

Software Defined Networks

Contents

1	About SDN	2
2	SDN in NetSim	2
	2.1 SDN Controller in NetSim	2
	2.2 CLI Commands for SDN in NetSim.....	3
	2.3 How to use CLI Commands for SDN	4
	2.4 Excluded Features.....	5
3	Featured Examples.....	6
	3.1 Example 1: Configuring One SDN Controller in a Simple Internetwork	6
	3.2 Example 2: Configuring Multiple SDN Controllers in a Wireless Sensor Network.....	9
	3.3 Example 3: How to Change the IP tables in devices in NetSim using SDN Commands.....	11
	3.4 Example 4: Configuring Static Routes for a MANET Network by using SDN.....	16
	3.5 Example 5: Configuring Static Routes for a WSN Network by using SDN.....	18
4	Latest FAQs	21

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 3.2 NetSim Interactive Simulation in User Manual.

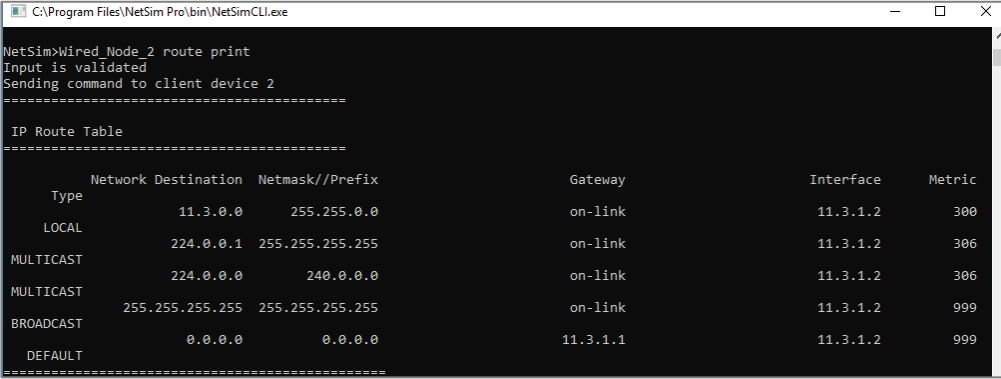
2.3 How to use CLI Commands for SDN

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.

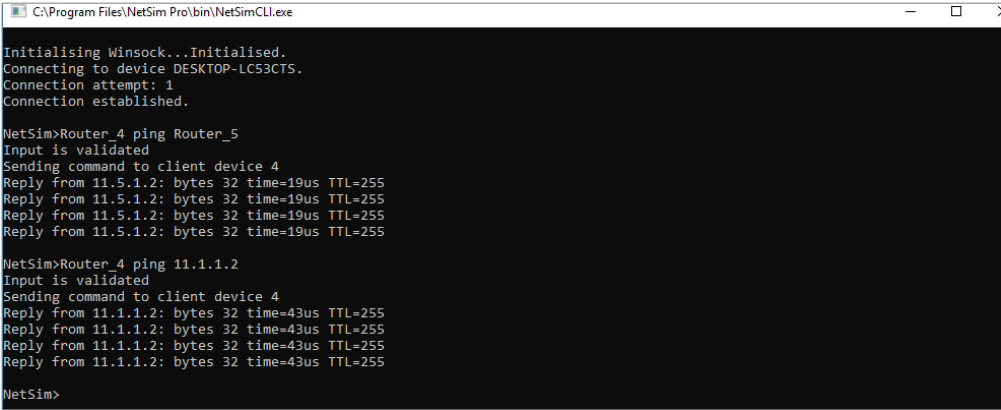


```
C:\Program Files\NetSim Pro\bin\NetSimCLI.exe
NetSim>Wired_Node_2 route print
Input is validated
Sending command to client device 2
=====
IP Route Table
=====
      Network Destination  Netmask//Prefix          Gateway          Interface        Metric
      Type
      LOCAL                11.3.0.0                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
      0.0.0.0                0.0.0.0                11.3.1.1         11.3.1.2         999
=====
```

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**.



```
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
Reply from 11.5.1.2: bytes 32 time=19us TTL=255
Reply from 11.5.1.2: bytes 32 time=19us TTL=255
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
Reply from 11.1.1.2: bytes 32 time=43us TTL=255
Reply from 11.1.1.2: bytes 32 time=43us TTL=255
Reply from 11.1.1.2: bytes 32 time=43us TTL=255

NetSim>
```

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

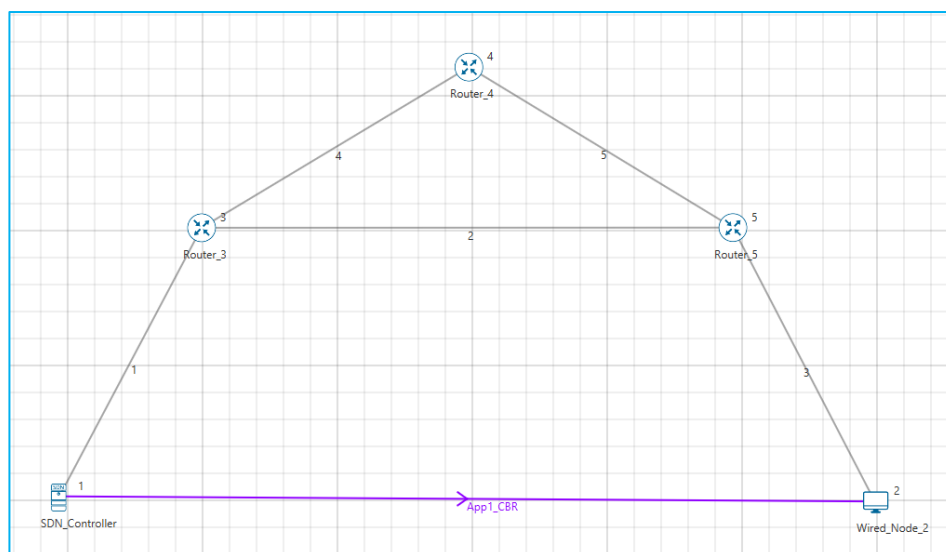
The Internetwork model in this example consists of the following configuration:

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

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

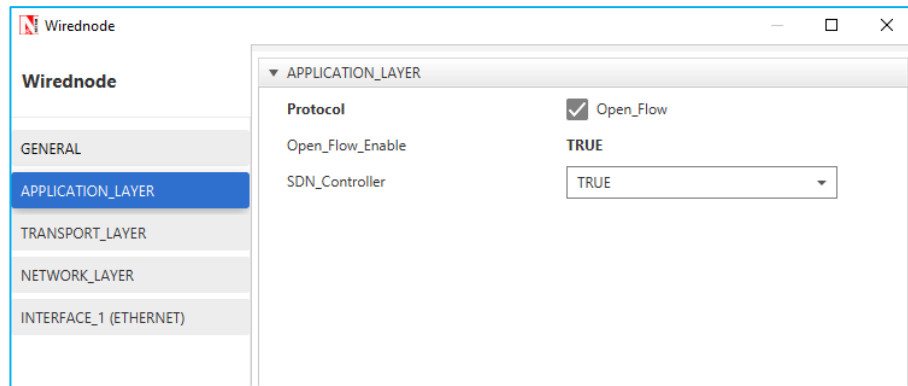
1. Open NetSim and click **Examples > Software-Defined-Networks > Configuring SDN > SDN-Internetworks**.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for SDN.



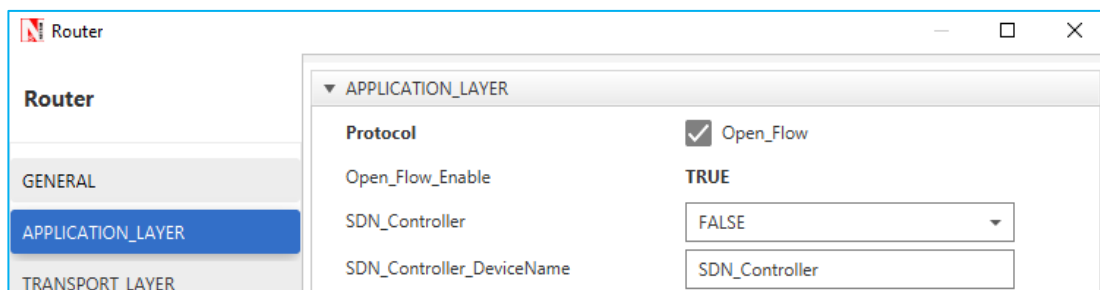
2. In the application layer of the SDN Controller node:
 - a. **SDN_Controller** option set to **TRUE**.
 - b. **Open_Flow** protocol is enabled

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



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:



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	TRUE	NA
Router_3	Enable	FALSE	SDN_Controller
Router_4	Enable	FALSE	SDN_Controller
Router_5	Enable	FALSE	SDN_Controller
Wired_Node_2	Enable	FALSE	SDN_Controller

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.

```

D:\NetSim_12_2_26_64_std_default\bin\bin_x64\NetSimCore.exe
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -
License Manager Output. Product>Edition>Maj_ver>Min_ver>Lic_type>Components>
netsim>pro>12>2>rlm_hw>111111111111>11101>
NetSim license validated
Installing heart-beat...
Heartbeat status = 0 (0 indicates successful)
NetworkStack loaded from path- D:\NetSim_12_2_26_64_std_default\bin\bin_x64\NetworkStack.dll
***
NetSim start
Network Stack loaded
Error in creating C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\log directory. Error number 17
Initializing simulation
Config file reading complete
License re-validation complete
Protocol binaries loaded
Stack variables initialized
Could Not Find C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\Plot_*
Metrics variables initialized
C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\log directory already exists
Protocol variables initialized
Executing command --- DEL "C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\*.pcap"
Could Not Find C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\*.pcap
Emulation is disabled
Could Not Find C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\Animation_*
NetSim Console Mode is enabled.
Waiting for first client to connect. Press ctrl+c to stop connection.
  
```

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.

```

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

NetSim>
  
```

- 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:

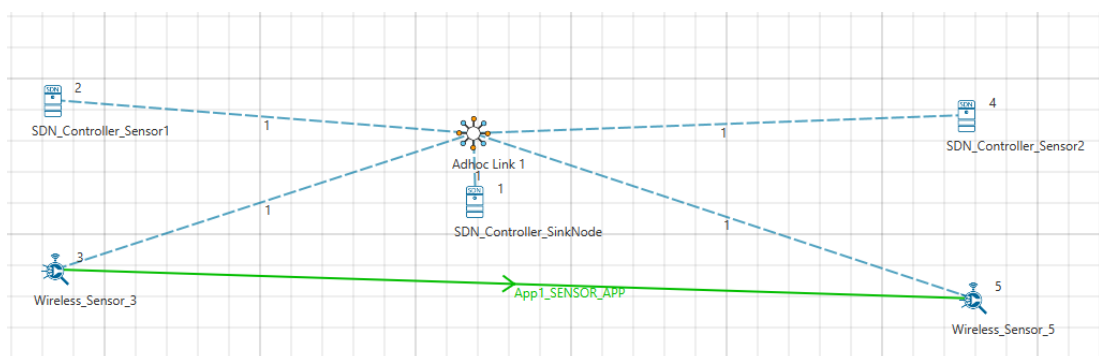
- The unicast application transmits data from Wireless Sensor_3 to Wireless Sensor_5.
- Simulation runs for 200 seconds.
- Plots, Packet trace, and Event trace is enabled.

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

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

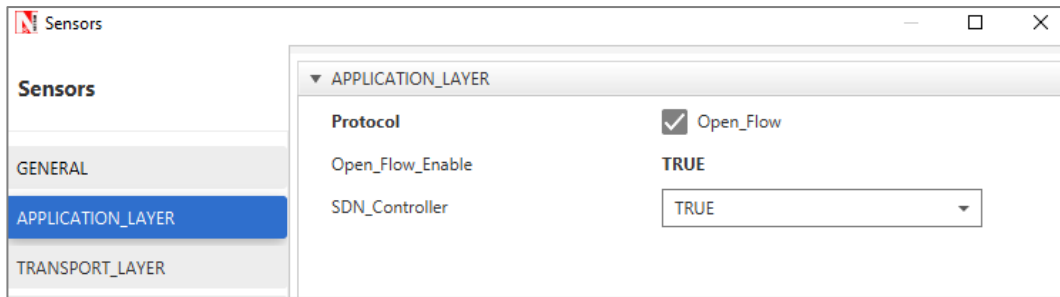
1. Open NetSim and click **Examples > Software-Defined-Networks > Configuring SDN > WSN-Setting-Multiple-SDN-Controllers**.

The following network diagram illustrates the network setup related to this example:



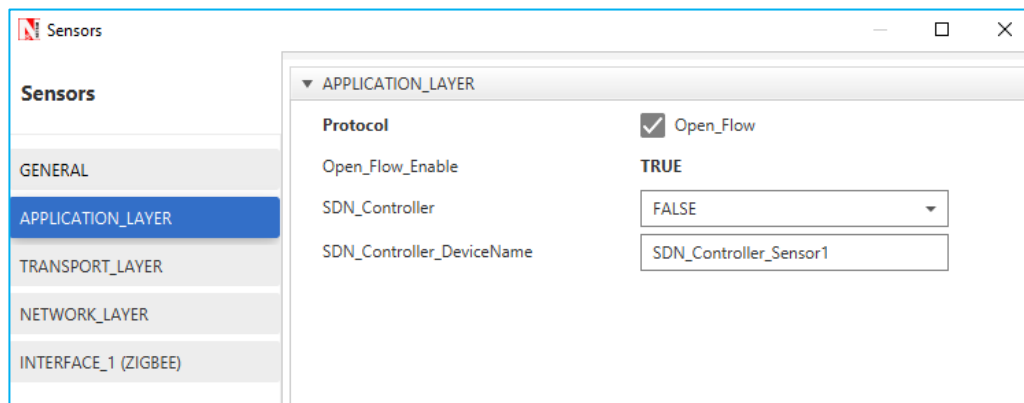
2. The sensors 1, 2 and the sinknode are configured as SDN controllers with Open Flow protocol enabled in all the nodes.

The following image illustrates the settings done in the sensors configured as SDN controller:



3. The sensor SDN_Controller_Sensor1 is set as the SDN controller for Wireless_Sensor_3 and SDN_Controller_Sensor2 is set as the SDN controller for Wireless_Sensor_5.

The following image illustrates the settings done in sensor 3



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

Device Name	Open_Flow	SDN_Controller	SDN_Controller_DeviceName
Wireless_Sensor_2	Enable	TRUE	NA
Wireless_Sensor_4	Enable	TRUE	NA
Wireless_Sensor_3	Enable	FALSE	SDN_Controller_Sensor1
Wireless_Sensor_5	Enable	FALSE	SDN_Controller_Sensor2

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.

```
D:\NetSim_12_2_26_64_std_default\bin\bin_x64\NetSimCore.exe
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -
License Manager Output. Product>Edition>Maj_ver>Min_ver>Lic_type>Components>
netsim>pro>12>2>rlm_hw>111111111111>11101>
NetSim license validated
Installing heart-beat...
Heartbeat status = 0 (0 indicates successful)
NetworkStack loaded from path- D:\NetSim_12_2_26_64_std_default\bin\bin_x64\NetworkStack.dll

***
NetSim start
Network Stack loaded
Error in creating C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\log directory. Error number 17
Initializing simulation
Config file reading complete
License re-validation complete
Protocol binaries loaded
Stack variables initialized
Could Not Find C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\Plot_*
Metrics variables initialized
C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\log directory already exists
Protocol variables initialized
Executing command --- DEL "C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\*.pcap"
Could Not Find C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\*.pcap
Emulation is disabled
Could Not Find C:\Users\Parvati\AppData\Local\Temp\NetSim\pro12.2.26_x64\Animation_*
NetSim Console Mode is enabled.
Waiting for first client to connect. Press ctrl+c to stop connection.
```

- 7. To use the SDN CLI Console:

- c. 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.

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

NetSim>
```

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

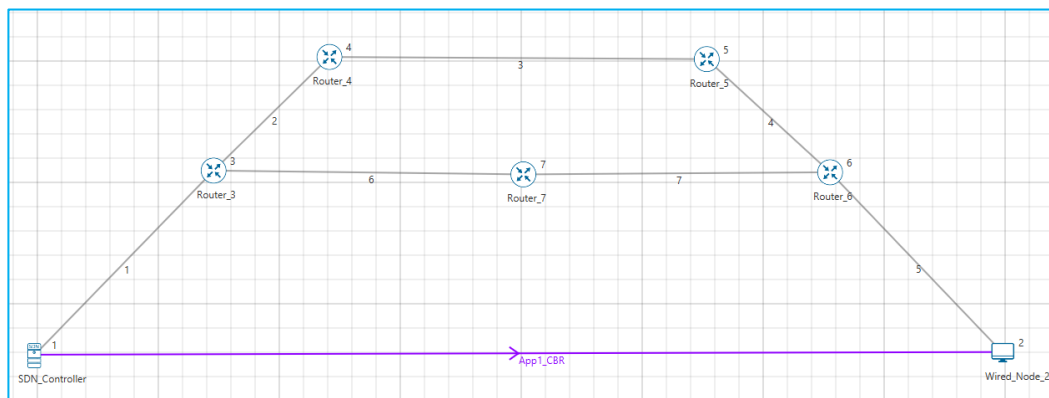
In the example the IP tables in the nodes and routers on an SDN network are modified using SDN CLI commands.

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 wired nodes.
- 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 SDN_Controller 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.

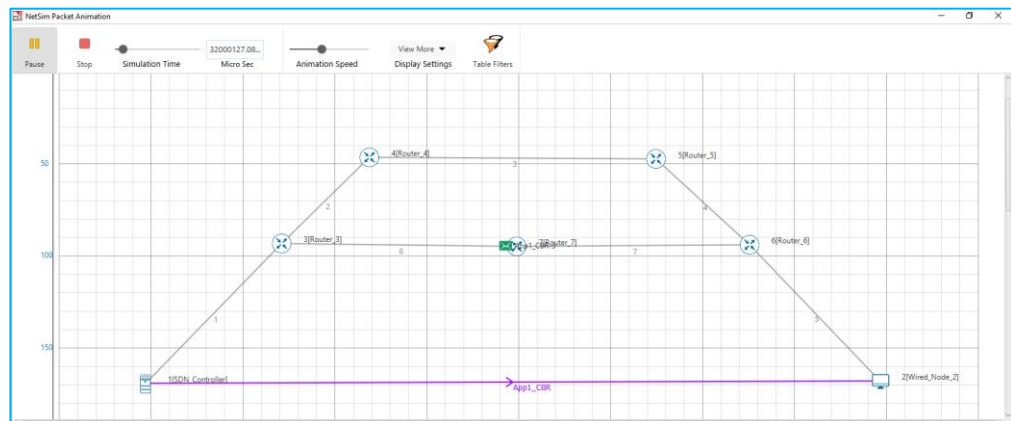
To simulate SDN and change the IP tables:

1. Open NetSim and click **Examples > Software-Defined-Networks > Internetworks-StaticRoute**. The following network diagram illustrates the network setup related to this example:



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 > Router_7 > Router 6.

The following figure illustrates step (8a).



- 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.

The following figure illustrates step (6a).

Link_Metrics_Table									
Link_Metrics <input checked="" type="checkbox"/> Detailed View									
Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided		Bytes_transmitted(bytes)	Payload_transmitted(bytes)
		Data	Control	Data	Control	Data	Control		
All	NA	1888	4373	3	1	0	0	3174526	2752100
1	Link throughput	473	473	0	0	0	0	753024	690580
2	Link throughput	0	495	0	0	0	0	43348	0
3	Link throughput	0	496	0	0	0	0	43524	0
4	Link throughput	0	495	0	1	0	0	43448	0
5	Link throughput	470	473	0	0	0	0	748446	686200
6	Link throughput	473	970	1	0	0	0	772000	689120
7	Link throughput	472	971	2	0	0	0	770736	686200

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 on 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 (9c) and (9d).

```

C:\Program Files\NetSim Standard\bin\NetSimCLI.exe
Initialising Winsock...Initialised.
Connecting to device DESKTOP-1TG2NRI.
Connection attempt: 1
Connection established.

NetSim>ROUTER_3 route add 11.5.1.2 mask 255.255.255.255 11.2.1.2 metric 1 if 2
Input is validated
Sending command to client device 3
OK!

NetSim>ROUTER_3 route print
Input is validated
Sending command to client device 3
=====
IP Route Table
=====
ic      Network Destination Netmask//Prefix      Gateway      Interface      Metr
1       STATIC      11.5.1.2      255.255.255.255      11.2.1.2      11.2.1.1
00      OSPF        11.6.1.2      255.255.0.0         11.6.1.2      11.6.1.1      2
00      OSPF        11.7.1.1      255.255.0.0         11.6.1.2      11.6.1.1      2
00      OSPF        11.2.1.2      255.255.0.0         11.2.1.2      11.2.1.1      2
00      OSPF        11.3.1.1      255.255.0.0         11.2.1.2      11.2.1.1      2

```

8. Interpret the results.

- Click **View Animation** and see the Packet Animation.
- The packets reach Wired_Node_2 via Router 3 > Router_4 > Router_5 > Router 6.
- 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.

The following figure illustrates step (8b).

Link_Metrics_Table									
Link_Metrics Detailed View									
Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided		Bytes_transmitted(bytes)	Payload_transmitted(bytes)
		Data	Control	Data	Control	Data	Control		
All	NA	2359	4388	3	1	0	0	3884538	3442146
1	Link_throughput	473	486	0	0	0	0	756426	692966
2	Link_throughput	473	497	2	0	0	0	752932	687660
3	Link_throughput	471	498	0	0	0	0	750108	687660
4	Link_throughput	471	497	0	1	0	0	750032	687660
5	Link_throughput	471	473	1	0	0	0	749972	686200
6	Link_throughput	0	968	0	0	0	0	62416	0
7	Link_throughput	0	969	0	0	0	0	62652	0

- 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 SDN_Controller to Router_3.

9. Use RIP as the routing protocol and Router_3 as the SDN controller. To do so:
- Right-click Router_3 and click **Properties**. The Router pop-up window appears.
 - Click **APPLICATION_LAYER** in the left area.
 - Select **TRUE** from the SDN_Controller drop-down list.
 - Select **RIP** from the Routing Protocol drop-down list.
 - Click **OK**.

10. Configure static routes on Router_3 as follows:

- Simulate SDN (Refer step 6).
- Right-click **Router_3** and click NetSim Console.
Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe. NetSim CLI console opens.
- 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 > Router_7 > Router_6 > Router 6.

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

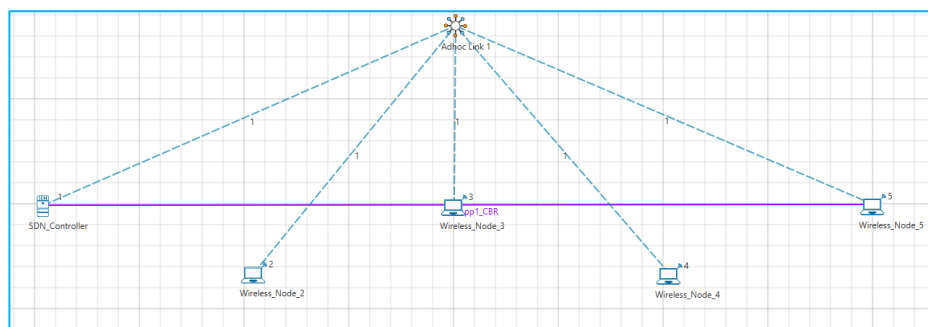
In this example static routes are configured using the SDN controllers, in a MANET network.

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.
- AODV 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 SDN_Controller 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**. The following network diagram illustrates the network setup:



2. AODV 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 1 as SDN controller and other nodes as its clients.

Device Name	Open_Flow	SDN_Controller	SDN_Controller_DeviceName
Wireless_Node_1	Enable	TRUE	NA
Wireless_Node_2	Enable	FALSE	SDN_Controller
Wireless_Node_3	Enable	FALSE	SDN_Controller
Wireless_Node_4	Enable	FALSE	SDN_Controller
Wireless_Node_5	Enable	FALSE	SDN_Controller

6. In NetSim GUI Plota re 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:

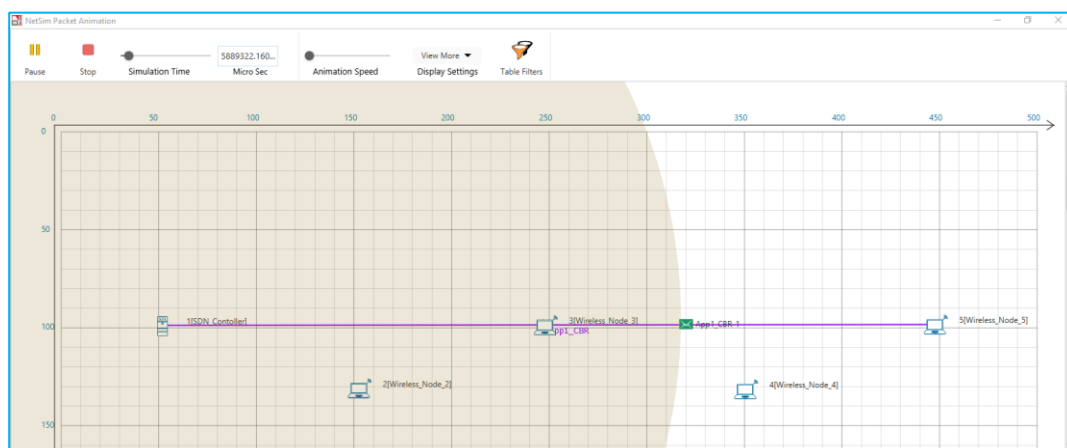
- a. Right-click **SDN_Controller** and select NetSim Console. Now, the client (NetSimCLI.exe) starts and attempts to establish a connection with NetSimCore.exe.

9. Interpret the results.

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

The packets reach Wireless_Node_5 directly from SDN_Controller.

The following figure illustrates step (9a).



10. Configure static routes such that the packets will go through `Wireless_Node_2 > Wireless_Node_4`, to `Wireless_Node_5`.

To configure the static routes on `SDN_Controller` and `Wireless_Node_2`:

- 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.
 - **SDN_Controller 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 11.1.1.4 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_2 > Wireless_Node_4`.
- 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 `SDN_Controller` to `Wireless_Node_2` 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.

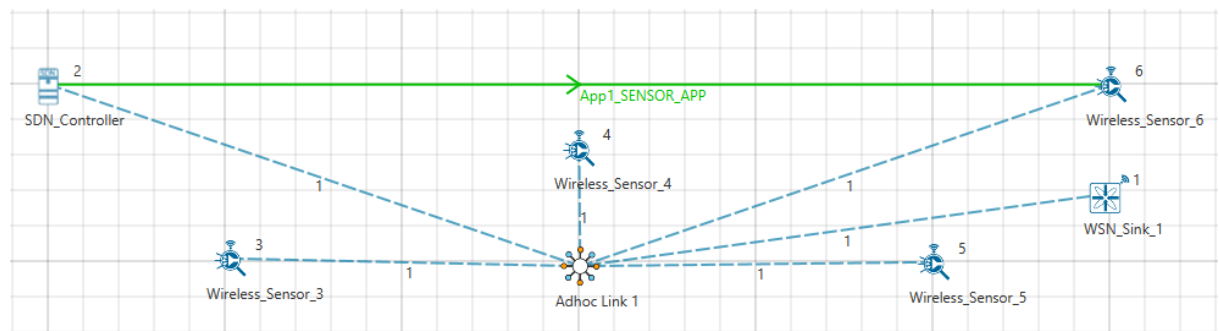
The WSN network model in this example consists of the following configuration:

- 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 SDN_Controller to Wireless_Sensor_6.
- 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**. The following network diagram illustrates the network setup considered in this example:



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

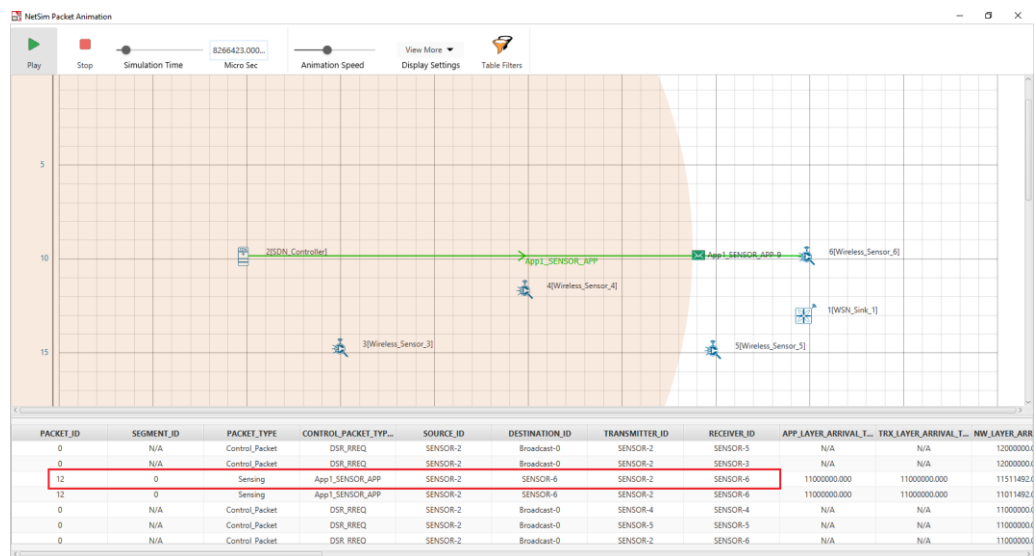
Device Name	Open_Flow	SDN_Controller	SDN_Controller_DeviceName
Wireless_Sensor_2	Enable	TRUE	NA
Wireless_Sensor_3	Enable	FALSE	SDN_Controller
Wireless_Sensor_4	Enable	FALSE	SDN_Controller
Wireless_Sensor_5	Enable	FALSE	SDN_Controller
Wireless_Sensor_6	Enable	FALSE	SDN_Controller
WSN_Sink_1	Enable	FALSE	SDN_Controller

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:
 - b. 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_6 directly from SDN_Controller.



7. Configure static routes such that the packets will go through Wireless_Sensor_4 to Wireless_Sensor_6.

To configure the static routes on SDN_Controller and Wireless_Sensor_4:

- 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.
 - **SDN_Controller route ADD 11.1.0.0 MASK 255.255.0.0 11.1.1.4 METRIC 1 IF 1**

- **Wireless_Sensor_4 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_6 via Wireless_Sensor_4.

- c. 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 SDN_Controller to Wireless_Sensor_4 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>.