

Software Defined Networks

Contents

1	Introduction.....	2
	1.1 SDN in NetSim.....	2
2	Model Features.....	3
	2.1 SDN Controller.....	3
	2.2 SDN Supported Commands.....	3
3	Featured Examples.....	5
	3.1 Configuring SDN in NetSim.....	5
	3.2 How to Change the IP tables in devices in NetSim using SDN Commands.....	8

1 Introduction

Software-Defined Networking (SDN) is an emerging architecture that decouples the network control and forwarding functions, enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

The SDN architecture is:

- Directly programmable: Network control is directly programmable because it is decoupled from forwarding functions.
- Agile: Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.
- Centrally managed: Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single logical switch.
- Programmatically configured: SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.
- Open standards-based and vendor-neutral: When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

1.1 SDN in NetSim

Software Defined Networking (SDN) module is featured from NetSim v11 onwards. This module features an SDN controller which can be used to control packet forwarding of all Layer 3 devices in the Network.

NetSim SDN is Open Flow Compatible. OpenFlow is an open interface for remotely controlling the forwarding tables in network switches, routers, and access points. Upon this low-level primitive, researchers can build networks with new high-level properties

2 Model Features

2.1 SDN Controller

An SDN controller is an application in software-defined networking (SDN) that manages flow control to enable intelligent networking. SDN controllers are based on protocols, such as OpenFlow, that allow servers to tell switches where to send packets.

The controller is the core of an SDN network. It lies between network devices at one end and applications at the other end. Any communications between applications and devices have to go through the controller. The controller also uses protocols such as OpenFlow to configure network devices and choose the optimal network path for application traffic.

In NetSim, any Layer 3 device can be configured to be as SDN Controller

1. Internetworks – Nodes (Wired, Wireless Node), L3 Switches, Routers
2. MANETs – Wireless Nodes
3. WSN / IoT - Sensors and Gateway (Sink Node in WSN, LowPAN Gateway in IoT)
4. LTE - LTE MME
5. VANETs - VANET nodes

In NetSim, the SDN controller features a command line interface through which users can configure properties such as the IP forwarding table for different devices

2.2 SDN Supported Commands

Note: Commands are not a case sensitive

To get detailed help of command syntax usage, please refer section 2.4.8 Interactive simulation commands

1. Commands Simulation specific

- Pause
- PauseAt
- Continue
- Stop
- Exit

- Reconnect

2. Route Commands

- route add
- route print
- route delete

3. Ping Command

- ping

4. ACL Configuration Commands

- ACL Enable
- ACL Disable
- ACL Print
- aclconfig

How to Use SDN Commands:

Use SDN commands in SDN Controller nodes to control other nodes in the network. Use the Node Names with Device_ID < **Wired_Node_2** > then follows SDN commands

For Example:

- In order to view the IP routing table of any node (e.g., Wired_Node_2), use following command

<DeviceName with Device_ID> route print

e.g. Wired_Node_2 route print

```

C:\Program Files\NetSim Pro\bin\NetSimCL.exe
NetSim>Wired_Node_2 route print
Input is validated
Sending command to client device 2
-----
IP Route Table
-----
Type      Network Destination  Netmask/Prefix          Gateway                Interface              Metric
-----
LOCAL    11.3.0.0             255.255.0.0             on-link                11.3.1.2                300
MULTICAST 224.0.0.1           255.255.255.255        on-link                11.3.1.2                306
MULTICAST 224.0.0.0            240.0.0.0              on-link                11.3.1.2                306
BROADCAST 255.255.255.255     255.255.255.255        on-link                11.3.1.2                999
DEFAULT  0.0.0.0              0.0.0.0                11.3.1.1              11.3.1.2                999
-----

```

- Use **ping** cmd with an device name and IP address

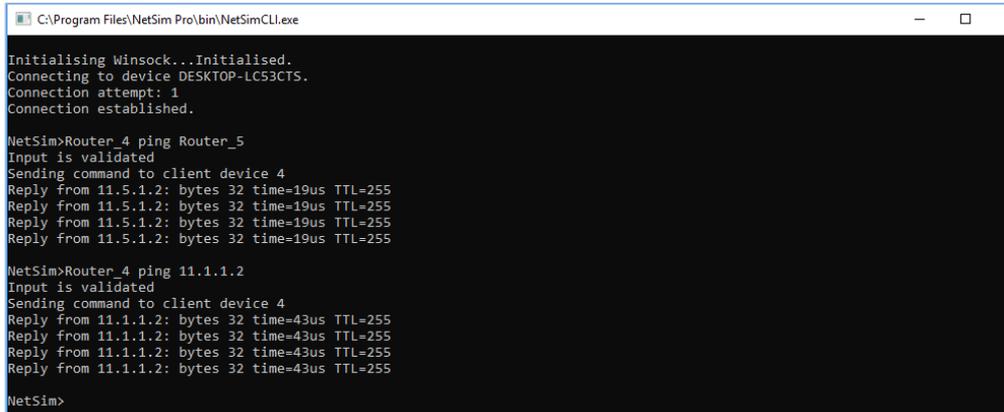
Note: ICMP_Status should be set as True in respective nodes for ping to work

<DeviceName with Device_ID> Ping <DeviceName with Device_ID >

<DeviceName with Device_ID> Ping < IP Address >

e.g. Router_4 ping Router_5

Router_4 ping 11.1.1.2



```
C:\Program Files\NetSim Pro\bin\NetSimCLI.exe
Initialising Winsock...Initialised.
Connecting to device DESKTOP-LC53CTS.
Connection attempt: 1
Connection established.

NetSim>Router_4 ping Router_5
Input is validated
Sending command to client device 4
Reply from 11.5.1.2: bytes 32 time=19us TTL=255

NetSim>Router_4 ping 11.1.1.2
Input is validated
Sending command to client device 4
Reply from 11.1.1.2: bytes 32 time=43us TTL=255

NetSim>
```

3 Featured Examples

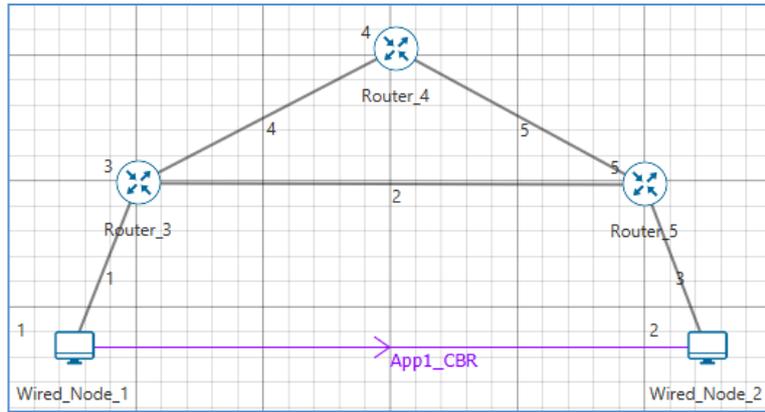
Sample configuration files for all networks are available in Examples Menu in NetSim Home Screen. These files provide examples on how NetSim can be used – the parameters that can be changed and the typical effect it has on performance.

3.1 Configuring SDN in NetSim

This section will demonstrate how to configure SDN controller in a simple network scenario.

Sample configuration files for all networks are available in Examples Menu in NetSim Home Screen. These files provide examples on how NetSim can be used – the parameters that can be changed and the typical effect it has on performance.

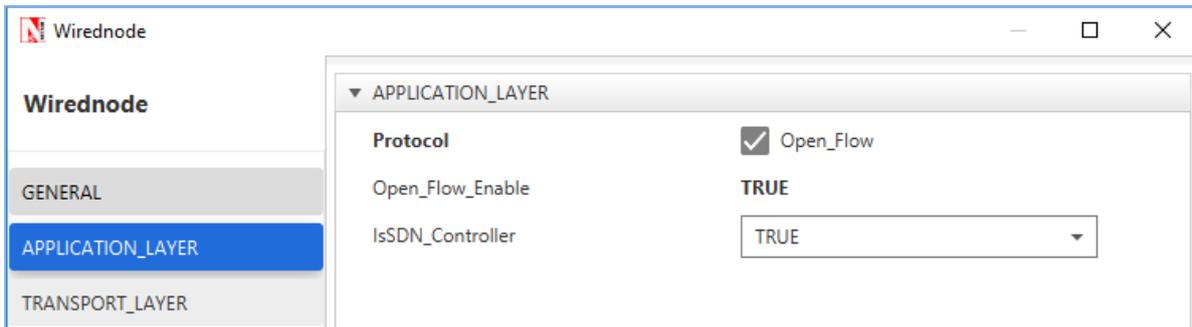
Open NetSim, Select Examples->Software-Defined-Networks->SDN-Internetworks



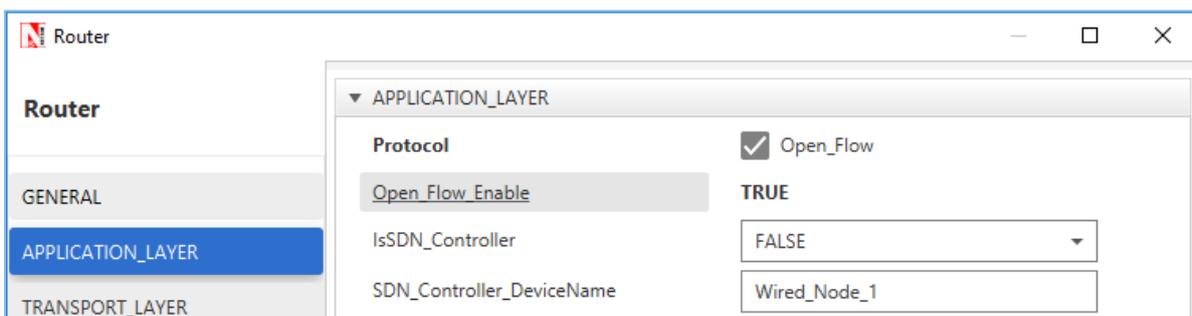
Setting the SDN controller:

In Transport Layer TCP should be enabled in all nodes. Any Layer_3 device can be a SDN Controller and a network can contain atmost one SDN controller.

For example, let us set Wired_Node_1 as SDN Controller. To set go to application layer properties of Wired_Node_1 and set IsSDN_Controller as TRUE and Enable Protocol as Open_Flow.



Once a SDN Controller is configured, go to other nodes (Router_3, 4, 5 and Wired_Node_2) and set Protocol as Open_Flow and enter SDN Controller's device name in SDN_Controller_DeviceName field. IN our example we enter Wired_Node_1 in this field as shown below

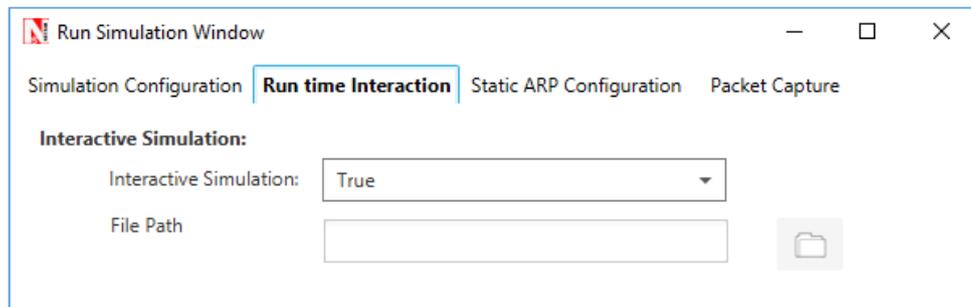


The setting for all the devices in the network would be as follows:

Device Name	Open_Flow	IsSDN_Controller	SDN_Controller_Device_Name
Wired_Node_1	Enable	TRUE	---
Router_3	Enable	FALSE	Wired_Node_1
Router_4	Enable	FALSE	Wired_Node_1
Router_5	Enable	FALSE	Wired_Node_1
Wired_Node_2	Enable	FALSE	Wired_Node_1

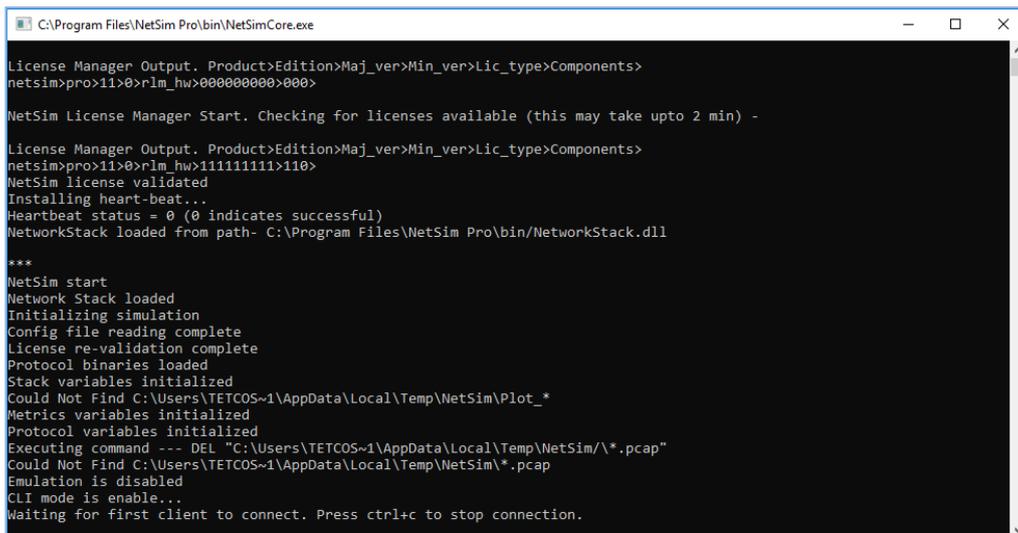
Click on Application icon present in the top ribbon and set the Application type as CBR. The Source_Id is 1 and Destination_Id is 2.

Enable plots, packet and event trace options. Click on run simulation and set Simulation Time as 200 sec. In the Run time Interaction tab set Interactive Simulation as True.

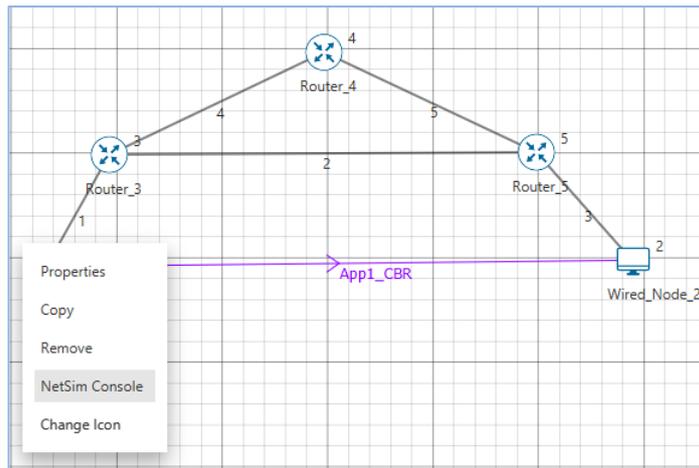


Click on Accept. And then click on OK to run the simulation.

Simulation (NetSimCore.exe) will start running and will display a message “waiting for first client to connect” as shown below:



Now go to network scenario right click on Wired_Node_1 (SDN_Controller) and select NetSim Console



Now Client (NetSimCLI.exe) will start running and it will try to establish a connection with NetSimCore.exe. After the connection is established the window will look like

```

C:\Program Files\NetSim Pro\bin\NetSimCLI.exe
Initialising Winsock...Initialised.
Connecting to device DESKTOP-LC53CTS.
Connection attempt: 1
Connection established.

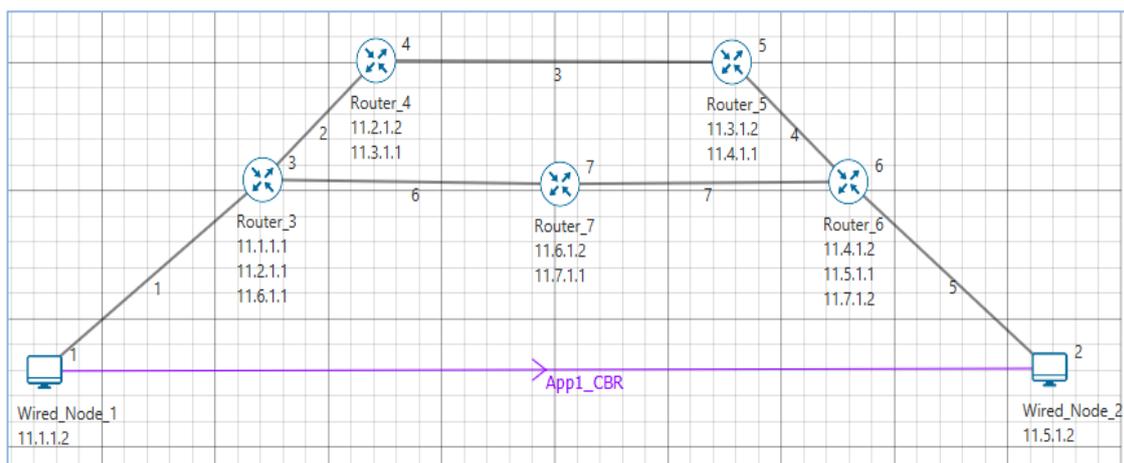
NetSim>

```

After this the command line interface can be used to execute the supported SDN commands.

3.2 How to Change the IP tables in devices in NetSim using SDN Commands

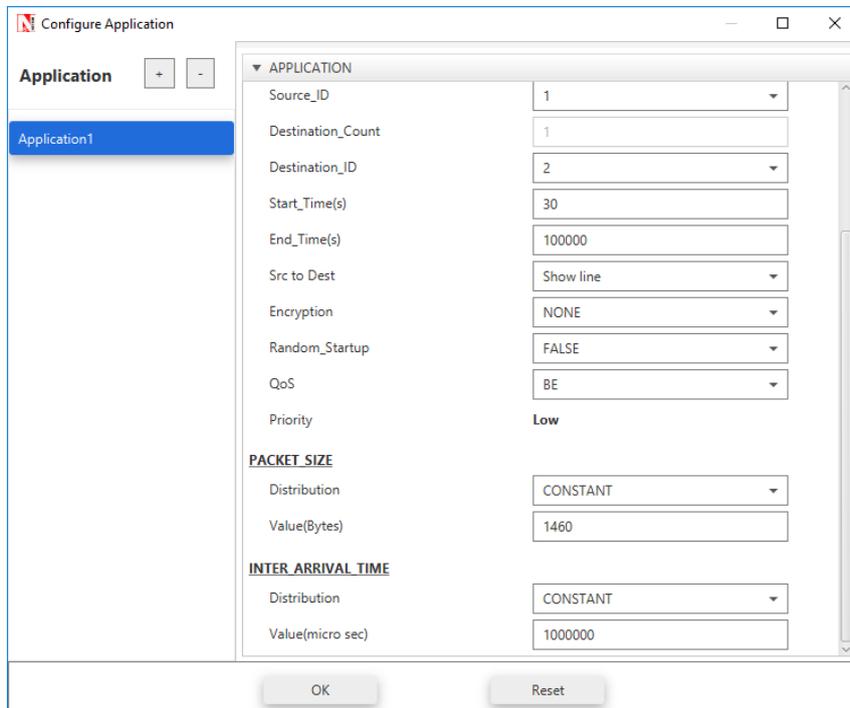
Open NetSim, Select Examples->Internetworks->Software-Defined-Networks->SDN-StaticRoute



Settings done for this sample experiment:

- Set Application type as CBR Source_Id as 1 and Destination_Id as 2

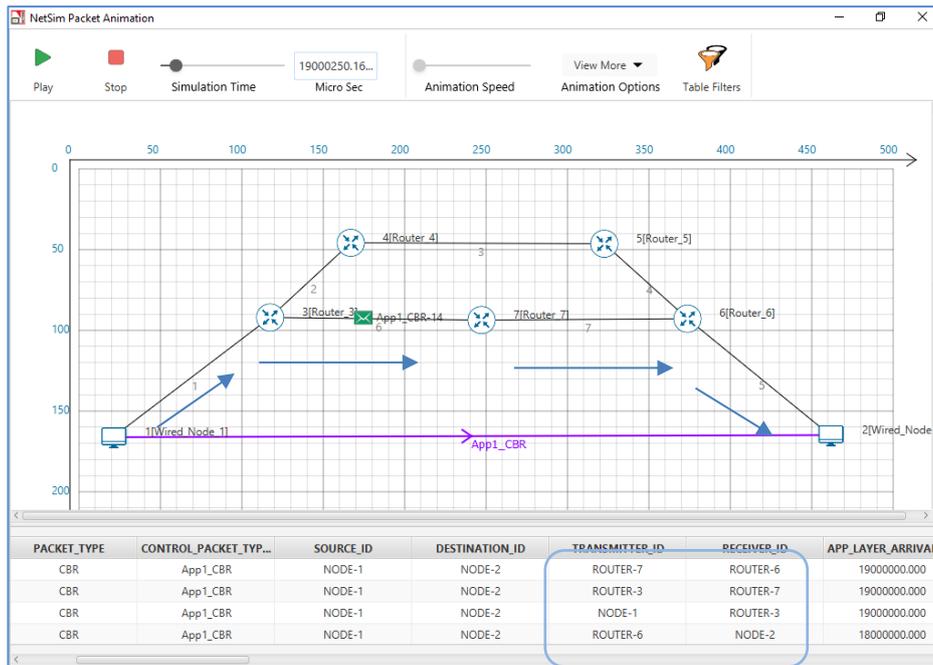
- Set Start Time as 30 sec and Inter Arrival Time as 1 Sec (1000000 microsec) as shown below:



- Enable Plots, Packet and event trace option

Please refer the Section 3.1: Configuring SDN in NetSim (In this example SDN Controller node is WIRED_NODE_1) after setting SDN Controller. Click on run simulation and set Simulation Time as 500 sec.

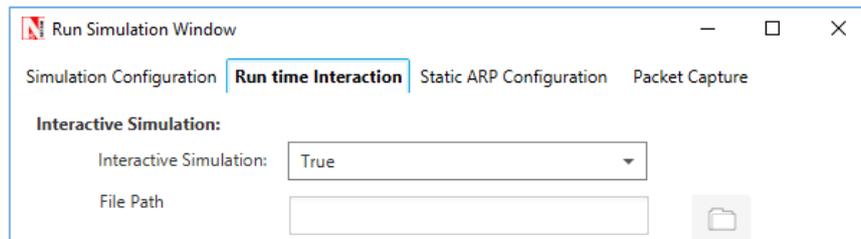
Check packet animation. The default routing protocol is OSPF. So, packets will reach destination via Router_7 (Refer related experiment in experiment manual for more information). In results window the Network Metrics_Table displays no data packet transmissions in Links 2, 3, and 4.



Configuring Routes:

You can configure a route on the router. For example will set up a route as follows Router_3 → Router_4 → Router_5 → Router_6.

Re-run same scenario in interactive simulation mode (In Run time Interaction tab set Interactive Simulation as True)



Run simulation for 500 seconds. Now go to network scenario right click on WIRED_NODE_1 (SDN_Controller) and select NetSim Console.

Commands to configure Static Routes:

- To view IP Routing table use command: **Router_3 route print**
- To add a route use **route add** command

```
route ADD destination_ip MASK subnet_mask gateway_ip metric interface
```

- So, for example, if you want to add a route specifying that all traffic bound for the 11.4.1.2 subnet will go to a gateway 11.2.1.2

```
route add 11.4.1.2 mask 255.255.255.255 11.2.1.2 metric 1 if 2
```

```
NetSim> ROUTER_3 route add 11.4.1.2 mask 255.255.255.255 11.2.1.2 metric 1 if 2
```

```
NetSim>ROUTER_3 route add 11.5.1.1 mask 255.255.255.255 11.2.1.2 metric 1 if 2
```

```
NetSim>ROUTER_3 route add 11.7.1.2 mask 255.255.255.255 11.2.1.2 metric 1 if 2
```

```
NetSim>ROUTER_3 route print
```

```

C:\Program Files\NetSim Pro\bin\NetSimCLI.exe
NetSim>ROUTER_3 route add 11.4.1.2 mask 255.255.255.255 11.2.1.2 metric 1 if 2
Input is validated
Command is for SDN Controller

NetSim>ROUTER_3 route add 11.5.1.1 mask 255.255.255.255 11.2.1.2 metric 1 if 2
Input is validated
Command is for SDN Controller

NetSim>ROUTER_3 route add 11.7.1.2 mask 255.255.255.255 11.2.1.2 metric 1 if 2
Input is validated
Command is for SDN Controller

NetSim>ROUTE PRINT
=====
IP Route Table
=====

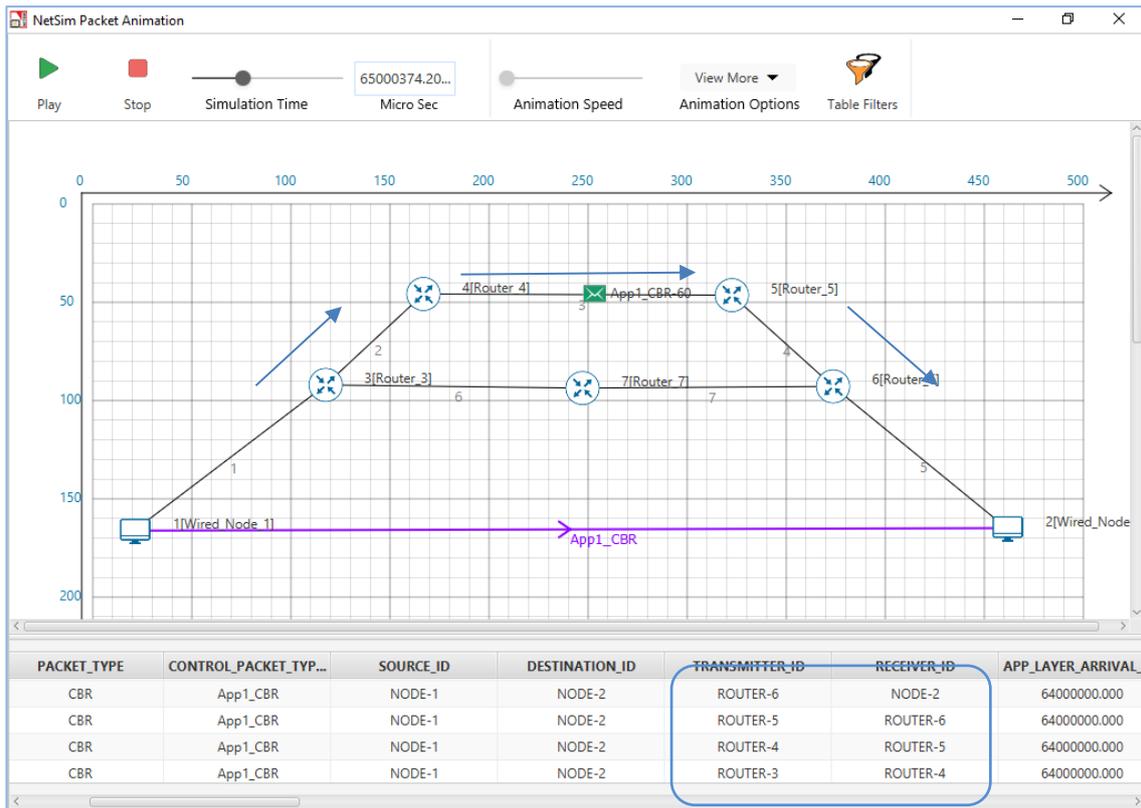
```

Type	Network	Destination	Netmask//Prefix	Gateway	Interface	Metric
STATIC	11.7.1.2		255.255.255.255	11.2.1.2	11.2.1.1	1
STATIC	11.5.1.1		255.255.255.255	11.2.1.2	11.2.1.1	1
STATIC	11.4.1.2		255.255.255.255	11.2.1.2	11.2.1.1	1
OSPF	11.2.1.2		255.255.0.0	11.2.1.2	11.2.1.1	200
OSPF	11.3.1.1		255.255.0.0	11.2.1.2	11.2.1.1	200
OSPF	11.6.1.2		255.255.0.0	11.6.1.2	11.6.1.1	200

After configuring the static routes, wait till end of the simulation and check packet animation now the packets will reach destination as per the static routes:

Wired_Node_1 → Router_3 → Router_4 → Router_5 → Router_6 → Wired_Node_2

Note: Packets may initially flow via Router_7 and routes will change via Router_4 after static routes are configured.



In results window the Network Metrics_Table displays data packet transmissions in Links 2, 3, and 4.

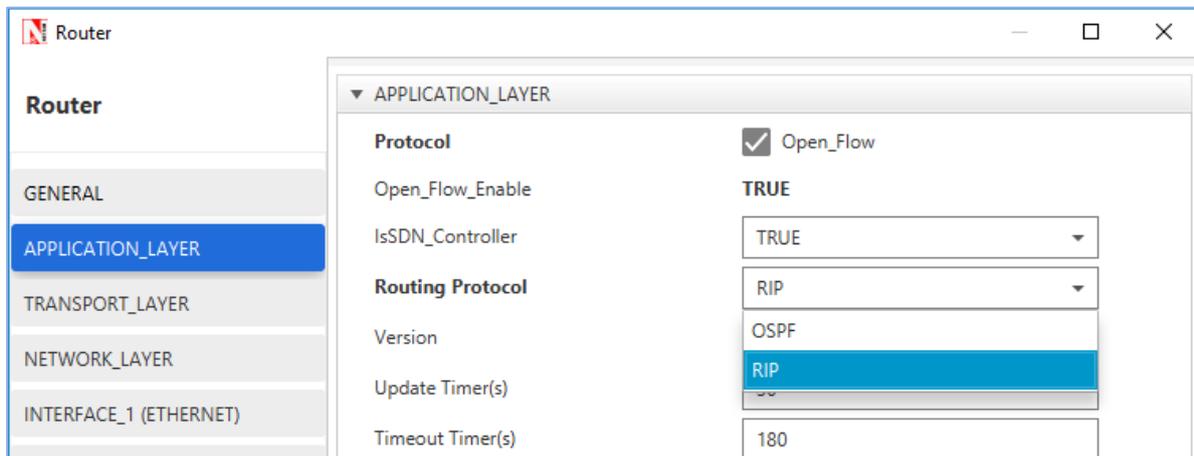
Link_id	Link_throughput_plot	Packet_transmi...		Packet_errored		Packet_collided		Bytes_transmitted(bytes)	Payload_transmitted(bytes)	Overhead_transmitted(bytes)
		Data	Control	Data	Control	Data	Control			
All	NA	903	297	3	1	0	0	1377140	1314000	63140
1	Link_throughput	194	0	0	0	0	0	293716	283240	10476
2	Link_throughput	157	61	0	0	0	0	238444	229220	9224
3	Link_throughput	157	62	0	0	0	0	238580	229220	9360
4	Link_throughput	157	57	0	1	0	0	238140	229220	8920
5	Link_throughput	184	0	2	0	0	0	278576	265720	12856
6	Link_throughput	27	59	0	0	0	0	44800	39420	5380
7	Link_throughput	27	58	1	0	0	0	44884	37960	6924

Open Packet trace option from results window and filter CONTROL_PACKET_TYPE column to OPENFLOW_COMMAND and OPENFLOW_RESPONSE and users can see that OpenFlow packets between WiredNode_1 to Router_3

	A	B	C	D	E	F	G	H
1	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
97	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-3	NODE-1	ROUTER-3
99	0	0	Control_Packet	OPENFLOW_RESPONSE	ROUTER-3	NODE-1	ROUTER-3	NODE-1
424	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-3	NODE-1	ROUTER-3
890	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-3	NODE-1	ROUTER-3
892	0	0	Control_Packet	OPENFLOW_RESPONSE	ROUTER-3	NODE-1	ROUTER-3	NODE-1
1372	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-3	NODE-1	ROUTER-3
1556	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-3	NODE-1	ROUTER-3
1558	0	0	Control_Packet	OPENFLOW_RESPONSE	ROUTER-3	NODE-1	ROUTER-3	NODE-1
1848	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-4	NODE-1	ROUTER-3
1849	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-4	ROUTER-3	ROUTER-4
1852	0	0	Control_Packet	OPENFLOW_RESPONSE	ROUTER-4	NODE-1	ROUTER-4	ROUTER-3
1853	0	0	Control_Packet	OPENFLOW_RESPONSE	ROUTER-4	NODE-1	ROUTER-3	NODE-1
1928	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-4	NODE-1	ROUTER-3
1929	0	0	Control_Packet	OPENFLOW_COMMAND	NODE-1	ROUTER-4	ROUTER-3	ROUTER-4

Running simulation using RIP Routing Protocol:

In this network scenario created about, Right click on any router and in application layer set Routing protocol as RIP

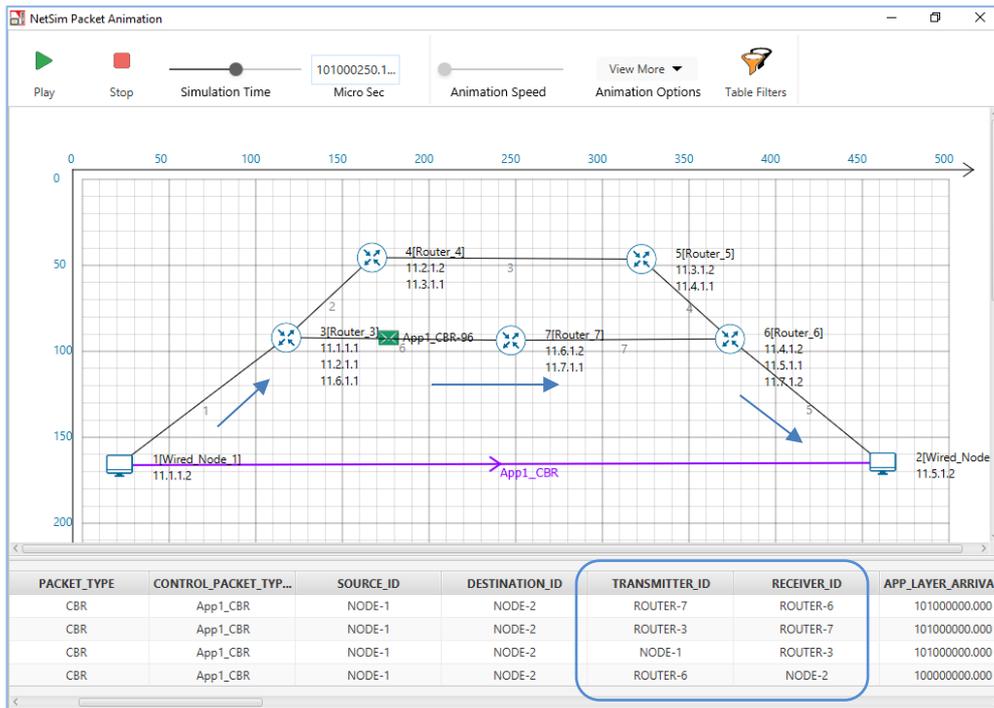


Commands to configure Static Routes (SDN Controller: Router_3):

route add 11.4.1.2 mask 255.255.255.255 11.6.1.2 metric 1 if 3

route add 11.5.1.1 mask 255.255.255.255 11.6.1.2 metric 1 if 3

route add 11.7.1.2 mask 255.255.255.255 11.6.1.2 metric 1 if 3



After configuring the static routes, check packet animation. Now the packets will reach destination as per the static routes: Router_3 → Router_7 → Router_6.