NetSim®

Accelerate Network R & D

Vehicular Adhoc Networks (VANETs)

A Network Simulation & Emulation Software

By

TETCOS LLP

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 Introduction

***NOTE*: NetSim VANET library is available only in standard and pro versions**

Connected vehicle (CV) technologies enable a wide range of transportation applications in safety, mobility, and infotainment. While holding tremendous promise, the success of these CV-enabled applications will rely on the quality of the underlying information flow [1]. NetSim is a simulation tool to model, simulate and analyses this information flow. The vehicular communication architecture in NetSim is based on a combination of the IEEE 1609 standard and IEEE 802.11p standard. The 802.11p standard defines the PHY and MAC layers while IEEE 1609 defines the upper layers.

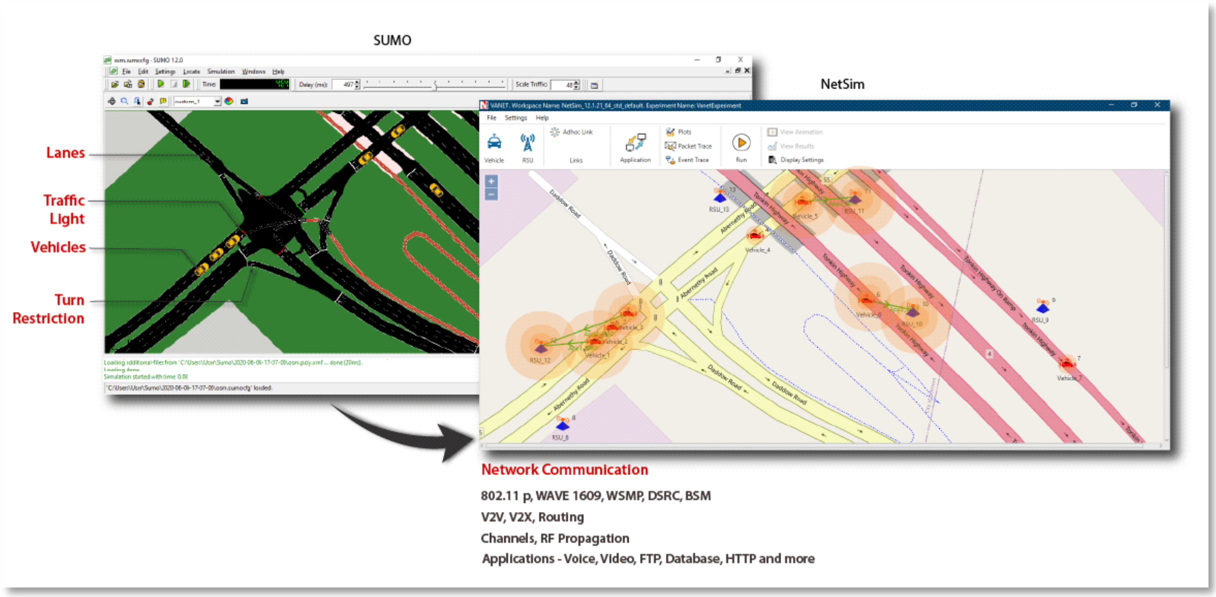

Figure 1-1: NetSim-SUMO interfacing for VANET simulation. Top left is a SUMO screen shot while bottom right is a NetSim screen shot.

NetSim's VANET library features:

- IEEE 802.11p PHY operating in the 5.9 GHz band with a channel bandwidth of 10 MHz. 802.11p is an adaptation of the famous IEEE 802.11a standard previously used in Wi-Fi systems.
- Radio propagation in the PHY layer covering various pathloss, shadowing and fading models.
- IEEE 802.11p MAC layer. Stations communicate directly outside the context of a BSS.
- IEEE 1609-2, which defines security services for application messages and management messages in WAVE.
- IEEE 1609-3, which defines connection set up and management of WAVE compliant devices.

- IEEE 1609-4, which enables upper layer operational aspects across multiple channels without knowledge of PHY layer parameters.
- DSRC SAE J2735
- BSM packets that are transmitted using WSMP
- A spontaneous Adhoc network formation between the VANET nodes; layer-3 IP routing can be through DSR, AODV, OLSR or ZRP for non-BSM packets
- Vehicular mobility using in-built mobility models or by interfacing with SUMO software
- Interfacing between SUMO & NetSim via Traffic control interface (TraCI). Automatic import of road network and vehicle mobility from SUMO
- Wide range of output metrics including Delay, Throughput, Error, Retransmission, etc.
- Protocol source C code is provided along with NetSim software

In VANETs, Vehicles and roadside units (RSUs) are the communicating nodes, providing each other with (i) safety information using BSM application and (ii) infotainment applications. Both types of nodes are dedicated short-range communications (DSRC) devices. The RSU is a WAVE device usually fixed along the roadside or in dedicated locations such as at junctions or near parking spaces. In NetSim, users can model network traffic flows:

- between two or more Vehicles, known as V2V
- from vehicles to RSUs (infrastructure), known as V2I
- between two or more RSUs
- from vehicles or RSUs to remove servers, by connecting RSUs in backhaul to a wired network comprising of switches, routers, and servers for end-to-end simulation.

2 Simulation GUI

2.1 Create Scenario

- Open NetSim and click **New Simulation → Vehicular Adhoc Network (Vanet)** as shown Figure 2-1.

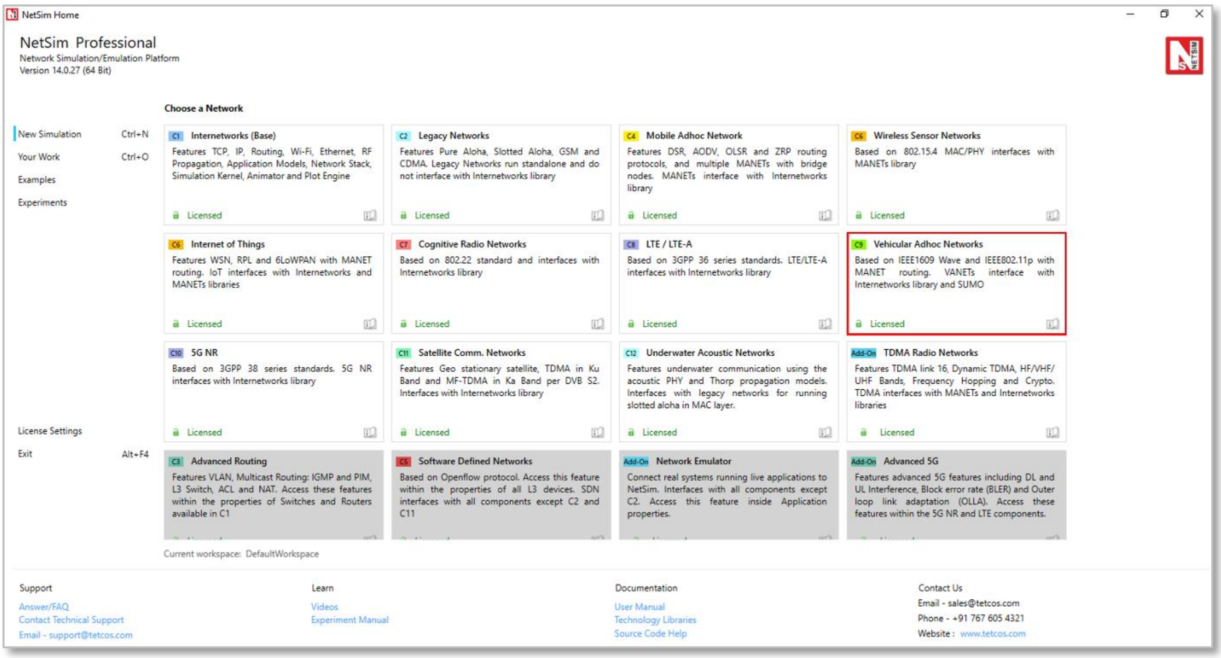

Figure 2-1: NetSim Home Screen

- A dialogue box appears as shown below, in that browse the Sumo Configuration File path. The general format of such file is **"*.Sumo.cfg".**

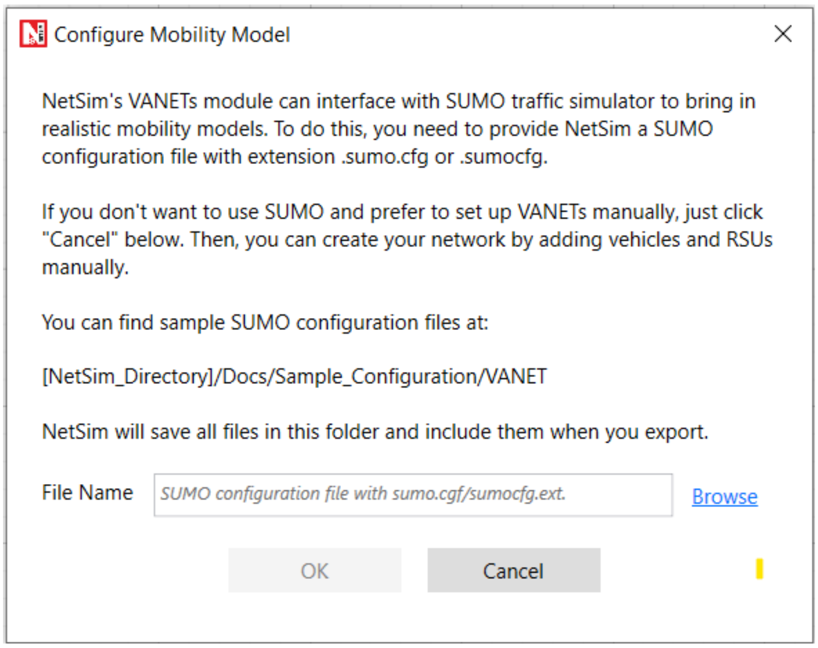

Figure 2-2: Sumo Configuration File path

- NetSim VANET module is designed to interface directly with SUMO.
- A SUMO configuration file is required as an input to NetSim.
- Sample SUMO configuration files are available inside <NetSim-Installation-Directory>**\Docs\Sample_Configuration\VANET** folder.
- Users can either use a Sumo configuration file which is provided inside NetSim's installation directory or use a different user specified SUMO configuration file. This .cfg file contains the path of NETWORK file and VEHICLES file.
- Further help on how to create SUMO configuration files is available at **http://sumo.dlr.de/wiki/Networks/Building_Networks_from_own_XML-descriptions**.

After selecting the Sumo configuration file name, the scenario is opened, with nodes placed at their respective starting positions (tracked from Sumo). Roads and Traffic Lights are also placed exactly as present in SUMO Configuration file.

2.2 Devices specific to NetSim VANETs Library

- **Vehicle (with one OBU):** In NetSim, a vehicle is a mobile communications device. It is assumed to have one (1) on board unit (OBU) which is a 5-layer device. The OBU can communicate with other OBUs or with RSUs via an Adhoc link. The OBU is assumed to have one wireless interface and has its own IP and MAC addresses.
- **Roadside Unit (RSU):** In NetSim, an RSU is a fixed communicating device. RSUs are generally termed as infrastructure. Vehicle (OBU) to RSU is termed as V2I communication. The RSU is a 5-layer device that can be connected to a Vehicle or to a Router. RSUs cannot be directly connected to other RSUs. RSUs have one (1) wireless interface and one (1) serial interface, and each interface has its own IP and MAC addresses.
- **Wired node:** A Wired node can be an end-node or for a server. It is a 5-layer device that can be connected to a switch and router. It supports only 1 Ethernet interface and has its own IP and MAC Addresses.
- **Wireless Nodes:** A Wireless node can be an end-node or a server. It is a 5-layer wireless device that can be connected to an Access point. It supports only 1 Wireless interface and has its own IP and MAC Addresses.
- **L2 Switch:** A Switch is a layer-2 device that uses the devices' MAC address to make forwarding decisions. It does not have an IP address.
- **Router:** Router is a layer-3 device and supports a maximum of 24 interfaces each of which has its own IP address.

- **Access point:** Access point (AP) is a layer-2 wireless device working per 802.11 Wi-Fi protocol. It can be connected to wireless nodes via wireless links and to a router or a switch via a wired link.

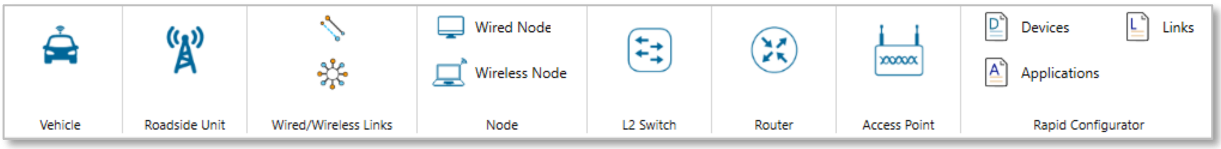

Figure 2-3: VANET Library Device Palette in GUI

2.3 Set Node, Link and Application Properties

- Right click on the appropriate node or link and select Properties. Then modify the parameters according to the requirements.
- Routing Protocol in Network Layer and all user editable properties in Data Link Layer few properties are Global or Local, Physical Layer and Power are Local.
- In Physical layer standards are acting as Link global.
- In the General properties, Mobility Model is set to SUMO, and it is Editable. This signifies that the Node movements will be traced from SUMO.
- File name gives the path to Sumo Configuration file that was given by the user.
- Step Size is taken from the Sumo Configuration file specified which tells the amount of time paused in sumo corresponding to single step of SUMO Simulation.
- In Interface (wireless) properties, under Physical layer, by default Standard is set to IEEE 802.11p in case of VANET.
- The following are the important properties of VANET Wireless Node (RSU/Vehicle) in Data link and Physical layers.

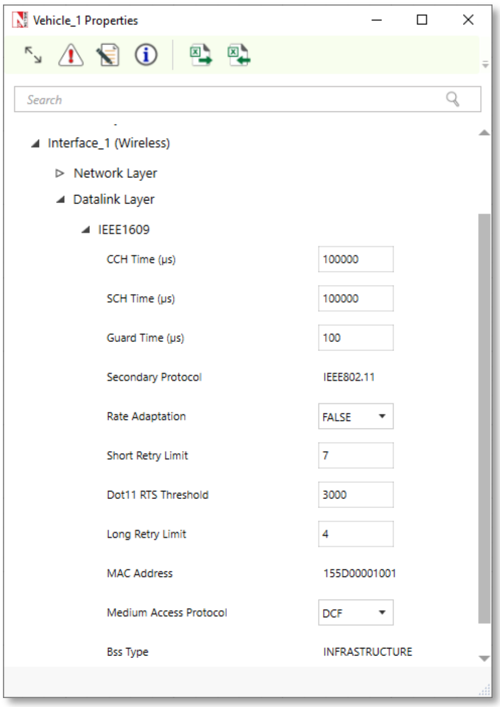

Figure 2-4: Vanet > Datalink layer Properties Window

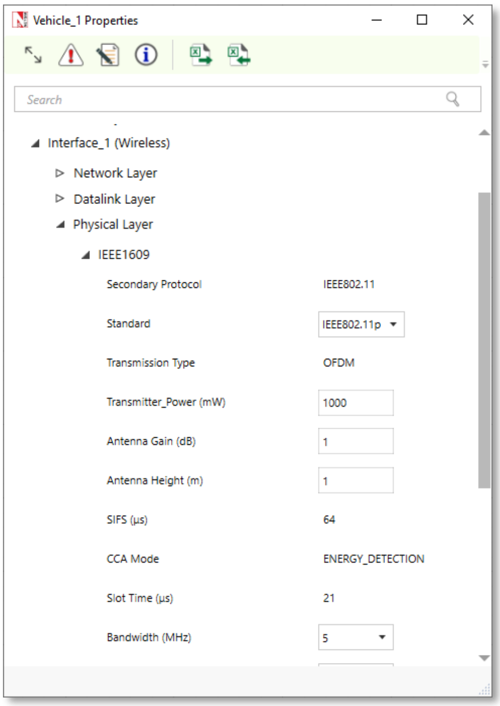

Figure 2-5: Vanet > Physical layer Properties Window

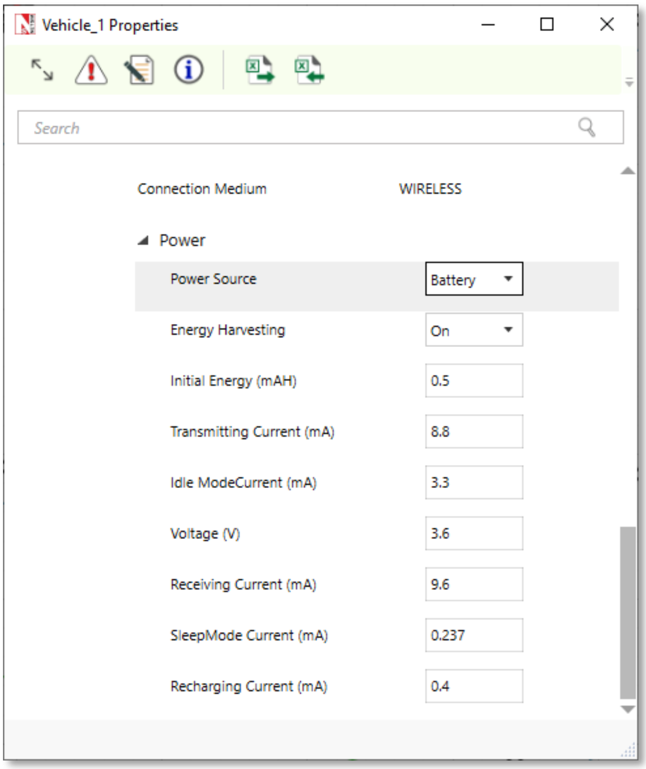

Figure 2-6: Vanet > Physical layer Properties Window > Battery Model

- Click on the Application icon present in the Set Traffic option and set properties. Multiple applications can be generated by using add button in Application properties.

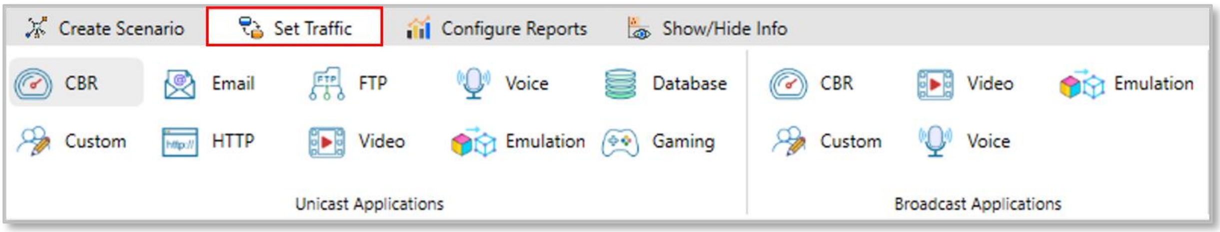

Figure 2-7: Application icon present on top ribbon

- Set the values according to requirement and click OK.

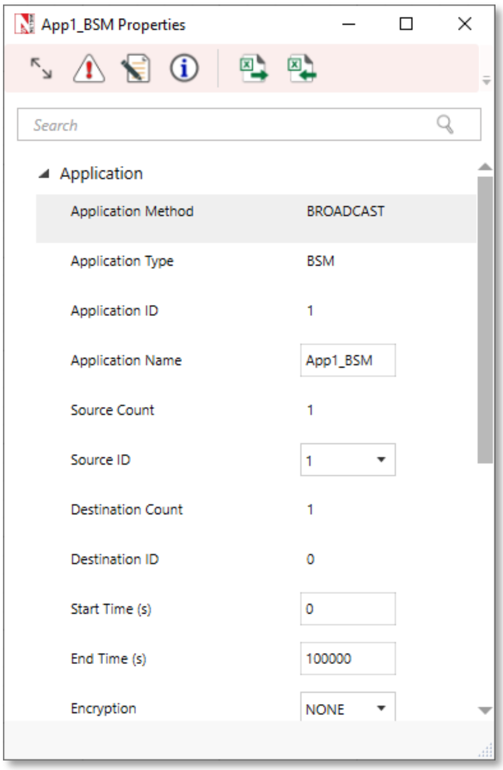

Figure 2-8: Application Configuration window

Detailed information on Application properties is available in **section 6** of NetSim User Manual.

2.4 Enable Packet Trace, Event Trace & Plots (Optional)

Click Packet Trace / Event Trace icon in the Configure Reports option and click OK. To get detailed help, please refer to **sections 8.4 and 8.5** in User Manual. Select Plots icon for enabling Plots and click OK.

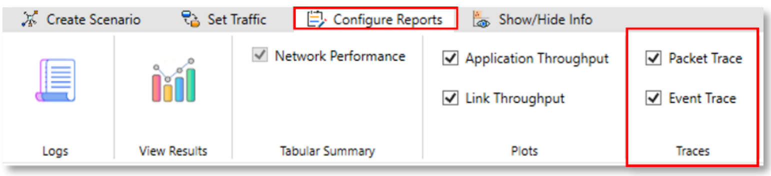

Figure 2-9: Enable Packet Trace, Event Trace & Plots options on top ribbon

2.5 Run Simulation

Click on **Run Simulation** icon on the top toolbar. Simulation Time is set from the Configuration File of Sumo. The simulation has three options.

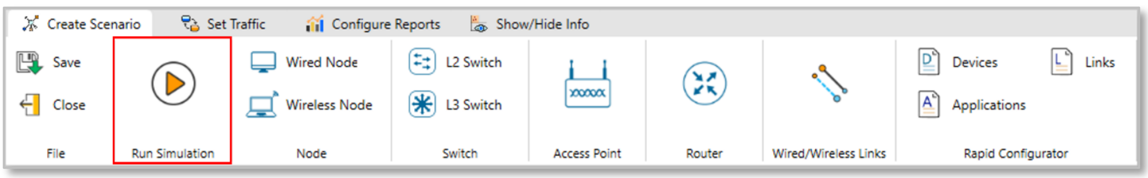

Figure 2-10: Run Simulation option on top ribbon

SUMO determines the positions of vehicles with respect to time as per the road conditions. NetSim reads the coordinates of vehicles from SUMO (through pipe) during runtime and uses it as input for vehicles mobility.

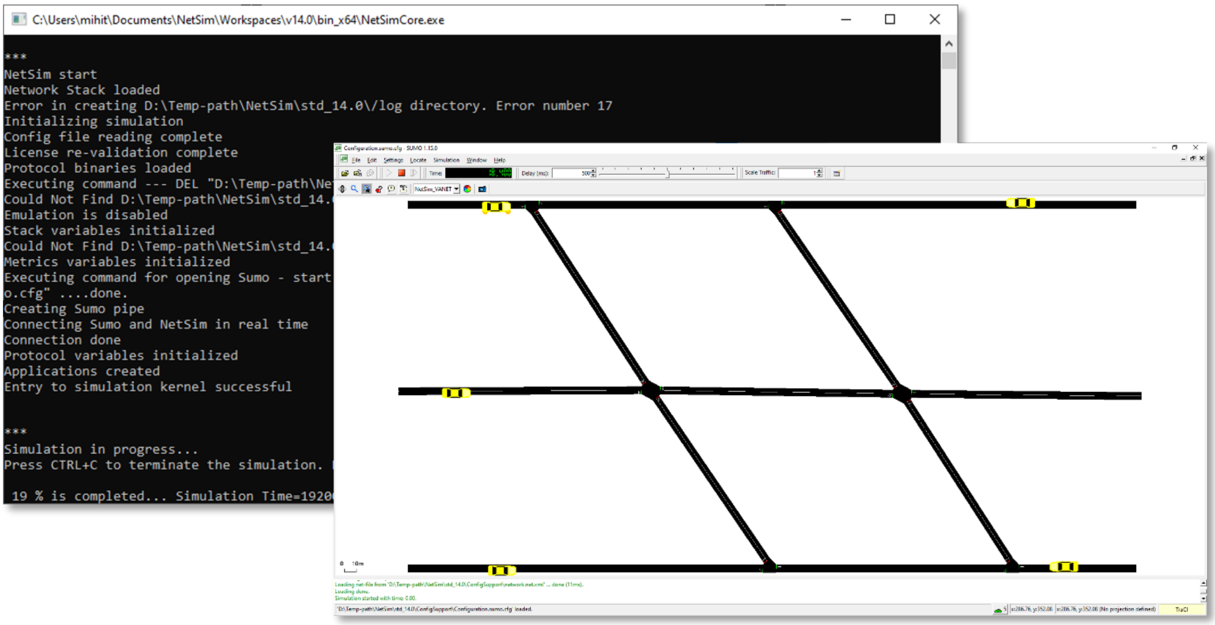

Figure 2-11: NetSim simulation window and SUMO simulation window runs simultaneously

Users can see the movement of vehicles and observe vehicular simulation in SUMO and observe equivalent simulation in NetSim.

3 Model Features

3.1 Implementation of the 802.11p in NetSim

- The Adhoc Wi-fi MAC allows for STA transmissions of data frames outside-the-context-of-a-BSS (OCB). Establishing a secure BSS necessitates announcement, scanning, synchronization, and association and the time required is extremely undesirable in vehicular environments [1]. NetSim therefore allows for direct and instantaneous link setups with no establishment of a BSS. There is no authentication nor association.
- Supports a channel bandwidth of 10 MHz in the 5.9 GHz band
- Supported PHY rates are 3, 4.5, 6, 9, 12, 18, 24 and 27 Mbps. The rate is auto determined at the sender based on the target packet error probability at the receiver (target PEP 0.1, 1000 B packets)
- Transmission type is OFDM with slot time equal to 9 μs and SIFS equal to 16 μs.
- Uses a Medium Access Control (MAC) protocol based on the Carrier Sense Multiple Access Collision Avoidance (CSMA/CA) protocol, which is explained below.
  - When a node wants to send a message, the channel must be idle for a duration of SIFS. If the channel is idle, it starts transmission.
  - When a node finds the channel busy, it chooses a random backoff time from the interval [0, CW] and transmits only when the backoff timer has elapsed. The variable CW represents the size of the Contention Window.
  - When the SCH is used and a node does not receive an acknowledgement for a message, it concludes that the message has collided and is lost, so the value of CW is doubled, and it will retry transmission.
  - In the CCH however, beacons are broadcast in the channel and no acknowledgments are sent. Therefore, the value of CW is never doubled in the CCH.

3.2 DSRC Channels: CCH and SCH

Vehicles (OBUs) and RSUs can operate in (switch between) multiple channels i.e., in the SCH and CCH as explained below.

- **Control channel (CCH):** A radio channel, intended for the exchange of management information. In NetSim when a BSM (safety) application is configured, it is transmitted on the CCH.
- **Service channel (SCH):** These are radio channels used for non-safety applications. In NetSim, when non safety application such as CBR, Voice, Video, FTP etc., are configured, they are transmitted on the CCH.

- **Guard interval:** A time interval at the start of each control channel (CCH) interval and service channel (SCH) interval during which devices cannot transmit.

Each synchronization interval SI is split as follows

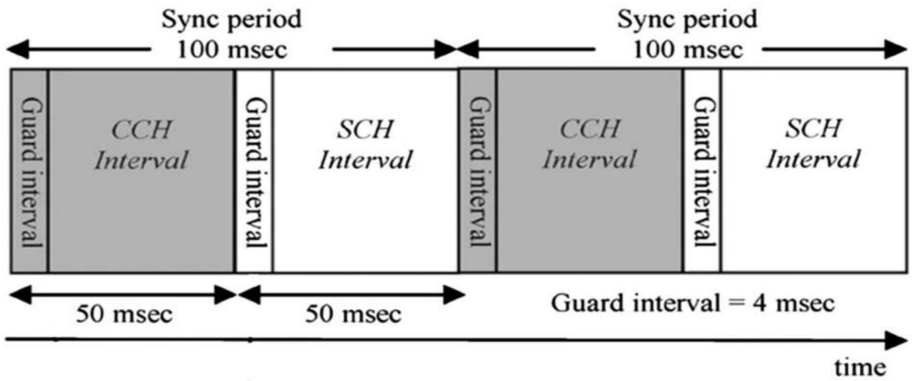

Figure 3-1: We see the time divisions in DSRC. Each synchronization period consists of 1 CCH, 1 SCH and 1 guard interval. While the sync period is generally equal to 100 ms. NetSim allows users to modify the CCH and SCH interval, and in turn the total synchronization period.

All devices (Vehicles and RSUs) switch between SCH and CCH and the alternation is based on the time divisions. NetSim allows the user to configure values of CCH interval, SCH interval and Guard interval. NetSim supports 6 service channels (SCH):172 (5860 MHz), 174 (5870MHz), 176 (5880 MHz), 180(5900MHz), 182(5910MHz) and 184 (5920 MHz), and 1 control channel (CCH): 178 (5890 MHz). The default channels used in NetSim are SCH 171 (5.855 GHz) and CCH 178 (5.890 GHz).

3.3 BSM Application

- DSRC protocol runs with BSM (Basic Safety Message) applications. BSM is a broadcast packet transmitted at a regular interval
- The BSM Application class sends and receives the IEEE 1609 WAVE (Wireless Access in Vehicular Environments) Basic Safety Messages (BSMs). The BSM is a 20-byte packet that is generally broadcast from every vehicle at a nominal rate of 10 Hz. In NetSim this can be configured as a broadcast or as a unicast application. Note that a broadcast application can only be transmitted over a single hop. NetSim does not transmit broadcast applications over multiple hops.
- This application does not follow the IP stack. It runs WSMP protocol over IEEE 1609. There is no routing; static routes cannot be set, and packets are sent directly to the destination.

3.4 NetSim – SUMO interfacing

NetSim's VANET module allows users to interface with SUMO which is an open-source road traffic simulation package designed to handle vehicular & road networks. The road traffic

simulation is done by SUMO while NetSim does the network simulation along with RF propagation modelling in the physical layer. While SUMO Simulates the road traffic conditions and movements, NetSim Simulates the communication occurring between the Vehicles.
NetSim and SUMO are interfaced using 'pipes'. A pipe is a section of shared memory that processes use for communication. SUMO process writes information to pipe, then NetSim process reads the information from pipe. On running the Simulation, SUMO determines the positions of vehicles with respect to time as per the road conditions. NetSim reads the coordinates of vehicles from SUMO (through pipes) in runtime and uses it as input for vehicles mobility.
Users will notice an inversion along X axis in the NetSim GUI, since origin (0, 0) in SUMO is at the left bottom, while origin is at the left top in NetSim.
VANET operates in wireless environment and hence RF channel loss occurs. The amount of loss can be configured by users. To modify the Wireless channel characteristics users can right click on the adhoc/wireless link and modify the channel characteristics as per the requirement.
Source code related to interfacing of SUMO and NetSim is available in Sumo_interface.c file inside the mobility folder/project.

3.5 How to create a VANET using SUMO and simulate with NetSim

A SUMO network can be created either **manually** or using **SUMO NetEdit.**

3.5.1 Using SUMO NetEdit utility and randomtrips.py to configure road traffic models

Netedit is a Road network editor for the road traffic simulation in SUMO. Using this utility, users can quickly design road networks and obtain Network xml file which is part of SUMO configuration.

Steps to create a simple SUMO network using netedit utility

Step 1: Open **netedit** from **<SUMO_INSTALL_DIRECTORY>/bin** (C:\Program Files (x86)\Eclipse\Sumo\bin) and select **File-->New Network**

Refer SUMO Documentation: *"http://sumo.dlr.de/wiki/NETEDIT" for more details on modes of operation.*

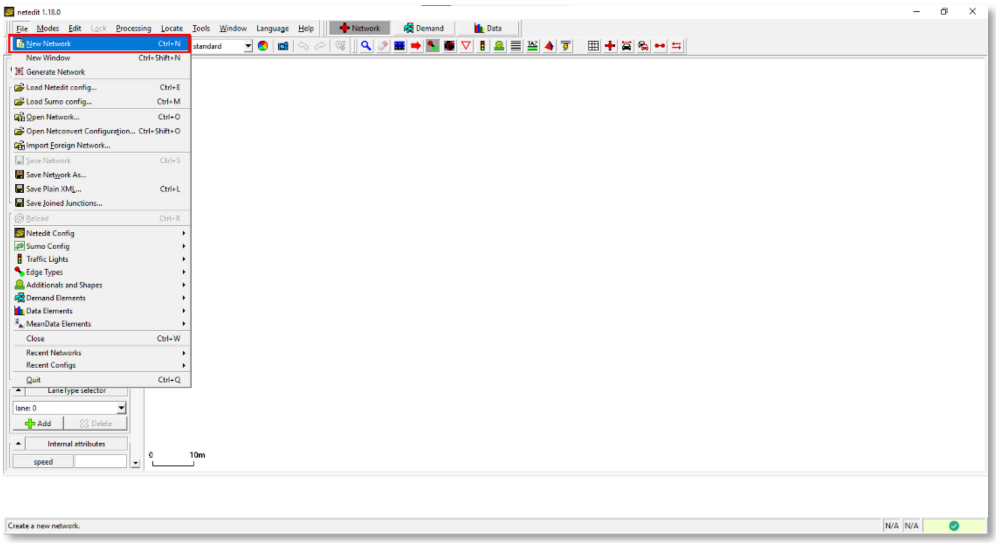

Figure 3-2: SUMO NetEdit New Network Screen

Step 2: Select Creating junction and edges option as shown below or click on character "**e**" in the keyboard.

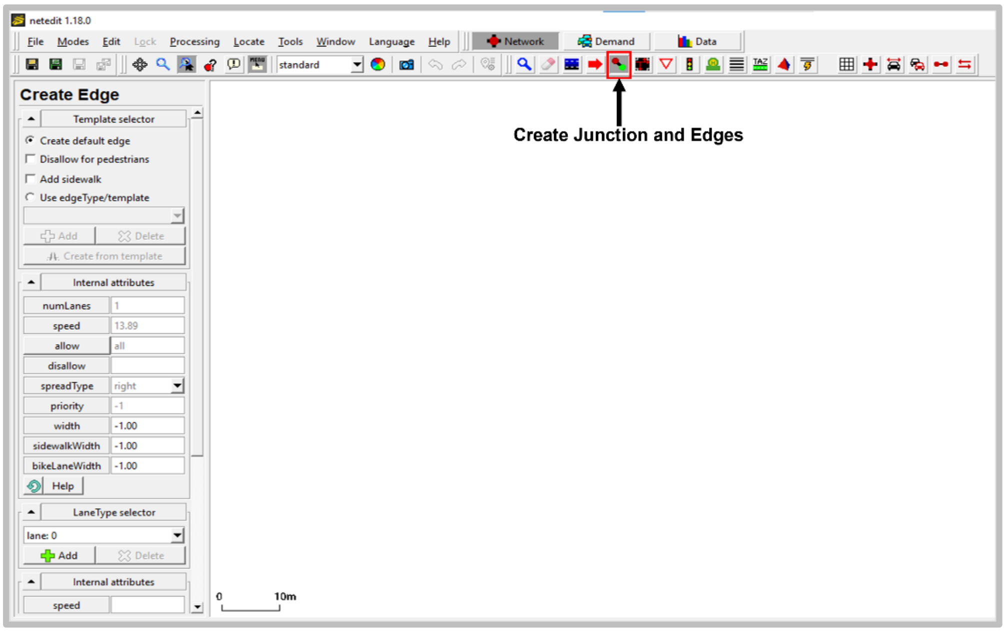

Figure 3-3: SUMO NetEdit Screen

Step 3: Enable the check boxes "**chain**", "**two-way**" **and "Grid"** which are present in the right-side corner.

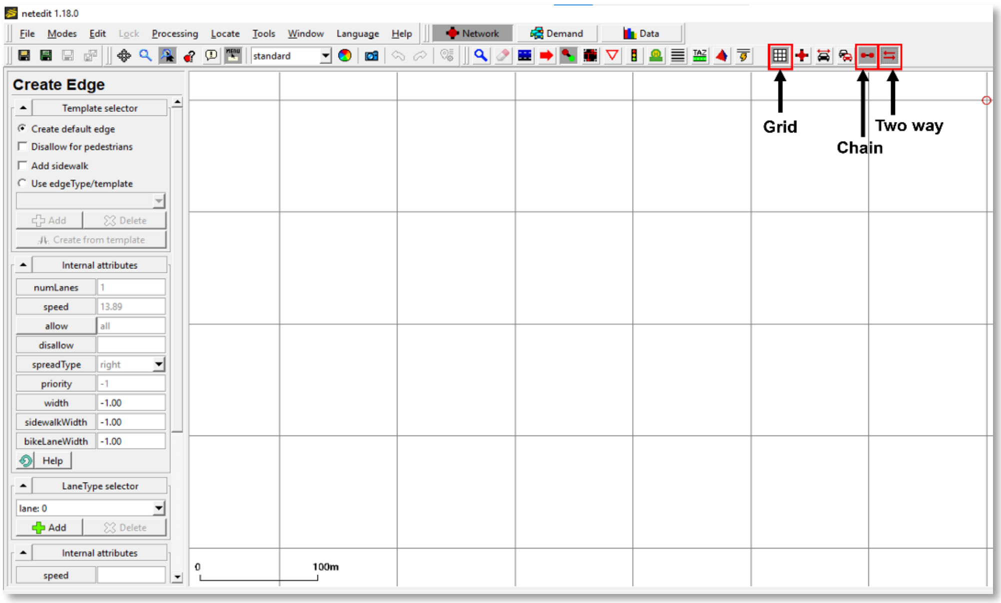

Figure 3-4: SUMO NetEdit design window

Step 4: Adjust the design area to ensure that the road network lies in the **Positive XY** quadrant. This will help in avoiding complexities when opening the network scenario in NetSim.

Step 5: Click on grid area to create edges, clicking again will create a new edge which will automatically get connected to the previous edge as shown below.

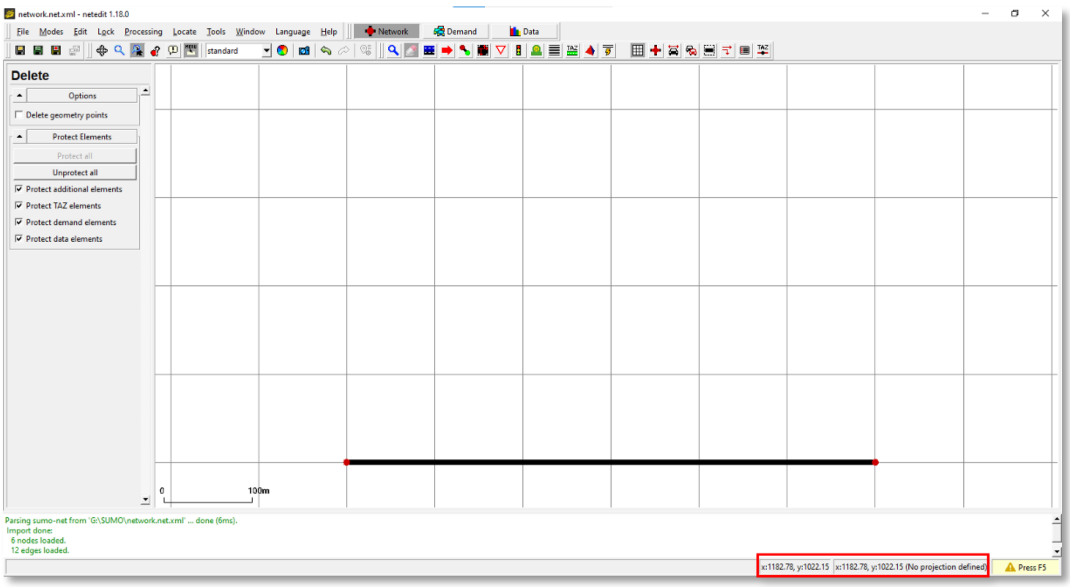

Figure 3-5: Creating edges SUMO NetEdit design window

Step 6: Select "**(t) Traffic Lights**". Select the junctions and click on **Create TLS** button on the left to add Traffic Signal to it.

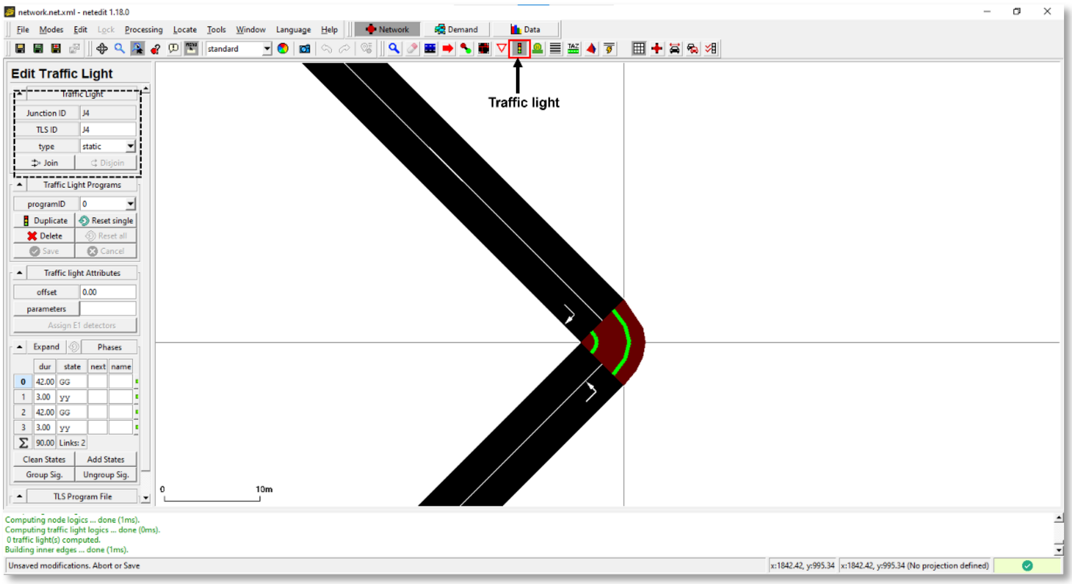

Figure 3-6: Adding Traffic Signal to Network

Step 7: Select "**(c) Connect**" icon Select the lanes and ensure connectivity between them.

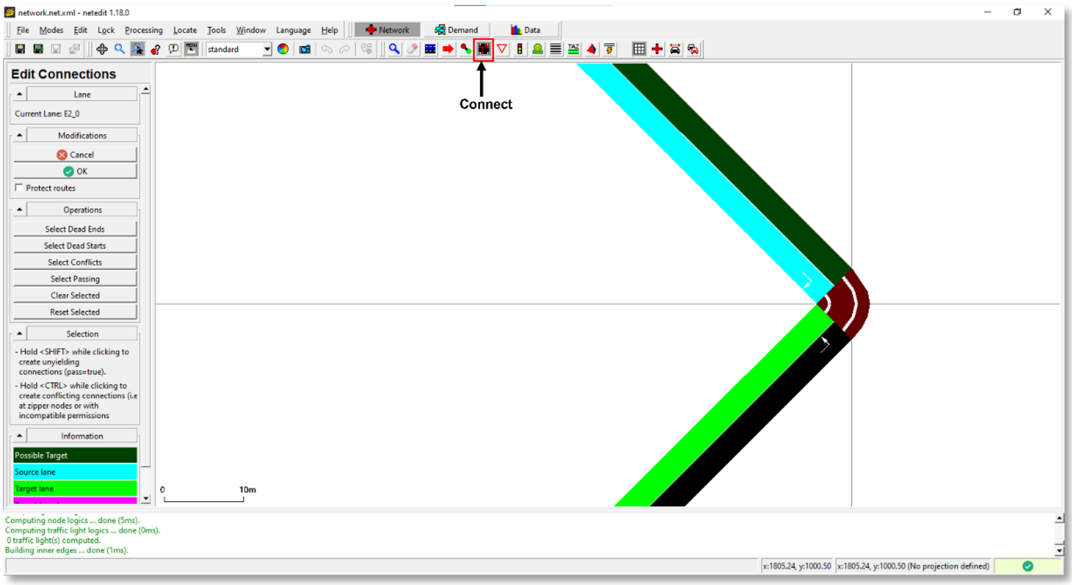

Figure 3-7: Select the lanes and connectivity between them

Step 8: Create a new folder and save the network file (***.net.xml**) over there, say with a name **network.net.xml**

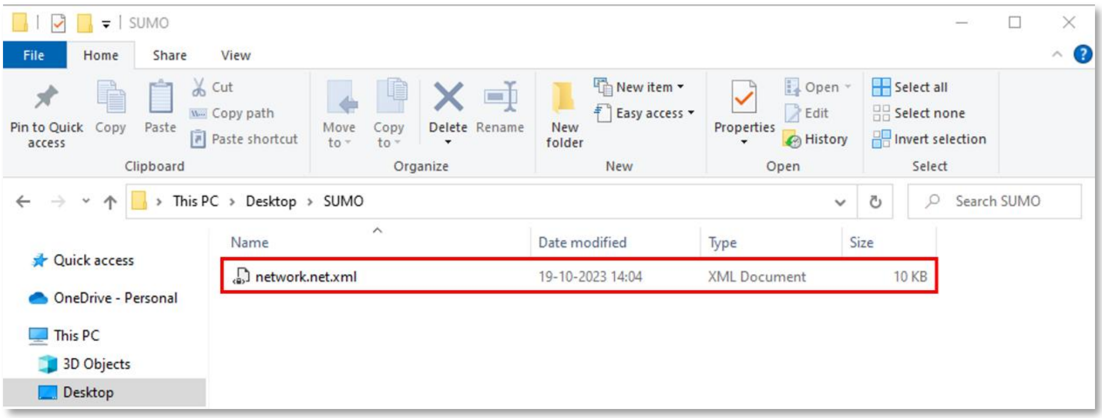

Figure 3-8: network.net.xml inside the folder

Step 9: Open command prompt with the current working directory as the folder where you have saved the network file in the previous step.

Step 10: Using **randomtrips.py** utility present in **<SUMO_INSTALL_DIRECTORY>/tools** directory create trips file with the command.

COMMAND SYNTAX >" C:\Program Files (x86)\Eclipse\Sumo\tools\randomTrips.py" -n " ***.net.xml**" -e <**NO_OF_TRIPS**> --route-file "**trips.xml**"

Example Command >" C:\Program Files (x86)\Eclipse\Sumo\tools\randomTrips.py" -n "network.net.xml" -e 5 --route-file "trips.xml"

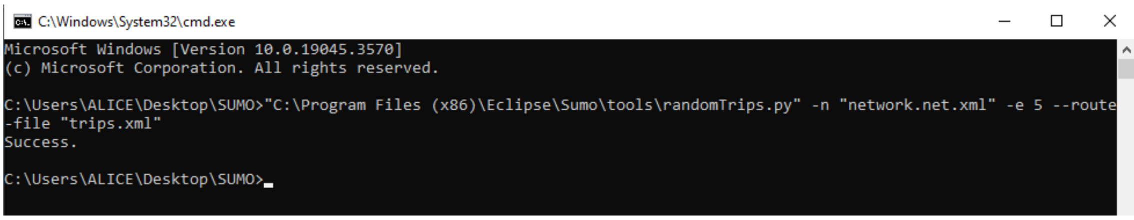

Figure 3-9: Generating route file (trips.xml)

This will create a trips file in your folder along with associated files.

Step 11: Create a “file-settings.xml” file in your folder which contains the network and route file for improving the visualization of vehicles and their movements in a graphical user interface (GUI).

Following is a “file-settings.xml” file:

<viewsettings>

<scheme name="NetSim_VANET">

<vehicles vehicleQuality="2" vehicle_exaggeration="4.00">

</vehicles>

</scheme>

<delay value="500"/>

</viewsettings>

Step 12: Create a SUMO configuration file (***sumo.cfg**) which points to the network and trips file, in your folder which contains the network and route file.

Refer: http://sumo.dlr.de/wiki/Tutorials/Hello_Sumo

Include parameter (To Run in NetSim)

"<step-length value="0.4"/>"

Following is a sample SUMO Configuration:

<configuration>

<input>

<net-file value="network.net.xml"/>

<route-files value="trips.xml"/>

<gui-settings-file value="file.settings.xml"/>

</input>

<time>

<begin value="0"/>

<end value="100"/>

<step-length value="0.4"/>

</time>

</configuration>

NOTE: *Save above content as* ***Configuration.sumo.cfg***

You can copy the above contents to create a SUMO configuration file in your folder.

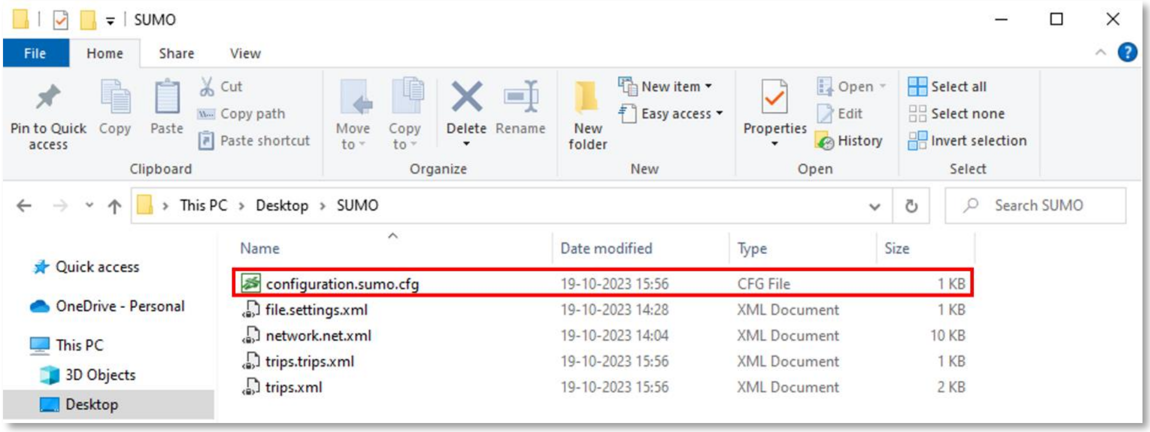

Figure 3-10: Double click on Configuration.sumo.cfg

Step 12: Open Configuration.sumo.cfg by double clicking or open SUMO using **sumo-gui.exe** present in **<SUMO_INSTALL_DIRECTORY>/bin**. Open scenario in SUMO using **Open->Simulation** and verify whether the network loads and simulation happens as per the configuration done.

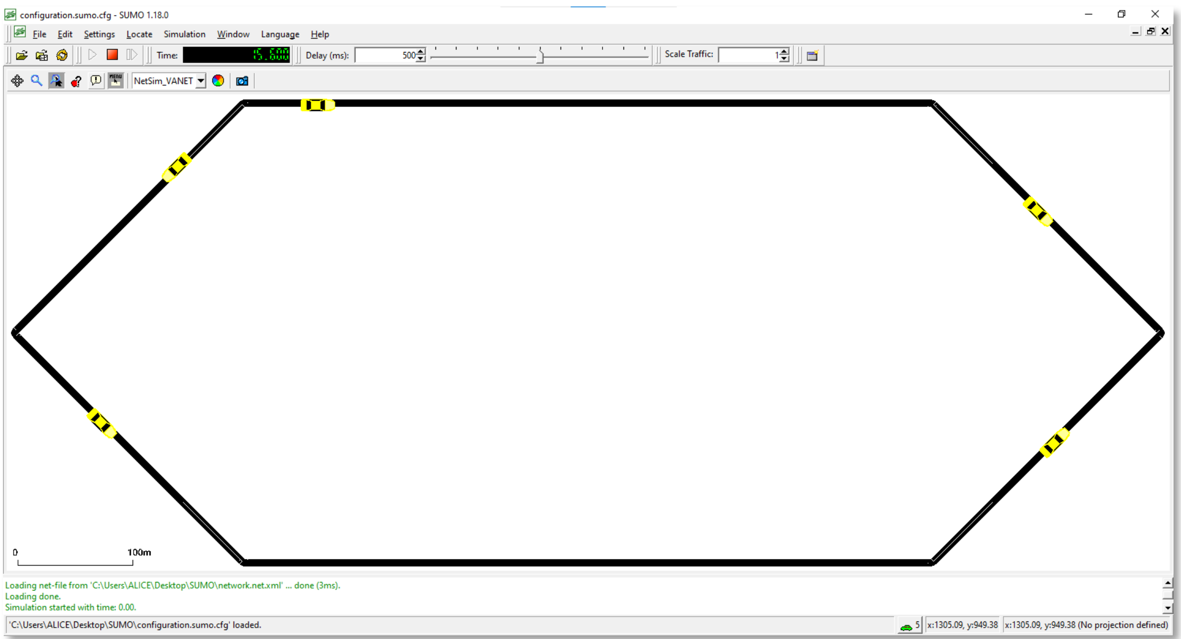

Figure 3-11: SUMO simulation window

Step 13: Open the SUMO scenario via NetSim VANET by selecting VANET under the New Simulation in the NetSim Home Screen. Browse and locate the SUMO Configuration file present in your directory to load the road traffic network in NetSim GUI. The road network created in SUMO will be automatically replicated in NetSim GUI environment.

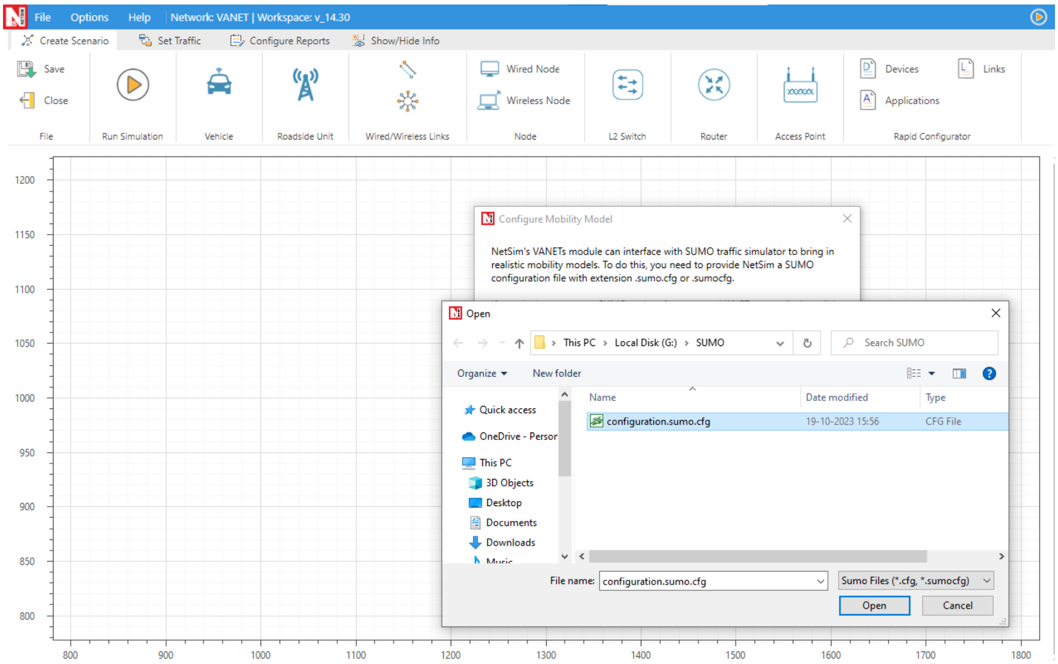

Figure 3-12: Importing a sumo network configuration into NetSim

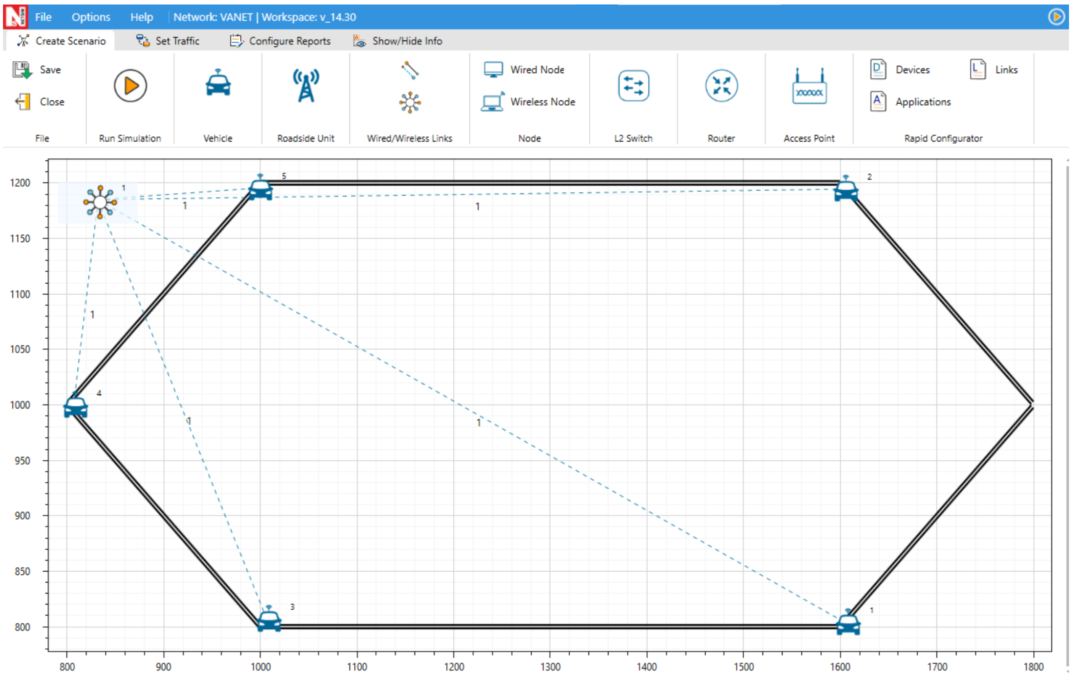

Figure 3-13: Network Topology in this experiment

Step 14: Configure traffic between vehicles using the Application icon, enable trace files and plots.

Step 15: Click on Run Simulation button. The **Simulation Time** is equal to the end time specified in sumo configuration (**sumo.cfg**) file and Simulation Time option is Non editable.

3.5.2 Creating your own network in SUMO manually

Step 1: Create a node file using any code editor (like notepad, notepad++ etc.) and the file extension will be.nod.xml. It represents the junctions in the road. Each of these attributes has a certain meaning and value range: node_id means unique name of each junction, x-y is the positions of node and type can be "priority", "traffic_light", "rail_crossing", “rail_ signal”etc.(Refer: https://sumo.dlr.de/wiki/Networks/PlainXML#Node_Descriptions).

```
NODES.nod.xml
<nodes>
    <node id="n0" x="100" y="300" type="priority"/>
    <node id="n1" x="300" y="300" type="traffic_light"/>
    <node id="n2" x="500" y="300" type="rail_crossing"/>
    <node id="n3" x="700" y="300" type="priority"/>
    <node id="n4" x="300" y="100" type="traffic_light"/>
    <node id="n5" x="500" y="100" type="traffic_light"/>
</nodes>
```

Figure 3-14: Device Positions in nodes file

Step 2: Create an edge file that describes how the junctions or nodes are connected to each other. The extension of this file is .edg.xml. Each edge is unidirectional and starts at the "from"-node and ends at the "to"-node. For each edge, some further attributes should be supplied, being the number of lanes, the edge has (numLanes), the maximum speed allowed on the edge speed. Furthermore, the priority may be defined optionally. (Refer: https://sumo.dlr.de/wiki/Networks/PlainXML#Edge_Descriptions).

```
EDGE.edg.xml
<edges>
  <edge id="L01" from="n1" to="n0" priority="75" nolanes="2" speed="40"/>
  <edge id="R01" from="n0" to="n1" priority="75" nolanes="2" speed="40"/>
  <edge id="L12" from="n2" to="n1" priority="75" nolanes="2" speed="40"/>
  <edge id="R12" from="n1" to="n2" priority="75" nolanes="2" speed="40"/>
  <edge id="L23" from="n3" to="n2" priority="75" nolanes="2" speed="40"/>
  <edge id="R23" from="n2" to="n3" priority="75" nolanes="2" speed="40"/>
  <edge id="D04" from="n0" to="n4" priority="75" nolanes="2" speed="40"/>
  <edge id="U04" from="n4" to="n0" priority="75" nolanes="2" speed="40"/>
  <edge id="D14" from="n1" to="n4" priority="75" nolanes="2" speed="40"/>
  <edge id="U14" from="n4" to="n1" priority="75" nolanes="2" speed="40"/>
  <edge id="D35" from="n3" to="n5" priority="75" nolanes="2" speed="40"/>
  <edge id="U35" from="n5" to="n3" priority="75" nolanes="2" speed="40"/>
  <edge id="D25" from="n2" to="n5" priority="75" nolanes="2" speed="40"/>
  <edge id="U25" from="n5" to="n2" priority="75" nolanes="2" speed="40"/>
  <edge id="L45" from="n5" to="n4" priority="75" nolanes="2" speed="40"/>
  <edge id="R45" from="n4" to="n5" priority="75" nolanes="2" speed="40"/>
</edges>
```

Figure 3-15: Edge file

Step 3: Open Command Prompt and change the directory to the binary folder of sumo using cd command. **“cd C:\Program Files (x86)\Eclipse\Sumo\bin”**

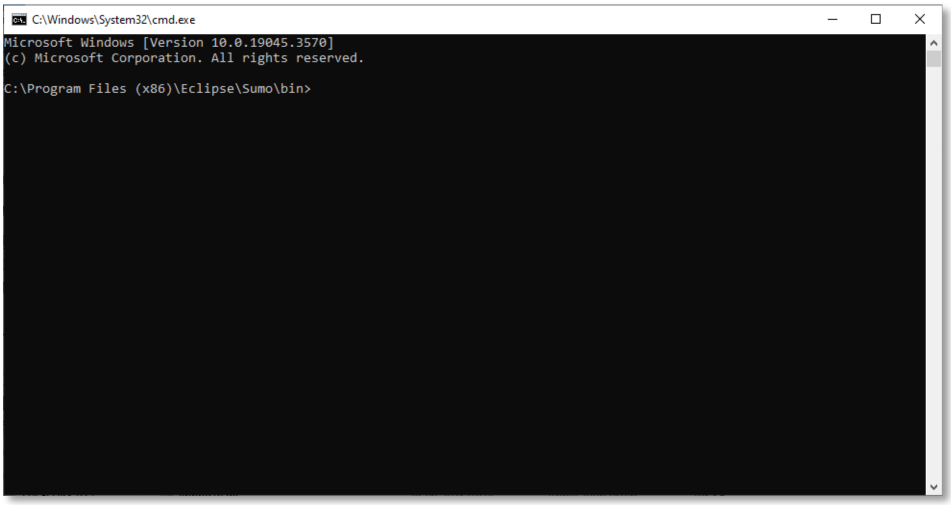

Figure 3-16: Open command prompt in installation directory

Step 4: Generate Network file by using NETCONVERT command. Make a folder named like VANET_Example and place the. nod.xml and .edg.xml files i.e. NODES.nod.xml and EDGE.edg.xml respectively.

netconvert --n "<path where the.nod.xml file is present>\<filename>.nod.xml" --e "<path where the .edg.xml file is present>\<filename>.edg.xml" --o "<path where both input files are present>\<filename>.net.xml"

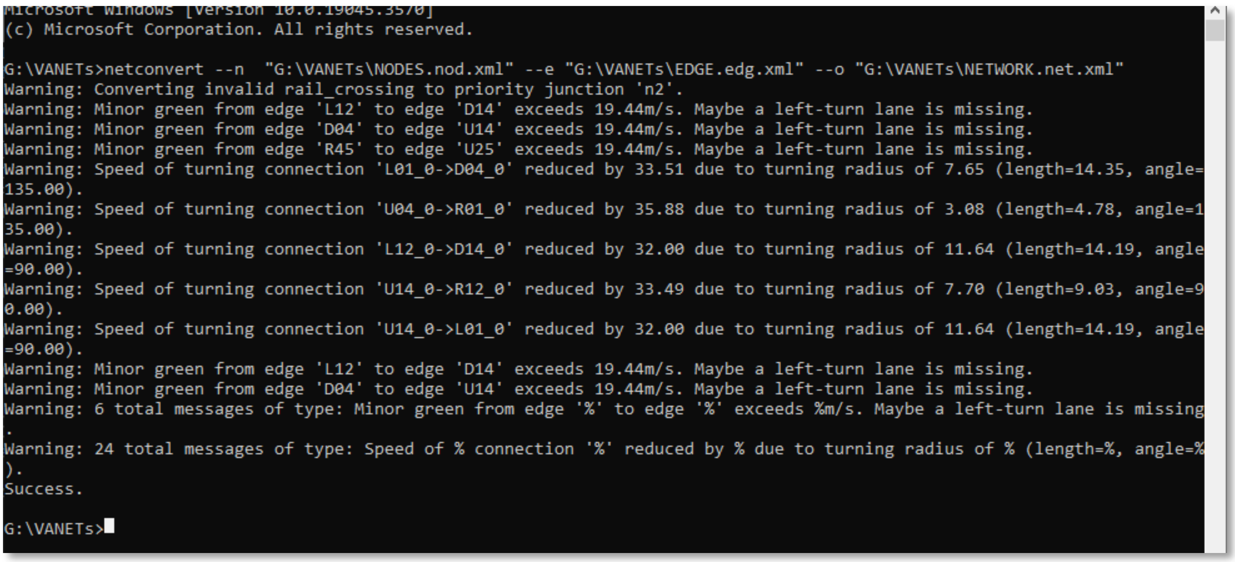

Figure 3-17: Generating Network file by using NETCONVERT command

Step 5: Create a .rou.xml file that describes the direction of the vehicle's movement.

```xml
<routes>
<!-- e -->
<vehicle id="V0" depart="0.00">
<route edges="D04 R45 U35 L23 L12 L01" />
</vehicle>
<vehicle id="V1" depart="0.00">
<route edges="R12 R23 D35 L45 U14 L01" />
</vehicle>
<vehicle id="V2" depart="0.00">
<route edges="R01 R12 R23 D35 L45 U04" />
</vehicle>
<vehicle id="V3" depart="0.00">
<route edges="R01 D14 R45 U35 L23 L12" />
</vehicle>
<vehicle id="V4" depart="0.00">
<route edges="D35 L45 U14 R12 D25 U35 L23" />
</vehicle>
<vehicle id="V5" depart="0.00">
<route edges="R45 U25 L12 L01 D04 R45" />
</vehicle>
</routes>
```

Figure 3-18: Direction of the vehicle's movement

Step 6: Create a sumo configuration file file using any code editor (like notepad, notepad++ etc.) and the extension is. sumo.cfg. Place the file inside the same folder where the network file (i.e. NETWORK.net.xml) and route file (i.e. VEHICLES.rou.xml) are present.

```xml
<Configuration>
<input>
<net-file value="NETWORK.net.xml"/>
<route-files value="VEHICLES.rou.xml"/>
</input>
<time>
<begin value="0"/>
<end value="200"/>
<step-length value="1"/>
</time>
</Configuration>
```

Figure 3-19: Sumo configuration file

Step 7: Now open "New Simulation → VANET". Choose the Configuration.sumo.cfg from the specified folder and run simulation using NetSim.

3.6 How to include Roadside Units (RSU's) in a VANET network?

Upon importing a sumo network configuration into NetSim, roads and vehicles are automatically added in NetSim as per the configuration done in SUMO.

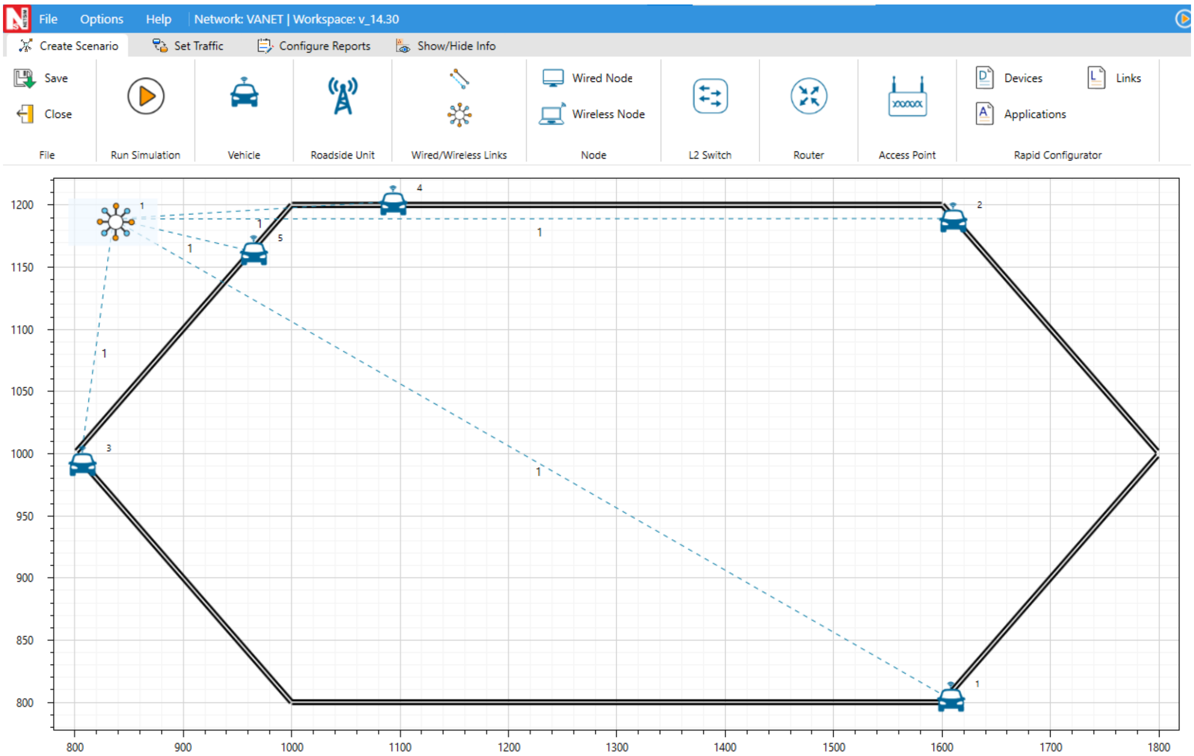

Figure 3-20: Importing a sumo network configuration into NetSim

Roadside Units can be optionally included in the network by manually clicking and dropping the RSU icon from the ribbon.

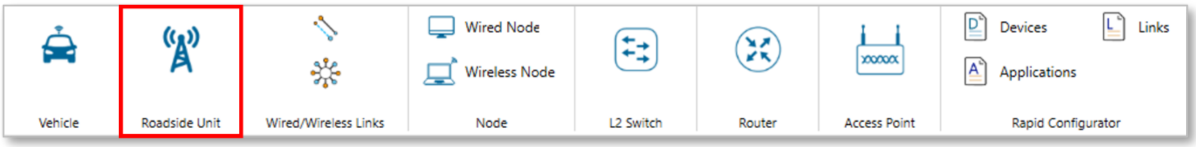

Figure 3-21: RSU icon from the ribbon

RSU’s should be connected using Adhoc links manually.

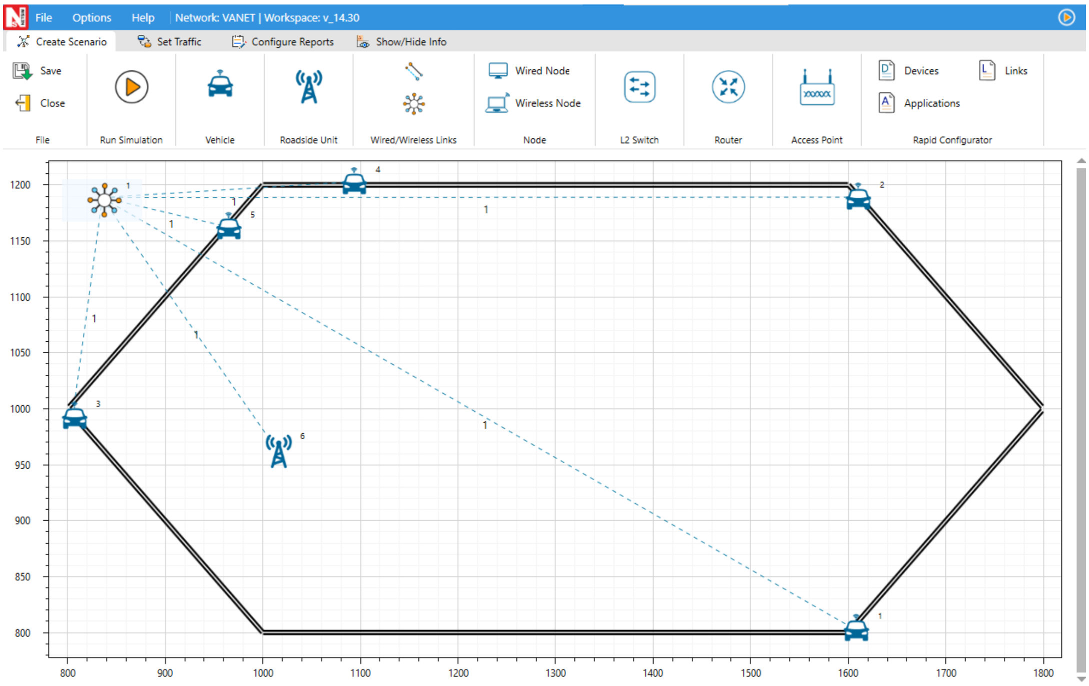

Figure 3-22: NetSim Design Window

Traffic can be configured between RSU's and vehicles via Application configuration.

4 Featured Examples

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

4.1 Importing a simple VANET scenario from SUMO

Open NetSim and Select **Examples > VANETs > Importing a simple VANET scenario from SUMO** then click on the tile in the middle panel to load the example as shown in below screenshot.

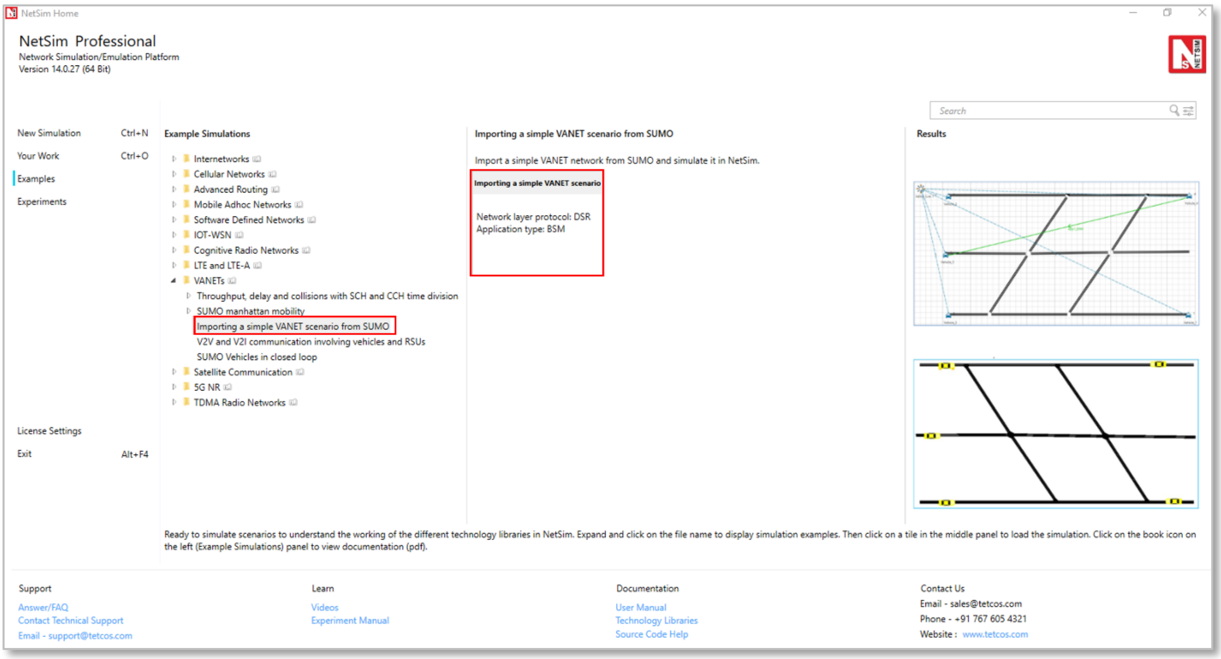

Figure 4-1: List of scenarios for the example of Importing a simple VANET scenario from SUMO

This scenario involves a simple road traffic network scenario as shown below:

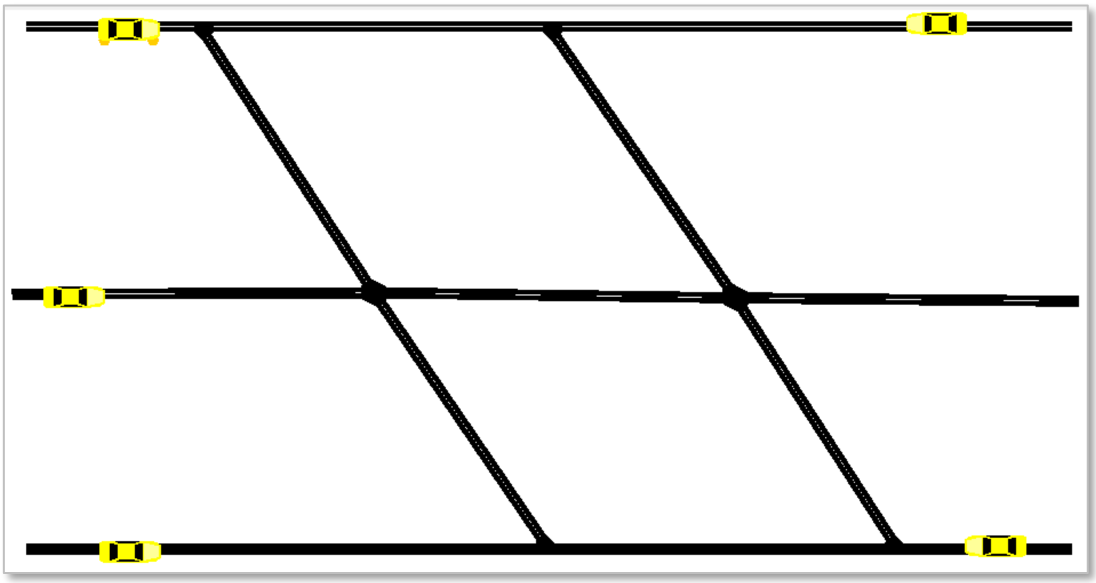

Figure 4-2: Network topology in Sumo

An equivalent network is created in NetSim by importing the SUMO configuration file. In NetSim the TCP/IP stack parameters of the devices are configured along with network traffic between vehicles.

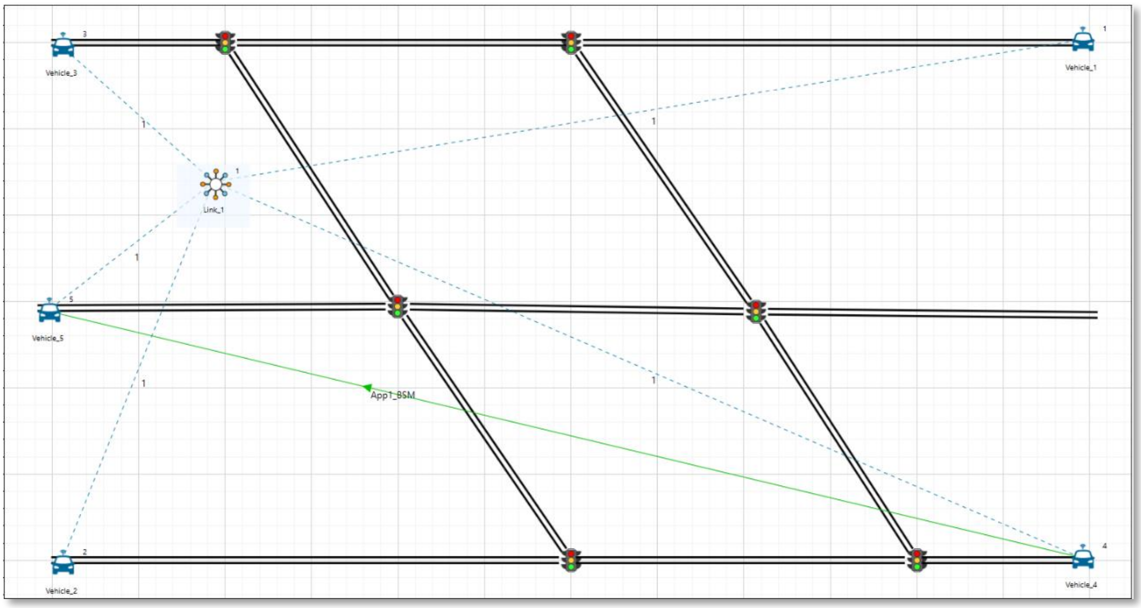

Figure 4-3: Network scenario after importing into NetSim and configuring application traffic

The properties of Application, vehicle and link are set as follows:

Properties	
Application Properties	
Application Type	BSM
Application Method	UNICAST
Packet Size	20 (Bytes)
Inter Arrival Time	1000000(μs)
Link Properties	

Channel Characteristics	No Pathloss
Physical Layer Properties (Vehicle)	
Standard	IEEE802.11p
Transmitter Power	40 mW
Antenna Gain	1 dBi
Antenna Height	1 dBi
Bandwidth	10 MHz

Table 4-1: Application, Link and Physical layer Properties

After running the simulation, Packet Trace can be used to visualize packet flow along with packet information and Mobility log can be used to record vehicle mobility. Time varying throughput plot can be opened from the Results window.

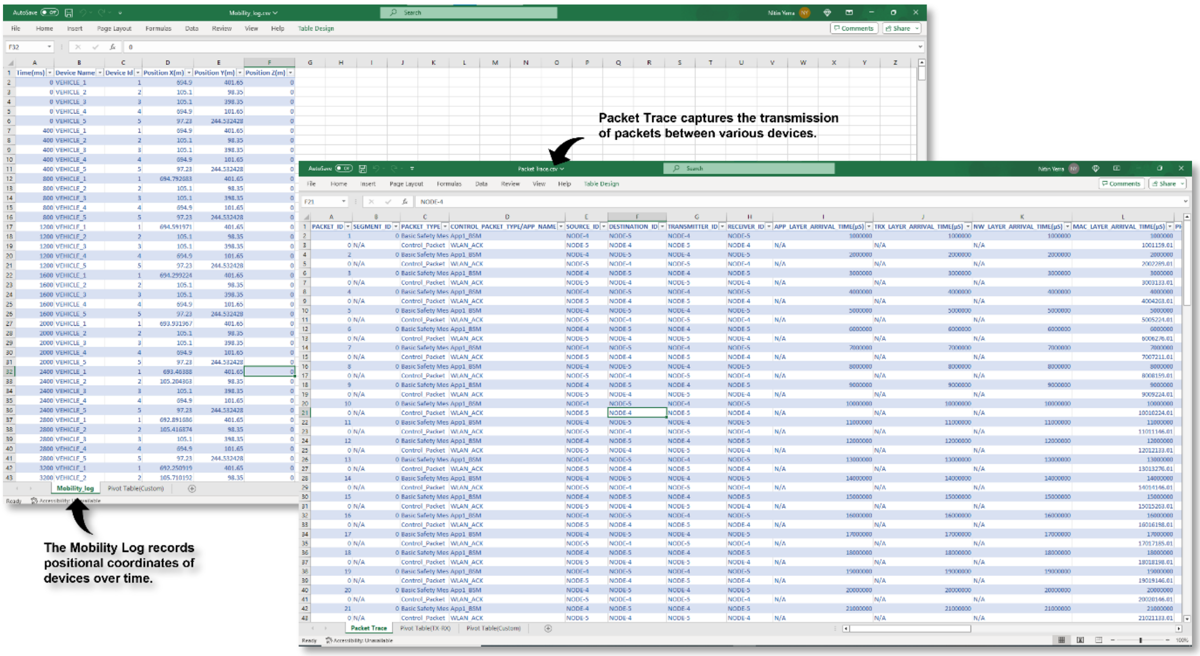

Figure 4-4: : Packet Trace and Mobility log window

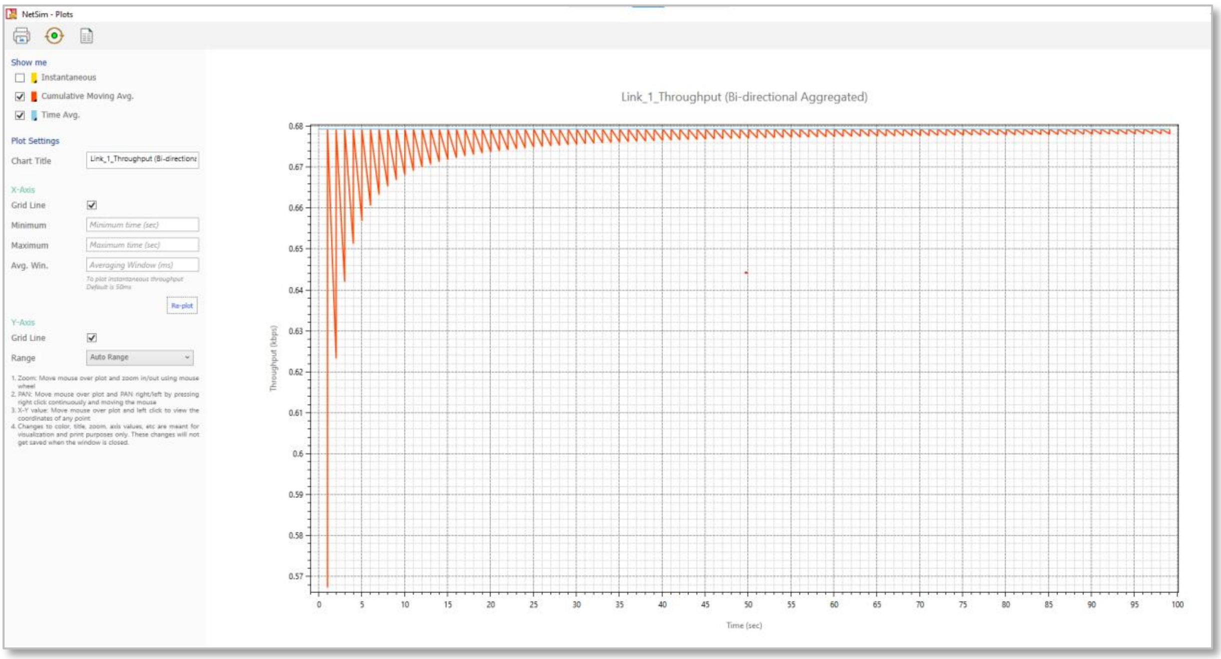

Figure 4-5: Link throughput plot

Simulation results dashboard provides the performance metrics of protocols running in different layers of the network stack of the devices.

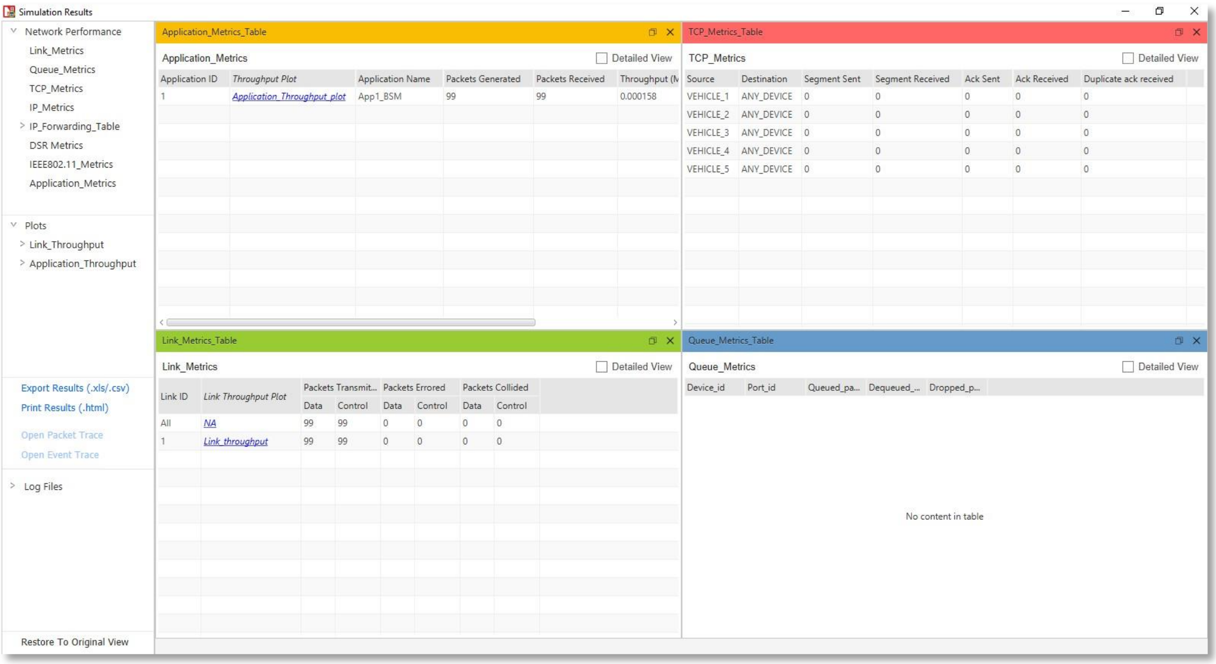

Figure 4-6: Result Dashboard

4.2 V2V and V2I communication involving Vehicles and RSUs

Open NetSim and Select **Examples > VANETs > V2V and V2I communication involving Vehicles and RSUs** then click on the tile in the middle panel to load the example as shown in below screenshot

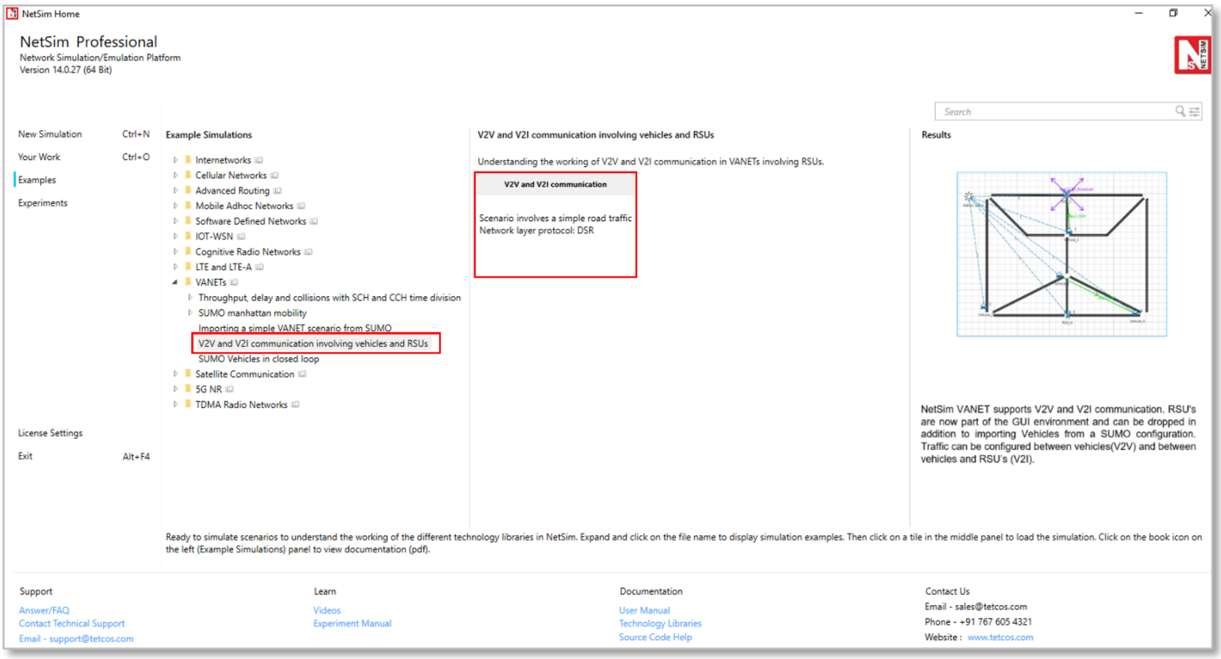

Figure 4-7: List of scenarios for the example of V2V and V2I communication involving Vehicles and RSUs

NetSim VANETs module supports V2V and V2I communication. RSU's are now part of the GUI environment and can be dropped in addition to importing Vehicles from a SUMO configuration. Traffic can be configured between vehicles (V2V) and between vehicles and RSU's (V2I).

This scenario involves a simple road traffic network scenario as shown below:

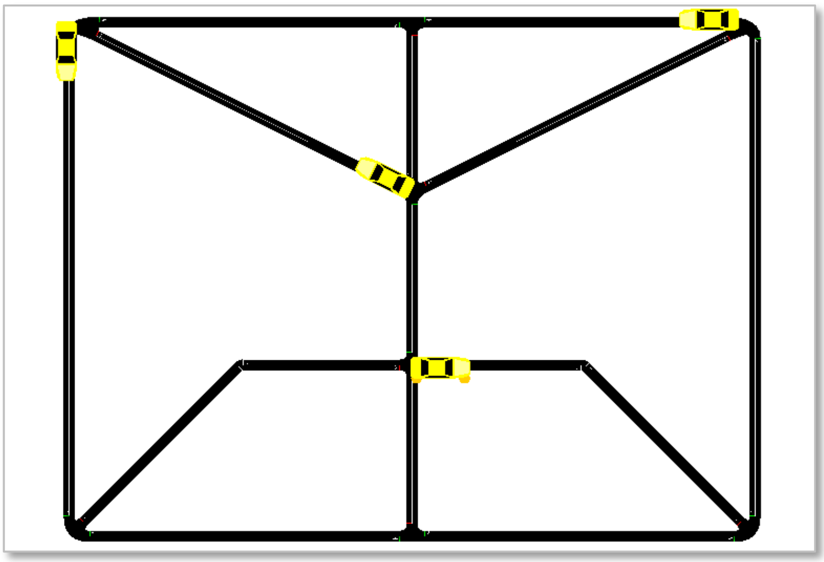

Figure 4-8: Network topology in Sumo

An equivalent network is created in NetSim by importing the SUMO configuration file as shown below:

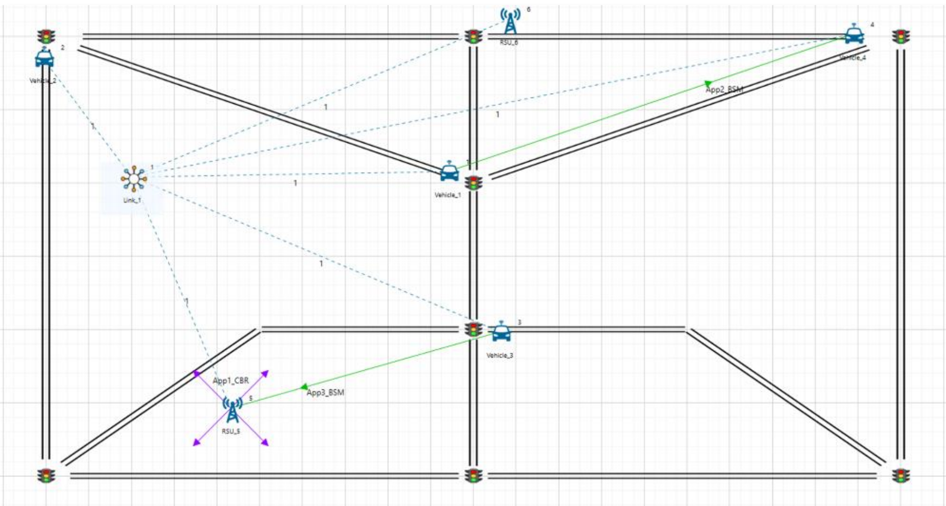

Figure 4-9: Network set up for scenario involving vehicles and RSUs for v2v and v2i communication

After importing the SUMO configuration file in NetSim, RSU's were added at junctions. In NetSim the TCP/IP stack parameters of the devices are configured along with network traffic between vehicles and between vehicles and RSU's.

Settings done for the Experiment:

Properties	
Application-1 Properties	
App Type	CBR
Application Method	UNICAST
Transport Protocol	UDP
Packet Size	1460 (Bytes)
Inter Arrival Time	1000000(µs)

Application-2 and 3 Properties	
App Type	BSM
Application Method	UNICAST
Packet Size	1460 (Bytes)
Inter Arrival Time	1000000(µs)
Link Properties	
Channel Characteristics	No Pathloss
Physical Layer Properties (Vehicle's & RSU)	
Standard	IEEE802.11p
Transmitter Power	40 mW
Antenna Gain	1 dBi
Antenna Height	1 dBi
Bandwidth	10 MHz

Table 4-2: Application, Link and Physical layer Properties

After running the simulation, Packet Trace can be used to visualize packet flow along with packet information and mobility log can be used to record vehicle mobility. Time varying throughput plot can be opened from the Results window.

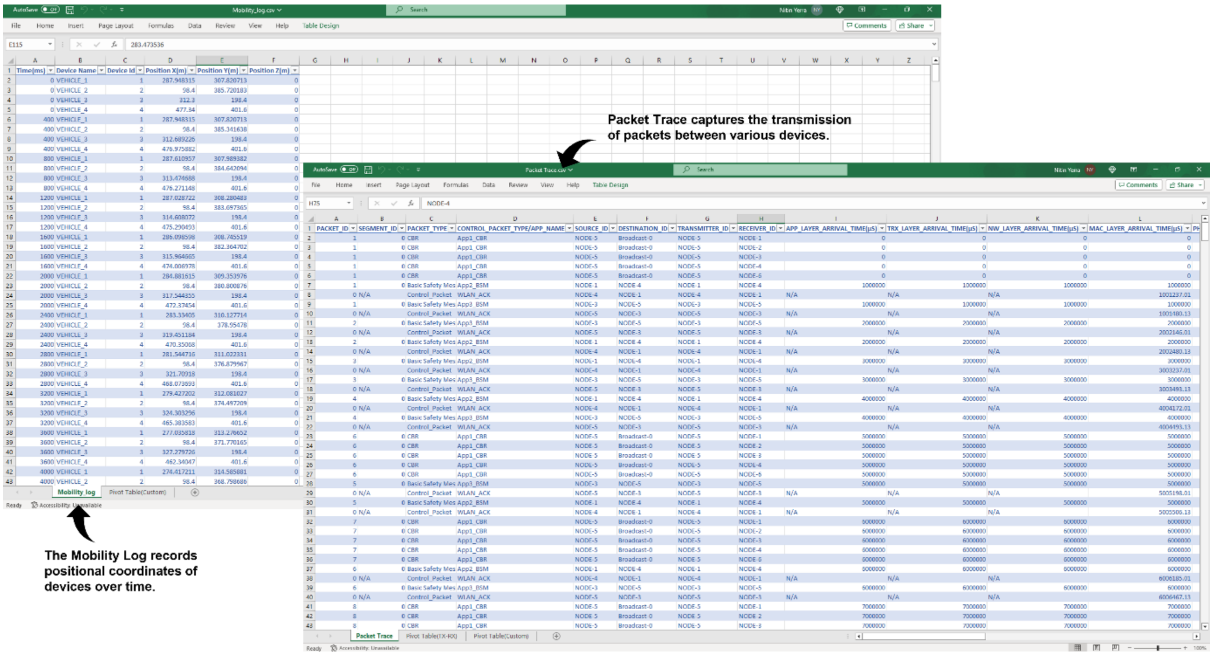

Figure 4-10: Packet Trace and Mobility log window

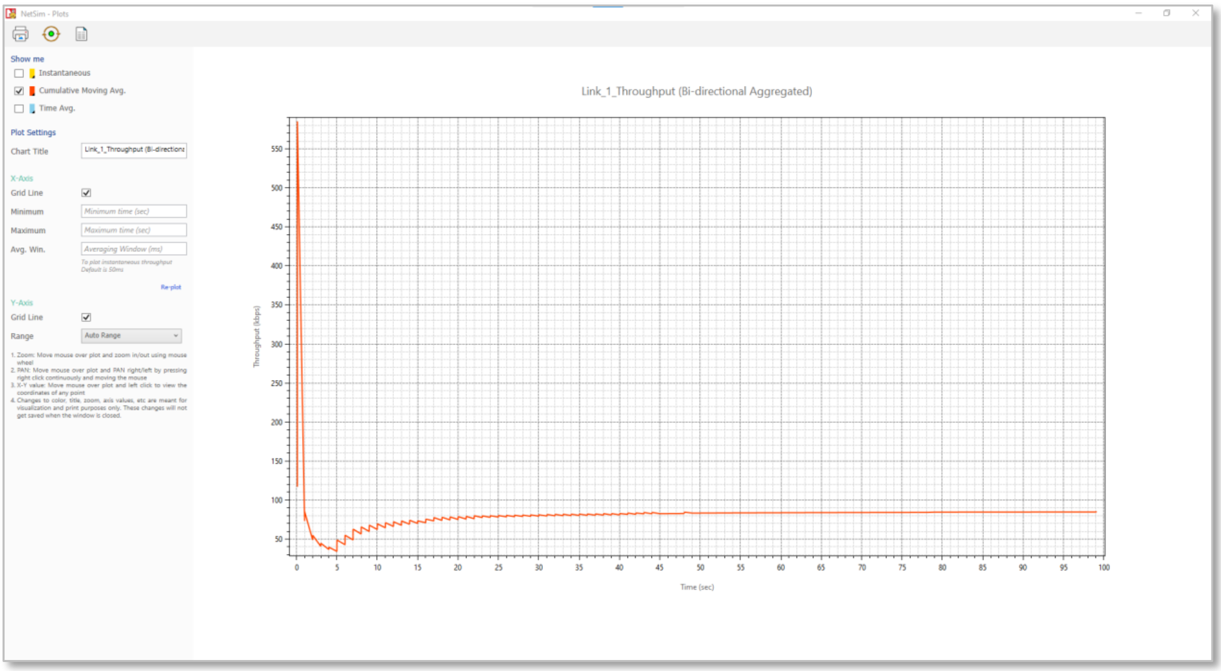

Figure 4-11: link throughput plot

Simulation results dashboard provides the performance metrics of protocols running in different layers of the network stack of the devices.

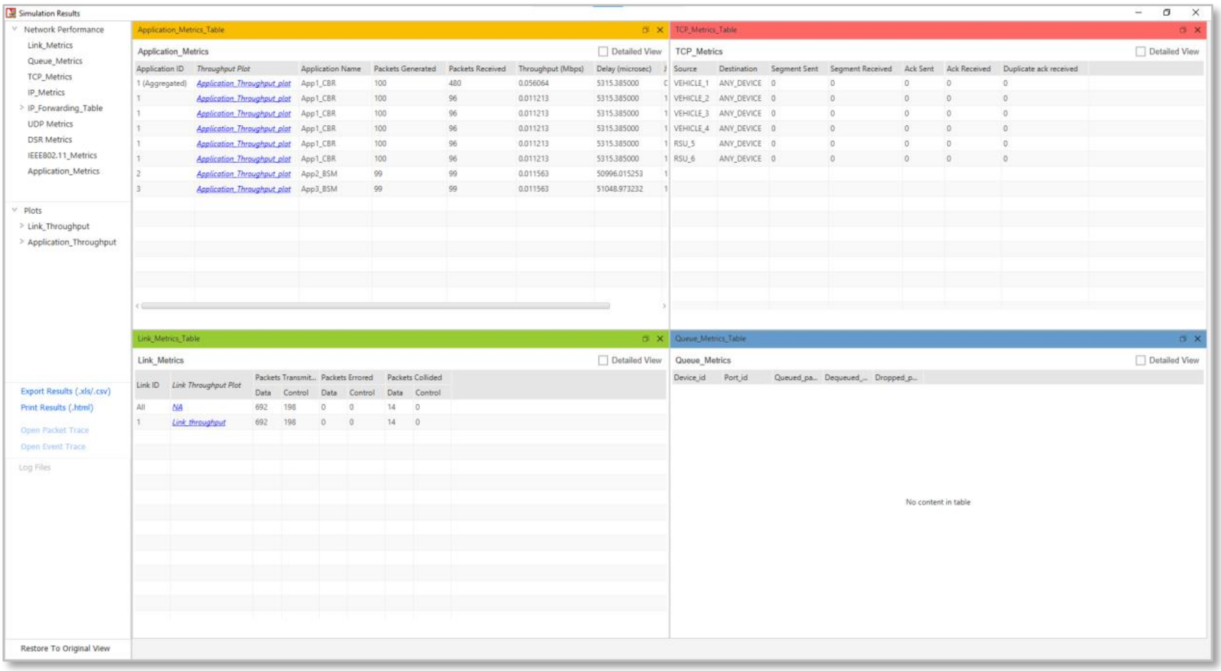

Figure 4-12: Result Dashboard

4.3 Throughput, delay and collisions with SCH and CCH time division

All the following examples are available in NetSim GUI. Navigate to Example > VANETs > Throughput, delay and collisions with SCH and CCH time division. Within Throughput, delay and collisions with SCH and CCH time division users will see four folders. Each folder comprises of simulation samples for Parts 1, 2, 3 and 4 as explained below.

4.3.1 Background

Dedicated short range communication (DSRC) which uses two channels: Service channel SCH and Control channel (CCH). Each synchronization interval SI is split as follows

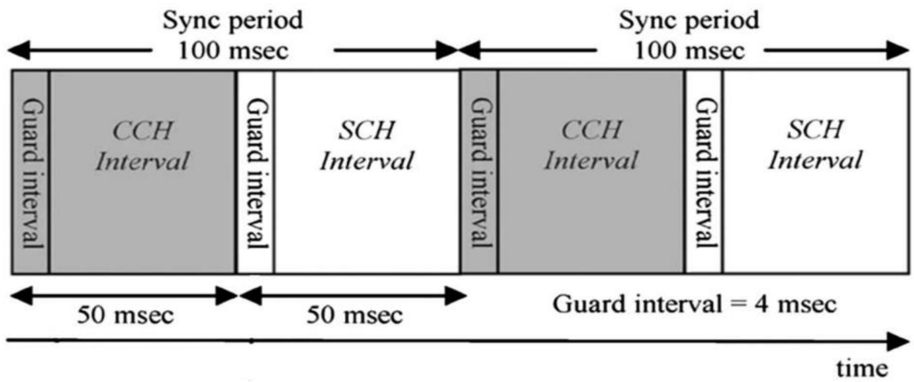

Figure 4-13: We see the time divisions in DSRC. Each synchronization period consists of 1 CCH, 1 SCH and 1 guard interval. While the sync period is generally equal to 100 ms. NetSim allows users to modify the CCH and SCH interval, and in turn the total Sync period.

All devices switch between SCH and CCH and the alternation is based on the time divisions. NetSim allows the user to configure values of CCH interval, SCH interval and Guard interval. The default channels used in NetSim are SCH 171 (5.855 GHz) and CCH 178 (5.890 GHz)

Multiple nodes access the medium using 802.11p protocol. IEEE 802.11p PHY operates in the 5.9 GHz band with a channel bandwidth of 10 MHz 802.11p is an adaptation of the IEEE 802.11a standard used in Wi-Fi systems.

4.3.2 Simulation scenario

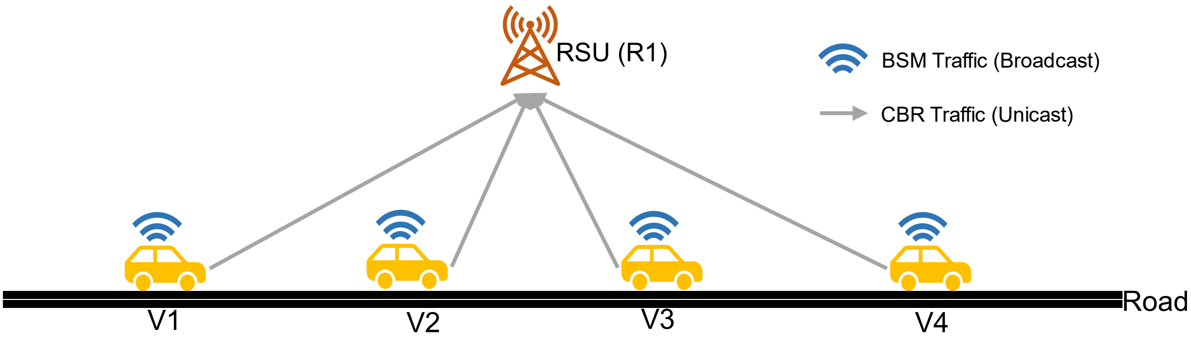

Figure 4-14: Illustration of the VANET scenario under study. The network comprises of 4 vehicles and 1 roadside unit. Each vehicle transmits two applications: (i) a BSM broadcast application that is sent to all other devices (vehicles plus RSU) within range and (ii) a CBR application transmitted to the RSU

The scenario comprises of four vehicles, V1, V2, V3 and V4 communication to the RSU, R1 and to one another. As explained in Figure 4-14 each vehicle sends unicast CBR traffic to the

RSU and broadcast BSM (safety messages) to one another. Recall that per DSRC functioning, BSM is sent on the CCH while CBR is sent on the SCH.

4.3.3 Simulation parameters and results

4.3.4 Part 1: Throughput

Open NetSim and Select **Examples > VANETs > Throughput, delay and collisions with SCH and CCH time division > Throughput** then click on the tile in the middle panel to load the example as shown in below Figure 4-15.

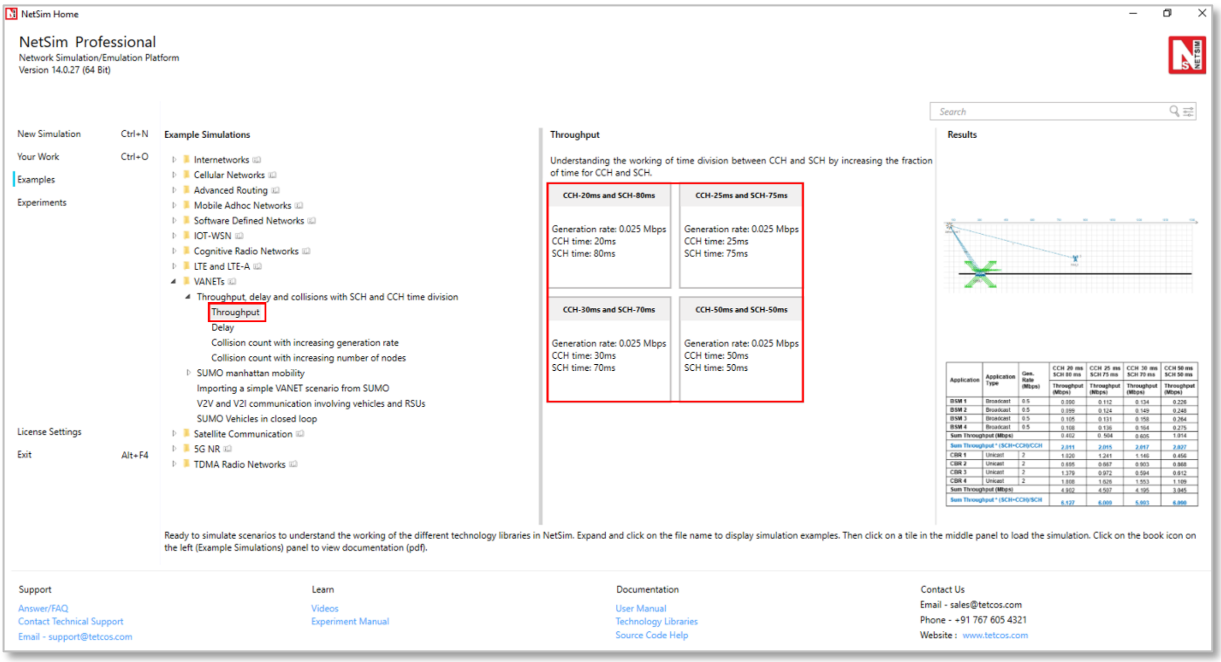

Figure 4-15: List of scenarios for the example of Throughput

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file throughput as shown in below Figure 4-16.

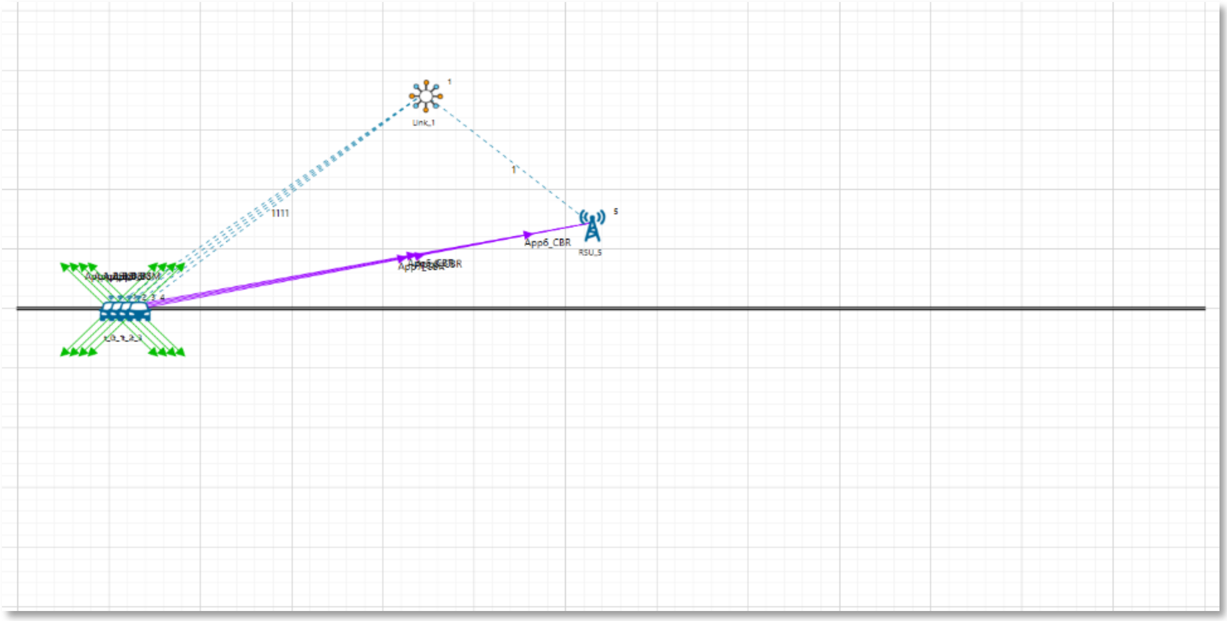

Figure 4-16: Network set up for studying the Throughput

Settings done for the Experiment:

Properties	
Application Properties	
Application Type	BSM (Applications 1-4)
Application Method	BROADCAST
Packet Size	20 (Bytes)
Inter Arrival Time	320 (µs)
Application Type	CBR (Applications 5-8)
Application Method	UNICAST
Packet Size	1460 (Bytes)
Inter Arrival Time	5840 (µs)
Link Properties	
Channel Characteristics	Pathloss Only
Pathloss Model	Log Distance
Pathloss Exponent	2
Data Link Properties	
CCH Time	20 ms, 25 ms, 30 ms, 50 ms (Varying)
SCH Time	80 ms, 75 ms, 70 ms, 50 ms (Varying)
Physical Layer Properties (Vehicle)	
Standard	IEEE802.11p
Transmitter Power	100 mW
Antenna Gain	1 dBi
Antenna Height	1 dBi
Bandwidth	10 MHz

Table 4-3: Application, Link and Physical layer Properties

4.3.4.1 Results

The BSM application is configured with packet size of 20B and inter-packet arrival time of 320 μs, while the CBR application is configured with packet size of 1460B and inter-packet arrival time of 5840 μs.

Application	Application Type	Gen. Rate (Mbps)	CCH 20 ms SCH 80 ms Throughput (Mbps)	CCH 25 ms SCH 75 ms Throughput (Mbps)	CCH 30 ms SCH 70 ms Throughput (Mbps)	CCH 50 ms SCH 50 ms Throughput (Mbps)
BSM 1	Broadcast	0.5	0.096	0.121	0.146	0.240
BSM 2	Broadcast	0.5	0.104	0.129	0.155	0.261
BSM 3	Broadcast	0.5	0.108	0.136	0.162	0.274
BSM 4	Broadcast	0.5	0.113	0.142	0.171	0.286
Sum Throughput (Mbps)			0.421	0.528	0.421	0.528
Sum Throughput * (SCH+CCH)/CCH			**2.105**	**2.111**	**2.105**	**2.111**
CBR 1	Unicast	2	1.838	1.690	1.592	1.085
CBR 2	Unicast	2	1.769	1.722	1.529	1.134
CBR 3	Unicast	2	1.848	1.707	1.601	1.190
CBR 4	Unicast	2	1.861	1.718	1.659	1.143
Sum Throughput (Mbps)			7.316	6.837	7.316	6.837
CASum Throughput * (SCH+CCH)/SCH			**9.145**	**9.116**	**9.145**	**9.116**

Table 4-4: We see that the as the CCH interval increases, BSM application has higher throughput rate. Similarly, as the SCH Interval decreases there is a decrease in throughput rate.

4.3.4.2 Observations

1. BSM in sent on CCH; CBR is sent on SCH. Increasing the fraction of time for CCH increases BSM throughput. Increasing the fraction of time for SCH increases CBR throughput.
2. As expected, Sum throughput divided by SCH fraction is equal for all cases. Similarly, Sum throughput divided by CCH fraction is equal in all cases. This verifies the working of time division between CCH and SCH.

4.3.5 Part 2: Delay

Open NetSim and Select **Examples > VANETs > Throughput, delay and collisions with SCH and CCH time division > Delay** then click on the tile in the middle panel to load the example as shown in below screenshot

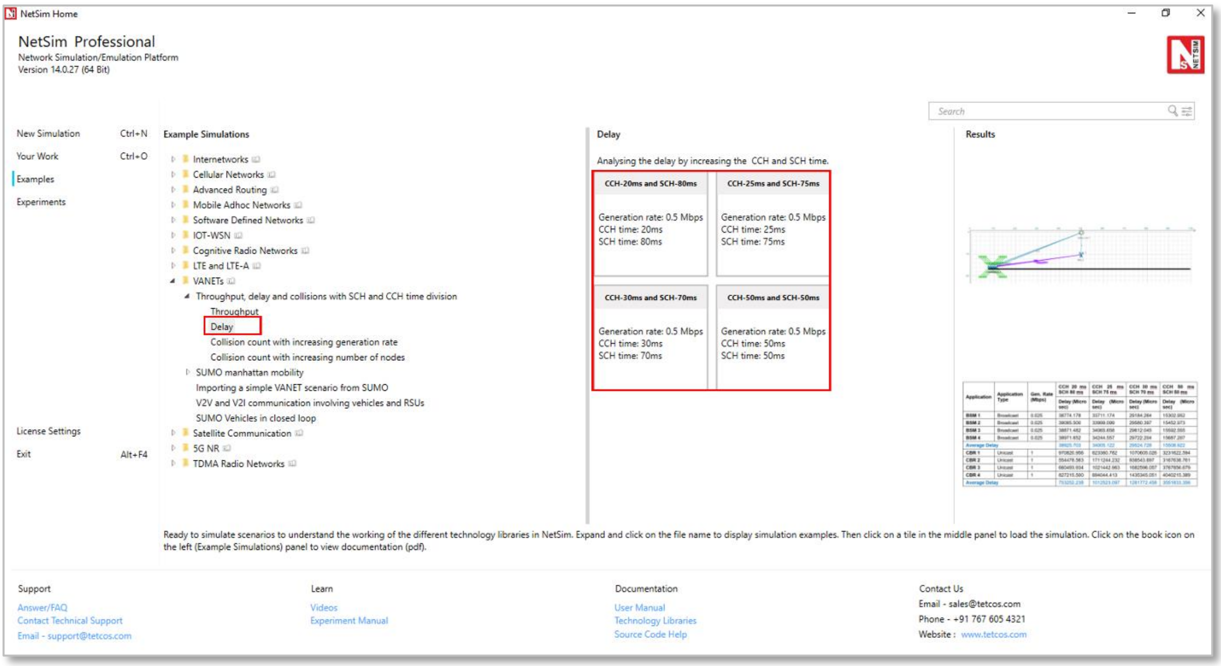

Figure 4-17: List of scenarios for the example of Delay

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file throughput as shown in below Figure 4-28.

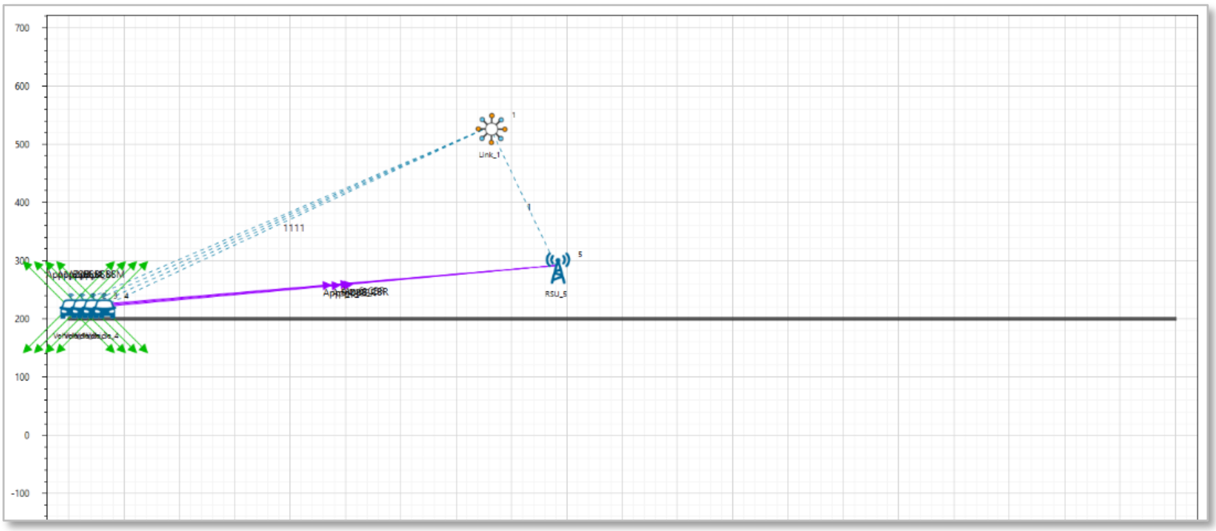

Figure 4-18: Network set up for studying the Delay

Settings done for the Experiment:

Properties	
Application Properties	
Application Type	BSM (Applications 1-4)
Application Method	BROADCAST
Transport Protocol	WSMP
Packet Size	20 (Bytes)
Inter Arrival Time	6400 (µs)
Application Type	CBR (Applications 5-8)

Application Method	UNICAST
Transport Protocol	UDP
Packet Size	1460 (Bytes)
Inter Arrival Time	11680 (µs)
Link Properties	
Channel Characteristics	Pathloss Only
Pathloss Model	Log Distance
Pathloss Exponent	2.5
Data Link Properties	
CCH Time	20 ms, 25 ms, 30 ms, 50 ms (Varying)
SCH Time	80 ms, 75 ms, 70 ms, 50 ms (Varying)
Physical Layer Properties (Vehicle)	
Standard	IEEE802.11p
Transmitter Power	100 mW
Antenna Gain	1 dBi
Antenna Height	1 dBi
Bandwidth	10 MHz

Table 4-5: Application, Link and Physical layer Properties

4.3.5.1 Results

When analyzing delay, we change the generation rate such that it is below the saturation capacity of the network. If this were not so, then queuing delay would blow-up at (and post) saturation.

APPLICATION	Application Type	Gen. Rate (Mbps)	CCH 20 ms SCH 80 ms Delay (Micro sec)	CCH 25 ms SCH 75 ms Delay (Micro sec)	CCH 30 ms SCH 70 ms Delay (Micro sec)	CCH 50 ms SCH 50 ms Delay (Micro sec)
BSM 1	Broadcast	0.025	39157.0172	33603.5655	29297.1917	14963.2294
BSM 2	Broadcast	0.025	38777.6930	34097.4895	29631.2976	15339.1878
BSM 3	Broadcast	0.025	39401.9765	34194.9543	29703.8060	15353.3003
BSM 4	Broadcast	0.025	39305.3040	34398.1667	29990.2463	15501.4813
Average Delay			39160.4977	34073.5440	29655.6354	15289.2997
CBR 1	Unicast	1	8405481.9254	8627465.0801	46870809.2009	47045337.6550
CBR 2	Unicast	1	22775844.4111	24481833.4339	26768319.4297	10179134.9685
CBR 3	Unicast	1	3926050.1513	4110107.3368	8731827.9376	32922483.4220
CBR 4	Unicast	1	45846676.7056	45828534.3322	4052726.1651	4912706.2903
Average Delay			20238513.2984	20761985.0457	216055920.6834	23764915.5840

Table 4-6: We see that as the CCH interval increases, the delay for BSM application decreases. Similarly, as the SCH interval decreases the delay for CBR application increases.

4.3.5.2 Observations

1. CCH Delay has 3 components (a) waiting time where the packet is waiting for the SCH to complete (b) Medium access time and (c) Transmission time
2. The mathematical analysis of delay is complex. It involves two evaluating two difficult components (i) CCH packet waiting time while SCH packet is served and vice versa, and (ii) medium access time. We leave the mathematical analysis to interested researchers, and restrict ourselves to stating that the trends are as expected i.e., increasing CCH time (and reducing SCH time) reduces the CCH delay (and increases SCH delay)

4.3.6 Part 3: Collision count with increasing generation rate

Open NetSim and Select **Examples > VANETs > Throughput, delay and collisions with SCH and CCH time division > Collision count with increasing generation rate** then click on the tile in the middle panel to load the example as shown in below screenshot

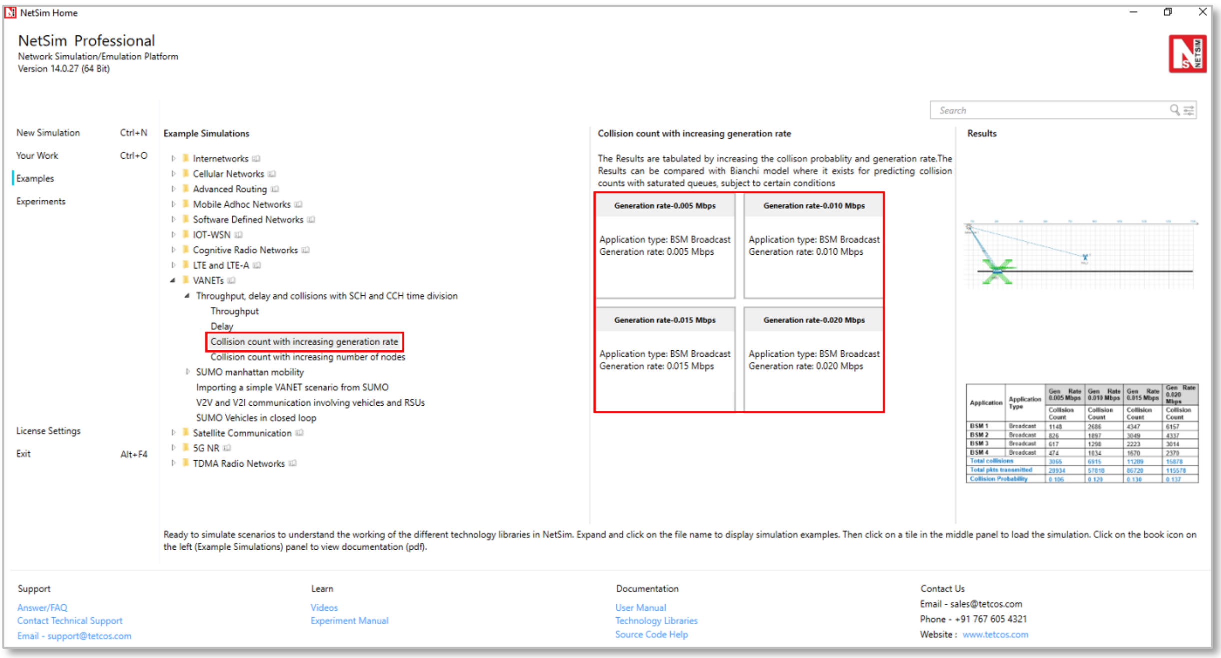

Figure 4-19: List of scenarios for the example of Collision count with increasing generation rate

The following network diagram illustrates what the NetSim Figure 4-20UI displays when you open the example configuration file throughput as shown in below Figure 4-20.

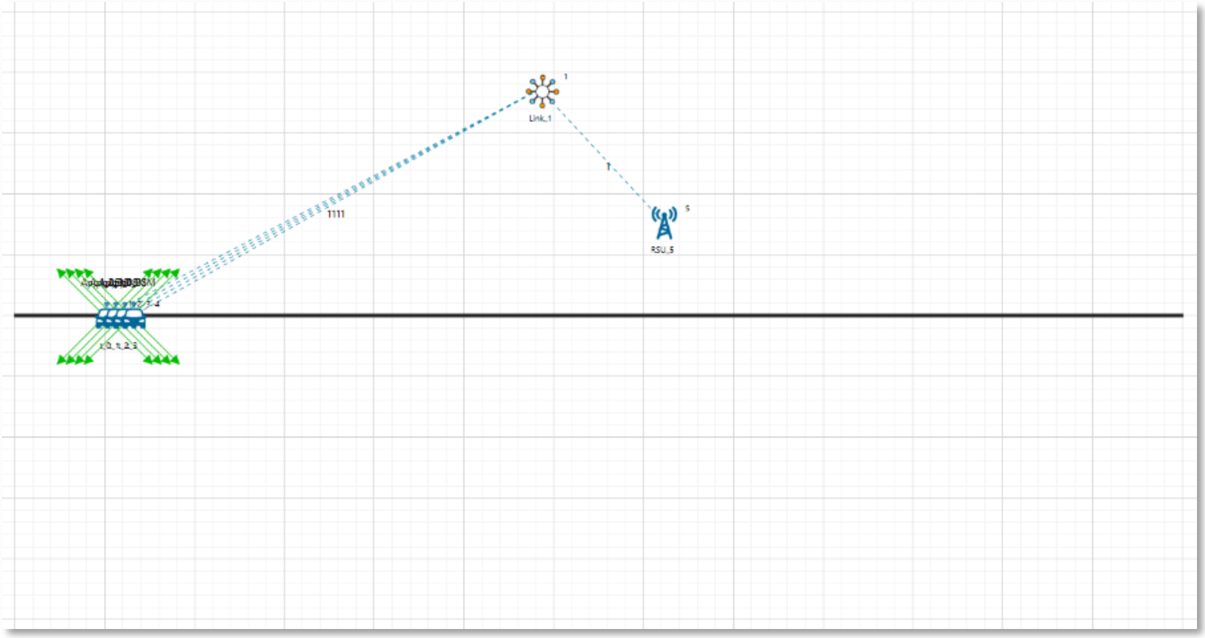

Figure 4-20: Network set up for studying the Collision count with increasing generation rate

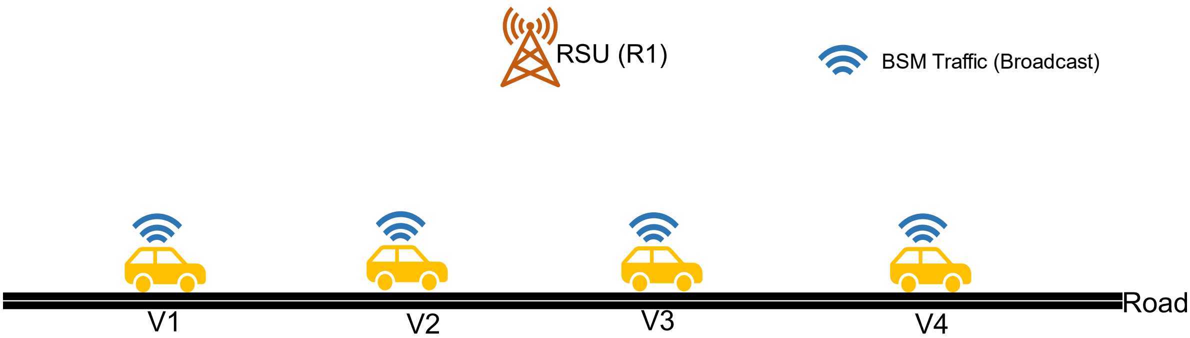

Figure 4-21: The scenario layout remains the same, however we change the application settings. In this example we only have 4 BSM applications. There are no CBR applications.

Settings done for the Experiment:

Properties	
Application Properties	
App Type	BSM
Application Method	BROADCAST
Packet Size	20 (Bytes)
Inter Arrival Time	32000(µs), 16000 (µs), 10666.67(µs), 8000 (µs) (Varying..)
Link Properties	
Channel Characteristics	No Pathloss
Data Link Properties	
CCH Time	20 ms
SCH Time	80 ms
Physical Layer Properties (Vehicle)	
Standard	IEEE802.11p

Transmitter Power	100 mW
Antenna Gain	1 dBi
Antenna Height	1 dBi
Bandwidth	10 MHz

Table 4-7: Application, Link and Physical layer Properties

4.3.6.1 Results

The application generation rates are mentioned in Row 1 (shaded grey).

Application	Application Type	Gen Rate 0.005 Mbps Collision Count	Gen Rate 0.010 Mbps Collision Count	Gen Rate 0.015 Mbps Collision Count	Gen Rate 0.020 Mbps Collision Count
BSM 1	Broadcast	168	468	793	1244
BSM 2	Broadcast	157	453	810	1144
BSM 3	Broadcast	151	365	688	926
BSM 4	Broadcast	44	121	214	301
Total collisions		520	1407	2505	3615
Total pkts transmitted		29932	59820	89724	119584
Collision Probability		0.017	0.024	0.028	0.030

Table 4-8: Comparison of Collision count of BSM applications with changing generation rate

The Collision probability is the ratio between Collision count to total number of packets transmitted

$$Collision\ probability = \frac{Collision\ count}{packets\ transmitted}$$

To find the Collision count of each individual application,

- Click on Packet Trace in the results dashboard window (Please make sure the packet trace is enabled before running the simulation)

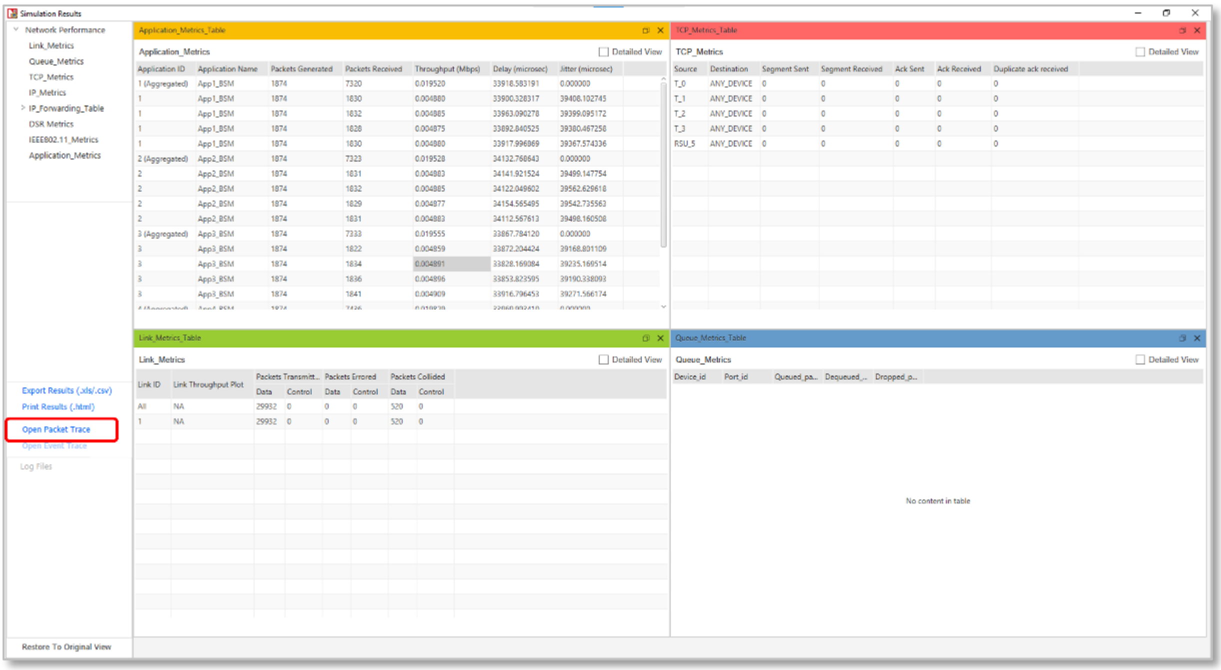

Figure 4-22 : Results dashboard window

- **total no of collision counts,** and **packets transmitted** can be viewed in the link metrics table over results dashboard
- In packet trace, filter the control packet type/ App Name to App1_BSM to find the individual collision count
- Along with that filter the packet status field to collisions to view the collisions of that application (APP1_BSM)

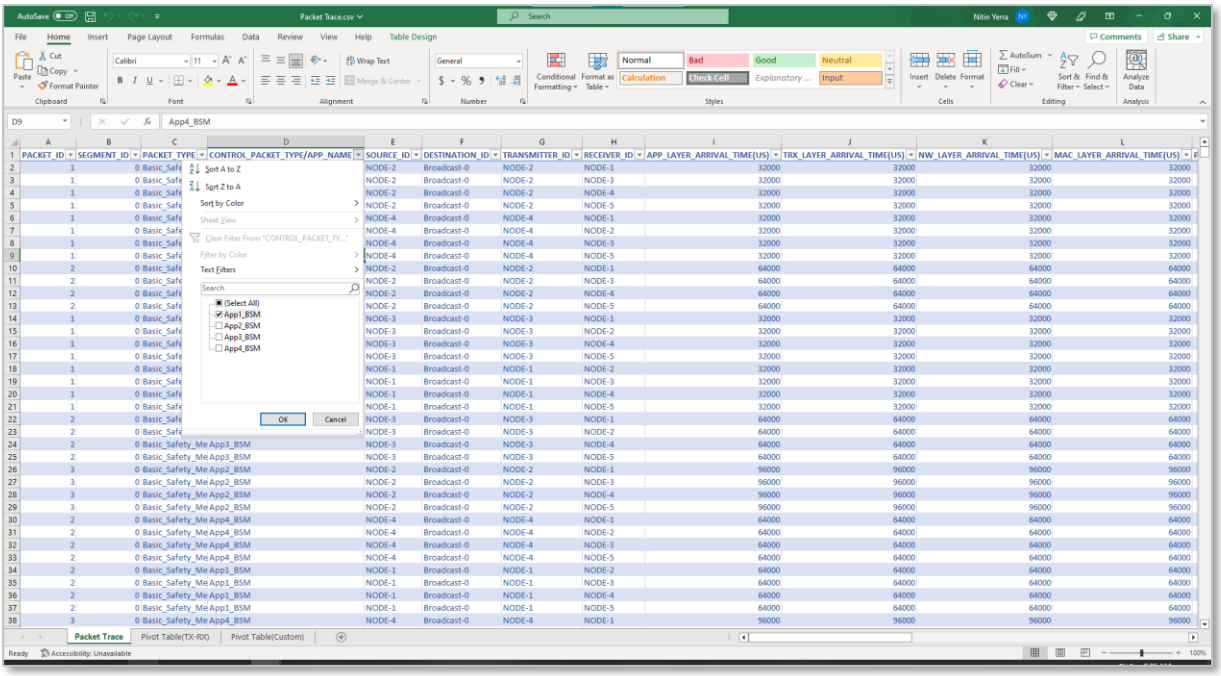

Figure 4-23 : Packet trace which depicts filtering of applications

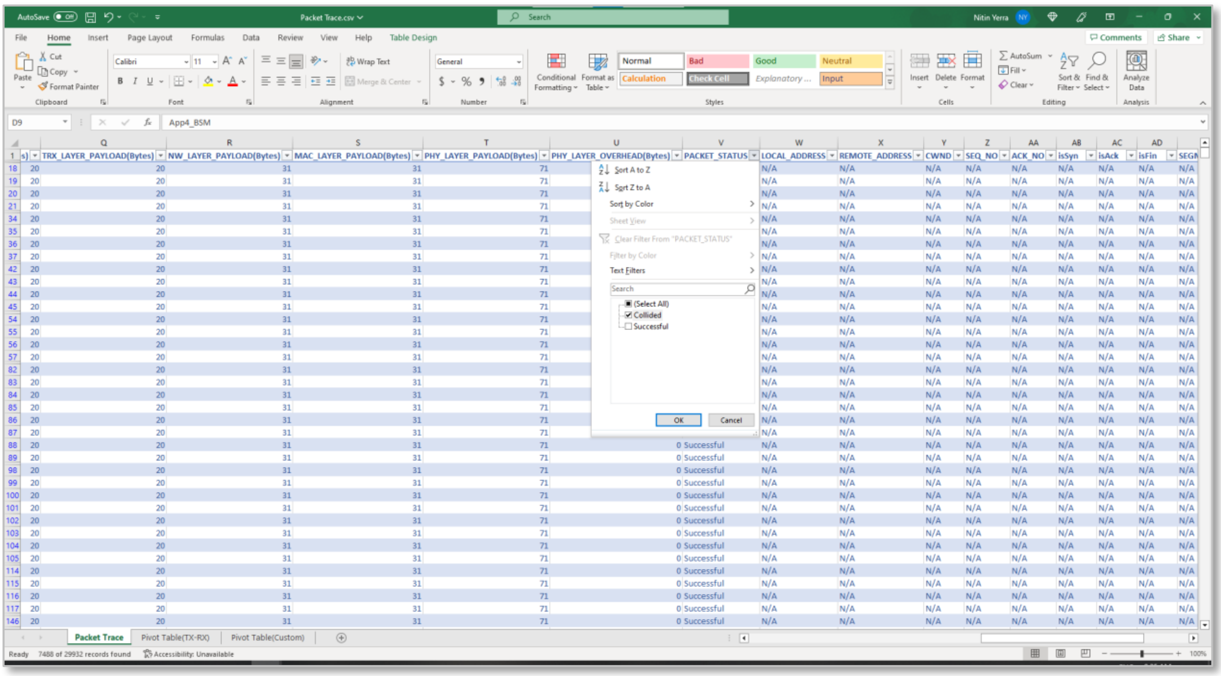

Figure 4-24 : Packet trace that depicts filtering of packet status of each application

- After applying the filters, the total collision count of APP1_BSM applications can be viewed

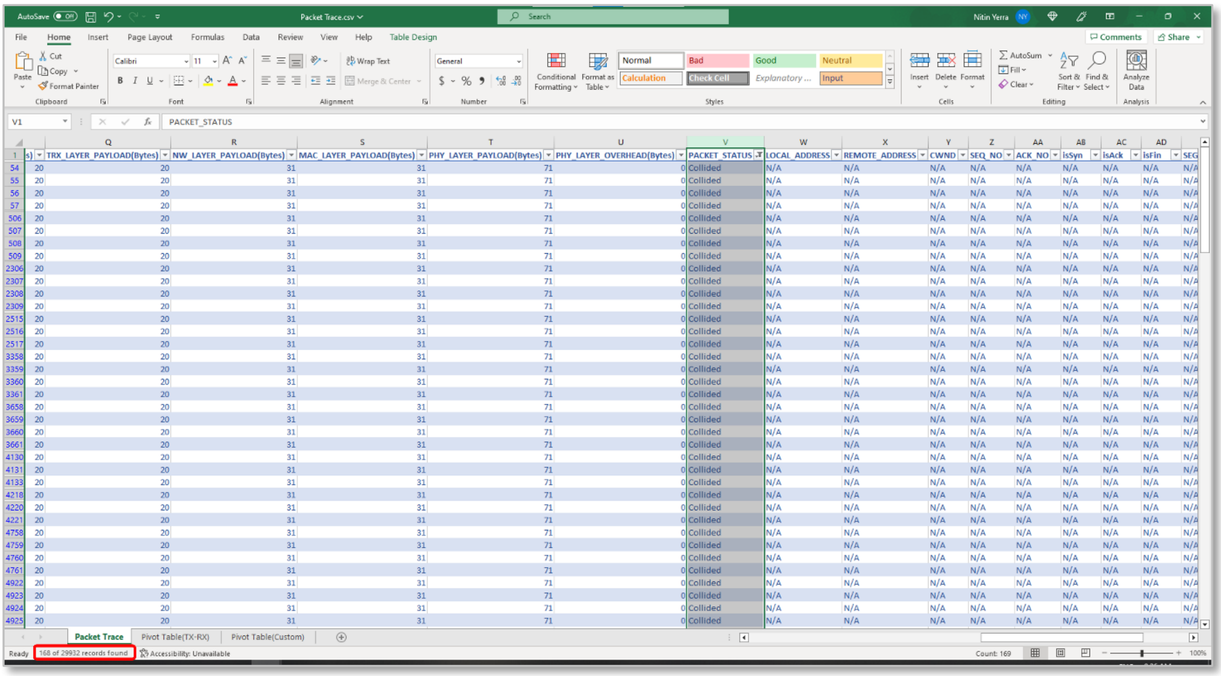

Figure 4-25 : packet trace

- Same process can be done for all the remaining applications of the network.

4.3.6.2 Observations

- Saturation throughput is about 0.25 Mbps per app or 1 Mbps total. Note the generation rate is below the saturation capacity of the network

- We see collision probability increases as generation rate increases
- To the best of our knowledge the mathematical modelling of collisions with non-saturated queues is an open problem. The Bianchi model exists for predicting collision counts with saturated queues, subject to certain conditions.

4.3.7 Part 4: Collisions count with increasing number of nodes

Open NetSim and Select **Examples > VANETs > Throughput, delay and collisions with SCH and CCH time division > Collisions count with increasing number of nodes** then click on the tile in the middle panel to load the example as shown in below screenshot

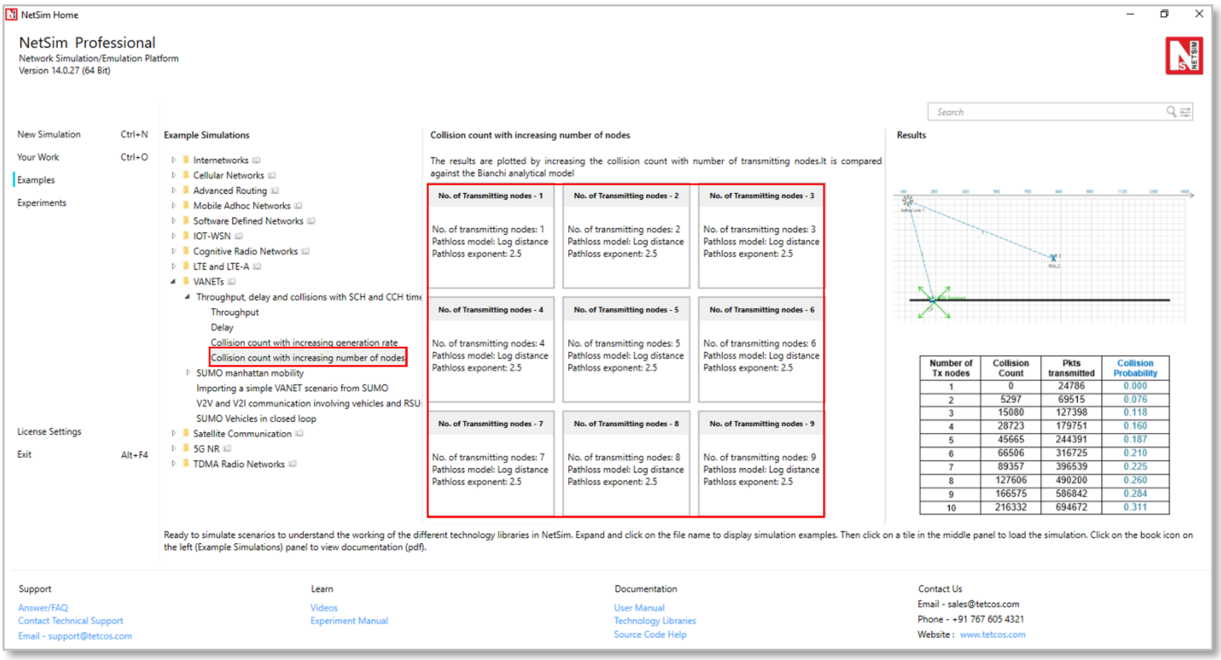

Figure 4-26: List of scenarios for the example of Collisions count with increasing number of nodes

The following network diagram illustrates what the NetSim Figure 4-20UI displays when you open the example configuration file throughput as shown in below Figure 4-27.

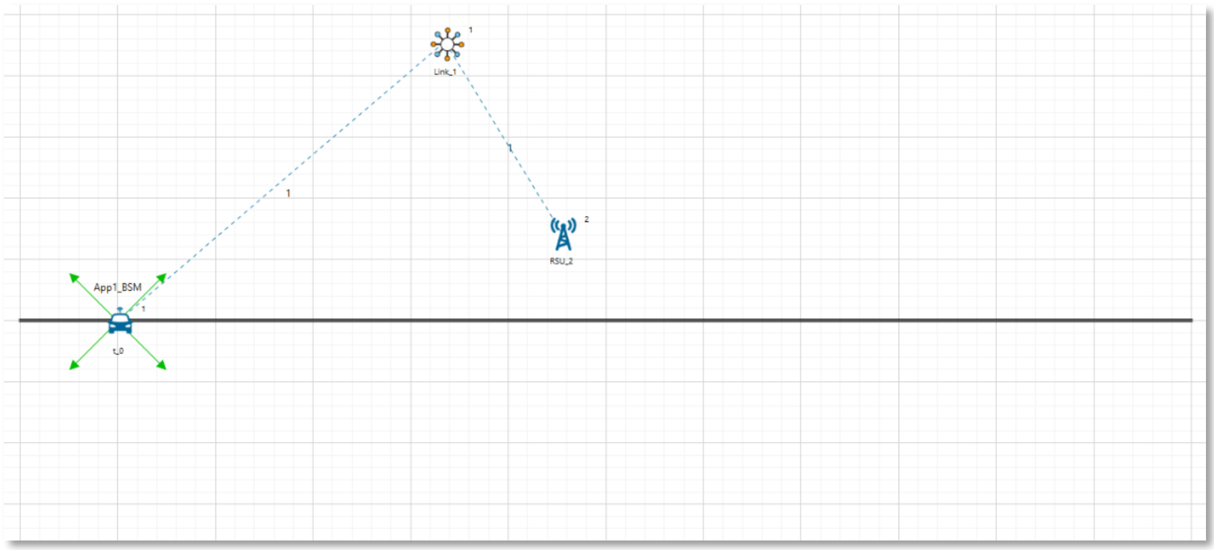

Figure 4-27: Network set up for studying the Collisions count with increasing number of nodes

This scenario has 10 vehicles in a line on a road. The vehicles transmit power $P_t = 20\ dBm$, Carrier sense threshold $CS_{th} = -85\ dBm$, and we assumed log distance pathloss with $\eta = 2.5$. The received power between nodes with maximum separation, $d = 100$, is

$$P_r = 20 - 47.88 - 10 \times 2.5 \times \log(100) = -77.88\ dBm$$

Since $P_r > CS_{th}$ all nodes can *hear* one another which means that they are all within CS Range.

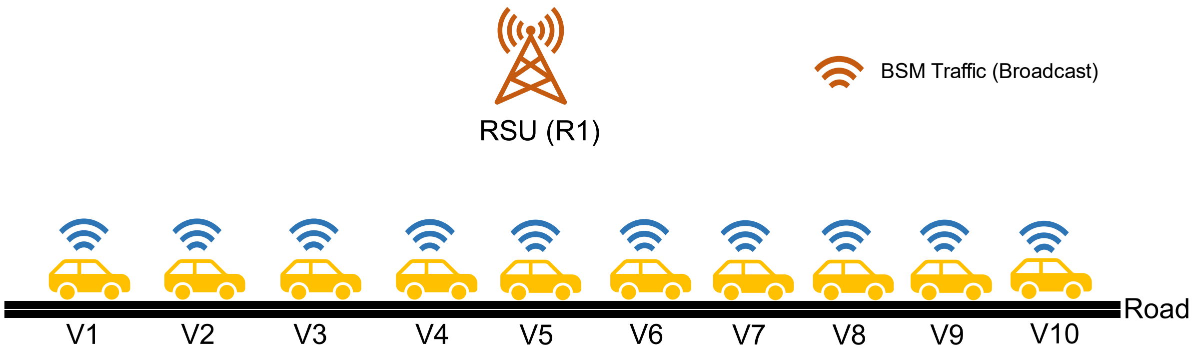

Figure 4-28: Illustration of the VANET scenario under study. The network comprises of 10 vehicles and 1 roadside unit. Each vehicle transmits one application (i) a BSM broadcast application that is sent to all other devices (vehicles plus RSU) within range. In this study we are increasing the Tx nodes from 1-10

Settings done for the Experiment:

Properties	
Application Properties	
App Type	BSM
Application Method	BROADCAST
Packet Size	20 (Bytes)
Inter Arrival Time	320(μs)
Link Properties	
Channel Characteristics	No Pathloss

Data Link Properties	
CCH Time	20 ms
SCH Time	80 ms
Physical Layer Properties (Vehicle)	
Standard	IEEE802.11p
Transmitter Power	100 mW
Antenna Gain	1 dBi
Antenna Height	1 dBi
Bandwidth	10 MHz

Table 4-9: Application, Link and Physical layer Properties

4.3.7.1 Results

Number of Tx nodes	Collision Count	Pkts transmitted	Collision Probability
1	0	31261	0.000
2	0	75726	0.000
3	1731	127041	0.014
4	7142	187016	0.038
5	16724	252210	0.066
6	32045	325308	0.099
7	53946	405678	0.133
8	86203	499376	0.173
9	126970	596448	0.213
10	176435	704510	0.250

Table 4-10: Collision probability comparison with change in number of transmitting nodes

The total no of collision count and packets transmitted can be viewed in link metrics window of results dashboard.

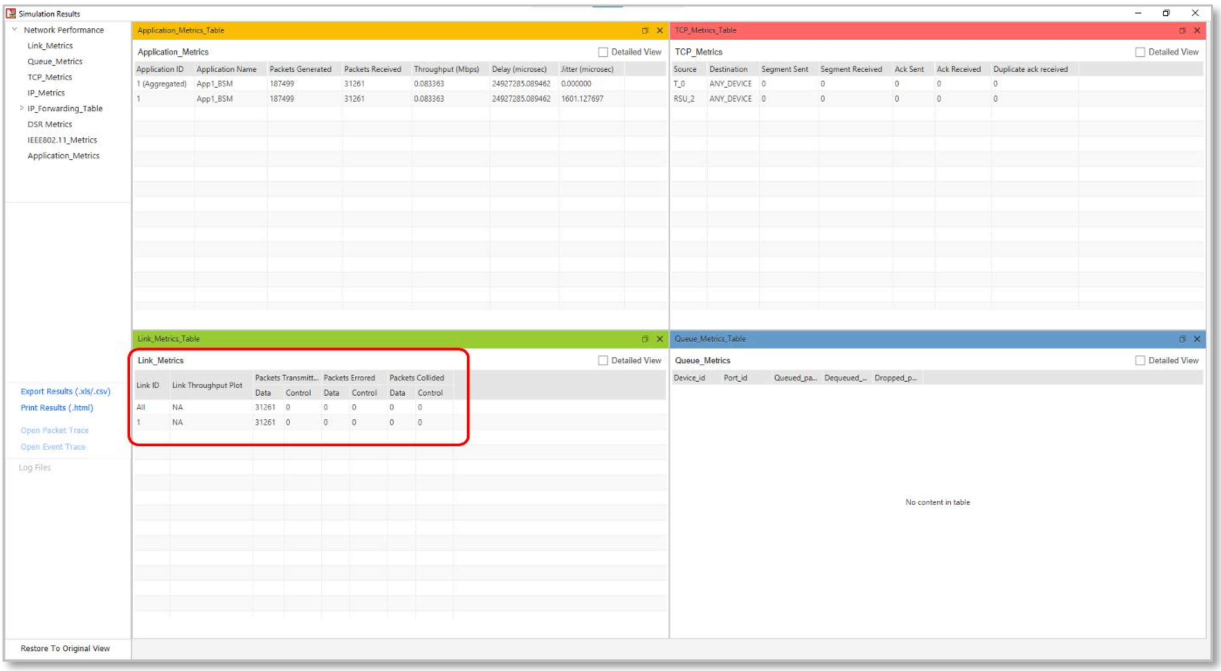

Figure 4-29 : Results Dashboard window

The Collision probability is the ratio between Collision count to total number of packets transmitted

$$Collision\ probability = \frac{Collision\ count}{packets\ transmitted}$$

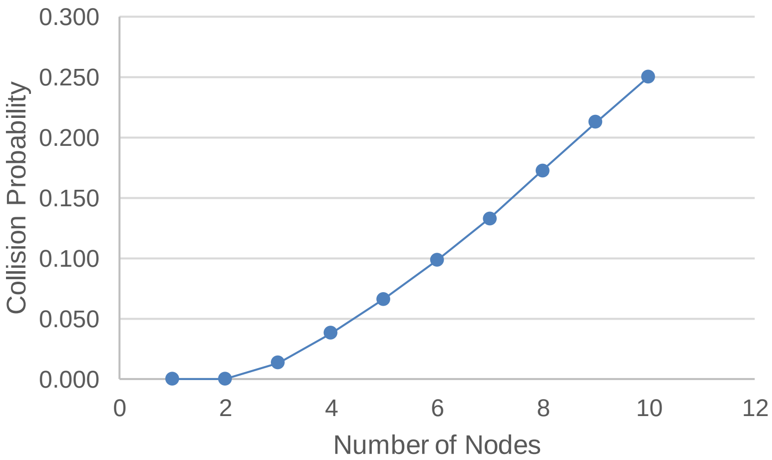

Figure 4-30: Collision probability vs. number of transmitting nodes

4.3.7.2 Observations

- We see collision count increasing with number of transmitting nodes
- This can be compared against the Bianchi analytical model

4.4 SUMO Manhattan Mobility with Single and Multi-hop Communication

Introduction

The Manhattan mobility in SUMO features a grid topology as shown below. It is composed of a number of horizontal and vertical streets. Each street has two lanes for each direction (North and South direction for vertical streets, East and West for horizontal streets). The mobile node is allowed to move along the grid of horizontal and vertical streets. At an intersection of a horizontal and a vertical street, the mobile node can turn left, right or go straight with certain probability.

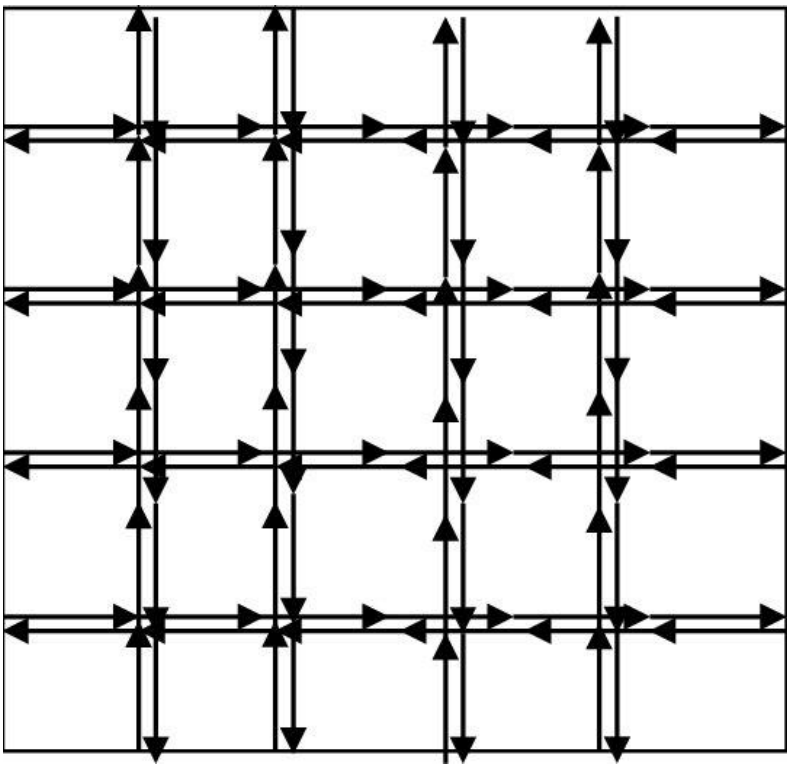

Figure 4-31: Manhattan mobility in SUMO features a grid topology

Case 1: Manhattan mobility Single-hop RSU to vehicles

Objective

To create, using SUMO, a Manhattan Road network in which vehicles drive randomly, and to have a Roadside unit (RSU) which sends safety messages continuously to vehicles. The network performance is analyzed for different environments each having different RF channel characteristics.

Procedure

Open NetSim and Select **Examples > VANETs > SUMO Manhattan mobility > Single hop communication** then click on the tile in the middle panel to load the example as shown in below screenshot

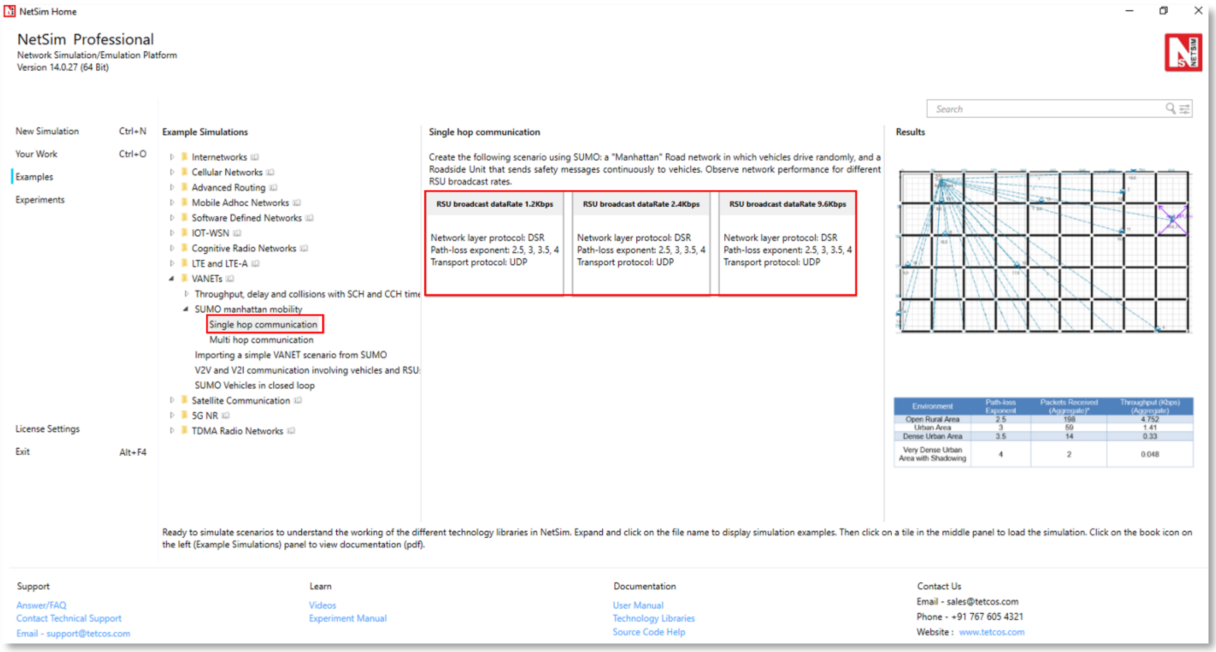

Figure 4-32: List of scenarios for the example of Single hop communication

The NetSim UI would display as shown below.

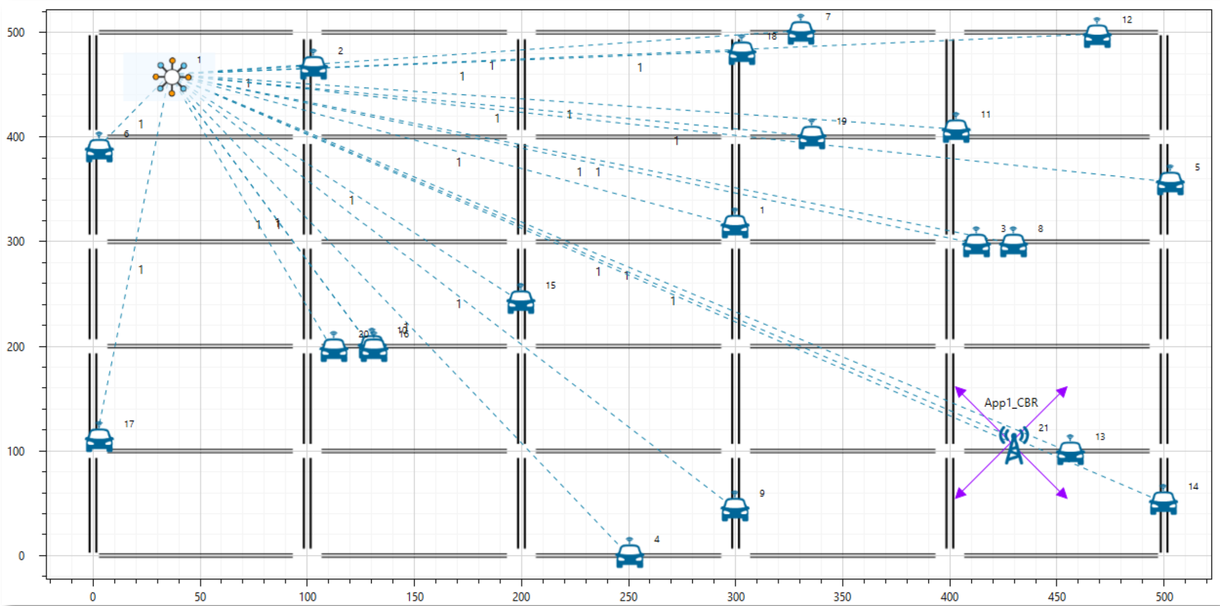

Figure 4-33: Network set up for studying the Single hop communication

Settings done for this sample experiment.

1. Applications set as CBR (Broadcast application)

Application Method	Application Type	Application Name	Source ID	Destination ID	Packet Size (Bytes)	Inter-Arrival Time (µs)
Broadcast	CBR	APP_1_CBR_Broadcast	21	Broadcast to all 20 vehicles	300	2,000,000

Table 4-11: CBR Applications Settings

2. Transport protocol set as UDP in application Configuring window.
3. Adhoc link/Wireless link properties were set as follows:

Channel characteristics	Pathloss Model	Pathloss Exponent
Pathloss Only	Log Distance	2.5

Table 4-12: Wireless link properties

4. Co-ordinates of RSU are set as X = 834.62, and Y = 133.85.
5. Set transmitter power to 40mW under INTERFACE_1(Wireless) > Physical layer properties of Vehicles and RSU.
6. Plots and packet trace are enabled and run simulation..
7. Increase the pathloss exponent (in the order 2.5, 3, 3.5, 4) and note down the aggregate throughput and packets received for different application generation rates.
8. After running the simulation, Packet Trace can be used to visualize packet flow along with packet information and Mobility log can be used to record vehicle mobility. Time varying throughput plot can be opened from the Results window.

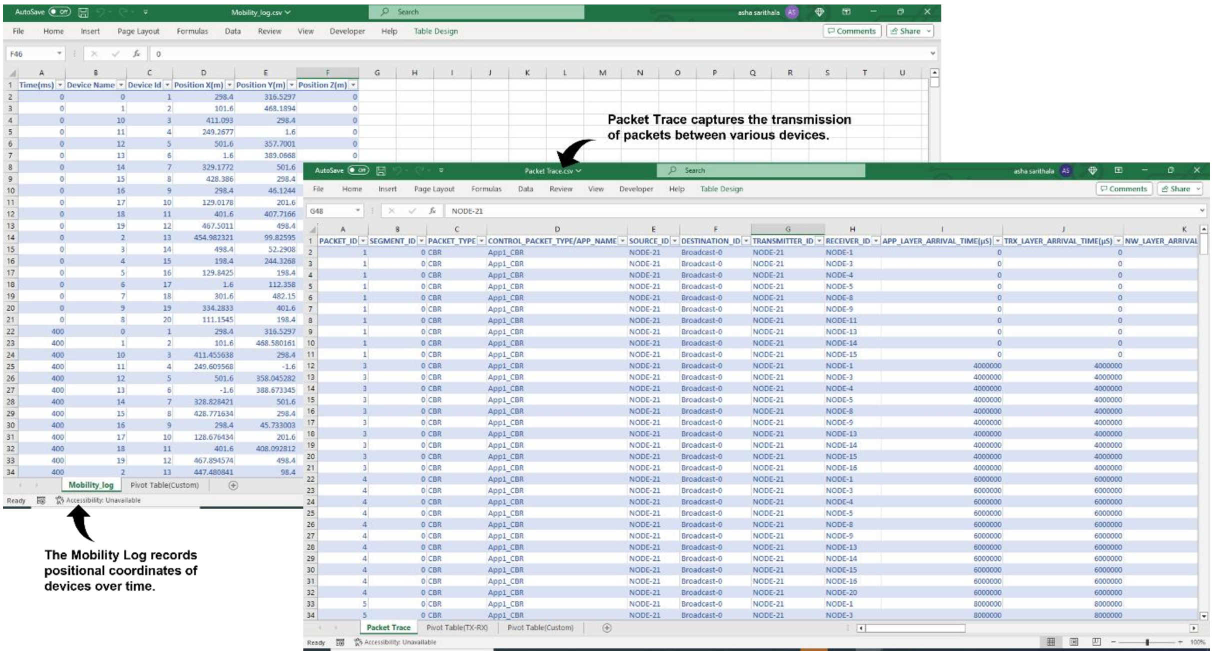

Figure 4-34: Packet Trace and Mobility log window

9. In SUMO GUI, you can see that vehicles choose random directions when they reach a junction in the Manhattan grid network.

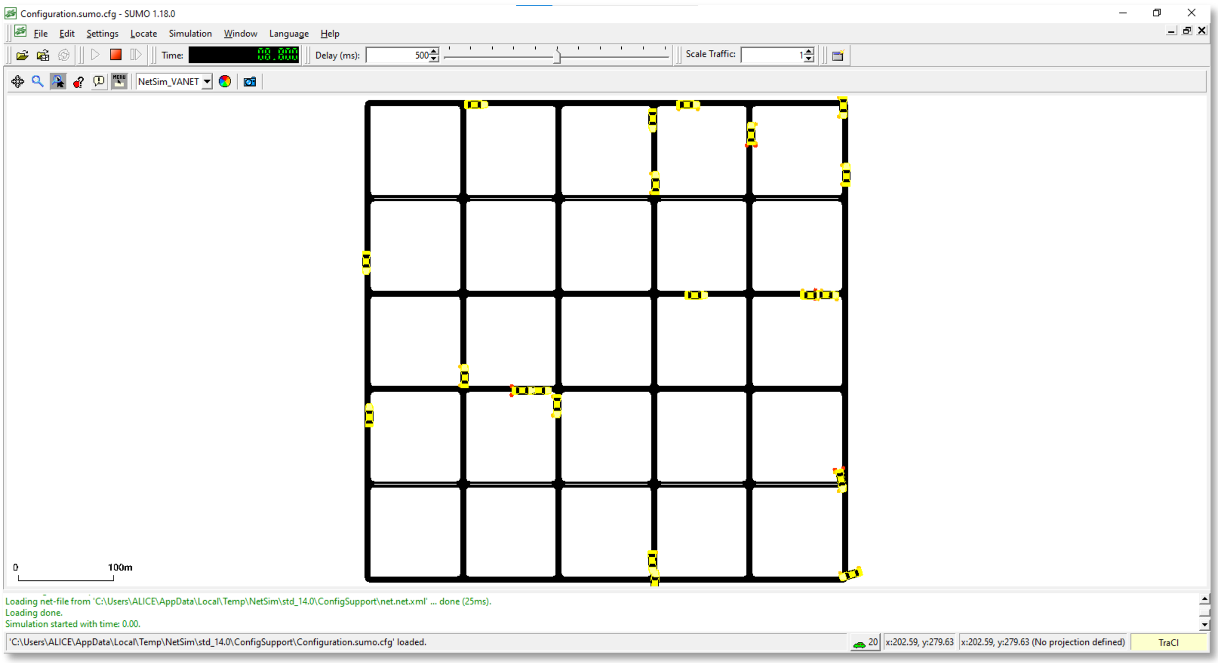

Figure 4-35: Animation Window for Sumo

Results and Observations

For sample **RSU Broadcast Data Rate = 1.2 Kbps** (Packet size = 300 bytes, IAT = 2,000,000µs. This means packets of size 300 Bytes are sent every 2 seconds).

Environment	Path-loss Exponent	Packets Received (Aggregate)*	Throughput (Kbps) (Aggregate)
Open Rural Area	2.5	601	14.424
Urban Area	3	214	5.136
Dense Urban Area	3.5	54	1.296
Very Dense Urban Area with Shadowing	4	25	0.600

Table 4-13: Results Comparison for RSU Broadcast Data Rate = 1.2 Kbps

***** Aggregate is the sum of the packet/throughputs obtained by all applications.

For sample **RSU Broadcast Data Rate = 2.4 Kbps** (Packet size =300 Bytes, IAT = 1,000,000µs or 1 seconds. This means packets of size 300 Bytes are sent every second)

Environment	Path-loss Exponent	Packets Received (Aggregate)	Throughput (Kbps) (Aggregate)
Open Rural Area	2.5	1201	28.824
Urban Area	3	429	10.296
Dense Urban Area	3.5	108	2.592
Very Dense Urban Area with Shadowing	4	51	1.224

Table 4-14: Results Comparison for RSU Broadcast Data Rate = 2.4 Kbps

For sample RSU Broadcast Data Rate = 9.6 Kbps (Packet size =300Bytes, IAT =250,000µs or 0.25 seconds. This means four packets of size 300 Bytes are sent every second)

Environment	Path-loss Exponent	Packets Received (Aggregate)	Throughput (Kbps) (Aggregate)
Open Rural Area	2.5	4843	116.232
Urban Area	3	1718	41.232
Dense Urban Area	3.5	430	10.32
Very Dense Urban Area with Shadowing	4	201	4.824

Table 4-15: Results Comparison for RSU Broadcast Data Rate = 9.6 Kbps

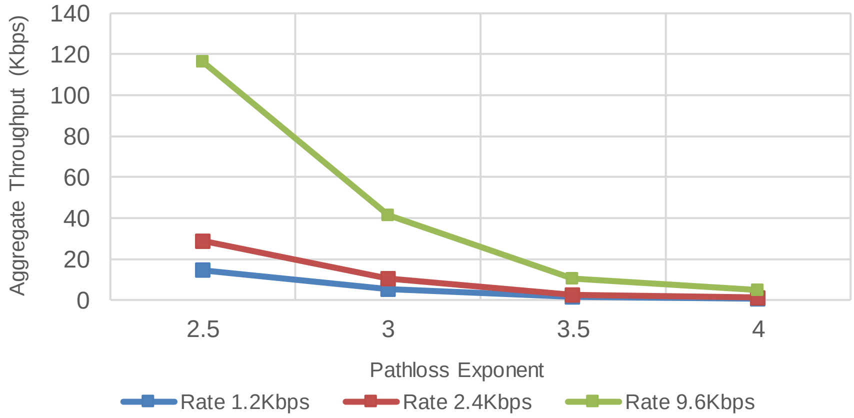

Figure 4-36: Plot of Throughput vs. Pathloss Exponent for different RSU broadcast for different DR (Data Rates)

Case 2: Manhattan mobility Multi-hop Vehicles to RSU

Objective

To create, using SUMO, a Manhattan Road network in which vehicles drive randomly, and to have a Roadside unit (RSU) to which vehicles continuously send unicast traffic via multi-hop (hopping via other vehicles if the RSU is beyond communication range). The network performance is analyzed for different vehicle counts.

Procedure

Open NetSim and Select **Examples > VANETs > SUMO Manhattan mobility > Multi hop communication** then click on the tile in the middle panel to load the example

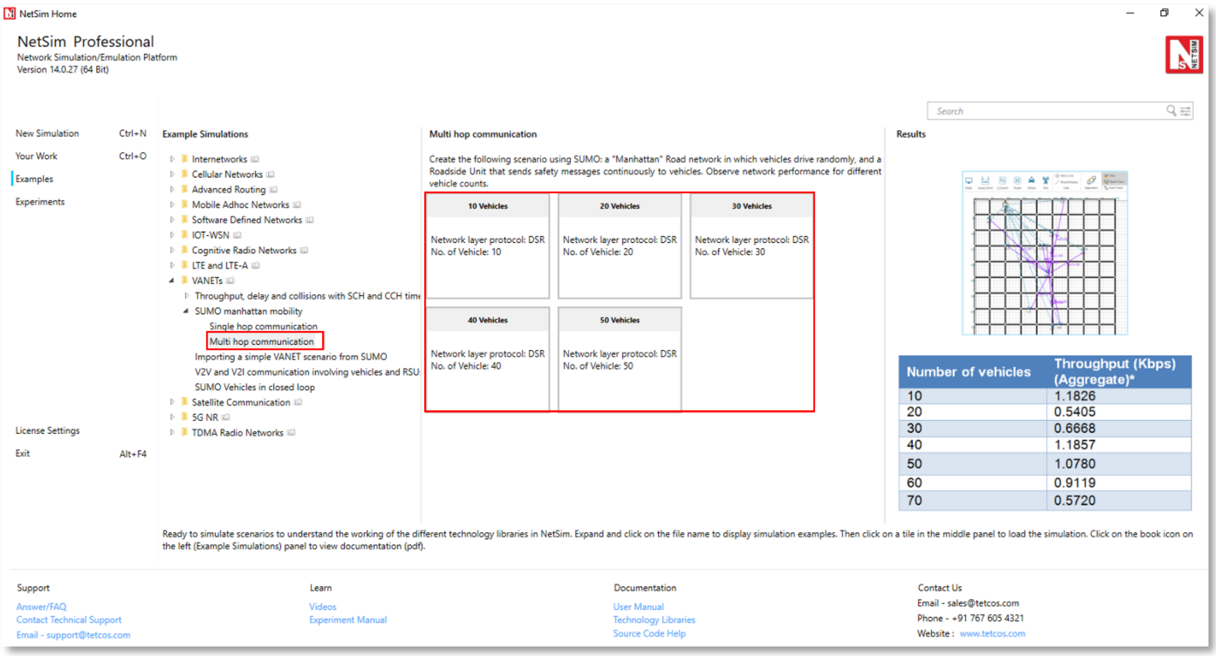

Figure 4-37: List of scenarios for the example of Multi hop communication.

The NetSim UI would display as shown below.

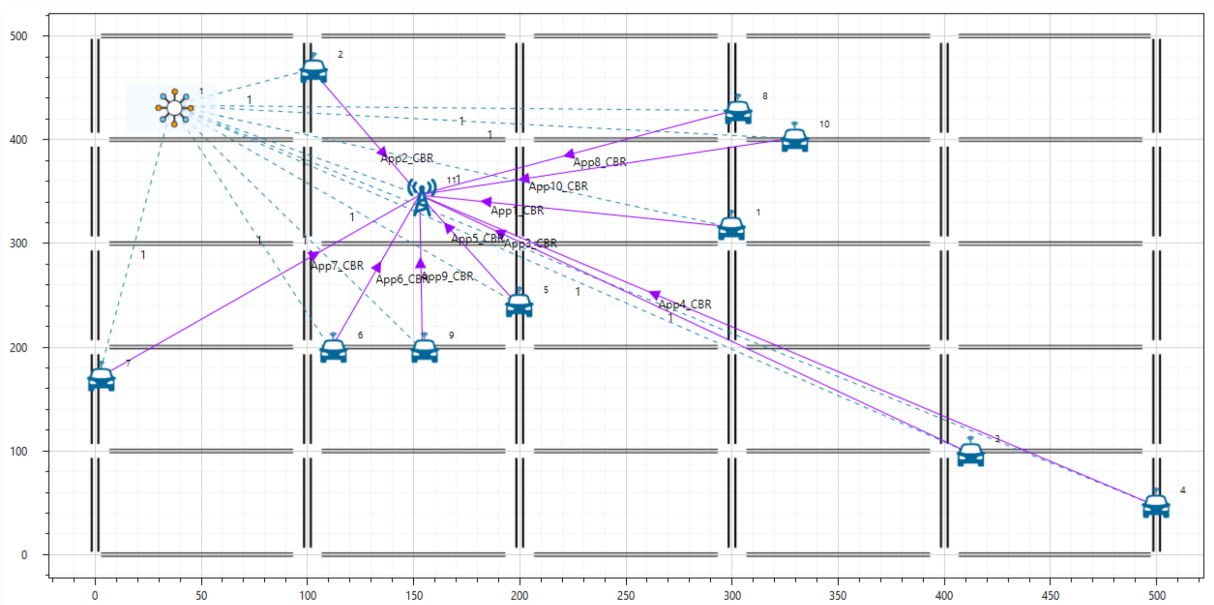

Figure 4-38: Network set up for studying the Multi hop communication

Settings done for this sample experiment.

1. Applications set as CBR.

Application Method	Application Type	Source_Id	Destination_Id	Packet size (Bytes)	Inter-Arrival Time (µs)
Unicast	CBR	(All vehicles)	RSU	1460	20,000

Table 4-16: CBR Applications settings

2. In Vehicle General Properties, under SUMO file manhattan.sumo.cfg file was selected from the Docs folder of NetSim Install Directory < C:\Program Files\NetSim Standard\Docs\Sample_Configuration\VANET\SUMO-Manhattan-mobility-Single-hop-and-Multi-hop\Multi-hop-communication>

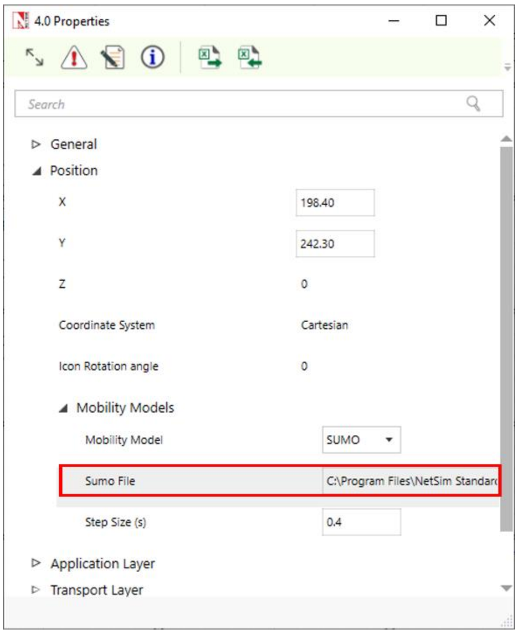

Figure 4-39: General Properties window

3. Transport protocol set as UDP in application Configuration window.
4. Adhoc link/ Wireless link properties set as NO PATHLOSS.
5. Co-ordinates of RSU are set as X = 450, and Y = 450
6. Network layer routing protocol is set as DSR.
7. Set transmitter power to 40mW under INTERFACE_1(Wireless) > Physical layer properties of Vehicles and RSU.
8. Plots are enabled and run the simulation.
9. Increase the number of vehicles in the order 10, 20, 30 etc. and note down the aggregate throughput.

Result:

Number of vehicles	Throughput (Kbps) (Aggregate)*
10	3233.96
20	3466.51
30	3432.87
40	3046.73
50	3086.67

Table 4-17: Results Comparison

*Aggregate is the sum of the packet/throughputs obtained by all applications.

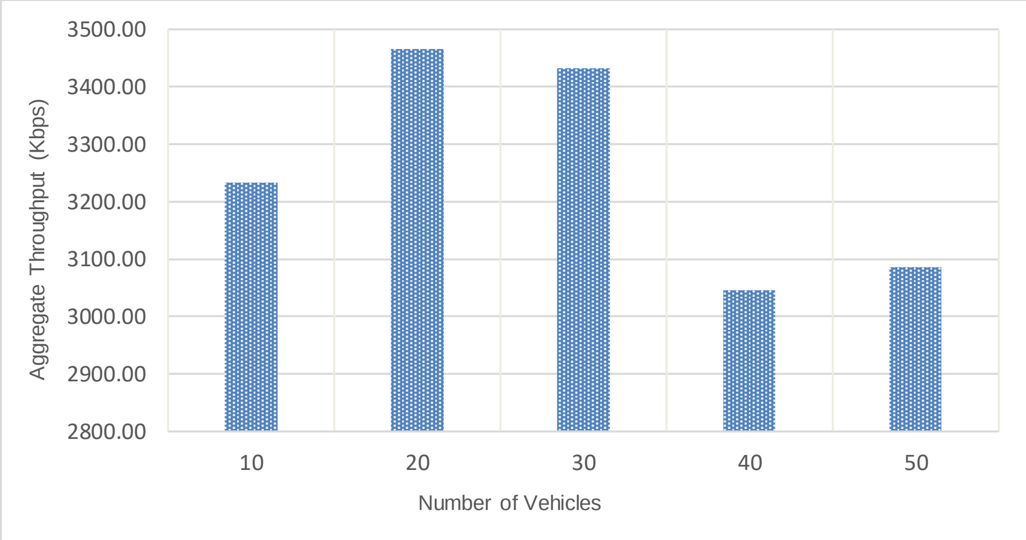

Figure 4-40: Aggregate Throughput vs. Number of Vehicles

4.5 SUMO Interfacing with vehicles moving in a closed loop

Open NetSim and Select **Examples > VANETs > SUMO Vehicles in closed loop** then click on the tile in the middle panel to load the example as shown in below Figure 4-41.

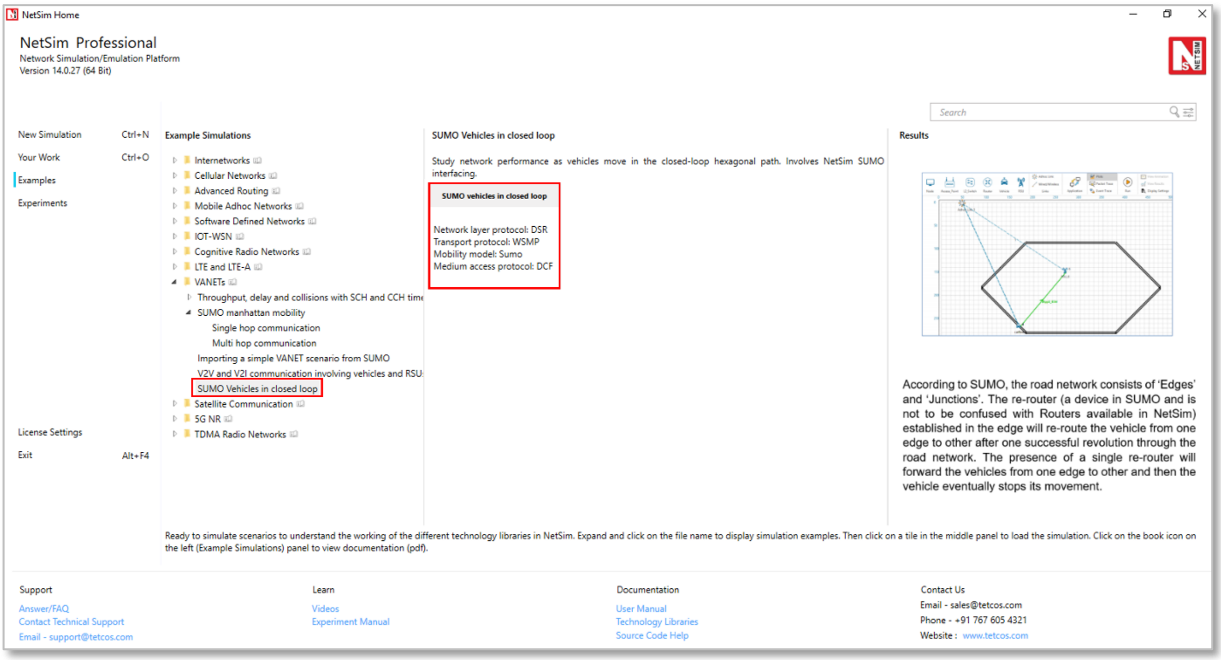

Figure 4-41: List of scenarios for the example of SUMO Vehicles in closed loop

The NetSim UI would display as shown below.

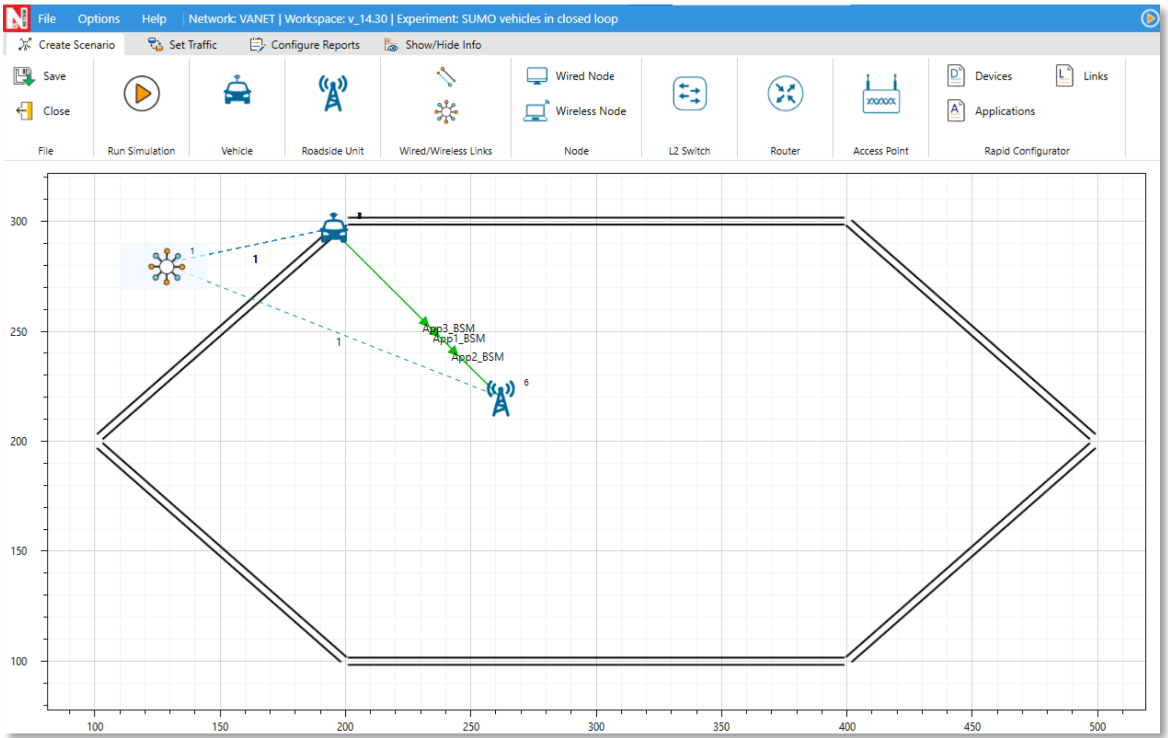

Figure 4-42: Network set up for studying the SUMO Vehicles in closed loop

Settings done for this sample experiment:

1. Applications set as BSM (Basic_Safety_Message)

APP_ID	Source ID	Destination ID	Packet Size (Bytes)	Inter-Arrival Time (µs)
APP_1_BSM	1	6 (RSU)	100	2,000,000
APP_2_BSM	2	6 (RSU)	100	2,000,000
APP_3_BSM	3	6 (RSU)	100	2,000,000

Table 4-18: CBR Applications settings

2. Transport protocol set as WSMP for all applications in Application window.
3. In Vehicle Position Properties, under SUMO file Configuration.sumo.cfg file was selected from the Docs folder of NetSim Install Directory < C:\Program Files\NetSim Standard\Docs\Sample_Configuration\VANET\SUMO-Vehicles-moving-in-closed-loop >

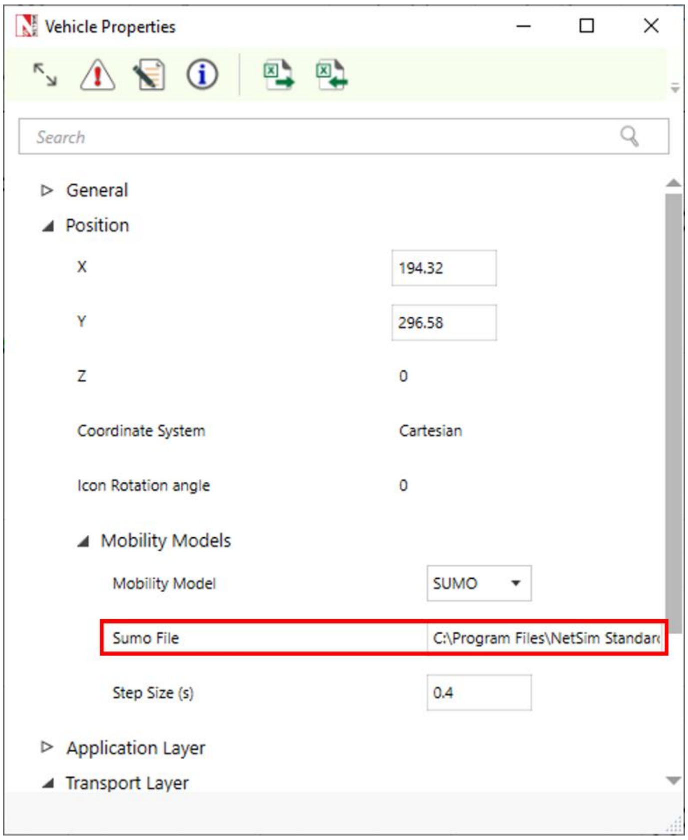

Figure 4-43: General Properties window

4. Adhock link/Wireless link properties were set as NO PATHLOSS.
5. Co-ordinates of RSU are set as X = 278.31 and Y = 153.48
6. Medium access protocol set as DCF in all vehicles and RSU.
7. Set transmitter power to 40mW under INTERFACE_1(Wireless) > Physical layer properties of Vehicles and RSU.
8. Enable Plot and Run simulation.
9. After running the simulation, Packet Trace can be used to visualize packet flow along with packet information and mobility log can be used to record vehicle mobility. Time varying throughput plot can be opened from the Results window.

Result:

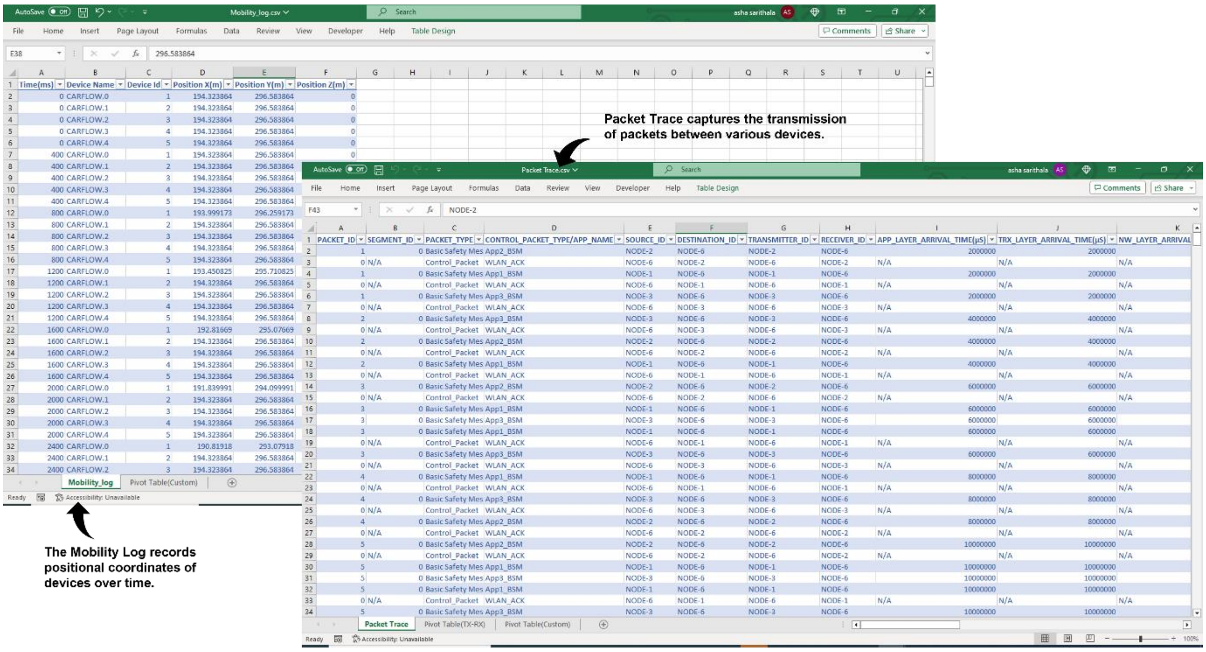

Figure 4-44: Packet Trace and Mobility log window.

same can be observed in SUMO as well

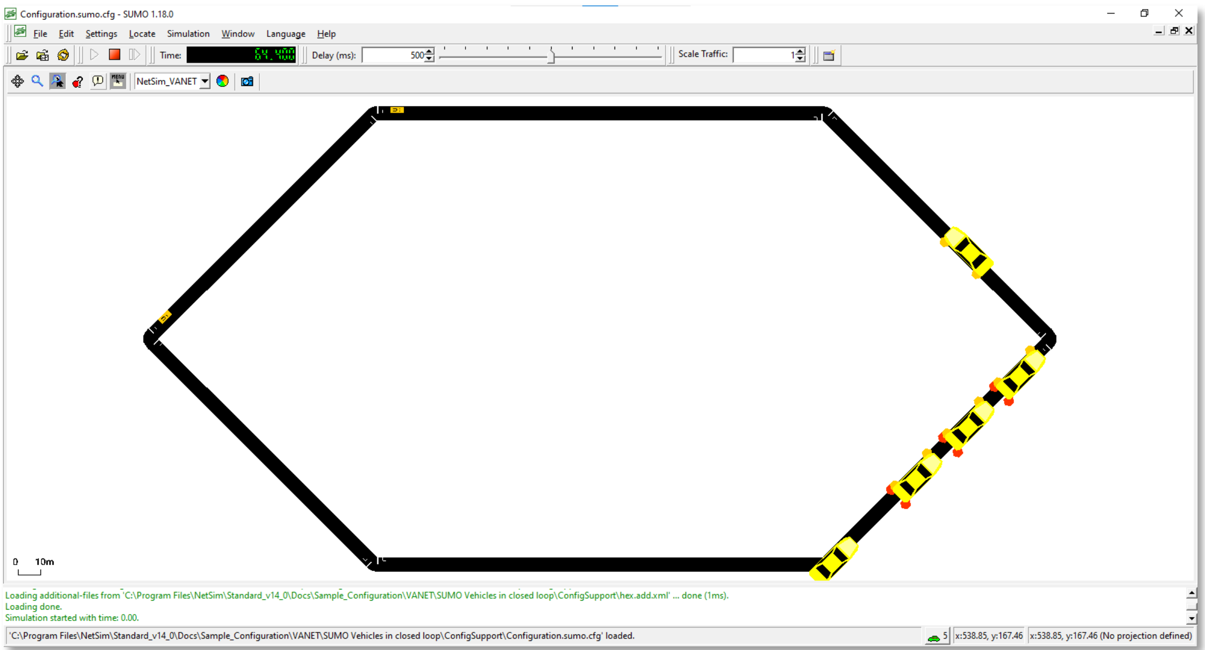

Figure 4-45: Animation window for Sumo

According to SUMO, the road network consists of 'Edges' and 'Junctions'. The re-router (a device in SUMO and is not to be confused with Routers available in NetSim) established in the edge will re-route the vehicle from one edge to other after one successful revolution through the road network. The presence of a single re-router will forward the vehicles from one edge to other and then the vehicle eventually stops its movement. Hence, two re-routers have been established in two edges which re-route the vehicle from one edge to other. The above road network consists of six edges in which re-routers are established in the starting and ending edges, which re-routes the vehicles present in the network from starting edge to the finishing edge after one complete revolution through the road or path. As a result, the vehicles will move through the closed loop continuously, until the end time configured in the configuration file.

In the animation window, we can observe that the vehicles will start from a point in one of the edges, moves through other five edges and finally reach back the point where it started. At this point, the re-router will direct the vehicles to the next edge. This cycle will continue till the end time configured.

The RSU configured in the network will allow V2I communication. Per the application configuration a 100 bytes packet is transmitted from vehicle to RSU every 2 seconds. This can also be observed in the packet trace.

5 Reference Documents

- IEEE 802.11p 2010. Wireless Access for Vehicular Environments
- IEEE1609: Standards for Wireless Access in Vehicular Environment (WAVE)

6 Latest FAQs

Up to date FAQs on NetSim's VANETs library is available at
https://tetcos.freshdesk.com/support/solutions/folders/14000118424

7 References

[1] C. Bhat, "Evaluation of Routing Protocols for Vehicular Ad hoc Networks (VANETs) in Connected Transportation Systems," D-STOP, University of Texas at Austin, Austin, 2018.