

NetSim[®]

Accelerate Network R & D

Software Defined Networks

A Network Simulation & Emulation Software

By



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

Copyright in the whole and every part of this manual belongs to TETCOS LLP and may not be used, sold, transferred, copied or reproduced in whole or in part in any manner or in any media to any person, without the prior written consent of TETCOS LLP. If you use this manual you do so at your own risk and on the understanding that TETCOS LLP shall not be liable for any loss or damage of any kind.

TETCOS LLP may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from TETCOS LLP, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Rev 14.0 (V), Oct 2023, TETCOS LLP. All rights reserved.

All trademarks are property of their respective owner.

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	4
2	SDN in NetSim	5
2.1	SDN Controller in NetSim	5
2.2	CLI Commands for SDN in NetSim.....	5
2.3	Configuring SDN in NetSim	6
2.4	How to use CLI Commands for SDN	6
2.5	Excluded Features.....	7
2.6	Troubleshooting issues in executing SDN commands	7
3	Featured Examples.....	9
3.1	Configuring SDN	9
3.1.1	Configuring One SDN Controller in a Simple Internetwork.....	9
3.1.2	Configuring Multiple SDN Controllers in a Wireless Sensor Network	13
3.2	Configuring Static Routes for a MANET Network by using SDN	19
3.3	Configuring Static Routes for Internetworks using SDN	23
3.4	Configuring Static Routes for a WSN Network by using SDN	28
4	Latest FAQs	32

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.

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

2.3 Configuring SDN in NetSim

SDN protocol parameters can be accessed and configured from the Application layer properties of the devices.

- Open-flow protocol must be enabled in the devices which are configured as SDN Controller or SDN clients.
- SDN Controller device must have the SDN Controller option set to True.
- SDN Client devices must have the SDN Controller option set to False along with the Controller Device Name set as the SDN Controller Device Name.
- TCP protocol must be enabled in the SDN Controller and client devices.
- Interactive Simulation option needs to be set to True in the Run time interaction tab that is part of the Run Simulation window.

When SDN commands are executed during runtime either via console or via file input, commands and response are exchanged between the controller and clients using TCP

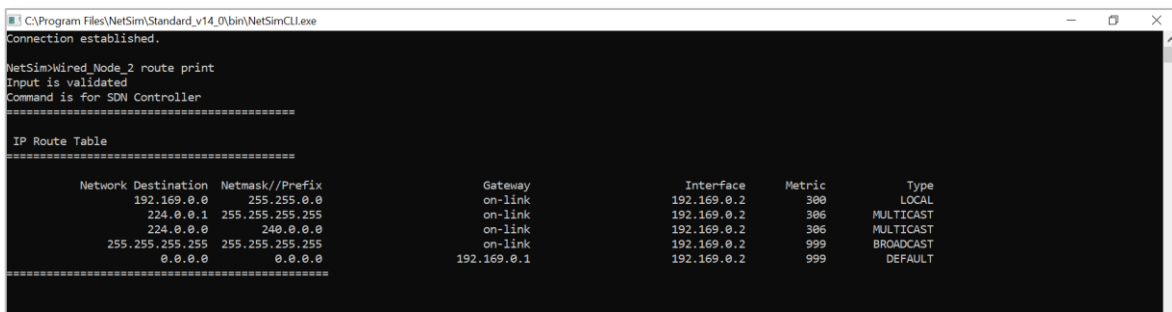
2.4 How to use CLI Commands for SDN

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\Standard_v14_0\bin\NetSimCLI.exe
Connection established.

NetSim>Wired_Node_2 route print
Input is validated
Command is for SDN Controller
=====
IP Route Table
=====
Network Destination  Netmask/Prefix          Gateway          InterFace      Metric      Type
-----
192.169.0.0          255.255.0.0            on-link         192.169.0.2    300         LOCAL
224.0.0.1            255.255.255.255       on-link         192.169.0.2    306         MULTICAST
224.0.0.0            240.0.0.0              on-link         192.169.0.2    306         MULTICAST
255.255.255.255     255.255.255.255       on-link         192.169.0.2    999         BROADCAST
0.0.0.0              0.0.0.0                192.169.0.1    192.169.0.2    999         DEFAULT
=====

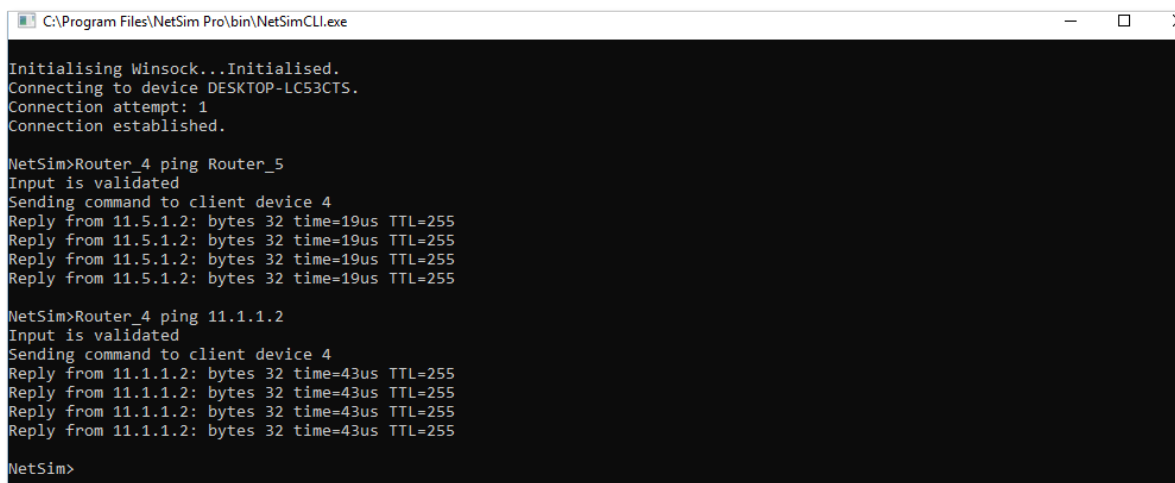
```

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.



```

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>

```

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.

2.5 Excluded Features

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

2.6 Troubleshooting issues in executing SDN commands

When executing SDN commands in the SDN controller, there can be cases where there is no response coming back. Some of the reasons for this are listed below:

- SDN Controller and SDN client are unable to communicate (not reachable).
 - Part of two different networks

- Due to the distance between the devices in the case of wireless nodes
- Failed TCP connection attempts due to high error or collision
 - Commands for Controller to the client and response from the client to the controller are passed as TCP packets.
 - The TCP connection attempt fails if the maximum connection attempt is reached.
- High network traffic due to application packets exchanged during the simulation

Some suggestions to resolve issues with SDN command execution are listed below:

- Connectivity between the SDN controller and clients should be ensured.
- Application start time can be increased to allow necessary SDN commands to be executed prior to application traffic flow which will also avoid contention especially in the case of Wireless devices.
- SDN commands can be passed as input via file-based interactive simulation for higher accuracy in terms of the time at which the commands are to be executed.

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 Configuring SDN

3.1.1 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:

Open NetSim and Select **Examples > Software Defined Networks > Configuring SDN > SDN Internetworks** then click on the tile in the middle panel to load the example as shown in below

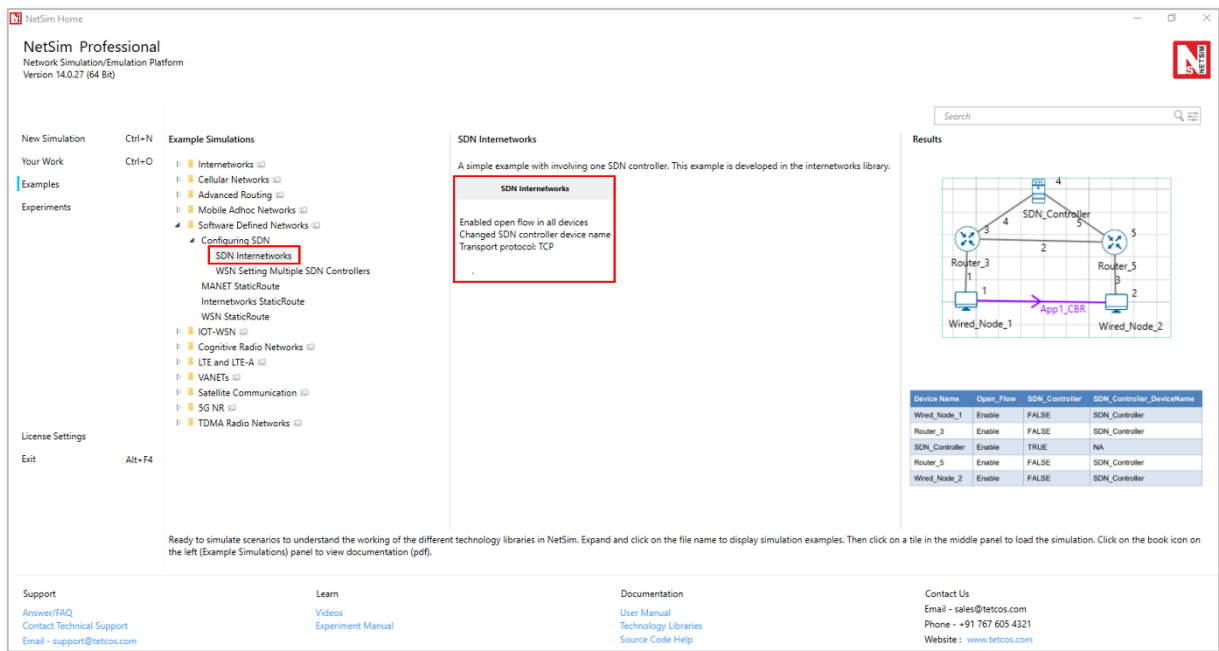


Figure 3-1: List of scenarios for the example of SDN Internetworks

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

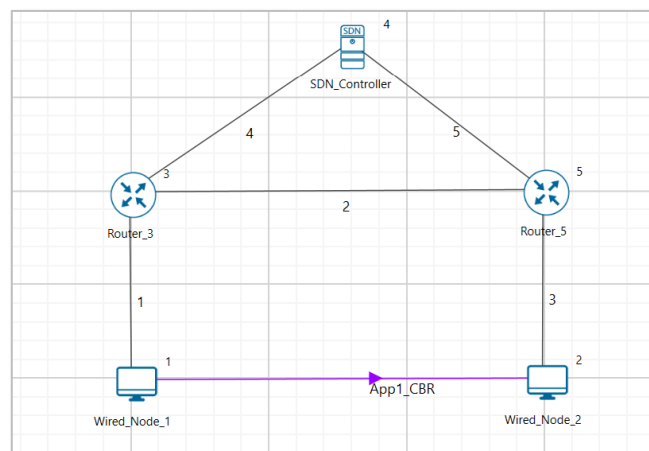


Figure 3-2: Network set up for studying the SDN Internetworks

1. 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:

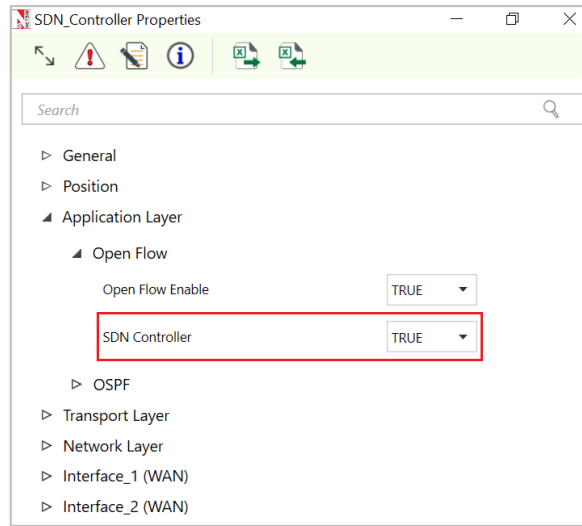


Figure 3-3: Application Layer properties Window of SDN_Controller device

2. 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:

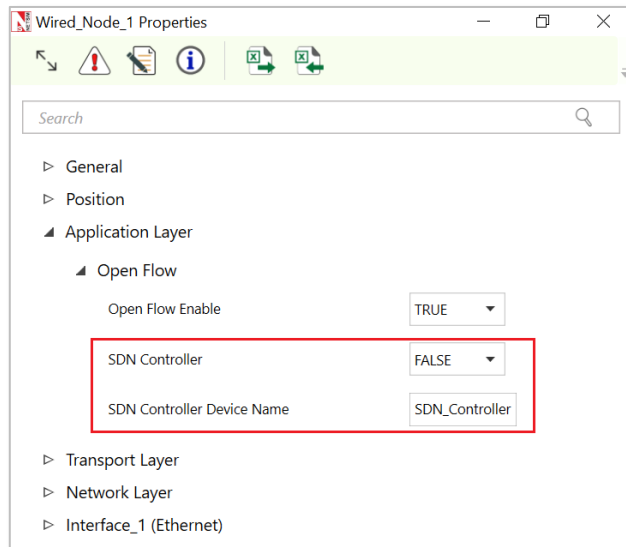


Figure 3-4: SDN_Controller_DeviceName is set as SDN_Controller in Application layer

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

3. Click on the Application icon present in the top ribbon/toolbar.
 - a. CBR Application from Wired Node 1 to Wired Node 2 with 50 Mbps Generation Rate (Packet Size: 1460Bytes, Inter Arrival Time: 233.6µs).
 - b. Set Transport Protocol to TCP.
4. Before simulating the example, Right-click SDN_Controller and select NetSim Console. Now, the client (NetSimCLI.exe) waits to establish a connection with NetSimCore.exe as shown below.

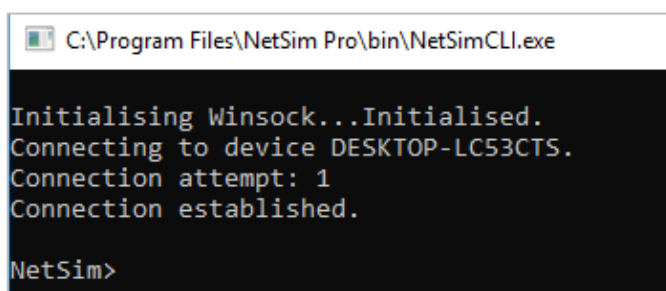


Figure 3-5: Connection Establishment

5. Follow the below steps to simulate an example.
 - a. Click on the Options tab. Select Run time interaction.

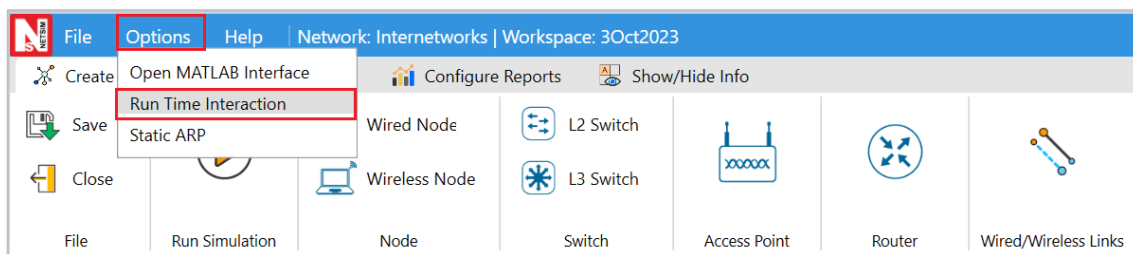


Figure 3-6: Run Time Interaction option in NetSim GUI

- b. In the Run Time Interaction tab, set Interactive Simulation to True and click on OK.

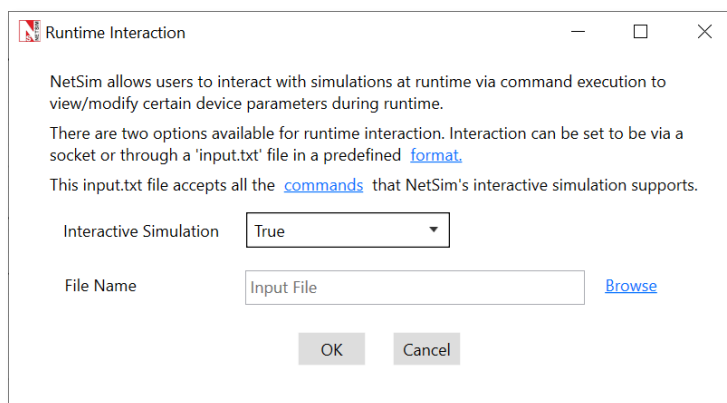


Figure 3-7: Enable Interactive Simulation

- c. Click on Run Simulation window and simulate it to 500 secs.

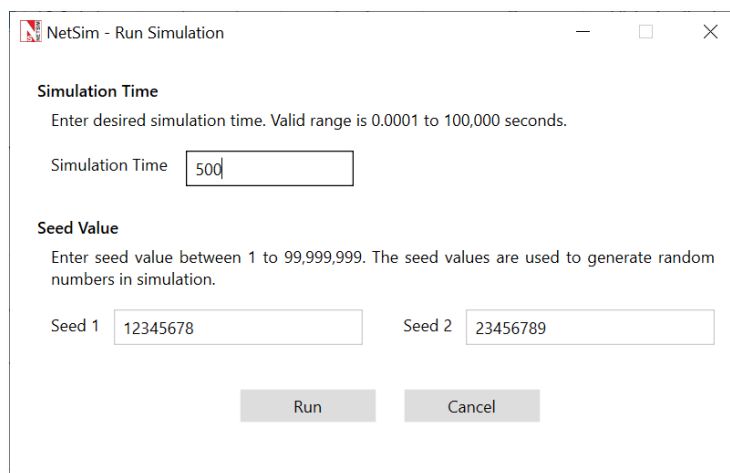


Figure 3-8: Run Simulation Window

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

```

C:\Users\User\OneDrive\OneDrive\Documents\NetSim\Workspaces\30ct2023\bin_x64\NetSimCore.exe
NetSim License Manager will first check for node lock licenses.
If not available, it will then check for floating/cloud licenses
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -
No license for product (-1)

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>std>14>0>rlm_hw>1111111111111111>10100>
NetworkStack loaded from path- C:\Users\User\OneDrive\OneDrive\Documents\NetSim\Workspaces\30ct2023\bin_x64\NetworkStack.dll

***
NetSim start
Network Stack loaded
Error in creating C:\Users\User\AppData\Local\Temp\NetSim\std_14.0\log directory. Error number 17
Initializing simulation
Config file reading complete
License re-validation complete
Protocol binaries loaded
Executing command --- DEL "C:\Users\User\AppData\Local\Temp\NetSim\std_14.0\*.pcap"
Could Not Find C:\Users\User\AppData\Local\Temp\NetSim\std_14.0\*.pcap
Emulation is disabled
Stack variables initialized
Could Not Find C:\Users\User\AppData\Local\Temp\NetSim\std_14.0\Plot_*
Metrics variables initialized
Protocol variables initialized
NetSim Console Mode is enabled.
Waiting for first client to connect. Press ctrl+c to stop connection.

```

Figure 3-9: Waiting for first client to connect.

- d. Use this console to execute SDN commands as explained in section **Error! Reference source not found.**

3.1.2 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 500 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:

Open NetSim and Select **Examples > Software Defined Networks > Configuring SDN > WSN Setting Multiple SDN Controllers** then click on the tile in the middle panel to load the example as shown in below screenshot

The screenshot shows the NetSim Professional interface. The left panel lists 'Example Simulations' with a tree view under 'Software Defined Networks' > 'Configuring SDN' > 'SDN Internetworks' > 'WSN Setting Multiple SDN Controllers'. The middle panel displays the simulation details for 'WSN Setting Multiple SDN Controllers', including the description: 'Simulating multiple SDN controllers. This example is developed for Wireless Sensor Networks'. Below the description, a table lists the configuration parameters for the simulation.

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

The right panel shows a network diagram with nodes: SDN_Controller_Sensor1, SDN_Controller_Sensor2, SDN_Controller_SinkNode, Wireless_Sensor_3, and Wireless_Sensor_5. A green arrow labeled 'App1_SENSOR_APP' points from Wireless_Sensor_3 to Wireless_Sensor_5. Below the diagram is a table with the same configuration parameters as shown in the middle panel.

Figure 3-10: List of scenarios for the example of WSN Setting Multiple SDN Controllers

The following network diagram illustrates the network setup related to this example as shown below Figure 3-11.

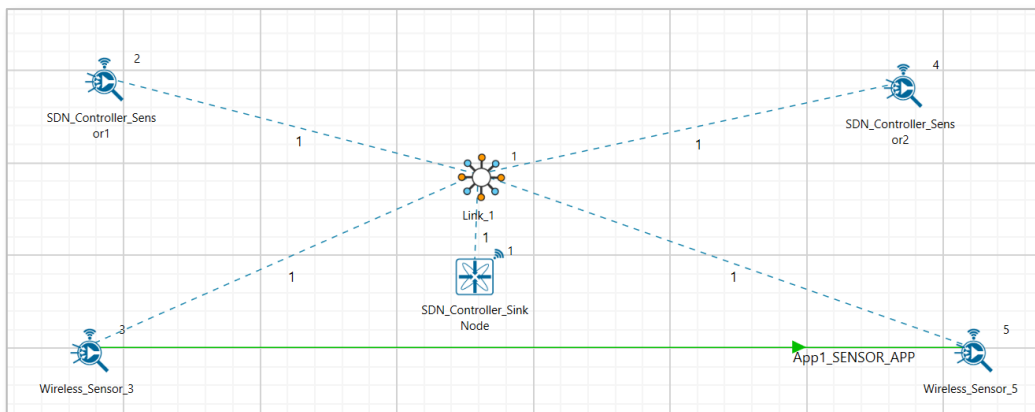


Figure 3-11: Network set up for studying the WSN Setting Multiple SDN Controllers

1. 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 3-12.

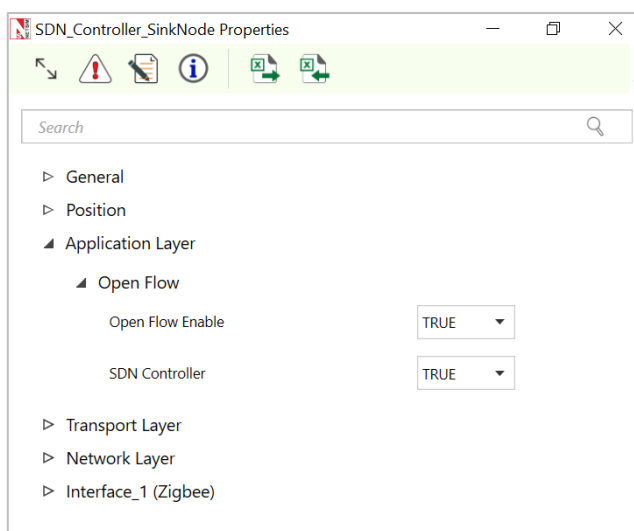


Figure 3-12: Application Layer properties Window

2. 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 3-13.

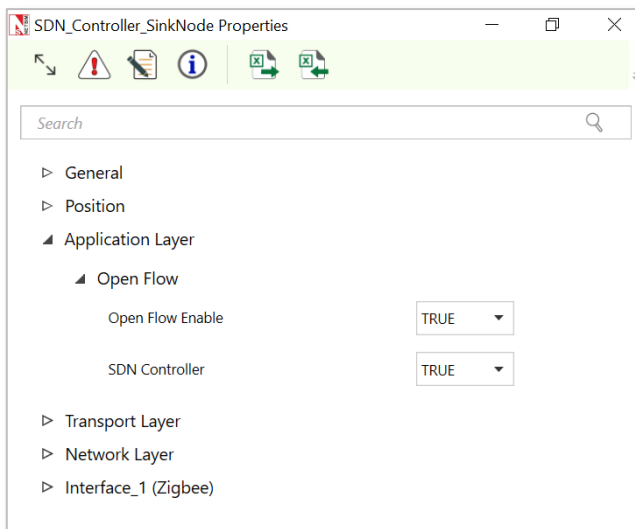


Figure 3-13: SDN_Controller_DeviceName is set as SDN_Controller in Application layer

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

3. Click on the Application icon present in the top ribbon/toolbar.
 - a. Sensor Application from Wireless_Sensor_3 to Wireless_Sensor_5 with Packet Size: 50Bytes, Inter Arrival Time: 1000000µs).
 - b. Set Transport Protocol to TCP.
4. Additional analysis options such as plots, packet trace and event trace are enabled.
5. Simulate the example. 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.
 - d. Click Accept.
 - e. Click OK.

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

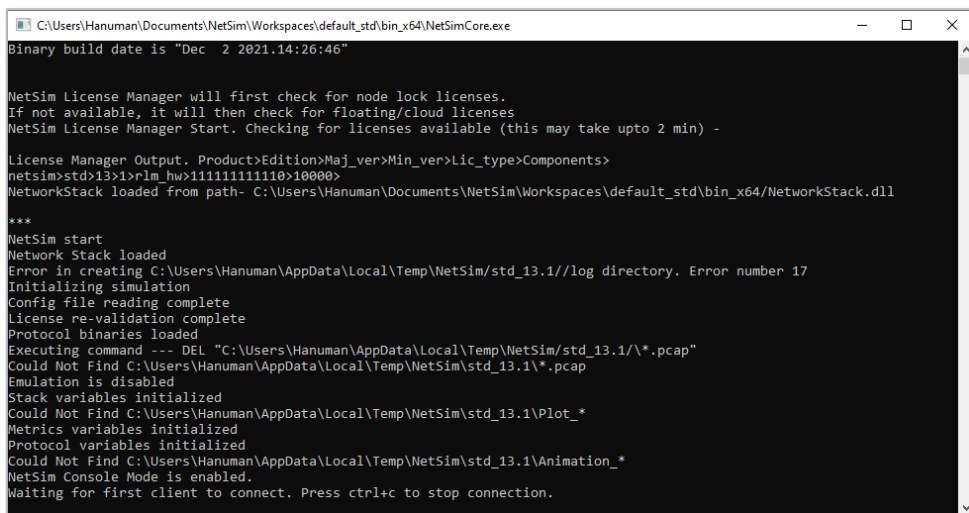


Figure 3-14: Waiting for first client to connect

6. 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.

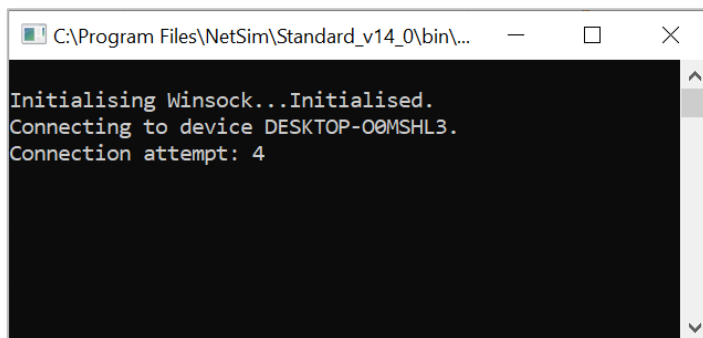


Figure 3-15: Connection established.

7. Simulate the example. To do so:

- a. Click on the Options tab. Select Run time interaction.

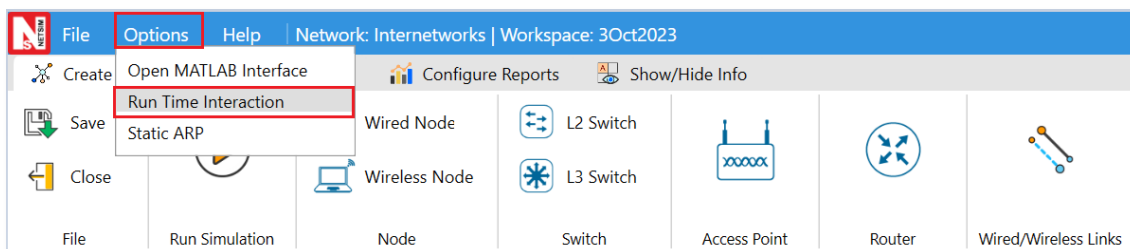


Figure 3-16: Run Time Interaction in NetSim GUI

- b. In the Run time Interaction tab, Interactive Simulation option is set to True and click on OK.

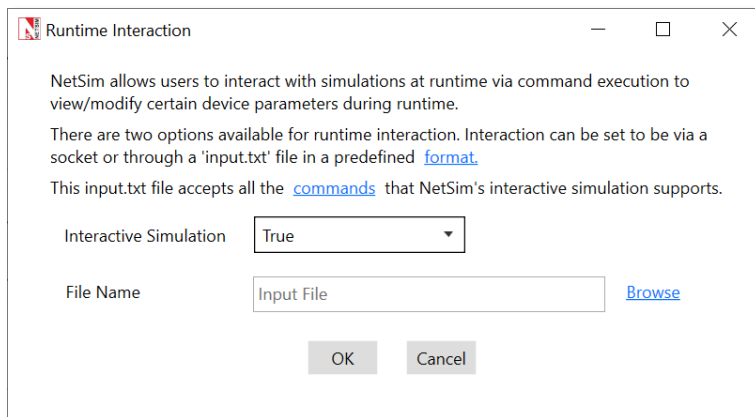


Figure 3-17: Runtime Interaction window

c. Click on the Run Simulation window and simulate it to 500 sec.

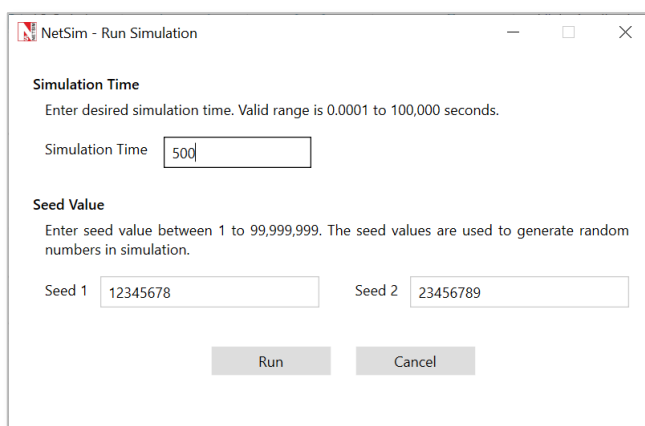


Figure 3-18: Run Simulation Window

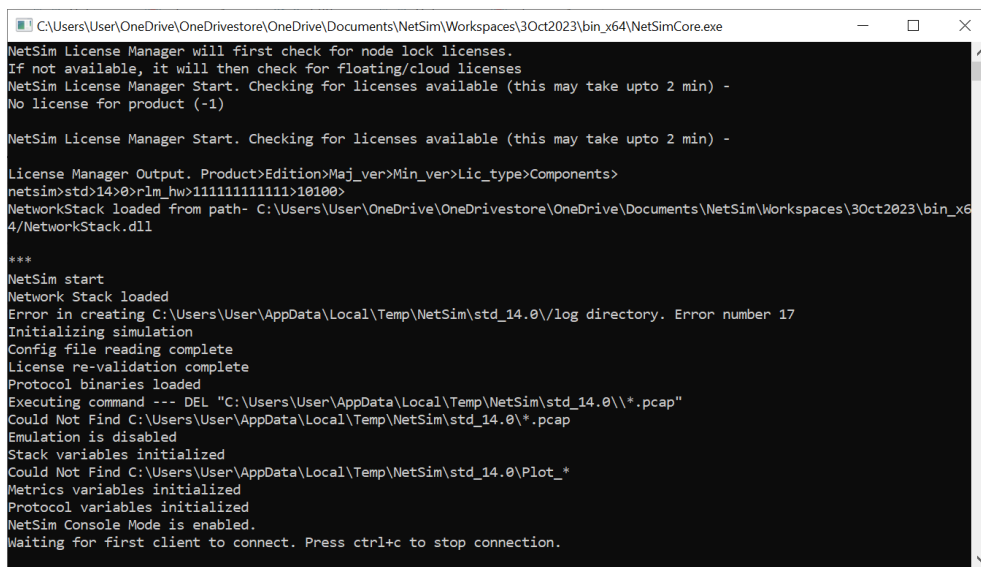


Figure 3-19: Waiting for first Client to Connect

You can execute various SDN commands that are supported in any of the SDN controller Sensor CLI.

3.2 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 5 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 1000 seconds.
- Packet trace, and Event trace is enabled.

To simulate the example for MANET using SDN:

Open NetSim and Select **Examples > Software Defined Networks > MANET StaticRoute** then click on the tile in the middle panel to load the example as shown in below screenshot

The screenshot displays the NetSim Professional interface. On the left, the 'Example Simulations' tree is expanded to 'Configuring SDN' > 'SDN Internetworks' > 'WSN Setting Multiple SDN Controllers' > 'MANET StaticRoute'. The middle panel shows the scenario details for 'MANET StaticRoute', including a description: 'In this example, static routes are configured using the SDN controllers, in a MANET network.' and configuration parameters: 'Enabled open flow in all devices', 'Changed SDN controller device name', 'Network layer protocol: DSR', and 'Interactive simulation: TRUE'. The right panel shows a network diagram with 5 wireless nodes and 1 SDN controller, and a list of CLI commands for configuring static routes on each device.

MANET StaticRoute

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

MANET StaticRoute

Enabled open flow in all devices
 Changed SDN controller device name
 Network layer protocol: DSR
 Interactive simulation: TRUE

Results

Type the following commands on the NetSim CLI console, in the <<Device name 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
```

Figure 3-20: List of scenarios for the example of MANET StaticRoute

The following network diagram illustrates the network setup as shown Figure 3-21.

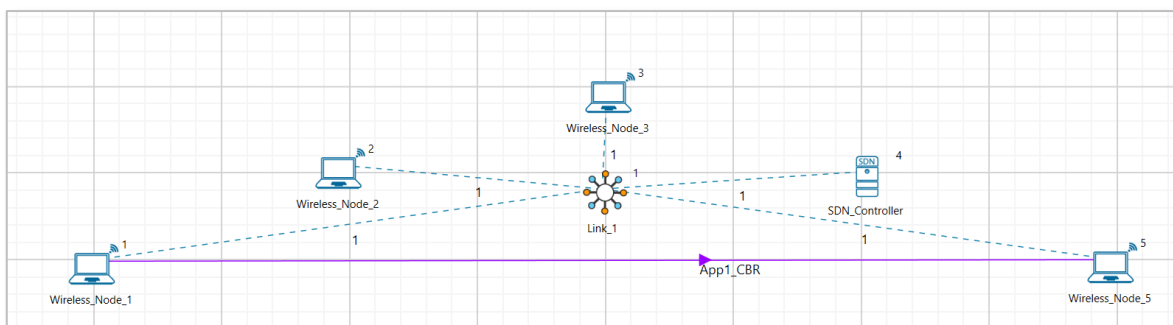


Figure 3-21: Network set up for studying the MANET StaticRoute

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

1. Click on the Application icon present in the top ribbon/toolbar.
 - a. CBR Application from Wireless_Node_1 to Wireless_Node_5 with 0.584 Mbps Generation Rate (Packet Size: 1460Bytes, Inter Arrival Time: 20000 μ s).
 - b. Set Transport Protocol to UDP.
2. Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, Path Loss Exponent:3.2.
3. In NetSim GUI Plots, Packet Trace and Event Trace are enabled.
4. Simulate the example. To do so:
 - a. Click the Run button. The Run Simulation pop-up window appears.
 - b. Simulation time is set to 1000 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 select NetSim Console. Now, the client (NetSimCLI.exe) starts and attempts to establish a connection with NetSimCore.exe.
6. Simulate the example. To do so:
 - a. Click on the Options tab. Select Run time interaction.

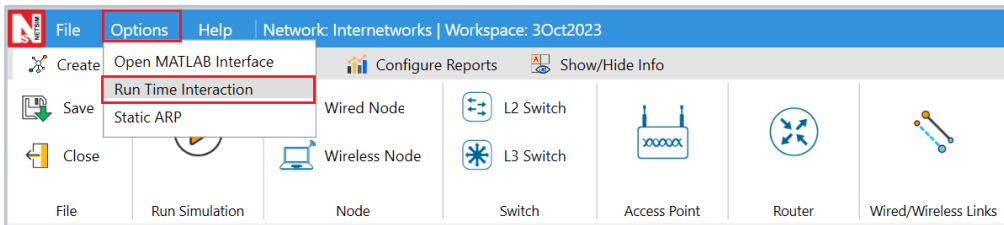


Figure 3-22: Runtime Interaction in GUI

- b. In the Run time Interaction tab, Interactive Simulation option is set to True and click on OK.

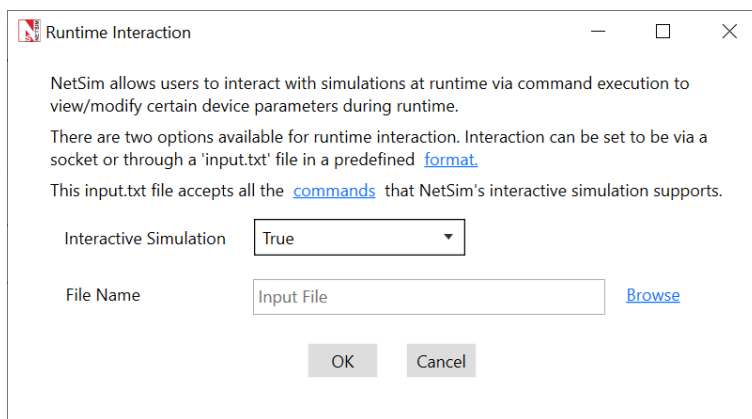


Figure 3-23: Enabling Run Time Interaction option.

- c. Click on Run simulation window and simulate it to 500 sec.

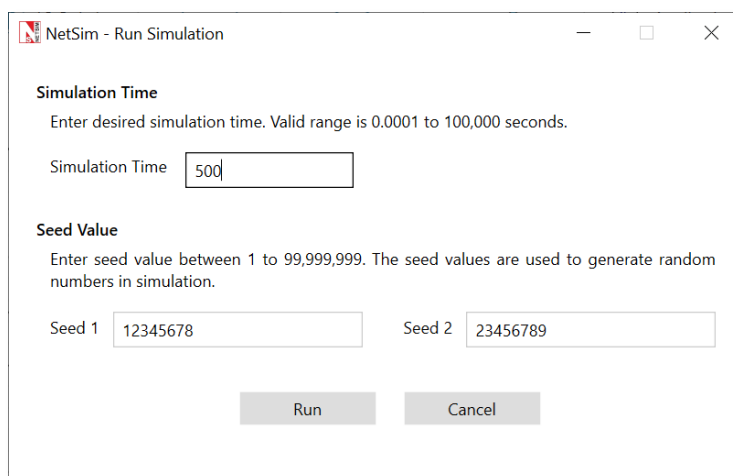


Figure 3-24: Run Simulation window

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

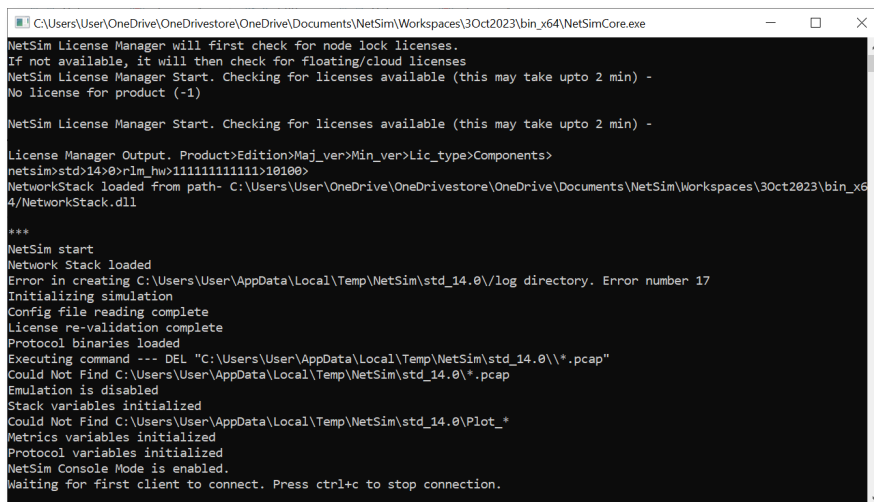


Figure 3-25: Waiting for the first Client to connect.

7. Interpret the Results

- a. 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 for APP1_CBR.
- b. The packets reach Wireless_Node_5 from Wireless_Node_1-> Wireless_Node_3-> Wireless_Node_5.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-1	NODE-3
1	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-5

Figure 3-26 : Packet trace screenshot

- 8. For the same example, 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. Right-click SDN_Controller and select NetSim Console.
 - b. Re- run the simulation.
 - 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.

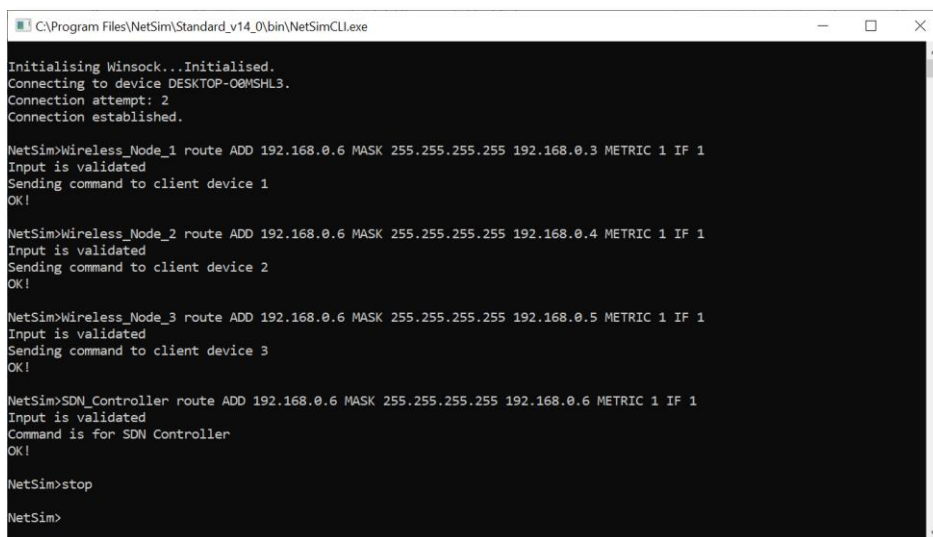


Figure 3-27: NetSim Console Window

9. Interpret the Results.

- a. Open Packet trace and filter CONTROL_PACKET_TYPE/APP_NAME to APP1_CBR, you will see that data packets transmitted 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.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1558	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-5
1559	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-1	NODE-2
1559	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-2	NODE-3
1559	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-5
1560	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-1	NODE-2
1560	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-2	NODE-3
1560	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-4
1560	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-4	NODE-5
1561	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-1	NODE-2
1561	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-2	NODE-3
1561	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-4
1561	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-4	NODE-5
1562	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-1	NODE-2
1562	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-2	NODE-3
1562	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-4
1562	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-4	NODE-5
1563	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-1	NODE-2
1563	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-2	NODE-3
1563	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-4

Figure 3-28: Packet Trace Window

3.3 Configuring Static Routes for Internetworks using SDN

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:

Open NetSim and Select **Examples > Software Defined Networks > Internetworks StaticRoute** then click on the tile in the middle panel to load the example as shown in below screenshot

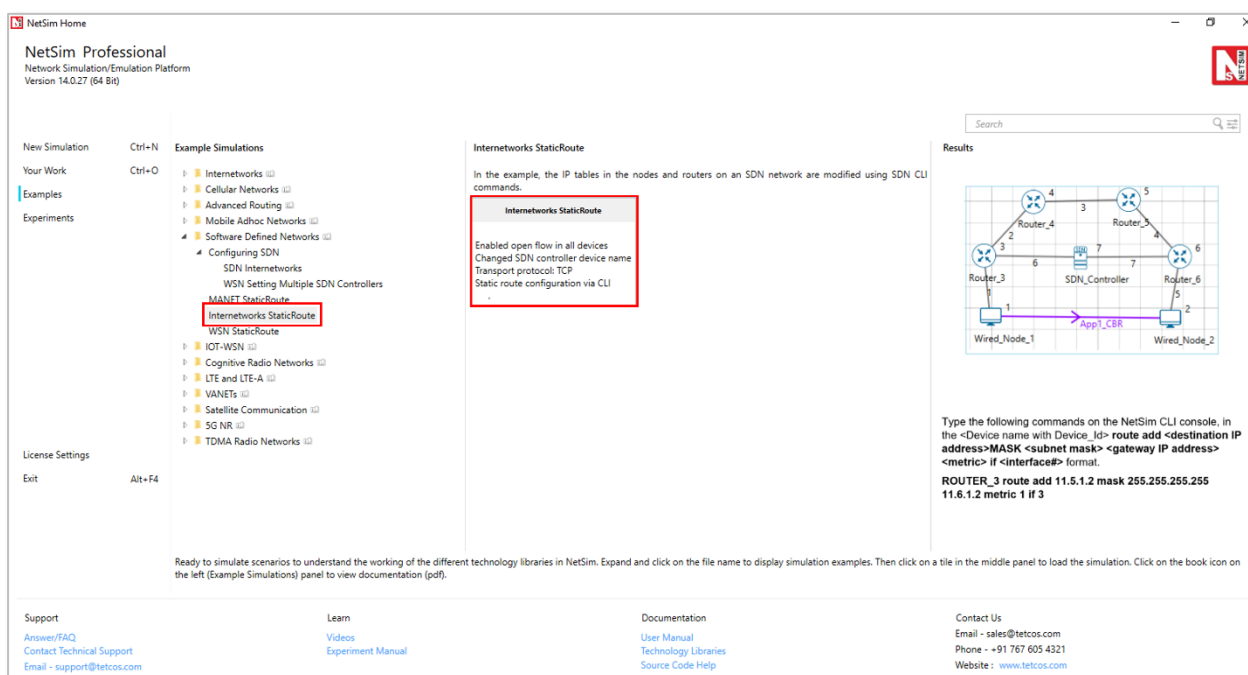


Figure 3-29: List of scenarios for the example of Internetworks StaticRoute

The following network diagram illustrates the network setup related to this example as shown below Figure 3-30.

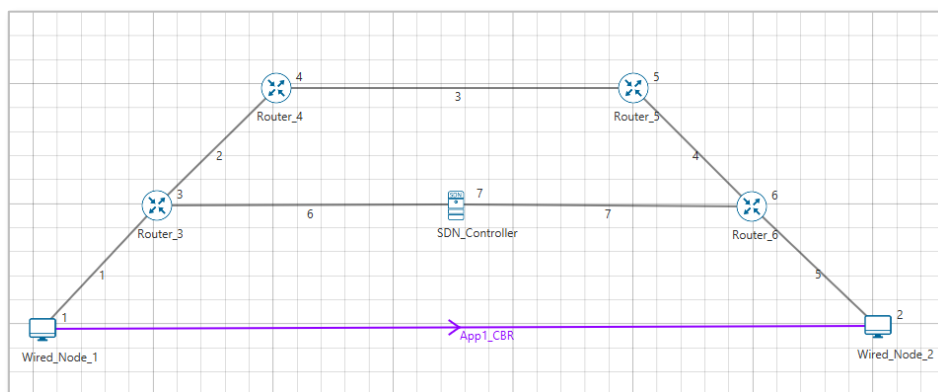


Figure 3-30: Network set up for studying the Internetworks StaticRoute

1. Click on the Application icon present in the top ribbon/toolbar.
 - a. CBR Application from Wired Node 1 to Wired Node 2 with 50 Mbps Generation Rate (Packet Size: 1460Bytes, Inter Arrival Time: 233.6μs).
 - b. Application Start time is 30 sec.
 - c. Set Transport Protocol to TCP.
2. Plots, Packet Trace and Event Trace features are enabled.
3. 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.
4. Right-click **SDN_Controller** and click NetSim Console.
5. 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.
6. Interpret the results.
 - a. Open packet trace and filter the CONTROL_PACKET_TYPE/APP_NAME to APP1_CBR, and PACKET_ID to 1. Packets reach Wired_Node_2 via Wired_Node_1 > Router 3 > SDN_Controller > Router 6 as shown in Figure 3 15.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-7
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-7	ROUTER-6
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-6	NODE-2

Figure 3-31: Packet Animation window

7. 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	Packets Transmitted		Packets Errored		Packets Collided	
		Data	Control	Data	Control	Data	Control
All	NA	184414	184006	242	10	0	0
1	NA	46196	45950	61	2	0	0
2	NA	0	39	0	0	0	0
3	NA	0	39	0	0	0	0
4	NA	0	40	0	0	0	0
5	NA	46010	45957	56	4	0	0
6	NA	46135	45988	62	2	0	0
7	NA	46073	45993	63	2	0	0

Figure 3-32: Link Metrics table in Result Window

8. For the same example, 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:

- Simulate the network again.
- 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.
- 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 192.168.0.2 MASK 255.255.255.0 11.0.0.2 METRIC 1 IF 2**
- (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 3-33**.

```

C:\Program Files\NetSim\Standard_v14_0\bin\NetSimCLI.exe
NetSim>Router_3 route print
Input is validated
Sending command to client device 3
=====
IP Route Table
=====
Network Destination Netmask//Prefix Gateway Interface Metric Type
-----
192.168.0.2 255.255.255.0 11.0.0.2 11.0.0.3 1 STATIC
11.0.0.2 255.255.255.255 11.0.0.2 11.0.0.3 100 OSPF
11.0.0.11 255.255.255.255 11.0.0.2 11.0.0.3 100 OSPF
11.0.0.26 255.255.255.255 11.0.0.26 11.0.0.27 100 OSPF
11.0.0.35 255.255.255.255 11.0.0.26 11.0.0.27 100 OSPF
11.0.0.10 255.255.255.255 11.0.0.2 11.0.0.3 200 OSPF
11.0.0.19 255.255.255.255 11.0.0.2 11.0.0.3 200 OSPF
11.0.0.18 255.255.255.255 11.0.0.26 11.0.0.27 200 OSPF
11.0.0.34 255.255.255.255 11.0.0.26 11.0.0.27 200 OSPF
11.0.0.24 255.255.255.248 on-link 11.0.0.27 300 LOCAL
11.0.0.0 255.255.255.248 on-link 11.0.0.3 300 LOCAL
192.168.0.0 255.255.0.0 on-link 192.168.0.1 300 LOCAL
224.0.0.1 255.255.255.255 on-link 192.168.0.1 306 MULTICAST
224.0.0.0 240.0.0.0 on-link 192.168.0.1 306 MULTICAST
255.255.255.255 255.255.255.255 on-link 192.168.0.1 999 BROADCAST
=====

```

Figure 3-33: Route added to network and route printed in command.

9. 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 (7c) as shown in below.

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 3-34: 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.
10. In Router_3 under application layer use routing protocol as RIP
 11. 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

12. 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 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 5 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 500 seconds.
- Packet trace is enabled.

To simulate the example for WSN using SDN:

Open NetSim and Select **Examples > Software Defined Networks > WSN StaticRoute** then click on the tile in the middle panel to load the example as shown in below screenshot

The screenshot displays the NetSim Professional interface. On the left, the 'Example Simulations' list is expanded to 'Software Defined Networks > Configuring SDN > Internetworks StaticRoute > WSN StaticRoute'. The central panel shows the example title 'WSN StaticRoute' and a description: 'In this example, static routes are configured in sensors using SDN commands executed in an SDN controller.' Below this, it lists configuration details: 'Enabled open flow in all devices', 'Changed SDN controller device name', 'Transport protocol: UDP', and 'Static route configuration via CLI'. The right panel features a network diagram with nodes labeled 'Wireless_Sensor_1' through 'Wireless_Sensor_5', 'SDN_Controller', 'AdHoc Link 1', and 'WSN_Sink_6'. Below the diagram, it provides CLI commands for configuring static routes on the sensors and the controller.

```

Type the following commands on the NetSim CLI console, in the <Device name 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
  
```

Figure 3-35: List of scenarios for the example of WSN StaticRoute

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

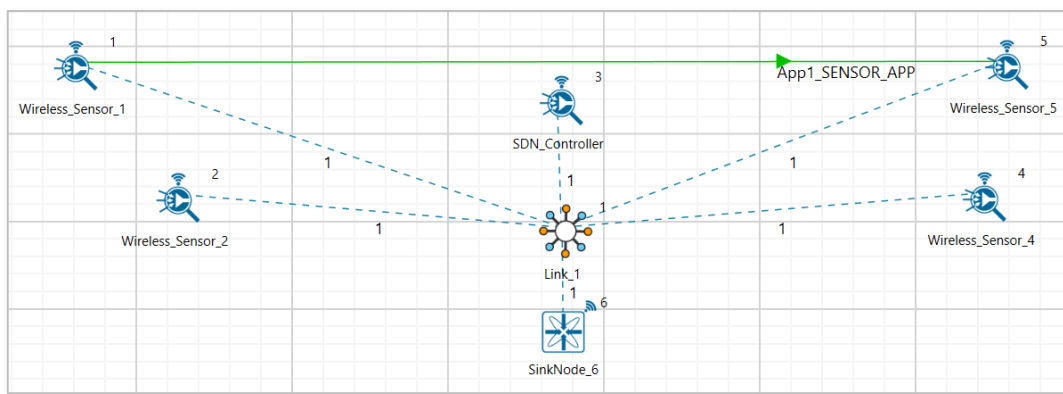


Figure 3-36: Network set up for studying the WSN StaticRoute

1. 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

2. Click on the Application icon present in the top ribbon/toolbar.
 - a. Sensor Application from Wireless_Sensor_1 to Wireless_Sensor_5 with default Generation Rate (Packet Size: 50Bytes, Inter Arrival Time: 1000000µs).
 - b. Set Transport Protocol to UDP.
3. Before Simulating the scenario, Right-click SDN_Controller and click NetSim Console. Now, the client (NetSimCLI.exe) waits to establish a connection with NetSimCore.exe.
 - a. Enable the packet trace and click on the Options tab. Select Run time interaction.

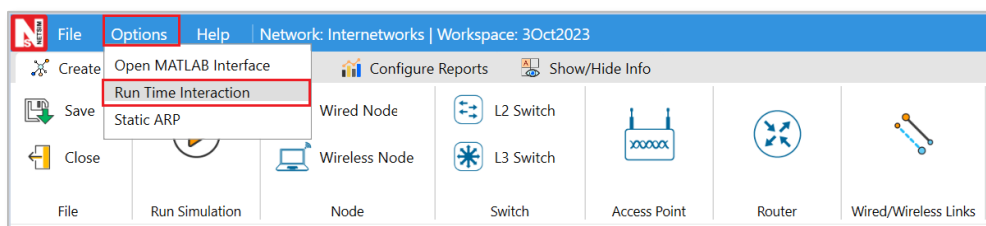


Figure 3-37: Runtime Interaction in GUI

- b. In the Run time Interaction tab, Set the Interactive Simulation option as True and click on OK.

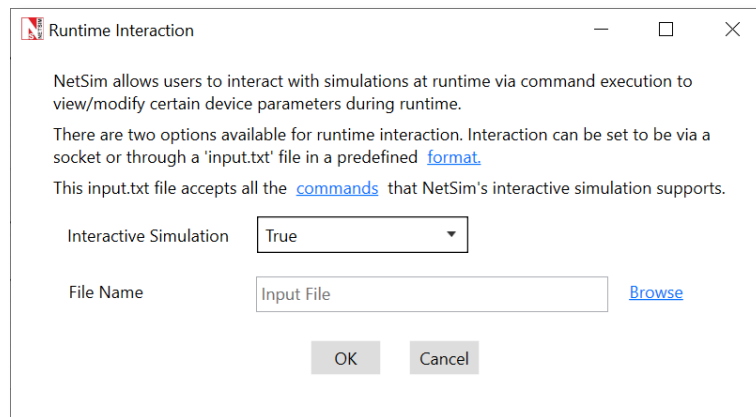


Figure 3-38: Runtime Interaction Window

- c. Click on Run simulation window and simulate it to 500 sec.

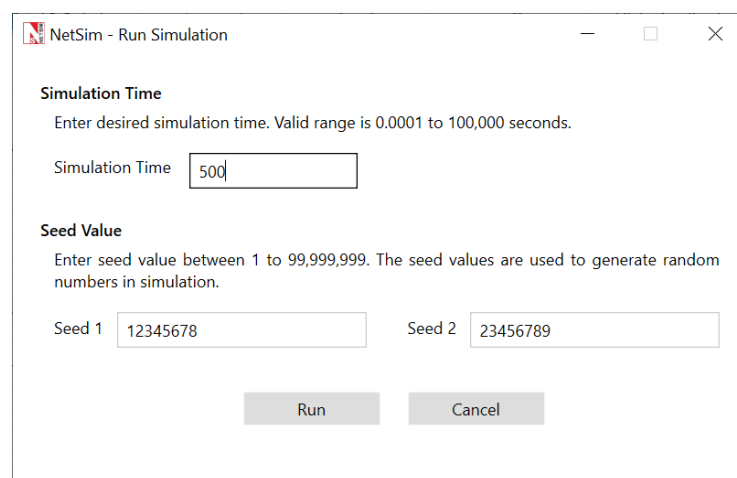


Figure 3-39: Run Simulation Window

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

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

```

C:\Users\User\OneDrive\OneDrive\Documents\NetSim\Workspaces\3Oct2023\bin_x64\NetSimCore.exe
NetSim License Manager will first check for node lock licenses.
If not available, it will then check for floating/cloud licenses
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -
License server does not support this product (-18)

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>std>14>0>rlm_hw>1111111111111111>00100>
NetworkStack loaded from path- C:\Users\User\OneDrive\OneDrive\Documents\NetSim\Workspaces\3Oct2023\bin_x64\NetworkStack.dll

***
NetSim start
Network Stack loaded
Error in creating c:\Users\User\AppData\Local\Temp\NetSim\std_14.0\log directory. Error number 17
Initializing simulation
Config file reading complete
License re-validation complete
Protocol binaries loaded
Executing command --- DEL "C:\Users\User\AppData\Local\Temp\NetSim\std_14.0\*.pcap"
Could Not Find C:\Users\User\AppData\Local\Temp\NetSim\std_14.0\*.pcap
Emulation is disabled
Stack variables initialized
Could Not Find C:\Users\User\AppData\Local\Temp\NetSim\std_14.0\Plot_*.
Metrics variables initialized
Protocol variables initialized
NetSim Console Mode is enabled.
Waiting for first client to connect. Press ctrl+c to stop connection.
    
```

Figure 3-40: Waiting for Client to connect.

- Observe the packet trace after simulation, you will see the direct communication between wireless sensor 1 and 5.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
2	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
3	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
4	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
5	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
6	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
7	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
8	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5

Figure 3-41: Direct communication between node 1 and node 5 in packet Animation window

- For the same example, 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
 - e. Follow the above step 3 as explained above
- 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 192.168.0.6 MASK 255.255.255.0 192.168.0.4 METRIC 1 IF 1
SDN_Controller route ADD 192.168.0.6 MASK 255.255.255.0 192.168.0.6 METRIC 1 IF
    
```

- Interpret the results.

Open Packet trace and filter the **CONTROL_PACKET_TYPE/APP_NAME** to show only **APP1_SENSOR_APP**, you will see that data packet flow from wireless sensor 1 to wireless sensor 5 through SDN Controller.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
9	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
10	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
11	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-5
12	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
12	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5
13	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
13	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5
14	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
14	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5
15	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
15	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5
16	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
16	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5
17	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
17	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5
18	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
18	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5
19	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-1	SENSOR-3
19	0	Sensing	App1_SENSOR_APP	SENSOR-1	SENSOR-5	SENSOR-3	SENSOR-5

Figure 3-42: Communication between node 1 > SDN Controller > node 5 in packet Animation window

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>.