

Отчёт по лабораторной работе «Динамическая IP-маршрутизация»

Trung Luong

13 октября 2020 г.

Содержание

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

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

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 10.10.2.2
netmask 255.255.255.0

auto eth1
```

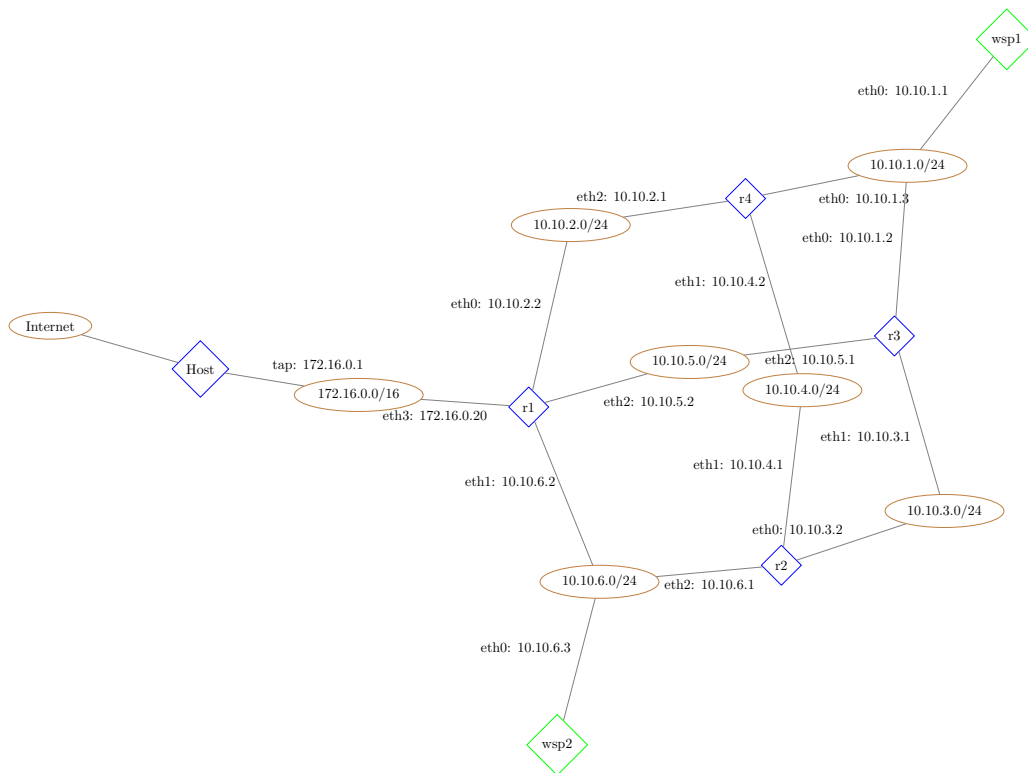


Рис. 1. Топология сети

```

iface eth1 inet static
address 10.10.6.2
netmask 255.255.255.0

auto eth2
iface eth2 inet static
address 10.10.5.2
netmask 255.255.255.0

```

Ниже приведён файл сетевой настройки рабочей станции **wsp1**.

```

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.1.1
netmask 255.255.255.0

```

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 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

```

Ниже приведен файл `/etc/quagga/ripd.conf` рабочей станции, связанной с несколькими маршрутизаторами **wsp1**.

```

! 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 **wsp2** до **wsp1** при нормальной работе сети.

```

wsp2:~# traceroute 10.10.1.1
traceroute to 10.10.1.1 (10.10.1.1), 64 hops max, 40 byte packets

```

```

1 10.10.6.1 (10.10.6.1) 1 ms 0 ms 1 ms
2 10.10.3.1 (10.10.3.1) 0 ms 0 ms 0 ms
3 10.10.1.1 (10.10.1.1) 1 ms 1 ms 0 ms

```

Вывод **traceroute** от узла такого-то до внешнего IP (195.19.38.2 сгодится).

Сюда нужно поместить вывод traceroute.

Вывод сообщения RIP.

```

r2:~# tcpdump -tnv udp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 152) 10.10.3.2.520 > 224.0.0.0.520:
    RIPv2, Response, length: 124, routes: 6
        AFI: IPv4:      0.0.0.0/0 , 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) 10.10.3.1.520 > 224.0.0.0.520:
    RIPv2, Response, length: 104, routes: 5
        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) 10.10.3.2.520 > 224.0.0.0.520:
    RIPv2, Response, length: 124, routes: 6
        AFI: IPv4:      0.0.0.0/0 , 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 152) 10.10.3.2.520 > 224.0.0.0.520:
    RIPv2, Response, length: 124, routes: 6
        AFI: IPv4:      0.0.0.0/0 , 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) 10.10.3.1.520 > 224.0.0.0.520:
    RIPv2, Response, length: 104, routes: 5
        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]
...

```

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
K(r) 0.0.0.0/0         172.16.0.1         1 self              0
R(n) 10.10.1.0/24      10.10.5.1          2 10.10.5.1          0 00:56
C(i) 10.10.2.0/24      0.0.0.0            1 self              0

```

```
R(n) 10.10.3.0/24      10.10.5.1      2 10.10.5.1      0 00:56
R(n) 10.10.4.0/24      10.10.2.1      2 10.10.2.1      0 00:58
C(i) 10.10.5.0/24      0.0.0.0        1 self          0
C(i) 10.10.6.0/24      0.0.0.0        1 self          0
C(r) 172.16.0.0/16     0.0.0.0        1 self (connected:1) 0
```

Вывод таблицы маршрутизации.

```
r1:~# ip r
10.10.6.0/24 dev eth1  proto kernel  scope link  src 10.10.6.2
10.10.4.0/24 via 10.10.2.1 dev eth0  proto zebra  metric 2
10.10.5.0/24 dev eth2  proto kernel  scope link  src 10.10.5.2
10.10.2.0/24 dev eth0  proto kernel  scope link  src 10.10.2.2
10.10.3.0/24 via 10.10.5.1 dev eth2  proto zebra  metric 2
10.10.1.0/24 via 10.10.5.1 dev eth2  proto zebra  metric 2
172.16.0.0/16 dev eth3  proto kernel  scope link  src 172.16.0.20
default via 172.16.0.1 dev eth3
```

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.

```
# r4/etc/quagga/ripd.conf
interface eth0
ip rip split-horizon poisoned-reverse

# bash
r4:~# tcpdump -nvt -i eth0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 172) 10.10.1.2.520 > 224.0.0.0.520:
    RIPv2, Response, length: 144, routes: 7
        AFI: IPv4:      0.0.0.0/0 , 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) 10.10.1.3.520 > 224.0.0.0.520:
    RIPv2, Response, length: 104, routes: 5
        AFI: IPv4:      0.0.0.0/0 , tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4:      10.10.2.0/24, tag 0x0000, metric: 1, next-hop: self[|rip]
...

# r4/etc/quagga/ripd.conf
interface eth0
no ip rip split-horizon

# bash
r4:~# tcpdump -nvt -i eth0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

```
| IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 172) 10.10.1.3.520 > 224
    RIPv2, Response, length: 144, routes: 7
        AFI: IPv4:      0.0.0.0/0 , tag 0x0000, metric: 2, 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 172) 10.10.1.2.520 > 224
    RIPv2, Response, length: 144, routes: 7
        AFI: IPv4:      0.0.0.0/0 , tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4:      10.10.2.0/24, tag 0x0000, metric: 2, next-hop: self[|rip]
| ...
```

Вернуть настройки в исходное состояние (включенный без испорченных).

4. Имитация устранимой поломки в сети

Какой маршрутизатор выключили?

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

| Таблица RIP

Перестроенная таблица на этом же маршрутизаторе

| Таблица RIP

Вывод **traceroute** от узла такого-то до такого-то после того, как служба RIP перестроила таблицы маршрутизации.

| Сюда нужно поместить вывод traceroute после "поломки".

5. Имитация неустранимой 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.