



Figure 1: Network Topology

### B1:

I created the above topology on mininet using the mininet Python API. It consists of 2 hosts, H1 and H2, connected through a network of 4 routers R1-R4.

We need to spawn a BIRD daemon on every router to enable dynamic routing using the RIP protocol. The main configuration file, bird.conf can be found for every router in its corresponding folder (R1/R2/R3/R4) under the main partB subdirectory. In Figure 2 routing tables of all routers are shown. Result of the pingall command can be found in Figure 3, which indicates that all the nodes in the network are connected to each other and there are no message drops.

R1's Routing Table:							
Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.10.1.0	0.0.0.0	255.255.255.0	U	0	0	0	r1-eth0
172.10.1.0	0.0.0.0	255.255.255.0	U	32	0	0	r1-eth0
172.10.2.0	0.0.0.0	255.255.255.0	U	0	0	0	r1-eth1
172.10.2.0	0.0.0.0	255.255.255.0	U	32	0	0	r1-eth1
172.10.3.0	0.0.0.0	255.255.255.0	U	0	0	0	r1-eth2
172.10.3.0	0.0.0.0	255.255.255.0	U	32	0	0	r1-eth2
172.10.4.0	172.10.2.1	255.255.255.0	UG	32	0	0	r1-eth1
172.10.5.0	172.10.3.1	255.255.255.0	UG	32	0	0	r1-eth2
172.10.6.0	172.10.2.1	255.255.255.0	UG	32	0	0	r1-eth1
R2's Routing Table:							
Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.10.1.0	172.10.2.2	255.255.255.0	UG	32	0	0	r2-eth0
172.10.2.0	0.0.0.0	255.255.255.0	U	0	0	0	r2-eth0
172.10.2.0	0.0.0.0	255.255.255.0	U	32	0	0	r2-eth0
172.10.3.0	172.10.2.2	255.255.255.0	UG	32	0	0	r2-eth0
172.10.4.0	0.0.0.0	255.255.255.0	U	0	0	0	r2-eth1
172.10.4.0	0.0.0.0	255.255.255.0	U	32	0	0	r2-eth1
172.10.5.0	172.10.4.2	255.255.255.0	UG	32	0	0	r2-eth1
172.10.6.0	172.10.4.2	255.255.255.0	UG	32	0	0	r2-eth1
R3's Routing Table:							
Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.10.1.0	172.10.3.2	255.255.255.0	UG	32	0	0	r3-eth0
172.10.2.0	172.10.3.2	255.255.255.0	UG	32	0	0	r3-eth0
172.10.3.0	0.0.0.0	255.255.255.0	U	0	0	0	r3-eth0
172.10.3.0	0.0.0.0	255.255.255.0	U	32	0	0	r3-eth0
172.10.4.0	172.10.5.2	255.255.255.0	UG	32	0	0	r3-eth1
172.10.5.0	0.0.0.0	255.255.255.0	U	0	0	0	r3-eth1
172.10.5.0	0.0.0.0	255.255.255.0	U	32	0	0	r3-eth1
172.10.6.0	172.10.5.2	255.255.255.0	UG	32	0	0	r3-eth1
R4's Routing Table:							
Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.10.2.0	172.10.4.1	255.255.255.0	UG	32	0	0	r4-eth1
172.10.3.0	172.10.5.1	255.255.255.0	UG	32	0	0	r4-eth2
172.10.4.0	0.0.0.0	255.255.255.0	U	0	0	0	r4-eth1
172.10.4.0	0.0.0.0	255.255.255.0	U	32	0	0	r4-eth1
172.10.5.0	0.0.0.0	255.255.255.0	U	0	0	0	r4-eth2
172.10.5.0	0.0.0.0	255.255.255.0	U	32	0	0	r4-eth2
172.10.6.0	0.0.0.0	255.255.255.0	U	0	0	0	r4-eth0
172.10.6.0	0.0.0.0	255.255.255.0	U	32	0	0	r4-eth0

Figure 2: Routing tables of all routers

```

Pinging all nodes and routers
*** Ping: testing ping reachability
H1 -> H2 R1 R2 R3 R4
H2 -> H1 R1 R2 R3 R4
R1 -> H1 H2 R2 R3 R4
R2 -> H1 H2 R1 R3 R4
R3 -> H1 H2 R1 R2 R4
R4 -> H1 H2 R1 R2 R3
*** Results: 0% dropped (30/30 received)
0.0*** Starting CLI:

```

Figure 3: pingall command result

Also, result of traceroute command between H1 and H2 is shown in Figure 4.

```

mininet> H1 traceroute H2
traceroute to 172.10.6.2 (172.10.6.2), 30 hops max, 60 byte packets
 1  172.10.1.1 (172.10.1.1)  0.023 ms  0.003 ms  0.003 ms
 2  172.10.3.1 (172.10.3.1)  0.011 ms  0.004 ms  0.004 ms
 3  172.10.5.2 (172.10.5.2)  0.011 ms  0.005 ms  0.005 ms
 4  172.10.6.2 (172.10.6.2)  0.032 ms  0.008 ms  0.007 ms
mininet> H2 traceroute H1
traceroute to 172.10.1.2 (172.10.1.2), 30 hops max, 60 byte packets
 1  172.10.6.1 (172.10.6.1)  0.108 ms  0.093 ms  0.089 ms
 2  172.10.5.1 (172.10.5.1)  0.086 ms  0.078 ms  0.074 ms
 3  172.10.3.2 (172.10.3.2)  0.071 ms  0.072 ms  0.067 ms
 4  172.10.1.2 (172.10.1.2)  0.065 ms  0.058 ms  0.052 ms

```

Figure 4: output of traceroute command between H1 and H2

**B2:**

I then bring down the R1-R2 link with the following command:

link R1 R3 down

We can now see the updated traceroute output as shown in the Figure 5. The new route goes through the 172.10.2.x/24 and 172.10.4.x/24 subnets (previously 172.10.3.x/24 and 172.10.5.x/24).

```

mininet> link R1 R3 down
mininet> H1 traceroute H2
traceroute to 172.10.6.2 (172.10.6.2), 30 hops max, 60 byte packets
 1  172.10.1.1 (172.10.1.1)  0.304 ms  0.252 ms  0.249 ms
 2  172.10.2.1 (172.10.2.1)  0.246 ms  0.181 ms  0.154 ms
 3  172.10.4.2 (172.10.4.2)  0.126 ms  0.116 ms  0.111 ms
 4  172.10.6.2 (172.10.6.2)  0.058 ms  0.050 ms  0.045 ms
mininet> H2 traceroute H1
traceroute to 172.10.1.2 (172.10.1.2), 30 hops max, 60 byte packets
 1  172.10.6.1 (172.10.6.1)  0.106 ms  0.091 ms  0.087 ms
 2  172.10.4.1 (172.10.4.1)  0.084 ms  0.076 ms  0.073 ms
 3  172.10.2.2 (172.10.2.2)  0.070 ms  0.061 ms  0.057 ms
 4  172.10.1.2 (172.10.1.2)  0.055 ms  0.048 ms  0.043 ms

```

Figure 5: traceroute command result after the link between R1 and R3 goes down