



Islamic University of Technology

CSE 4840

Internetworking Protocols Lab

Lab 2

Simulating and Understanding Routing in ns3

Name: Aashnan Rahman

ID: 190041204

Section 2B

February 24, 2024

Contents

1 Task	1
1.1 Objective	1
1.2 Topology Description:	1
1.3 Configuration Steps	1
2 Solution	2
2.1 Creating Object	2
2.2 Installing CSMA in Net Device Container	3
2.3 Excluding Interfaces from RIP Routing	3
2.4 Assigning IPv4 Address to the Interfaces	4
2.5 Configuring Static Routes	5
2.6 Link Teardown	6
2.7 Pinging Destination	7
3 Result	8
3.1 Ping Status	8
3.2 Routing Table	10
3.3 Ping Statistics	18
3.4 Result Summary	18
4 NetAnim Output	19

1 Task

1.1 Objective

Configure RIP routing protocols for a given network topology using ns-3. Monitor the routing tables at specified time intervals and observe the impact of link teardown events.

1.2 Topology Description:

Consider the network topology shown in Figure 1. Nodes n0, n1, n2, and n8 are hosts, while n3, n4, n5, n6, and n7 are routers. All networks have a cost of 1, except for the direct links from n5 to n7 (cost 5) and n6 to n7 (cost 10).

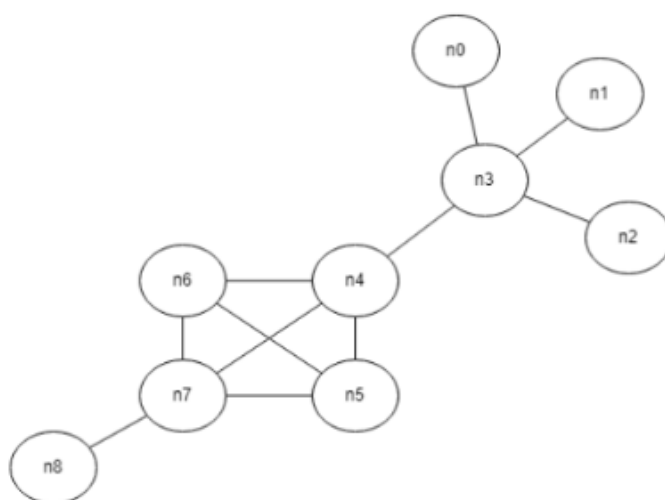


Figure 1: Network Topology

1.3 Configuration Steps

- Use the `ns3::RipHelper` class along with `ns3::Ipv4ListRoutingHelper` to configure RIP routing protocols for all routers.
- Set the costs for the direct links from n5 to n7 and n6 to n7 as mentioned.
- Simulate link teardown events: The link between n4 and n7 tears down at 30 seconds, and the link between n5 and n6 tears down at 50 seconds.
- Initiate ping messages from sources n0, n1, and n2 to destination n8.
- Simulate the network, and at 20, 40, 60, and 80 seconds, print the routing tables of all routers.
- All connected networks have different network addresses.

2 Solution

The code can be broken down into few segments –

2.1 Creating Object

```
Ptr<Node> n0 = CreateObject<Node>();
Names::Add("Hostn0", n0);
Ptr<Node> n1 = CreateObject<Node>();
Names::Add("Hostn1", n1);
Ptr<Node> n2 = CreateObject<Node>();
Names::Add("Hostn2", n2);

Ptr<Node> n3 = CreateObject<Node>();
Names::Add("Routern3", n3);
Ptr<Node> n4 = CreateObject<Node>();
Names::Add("Routern4", n4);
Ptr<Node> n5 = CreateObject<Node>();
Names::Add("Routern5", n5);
Ptr<Node> n6 = CreateObject<Node>();
Names::Add("Routern6", n6);
Ptr<Node> n7 = CreateObject<Node>();
Names::Add("Routern7", n7);

Ptr<Node> n8 = CreateObject<Node>();
Names::Add("Hostn8", n8);
```

This code segment initializes a network simulation in the ns-3 framework by creating nine nodes, each represented by a ‘Ptr <Node>’ smart pointer object. Nodes ‘n0’ to ‘n2’ are designated as hosts, while nodes ‘n3’ to ‘n7’ are labeled as routers. Another host node ‘n8’ is also created. Each node is given a unique name using the ‘Names::Add()’ function, allowing them to be referenced later in the simulation. This setup lays the foundation for configuring and connecting the nodes to simulate various network scenarios within the ns-3 environment.

2.2 Installing CSMA in Net Device Container

```
CsmaHelper csma;  
csma.SetChannelAttribute("DataRate", DataRateValue(5000000));  
csma.SetChannelAttribute("Delay", TimeValue(MilliSeconds(2)));  
NetDeviceContainer ndc1 = csma.Install(net1);  
NetDeviceContainer ndc2 = csma.Install(net2);  
NetDeviceContainer ndc3 = csma.Install(net3);  
NetDeviceContainer ndc4 = csma.Install(net4);  
NetDeviceContainer ndc5 = csma.Install(net5);  
NetDeviceContainer ndc6 = csma.Install(net6);  
NetDeviceContainer ndc7 = csma.Install(net7);  
NetDeviceContainer ndc8 = csma.Install(net8);  
NetDeviceContainer ndc9 = csma.Install(net9);  
NetDeviceContainer ndc10 = csma.Install(net10);  
NetDeviceContainer ndc11 = csma.Install(net11);
```

This code snippet configures a CSMA (Carrier Sense Multiple Access) network in the ns-3 simulation framework. It uses the ‘CsmaHelper’ class to set attributes for the channel, specifying a data rate of 5 Mbps and a delay of 2 milliseconds. Then, it installs CSMA net devices for each network container (‘net1’ to ‘net11’) using the ‘Install()’ method, creating separate ‘NetDeviceContainer’ objects (‘ndc1’ to ‘ndc11’) for each network. These net devices represent the communication interfaces between nodes within the simulated CSMA network, facilitating data transmission and reception with the specified channel attributes.

2.3 Excluding Interfaces from RIP Routing

```
RipHelper ripRouting;  
  
// Rule of thumb:  
// Interfaces are added sequentially, starting from 0  
// However, interface 0 is always the loopback...  
ripRouting.ExcludeInterface(n3, 1);  
ripRouting.ExcludeInterface(n3, 2);  
ripRouting.ExcludeInterface(n3, 3);  
ripRouting.ExcludeInterface(n7, 4);  
  
ripRouting.SetInterfaceMetric(n5, 3, 5);
```

```

ripRouting.SetInterfaceMetric(n6, 3, 10);

Ipv4ListRoutingHelper listRH;
listRH.Add(ripRouting, 0);10);
NetDeviceContainer ndc11 = csma.Install(net11);

```

In this code snippet, the ‘RipHelper’ class is used to configure the RIP (Routing Information Protocol) routing protocol for a network simulation in the ns-3 framework. Initially, the ‘ExcludeInterface()’ method is called multiple times to exclude specific interfaces from participating in RIP routing updates. For instance, interfaces 1, 2, and 3 of node n3, as well as interface 4 of node n7, are excluded from RIP routing. Additionally, the ‘SetInterfaceMetric()’ method is invoked to set custom metrics for specific interfaces. Here, the metric for interface 3 of nodes n5 and n6 is set to 5 and 10, respectively. Finally, the configured RIP routing settings are added to the ‘Ipv4ListRoutingHelper’ object ‘listRH’ using the ‘Add()’ method, specifying the RIP routing configuration for interface 0.

2.4 Assigning IPv4 Address to the Interfaces

```

Ipv4AddressHelper ipv4;

ipv4.SetBase(Ipv4Address("10.0.0.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic1 = ipv4.Assign(ndc1);

ipv4.SetBase(Ipv4Address("10.0.1.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic2 = ipv4.Assign(ndc2);

ipv4.SetBase(Ipv4Address("10.0.2.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic3 = ipv4.Assign(ndc3);

ipv4.SetBase(Ipv4Address("10.0.3.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic4 = ipv4.Assign(ndc4);

ipv4.SetBase(Ipv4Address("10.0.4.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic5 = ipv4.Assign(ndc5);

ipv4.SetBase(Ipv4Address("10.0.5.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic6 = ipv4.Assign(ndc6);

ipv4.SetBase(Ipv4Address("10.0.6.0"), Ipv4Mask("255.255.255.0"));

```

```

Ipv4InterfaceContainer iic7 = ipv4.Assign(ndc7);

ipv4.SetBase(Ipv4Address("10.0.7.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic8 = ipv4.Assign(ndc8);

ipv4.SetBase(Ipv4Address("10.0.8.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic9 = ipv4.Assign(ndc9);

ipv4.SetBase(Ipv4Address("10.0.9.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic10 = ipv4.Assign(ndc10);

ipv4.SetBase(Ipv4Address("10.0.10.0"), Ipv4Mask("255.255.255.0"));
Ipv4InterfaceContainer iic11 = ipv4.Assign(ndc11);

```

In this code snippet, the ‘Ipv4AddressHelper’ object ‘ipv4’ is used to assign IPv4 addresses to the network interfaces created by the ‘csma.Install()’ method for each subnet in the network. The ‘SetBase()’ method is called multiple times to set the base IP address and subnet mask for each subnet. Subsequently, the ‘Assign()’ method is invoked for each subnet, passing the corresponding ‘NetDeviceContainer’ object created by the ‘csma.Install()’ method. This assigns IPv4 addresses to the network interfaces within each subnet according to the specified base address and subnet mask. The resulting ‘Ipv4InterfaceContainer’ objects (‘iic1’ through ‘iic11’) store the assigned IPv4 addresses for each subnet’s interfaces, facilitating communication between nodes within the simulated network topology.

2.5 Configuring Static Routes

```

Ptr<Ipv4StaticRouting> staticRouting;

staticRouting = Ipv4RoutingHelper::GetRouting<Ipv4StaticRouting>(
    n0->GetObject<Ipv4>()->GetRoutingProtocol());
staticRouting->SetDefaultRoute("10.0.0.2", 1);

staticRouting = Ipv4RoutingHelper::GetRouting<Ipv4StaticRouting>(
    n1->GetObject<Ipv4>()->GetRoutingProtocol());
staticRouting->SetDefaultRoute("10.0.1.2", 1);

staticRouting = Ipv4RoutingHelper::GetRouting<Ipv4StaticRouting>(

```

```

        n2->GetObject<Ipv4>()->GetRoutingProtocol());
staticRouting->SetDefaultRoute("10.0.2.2", 1);

staticRouting = Ipv4RoutingHelper::GetRouting<Ipv4StaticRouting>(
    n8->GetObject<Ipv4>()->GetRoutingProtocol());
staticRouting->SetDefaultRoute("10.0.10.1", 1);

```

In this code snippet, static routes are configured for specific nodes in the simulated network topology. For each node ('n0', 'n1', 'n2', and 'n8'), the 'GetObject <Ipv4 >()->GetRoutingProtocol()' method is used to retrieve a pointer to the 'Ipv4StaticRouting' protocol module associated with the node's IPv4 protocol stack. This pointer is then assigned to the 'staticRouting' variable. Subsequently, the 'SetDefaultRoute()' method is called on the 'staticRouting' object to set a default route for each node. The first parameter of 'SetDefaultRoute()' specifies the next hop IP address for the default route, while the second parameter indicates the index of the outgoing interface through which packets should be forwarded. By configuring these static routes, the specified nodes are able to forward packets to the specified next-hop IP addresses via the designated outgoing interfaces, enabling routing within the simulated network topology.

2.6 Link Teardown

```

Simulator::Schedule(Seconds(30), &TearDownLink, n4, n7, 4, 1);
Simulator::Schedule(Seconds(50), &TearDownLink, n5, n6, 2, 2);

```

These lines of code utilize the 'Simulator::Schedule' method in NS-3 to set up events for tearing down network links between nodes at specified times during the simulation. In the first line, a link between nodes 'n4' and 'n7' is scheduled to be torn down at 30 seconds into the simulation, affecting interfaces 4 and 1 on nodes 'n7' and 'n4', respectively. Similarly, the second line schedules a link teardown event between nodes 'n5' and 'n6' at 50 seconds into the simulation, affecting interfaces 2 and 2 on nodes 'n6' and 'n5', respectively. These scheduled events facilitate the simulation of link failures and network disruptions at predetermined time points.

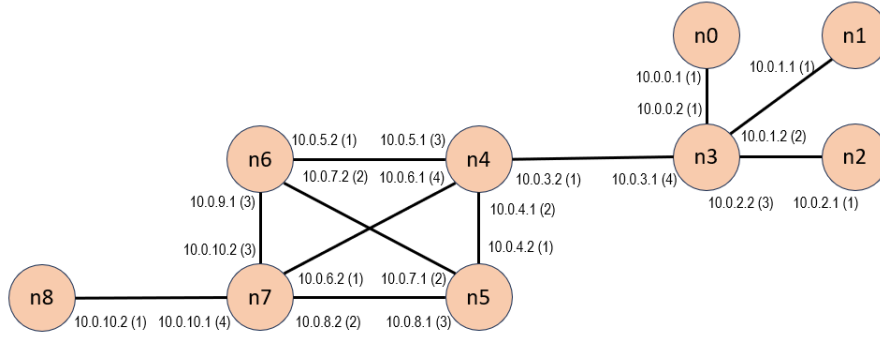


Figure 2: Enter Caption

2.7 Pinging Destination

```

NS_LOG_INFO("Create Applications.");
uint32_t packetSize = 1024;
Time interPacketInterval = Seconds(1.0);
V4PingHelper ping("10.0.10.2");

ping.SetAttribute("Interval", TimeValue(interPacketInterval));
ping.SetAttribute("Size", UIntegerValue(packetSize));
if (showPings)
{
    ping.SetAttribute("Verbose", BooleanValue(true));
}
ApplicationContainer apps = ping.Install(n0);
apps.Start(Seconds(1.0));
apps.Stop(Seconds(110.0));

```

This code segment configures a ping application to generate ICMP echo requests targeting the IPv4 address "10.0.10.2". The 'V4PingHelper' class is utilized to create the ping application, with attributes set for the packet interval ('Interval') and size ('Size'). Additionally, if the 'showPings' flag is set to true, verbose output is enabled for the ping application. The 'Install()' method is then used to install the ping application on node 'n0'. The ping application is scheduled to start at simulation time 1.0 seconds and stop at simulation time 110.0 seconds, as specified by the 'Start()' and 'Stop()' methods, respectively.

3 Result

3.1 Ping Status

```
1032 bytes from 10.0.10.2: icmp_seq=0 ttl=61 time=+57.5232ms
1032 bytes from 10.0.10.2: icmp_seq=1 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=2 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=3 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=4 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=5 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=6 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=7 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=8 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=9 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=10 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=11 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=12 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=13 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=14 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=15 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=16 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=17 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=18 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=19 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=20 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=21 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=22 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=23 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=24 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=25 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=26 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=27 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=28 ttl=61 time=+29.697ms
1032 bytes from 10.0.10.2: icmp_seq=42 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=43 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=44 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=45 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=46 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=47 ttl=60 time=+37.121ms
```

[illegible]

```

1032 bytes from 10.0.10.2: icmp_seq=86 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=87 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=88 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=89 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=90 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=91 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=92 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=93 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=94 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=95 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=96 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=97 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=98 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=99 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=100 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=101 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=102 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=103 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=104 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=105 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=106 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=107 ttl=60 time=+37.121ms
1032 bytes from 10.0.10.2: icmp_seq=108 ttl=60 time=+37.121ms

```

3.2 Routing Table

Node: 0, Time: +20s, Local time: +20s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Ipv4StaticRouting

Node: 0, Time: +20s, Local time: +20s, Ipv4StaticRouting table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	- 0
10.0.0.0	0.0.0.0	255.255.255.0	U	0	-	- 1
0.0.0.0	10.0.0.2	0.0.0.0	UGS	0	-	- 1

Priority: -10 Protocol: ns3::Ipv4GlobalRouting

Node: 0, Time: +20s, Local time: +20s, Ipv4GlobalRouting table

Node: 4, Time: +20s, Local time: +20s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 4, Time: +20s, Local time: +20s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.10.0	10.0.6.2	255.255.255.0	UGS	2	-	- 4
10.0.8.0	10.0.6.2	255.255.255.0	UGS	2	-	- 4
10.0.9.0	10.0.5.2	255.255.255.0	UGS	2	-	- 3
10.0.7.0	10.0.5.2	255.255.255.0	UGS	2	-	- 3
10.0.2.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.1.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	- 3
10.0.6.0	0.0.0.0	255.255.255.0	U	1	-	- 4

Node: 5, Time: +20s, Local time: +20s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 5, Time: +20s, Local time: +20s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.6.0	10.0.4.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	10.0.4.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.10.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.9.0	10.0.7.2	255.255.255.0	UGS	2	-	- 2
10.0.5.0	10.0.7.2	255.255.255.0	UGS	2	-	- 2
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.7.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 6, Time: +20s, Local time: +20s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 6, Time: +20s, Local time: +20s, IPv4 RIP table

Destination ↪ Iface	Gateway	Genmask	Flags	Metric	Ref	Use
10.0.6.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.4.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.8.0	10.0.7.1	255.255.255.0	UGS	2	-	- 2
10.0.10.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.7.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 7, Time: +20s, Local time: +20s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 7, Time: +20s, Local time: +20s, IPv4 RIP table

Destination ↪ Iface	Gateway	Genmask	Flags	Metric	Ref	Use
10.0.4.0	10.0.6.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	10.0.6.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.6.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.6.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.6.1	255.255.255.0	UGS	3	-	- 1
10.0.7.0	10.0.9.1	255.255.255.0	UGS	2	-	- 3
10.0.5.0	10.0.9.1	255.255.255.0	UGS	2	-	- 3
10.0.6.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	- 3
10.0.10.0	0.0.0.0	255.255.255.0	U	1	-	- 4

Node: 0, Time: +40s, Local time: +40s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Ipv4StaticRouting

Node: 0, Time: +40s, Local time: +40s, Ipv4StaticRouting table

Destination ↪ Iface	Gateway	Genmask	Flags	Metric	Ref	Use
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	- 0
10.0.0.0	0.0.0.0	255.255.255.0	U	0	-	- 1

```
0.0.0.0      10.0.0.2      0.0.0.0      UGS    0      -      -      1
```

Priority: -10 Protocol: ns3::Ipv4GlobalRouting

Node: 0, Time: +40s, Local time: +40s, Ipv4GlobalRouting table

Node: 4, Time: +40s, Local time: +40s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 4, Time: +40s, Local time: +40s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.10.0	10.0.4.2	255.255.255.0	UGS	7	-	- 2
10.0.9.0	10.0.5.2	255.255.255.0	UGS	2	-	- 3
10.0.7.0	10.0.5.2	255.255.255.0	UGS	2	-	- 3
10.0.2.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.1.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 5, Time: +40s, Local time: +40s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 5, Time: +40s, Local time: +40s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.3.0	10.0.4.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.10.0	10.0.8.2	255.255.255.0	UGS	6	-	- 3
10.0.9.0	10.0.7.2	255.255.255.0	UGS	2	-	- 2
10.0.5.0	10.0.7.2	255.255.255.0	UGS	2	-	- 2
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.7.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 6, Time: +40s, Local time: +40s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 6, Time: +40s, Local time: +40s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.4.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.8.0	10.0.7.1	255.255.255.0	UGS	2	-	- 2
10.0.10.0	10.0.7.1	255.255.255.0	UGS	7	-	- 2
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.7.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 7, Time: +40s, Local time: +40s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 7, Time: +40s, Local time: +40s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.7.0	10.0.9.1	255.255.255.0	UGS	2	-	- 3
10.0.5.0	10.0.9.1	255.255.255.0	UGS	2	-	- 3
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	- 3
10.0.10.0	0.0.0.0	255.255.255.0	U	1	-	- 4

Node: 0, Time: +60s, Local time: +60s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Ipv4StaticRouting

Node: 0, Time: +60s, Local time: +60s, Ipv4StaticRouting table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	- 0
10.0.0.0	0.0.0.0	255.255.255.0	U	0	-	- 1
0.0.0.0	10.0.0.2	0.0.0.0	UGS	0	-	- 1

Priority: -10 Protocol: ns3::Ipv4GlobalRouting

Node: 0, Time: +60s, Local time: +60s, Ipv4GlobalRouting table

Node: 4, Time: +60s, Local time: +60s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 4, Time: +60s, Local time: +60s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.10.0	10.0.4.2	255.255.255.0	UGS	7	-	- 2
10.0.8.0	10.0.4.2	255.255.255.0	UGS	2	-	- 2
10.0.9.0	10.0.5.2	255.255.255.0	UGS	2	-	- 3
10.0.2.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.1.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 5, Time: +60s, Local time: +60s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 5, Time: +60s, Local time: +60s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.3.0	10.0.4.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.10.0	10.0.8.2	255.255.255.0	UGS	6	-	- 3
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 6, Time: +60s, Local time: +60s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 6, Time: +60s, Local time: +60s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.4.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	- 1

10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	-	3
----------	---------	---------------	---	---	---	---	---

Node: 7, Time: +60s, Local time: +60s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 7, Time: +60s, Local time: +60s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.4.0	10.0.8.1	255.255.255.0	UGS	2	-	- 2
10.0.3.0	10.0.9.1	255.255.255.0	UGS	3	-	- 3
10.0.0.0	10.0.9.1	255.255.255.0	UGS	4	-	- 3
10.0.1.0	10.0.9.1	255.255.255.0	UGS	4	-	- 3
10.0.2.0	10.0.9.1	255.255.255.0	UGS	4	-	- 3
10.0.5.0	10.0.9.1	255.255.255.0	UGS	2	-	- 3
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	- 3
10.0.10.0	0.0.0.0	255.255.255.0	U	1	-	- 4

Node: 0, Time: +80s, Local time: +80s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Ipv4StaticRouting

Node: 0, Time: +80s, Local time: +80s, Ipv4StaticRouting table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
127.0.0.0	0.0.0.0	255.0.0.0	U	0	-	- 0
10.0.0.0	0.0.0.0	255.255.255.0	U	0	-	- 1
0.0.0.0	10.0.0.2	0.0.0.0	UGS	0	-	- 1

Priority: -10 Protocol: ns3::Ipv4GlobalRouting

Node: 0, Time: +80s, Local time: +80s, Ipv4GlobalRouting table

Node: 4, Time: +80s, Local time: +80s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 4, Time: +80s, Local time: +80s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.10.0	10.0.4.2	255.255.255.0	UGS	7	-	- 2
10.0.8.0	10.0.4.2	255.255.255.0	UGS	2	-	- 2
10.0.9.0	10.0.5.2	255.255.255.0	UGS	2	-	- 3
10.0.2.0	10.0.3.1	255.255.255.0	UGS	2	-	- 1

10.0.1.0	10.0.3.1	255.255.255.0	UGS	2	-	-	1
10.0.0.0	10.0.3.1	255.255.255.0	UGS	2	-	-	1
10.0.3.0	0.0.0.0	255.255.255.0	U	1	-	-	1
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	-	2
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	-	3
10.0.6.0	10.0.3.1	255.255.255.0	UGS	16	-	-	1

Node: 5, Time: +80s, Local time: +80s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 5, Time: +80s, Local time: +80s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.3.0	10.0.4.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.10.0	10.0.8.2	255.255.255.0	UGS	6	-	- 3
10.0.9.0	10.0.4.1	255.255.255.0	UGS	3	-	- 1
10.0.5.0	10.0.4.1	255.255.255.0	UGS	2	-	- 1
10.0.4.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 6, Time: +80s, Local time: +80s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 6, Time: +80s, Local time: +80s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.4.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.3.0	10.0.5.1	255.255.255.0	UGS	2	-	- 1
10.0.0.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.1.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.2.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.8.0	10.0.5.1	255.255.255.0	UGS	3	-	- 1
10.0.10.0	10.0.5.1	255.255.255.0	UGS	8	-	- 1
10.0.5.0	0.0.0.0	255.255.255.0	U	1	-	- 1
10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	- 3

Node: 7, Time: +80s, Local time: +80s, Ipv4ListRouting table

Priority: 0 Protocol: ns3::Rip

Node: 7, Time: +80s, Local time: +80s, IPv4 RIP table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
↪ Iface						
10.0.4.0	10.0.8.1	255.255.255.0	UGS	2	-	- 2
10.0.3.0	10.0.9.1	255.255.255.0	UGS	3	-	- 3
10.0.0.0	10.0.9.1	255.255.255.0	UGS	4	-	- 3
10.0.1.0	10.0.9.1	255.255.255.0	UGS	4	-	- 3
10.0.2.0	10.0.9.1	255.255.255.0	UGS	4	-	- 3
10.0.5.0	10.0.9.1	255.255.255.0	UGS	2	-	- 3
10.0.6.0	10.0.8.1	255.255.255.0	UGS	16	-	- 2
10.0.8.0	0.0.0.0	255.255.255.0	U	1	-	- 2
10.0.9.0	0.0.0.0	255.255.255.0	U	1	-	- 3
10.0.10.0	0.0.0.0	255.255.255.0	U	1	-	- 4

3.3 Ping Statistics

--- 10.0.10.2 ping statistics ---

109 packets transmitted, 96 received, 11% packet loss, time +1.09e+05ms
rtt min/avg/max/mdev = 29/34.87/57/4.302 ms

3.4 Result Summary

Node 8 (10.0.10.2) was successfully pinged from Node 0 (10.0.0.1).

4 NetAnim Output

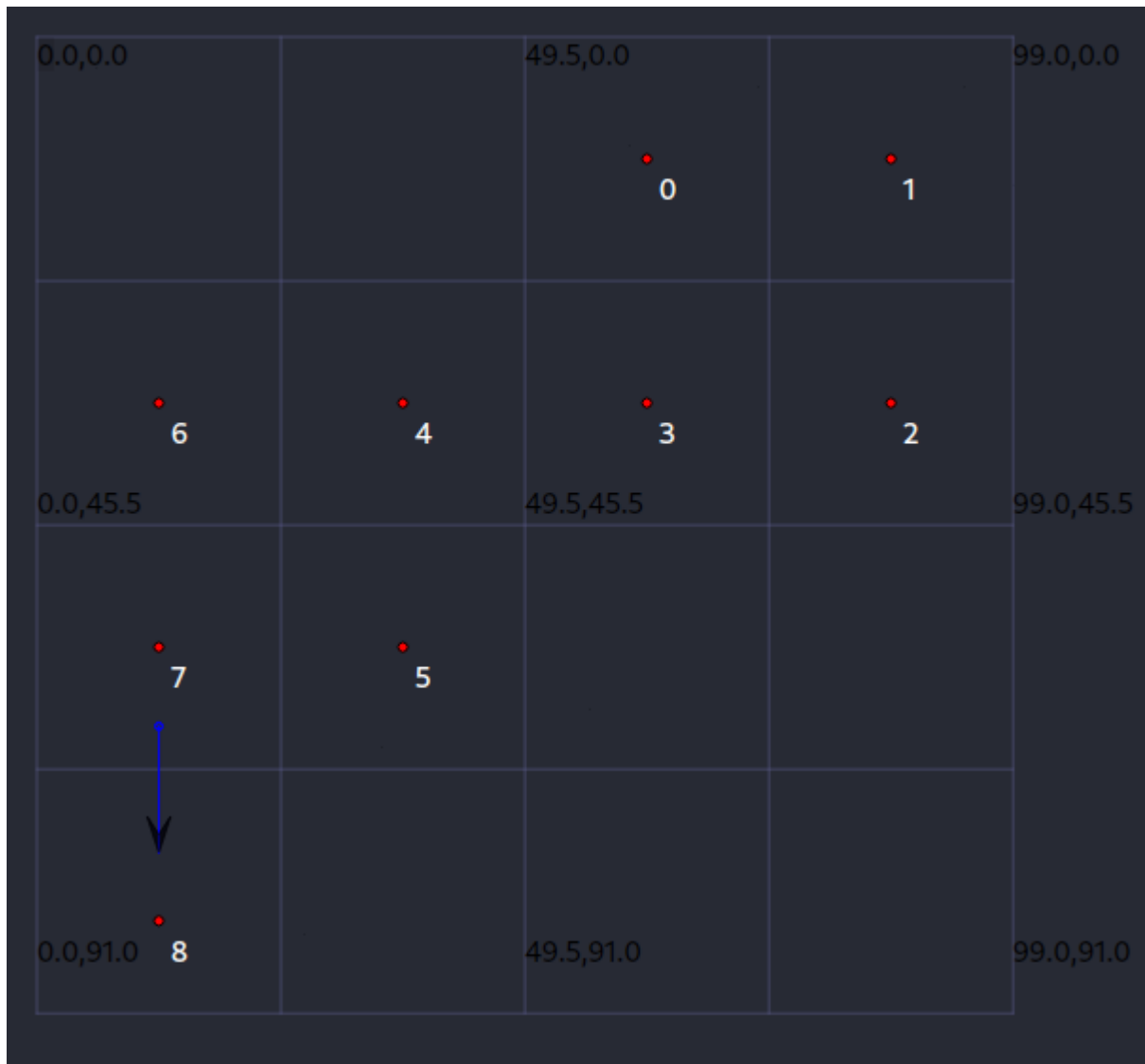


Figure 3: NetAnim

Node 8 was successfully pinged from Node 0.