NetSim®

Accelerate Network R & D

Software Defined Networks

A Network Simulation & Emulation Software

By

TETCOS LLP

The information contained in this document represents the current view of TETCOS LLP on the issues discussed as of the date of publication. Because TETCOS LLP must respond to changing market conditions, it should not be interpreted to be a commitment on the part of TETCOS LLP, and TETCOS LLP cannot guarantee the accuracy of any information presented after the date of publication.

This manual is for informational purposes only.

The publisher has taken care in the preparation of this document but makes no expressed or implied warranty of any kind and assumes no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained herein.

Warning! DO NOT COPY

Contact us at
TETCOS LLP
214, 39th A Cross, 7th Main, 5th Block Jayanagar,
Bangalore - 560 041, Karnataka, INDIA.
Phone: +91 80 26630624
E-Mail: sales@tetcos.com
Visit: www.tetcos.com

Table of Contents

1 About SDN

The salient features of SDN are:

- **Directly programmable**: SDN decouples the network control and forwarding functions. SDN allows programming the network control plane by abstracting the physical infrastructure.
- **Agile**: Decoupling and abstracting network control from forwarding helps administrators dynamically adjust network-wide traffic flow to meet the changing needs in the networks.
- **Centrally managed**: Network intelligence is centralized (logically) in software based SDN controllers that maintain a global view of the network, which appear 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 easily and quickly, because the programs do not depend on proprietary software.
- **Open standards-based and vendor-neutral**: Because SDN is 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.

2 SDN in NetSim

SDN is available in NetSim from version 11 onwards. NetSim 'simulates' OpenFlow protocol. OpenFlow is an open interface to remotely control the forwarding tables in network switches, routers, etc.

2.1 SDN Controller in NetSim

An SDN controller is an application in SDN that manages flow control to enable intelligent networking. SDN controller can be used to control the packet forwarding of all Layer 3 devices in the network.

SDN controller lies between the network devices and the applications. Any communication between applications and devices must go through the controller.

NetSim has inbuilt controllers that 'simulate' SDN. An SDN controller in NetSim contains a Command Line Interface (CLI) to allow you to configure properties, such as, the IP forwarding table for different devices in the network.

NetSim also provides a platform whereby users can develop various kind of commands/interface compatible to any SDN enabled device.

In NetSim, any Layer 3 device can be configured as an SDN Controller. Multiple controllers can be configured in a network scenario. The following is a list of Layer 3 devices you can configure as an SDN controller:

- Internetworks – Nodes (Wired, Wireless Node), L3 Switches, Routers
- MANETs – Nodes (Wired, Wireless Node), Bridge Node (Wired, Wireless Node), Routers
- WSN - Sensors and Sink Node
- IOT - Sensors and Gateway (LowPAN Gateway), Nodes (Wired, Wireless Node), Routers
- Cognitive Radio – CR CPE, Nodes (Wired, Wireless Node), Routers
- LTE – UE, EPC, Nodes (Wired, Wireless Node), Routers
- VANETs – Vehicle and RSU

- 5G mmWave – UE, EPC, Nodes (Wired, Wireless Node), Routers

Note: NetSim 'simulates' SDN protocol and cannot connect to real controllers such as Open Daylight.

2.2 CLI Commands for SDN in NetSim

You can use the following commands when you simulate SDN in NetSim:

- **Simulation-specific –** Pause, PauseAt, Continue, Stop, Exit, and Reconnect.
- **Route –** route add, route print, and route delete.
- **Ping Command –** ping (not supported on some network types, for example, Wireless Sensor Network)
- **ACL configuration –** ACL Enable, ACL Disable, ACL Print, and aclconfig.

Note: CLI commands in NetSim are NOT case-sensitive. To get detailed help about how to use CLI commands in NetSim, see 3.5 NetSim Interactive Simulation in User Manual.

2.3 How to use CLI Commands for SDN

Each device in NetSim has a console which can be accessed by right clicking on the device. Users can execute supported SDN commands on the console of the SDN controller node to configure other nodes in the network. Let us look at a few examples.

Example 1: To view the IP routing table of a node that is controlled by the SDN controller, use the following command syntax: ***<DeviceName with Device_ID>* route print.**

- For example, type **Wired_Node_2 route print** and press **Enter**.

The following image illustrates the output for the command.

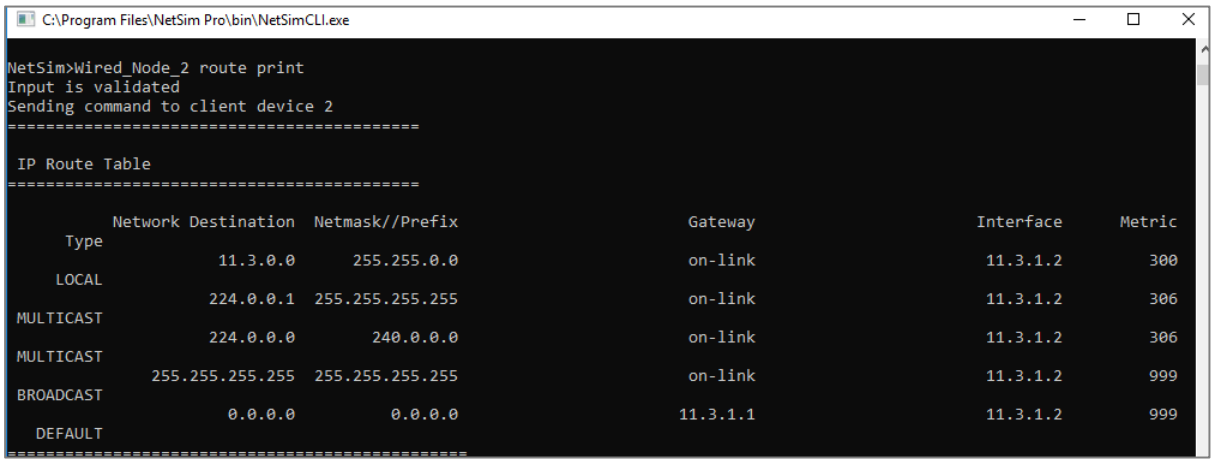


Figure 2-1: Printing IP routing table in Console

Example 2: To initiate ping from one node to another node where both nodes are controlled by the SDN controller, following command can be executed in the SDN controller:

<DeviceName with Device_ID> **Ping** ***<DeviceName with Device_ID>*** or ***<DeviceName with Device_ID>*** **Ping** ***<IP Address>*****.**

- For example, type Router_4 ping Router_5 or Router_4 ping 11.1.1.2 and press Enter.

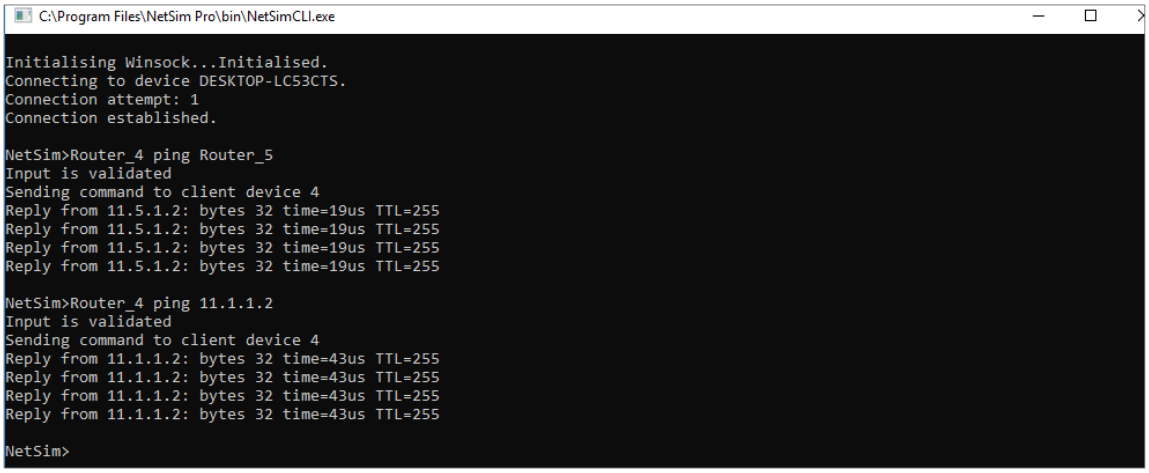


Figure 2-2: Pinging Router_4

Note: In order to initiate ping between the devices, ICMP protocol which is present in the Network Layer properties of the devices must be set to TRUE.

2.4 Excluded Features

Multiple controllers can be configured in NetSim. However, intercontroller communication requires the user to write their own code in NetSim.

3 Featured Examples

NetSim provides inbuilt examples to get started with the SDN module. To access these examples, you can go to **Examples** > **Software-Defined-Networks** in the NetSim Home Screen. You can change the default values of the parameters in these examples and see how they affect the performance of the SDN network.

3.1 Example 1: Configuring One SDN Controller in a Simple Internetwork

The Internetwork model in this example consists of the following configuration:

1. A subnet with 2 wired nodes, 3 routers, and a unicast application running on one of the wired nodes.
2. Only one router is configured as the SDN controller.
3. Open Flow protocol is enabled on all wired nodes and routers.
4. A unicast application is set between Wired_Node_1 to Wired_Node_2.
5. Set Transport Protocol to TCP in Application icon present in the top ribbon/toolbar.
6. Simulation time is set to 500 seconds.
7. Plots, Packet trace, and Event trace logs are enabled.

To simulate the example for One SDN controller in an Internetwork:

1. Open NetSim and click **Examples > Software-Defined-Networks > Configuring SDN > SDN-Internetworks**.

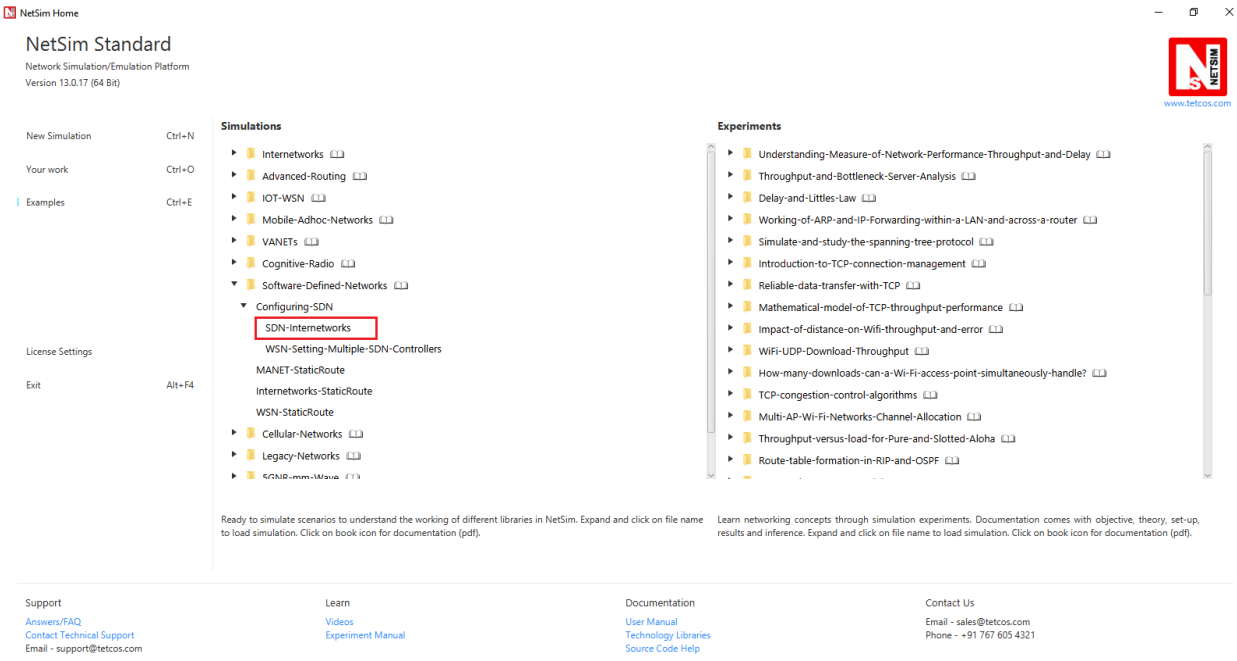


Figure 3-1: Featured Example list

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for SDN as shown **Figure 3-2.**

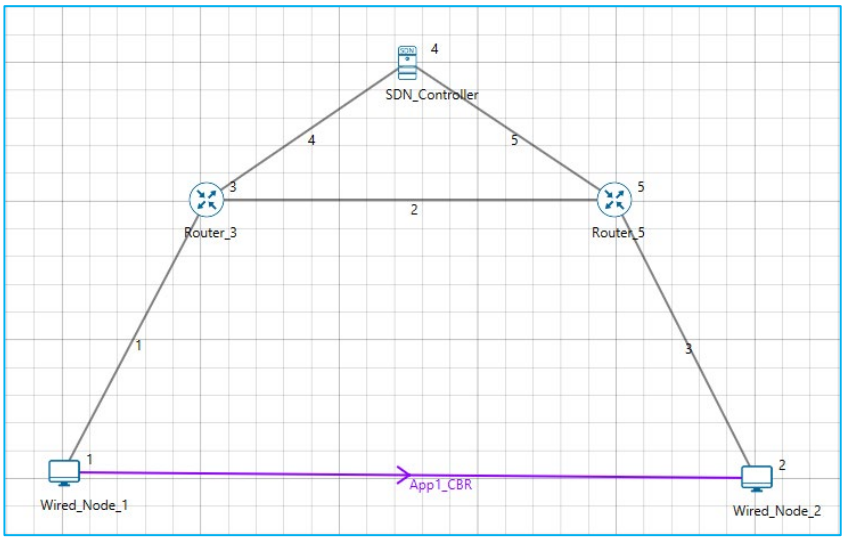


Figure 3-2: Network topology in the sample scenario

2. In the application layer of the SDN Controller node:

   a. **SDN_Controller** option set to **TRUE**.

   b. **Open_Flow** protocol is enabled.

   The following image illustrates the settings done in the SDN controller device:

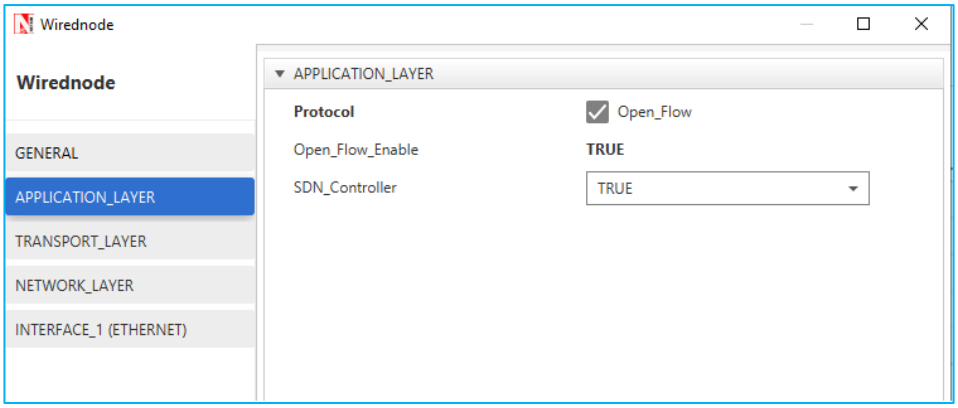


Figure 3-3: Application Layer properties Window

3. In the other devices (Routers and wired node) that are part of the network,
   a. **SDN_Controller** option is set to **FALSE.**
   b. **SDN_Controller_DeviceName** is set as **"SDN_Controller"** which is the name of the controller node.
   c. **Open_Flow** protocol is enabled.

   The following image illustrates settings done in the other devices:

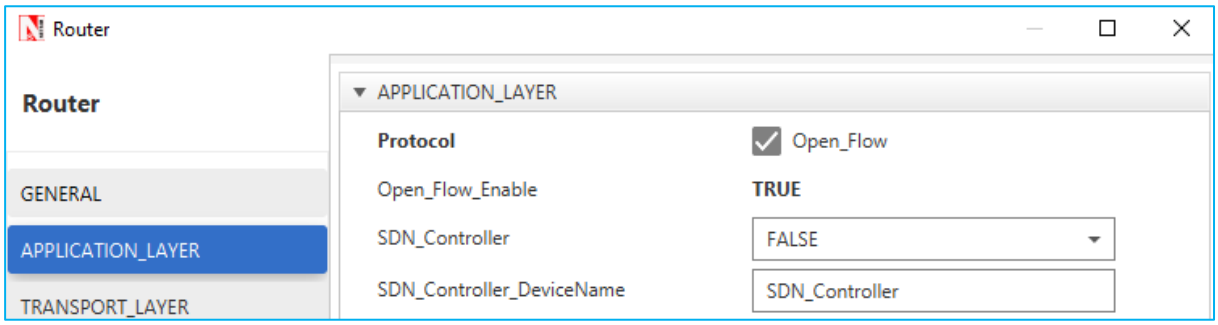


Figure 3-4: SDN_Controller_DeviceName is set as SDN_Controller in Application layer

The following table lists the settings done in all the devices that are part of the network:

Device Name	Open_Flow	SDN_Controller	SDN_Controller_DeviceName
Wired_Node_1	Enable	FALSE	SDN_Controller
Router_3	Enable	FALSE	SDN_Controller
SDN_Controller	Enable	TRUE	NA
Router_5	Enable	FALSE	SDN_Controller
Wired_Node_2	Enable	FALSE	SDN_Controller

Table 3-1: Set the properties for all the devices

4. A Unicast application has been configured between nodes 1 and 2.
5. Additional analysis options such as plots, packet trace and event trace are enabled.
6. Simulate the example. To do so:
    a. Click on the Run. The Run Simulation pop-up window appears.
    b. Simulation time is set to 500 Seconds.
    c. In the Run time Interaction tab, Interactive Simulation option is set to True.
    d. Click **Accept**.
    e. Click **OK**.

Simulation (NetSimCore.exe) starts to run. NetSimCore.exe displays the following message: waiting for first client to connect.

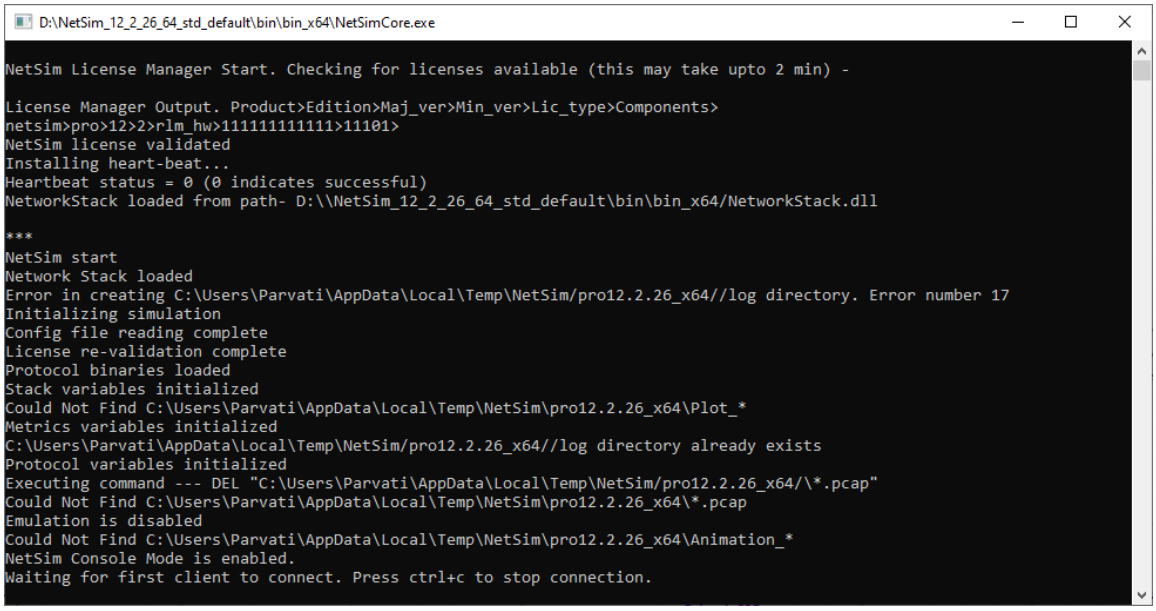


Figure 3-5: Waiting for first client to connect

7. To use the SDN CLI Console:
    a. Right-click **SDN_Controller** and select NetSim Console. Now, the client (NetSimCLI.exe) starts and attempts to establish a connection with NetSimCore.exe.

    The following image illustrates the NetSim CLI console.

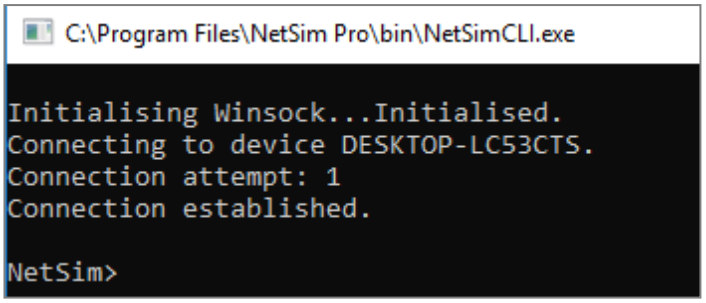


Figure 3-6: Connection established

b. Use this console to execute SDN commands.

3.2 Example 2: Configuring Multiple SDN Controllers in a Wireless Sensor Network

In this example, multiple SDN controllers are configured in a Wireless Sensor Network.

The Wireless Sensor Network model in this example consists of the following configuration:

- A subnet with 4 wireless sensors,1 ad-hoc link, 1 WSN sink node, and a unicast sensor application running on one of the wireless sensors.
- Set Transport Protocol to TCP in Application icon present in the top ribbon/toolbar.
- Two wireless sensors are configured as SDN controllers.
- Open Flow protocol is enabled on all wireless sensors and sink node.

NetSim uses the following defaults for this SDN example:

a. The unicast application transmits data from Wireless Sensor_3 to Wireless Sensor_5.

b. Simulation runs for 200 seconds.

c. Plots, Packet trace, and Event trace is enabled.

Note: For a WSN network, ping command is not supported on ZigBee device since these these nodes do not support ICMP protocol.

To simulate the example for multiple SDN controllers in WSN for SDN:

1. Open NetSim and click **Examples > Software-Defined-Networks > Configuring SDN > WSN-Setting-Multiple-SDN-Controllers**.

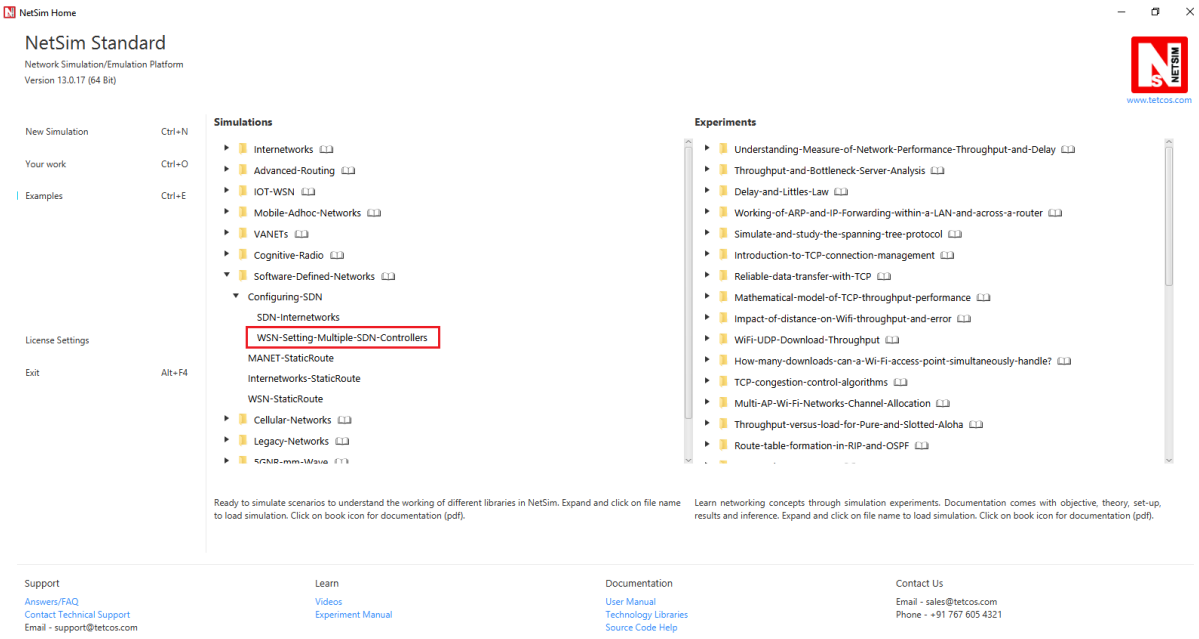


Figure 3-7: Featured Example list

The following network diagram illustrates the network setup related to this example as shown below **Figure 3-8.**

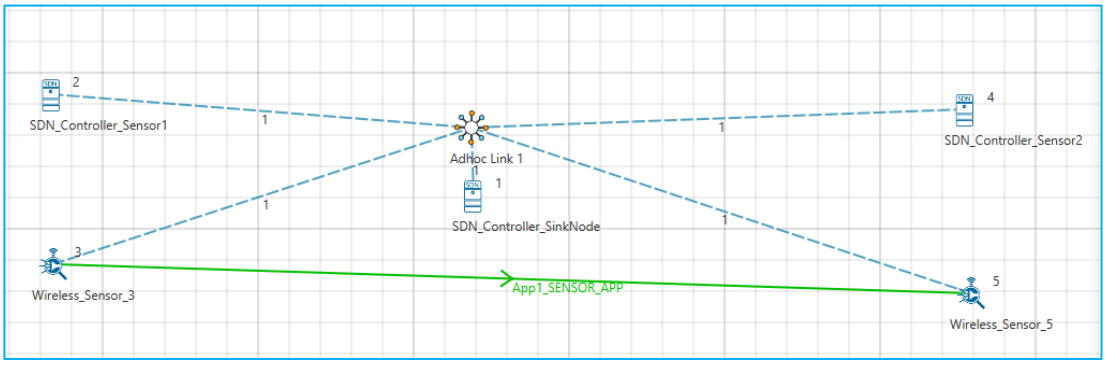


Figure 3-8: Network topology in the sample scenario

2. The sensors 1, 2 and the sinknode are configured as SDN controllers with Open Flow protocol enabled in all the nodes.

   The following image illustrates the settings done in the sensors configured as SDN controller as shown below **Figure 3-9.**

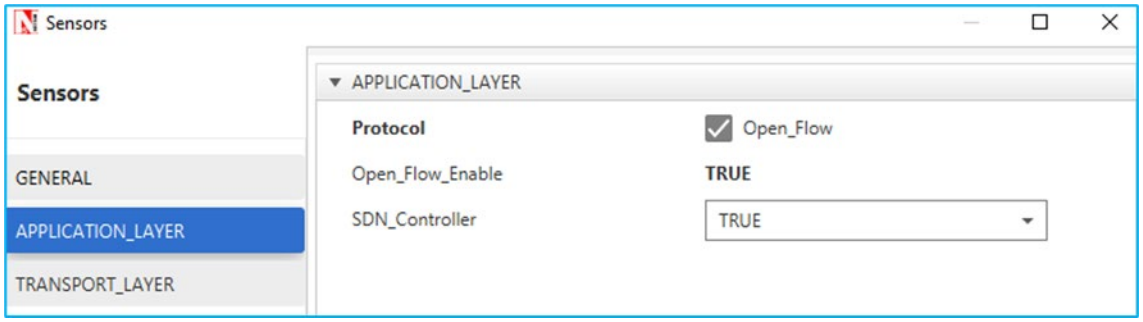


Figure 3-9: Application Layer properties Window

3. The sensor SDN_Controller_Sensor_1 is set as the SDN controller for Wireless_Sensor_3 and SDN_Controller_Sensor_2 is set as the SDN controller for Wireless_Sensor_5.

   The following image illustrates the settings done in sensor 3 as shown **Figure 3-10.**

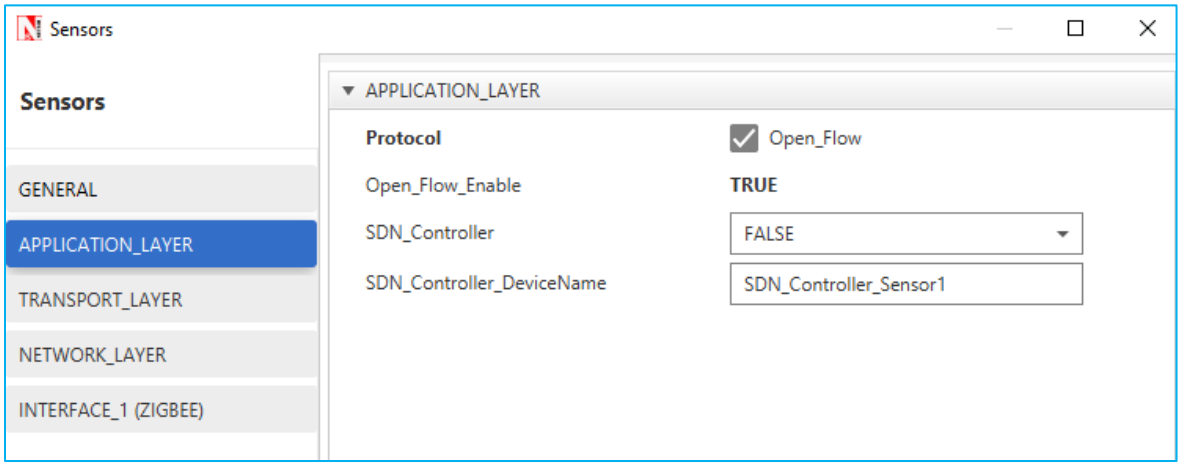


Figure 3-10: SDN_Controller_DeviceName is set as SDN_Controller in Application layer

The following table lists the default setting for the wireless sensors.

Device Name	Open_Flow	SDN_Controller	SDN_Controller_DeviceName
SDN_Controller_Sensor1	Enable	TRUE	NA
SDN_Controller_Sensor2	Enable	TRUE	NA
Wireless_Sensor_3	Enable	FALSE	SDN_Controller_Sensor1
Wireless_Sensor_5	Enable	FALSE	SDN_Controller_Sensor2

Table 3-2: Set the properties for all the devices

4. A unicast application is configured between sensor 3 and the sensor 5.
5. Additional analysis options such as plots, packet trace and event trace are enabled.

6. Simulate the example. To do so:

   a. Click the Run button. The Run Simulation pop-up window appears.

   b. Simulation time is set to 200 seconds.

   c. In the Run time Interaction tab, Interactive Simulation option is set to True.

   d. Click **Accept**.

   e. Click **OK**.

Simulation (NetSimCore.exe) starts to run. NetSimCore.exe displays the following message: waiting for first client to connect.

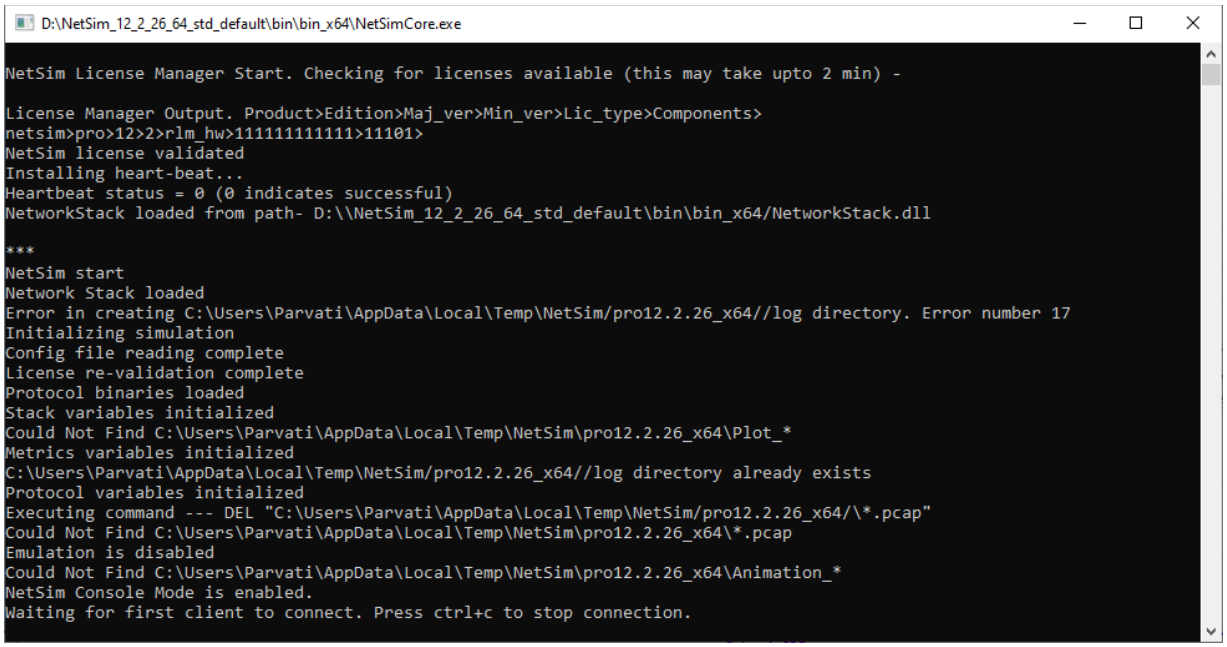


Figure 3-11: Waiting for first client to connect

7. To use the SDN CLI Console:

   a. Right-click **SDN_Controller_Sensor1** or **SDN_Controller_Sensor2** and click NetSim Console.

Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe.

The following image illustrates the NetSim CLI console.

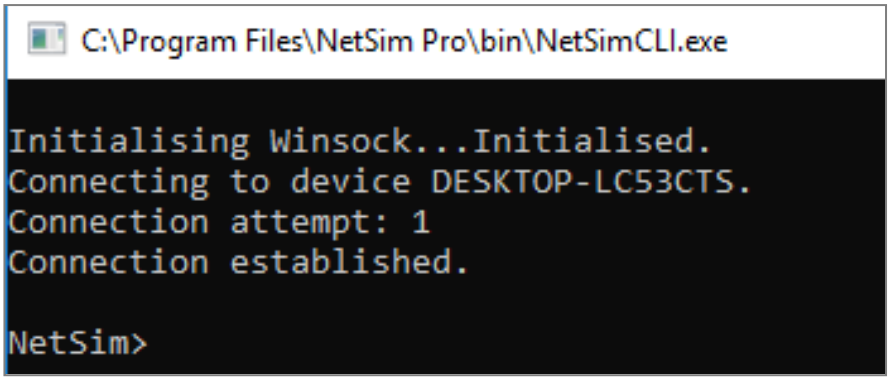


Figure 3-12: Connection established

You can execute various SDN commands that are supported.

3.3 Example 3: How to Change the IP tables in devices in NetSim using SDN Commands

In the example the IP tables in the nodes and routers on an SDN network are modified using SDN CLI commands.

The network model in this example consists of the following configuration:

- A subnet with 2 wired nodes, 5 routers, and a unicast application running on one of the wired nodes.
- SDN controller running on one of the router.
- Set Transport Protocol to TCP in Application icon present in the top ribbon/toolbar. Open Flow protocol is enabled on all wired nodes and routers.
- A unicast application set from Wired_Node_1 to Wired_Node_2.
- OSPF is the routing protocol in the routers.
- The node SDN_Controller is configured to be the SDN controller.
- Simulation time is set to 500 seconds.
- Plots, Packet trace, and Event trace is enabled.

To simulate SDN and change the IP tables:

1. Open NetSim and click **Examples > Software-Defined-Networks > Internetworks-StaticRoute**.

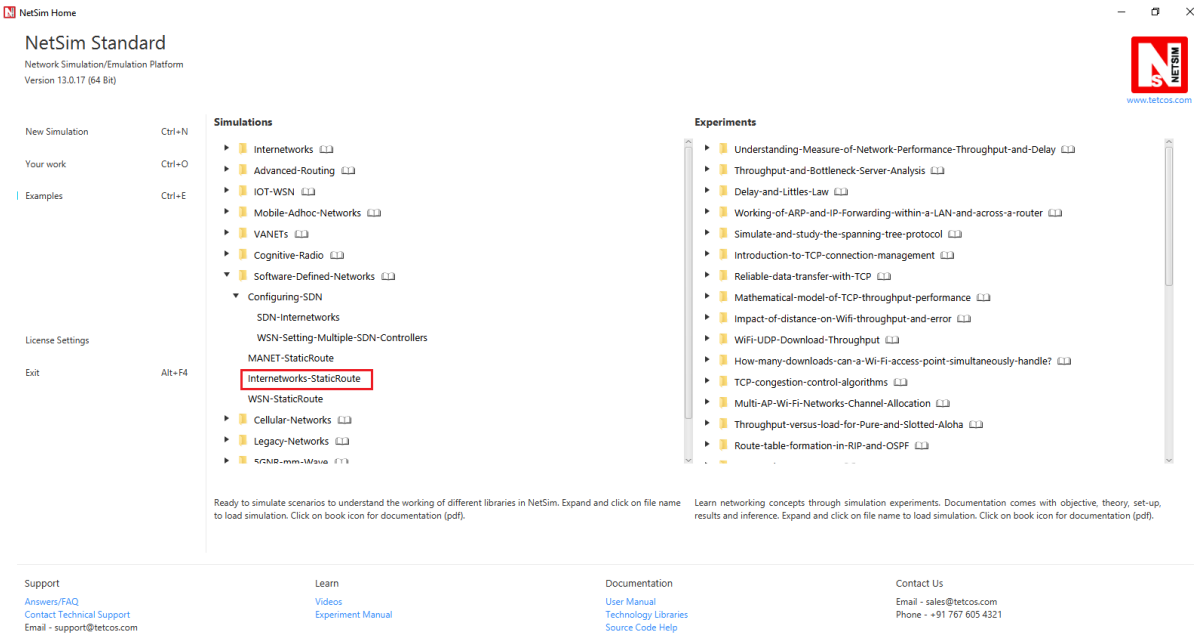


Figure 3-13: Featured Example list

The following network diagram illustrates the network setup related to this example as shown below **Figure 3-14.**

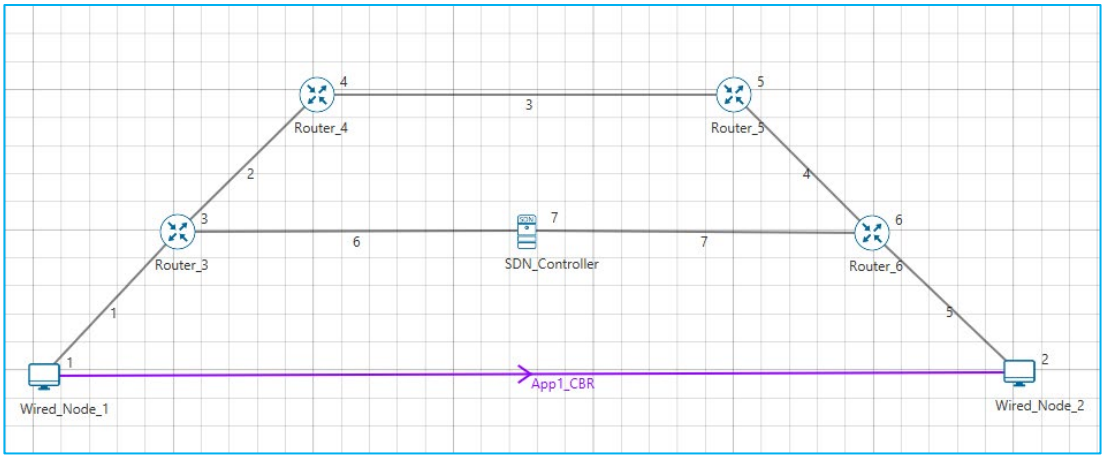


Figure 3-14: Network topology in the sample scenario

2. Unicast traffic is configured from Node 1 to Node 2 and Start Time 30s.
3. Plots, Packet Trace and Event Trace features are enabled.
4. Simulate SDN. To do so:
   a. Click the Run button. The Run Simulation pop-up window appears.
   b. Simulation time is set to 500 Seconds.
   c. In the Run time Interaction tab, Interactive Simulation option is set to True.
5. Right-click **SDN_Controller** and click NetSim Console. NetSim simulates SDN.

6. Interpret the results.

   a. Click View Animation and see the Packet Animation.

The packets reach Wired_Node_2 via Router 3 > SDN_Controller > Router 6 as shown in **Figure 3-15**

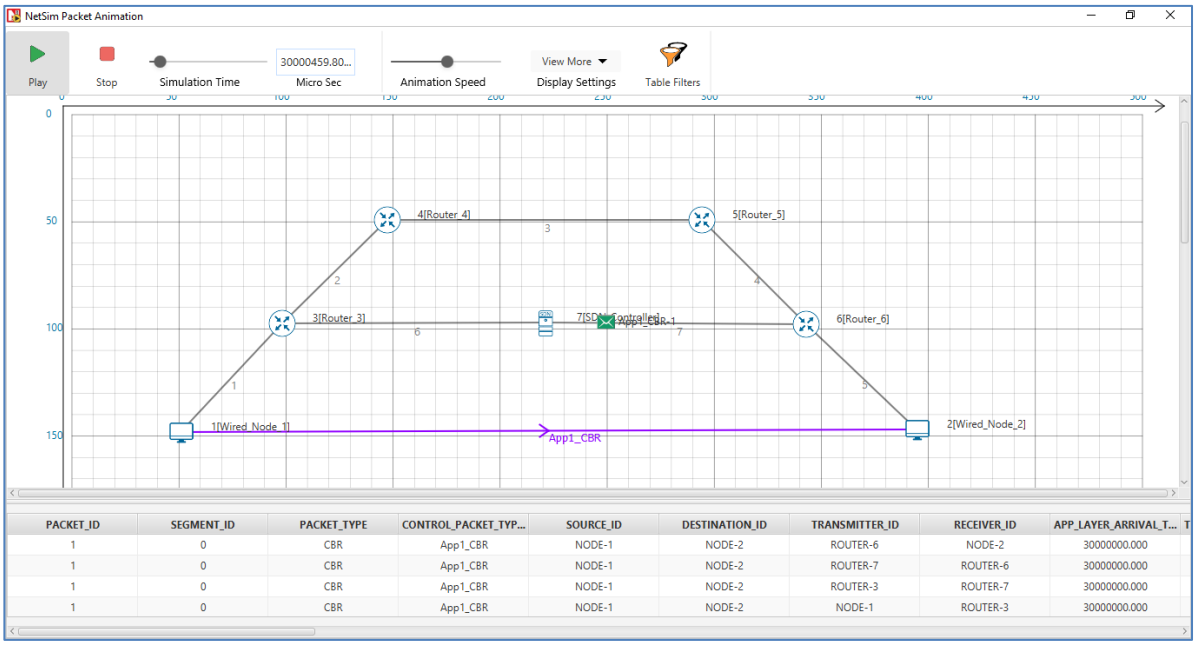


Figure 3-15: Packet Animation window

   b. Click **View Results** on the tool bar, **Link metrics** in the left area and check the **Detailed View** check box in the **Link_Metrics_Table** pop-up window.

   You will not see data packet transmissions in Links 2, 3, and 4 as shown in below **Figure 3-16**.

Link_Metrics_Table

Link_Metrics

Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided		Bytes_transmitted(bytes)	Payload_transmitted(bytes)	Overhead_transmitted(bytes)
		Data	Control	Data	Control	Data	Control			
All	NA	1888	4373	3	1	0	0	3174526	2752100	422426
1	Link_throughput	473	473	0	0	0	0	753024	690580	62444
2	Link_throughput	0	495	0	0	0	0	43348	0	43348
3	Link_throughput	0	496	0	0	0	0	43524	0	43524
4	Link_throughput	0	495	0	1	0	0	43448	0	43448
5	Link_throughput	470	473	0	0	0	0	748446	686200	62246
6	Link_throughput	473	970	1	0	0	0	772000	689120	82880
7	Link_throughput	472	971	2	0	0	0	770736	686200	84536

Figure 3-16: Link Metrics Table in result window

7. Configure static routes on Router_3 such that all traffic bound for the 11.4.1.2 subnet will go to a gateway 11.2.1.2, that is, from Router_3 > Router_4 > Router_5 > Router_6.

To configure the static routes for Router_3:

a. Simulate the network again.

b. Now, right-click **SDN_Controller** and click NetSim Console.

   Now, the client (NetSimCLI.exe) starts and attempts to establish a connection with NetSimCore.exe.

   NetSim CLI console opens.

c. Type the following commands on the NetSim CLI console, in the <*DeviceName with Device_ID*> **route add <*destination IP address*> MASK <*subnet mask*> <*gateway IP address*> <*metric*> if <*interface #*>** format.

   ROUTER_3 route add 11.5.1.2 mask 255.255.255.255 11.2.1.2 metric 1 if 2

d. (Optional) To check the static routes on ROUTER_3, type **ROUTER_3 route print**. The following image illustrates step (c) and (d) as shown below **Figure 3-17.**

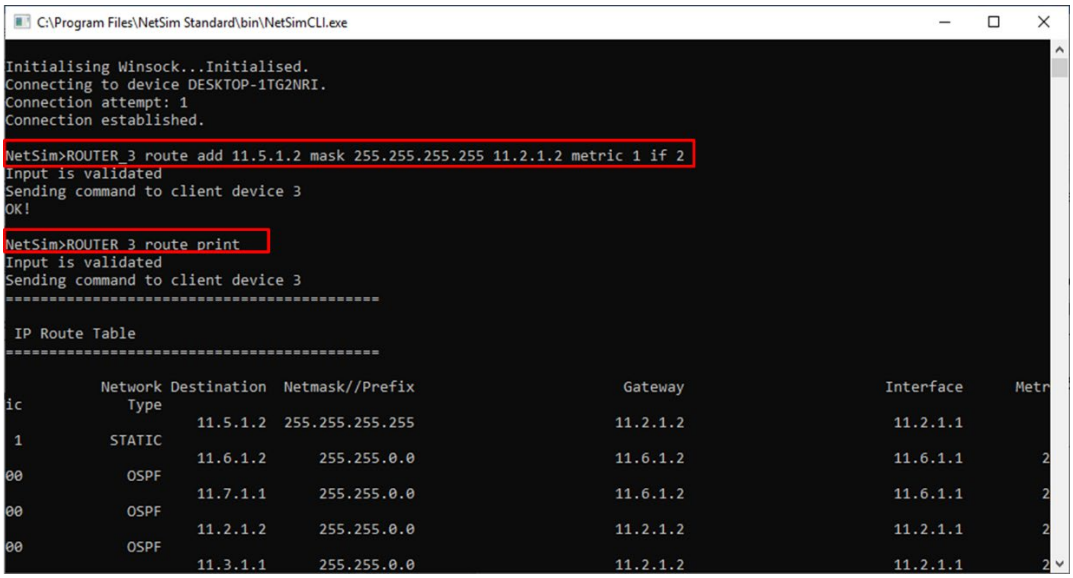


Figure 3-17: Route added to network and route printed in command

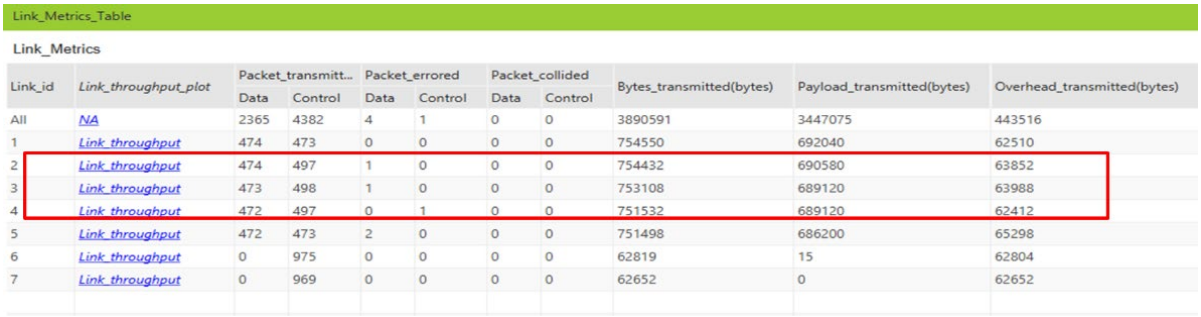
8. Interpret the results.

   a. Click **View Animation** and see the Packet Animation.

   b. The packets reach Wired_Node_2 via Router 3 > Router_4 > Router_5 > Router 6.

   c. Click **View Results** on the tool bar, **Link metrics** in the left area and check the **Detailed View** check box in the **Link_Metrics_Table** pop-up window. You will see data packet transmissions in Links 2, 3, and 4.

The following figure illustrates step (8c) as shown in below **Figure 3-18.**



Link_Metrics_Table

Link_Metrics

Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided		Bytes_transmitted(bytes)	Payload_transmitted(bytes)	Overhead_transmitted(bytes)
		Data	Control	Data	Control	Data	Control			
All	NA	2365	4382	4	1	0	0	3890591	3447075	443516
1	Link_throughput	474	473	0	0	0	0	754550	692040	62510
2	Link_throughput	474	497	1	0	0	0	754432	690580	63852
3	Link_throughput	473	498	1	0	0	0	753108	689120	63988
4	Link_throughput	472	497	0	1	0	0	751532	689120	62412
5	Link_throughput	472	473	2	0	0	0	751498	686200	65298
6	Link_throughput	0	975	0	0	0	0	62819	15	62804
7	Link_throughput	0	969	0	0	0	0	62652	0	62652

Figure 3-18: Link Metrics Table

d. In the simulation results window click on the Open **Packet trace** option in the left area. Once the packet trace opens, apply display filters to the **CONTROL_PACKET_TYPE/APP_NAME** column to show only **OPENFLOW_COMMAND** and **OPENFLOW_RESPONSE** packets. You will see that OpenFlow packets flow between Router_3 to SDN_Controller.

9. In Router_3 under appilication layer use routing protocol as RIP
10. Configure static routes for Router_3 as follows:
    a. Simulate SDN (**Refer step 4**).
    b. Right-click SDN_Controller and click NetSim Console.

       Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe. NetSim CLI console opens.

    c. Type the following commands on the NetSim CLI console, in the <*DeviceName with Device_ID*> **route add <*destination IP address*> MASK <*subnet mask*> <*gateway IP address*> <*metric*> if <*interface #*>** format.

       ROUTER_3 route add 11.5.1.2 mask 255.255.255.255 11.6.1.2 metric 1 if 3
11. Interpret the results.
    a. Click **View Animation** and see the Packet Animation. The packets reach Wired_Node_2 via Router 3 > SDN_Controller > Router_6.

3.4 Example 4: Configuring Static Routes for a MANET Network by using SDN

In this example static routes are configured using the SDN controllers, in a MANET network.

The MANET network model in this example consists of the following configuration:

- A subnet with 4 wireless nodes and 1 ad-hoc link, and a unicast CBR application running on one of the wireless nodes.
- DSR is the routing protocol that is enabled on all wireless nodes.
- One wireless node is configured as the SDN controller.
- Wireless nodes do not have mobility.
- A unicast application is configured from Wireless_Node_1 to Wireless_Node_5.
- Simulation time is set to 100 seconds.
- Packet trace, and Event trace is enabled.

To simulate the example for MANET using SDN:

1. Open NetSim and click **Examples > Software-Defined-Networks > MANET-StaticRoute**.

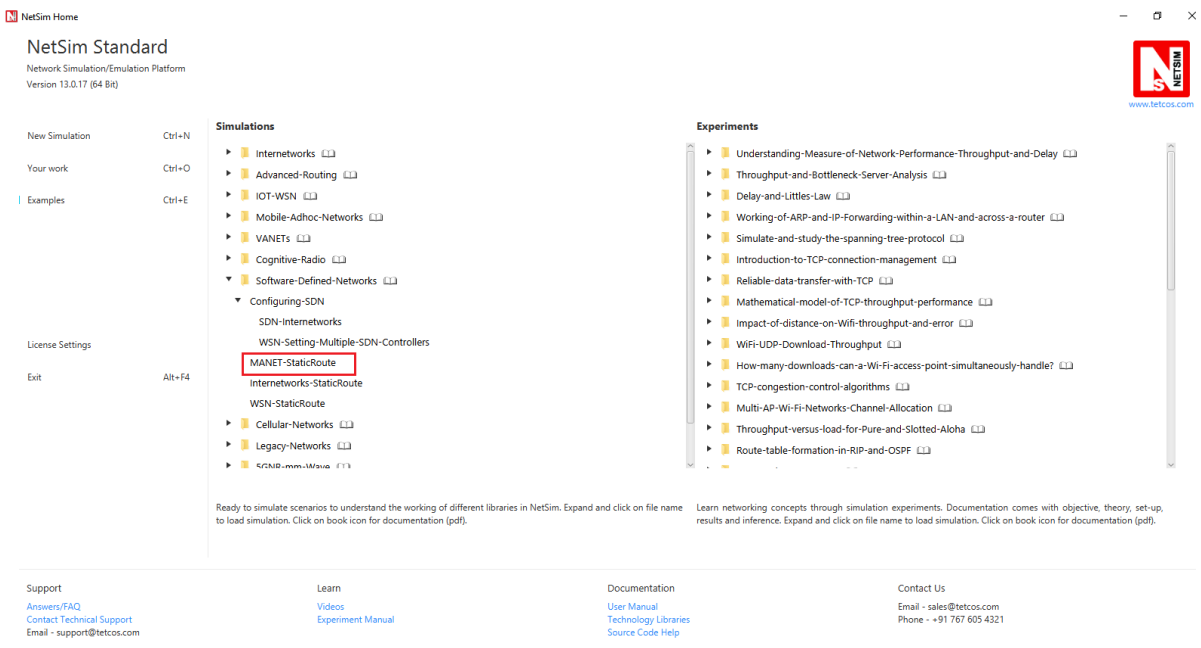


Figure 3-19: Featured Example list

The following network diagram illustrates the network setup as shown **Figure 3-20.**

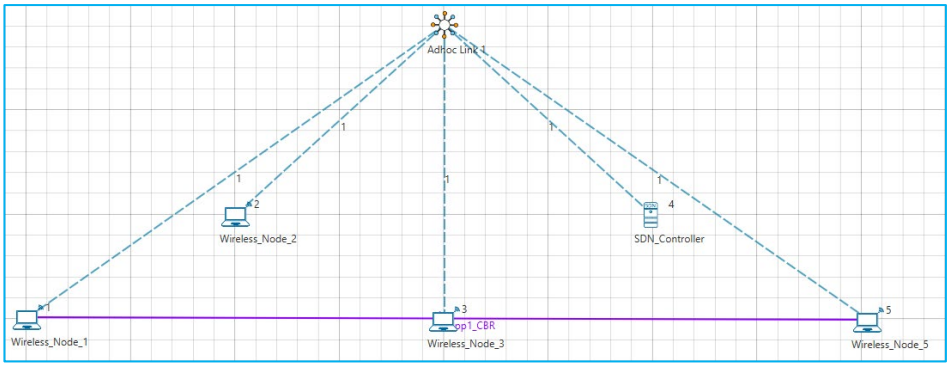


Figure 3-20: Network topology in the sample scenario

2. DSR routing protocol is configured in Network layer for all the nodes and set the nodes to be stationary.
3. Mobility model is set to NO_MOBILITY in all the nodes.
4. ICMP protocol is enabled in the network layer properties of all the wireless nodes.
5. Open flow protocol is enabled in all the nodes.

   The following settings were done in the wireless nodes to configure Wireless Node 4 as SDN controller and other nodes as its clients.

Device Name	Open_Flow	SDN_Controller	SDN_Controller_DeviceName
Wireless_Node_1	Enable	FALSE	SDN_Controller
Wireless_Node_2	Enable	FALSE	SDN_Controller
Wireless_Node_3	Enable	FALSE	SDN_Controller
SDN_Controller	Enable	TRUE	FALSE
Wireless_Node_5	Enable	FALSE	SDN_Controller

Table 3-3: Set the properties for all the devices

6. In NetSim GUI Plots are enabled.
7. Simulate the example. To do so:
   a. Click the Run button. The Run Simulation pop-up window appears.
   b. Simulation time is set to 100 seconds.
   c. In the Run time Interaction tab, the Interactive Simulation option is set to True.
   d. Click **Accept**.
   e. Click **OK**.

Simulation (NetSimCore.exe) starts to run. NetSimCore.exe displays the following message: waiting for first client to connect.

8. To use the SDN CLI Console:

   a. Right-click **SDN_Controller** and select NetSim Console. Now, the client (NetSimCLI.exe) starts and attempts to establish a connection with NetSimCore.exe.

9. Interpret the results.

   a. Click View Animation and see the Packet Animation.

The packets reach Wireless_Node_5 from Wireless_Node_1-> SDN_Controller Wireless_Node_5.

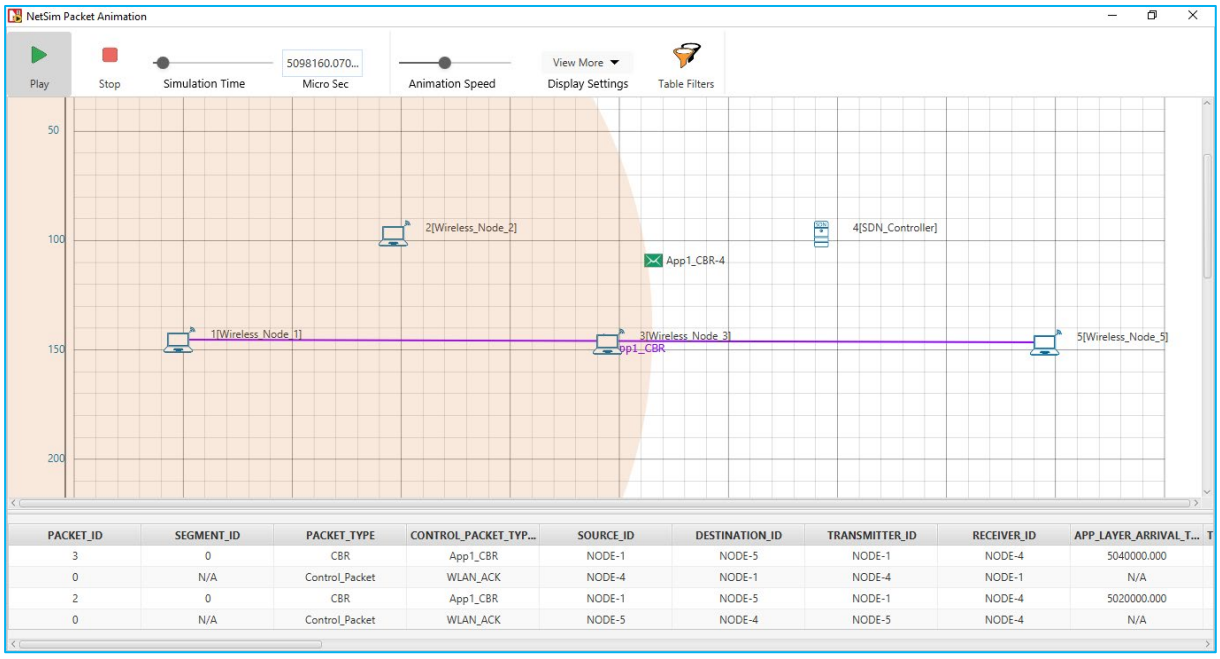


Figure 3-21 : Packet Animation window

10. Configure static routes such that the packets will go through Wireless_Node_1 > Wireless_Node_2, Wireless_Node_3, SDN_Controller to Wireless_Node_5.

To configure the static routes on SDN_Controller for all the nodes.

a. Run the simulation again.

b. Right-click **SDN_Controller** and select NetSim Console

Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe.

NetSim CLI console opens.

c. Type the following commands on the NetSim CLI console, in the <*DeviceName with Device_ID*> **route add <*destination IP address*> MASK <*subnet mask*> <*gateway IP address*> <*metric*> if <*interface #*>** format.

Wireless_Node_1 route add 11.1.1.5 mask 255.255.0.0 11.1.1.2 metric 1 if 1

Wireless_Node_2 route add 11.1.1.5 mask 255.255.0.0 11.1.1.3 metric 1 if 1

Wireless_Node_3 route add 11.1.1.5 mask 255.255.0.0 11.1.1.4 metric 1 if 1

SDN_Controller route add 11.1.1.5 mask 255.255.0.0 11.1.1.5 metric 1 if 1

11. Interpret the results.

a. Click **View Animation** and see the Packet Animation. The packets reach Wireless_Node_5 via Wireless_Node_1 > Wireless_Node_2 > Wireless_Node_3 > **SDN_Controller >** Wireless_Node_5

b. In the simulation results window select the **Open Packet trace** option in the left area. Once the packet trace opens apply filter in the **CONTROL_PACKET_TYPE/APP_NAME** column to show only **OPENFLOW_COMMAND** and **OPENFLOW_RESPONSE** messages.

You will see that OpenFlow packets flow from Wireless_Node_1 > Wireless_Node_2 > Wireless_Node_3 > SDN Controller to Wireless_Node_5 as part of the commands executed for static route configuration.

3.5 Example 5: Configuring Static Routes for a WSN Network by using SDN

In this example static routes are configured in sensors using SDN commands executed in a SDN controller.

The WSN network model in this example consists of the following configuration:

- A subnet with 4 wireless sensor, 1 ad-hoc link, 1 wireless sink node, and a unicast sensor application running on one of the wireless sensors.
- One wireless Sensor is configured as the SDN controller.
- A unicast application from Wireless_Sensor_1 to Wireless_Sensor_5.

- Simulation time of 100 seconds.
- Packet trace is enabled.

To simulate the example for WSN using SDN:

1. Open NetSim and click **Examples > Software-Defined-Networks > WSN-StaticRoute**.

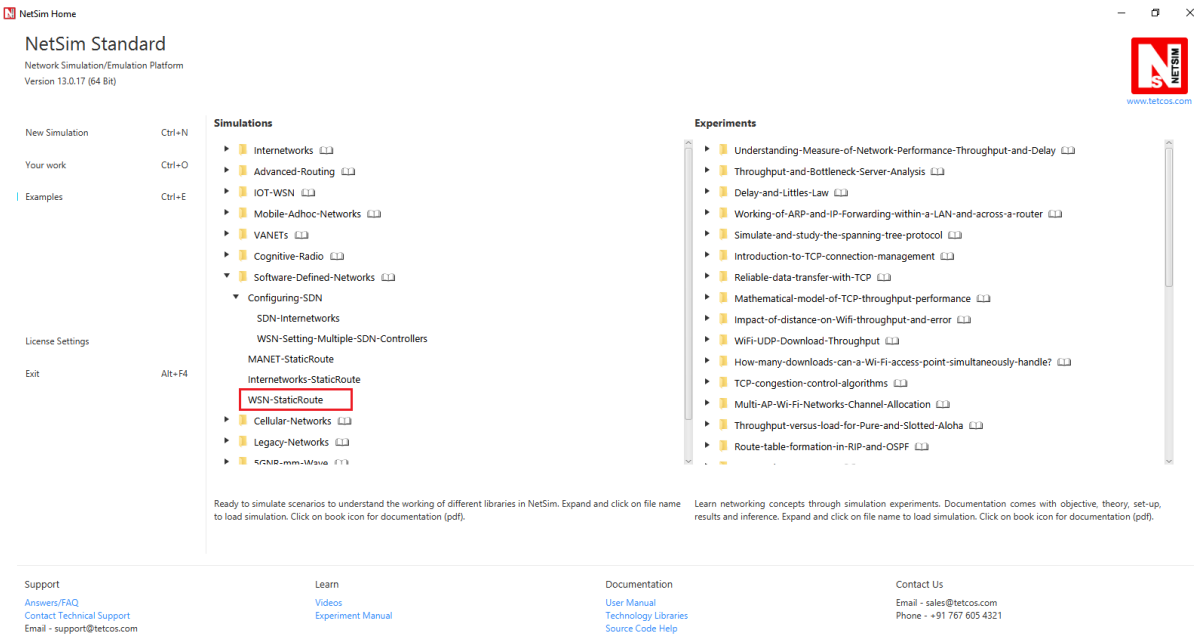


Figure 3-22: Featured Example list

The following network diagram illustrates the network setup considered in this example:

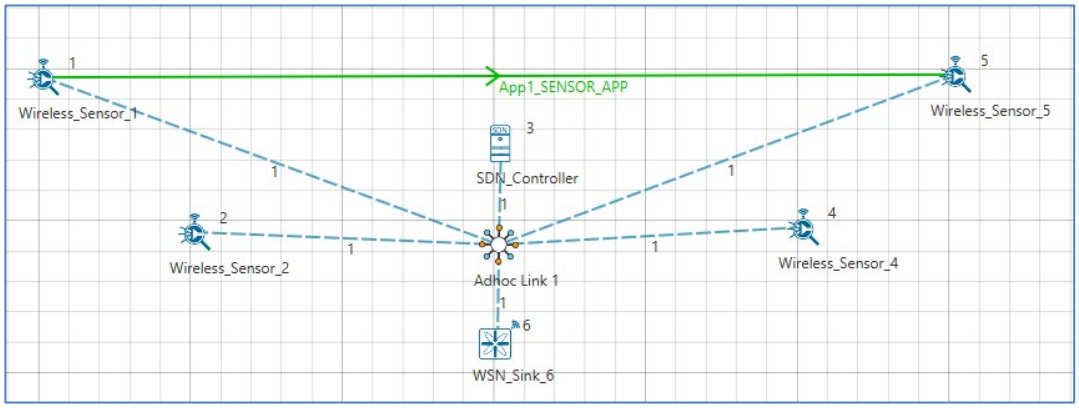


Figure 3-23: Network topology in the sample scenario

2. The following settings are done in the application layer properties of the devices for configuring SDN:

Device Name	Open_Flow	SDN_Controller	SDN_Controller_DeviceName
Wireless_Sensor_1	Enable	FALSE	SDN_Controller

Wireless_Sensor_2	Enable	FALSE	SDN_Controller
SDN_Controller	Enable	TRUE	NA
Wireless_Sensor_4	Enable	FALSE	SDN_Controller
Wireless_Sensor_5	Enable	FALSE	SDN_Controller
WSN_Sink_6	Enable	FALSE	SDN_Controller

Table 3-4: Set the properties for all the devices

3. In NetSim GUI Plots are enabled.
4. Simulate the example. To do so:
   a. Click on the Run button. The Run Simulation pop-up window appears.
   b. Simulation time is set to 100 seconds.
   c. In the Run time Interaction tab, the Interactive Simulation option is set to True.
   d. Click **Accept**.
   e. Click **OK**.

Simulation (NetSimCore.exe) starts to run. NetSimCore.exe displays the following message: waiting for first client to connect.

5. To use the SDN CLI Console:
   a. Right-click **SDN_Controller** and click NetSim Console. Now, the client (NetSimCLI.exe) starts and attempts to establish a connection with NetSimCore.exe.
6. Interpret the results.
   a. Click View Animation and see the Packet Animation.

The packets reach Wireless_Sensor_5 from Wireless_Sensor_1 to Wireless_Sensor_5.

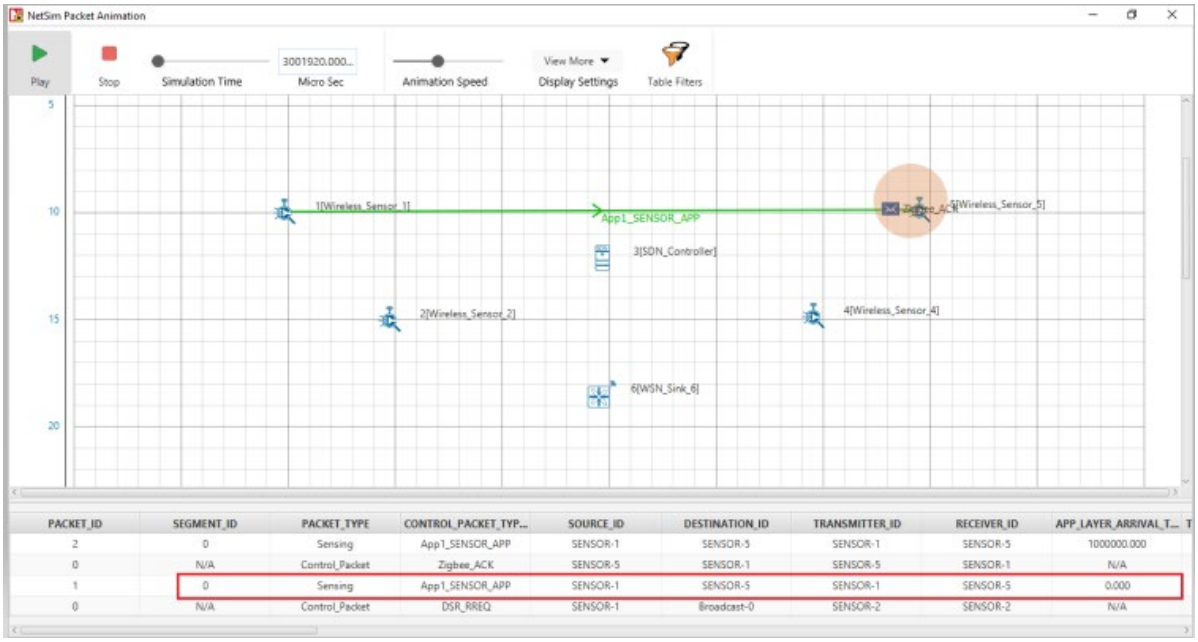


Figure 3-24: Direct communication between node 1 and node 5 in packet Animation window

7. Configure static routes such that the packets will go through Wireless_Sensor_1 to Wireless_Sensor_5 via SDN_Controller.

   To configure the static routes on SDN_Controller for Wireless_Sensor_1

   a. Run simulation again.

   b. Right-click SDN_Controller and click NetSim Console.

      Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe.

      NetSim CLI console opens.

   c. Type the following commands on the NetSim CLI console, in the <*DeviceName with Device_ID*> **route add <*destination IP address*> MASK <*subnet mask*> <*gateway IP address*> <*metric*> if <*interface #*>** format.

      Wireless_Sensor_1 route ADD 11.1.0.0 MASK 255.255.0.0 11.1.1.4 METRIC 1 IF 1

      SDN_Controller route ADD 11.1.0.0 MASK 255.255.0.0 11.1.1.6 METRIC 1 IF 1

8. Interpret the results.

   a. Click **View Animation** and see the Packet Animation.

      The packets reach Wireless_Sensor_5 from Wireless_Sensor_4 > SDN_Controller to Wireless_Sensor_5.

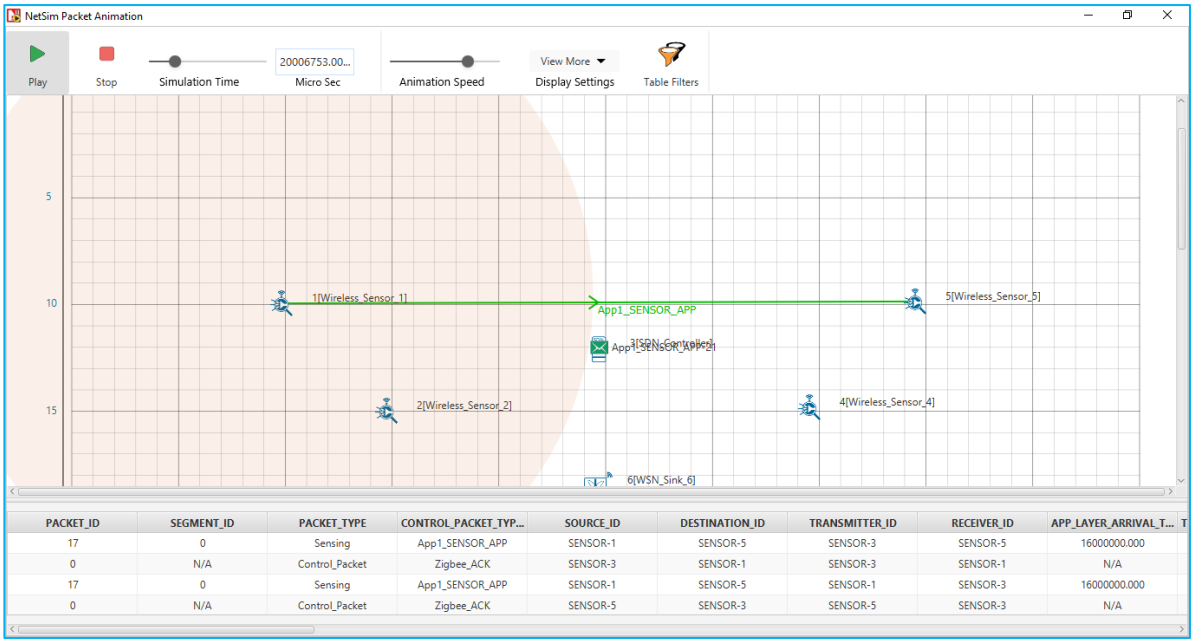


Figure 3-25: Communication between node 1 > SDN Controller > node 5 in packet Animation window

Packet Animation window

b. In the simulation results window select the **Open Packet trace** option in the left area. Once the packet trace opens apply filter in the **CONTROL_PACKET_TYPE/APP_NAME** column to show only **OPENFLOW_COMMAND** and **OPENFLOW_RESPONSE** messages.

   You will see that OpenFlow packets flow from Wireless_Sensor_4 to SDN_Controller as part of the static route configuration done via console.

4 Latest FAQs

You can refer to the up-to-date FAQs about NetSim's SDN library at

https://tetcos.freshdesk.com/support/solutions/folders/14000122307.