```
INR Lab Assignment:
RIPng∗
```

Preparation

Task 1. Take a look at the documentation available at http://www.nongnu.org/quagga/docs/quagga.html and explain what zebra is.

Answer:

As stated in the documentation, Zebra is “an IP routing manager. It provides kernel routing table updates, interface lookups, and redistribution of routes between different routing protocols.”

A simple RIPng network

Task 2. In the examples folder of pogo you will find a config file for the simple RIPng network of Figure 1. Edit the ripng.cfg so the pass quaggadir variables contain the absolute paths on the host of the routeri directories. Start the network. The RIPng protocol should start automatically.

Answer:

The routers directories are located in /pogo/configurations/ripng folder:

```
root@Guest-02:~/pogo/configurations/ripng# ls
ripng.cfg  router1  router2  router3
```

So I edited the following and I added the full path:

```
pass_quaggadir="/root/pogo/configurations/ripng/router1/"
pass_quaggadir="/root/pogo/configurations/ripng/router2/"
pass_quaggadir="/root/pogo/configurations/ripng/router3/"
```

Now, Let's create the configuration then start it:

```
root@Guest-02:~/pogo# python pogo.py create configurations/ripng/ripng.cfg
INFO: Creating ogo-ovs0 switch
ovs-vsctl: cannot create a bridge named ogo-ovs0 because a bridge named ogo-
ovs0 already exists
```

```
ovs-vsctl: multiple rows in Mirror match "ogo-ovs0mirror"
INFO: Creating ogo-ovs1 switch
ovs-vsctl: cannot create a bridge named ogo-ovs1 because a bridge named ogo-
ovs1 already exists
ovs-vsctl: multiple rows in Mirror match "ogo-ovs1mirror"
INFO: Creating host router1. This will take some time.
INFO: Creating host router2. This will take some time.
INFO: Creating host router3. This will take some time.

root@Guest-02:~/pogo# python pogo.py start configurations/ripng/ripng.cfg
INFO: Starting sniffer on ogo-ovs0 switch
INFO: Starting sniffer on ogo-ovs1 switch
INFO: Starting host router1.
INFO: Starting host router2.
INFO: Starting host router3.
There are screens on:
    7951.router3    (11/19/17 16:52:51)    (Detached)
    7713.router2    (11/19/17 16:52:48)    (Detached)
    7346.router1    (11/19/17 16:52:45)    (Detached)
    26264.bridge3    (11/14/17 12:20:31)    (Detached)
    25833.bridge2    (11/14/17 12:20:27)    (Detached)
    25384.bridge1    (11/14/17 12:20:24)    (Detached)
    24749.host2    (11/14/17 12:20:20)    (Detached)
    24280.host1    (11/14/17 12:20:18)    (Detached)
8 Sockets in /run/screen/S-root.
```

host1, host2, bridge1, bridge2, and bridge3 are from previous assignments.

Task 3. Test the connectivity between router1 and router3 using the traceroute6 utility.

Answer:

I will use traceroute6 to the IP address of router3 which is “2001:db8:0:f102::2/64”:

```
root@Guest-02:~/pogo# screen -r router1

root@router1:~# traceroute6 2001:db8:0:f102::2
traceroute to 2001:db8:0:f102::2 (2001:db8:0:f102::2) from
2001:db8:0:f101::2, 30 hops max, 24 byte packets
 1  2001:db8:0:f101::1 (2001:db8:0:f101::1)  1.41 ms  0.134 ms  0.096 ms
 2  2001:db8:0:f102::2 (2001:db8:0:f102::2)  0.863 ms  0.122 ms  0.279 ms
```

Task 4. Examine the config files for both the network script and quagga and then on one of the routers connect to (via vtysh) and explore the ripng and zebra daemons. Very briefly explain what the following commands do: show ?, show run, config terminal, router ?, network, exit, write memory . Hint: some commands only exist in certain contexts

Answer:

I will use router1 and connect to the VTY (is integrated shell of Quagga software.) shell via vtysh.

show ? : it will display the options.

```
root@router1:~# vtysh

Hello, this is Quagga (version 0.99.24.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router1#
  clear         Reset functions
  configure     Configuration from vty interface
  copy          Copy from one file to another
  debug         Debugging functions (see also 'undebug')
  disable       Turn off privileged mode command
  end           End current mode and change to enable mode
  exit          Exit current mode and down to previous mode
  list          Print command list
  no            Negate a command or set its defaults
  ping          Send echo messages
  quit          Exit current mode and down to previous mode
  show          Show running system information
  ssh           Open an ssh connection
  start-shell   Start UNIX shell
  telnet        Open a telnet connection
  terminal      Set terminal line parameters
  test          Test
  traceroute    Trace route to destination
  undebug       Disable debugging functions (see also 'debug')
  write         Write running configuration to memory, network, or terminal
```

show run: displays the current configuration.

```
router1# show run
Building configuration...

Current configuration:
!
hostname Router
hostname router1
log stdout
!
debug rip events
debug rip packet
debug ripng events
debug ripng packet
!
password 1
enable password 1
!
interface eth12
 ipv6 nd suppress-ra
 no link-detect
!
interface lo
 no link-detect
!
router rip
 version 2
 network eth12
!
router ripng
 network ::/0
 redistribute connected
!
ip forwarding
ipv6 forwarding
!
line vty
!
end
```

config terminal: used to configure the environment by entering the config mode.

```
router1# config terminal
router1(config)#
```

router ? : List the options to the router command and its descriptions.

```
router1(config)# router
  babel  Babel
  bgp    BGP information
  isis   ISO IS-IS
  ospf   Start OSPF configuration
  ospf6  Open Shortest Path First (OSPF) for IPv6
  rip    RIP
  ripng  RIPng
```

exit: Exits the shell and if we are inside configuration also exit.

```
router1(config)# exit
router1#

router1# exit
```

```
root@router1:~#
```

Can't find the network command. However I found its uses. “Set the RIP enable interface by network. The interfaces which have addresses matching with network are enabled.”

write memory: saves the current configuration to the startup configuration.

```
router1# write memory
Building Configuration...
Can't open configuration file /etc/quagga/zebra.conf.FFOzn7.
Can't open configuration file /etc/quagga/ripd.conf.v5ZSt7.
Can't open configuration file /etc/quagga/ripngd.conf.uQqYq7.
[OK]
```

Source:

1- http://www.nongnu.org/quagga/docs/quagga.html#Starting-and-Stopping-ripd

Task 5. Connect to the RIPng process on router2 and display the RIPng routes and protocol status.

Answer:

On router2 to display protocol status and RIPng routes:

```
router2# show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 24 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:
  Default version control: send version 2, receive version 2
    Interface        Send  Recv   Key-chain
    eth12            2     2
    eth23            2     2
  Routing for Networks:
    eth12
    eth23
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
  Distance: (default is 120)
```

```
router2# 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
C(i) 10.0.0.0/8         0.0.0.0               1 self              0
C(i) 192.168.0.0/24     0.0.0.0               1 self              0


router2# show ipv6 rip status
Routing Protocol is "RIPng"
  Sending updates every 30 seconds with +/-50%, next due in 12 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:    connected
  Default version control: send version 1, receive version 1
    Interface        Send  Recv
    eth12            1     1
    eth23            1     1
  Routing for Networks:
    ::/0
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    fe80::216:3eff:fe2a:330e
                        0          0        120      00:00:07
    fe80::216:3eff:fe8e:98dd
                        0          0        120      00:00:15

router2# show ipv6 rip
Codes: R - RIPng, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface, (a/S) - aggregated/Suppressed

   Network      Next Hop                      Via     Metric Tag Time
C(i) 2001:db8:0:f101::/64
                  ::                          self       1    0
C(i) 2001:db8:0:f102::/64
                  ::                          self       1    0
```

Task 6. Start a ping6 from router1 to router3. Remove the global IPv6 address configured on router2’s eth1. What do you observe after max. 30 seconds? Why is the ping still working?

Answer:

I started the ping, and I removed the global IPv6 address on router2 on eth23 (eth1):

```
root@router2:~# ifconfig eth23 inet6 del 2001:db8:0:f102::1/64
root@router2:~# ifconfig
eth12     Link encap:Ethernet  HWaddr 00:16:3e:b1:ef:11
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: 2001:db8:0:f101::1/64 Scope:Global
          inet6 addr: fe80::216:3eff:feb1:ef11/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2388 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4198 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:203160 (203.1 KB)  TX bytes:362816 (362.8 KB)

eth23     Link encap:Ethernet  HWaddr 00:16:3e:41:03:d5
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::216:3eff:fe41:3d5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2408 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:204528 (204.5 KB)  TX bytes:362784 (362.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:168 errors:0 dropped:0 overruns:0 frame:0
          TX packets:168 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13408 (13.4 KB)  TX bytes:13408 (13.4 KB)
```

Ping output:

```
root@router1:~# ping6 2001:db8:0:f102::2
PING 2001:db8:0:f102::2(2001:db8:0:f102::2) 56 data bytes
64 bytes from 2001:db8:0:f102::2: icmp_seq=1 ttl=63 time=1.01 ms
          TX packets:4129 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:196804 (196.8 KB)  TX bytes:355474 (355.4 KB)

64 bytes from 2001:db8:0:f102::2: icmp_seq=17 ttl=63 time=0.160 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=18 ttl=63 time=0.157 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=19 ttl=63 time=0.160 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=20 ttl=63 time=0.159 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=21 ttl=63 time=0.162 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=22 ttl=63 time=0.177 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=23 ttl=63 time=0.162 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=24 ttl=63 time=0.160 ms
```

```
64 bytes from 2001:db8:0:f102::2: icmp_seq=25 ttl=63 time=0.167 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=26 ttl=63 time=0.166 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=27 ttl=63 time=0.165 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=28 ttl=63 time=0.165 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=29 ttl=63 time=0.159 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=30 ttl=63 time=0.136 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=31 ttl=63 time=0.156 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=32 ttl=63 time=0.147 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=33 ttl=63 time=0.170 ms
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
64 bytes from 2001:db8:0:f102::2: icmp_seq=39 ttl=63 time=0.149 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=40 ttl=63 time=0.156 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=41 ttl=63 time=0.158 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=42 ttl=63 time=0.160 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=43 ttl=63 time=0.160 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=44 ttl=63 time=0.159 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=45 ttl=63 time=0.153 ms
64 bytes from 2001:db8:0:f102::2: icmp_seq=46 ttl=63 time=0.139 ms
^C
--- 2001:db8:0:f102::2 ping statistics ---
46 packets transmitted, 41 received, 10% packet loss, time 46041ms
rtt min/avg/max/mdev = 0.134/0.179/1.011/0.132 ms
```

From the previous output we can't derive anything, It stopped working for few seconds. that's why I will use show ipv6 route:

```
router2# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
C>* 2001:db8:0:f101::/64 is directly connected, eth12
R>* 2001:db8:0:f102::/64 [120/2] via fe80::216:3eff:fe8e:98dd, eth23,
00:00:58
C * fe80::/64 is directly connected, eth23
C>* fe80::/64 is directly connected, eth12
```

As we see in above output, the interface that I deleted “eth23 which is eth01” on router2 became reachable via the link local address with RIPng.

```
R>* 2001:db8:0:f102::/64 [120/2] via fe80::216:3eff:fe8e:98dd, eth23,
00:00:58
```

A more complex RIPng network

Task 7. Create the config file required to start the network depicted in Figure 2. Each routeri has two or three network interfaces, one connected to a simple host via LAN Li (host not shown in the figure) and the rest connected to other routers (Li j...). You will use the IPv6 blocks defined in the previous labs. The addressing is as follows:

- For L1 : The : y01 : /64 block
- For L2 : The : y02 : /64 block
- For L3 : The : y03 : /64 block
- For L4 : The : y04 : /64 block
- For L5 : The : y05 : /64 block
- For L14 : The 8th /64 block
- For L35 : The 16th /64 block
- For L45 : The 32nd /64 block
- For L123 : The 42nd /64 block

As a convention, each router routeri will use the ::i address on the network segment. (e.g., assuming that the IP block for L35 is 2001:0db8::/64, router5 will use 2001:0db8::5). Hint: you can arbitrarily number eth devices!. The end hosts do not need to be created, but you do need to create the network segments. Show the config file on your log.

Answer:

First, I created a quagga directories for routers 4 + 5:

```
root@Guest-02:~/pogo/configurations/ripng# ll
total 36
drwxr-xr-x 7 root root 4096 Nov 20 11:00 ./
drwxr-xr-x 4 root root 4096 Nov 13 20:46 ../
-rwxr-xr-x 1 root root  909 Nov 19 16:48 oripng.cfg*
-rwxr-xr-x 1 root root  990 Nov 19 16:48 ripng.cfg*
drwxr-xr-x 2 root root 4096 Nov 13 20:46 router1/
drwxr-xr-x 2 root root 4096 Nov 13 20:46 router2/
drwxr-xr-x 2 root root 4096 Nov 13 20:46 router3/
drwxr-xr-x 2 root root 4096 Nov 13 20:46 router4/
drwxr-xr-x 2 root root 4096 Nov 13 20:46 router5/
```

It was quite complicated procedure, I followed the steps and used hex converter etc.. .

The IPv6 address from previous lab was 2001:0db8:0x00:0y00::1 with the values from the previous lab, it became:

```
X = 2
Y = 14 ( E in hex )

2001:0db8:0200:0e00::1
```

```
root@Guest-02:~/pogo/configurations/ripng# cat ripng.cfg
############################################################################
####
#GLOBAL CONFIG
############################################################################
####
[global]
session_path  = /tmp
tcpdump_path= /tmp
hubs = 9

############################################################################
####
#HOST CONFIG
############################################################################
####

#section name = device hostname
[router1]
role= router
home=/tmp
eth1 = 1,,2001:0db8:0200:0e01::1/64
eth123 = 6,,2001:0db8:0200:0e29::1/64
eth14 = 8,,2001:0db8:0200:0e07::1/64
pass_quaggadir="/root/pogo/configurations/ripng/router1/"

[router2]
role = router
home=/tmp
eth2 = 2,,2001:0db8:0200:0e02::2/64
eth123 = 6,,2001:0db8:0200:0e29::2/64
pass_quaggadir="/root/pogo/configurations/ripng/router2/"


[router3]
role = router
home=/tmp
eth3 = 3,,2001:0db8:0200:0e03::3/64
eth123 = 6,,2001:0db8:0200:0e29::3/64
eth35 = 7,,2001:0db8:0200:0e0f::3/64
pass_quaggadir="/root/pogo/configurations/ripng/router3/"


[router4]
role = router
home=/tmp
```

```
eth4 = 4,,2001:0db8:0200:0e04::4/64
eth14 = 8,,2001:0db8:0200:0e07::4/64
eth45 = 0,,2001:0db8:0200:0e1f::4/64
pass_quaggadir="/root/pogo/configurations/ripng/router4/"

[router5]
role = router
home=/tmp
eth5 = 5,,2001:0db8:0200:0e05::5/64
eth35 = 7,,2001:0db8:0200:0e0f::5/64
eth45 = 0,,2001:0db8:0200:0e1f::5/64
pass_quaggadir="/root/pogo/configurations/ripng/router5/"

[sniffer123]
role = sniffer
home=/tmp
eth0 = 6,,
mem=64M
```

Now, I create the configuration file, then start it (P.S I destroyed all the previous configurations and screens“):

```
root@Guest-02:~/pogo# python pogo.py create configurations/ripng/ripng.cfg
INFO: Creating ogo-ovs0 switch
INFO: Creating ogo-ovs1 switch
INFO: Creating ogo-ovs2 switch
INFO: Creating ogo-ovs3 switch
INFO: Creating ogo-ovs4 switch
INFO: Creating ogo-ovs5 switch
INFO: Creating ogo-ovs6 switch
INFO: Creating ogo-ovs7 switch
INFO: Creating ogo-ovs8 switch
WARNING: Sniffer is DEPRICATED!
INFO: Creating host router1. This will take some time.
INFO: Creating host router2. This will take some time.
INFO: Creating host router3. This will take some time.
INFO: Creating host router4. This will take some time.
INFO: Creating host router5. This will take some time.


root@Guest-02:~/pogo# python pogo.py start configurations/ripng/ripng.cfg
INFO: Starting sniffer on ogo-ovs0 switch
INFO: Starting sniffer on ogo-ovs1 switch
INFO: Starting sniffer on ogo-ovs2 switch
INFO: Starting sniffer on ogo-ovs3 switch
INFO: Starting sniffer on ogo-ovs4 switch
INFO: Starting sniffer on ogo-ovs5 switch
INFO: Starting sniffer on ogo-ovs6 switch
INFO: Starting sniffer on ogo-ovs7 switch
INFO: Starting sniffer on ogo-ovs8 switch
WARNING: Sniffer is DEPRICATED!
```

```
INFO: Starting host router1.
INFO: Starting host router2.
INFO: Starting host router3.
INFO: Starting host router4.
INFO: Starting host router5.
There are screens on:
    29121.router5    (11/20/17 11:40:20)    (Detached)
    28521.router4    (11/20/17 11:40:14)    (Detached)
    28185.router3    (11/20/17 11:40:11)    (Detached)
    27768.router2    (11/20/17 11:40:07)    (Detached)
    27387.router1    (11/20/17 11:40:04)    (Detached)
    19882.19447.router2    (11/20/17 10:42:45)    (Detached)
6 Sockets in /run/screen/S-root.
```

Task 8. Draw the diagram of the network including the IP address of each router interface.

Answer:

Task 9. For router1 display the following:

- the routing table
- the RIPng table
- the RIPng status

Are there any differences between the RIPng table and the routing table? (Hints: ip -6, vtysh, show)

Answer:

- **router1 routing table:**

```
root@router1:~# ip -6 route
2001:db8:200:e01::/64 dev eth1  proto kernel  metric 256  pref medium
2001:db8:200:e02::/64 via fe80::216:3eff:fe0d:ab86 dev eth123  proto zebra
metric 2  pref medium
2001:db8:200:e03::/64 via fe80::216:3eff:fee7:5118 dev eth123  proto zebra
metric 2  pref medium
2001:db8:200:e04::/64 via fe80::216:3eff:fe42:6980 dev eth14  proto zebra
metric 2  pref medium
2001:db8:200:e05::/64 via fe80::216:3eff:fe42:6980 dev eth14  proto zebra
metric 3  pref medium
2001:db8:200:e07::/64 dev eth14  proto kernel  metric 256  pref medium
2001:db8:200:e0f::/64 via fe80::216:3eff:fee7:5118 dev eth123  proto zebra
```

```
metric 2  pref medium
2001:db8:200:e1f::/64 via fe80::216:3eff:fe42:6980 dev eth14  proto zebra
metri



ic 2  pref medium
2001:db8:200:e29::/64 dev eth123  proto kernel  metric 256  pref medium
fe80::/64 dev eth1  proto kernel  metric 256  pref medium
fe80::/64 dev eth123  proto kernel  metric 256  pref medium
fe80::/64 dev eth14  proto kernel  metric 256  pref medium
```

- **router1 RIPng table:**

```
router1# show  ipv6 route  ripng
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* 2001:db8:200:e02::/64 [120/2] via fe80::216:3eff:fe0d:ab86, eth123,
00:03:41
R>* 2001:db8:200:e03::/64 [120/2] via fe80::216:3eff:fee7:5118, eth123,
00:03:34
R>* 2001:db8:200:e04::/64 [120/2] via fe80::216:3eff:fe42:6980, eth14,
00:03:30
R>* 2001:db8:200:e05::/64 [120/3] via fe80::216:3eff:fe42:6980, eth14,
00:03:29
R>* 2001:db8:200:e0f::/64 [120/2] via fe80::216:3eff:fee7:5118, eth123,
00:03:34
R>* 2001:db8:200:e1f::/64 [120/2] via fe80::216:3eff:fe42:6980, eth14,
00:03:30
```

- **router1 RIPng status:**

```
router1# show ipv6  ripng  status
Routing Protocol is "RIPng"
  Sending updates every 30 seconds with +/-50%, next due in 24 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:    connected
  Default version control: send version 1, receive version 1
    Interface        Send  Recv
    eth1             1     1
    eth14            1     1
    eth123           1     1
  Routing for Networks:
    ::/0
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    fe80::216:3eff:fe0d:ab86
                        0          0        120      00:00:20
    fe80::216:3eff:fee7:5118
                        0          0        120      00:00:01
    fe80::216:3eff:fe42:6980
                        0          0        120      00:00:20
```

The routing table is a

The routing table list of the best routes to a given network: Static is indicated with “the proto kernel metric”, and dynamic is indicated “proto zebra metric”, where metric indicates the costs of the route. However, in the RIPng table it doesn't show the static routes and the link local addresses, but it shows everything specifically related to RIPng protocol.

Source:

1- https://learningnetwork.cisco.com/thread/16204

Task 10. Configure router2 as a default gateway for all the other routers. Just adding static routes on each router is not allowed. Describe how you did this.

Answer:

We can use on router2 The default-information originate router configuration command. This instructs router to originate default information, by propagating the static default route in RIP updates.

```
router2# config terminal

router2(config)# router
  babel  Babel
  bgp    BGP information
  isis   ISO IS-IS
  ospf   Start OSPF configuration
  ospf6  Open Shortest Path First (OSPF) for IPv6
  rip    RIP
  ripng  RIPng
router2(config)# router ripng

router2(config-router)#
  aggregate-address    Set aggregate RIPng route announcement
  default-information  Default route information
  default-metric       Set a metric of redistribute routes
  end                  End current mode and change to enable mode
  exit                 Exit current mode and down to previous mode
  list                 Print command list
  network              RIPng enable on specified interface or network.
  no                   Negate a command or set its defaults
  offset-list          Modify RIPng metric
  passive-interface    Suppress routing updates on an interface
  quit                    Exit current mode and down to previous mode
  redistribute            Redistribute
  route                   Static route setup
  route-map               Route map set
  timers                  RIPng timers setup
router2(config-router)# default-information
  originate  Distribute default route
router2(config-router)# default-information originate
```

Now, Let's check on router1 (since in previous question we already have output) using “ip -6 route”:

```
default via fe80::216:3eff:fe0d:ab86 dev eth123  proto zebra  metric 2  pref
medium
```

As we see above, the default route is added to router1.

Source:

1- http://www.ciscopress.com/articles/article.asp?p=2180210&seqNum=10

Task 11. Start a ping from router3 to router1’s address on the L1 segment. Turn off the interface that connects router1 to L123. How much time does it take for router3 to reach router1 again Why? What can you say about the efficiency of RIPng in a network with unstable network paths?

Answer:

I will start a ping and set down the eth123 interface on router1:

```
root@router1:~# ip link set eth123 down
```

Ping Output, the destination unreachable started at sequence 64:

```
root@router3:~#  ping6 2001:0db8:0200:0e01::1
PING 2001:0db8:0200:0e01::1(2001:db8:200:e01::1) 56 data bytes
64 bytes from 2001:db8:200:e01::1: icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=3 ttl=64 time=0.136 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=4 ttl=64 time=0.129 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=5 ttl=64 time=0.159 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=6 ttl=64 time=0.122 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=7 ttl=64 time=0.125 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=8 ttl=64 time=0.123 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=9 ttl=64 time=0.126 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=10 ttl=64 time=0.125 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=11 ttl=64 time=0.122 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=12 ttl=64 time=0.124 ms
```

```
64 bytes from 2001:db8:200:e01::1: icmp_seq=13 ttl=64 time=0.123 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=14 ttl=64 time=0.130 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=15 ttl=64 time=0.123 ms

From 2001:db8:200:e29::3 icmp_seq=63 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=64 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=65 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=66 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=67 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=68 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=69 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=70 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=71 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=72 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=73 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=74 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=75 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=76 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=77 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=78 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=79 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=80 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=81 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=82 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=83 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=84 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=85 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=86 Destination unreachable: Address
```

```
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=87 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=88 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=89 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=90 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=91 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=92 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=93 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=94 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=95 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=96 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=97 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=98 Destination unreachable: Address
unreachabl                    le
From 2001:db8:200:e29::3 icmp_seq=99  Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=100 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=101 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=102 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=103 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=104 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=105 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=106 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=107 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=108 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=109 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=110 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=111 Destination unreachable: Address
unreachable
```

```
From 2001:db8:200:e29::3 icmp_seq=112 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=113 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=114 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=115 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=116 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=117 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=118 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=119 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=120 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=121 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=122 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=123 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=124 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=125 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=126 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=127 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=128 Destination unreachable: Address
unreachable
From 2001:db8:200:e29::3 icmp_seq=129 Destination unreachable: Address
unreachable
64 bytes from 2001:db8:200:e01::1: icmp_seq=214 ttl=62 time=1.40 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=215 ttl=62 time=0.094 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=216 ttl=62 time=0.181 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=217 ttl=62 time=0.178 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=218 ttl=62 time=0.179 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=219 ttl=62 time=0.176 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=220 ttl=62 time=0.191 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=221 ttl=62 time=0.183 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=222 ttl=62 time=0.183 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=223 ttl=62 time=0.187 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=224 ttl=62 time=0.178 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=225 ttl=62 time=0.177 ms
64 bytes from 2001:db8:200:e01::1: icmp_seq=226 ttl=62 time=0.184 ms
^C
--- 2001:0db8:0200:0e01::1 ping statistics ---
```

```
226 packets transmitted, 28 received, +67 errors, 87% packet loss, time
230353ms
rtt min/avg/max/mdev = 0.094/0.230/1.400/0.293 ms
```

- How much time does it take for router3 to reach router1 again Why?

Since I set a timer for that, it took around 3.32 minutes to reach router1 again. The reason for that is that It needs to learn another route to that destination. The reason is that the ping packet can't use the route known by the router since it's down. Now the packet has to travel all the way between the routers. But this is not known yet. The packets can flow again as soon as this is noticed and all nodes have to agree.

- What can you say about the efficiency of RIPng in a network with unstable network paths?

Efficiency ? for such a simple network it tools 3.32 minutes to get a new new network path, so what do you think it will take in suck a real life huge network with unstable network paths.

Feedback received: Can you explain the 3 minutes from the protocol parameters?

The 3 minutes parameter means the default time out value of RIP “180 seconds” + “30 seconds” for sending updates.

```
Sending updates every 30 seconds with +/-50%, next due in 5 seconds
Timeout after 180 seconds, garbage collect after 120 seconds
```

Answer:

Task 12. Using the packet dump for the L123 segment explain very briefly the startup and convergence of the RIPng protocol and the payload of the packets (look for interesting packets, when are they sent and why). Only mention the relevant packets. Upload the raw dump file (as generated by the sniffer) to your wiki and provide a link on your log.

Answer:

Routers starts by sending a RIPng request message to multicast group ff02::9. This is the multicast group “for RIP”

The request messages only sent when RIP protocol first starts specifying neighbors. Then, only response messages are sent. These messages contain the full or only a portion of the modified routing table of a RIP router.

As we notice above that all routing entries have a metric 1. This means that this message is part of the RIP startup. However, if the entry inside a response message has higher metric, that means that it is part of the periodic updates

packet dump for the L123 segment

Task 13. Configure router1 such that it will accept updates only from router2 and router3. Show the configuration changes. Warning: some solutions only work via ripngd’s telnet interface, not vtysh.

Answer:

Before we apply my solution:

```
router1# show ipv6  route  ripng
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* ::/0 [120/5] via fe80::216:3eff:fe42:6980, eth14, 23:39:31
R>* 2001:db8:200:e02::/64 [120/5] via fe80::216:3eff:fe42:6980, eth14,
23:39:31
R>* 2001:db8:200:e03::/64 [120/4] via fe80::216:3eff:fe42:6980, eth14,
23:39:31
R>* 2001:db8:200:e04::/64 [120/2] via fe80::216:3eff:fe42:6980, eth14,
1d01h04m
R>* 2001:db8:200:e05::/64 [120/3] via fe80::216:3eff:fe42:6980, eth14,
1d01h04m
R>* 2001:db8:200:e0f::/64 [120/3] via fe80::216:3eff:fe42:6980, eth14,
23:39:21
R>* 2001:db8:200:e1f::/64 [120/2] via fe80::216:3eff:fe42:6980, eth14,
1d01h04m
R>* 2001:db8:200:e29::/64 [120/4] via fe80::216:3eff:fe42:6980, eth14,
23:39:31
```

In order to make router1 accept updates only from router2 and router3, we can use access_list command as stated in the documentation “You can apply access lists to the interface with a distribute-list command. access_list is the access list name. direct is ‘in’ or ‘out’. If direct is ‘in’ the access list is applied to input packets.”

Finally, after a long time trying to find the distribute-list. I connected to the router using ripngd’s telnet interface and applied my solution because distribute-list is not working using vtysh.

```
root@router1:~# telnet ::1 ripngd
Trying ::1...
Connected to ::1.
Escape character is '^]'.
```

```
Hello, this is Quagga (version 0.99.24.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router1> ena
router1> enable
router1# conf t
router1(config)# ipv6 access-list no-update deny any
router1(config)# router ripng
router1(config-router)# distribute-list no-update in eth14
```

Now, Let's test it:

```
router1# show ipv6  route  ripng
router1#
```

It is empty, maybe because in previous question we set down eth123.

Let's test again after making eth123 up again “ip link set eth123 up”:

```
router1# show ipv6  route  ripng
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* ::/0 [120/2] via fe80::216:3eff:fe0d:ab86, eth123, 00:00:42
R>* 2001:db8:200:e02::/64 [120/2] via fe80::216:3eff:fe0d:ab86, eth123,
00:00:42
R>* 2001:db8:200:e03::/64 [120/2] via fe80::216:3eff:fee7:5118, eth123,
00:00:42
R>* 2001:db8:200:e04::/64 [120/4] via fe80::216:3eff:fee7:5118, eth123,
00:00:42
R>* 2001:db8:200:e05::/64 [120/3] via fe80::216:3eff:fee7:5118, eth123,
00:00:42
R>* 2001:db8:200:e0f::/64 [120/2] via fe80::216:3eff:fee7:5118, eth123,
00:00:42
R>* 2001:db8:200:e1f::/64 [120/3] via fe80::216:3eff:fee7:5118, eth123,
00:00:42
R>* 2001:db8:200:e29::/64 [120/2] via fe80::216:3eff:fe0d:ab86, eth123,
00:00:42
```

It works :D :D :D :D :D

Source:

1- http://www.nongnu.org/quagga/docs/docs-multi/Filtering-RIP-Routes.html