  1 ms  2 ms
```

Перестроенная таблица на этом же маршрутизаторе

```
r5:~#halt

r2:~# uhl_net_start_xmit: failed(-111)
uhl_net_start_xmit: failed(-111)
uhl_net_start_xmit: failed(-111)
uhl_net_start_xmit: failed(-111)
uhl_net_start_xmit: failed(-111)

r2# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface
```

	Network	Next Hop	Metric From	Tag	Time
R(n)	0.0.0.0/0	10.10.3.31	2 10.10.3.31	0	00:53
R(n)	10.10.1.0/24	10.10.3.31	3 10.10.3.31	0	00:53
C(i)	10.10.2.0/24	0.0.0.0	1 self	0	
C(i)	10.10.3.0/24	0.0.0.0	1 self	0	
R(n)	10.10.4.0/24	10.10.3.31	2 10.10.3.31	0	00:53
R(n)	10.10.5.0/24	10.10.3.31	3 10.10.3.31	0	00:53
C(i)	10.10.6.0/24	0.0.0.0	1 self	0	

Вывод **traceroute** от узла ws2 до ws1 после того, как служба RIP перестроила таблицы маршрутизации.

```
ws2:~# ping -c 1 10.10.5.5
PING 10.10.5.5 (10.10.5.5) 56(84) bytes of data.
64 bytes from 10.10.5.5: icmp_seq=1 ttl=61 time=34.9 ms

--- 10.10.5.5 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 34.923/34.923/34.923/0.000 ms

ws2:~# traceroute 10.10.5.5
traceroute to 10.10.5.5 (10.10.5.5), 64 hops max, 40 byte packets
 1  10.10.6.62 (10.10.6.62)  1 ms  3 ms  1 ms
 2  10.10.3.31 (10.10.3.31)  1 ms  3 ms  1 ms
 3  10.10.4.43 (10.10.4.43)  2 ms  3 ms  1 ms
 4  10.10.5.5 (10.10.5.5)   2 ms  1 ms  2 ms
```

5. Имитация неустраняемой поломки в сети

Какой маршрутизатор выключили? (Теперь у нас нет связанной сети)

Далее поместить таблицы протокола RIP, где видна 16-ая метрика, и сообщения протокола RIP с 16-ой метрикой.