Software Defined Networks

Contents

1 About SDN

Software-Defined Networking (SDN) is a network architecture that makes networks agile and flexible and enables enterprises and service providers to respond quickly to the changing business requirements.

The salient features of SDN architecture are:

- **Directly programmable**: SDN decouples the network control and forwarding functions. SDN allows you to program your network control and abstracts the physical infrastructure for applications and network services.

- **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, MME, 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.

Note: To get detailed help about how to use CLI commands in NetSim, see 2.4.8 NetSim Interactive Simulation in User Manual.

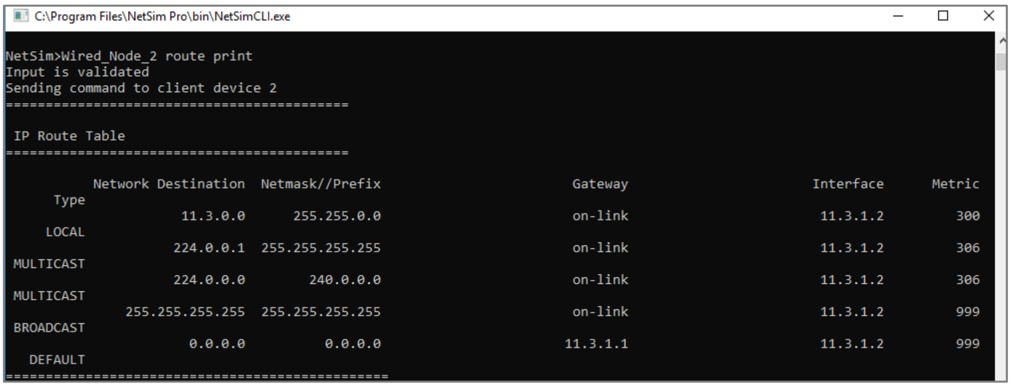
2.3 How to use CLI Commands for SDN

Users can set CLI commands on 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**.

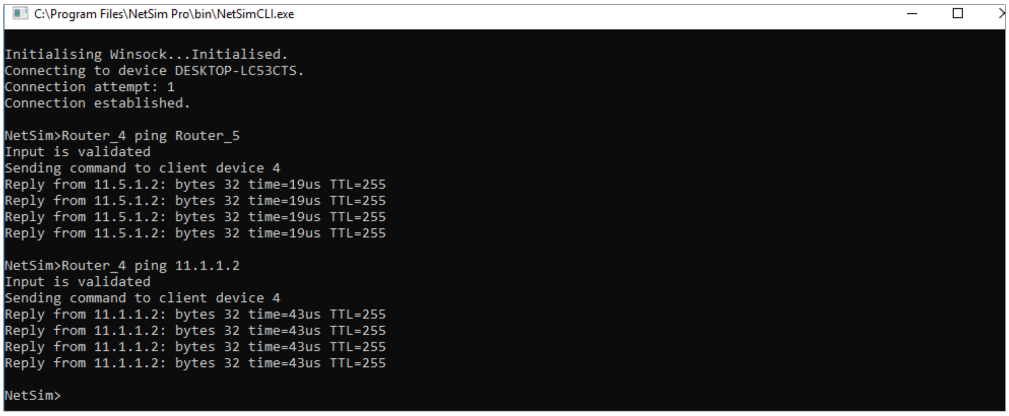
The following image illustrates the output for the command.



```
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
                      11.3.0.0      255.255.0.0                        on-link                        11.3.1.2        300
     LOCAL
                     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
 MULTICAST
               255.255.255.255  255.255.255.255                        on-link                        11.3.1.2        999
 BROADCAST
                       0.0.0.0          0.0.0.0                       11.3.1.1                        11.3.1.2        999
   DEFAULT
================================================
```

Example 2: To ping a node from another node where both nodes are controlled by the SDN controller, use one the following command syntaxes: ***<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**.



```
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 perform the ping cmd 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 contains some example configuration files to simulate and understand how SDN works. To simulate these examples, click **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

You configure one SDN controller in a simple Internetwork and simulate SDN from an example configuration file to see how data is transmitted in an SDN-based Internetwork.

The Internetwork you model from the example configuration file meets the following specifications:

- A subnet with 2 wired nodes, 3 routers, and a unicast application running on one of the wired nodes.
- TCP protocol is enabled in the Transport layer on all wired nodes and routers.
- Only one wired node is configured as the SDN controller.
- Open Flow protocol is enabled on all wired nodes and routers.

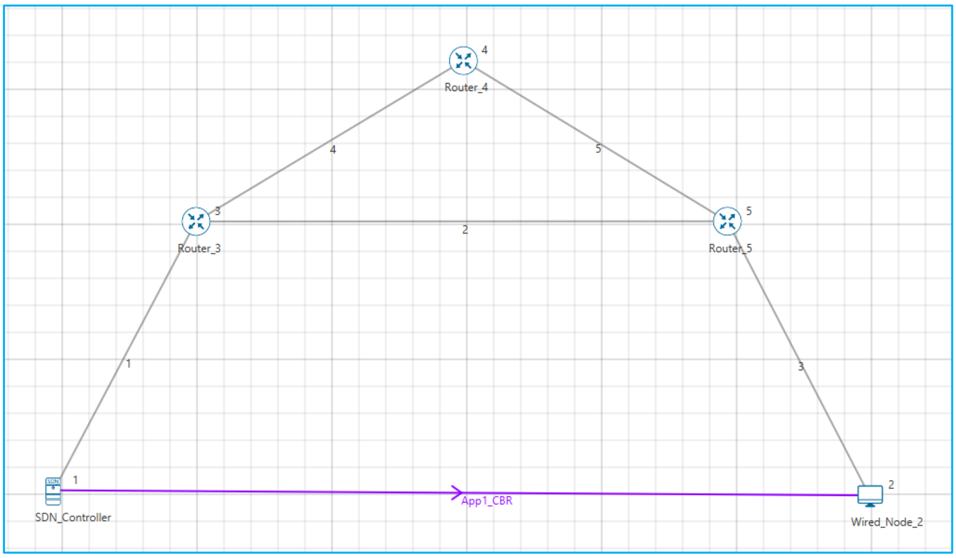
NetSim uses the following defaults for this SDN example:

- The unicast application transmits data at a constant bitrate and from SDN_Controller to Wired_Node_2.
- Simulation runs for 500 seconds.
- Plots, Packet trace, and Event trace is 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. See that by default, NetSim has enabled SDN_Controller as the SDN controller and enabled the Open Flow protocol. To do so:

   a. Right-click **SDN_Controller** and click **Properties.** The Wirednode pop-up window appears.

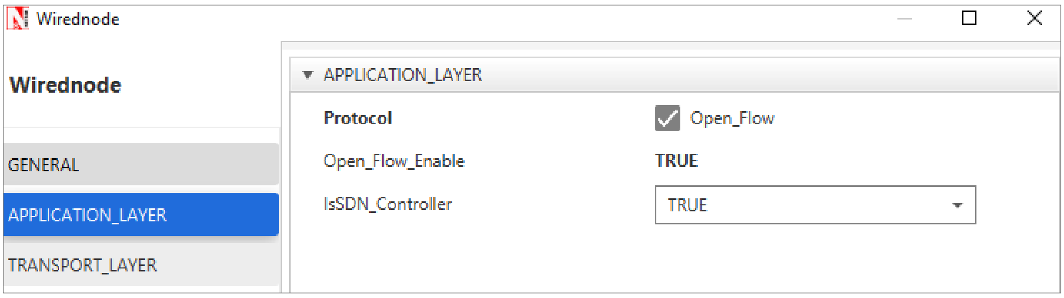
   b. Click **APPLICATION_LAYER** in the left area.

   c. **SDN_Controller** drop-down list is set to **TRUE**.

   d. NetSim has selected the **Open_Flow** check box.

   e. Click **OK**.

   The following image illustrates the Wirednode pop-up window and the default settings.



Protocol	☑ Open_Flow
Open_Flow_Enable	TRUE
IsSDN_Controller	TRUE

3. See that by default, NetSim has enabled the Open Flow protocol on Wired_Node_2 and the 3 routers. To do so:

   a. Right-click **Wired_Node_2** and click **Properties.** The Wirednode pop-up window appears.

   b. Click **APPLICATION_LAYER** in the left area.

   c. NetSim has selected the **Open_Flow** check box.

   d. Click **OK**.

e. Repeat steps (a) to (d) for the 3 routers.

4. See that by default, NetSim has specified SDN_Controller as the SDN controller node for Wired_Node_2 and the 3 routers. To do so:

   a. Right-click **Wired_Node_2** and click **Properties.** The Wirednode pop-up window appears.

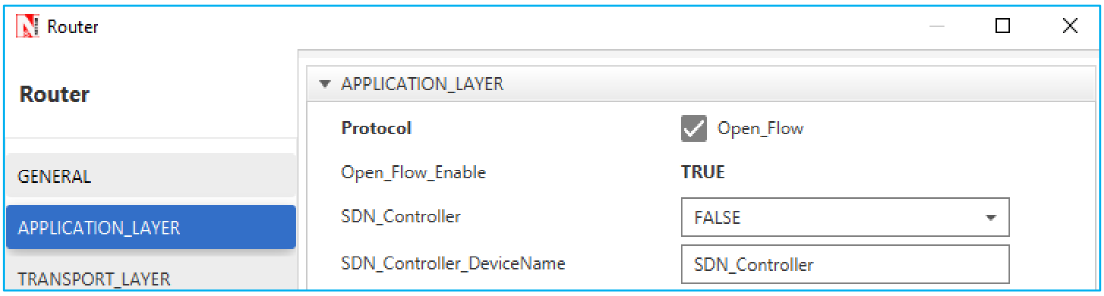
   b. Click **APPLICATION_LAYER** in the left area.

   c. NetSim has specified SDN_Controller in the SDN_Controller_DeviceName field.

   d. Click **OK**.

   e. Repeat steps (a) to (d) for the 3 routers.

The following image illustrates the Router pop-up window and the default settings.



Protocol ☑ Open_Flow

Open_Flow_Enable TRUE

SDN_Controller FALSE

SDN_Controller_DeviceName SDN_Controller

The following table lists the default setting for the nodes and the routers.

Device Name	Open_Flow	IsSDN_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

5. See that by default, NetSim has configured the unicast application and specified SDN_Controller as the source of the application and Wired_Node_2 as the destination. To do so:

   a. Click **Application** in the toolbar**.** The Configure Application pop-up window appears.

   b. NetSim has specified UNICAST in the Application_Method field.

   c. NetSim has specified CBR in the Application_Type field.

    d. NetSim has specified 1 in the Source_ID field.
    e. NetSim has specified 2 in the Destination_ID field.
    f. Click **OK**.

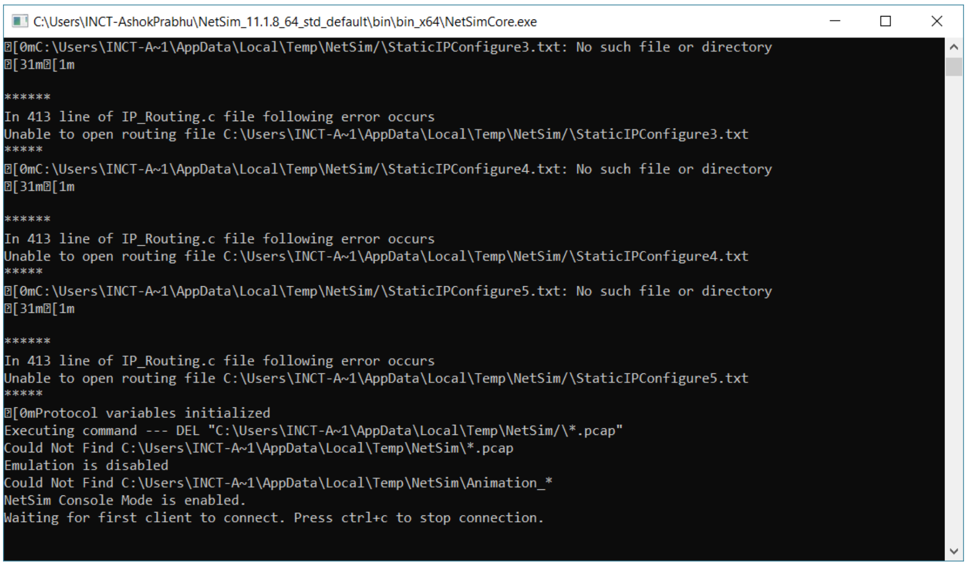
6. See that by default, NetSim has enabled the Plots, Packet Trace and Event Trace icons located in the toolbar.

7. Simulate the example. To do so:

    a. Click the Run icon located in the toolbar. The Run Simulation pop-up window appears.

    b. Retain the default settings in the Simulation Configuration tab (Simulation Time = 500)
    c. Retain the default settings in the Run time Interaction tab (Interactive Simulation = 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.

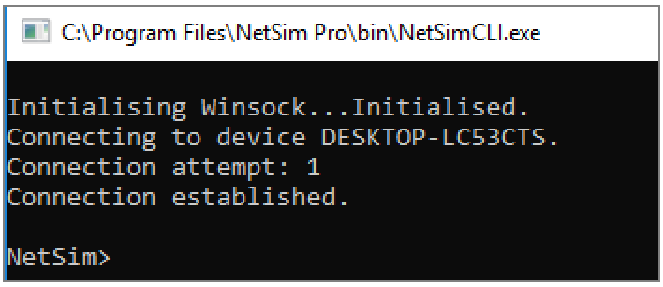

C:\Users\INCT-AshokPrabhu\NetSim_11.1.8_64_std_default\bin\bin_x64\NetSimCore.exe

```
▯[0mC:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure3.txt: No such file or directory
▯[31m▯[1m

******
In 413 line of IP_Routing.c file following error occurs
Unable to open routing file C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure3.txt
*****
▯[0mC:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure4.txt: No such file or directory
▯[31m▯[1m

******
In 413 line of IP_Routing.c file following error occurs
Unable to open routing file C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure4.txt
*****
▯[0mC:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure5.txt: No such file or directory
▯[31m▯[1m

******
In 413 line of IP_Routing.c file following error occurs
Unable to open routing file C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure5.txt
*****
▯[0mProtocol variables initialized
Executing command --- DEL "C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\*.pcap"
Could Not Find C:\Users\INCT-A~1\AppData\Local\Temp\NetSim\*.pcap
Emulation is disabled
Could Not Find C:\Users\INCT-A~1\AppData\Local\Temp\NetSim\Animation_*
NetSim Console Mode is enabled.
Waiting for first client to connect. Press ctrl+c to stop connection.
```

8. To use the SDN CLI Console:

   a. Right-click **SDN_Controller** 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.



```
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

You configure multiple SDN controllers in a Wireless Sensor Network and simulate SDN from an example configuration file to see how data is transmitted in an SDN-based Wireless Sensor Network.

The Wireless Sensor Network you model from the example configuration file meets the following specifications:

- 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.
- TCP protocol is enabled in the Transport layer on all wireless sensors and sink node.
- 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:

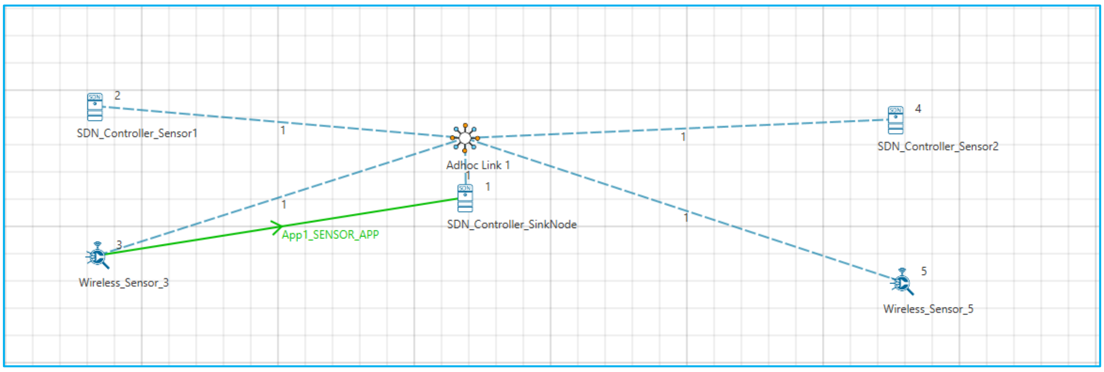
- The unicast application transmits data from Wireless Sensor_3 to SDN_Controller_SinkNode.
- Simulation runs for 200 seconds.
- Plots, Packet trace, and Event trace is enabled.

Note: For a WSN network, SDN commands are NOT supported. Because, the sensor nodes do not support for ICMP protocol, ping command is not supported on sensor nodes.

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 what the NetSim UI displays when you open the example configuration file for SDN.



2. The following Environment properties are already set, Grid Length as 500m and Side Length as 250m.

3. See that by default, NetSim has enabled SDN_Controller_Sensor1 and SDN_Controller_Sensor2 as the SDN controllers and enabled the Open Flow protocol. To do so:

   a. Right-click **SDN_Controller_Sensor1** and click **Properties.** The Sensors pop-up window appears.

   b. Click **APPLICATION_LAYER** in the left area.

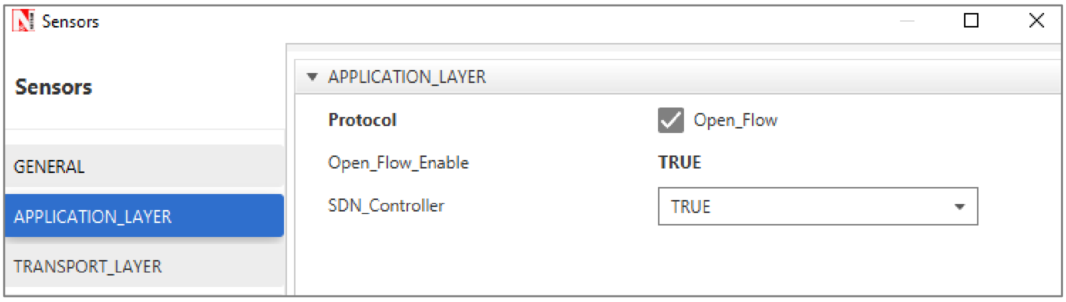
   c. **SDN_Controller** drop-down list is set to **TRUE**.

   d. NetSim has selected the **Open_Flow** check box.

   e. Click **OK**.

   f. Repeat steps (a) to (e) for **SDN_Controller_Sensor2**.

The following image illustrates the Sensor pop-up window and the default settings.



Sensors

Sensors

GENERAL

APPLICATION_LAYER

TRANSPORT_LAYER

APPLICATION_LAYER

Protocol	☑ Open_Flow
Open_Flow_Enable	**TRUE**
SDN_Controller	TRUE ▾

4. See that by default, NetSim has enabled the Open Flow protocol on Wireless_Sensor_3, and Wireless_Sensor_5 and the WSN sink. To do so:

   a. Right-click **Wired_Node_3** and click **Properties.** The Wirednode pop-up window appears.

   b. Click **APPLICATION_LAYER** in the left area.

   c. NetSim has selected the **Open_Flow** check box.

   d. Click **OK**.

   e. Repeat steps (a) to (d) for Wireless_Sensor_5, and the WSN sink.

5. See that by default, NetSim has specified SDN_Controller_Sensor1 as the SDN controller for Wireless_Sensor_3 and SDN_Controller_Sensor2 as the SDN controller for Wireless_Sensor_5. To do so:

   a. Right-click **Wireless_Sensor_3** and click **Properties.** The Wirednode pop-up window appears.

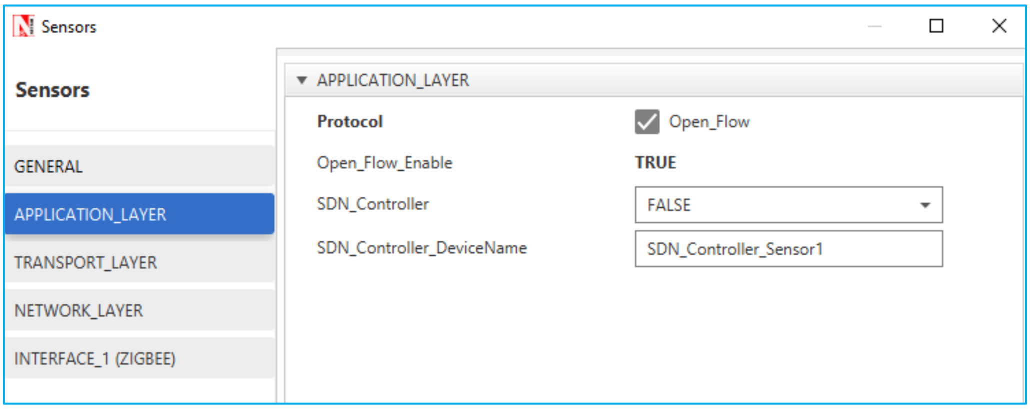
   b. Click **APPLICATION_LAYER** in the left area.

   c. NetSim has specified **SDN_Controller_Sensor1** in the SDN_Controller_DeviceName field.

   d. Click **OK**.

   e. Repeat steps (a) to (d) for Wireless_Sensor_5.

The following image illustrates the Sensors pop-up window and the default settings.



Protocol	☑ Open_Flow
Open_Flow_Enable	**TRUE**
SDN_Controller	FALSE
SDN_Controller_DeviceName	SDN_Controller_Sensor1

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

Device Name	Open_Flow	IsSDN_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

6. See that by default, NetSim has configured the unicast application and specified Wireless_Sensor_3 as the source of the sensor application and SDN_Controller_SinkNode as the destination. To do so:

    a. Click **Application** in the toolbar**.** The Configure Application pop-up window appears.

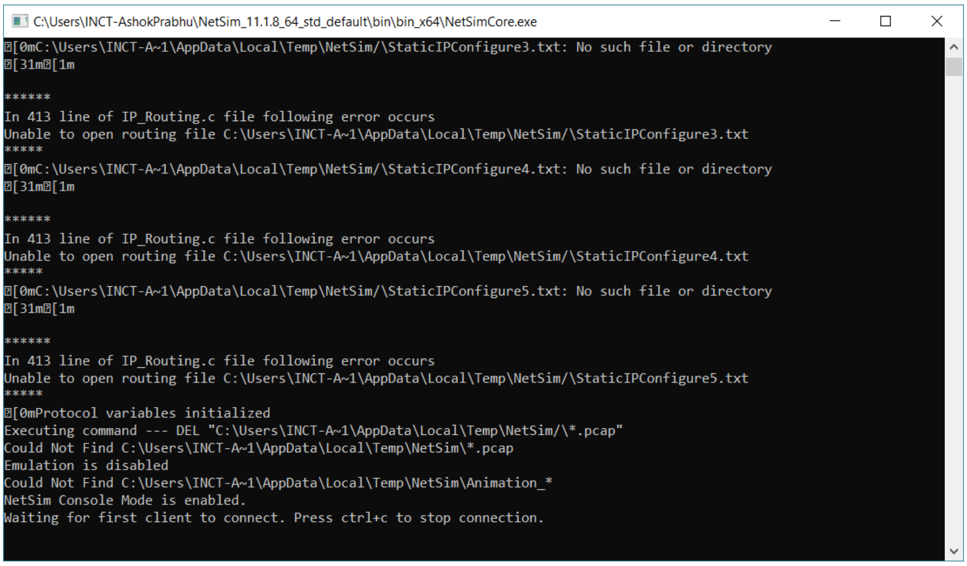
    b. NetSim has specified UNICAST in the Application_Method field.
    c. NetSim has specified SENSOR_APP in the Application_Type field.
    d. NetSim has specified 3 in the Source_ID field.
    e. NetSim has specified 1 in the Destination_ID field.
    f. Click **OK**.

7. See that by default, NetSim has enabled the Plots, Packet Trace and Event Trace icons located in the toolbar.

8. Simulate the example. To do so:

    a. Click the Run icon located in the toolbar. The Run Simulation pop-up window appears.

    b. Retain the default settings in the Simulation Configuration tab (Simulation Time = 200)
    c. Retain the default settings in the Run time Interaction tab (Interactive Simulation = 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.



C:\Users\INCT-AshokPrabhu\NetSim_11.1.8_64_std_default\bin\bin_x64\NetSimCore.exe

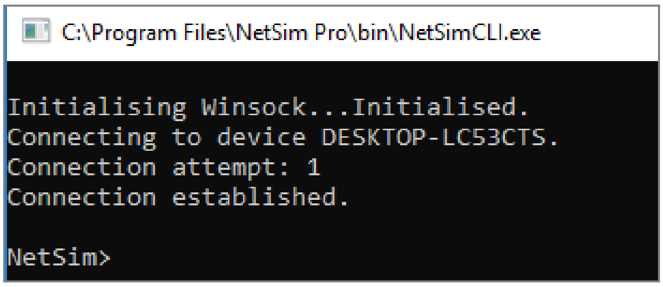
```
[0mC:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure3.txt: No such file or directory
[31m[1m

******
In 413 line of IP_Routing.c file following error occurs
Unable to open routing file C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure3.txt
*****
[0mC:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure4.txt: No such file or directory
[31m[1m

******
In 413 line of IP_Routing.c file following error occurs
Unable to open routing file C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure4.txt
*****
[0mC:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure5.txt: No such file or directory
[31m[1m

******
In 413 line of IP_Routing.c file following error occurs
Unable to open routing file C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\StaticIPConfigure5.txt
*****
[0mProtocol variables initialized
Executing command --- DEL "C:\Users\INCT-A~1\AppData\Local\Temp\NetSim/\*.pcap"
Could Not Find C:\Users\INCT-A~1\AppData\Local\Temp\NetSim\*.pcap
Emulation is disabled
Could Not Find C:\Users\INCT-A~1\AppData\Local\Temp\NetSim\Animation_*
NetSim Console Mode is enabled.
Waiting for first client to connect. Press ctrl+c to stop connection.
```

9. 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>
```

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

You change the IP tables in the nodes and routers on an SDN network from an example configuration file, to understand how to use the SDN CLI commands.

The network you model from the example configuration file meets the following specifications:

- 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.

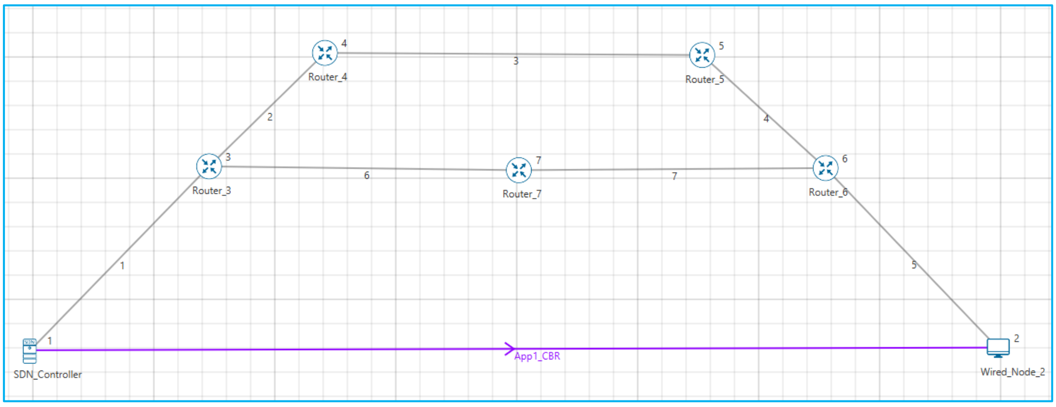
- TCP protocol is enabled in the Transport layer on all wired nodes and routers.
- Open Flow protocol is enables on all wired nodes and routers.
- Only one node can be the SDN controller.

NetSim uses the following defaults for this SDN example:

- The unicast application transmits data at a constant bitrate and from SDN_Controller to Wired_Node_2.
- OSPF is the routing protocol.
- SDN_Controller is the SDN controller.
- Simulation runs for 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 what the NetSim UI displays when you open the example configuration file for SDN.



2. See that by default, NetSim has assigned the following IP addresses to the interfaces of the nodes and routers. To do so:

   a. Right-click **SDN_Controller** and click **Properties.** The Wirednode pop-up window appears.

   b. Click **INTERFACE_1** in the left area.

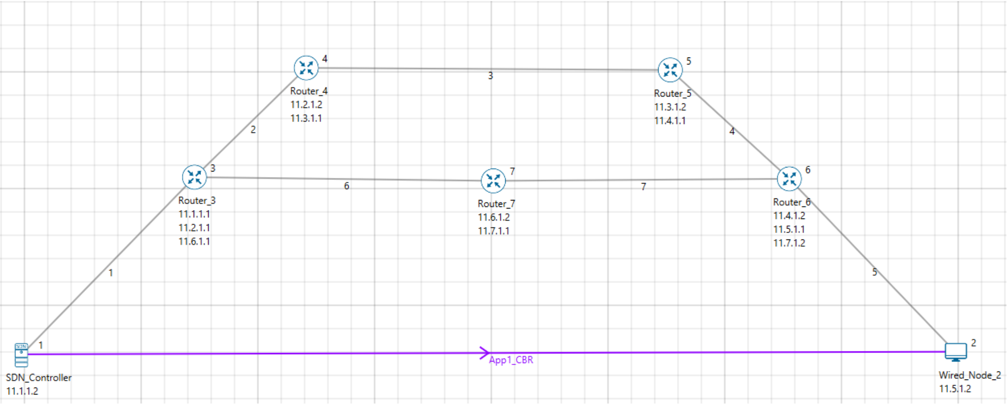
   c. Click **NETWORK_LAYER** in the right area.

   d. NetSim has assigned the following values: IP_Address = 11.1.1.2, Subnet_Mask = 255.255.0.0, and Default_Gateway = 11.1.1.1.

   e. Click **OK**.

   f. Repeat step (a) to (e) for Wired_Node_2 and the routers.

The following network diagram illustrates the IP addresses assigned to the nodes' and routers' interfaces.



3. See that by default NetSim has configured the default settings for SDN controller and Open_Flow on all nodes and routers.

   For more information about seeing the default settings, see the Configuring SDN in NetSim section.

4. See that by default, NetSim has configured the unicast application and specified the default settings. To do so:

   a. Click **Application** in the toolbar**.** The Configure Application pop-up window appears.

   b. NetSim has specified UNICAST in the Application_Method field.

   c. NetSim has specified CBR in the Application_Type field.

   d. NetSim has specified 1 in the Source_ID field.

   e. NetSim has specified 2 in the Destination_ID field.

   f. NetSim has specified 30 in the Start_Time(s) field.

   g. NetSim has specified 1000000 in the Value (micro sec) field, in the INTER ARRIVAL TIME area.

   h. Click **OK**.

5. See that by default, NetSim has enabled the Plots, Packet Trace and Event Trace icons located in the toolbar.

6. Simulate SDN. To do so:

   a. Click the Run icon located in the toolbar. The Run Simulation pop-up window appears.

b. Retain the default settings in the Simulation Configuration tab (Simulation Time = 500)

c. Retain the default settings in the Run time Interaction tab (Interactive Simulation = True).

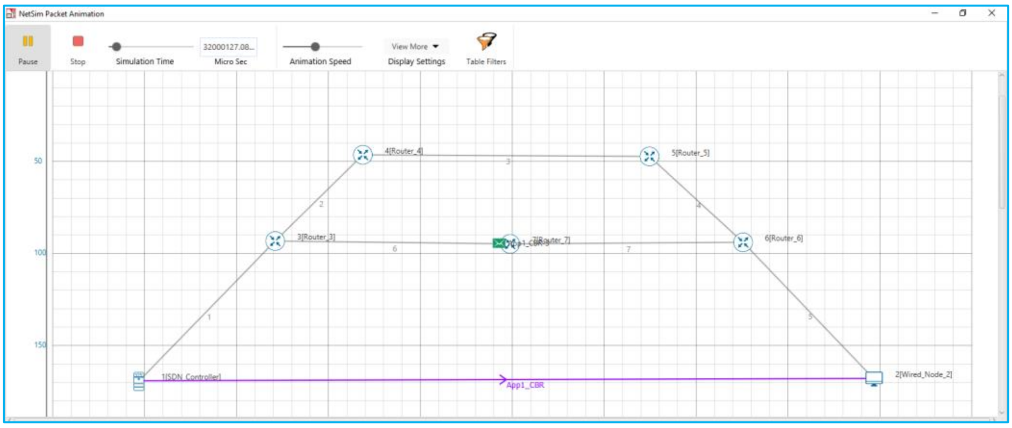
7. Right-click **SDN_Controller** and click NetSim Console. NetSim simulates SDN.

8. 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.

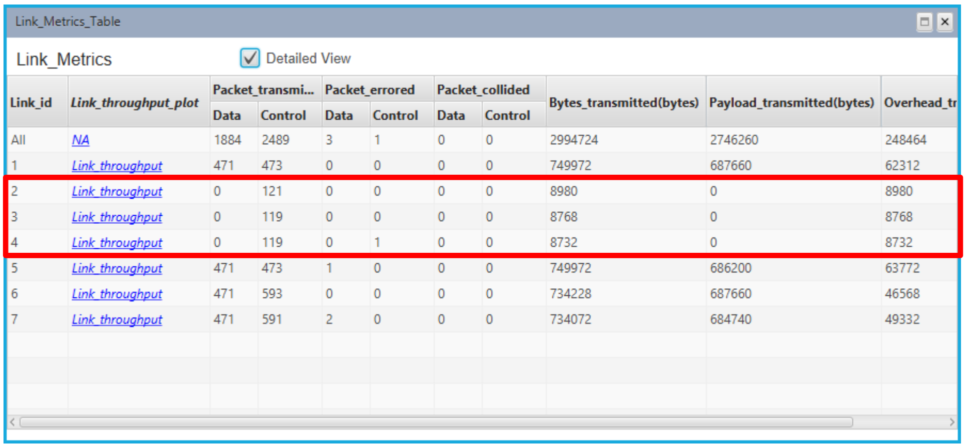
The following figure illustrates step (8a).



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 (8b).



Link_Metrics_Table

Link_Metrics ☑ Detailed View

Link_id	Link_throughput_plot	Packet_transmi...		Packet_errored		Packet_collided		Bytes_transmitted(bytes)	Payload_transmitted(bytes)	Overhead_tr
		Data	Control	Data	Control	Data	Control			
All	NA	1884	2489	3	1	0	0	2994724	2746260	248464
1	Link_throughput	471	473	0	0	0	0	749972	687660	62312
2	Link_throughput	0	121	0	0	0	0	8980	0	8980
3	Link_throughput	0	119	0	0	0	0	8768	0	8768
4	Link_throughput	0	119	0	1	0	0	8732	0	8732
5	Link_throughput	471	473	1	0	0	0	749972	686200	63772
6	Link_throughput	471	593	0	0	0	0	734228	687660	46568
7	Link_throughput	471	591	2	0	0	0	734072	684740	49332

9. 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 SDN (Refer step 6).

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

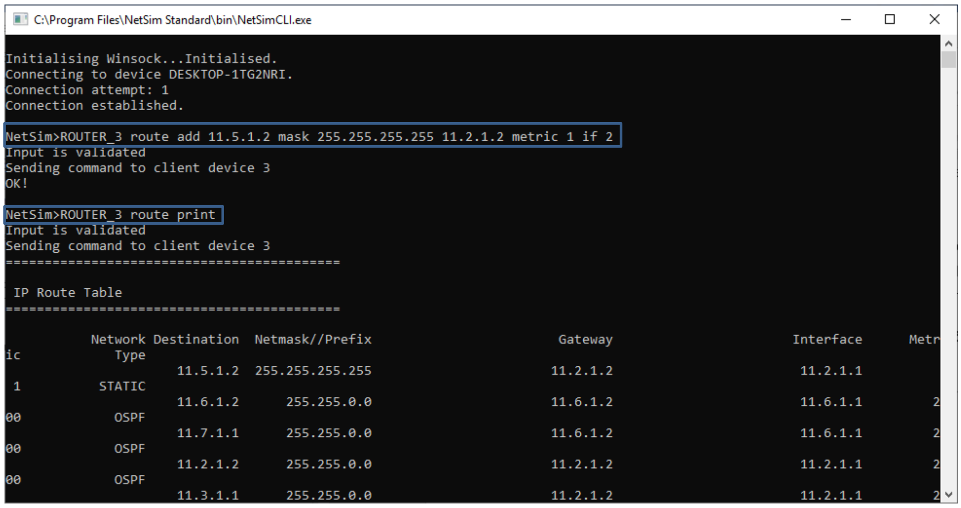
   Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe.

   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 11.2.1.2 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).



```
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
===========================================

           Network Destination  Netmask//Prefix                        Gateway                  Interface      Metr
ic            Type
                      11.5.1.2  255.255.255.255                       11.2.1.2                   11.2.1.1
 1          STATIC
                      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
```

10. Interpret the results.

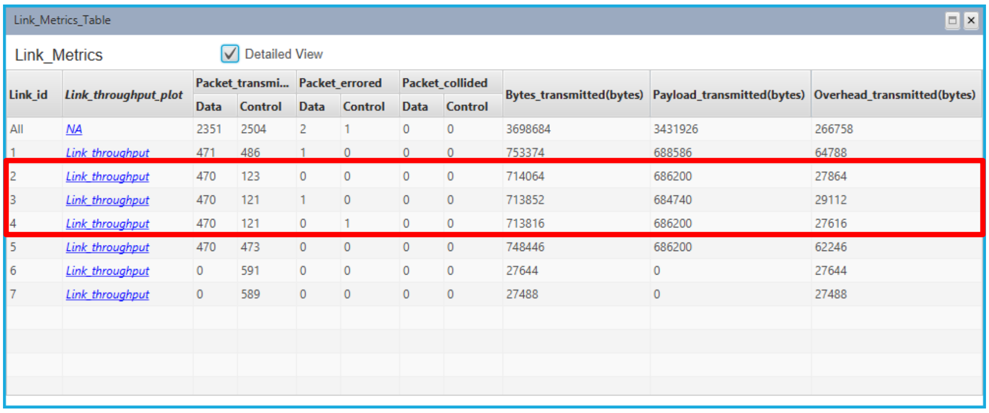
a. Click **View Animation** and see the Packet Animation.

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 not see data packet transmissions in Links 2, 3, and 4.

   The following figure illustrates step (10b).



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*	2351	2504	2	1	0	0	3698684	3431926	266758
1	*Link_throughput*	471	486	1	0	0	0	753374	688586	64788
2	*Link_throughput*	470	123	0	0	0	0	714064	686200	27864
3	*Link_throughput*	470	121	1	0	0	0	713852	684740	29112
4	*Link_throughput*	470	121	0	1	0	0	713816	686200	27616
5	*Link_throughput*	470	473	0	0	0	0	748446	686200	62246
6	*Link_throughput*	0	591	0	0	0	0	27644	0	27644
7	*Link_throughput*	0	589	0	0	0	0	27488	0	27488

d. Click **View Results** on the tool bar, **Open Packet trace** in the left area and filter the **CONTROL_PACKET_TYPE/APP_NAME** column by **OPENFLOW_COMMAND** and **OPENFLOW_RESPONSE**.

You will see that OpenFlow packets flow between SDN_Controller to Router_3.

11. Use RIP as the routing protocol and Router_3 as the SDN controller. To do so:

a. Right-click Router_3 and click **Properties**. The Router pop-up window appears.

b. Click **APPLICATION_LAYER** in the left area.

c. Select **TRUE** from the SDN_Controller drop-down list.

d. Select **RIP** from the Routing Protocol drop-down list.

e. Click **OK**.

12. Configure static routes on Router_3 as follows:

a. Simulate SDN (Refer step 6).

b. 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.

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**

13. 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

You configure static routes from the SDN controllers, in a MANET network to see how data flows through the static routes you specified from the SDN controllers.

The SDN-based MANET network you model from the example configuration file meets the following specifications:

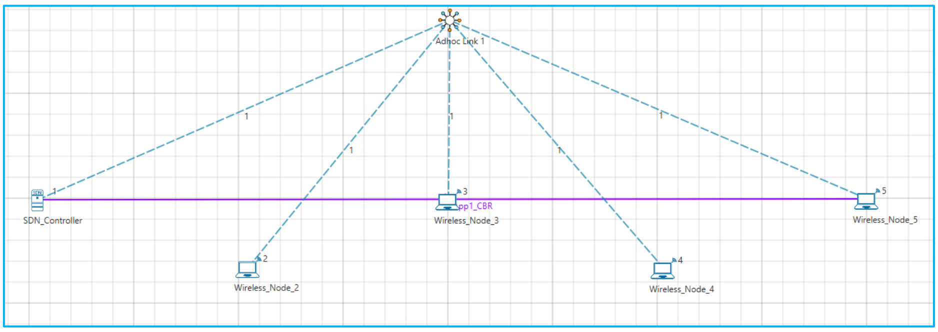
- A subnet with 4 wireless nodes and 1 ad-hoc link, and a unicast CBR application running on one of the wireless nodes.
- AODV is the routing protocol that is enabled on all wireless nodes.
- One wireless node is configured as the SDN controller.
- Wireless nodes do not go mobile.

NetSim uses the following defaults for this example:

- The unicast application transmits data at a constant bitrate and from SDN_Controller to Wireless_Node_5.
- Simulation runs for 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 what the NetSim UI displays when you open the example configuration file for SDN.



2. See that by default, NetSim has enabled the AODV protocol in the Network layer for all wireless nodes and set the nodes to be stationary. To do so:

a. Right-click **SDN_Controller** and click **Properties.** The Wirelessnode pop-up window appears.

b. NetSim has selected **NO_MOBILITY** in the Mobility_Model drop-down list, in the right area.

c. Click **NETWORK_LAYER** in the left area.

d. NetSim has selected TRUE in the ICMP_Status drop-down list.

e. NetSim has selected AODV in the Routing Protocol drop-down list.

f. Click **OK**.

g. Repeat steps (a) to (f) for other wireless nodes.

3. See that by default, NetSim has enabled SDN_Controller as the SDN controller and enabled the Open Flow protocol on all wireless nodes. To do so:

a. Right-click **SDN_Controller** and click **Properties.** The Wirelessnode pop-up window appears.

b. Click **APPLICATION_LAYER** in the left area.

c. **SDN_Controller** drop-down list is set to **TRUE**.

d. NetSim has selected the **Open_Flow** check box.

e. Click **OK**.

f. Repeat steps (a), (b), (d), and (e) for other wireless nodes.

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

Device Name	**Open_Flow**	**IsSDN_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

4. See that by default, NetSim has enabled some default settings for the ad-hoc link. To do so:

a. Right-click **Adhoc Link_1** and click **Properties.** The Link Properties pop-up window appears.

b. NetSim has selected PATHLOSS_ONLY in the Channel_Characteristics drop-down list.

c. NetSim has selected LOG_DISTANCE in the Path Loss Model drop-down list.

d. NetSim has specified 2 in the PathLoss_Exponent(n) field.

e. Click **OK**.

5. See that by default, NetSim has configured the unicast application and specified SDN_Controller as the source of the application and Wirless_Node_5 as the destination. To do so:

a. Click **Application** in the toolbar**.** The Configure Application pop-up window appears.

b. NetSim has specified UNICAST in the Application_Method field.

c. NetSim has specified CBR in the Application_Type field.

d. NetSim has specified 1 in the Source_ID field.

e. NetSim has specified 5 in the Destination_ID field.

f. Click **OK**.

6. See that by default, NetSim has enabled the Packet Trace and Event Trace icons located in the toolbar.

7. Simulate the example. To do so:

a. Click the Run icon located in the toolbar. The Run Simulation pop-up window appears.

b. Retain the default settings in the Simulation Configuration tab (Simulation Time = 100)

c. Retain the default settings in the Run time Interaction tab (Interactive Simulation = 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:

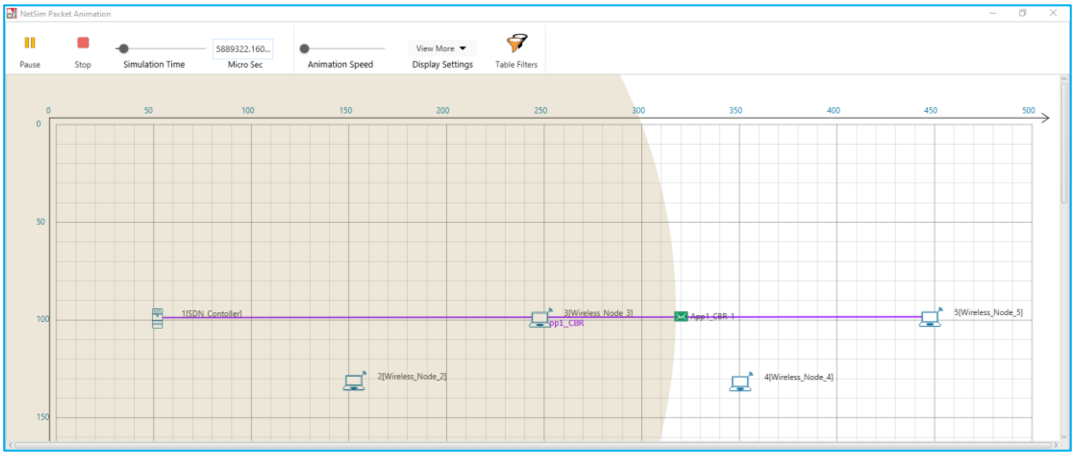
a. Right-click **SDN_Controller** and click NetSim Console. Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe.

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. Simulate SDN (Refer step 7).

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 11.1.1.5 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. Click **View Results** on the tool bar, **Open Packet trace** in the left area and filter the **CONTROL_PACKET_TYPE/APP_NAME** column by **OPENFLOW_COMMAND** and **OPENFLOW_RESPONSE**.

   You will see that OpenFlow packets flow between SDN_Controller to Wireless_Node_2.

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

You configure static routes from the SDN controllers, in a WSN network to see how data flows through the static routes you specified from the SDN controllers.

The SDN-based WSN network you model from the example configuration file meets the following specifications:

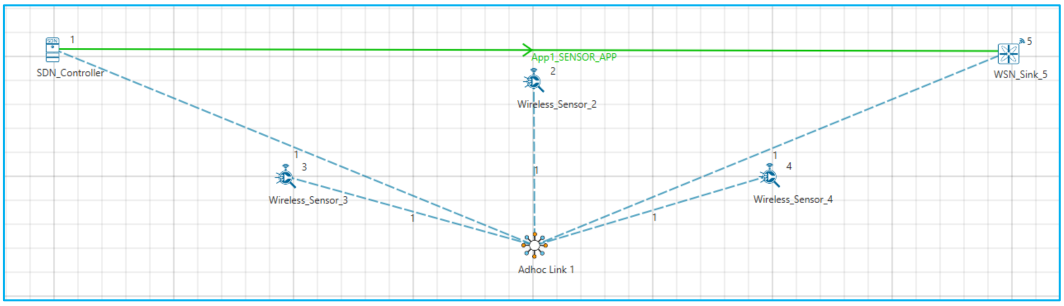
- A subnet with 4 wireless sensor, 1 ad-hoc link, 1 wireless sink node, and a unicast sensor application running on one of the wireless sensors.
- One wireless node is configured as the SDN controller.

NetSim uses the following defaults for this example:

- The unicast application transmits data from SDN_Controller to WSN_Sink_5.
- Simulation runs for 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 what the NetSim UI displays when you open the example configuration file for SDN.



2. The following Environment properties are already set, Grid Length as 500m and Side Length as 250m.

3. See that by default, NetSim has enabled SDN_Controller device as the SDN controller and enabled the Open Flow protocol on all wireless nodes. To do so:

   a. Right-click **SDN_Controller** and click **Properties.** The Sensors pop-up window appears.
   b. Click **APPLICATION_LAYER** in the left area.
   c. **SDN_Controller** drop-down list is set to **TRUE**.
   d. NetSim has selected the **Open_Flow** check box.
   e. Click **OK**.
   f. Repeat steps (a), (b), (d), and (e) for other wireless sensors and the sink node. The following table lists the default setting for the wireless sensors.

Device Name	**Open_Flow**	**IsSDN_Controller**	**SDN_Controller_DeviceName**
Wireless_Sensor_1	Enable	TRUE	NA
Wireless_Sensor_2	Enable	FALSE	SDN_Controller
Wireless_ Sensor_3	Enable	FALSE	SDN_Controller
Wireless_ Sensor_4	Enable	FALSE	SDN_Controller
WSN_Sink_5	Enable	FALSE	SDN_Controller

4. See that by default, NetSim has configured the unicast application and specified SDN_Controller as the source of the application and WSN_Sink_5 as the destination. To do so:

   a. Click **Application** in the toolbar**.** The Configure Application pop-up window appears.
   b. NetSim has specified UNICAST in the Application_Method field.
   c. NetSim has specified SENSOR_APP in the Application_Type field.
   d. NetSim has specified 1 in the Source_ID field.
   e. NetSim has specified 5 in the Destination_ID field.
   f. Click **OK**.

5. See that by default, NetSim has enabled the Packet Trace icon located in the toolbar.

6. Simulate the example. To do so:

   a. Click the Run icon located in the toolbar. The Run Simulation pop-up window appears.

b. Retain the default settings in the Simulation Configuration tab (Simulation Time = 100)

c. Retain the default settings in the Run time Interaction tab (Interactive Simulation = 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.

7. To use the SDN CLI Console:

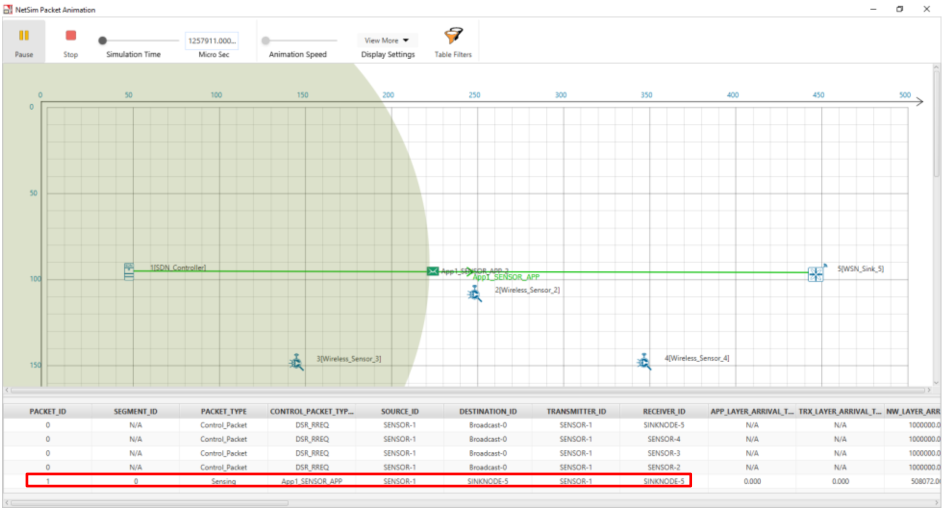
b. Right-click **SDN_Controller** and click NetSim Console. Now, the client (NetSimCLI.exe) starts and tries to establish a connection with NetSimCore.exe.

8. Interpret the results.

a. Click View Animation and see the Packet Animation.

The packets reach WSN_Sink_5 directly from SDN_Controller.

The following figure illustrates step (7a).



PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYP...	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_T...	TRX_LAYER_ARRIVAL_T...	NW_LAYER_ARR
0	N/A	Control_Packet	DSR_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SINKNODE-5	N/A	N/A	1000000.0
0	N/A	Control_Packet	DSR_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-4	N/A	N/A	1000000.0
0	N/A	Control_Packet	DSR_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-3	N/A	N/A	1000000.0
0	N/A	Control_Packet	DSR_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-2	N/A	N/A	1000000.0
1	0	Sensing	App1_SENSOR_APP	SENSOR-1	SINKNODE-5	SENSOR-1	SINKNODE-5	0.000	0.000	508072.0

9. Configure static routes such that the packets will go through Wireless_Sensor_2 to WSN_Sink_5.

To configure the static routes on SDN_Controller and Wireless_Sensor_2:

a. Simulate SDN (Refer step 5).

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 11.1.1.3 METRIC 1 IF 1**

- **Wireless_Sensor_2 route ADD 11.1.0.0 MASK 255.255.0.0 11.1.1.1 METRIC 1 IF 1**

10. Interpret the results.

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

The packets reach WSN_Sink_5 via Wireless_Sensor_2.

b. Click **View Results** on the tool bar, **Open Packet trace** in the left area and filter the **CONTROL_PACKET_TYPE/APP_NAME** column by **OPENFLOW_COMMAND** and **OPENFLOW_RESPONSE**.

You will see that OpenFlow packets flow between SDN_Controller to Wireless_Sensor_2.

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.