Modeling the Simulation: R1-R6 are routers and A-K are hosts/end devices

net1: A-C
net2: D-G
net5: H-I
net6: J-K

```
net1.Create(3);
net2.Create(4);
net5.Create(2);
net6.Create(2);
```

Routers: R1-R6

```
NodeContainer routers;
routers.Create(6);
```

For connections used CSMA channel of 100Mbps bandwidth and 1ms delay.

```
CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma.SetChannelAttribute ("Delay", StringValue ("1ms"));
```

All links have a default cost of 1, except the link R4-R5 has a cost of 5.

```
//R4-R5 cost=5
ripRouting.SetInterfaceMetric (routers.Get(3), 3, 5);
ripRouting.SetInterfaceMetric (routers.Get(4), 4, 5);
```

Assign IP addresses to each subnet connected to the router. For example, router-1 interfaces

```
Ipv4AddressHelper address;
  //For R1-A
address.SetBase ("192.168.1.0", "255.255.255.192");
Ipv4InterfaceContainer netR1_Ainterfaces = address.Assign(netR1_A);
//For R1-B
address.SetBase ("192.168.1.64", "255.255.255.192");
Ipv4InterfaceContainer netR1_Binterfaces = address.Assign(netR1_B);
//For R1-C
address.SetBase ("192.168.1.128", "255.255.255.192");
Ipv4InterfaceContainer netR1_Cinterfaces = address.Assign(netR1_C);
...
//For R1-R3
address.SetBase ("10.1.13.0", "255.255.255.0");
Ipv4InterfaceContainer router13Interfaces = address.Assign (netR1_R3);
...
```

Use RIP routing algorithm with Ipv4 and Poison Reverse as split horizon technique.

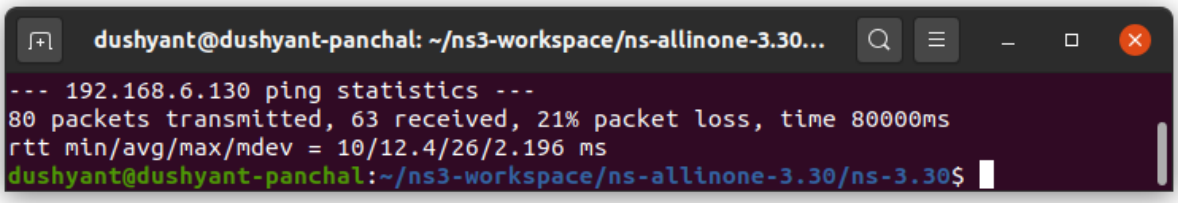
```
Config::SetDefault ("ns3::Rip::SplitHorizon", EnumValue
(RipNg::POISON_REVERSE));
```

After 25 - sec, break the link between R3-R4.

```
void TearDownLink (Ptr<Node> nodeA, Ptr<Node> nodeB, uint32_t interfaceA,
uint32_t interfaceB)
{
 nodeA->GetObject<Ipv4> ()->SetDown (interfaceA);
 nodeB->GetObject<Ipv4> ()->SetDown (interfaceB);
}
...
//Break R3-R4 link at 25s
 Simulator::Schedule(Seconds(25), &TearDownLink, routers34.Get(0),
routers34.Get(1), 2, 2);
//
```

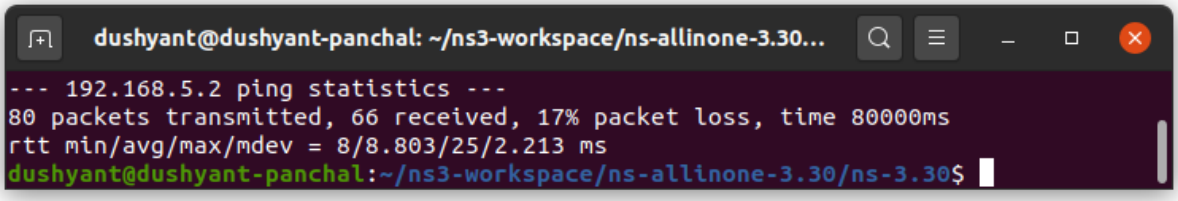
Ping for 80s and show the packet loss and average RTT in ms.

Host A pings K




```
 //A pings K
 V4PingHelper ping1 (netR6_Kinterfaces.GetAddress (1));
 ping1.SetAttribute ("Interval", TimeValue (Seconds(1.0)));
 ping1.SetAttribute ("Size", UintegerValue (1024));
 ping1.SetAttribute ("Verbose", BooleanValue (pingVerbose));
 ApplicationContainer ping1app = ping1.Install (net1.Get(0));
 ping1app.Start (Seconds (1.0));
 ping1app.Stop (Seconds (81.0));
```

Host G pings H




```
//G pings H
V4PingHelper ping2 (netR5_Hinterfaces.GetAddress(1));
ping2.SetAttribute ("Interval", TimeValue (Seconds(1.0)));
ping2.SetAttribute ("Size", UintegerValue (1024));
ping2.SetAttribute ("Verbose", BooleanValue (pingVerbose));
ApplicationContainer ping2app = ping2.Install (net2.Get(3));
ping2app.Start (Seconds (1.0));
ping2app.Stop (Seconds (81.0));
```

Run the simulation for 90 sec.

```
Simulator::Stop (Seconds (90));
//Start the simulation
Simulator::Run ();
Simulator::Destroy ();
```

Pcap Analysis

Files: A4-capture-A-0.pcap A4-capture-H-0.pcap A4-capture-G-0.pcap A4-capture-K-0.pcap

# Packets	A	G	H	K
ICMP	143	146	132	126
ARP	4	4	4	4

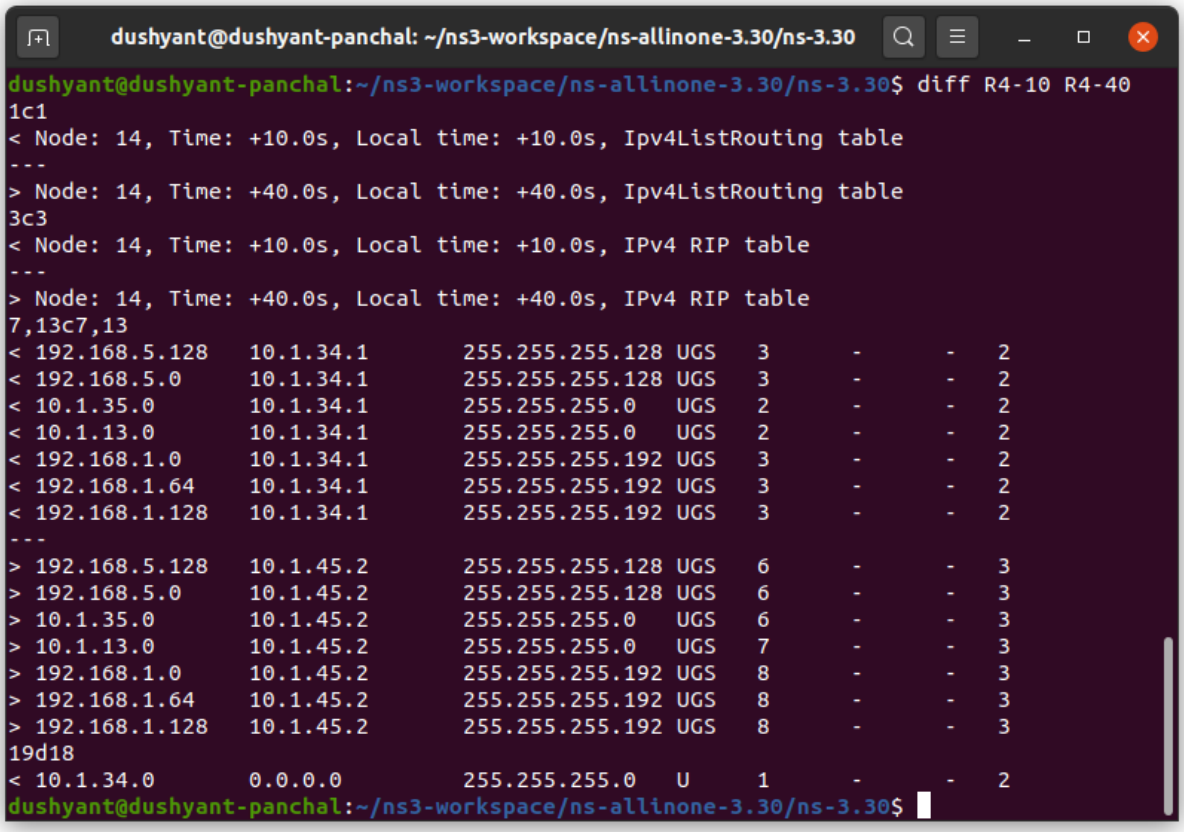
Routing Tables *(Node 11-16 ⇔ R1-R6)*

Stored in respective Files: routing-table-R1 routing-table-R3 routing-table-R5
routing-table-R2 routing-table-R4 routing-table-R6

Routing Table Changes for Router 4 (t=10, t=40, t=80)

Files Used: R4-10, R4-40, R4-80 (Contain Router 4 routing tables at t=10,40,80 resp)

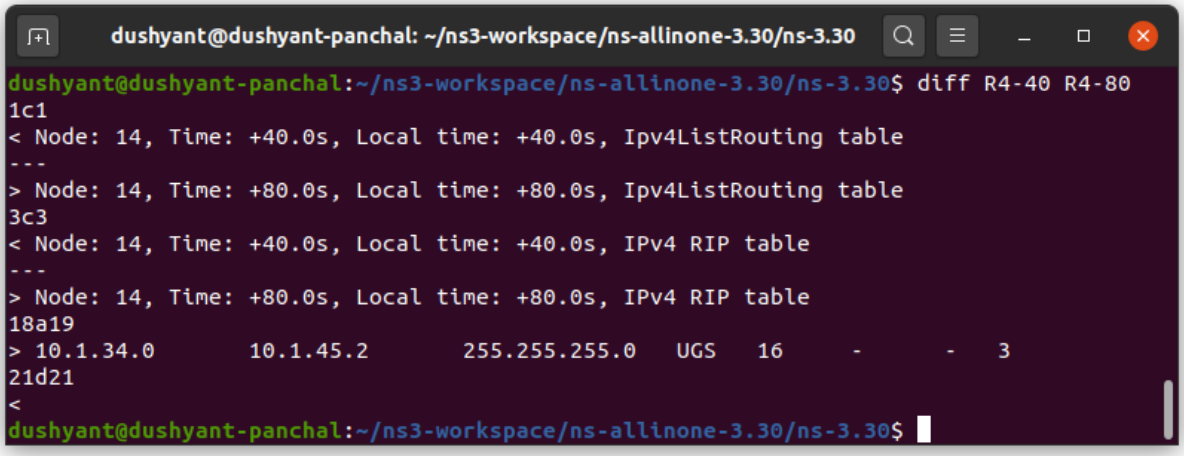
Difference between routing table at t=10 and t=40 for Router R4.



- All the table entries going through the link R3-R4 were re-routed through the R4-R5 link.

Before the R3-R4 link broke (t=10)	After the R3-R4 link broke (t=40)
Nodes H and I were reachable by R4 via R3 and R5 giving lower cost=3.	Nodes H and I are still reachable via R5 but with a higher cost path. (cost=6)
Nodes A, B, and C were reachable via R3 and R1 with cost =3.	Nodes A, B, and C need to take a longer path now via R5, R4, and R1 with cost=8.

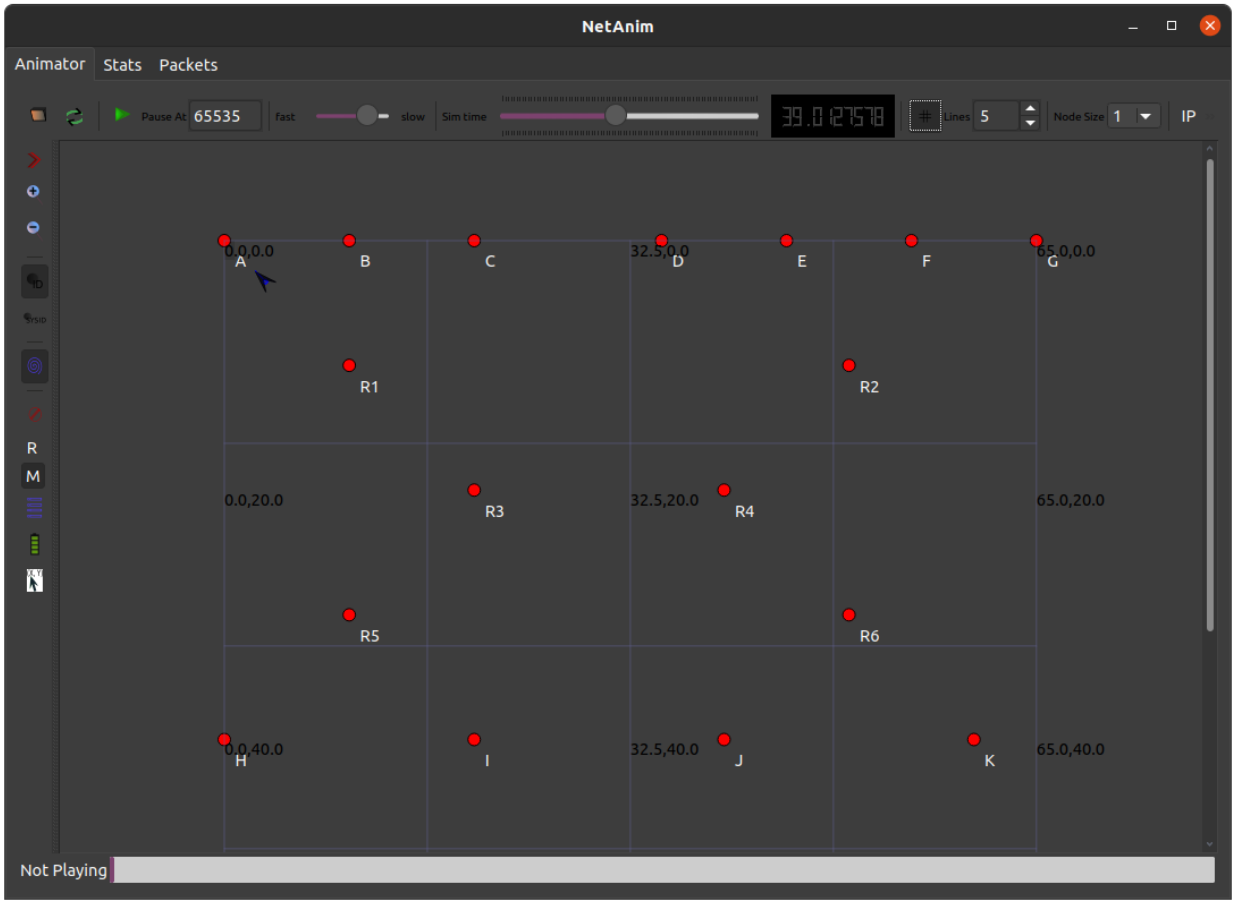
Difference between routing table at t=10 and t=40 for Router R4.



- The router R4 discovered (by t=40) that the R3-R4 network link is no more reachable, and eventually the entry for the network 10.1.34.0 was removed from its routing table.
- By t=80, a new path was discovered for the R4-R3 network, where R4 could reach R3 via R5, and hence a new entry was added to R4's routing table.

Visualization using NetAnim

File: animation.xml opened in NetAnim software



1. In the animation, we can, first of all, see the ARP requests and replies for creating IP to MAC mappings, populating the Routing tables through RIP, making the network scenario usable.
2. Next, we can see ICMP ping request and reply packets being exchanged between A and K as well as between G and H.
3. Before the R3-R4 link is broken, we can clearly see the paths taken by the ping packets.
   A pinging K takes the path: A→ R1→ R3→ R4→ R6→ K.
   G pinging H takes the path: G→ R2→ R4→ R3→ R5→ H.
   The paths are consistent with the routing tables (t=10).
4. After the R3-R4 link is broken, we can clearly see the packets from A getting lost at R3 while packets from G getting lost at R4, since the next-hop as per the existing entries in the routing tables becomes unreachable.
5. We also see that until somewhere around t=35, a new path is established through the R4-R5 link and the ping packets again start reaching their destinations.
   A pinging K now takes the path: A→ R1→ R3→ R5→ R4→ R6→ K.
   G pinging H now takes the path: G→ R2→ R4→ R5→ H.
   The paths are consistent with the new routing tables (t=40,80).