COL334 Computer Networks Assignment-4

Chinmay Mittal (2020CS10336)

Part-B

Explanation of Code

The code for this part is in 3 files. Namely, *ass-4-part-2.1.cc, ass-4-part-2.2.cc and ass-4-part-2.3.cc* for taks b.i, b.ii and b.iii respectively. These have been modified from the base file provided in the routing examples of ns3 namely *ns-allinone-3.29/ns-3.29/examples/routing/rip-simple-network.cc.*
For part b.i one has to change the csma channel delay values and run the script.

```
 // change csma channel delay values in ass-4-part-2.1.cc
 uint32_t csmaDelay = 80000 ;
 // to run ass-4-part-2.1.cc =>  ./waf --run scratch/ass-4-part-2.1
```

This file logs the routing tables at some timesteps to the console. Also it creates a more detailed version of the routing tables at each second in text files to check for convergence namely b.1-routing-a-100.txt, b.1-routing-b-100.txt and b.1-routing-c-100.txt (here 100 is the csma delay values). I have manually analyzed the routing table and created .txt files for the data points for plotting convergence namely ass-4-b.1.1.txt, ass-4-b.1.2.txt etc (here 1 and 2 are for delay values of 100 and 1000). These files are used by *convergence.py* to make and save the corresponding graphs ass-4-b.1.1.png, ass-4-b.1.2.png. To use convergence.py one has to download matplotlib

```
 // change filename in convergence.py to make corresponding plots
file = "ass-4-b.1.4"
 // to run convergence.py => python convergence.py
```

B.ii

```
 // to run ass-4-part-2.2.cc =>  ./waf --run scratch/ass-4-part-2.2
```

This will log the routing table at appropriate time steps to the console. I have manually analyzed them to check for reachability of destination from source. I have created a text file namely ass-4-b.2.txt with data points which will be used by the plotting file reachibility.py. To use this file one has to download matplotlib.

```
file = "ass-4-b.2" // change the filename in reachibility.py
 // to run reachibility.py => python reachibility.py
```

This creates a corresponding png file named ass-4-b.2.png with the reachability graph.

B.iii

```
 // to run ass-4-part-2.3.cc =>  ./waf --run scratch/ass-4-part-2.3
```

This will log the routing table at appropriate time steps to the console. I have manually analyzed them to check for reachability of destination from source. I have created a text file namely ass-4-b.3.txt with data points which will be used by the plotting file reachibility.py. To use this file one has to download matplotlib.

```
file = "ass-4-b.3" // change the filename in reachibility.py
 // to run reachibility.py => python reachibility.py
```

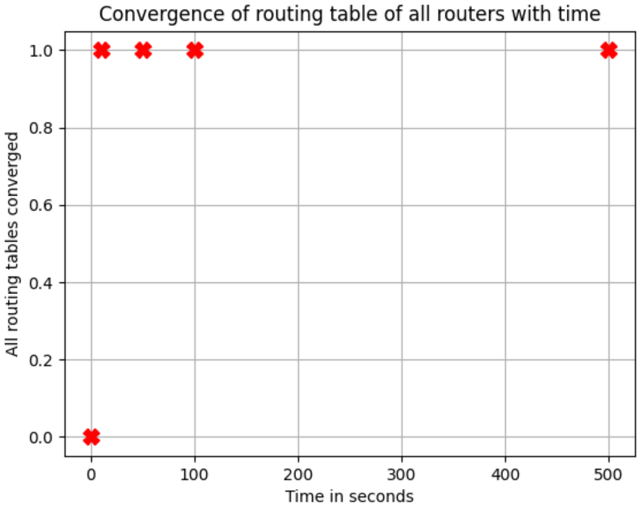
This creates a corresponding png file named ass-4-b.3.png with the reachability graph.

To re-establish a link I have used the following helper

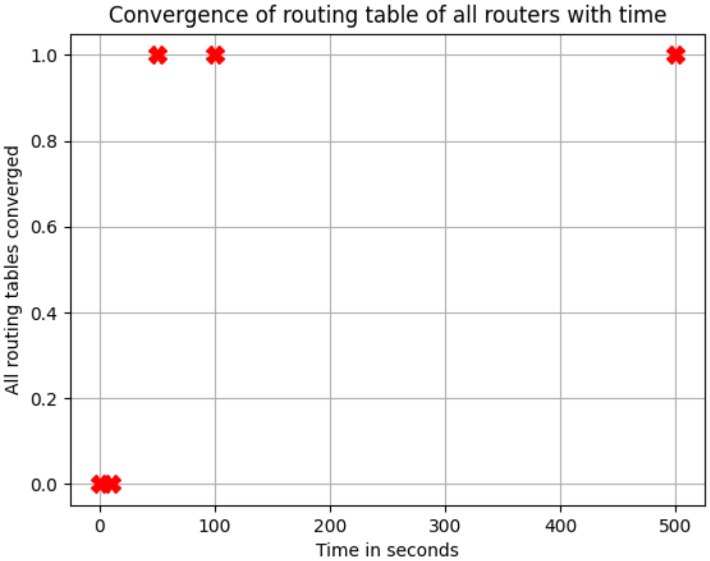
```
void ReJoinLink (Ptr<Node> nodeA, Ptr<Node> nodeB, uint32_t interfaceA,
uint32_t interfaceB)
{
   nodeA->GetObject<Ipv4> ()->SetUp (interfaceA);
   nodeB->GetObject<Ipv4> ()->SetUp (interfaceB);
}
 // This callback is used as follows:
 // re-establishing the links
 Simulator::Schedule (Seconds (120), &ReJoinLink, a, b, 2, 1);
 Simulator::Schedule (Seconds (120), &ReJoinLink, a, c, 3, 1);
```

Subpart B.i

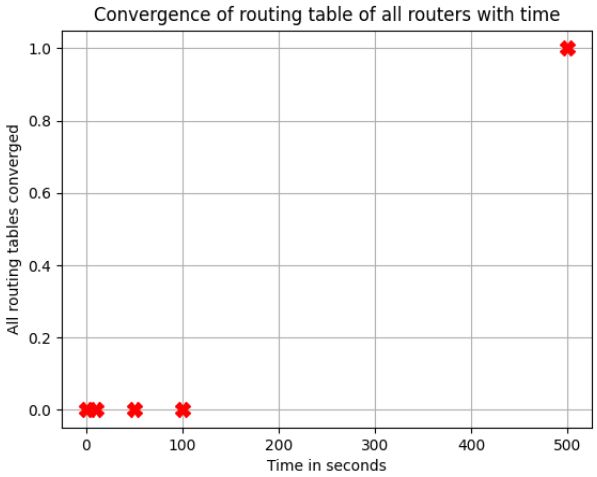
CSMA Delay: 100 msec



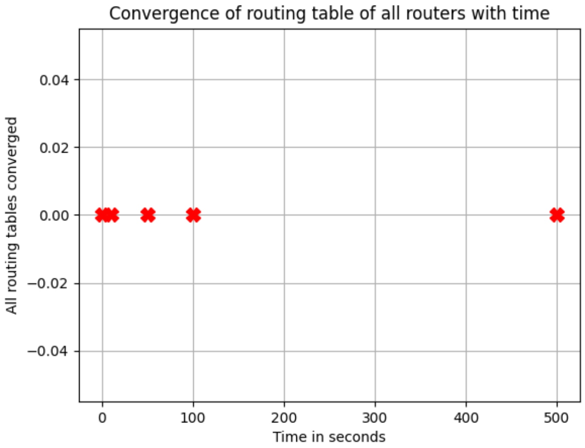
CSMA Delay: 1000 msec



CSMA Delay: 20000 msec



CSMA Delay: 80000 msec



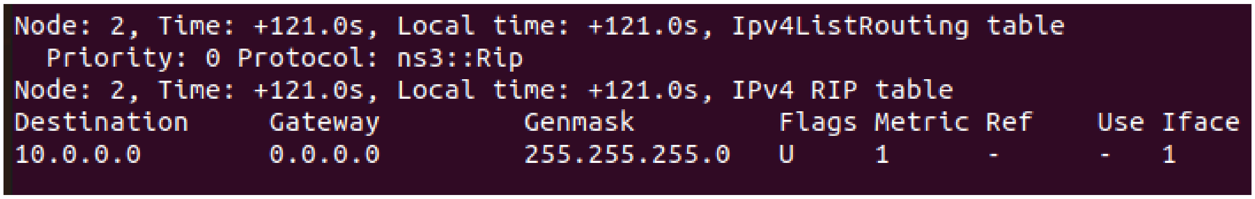
Explanation and observations for Part B.i

Ripng is an internal gateway protocol which uses distance vector approach to compute routing tables with hop counts as the metric. Routers exchange information with each other about their estimates of distances, the router which receives this information uses Bellman Ford algorithm to compute the shortest paths. As we increase the delay of the CSMA channels used for exchanging this information, this information flows from one node to the other slowly which slows down the convergence of the routing tables. As we can see from the graph, the delays increase the time required for convergence of routing tables of all routers.

Subpart B.ii

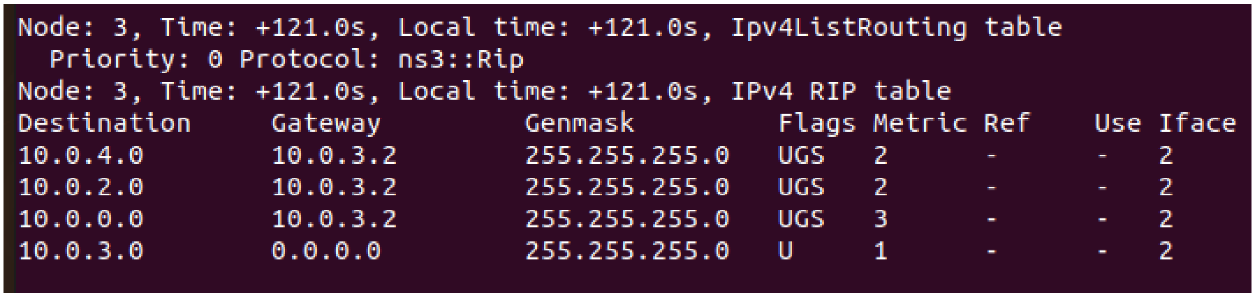
Routing Tables at 121 seconds

Router R1



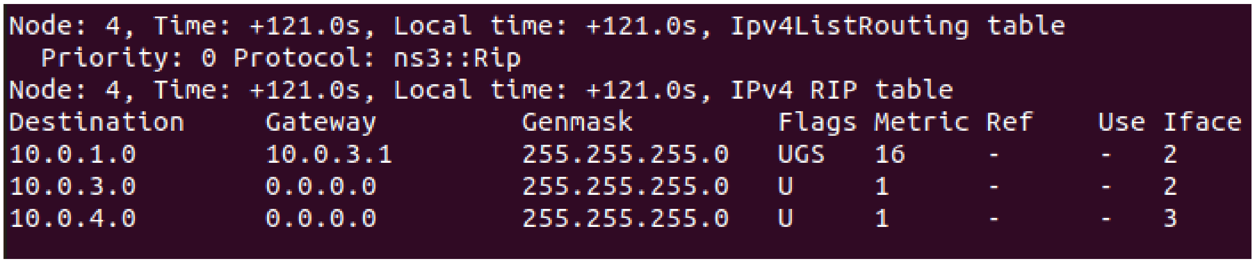
```
Node: 2, Time: +121.0s, Local time: +121.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 2, Time: +121.0s, Local time: +121.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     1      -      -   1
```

Router R2



```
Node: 3, Time: +121.0s, Local time: +121.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 3, Time: +121.0s, Local time: +121.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.4.0        10.0.3.2        255.255.255.0   UGS   2      -      -   2
10.0.2.0        10.0.3.2        255.255.255.0   UGS   2      -      -   2
10.0.0.0        10.0.3.2        255.255.255.0   UGS   3      -      -   2
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
```

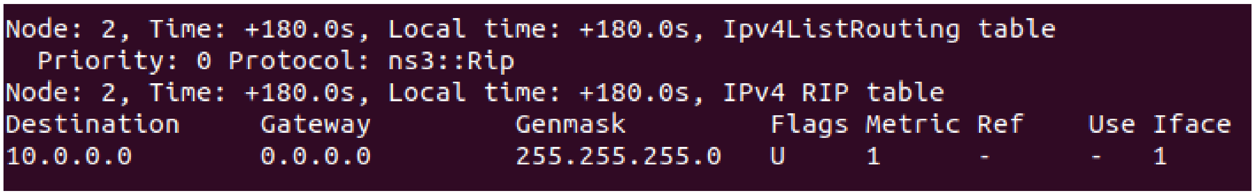
Router R3



```
Node: 4, Time: +121.0s, Local time: +121.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 4, Time: +121.0s, Local time: +121.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        10.0.3.1        255.255.255.0   UGS   16     -      -   2
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
10.0.4.0        0.0.0.0         255.255.255.0   U     1      -      -   3
```

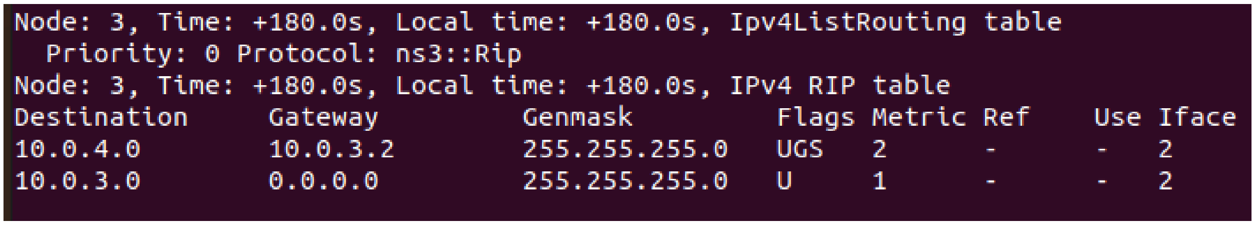
Routing Tables at 180 seconds

Router R1 (Node 2)



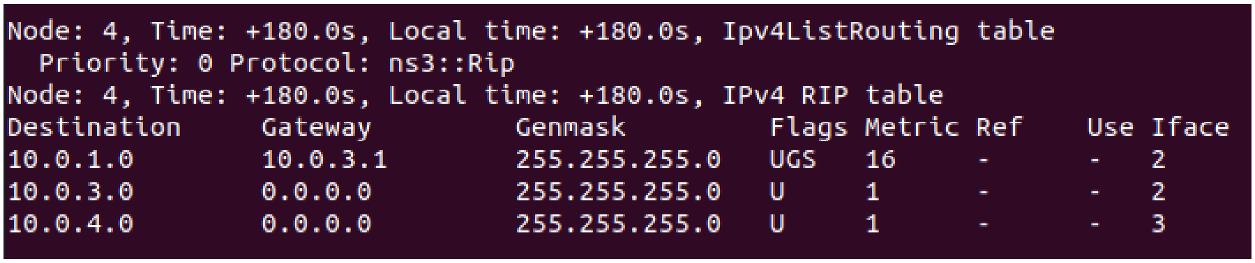
```
Node: 2, Time: +180.0s, Local time: +180.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 2, Time: +180.0s, Local time: +180.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     1      -      -   1
```

Router R2 (Node 3)



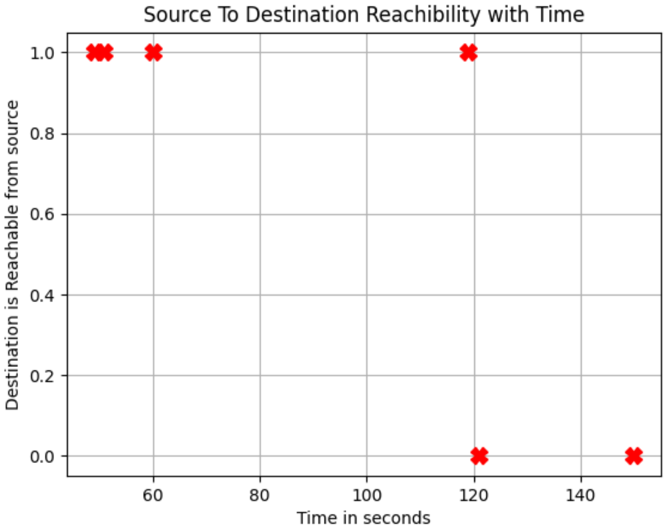
```
Node: 3, Time: +180.0s, Local time: +180.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 3, Time: +180.0s, Local time: +180.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.4.0        10.0.3.2        255.255.255.0   UGS   2      -      -   2
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
```

Router R3 (Node 4)



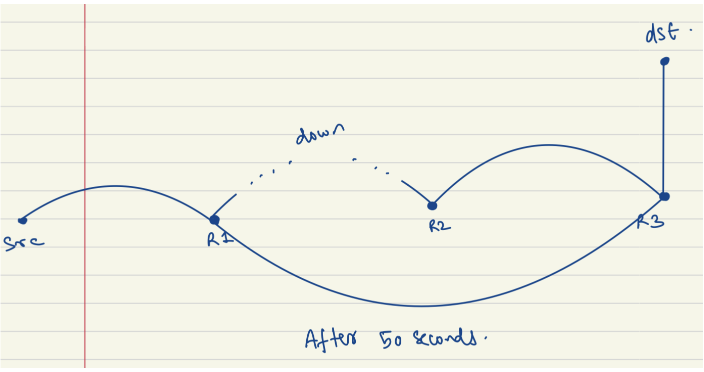
```
Node: 4, Time: +180.0s, Local time: +180.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 4, Time: +180.0s, Local time: +180.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        10.0.3.1        255.255.255.0   UGS   16     -      -   2
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
10.0.4.0        0.0.0.0         255.255.255.0   U     1      -      -   3
```

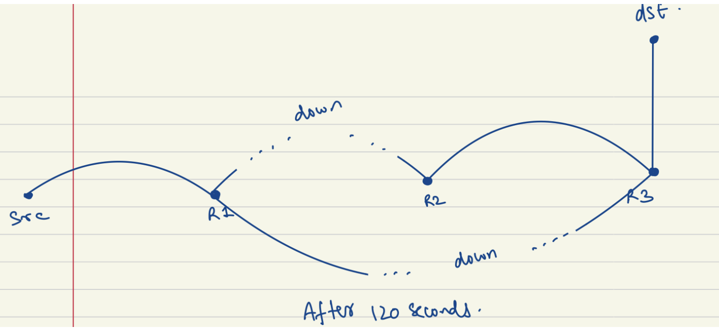
Source to Destination Reachability versus Time



Explanation and Observations

The network after 50 seconds and 120 seconds will look as follows





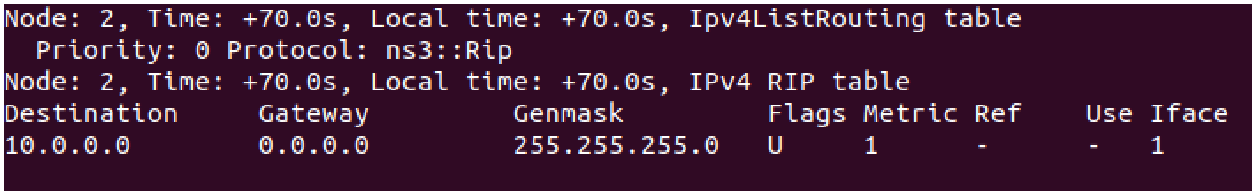
The shortest path from src to destination at the start of the simulation is src => R1 => R3 => dst. After the link R1 ⇔ R2 goes down this shortest path does not change and source is still reachable from the destination. Once link R1 ⇔ R3 also goes down there is no longer any path from the source to the destination and thus reachability in the plot drops to zero. The reachability behavior does not change from 51 seconds to 119 seconds since the shortest path remains the same i.e. src => R1 => R3 => dst. At 121 seconds the destination is no longer reachable from the source. This happens because the shortest distance from src to dest does not change till the link R1 => R3 is up.

Also from the routing tables for the three routers at 121 s and 180 s we can see that it takes some time for the routing table at R2 to update due to CSMA channel delays once the link R1 => R3 goes down.

Subpart B.iii

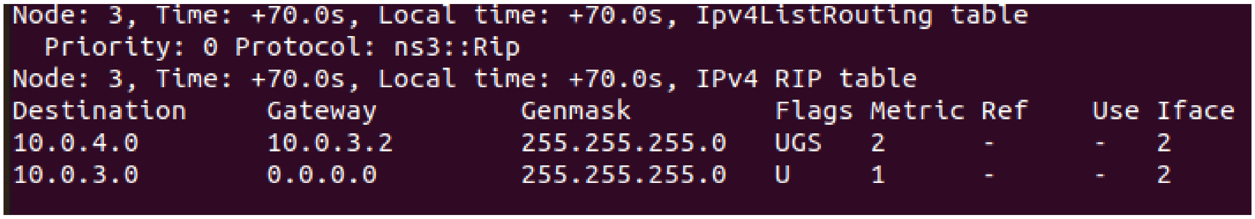
Routing Tables at 70 seconds

Router R1



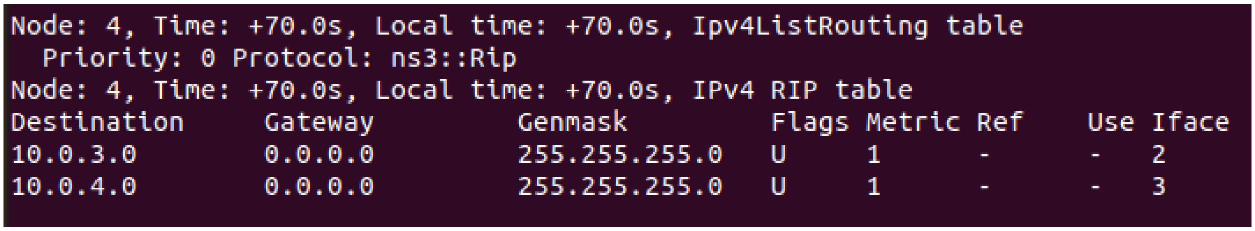
```
Node: 2, Time: +70.0s, Local time: +70.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 2, Time: +70.0s, Local time: +70.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     1      -      -   1
```

Router R2



```
Node: 3, Time: +70.0s, Local time: +70.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 3, Time: +70.0s, Local time: +70.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.4.0        10.0.3.2        255.255.255.0   UGS   2      -      -   2
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
```

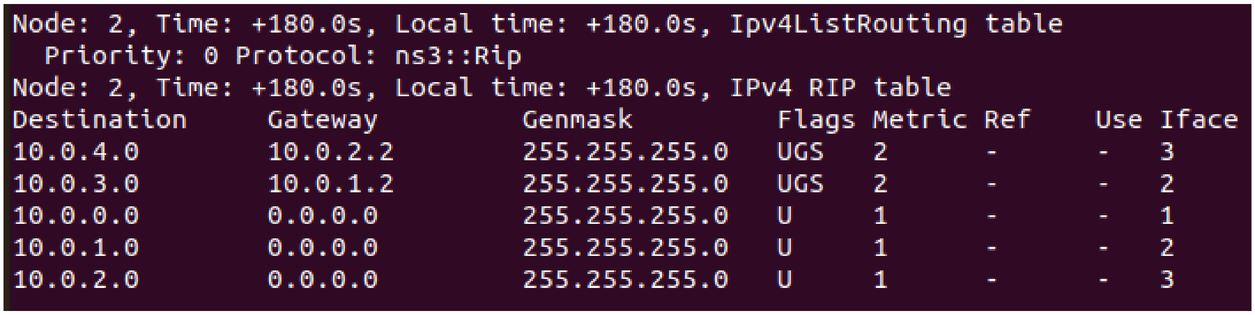
Router R3



```
Node: 4, Time: +70.0s, Local time: +70.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 4, Time: +70.0s, Local time: +70.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
10.0.4.0        0.0.0.0         255.255.255.0   U     1      -      -   3
```

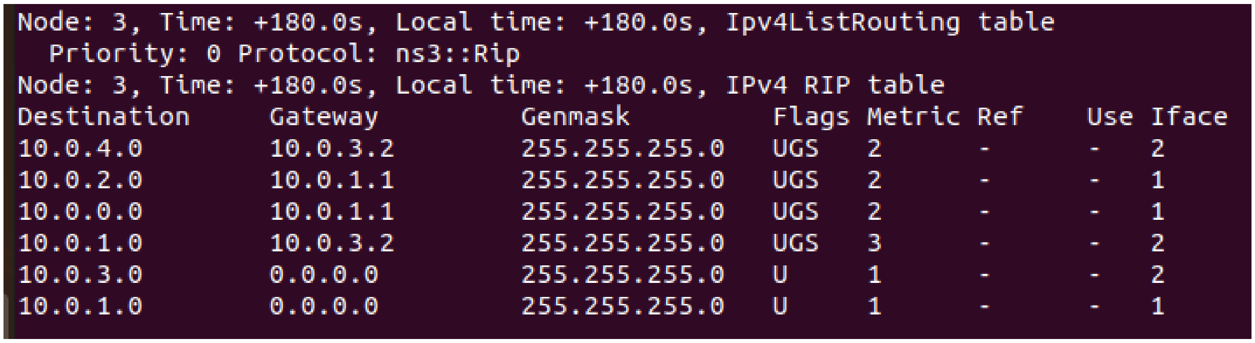
Routing Tables at 180 seconds

Router R1



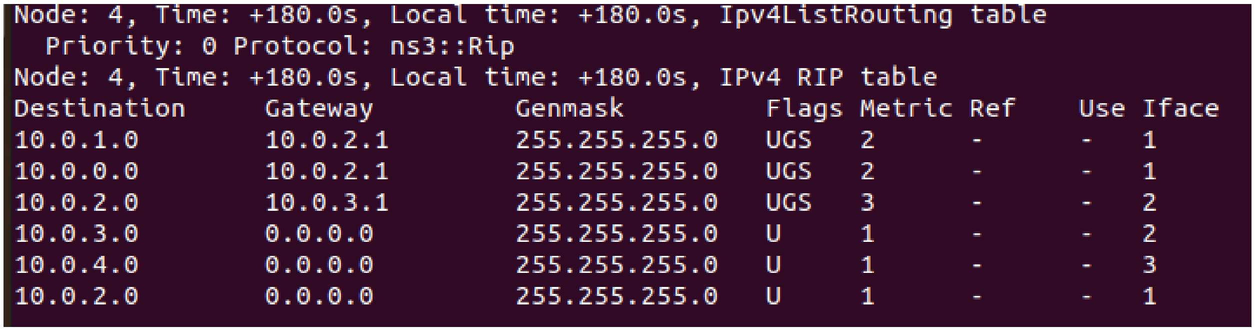
```
Node: 2, Time: +180.0s, Local time: +180.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 2, Time: +180.0s, Local time: +180.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.4.0        10.0.2.2        255.255.255.0   UGS   2      -      -   3
10.0.3.0        10.0.1.2        255.255.255.0   UGS   2      -      -   2
10.0.0.0        0.0.0.0         255.255.255.0   U     1      -      -   1
10.0.1.0        0.0.0.0         255.255.255.0   U     1      -      -   2
10.0.2.0        0.0.0.0         255.255.255.0   U     1      -      -   3
```

Router R2



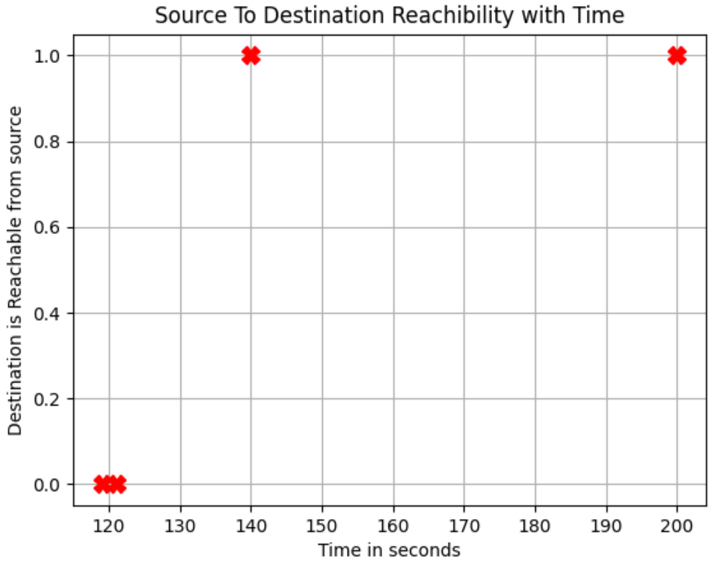
```
Node: 3, Time: +180.0s, Local time: +180.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 3, Time: +180.0s, Local time: +180.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.4.0        10.0.3.2        255.255.255.0   UGS   2      -      -   2
10.0.2.0        10.0.1.1        255.255.255.0   UGS   2      -      -   1
10.0.0.0        10.0.1.1        255.255.255.0   UGS   2      -      -   1
10.0.1.0        10.0.3.2        255.255.255.0   UGS   3      -      -   2
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
10.0.1.0        0.0.0.0         255.255.255.0   U     1      -      -   1
```

Router R3



```
Node: 4, Time: +180.0s, Local time: +180.0s, Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Rip
Node: 4, Time: +180.0s, Local time: +180.0s, IPv4 RIP table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        10.0.2.1        255.255.255.0   UGS   2      -      -   1
10.0.0.0        10.0.2.1        255.255.255.0   UGS   2      -      -   1
10.0.2.0        10.0.3.1        255.255.255.0   UGS   3      -      -   2
10.0.3.0        0.0.0.0         255.255.255.0   U     1      -      -   2
10.0.4.0        0.0.0.0         255.255.255.0   U     1      -      -   3
10.0.2.0        0.0.0.0         255.255.255.0   U     1      -      -   1
```

Source To Destination Reachability with Time



Explanations and Observations

Once both links R1 => R2 and R1 => R3 go down at 50 seconds. The routing tables converge relatively quickly and src is not reachable from the destination. This can be seen from the routing tables at 70 seconds as shown above. The state of the network does not change till 120 seconds thus destination is still not reachable from the source at 119 seconds. At 120 seconds both the links are revived and src => R1 => R3 => dst is the shortest path which can be seen from the routing tables at 180 seconds. Thus after 120 seconds in relatively small amount of time the routing tables converge and destination is reachable from source after 120 seconds.