Mininet Network Emulation

Dhaval Bagal

November 2021

Contents

Chapter 1

Static routing

1.1 Network topology

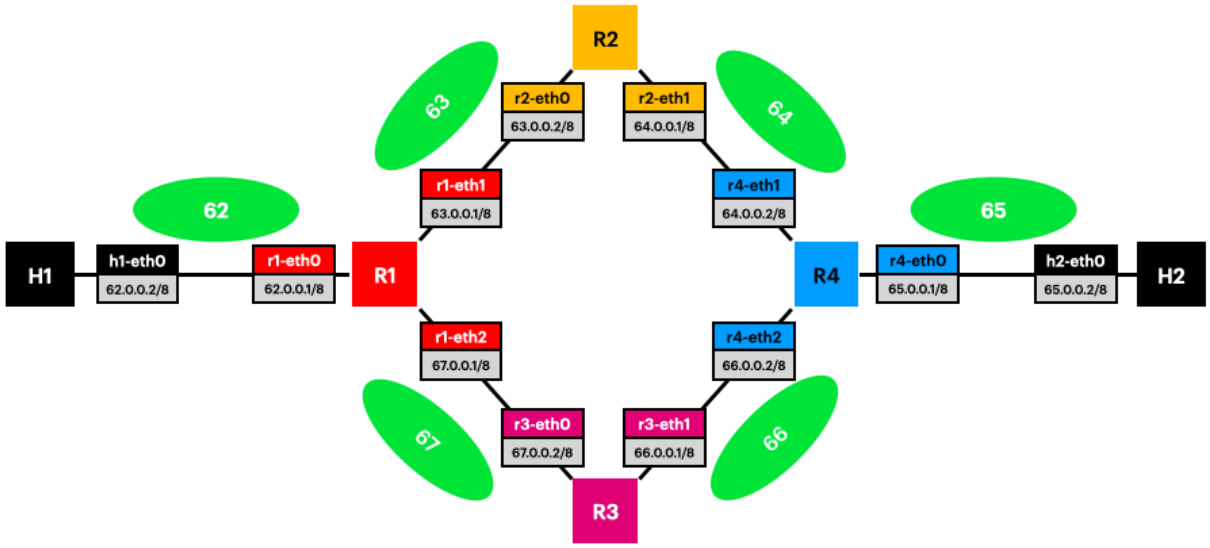

Figure 1.1: Topology

1.2 Routing tables

1.2.1 Router R1

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	63.0.0.2	0.0.0.0	UG	0	0	0	r1-eth1
62.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	r1-eth0
63.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	r1-eth1
66.0.0.0	67.0.0.2	255.0.0.0	UG	0	0	0	r1-eth2
67.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	r1-eth2

1.2.2 Router R2

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         64.0.0.2        0.0.0.0         UG    0      0        0 r2-eth1
62.0.0.0        63.0.0.1        255.0.0.0       UG    0      0        0 r2-eth0
63.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r2-eth0
64.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r2-eth1
67.0.0.0        63.0.0.1        255.0.0.0       UG    0      0        0 r2-eth0
```

1.2.3 Router R3

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         66.0.0.2        0.0.0.0         UG    0      0        0 r3-eth1
62.0.0.0        67.0.0.1        255.0.0.0       UG    0      0        0 r3-eth0
63.0.0.0        67.0.0.1        255.0.0.0       UG    0      0        0 r3-eth0
66.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth1
67.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth0
```

1.2.4 Router R4

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         64.0.0.1        0.0.0.0         UG    0      0        0 r4-eth1
64.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r4-eth1
65.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r4-eth0
66.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r4-eth2
67.0.0.0        66.0.0.1        255.0.0.0       UG    0      0        0 r4-eth2
```

1.2.5 Static routing

The **ip route** command is used to add static routes on a node. The syntax of the command is as follows:

```
ip route [dest-network-address] via [next-hop-ip]
    dev [src-exit-interface-name]
```

Few things to be careful about:

1. **dest-network-address** should contain the cidr decimal
2. **next-hop-ip** should NOT contain the cidr decimal
3. **next-hop-ip** and **src-exit-interface-name** should be on the same network

For every node in the network, static routes are added for every destination network which the node is not a part of or whose network prefix doesn't match the network prefix for any of the node's interfaces.

1.3 Traceroute output

1.3.1 H1 to H2

```
traceroute to 65.0.0.2 (65.0.0.2), 5 hops max, 60 byte packets
 1  62.0.0.1 (62.0.0.1)  0.049 ms  0.008 ms  0.006 ms
 2  63.0.0.2 (63.0.0.2)  0.024 ms  0.010 ms  0.009 ms
 3  64.0.0.2 (64.0.0.2)  0.025 ms  0.011 ms  0.011 ms
 4  65.0.0.2 (65.0.0.2)  0.027 ms  0.015 ms  0.013 ms
```

1.3.2 H2 to H1

```
traceroute to 62.0.0.2 (62.0.0.2), 5 hops max, 60 byte packets
 1  65.0.0.1 (65.0.0.1)  0.034 ms  0.006 ms  0.004 ms
 2  64.0.0.1 (64.0.0.1)  0.012 ms  0.006 ms  0.006 ms
 3  63.0.0.1 (63.0.0.1)  0.014 ms  0.008 ms  0.007 ms
 4  62.0.0.2 (62.0.0.2)  0.017 ms  0.010 ms  0.009 ms
```

Chapter 2

BIRD inter-domain routing

2.1 Routing tables

2.1.1 Router R1

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
62.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r1-eth0
62.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r1-eth0
63.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r1-eth1
63.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r1-eth1
64.0.0.0        63.0.0.2        255.0.0.0       UG    32     0        0 r1-eth1
66.0.0.0        67.0.0.2        255.0.0.0       UG    32     0        0 r1-eth2
67.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r1-eth2
67.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r1-eth2
```

2.1.2 Router R2

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
62.0.0.0        63.0.0.1        255.0.0.0       UG    32     0        0 r2-eth0
63.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r2-eth0
63.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r2-eth0
64.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r2-eth1
64.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r2-eth1
65.0.0.0        64.0.0.2        255.0.0.0       UG    32     0        0 r2-eth1
66.0.0.0        64.0.0.2        255.0.0.0       UG    32     0        0 r2-eth1
67.0.0.0        63.0.0.1        255.0.0.0       UG    32     0        0 r2-eth0
```

2.1.3 Router R3

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
62.0.0.0        67.0.0.1        255.0.0.0       UG    32     0        0 r3-eth0
63.0.0.0        67.0.0.1        255.0.0.0       UG    32     0        0 r3-eth0
64.0.0.0        66.0.0.2        255.0.0.0       UG    32     0        0 r3-eth1
65.0.0.0        66.0.0.2        255.0.0.0       UG    32     0        0 r3-eth1
66.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth1
66.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r3-eth1
```

```
67.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth0
67.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r3-eth0
```

2.1.4 Router R4

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
62.0.0.0        64.0.0.1        255.0.0.0       UG    32     0        0 r4-eth1
63.0.0.0        64.0.0.1        255.0.0.0       UG    32     0        0 r4-eth1
64.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r4-eth1
64.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r4-eth1
65.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r4-eth0
65.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r4-eth0
66.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r4-eth2
66.0.0.0        0.0.0.0         255.0.0.0       U     32     0        0 r4-eth2
67.0.0.0        66.0.0.1        255.0.0.0       UG    32     0        0 r4-eth2
```

2.2 Tracing routes

```
traceroute to 65.0.0.2 (65.0.0.2), 5 hops max, 60 byte packets
 1  62.0.0.1 (62.0.0.1)  0.030 ms  0.006 ms  0.005 ms
 2  63.0.0.2 (63.0.0.2)  0.014 ms  0.007 ms  0.008 ms
 3  64.0.0.2 (64.0.0.2)  0.015 ms  0.008 ms  0.009 ms
 4  65.0.0.2 (65.0.0.2)  0.017 ms  0.011 ms  0.010 ms
```

2.3 Getting a link down

The command to get any link down in mininet is as follows:

```
link [node-1-name] [node-2-name] down
```

You can also get the link down using python code as follows:

```
info(net.configLinkStatus('node-1-name','node-2-name','down'))
```

2.4 Tracing routes after a link is down

```
traceroute to 65.0.0.2 (65.0.0.2), 5 hops max, 60 byte packets
 1  62.0.0.1 (62.0.0.1)  0.026 ms  0.005 ms  0.004 ms
 2  67.0.0.2 (67.0.0.2)  0.014 ms  0.008 ms  0.007 ms
 3  66.0.0.2 (66.0.0.2)  0.015 ms  0.010 ms  0.008 ms
 4  65.0.0.2 (65.0.0.2)  0.015 ms  0.010 ms  0.010 ms
```

Chapter 3

Network performance

3.1 Experimental setup

We experiment with 3 buffer size settings viz 10Kb, 5Mb and 25Mb. The command that is used to set buffer size and delay at each router interface is as follows:

```
tc qdisc replace dev [intf-name] root netem limit [num-packets-waiting]
delay 30ms loss 0.1%%
```

netem allows setting the buffer size in terms of number of packets. Thus, we have to convert the buffer size into number of packets.

The calculation can be done as follows:

$$\begin{aligned} \frac{\text{10Kbits}}{\text{1500bytes}} &= 0.85 \approx 1 \\ \frac{\text{5Mbits}}{\text{1500bytes}} &\approx 437 \\ \frac{\text{25Mbits}}{\text{1500bytes}} &\approx 2185 \end{aligned} \tag{3.1}$$

3.1.1 Buffer size = 10Kb

Server

```
------------------------------------------------------------
Server listening on 8000
------------------------------------------------------------
Accepted connection from 62.0.0.2, port 41872
[  5] local 65.0.0.2 port 8000 connected to 62.0.0.2 port 41874
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.00  sec  80.6 KBytes  66.0 Kbits/sec
[  5]  10.00-11.17  sec  25.5 KBytes   179 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-11.17  sec   202 KBytes   148 Kbits/sec   30             sender
```

```
[  5]   0.00-11.17  sec   106 KBytes  77.8 Kbits/sec                  receiver
```

Client

```
Connecting to host 65.0.0.2, port 8000
[  4] local 62.0.0.2 port 41874 connected to 65.0.0.2 port 8000
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec   202 KBytes   166 Kbits/sec   30   5.66 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   202 KBytes   166 Kbits/sec   30             sender
[  4]   0.00-10.00  sec   106 KBytes  86.8 Kbits/sec                  receiver

iperf Done.
```

3.1.2 Buffer size = 5Mb

Server

```
-----------------------------------------------------------
Server listening on 8000
-----------------------------------------------------------
Accepted connection from 62.0.0.2, port 41876
[  5] local 65.0.0.2 port 8000 connected to 62.0.0.2 port 41878
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.00  sec  11.0 MBytes  9.23 Mbits/sec
[  5]  10.00-10.25  sec   132 KBytes  4.25 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-10.25  sec  13.7 MBytes  11.2 Mbits/sec   41             sender
[  5]   0.00-10.25  sec  11.1 MBytes  9.11 Mbits/sec                  receiver
```

Client

```
Connecting to host 65.0.0.2, port 8000
[  4] local 62.0.0.2 port 41878 connected to 65.0.0.2 port 8000
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  13.7 MBytes  11.5 Mbits/sec   41    132 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  13.7 MBytes  11.5 Mbits/sec   41             sender
[  4]   0.00-10.00  sec  11.1 MBytes  9.33 Mbits/sec                  receiver

iperf Done.
```

3.1.3 Buffer size = 25Mb

Server

```
-----------------------------------------------------------
Server listening on 8000
-----------------------------------------------------------
Accepted connection from 62.0.0.2, port 41880
[  5] local 65.0.0.2 port 8000 connected to 62.0.0.2 port 41882
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.00  sec  47.6 MBytes  39.9 Mbits/sec
```

```
[  5]  10.00-10.26  sec   272 KBytes  8.69 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-10.26  sec  58.8 MBytes  48.1 Mbits/sec   74             sender
[  5]   0.00-10.26  sec  47.9 MBytes  39.1 Mbits/sec                  receiver
```

Client

```
Connecting to host 65.0.0.2, port 8000
[  4] local 62.0.0.2 port 41882 connected to 65.0.0.2 port 8000
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  58.8 MBytes  49.3 Mbits/sec   74    247 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  58.8 MBytes  49.3 Mbits/sec   74             sender
[  4]   0.00-10.00  sec  47.9 MBytes  40.1 Mbits/sec                  receiver

iperf Done.
```

3.2 Analysis

Bandwidth-delay product (BDP) is given by the product of the bandwidth and the RTT.

We can calculate the BDP as follows:

```
Given:
------
    Bandwidth = 100Mbits

Calculation:
------------
    RTT = delay at each egress interface + propogation delay

    There are 3 routers and 4 links in between h1 and h2.

    The route taken is as follows:
    h1-eth0(egres) -> r1-eth0(ingress) ->
    r1-eth1(egress) -> r2-eth0(ingress) ->
    r2-eth1(egress) -> r4-eth1(ingress) ->
    r4-eth0(egress)

    RTT = 4*30ms + 4*30ms = 240ms

    Using ping, the RTT is verified to be 244ms. The additional 4ms is
    due to processing delays, and many other factors.

    BDP = 100Mb * 244ms = 24.4 Mbits
```

The throughput should be close to the bandwidth (100Mb) when the router buffer size is greater than or equal to the BDP. Similarly, when the router buffer size is less than the BDP, the throughput is reduced.

From the results obtained in section 1, we see that throughput with a buffer size of 10k is much lesser than that with buffer sizes of 5Mb which is further lesser than 25Mb.

Throughput as seen at the client for each of the three buffer setting is as follows:

```
10Kbits buffer  =>   throughput=166Kbps
5Mbits buffer   =>   throughput=11.5Mbps
25Mbits buffer  =>   throughput=49.3Mbps
```

You can clearly see that when buffer size is less than the BDP (24.4Mbits), the throughput suffers. Hence the throughput for 10Kbits and 5Mbits buffer setting is much less than the bandwidth (100Mbits).