

NetSim[®]

Accelerate Network R & D

Vehicular Adhoc Networks (VANETs)

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 13.2 (V), Jun 2022, 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	Introduction	5
2	Simulation GUI.....	7
2.1	Create Scenario	7
2.2	Devices specific to NetSim VANETs Library	8
2.3	Set Node, Link and Application Properties.....	9
2.4	Enable Packet Trace, Event Trace & Plots (Optional).....	11
2.5	Run Simulation	12
3	Model Features	14
3.1	Implementation of the 802.11p in NetSim	14
3.2	DSRC Channels: CCH and SCH	14
3.3	BSM Application	15
3.4	NetSim – SUMO interfacing.....	16
3.5	How to create a VANET using SUMO and simulate with NetSim.....	16
3.5.1	Using SUMO NetEdit utility and randomtrips.py to configure road traffic models	16
3.5.2	Creating your own network in SUMO manually	24
3.6	How to include Roadside Units (RSU's) in a VANET network?	26
4	Featured Examples.....	29
4.1	Importing a simple VANET scenario from SUMO	29
4.2	V2V and V2I communication involving Vehicles and RSUs	31
4.3	Throughput, delay and collisions with SCH and CCH time division.....	34
4.3.1	Background.....	34
4.3.2	Simulation scenario.....	35
4.3.3	Simulation parameters and results	35
4.3.4	Part 1: Throughput	35
4.3.5	Part 2: Delay	36
4.3.6	Part 3: Collision count with increasing generation rate	37
4.3.7	Part 4: Collisions count with increasing number of nodes.....	37
4.4	SUMO Manhattan Mobility with Single and Multi-hop Communication	39
4.5	SUMO Interfacing with vehicles moving in a closed loop	47
5	Reference Documents.....	52

6	Latest FAQs	52
7	References.....	52

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 remote 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.

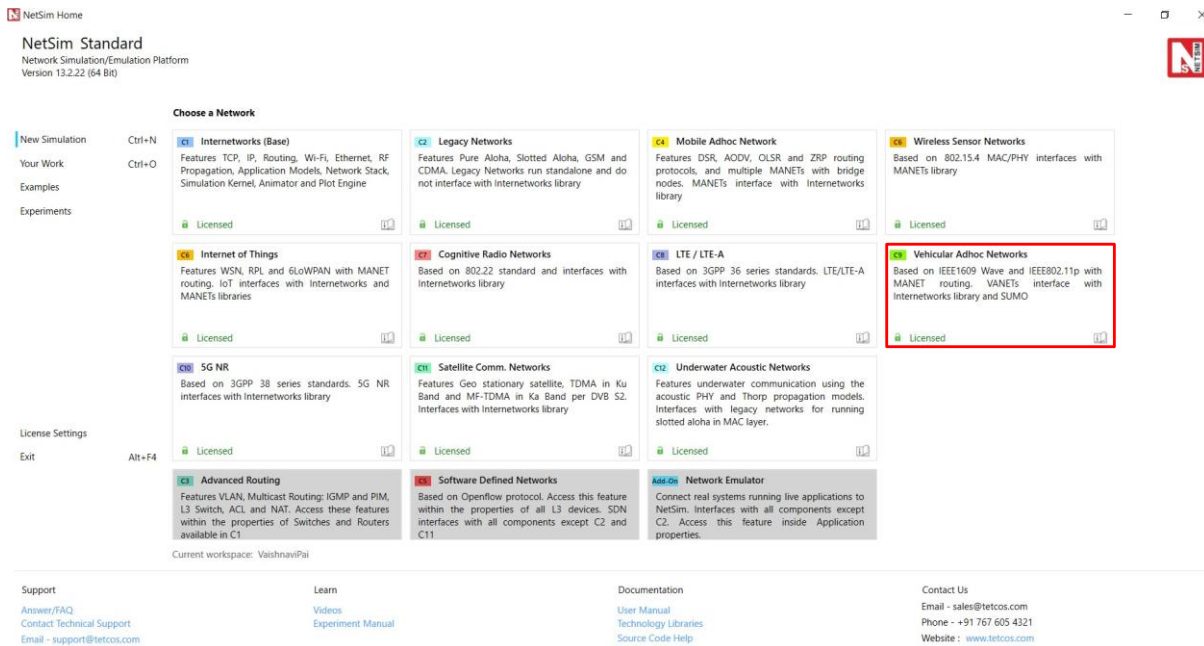


Figure 2-1: NetSim Home Screen

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

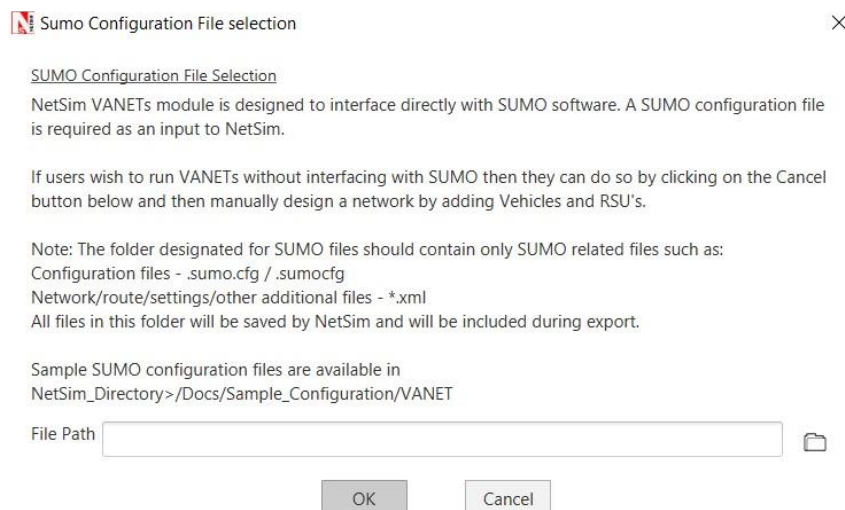


Figure 2-2: Sumo Configuration File path

- 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:** 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.

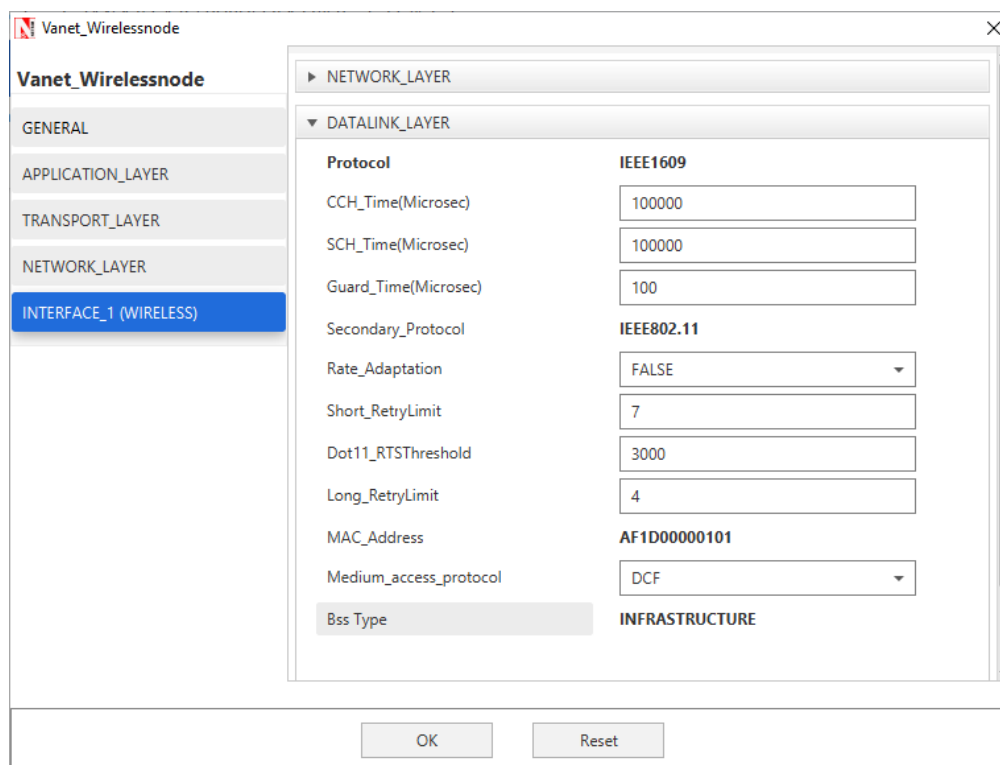


Figure 2-4: Vanet > Datalink layer Properties Window

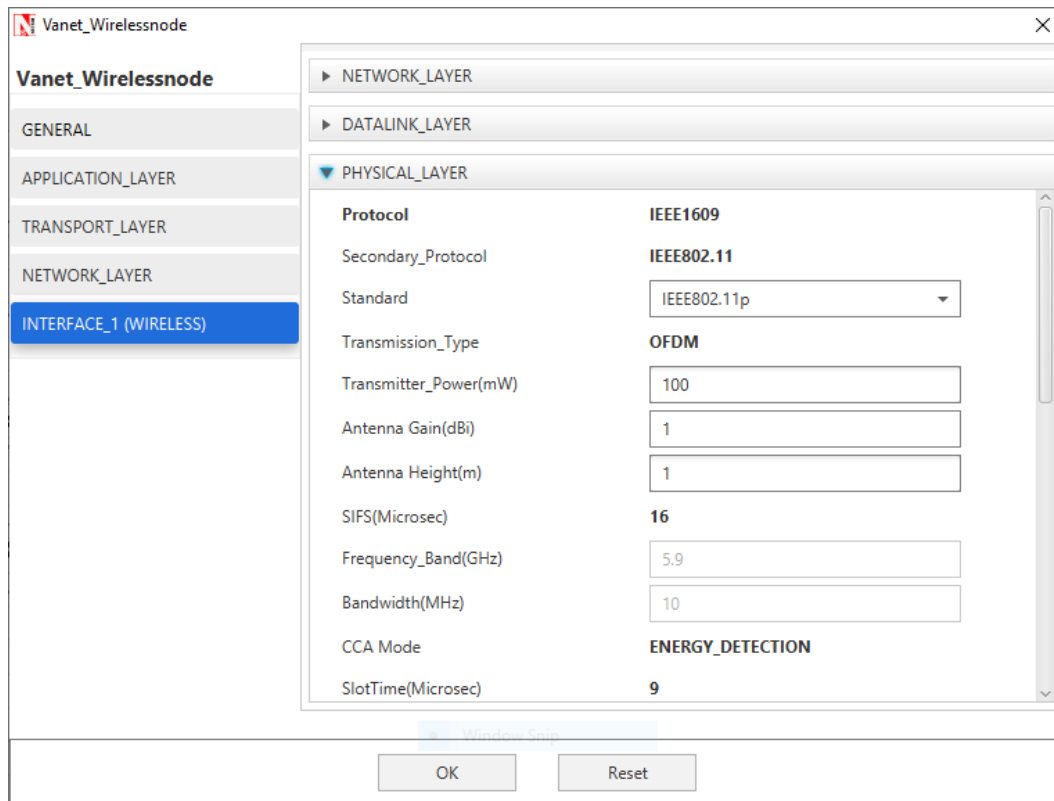


Figure 2-5: Vanet > Physical layer Properties Window

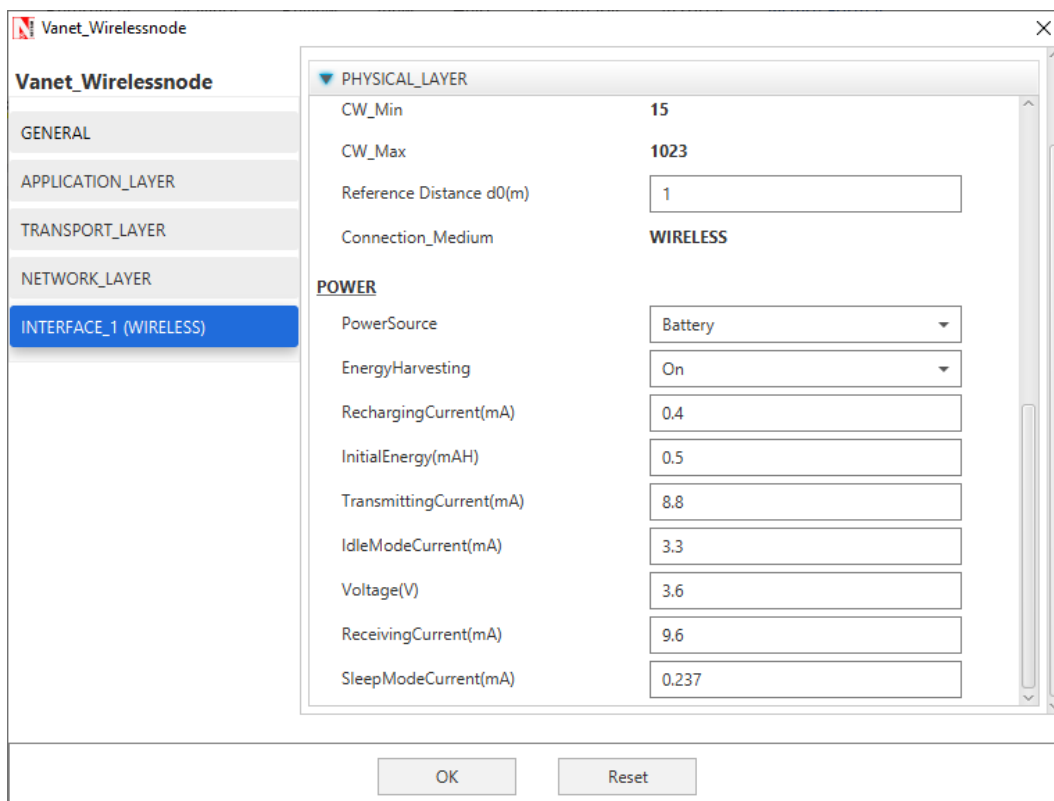


Figure 2-6: Vanet > Physical layer Properties Window > Battery Model

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

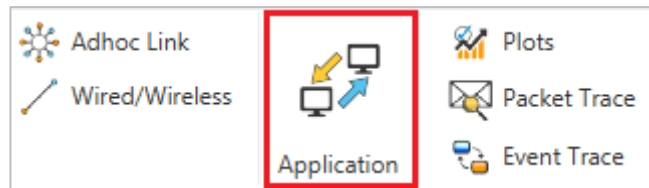


Figure 2-7: Application icon present on top ribbon

- Set the values according to requirement and click OK.

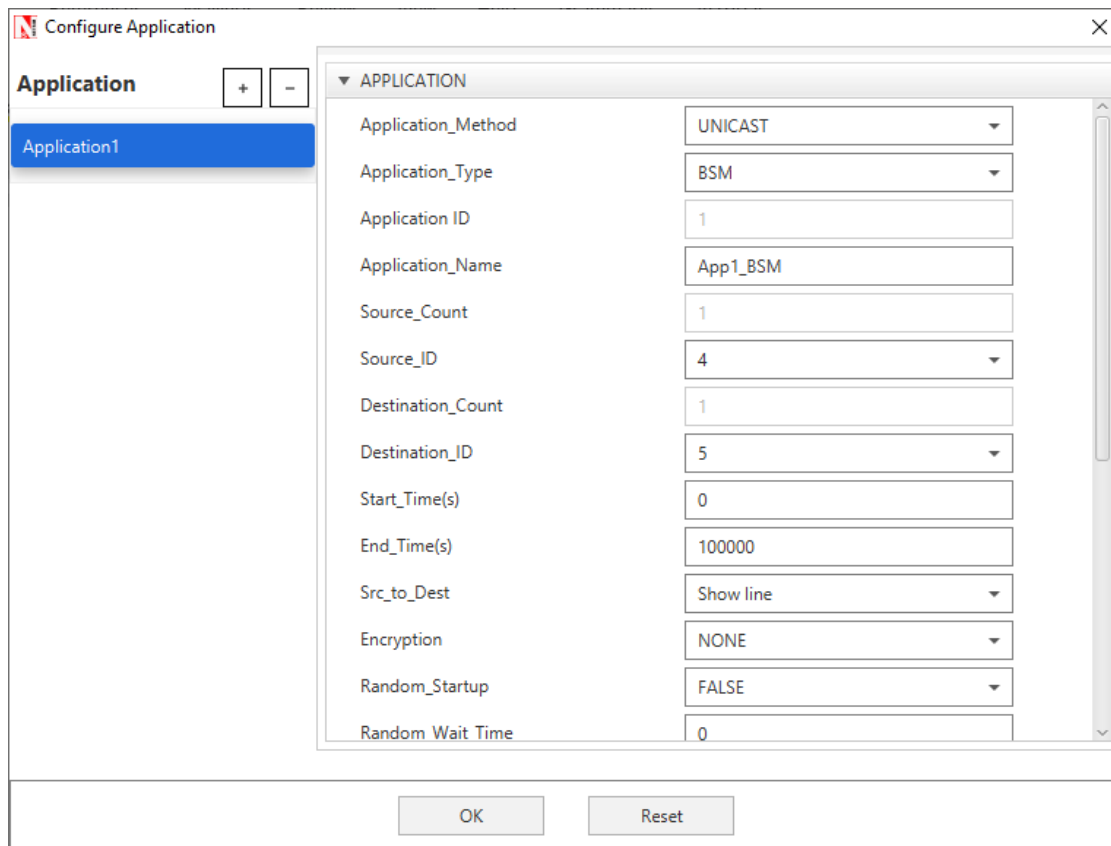


Figure 2-8: Application Configuration window

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 tool bar and click OK. To get detailed help, please refer **sections 8.4 and 8.5** in User Manual. Select Plots icon for enabling Plots and click OK.

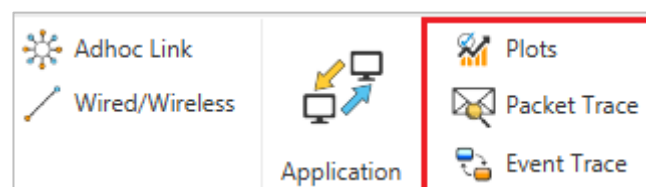


Figure 2-9: Enable Packet Trace, Event Trace & Plots options on top ribbon

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.

- **Record animation** - runs Sumo in background. Users can view animation after completion of Simulation.

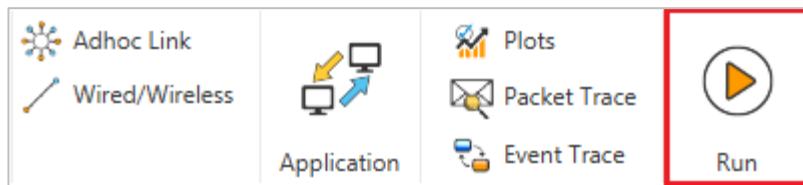


Figure 2-10: Run Simulation option on top ribbon

- **Play & Record animation** – opens NetSim GUI and Sumo GUI in parallel with parameters being continuously passed between the two Simulators.
- **Don't play/record animation** – runs Sumo in Backend. Animation is not recorded.

On running the Simulation by selecting **Play & Record** option, users can view NetSim Packet animation and SUMO simulation simultaneously.

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.

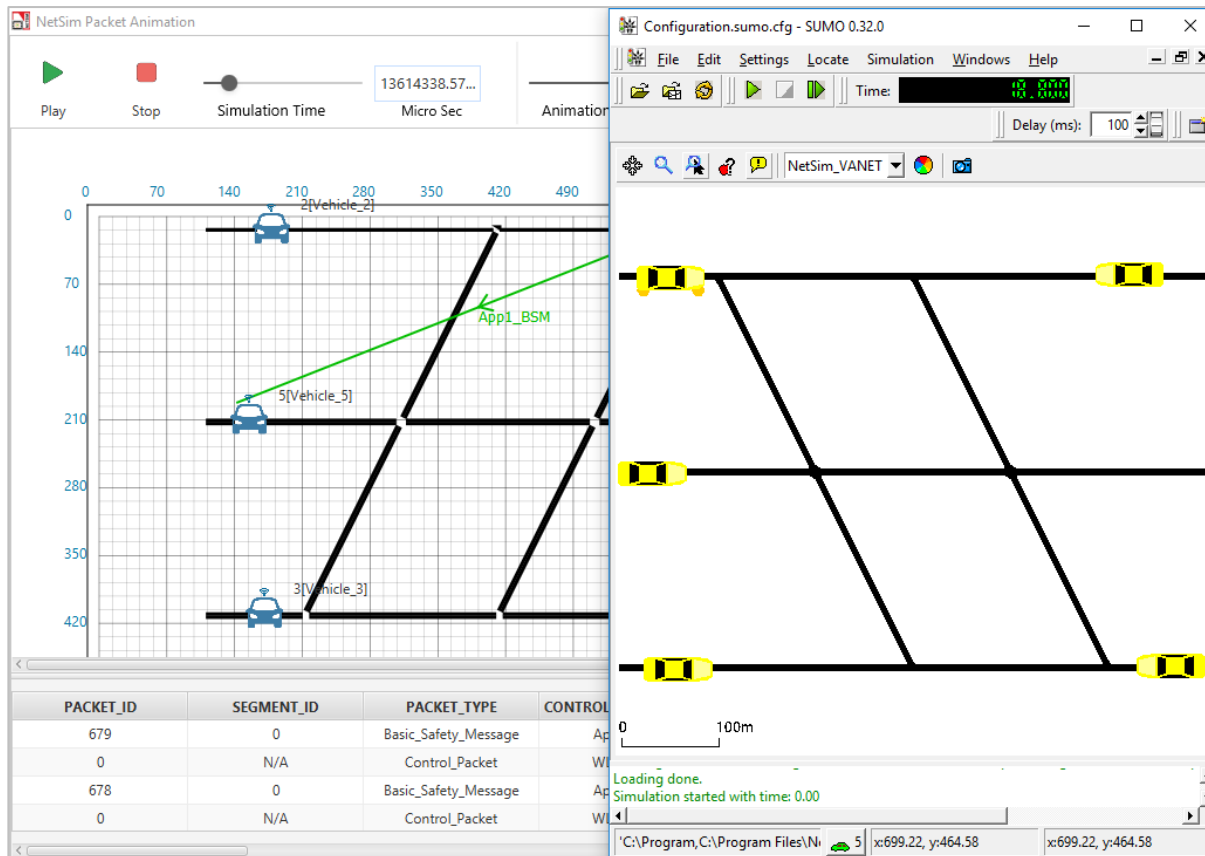


Figure 2-11: Vehicle moments on NetSim Packet animation and SUMO simulation window simultaneously. Users can see the movement of vehicles in SUMO and observe equivalent movement in NetSim. Here users can notice an inversion of view in the GUI, since origin (0, 0) in SUMO is at the left bottom, while origin is at the left top in NetSim.

When users select **Play and Record animation** option, NetSim and SUMO run separately, and users will find that the animation in SUMO is much faster than that of NetSim. This is because, NetSim has to animate the flow of packets between the vehicles in addition to the vehicle movement.

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 $9\ \mu s$ and SIFS equal to $16\ \mu s$.
- 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

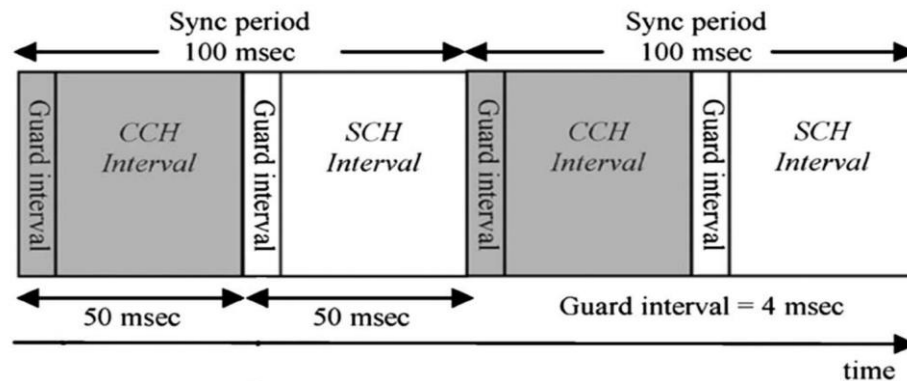


Figure 3-1: We see the time divisions in DSRC. Each synchronization period consists of 1 CCH, 1 SCH and 1 guard interval. While the sync period is generally equal to 100 ms. NetSim allows users to modify the CCH and SCH interval, and in turn the total synchronization period.

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.

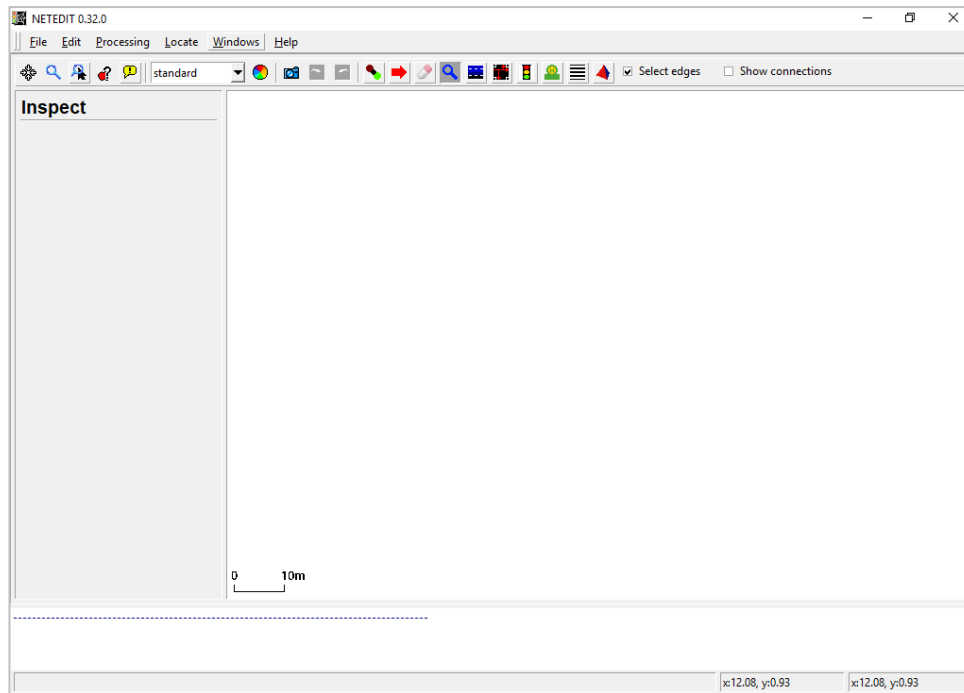


Figure 3-2: SUMO NetEdit Screen

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

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

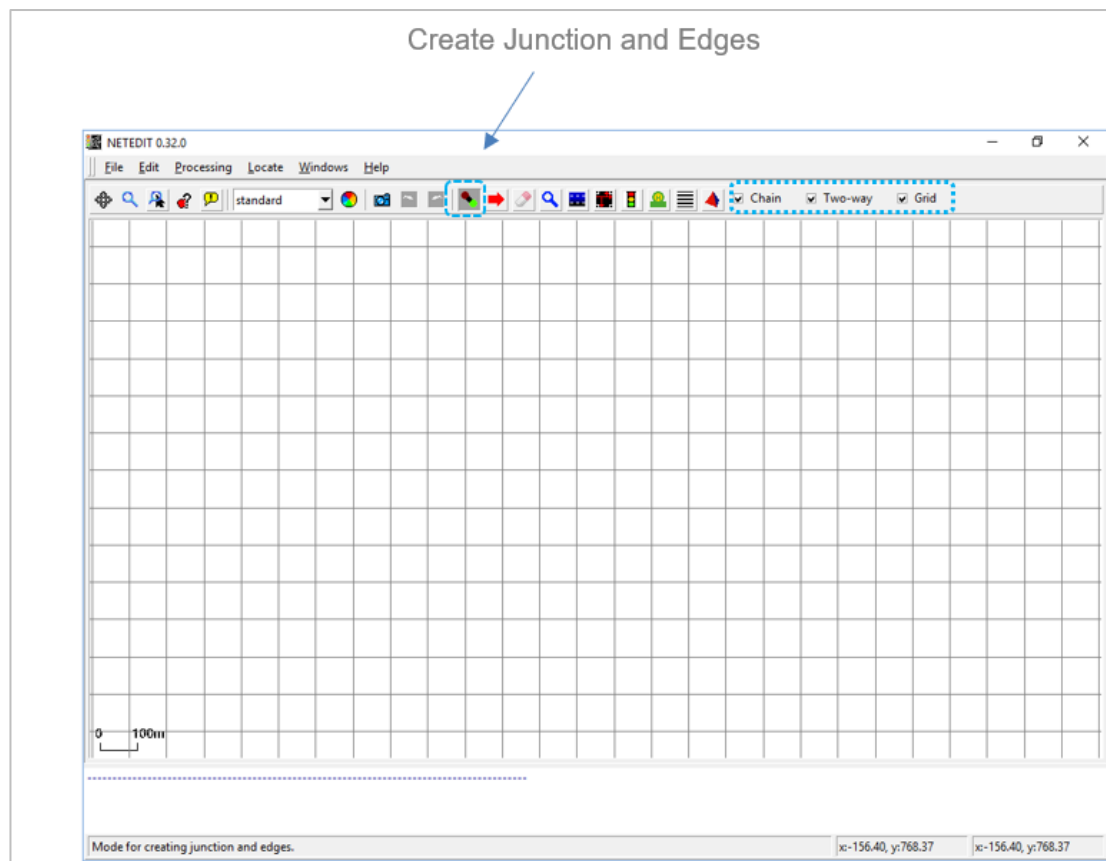


Figure 3-3: SUMO NetEdit design window

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.

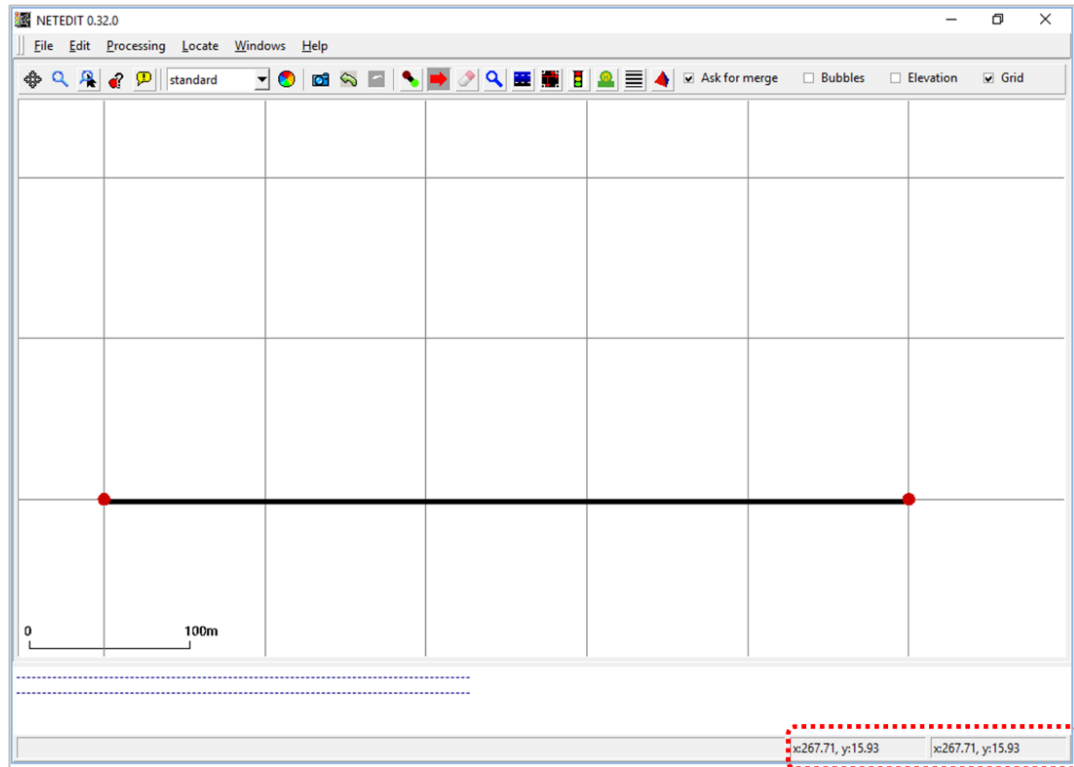


Figure 3-4: Creating edges SUMO NetEdit design window

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

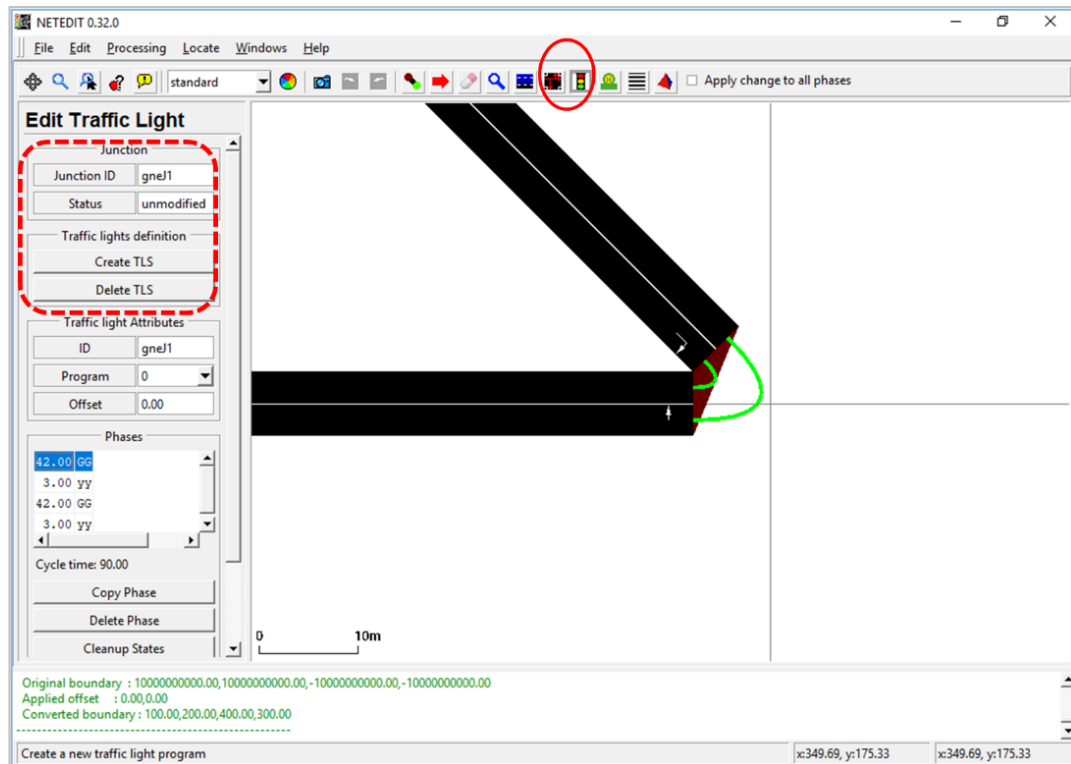


Figure 3-5: Adding Traffic Signal to Network

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

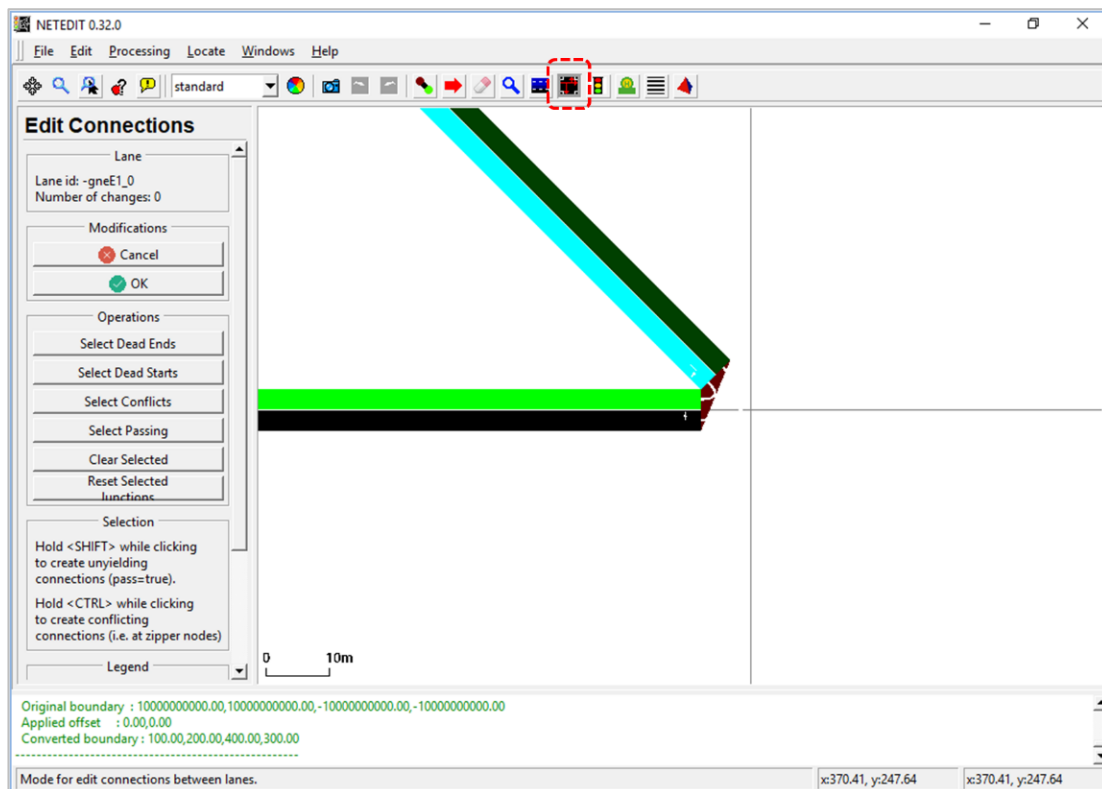


Figure 3-6: Select the lanes and connectivity between them

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

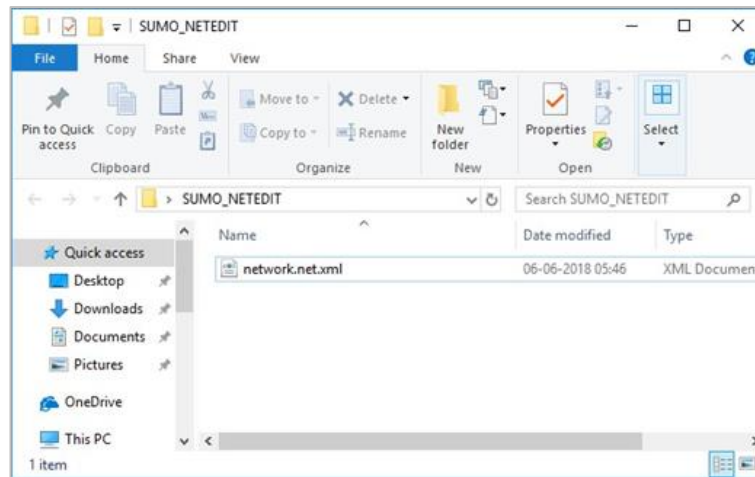


Figure 3-7: network.net.xml inside the folder

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 2 --route-file "trips.xml"

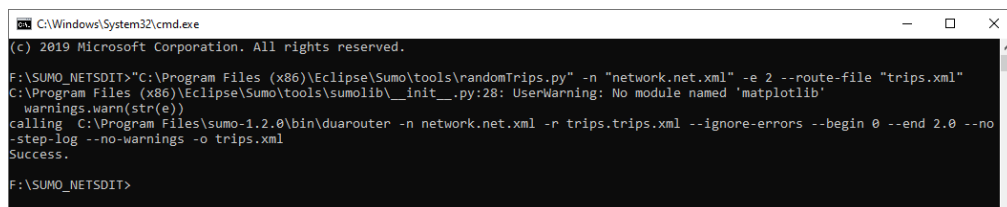


Figure 3-8: Generating route file (trips.xml)

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

Step 11: 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.trips.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.

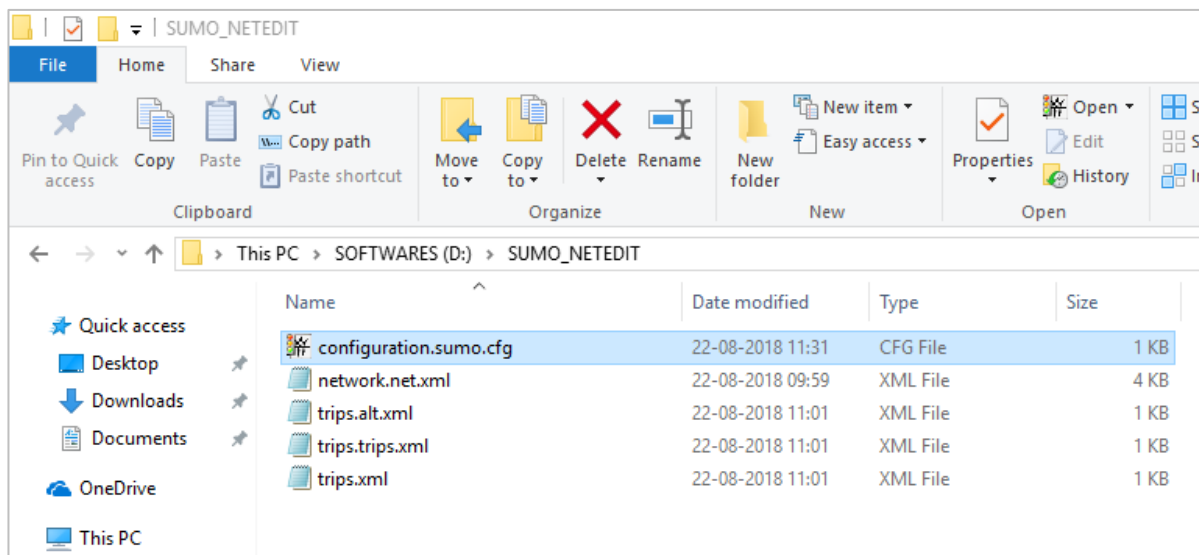


Figure 3-9: Double click on Configuration.sumo.cfg

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.

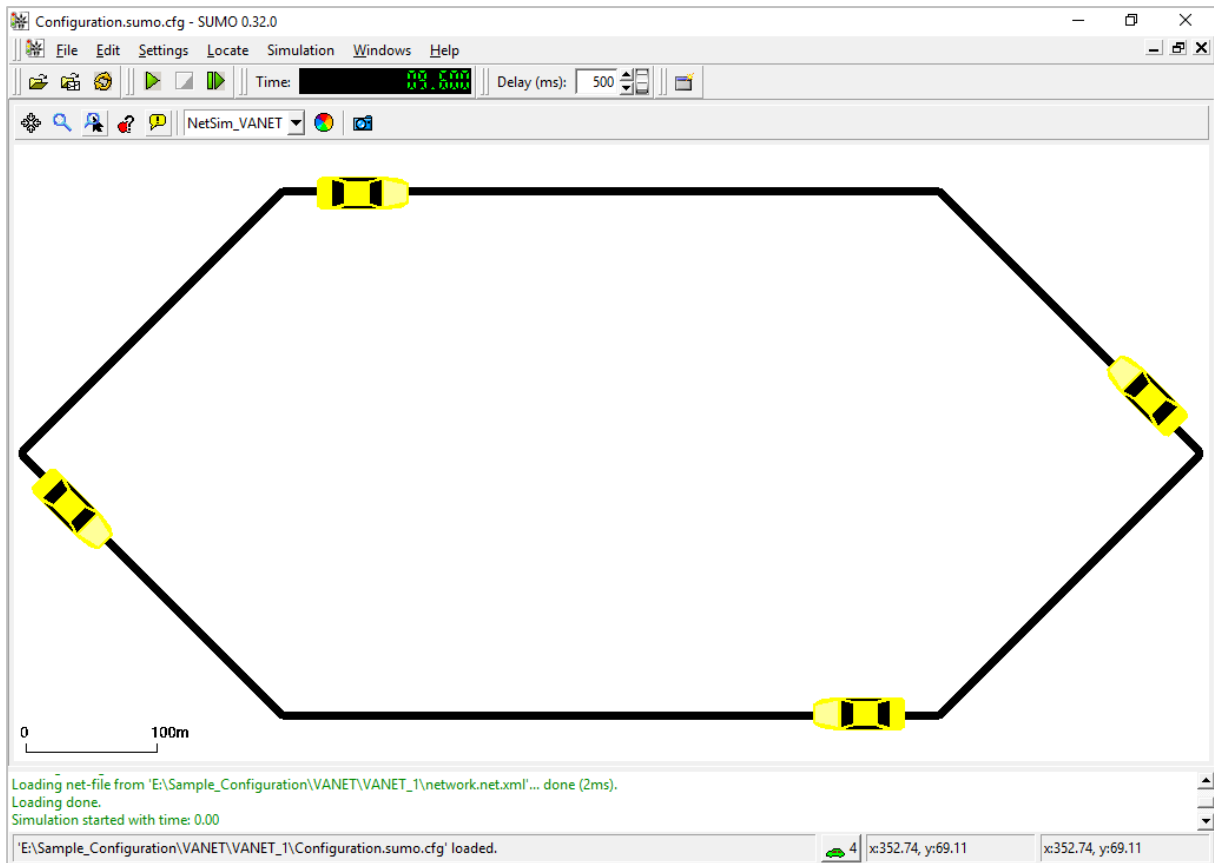


Figure 3-10: SUMO simulation window

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.

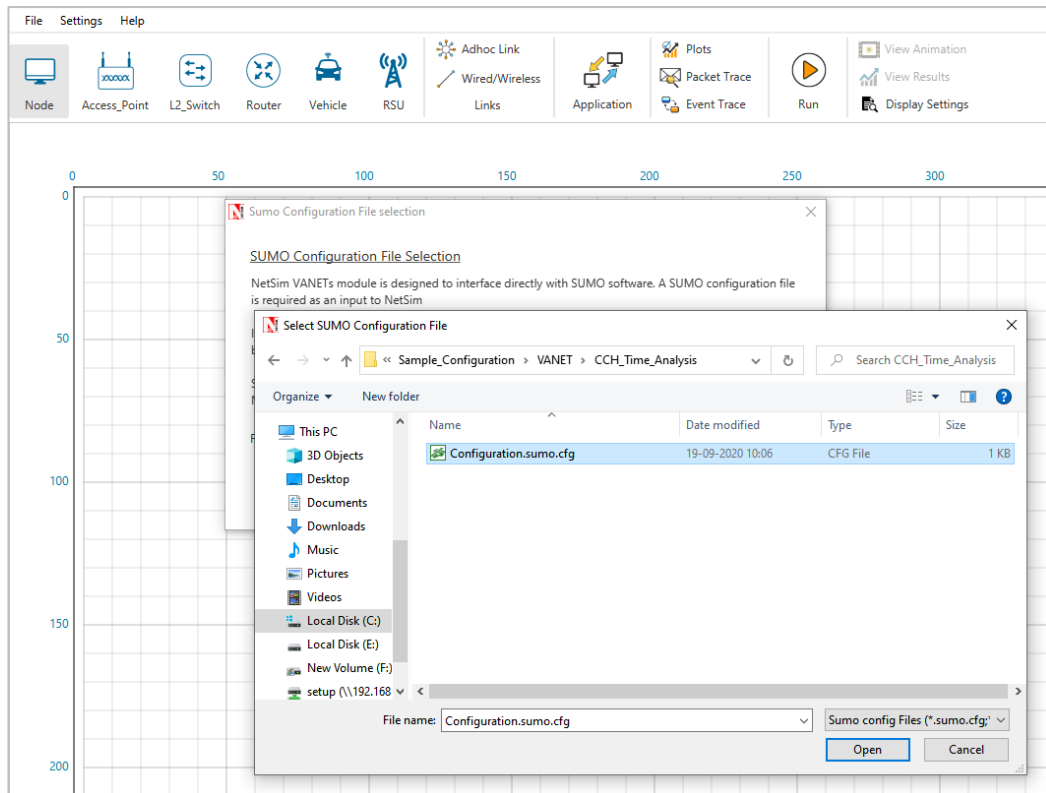


Figure 3-11: Importing a sumo network configuration into NetSim

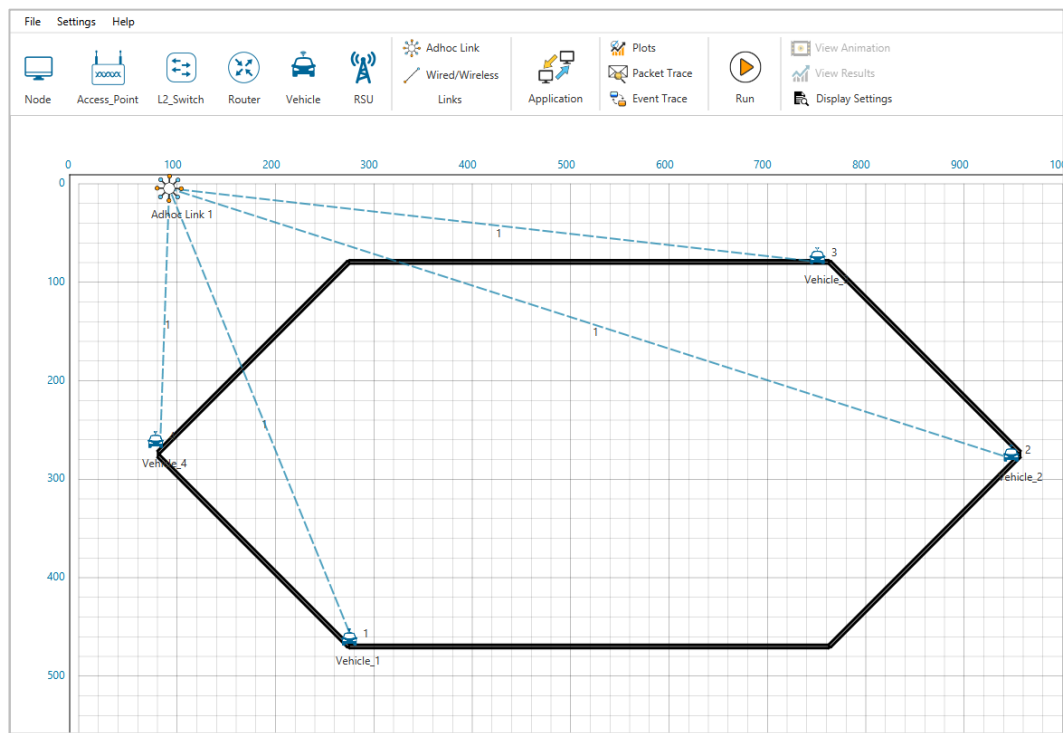


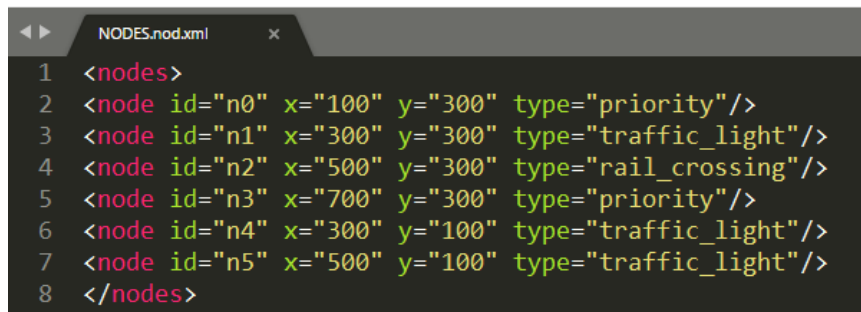
Figure 3-12: Network Topology in this experiment

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).



```
1 <nodes>
2 <node id="n0" x="100" y="300" type="priority"/>
3 <node id="n1" x="300" y="300" type="traffic_light"/>
4 <node id="n2" x="500" y="300" type="rail_crossing"/>
5 <node id="n3" x="700" y="300" type="priority"/>
6 <node id="n4" x="300" y="100" type="traffic_light"/>
7 <node id="n5" x="500" y="100" type="traffic_light"/>
8 </nodes>
```

Figure 3-13: 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).



```
1 <edges>
2 <edge id="L01" from="n1" to="n0" priority="75" nolanes="2" speed="40"/>
3 <edge id="R01" from="n0" to="n1" priority="75" nolanes="2" speed="40"/>
4 <edge id="L12" from="n2" to="n1" priority="75" nolanes="2" speed="30"/>
5 <edge id="R12" from="n1" to="n2" priority="75" nolanes="2" speed="30"/>
6 <edge id="L23" from="n3" to="n2" priority="75" nolanes="2" speed="30"/>
7 <edge id="R23" from="n2" to="n3" priority="75" nolanes="2" speed="30"/>
8 <edge id="D04" from="n0" to="n4" priority="75" nolanes="2" speed="60"/>
9 <edge id="U04" from="n4" to="n0" priority="75" nolanes="2" speed="60"/>
10 <edge id="D14" from="n1" to="n4" priority="75" nolanes="2" speed="40"/>
11 <edge id="U14" from="n4" to="n1" priority="75" nolanes="2" speed="40"/>
12 <edge id="D35" from="n3" to="n5" priority="75" nolanes="2" speed="60"/>
13 <edge id="U35" from="n5" to="n3" priority="75" nolanes="2" speed="60"/>
14 <edge id="D25" from="n2" to="n5" priority="75" nolanes="2" speed="40"/>
15 <edge id="U25" from="n5" to="n2" priority="75" nolanes="2" speed="40"/>
16 <edge id="L45" from="n5" to="n4" priority="75" nolanes="2" speed="40"/>
17 <edge id="R45" from="n4" to="n5" priority="75" nolanes="2" speed="40"/>
18 </edges>
```

Figure 3-14: 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"

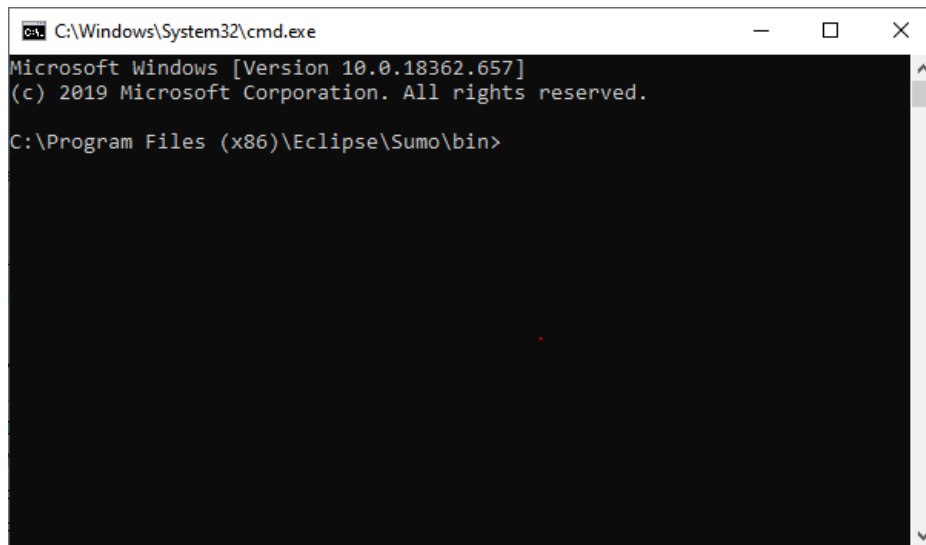


Figure 3-15: Open command prompt in installation directory

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"

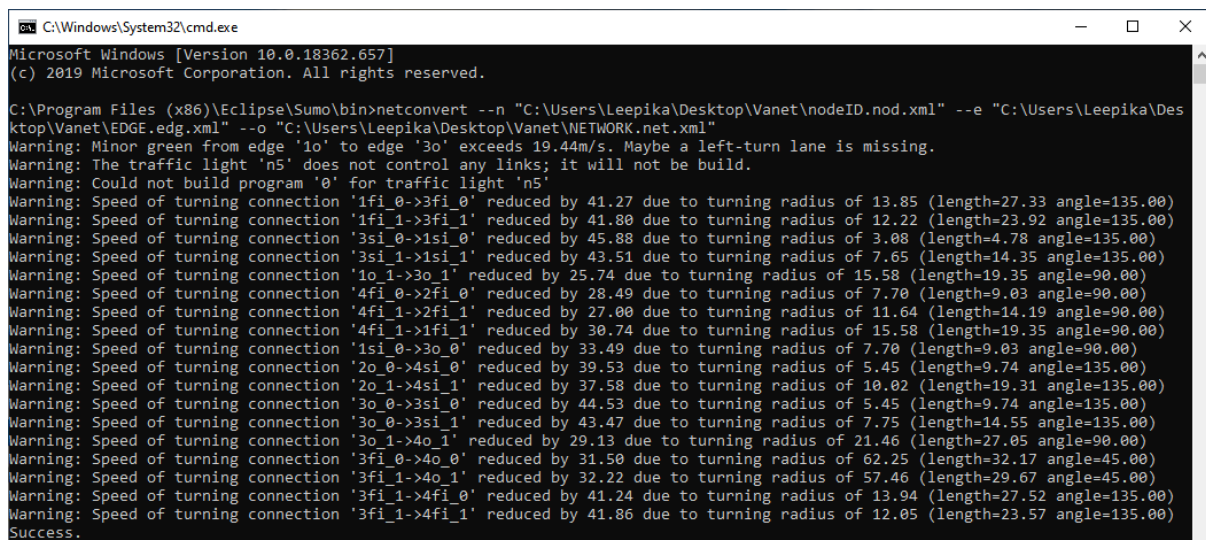


Figure 3-16: Generating Network file by using NETCONVERT command

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

```

1 <routes>
2   <!-- e -->
3   <vehicle id="v0" depart="0.00">
4     <route edges="D04 R45 U35 L23 L12 L01"/>
5   </vehicle>
6
7   <vehicle id="v1" depart="0.00">
8     <route edges="R12 R23 D35 L45 U14 L01"/>
9   </vehicle>
10
11  <vehicle id="v2" depart="0.00">
12    <route edges="R01 R12 R23 D35 L45 U04"/>
13  </vehicle>
14
15  <vehicle id="v3" depart="0.00">
16    <route edges="R01 D14 R45 U35 L23 L12"/>
17  </vehicle>
18
19  <vehicle id="v4" depart="0.00">
20    <route edges="D35 L45 U14 R12 D25 U35 L23"/>
21  </vehicle>
22
23  <vehicle id="v5" depart="0.00">
24    <route edges="R45 U25 L12 L01 D04 R45"/>
25  </vehicle>
26
27 </routes>

```

Figure 3-17: 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.

```

1 <configuration>
2   <input>
3     <net-file value="NETWORK.net.xml"/>
4     <route-files value="VEHICLES.rou.xml"/>
5   </input>
6   <time>
7     <begin value="0"/>
8     <end value="200"/>
9     <step-length value="1"/>
10  </time>
11 </configuration>

```

Figure 3-18: 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.

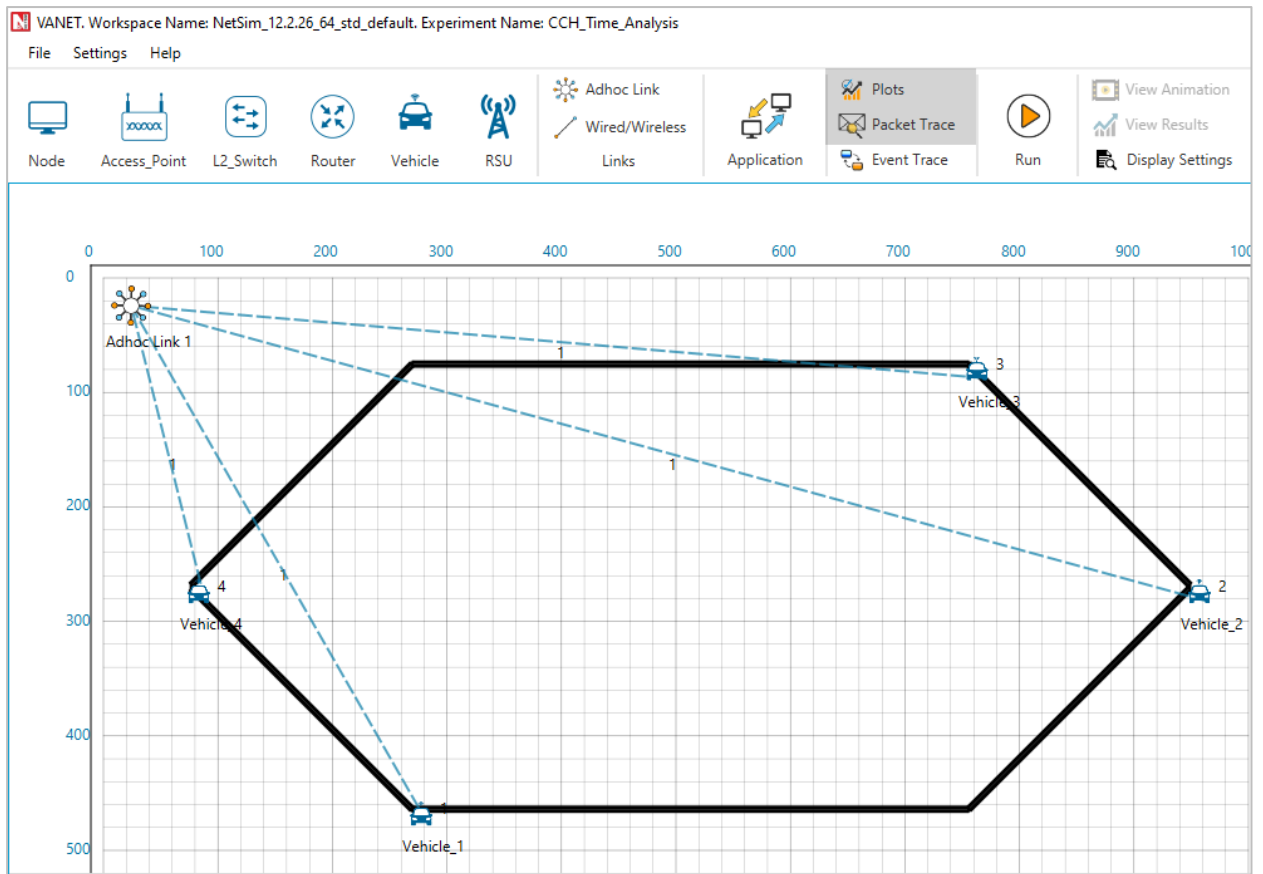


Figure 3-19: Importing a sumo network configuration into NetSim

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



Figure 3-20: RSU icon from the ribbon

RSU's should be connected using Adhoc links manually.

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 > Simple Scenario** then click on the tile in the middle panel to load the example as shown in below screenshot

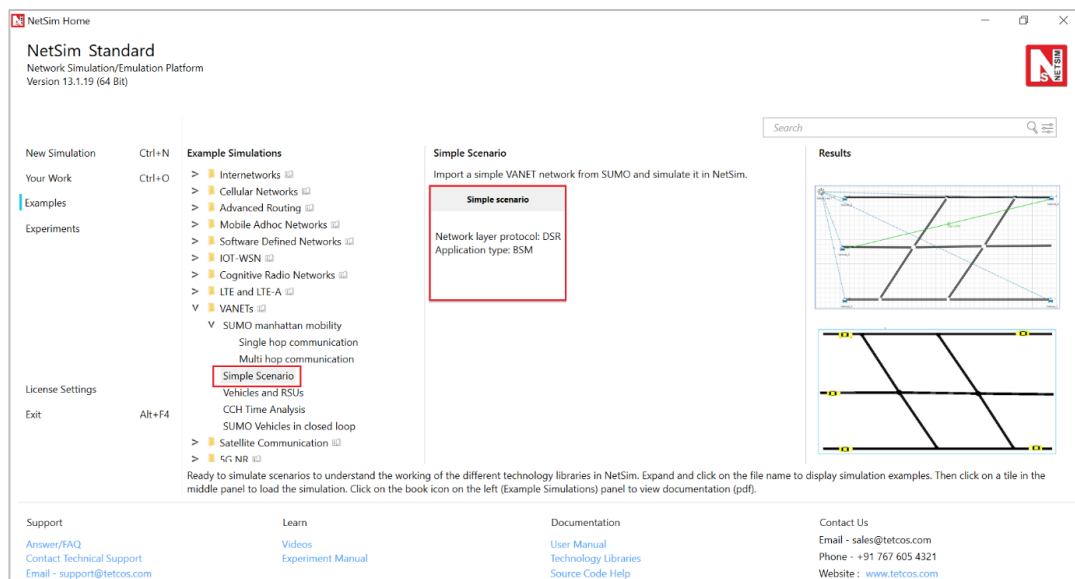


Figure 4-1: List of scenarios for the example of Simple Scenario

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

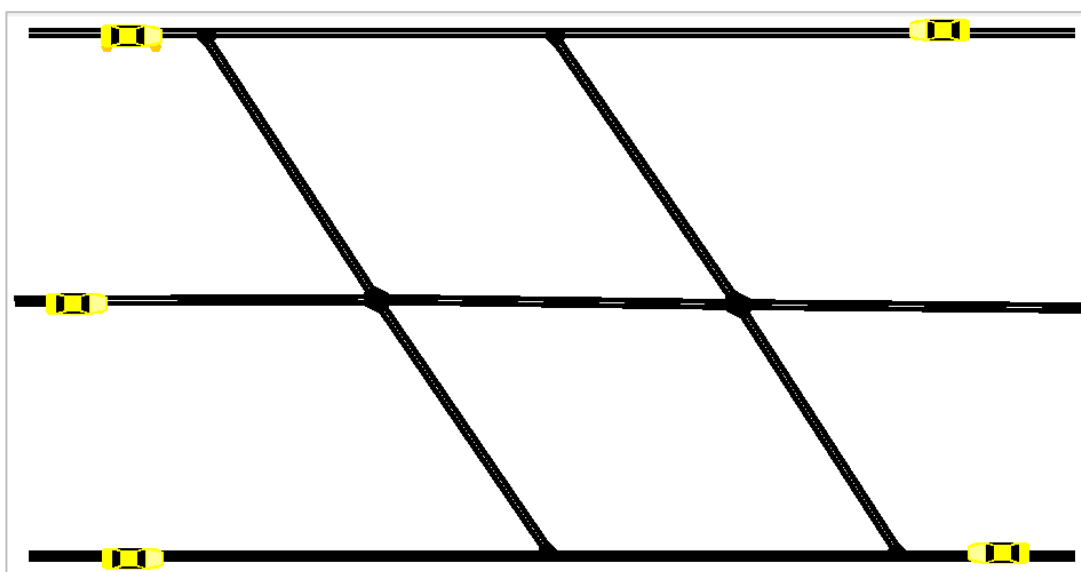


Figure 4-2: Network topology in Sumo

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.

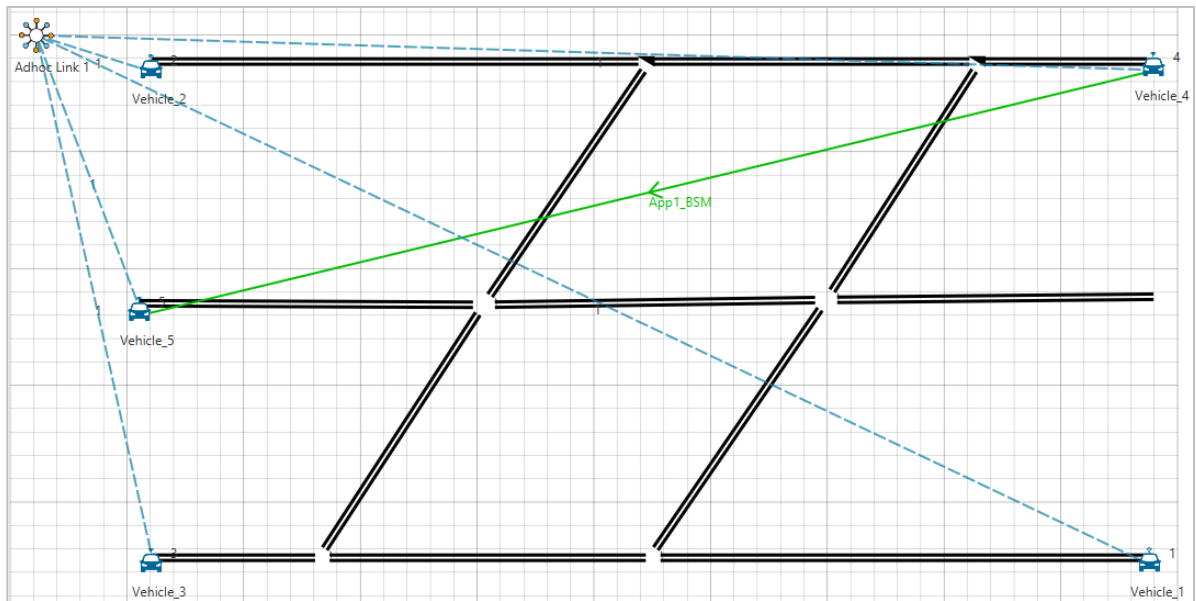


Figure 4-3: Network scenario after importing into NetSim and configuring application traffic

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

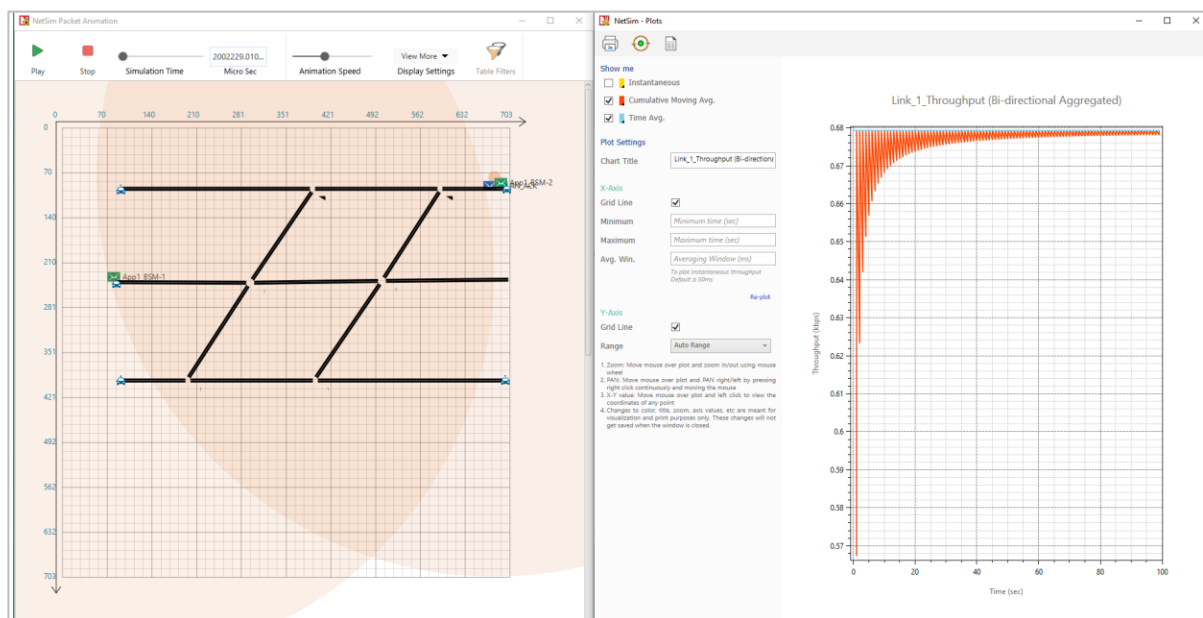


Figure 4-4: Packet animation window and link throughput plot

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

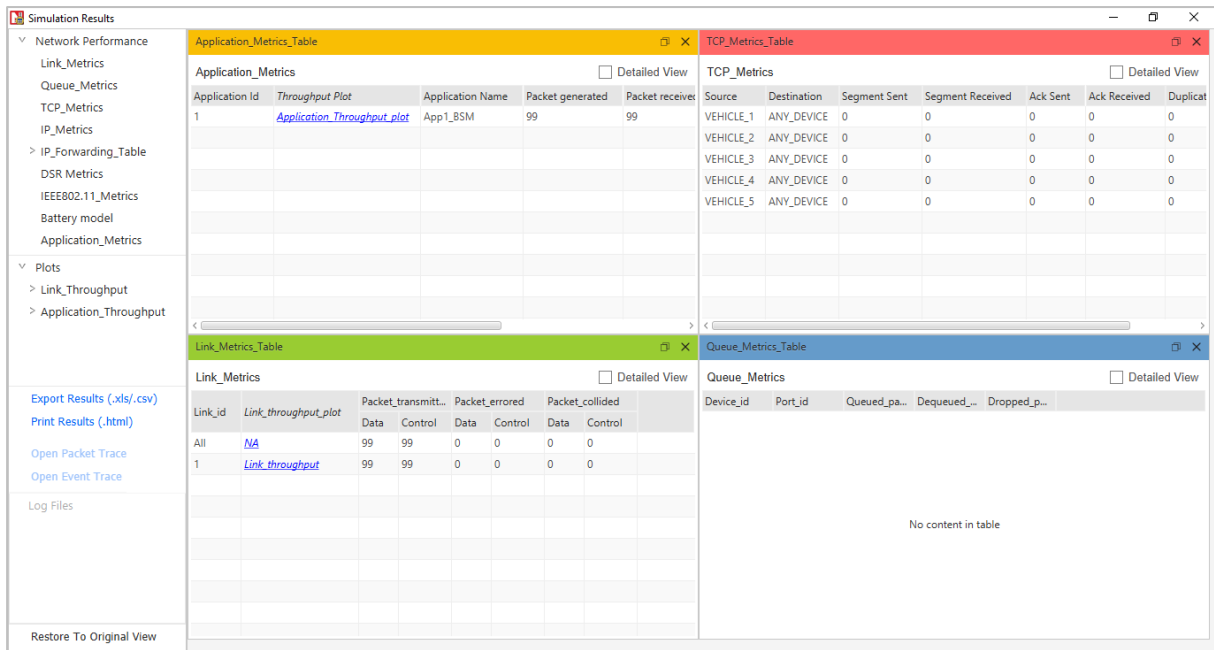


Figure 4-5: Result Dashboard

4.2 V2V and V2I communication involving Vehicles and RSUs

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

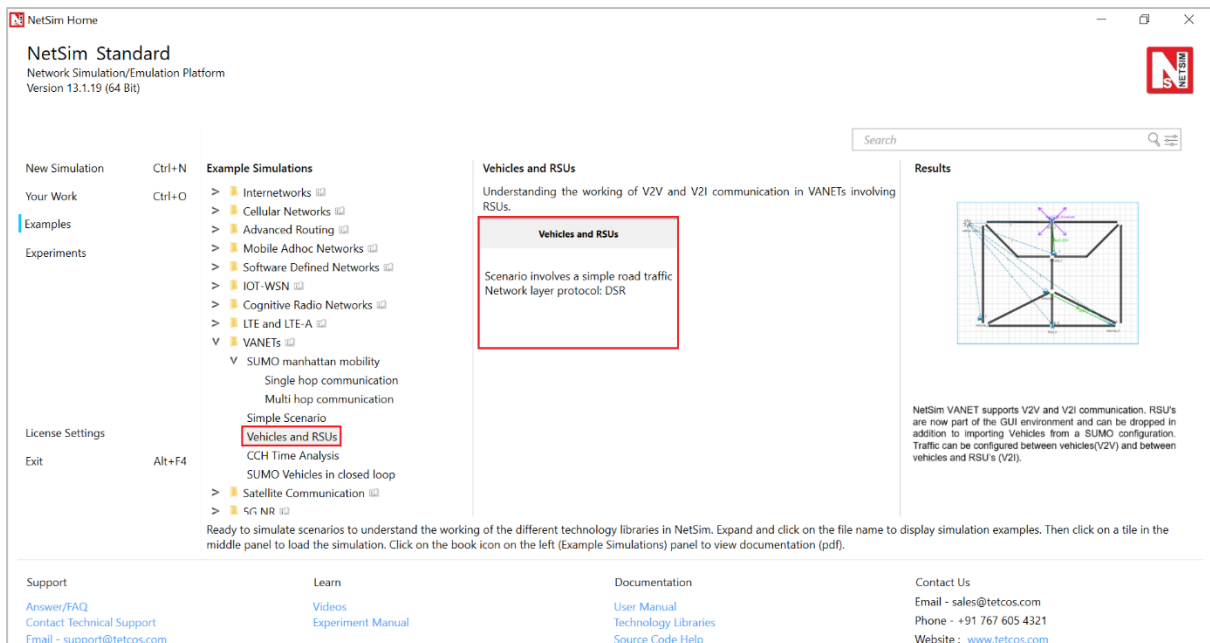


Figure 4-6: List of scenarios for the example of Vehicles and RSUs

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:

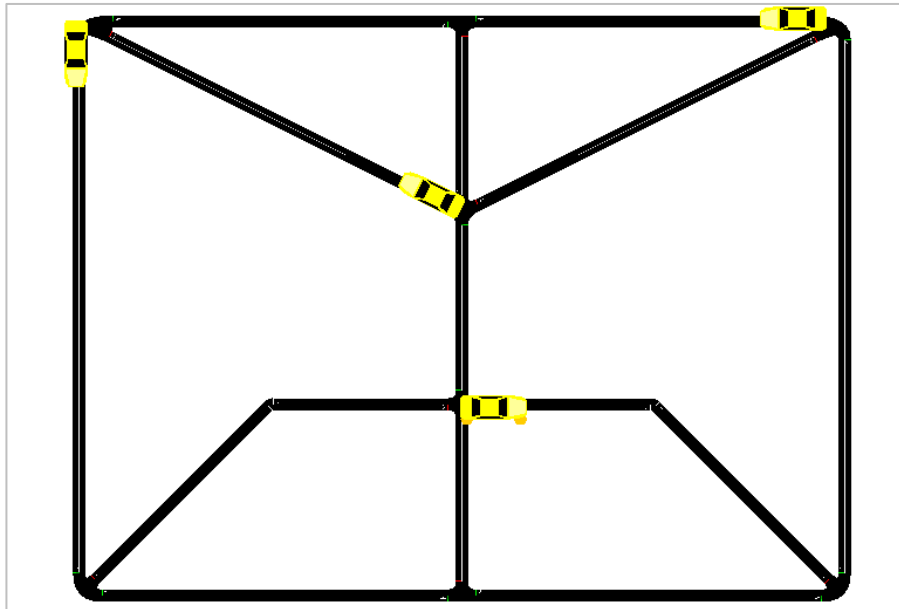


Figure 4-7: Network topology in Sumo

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

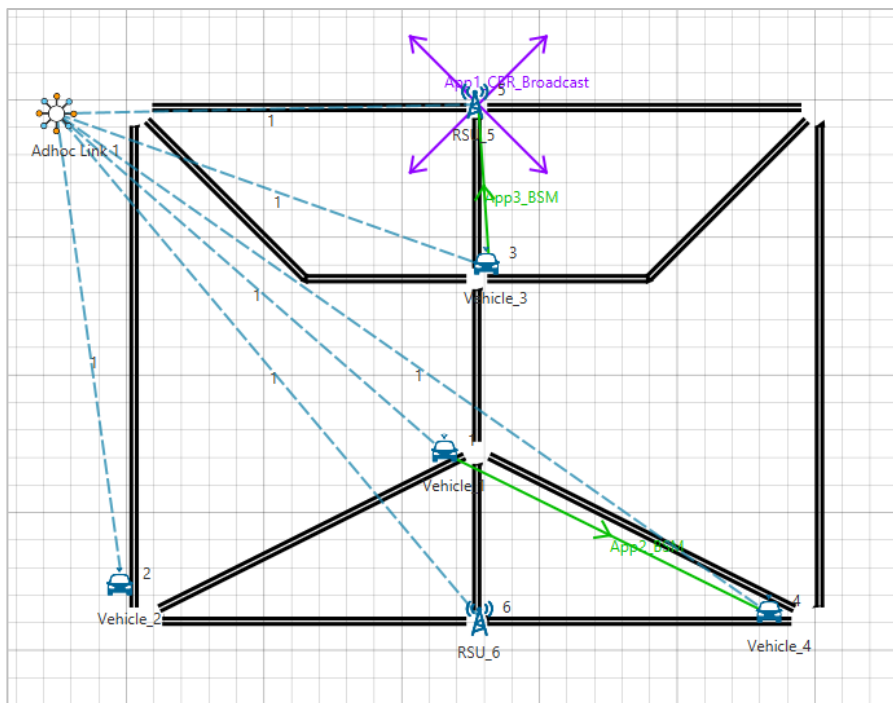


Figure 4-8: Network set up for scenario involving vehicles and RSUs for v2v and v2i communication

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.

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

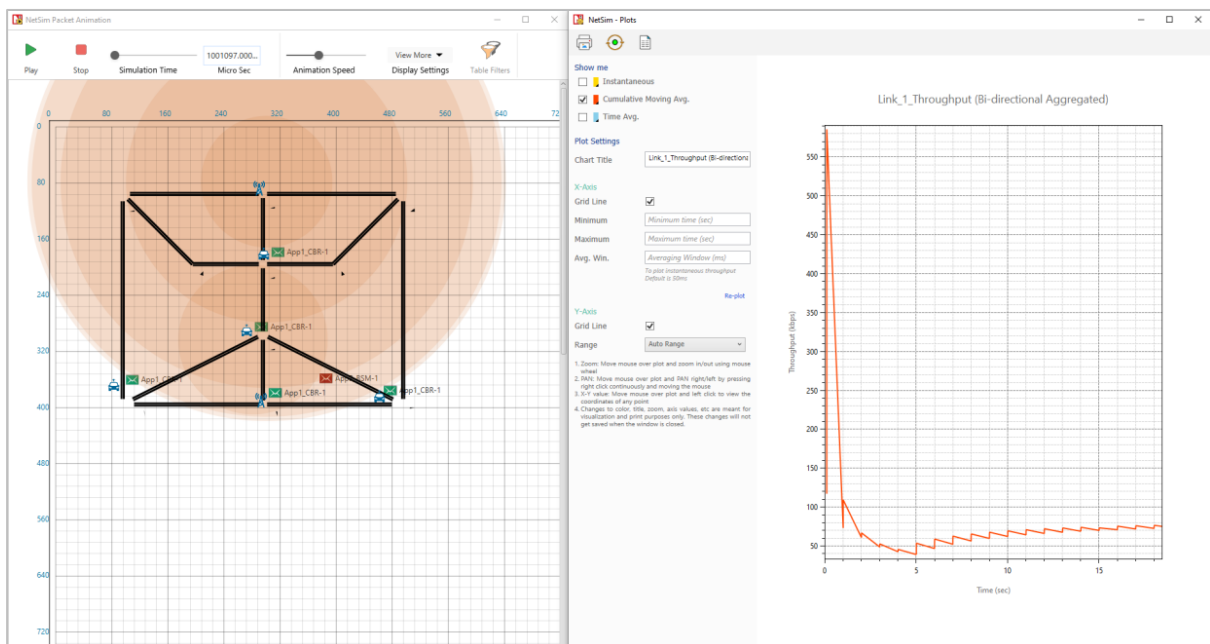


Figure 4-9: Packet animation window and link throughput plot

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

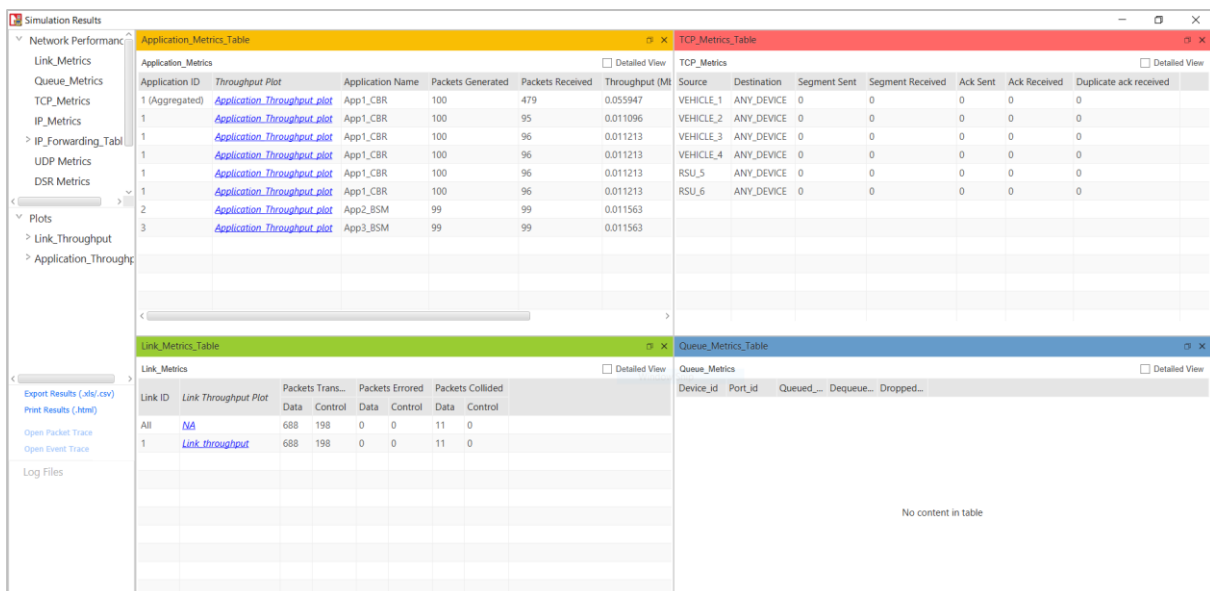


Figure 4-10: Result Dashboard

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

All of 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

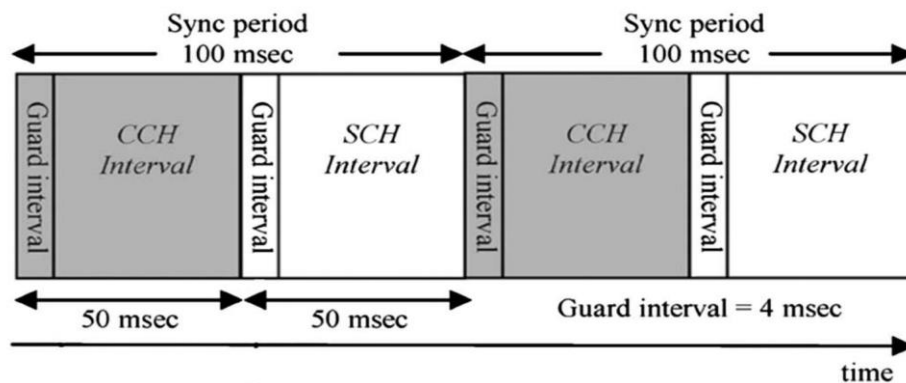


Figure 4-11: We see the time divisions in DSRC. Each synchronization period consists of 1 CCH, 1 SCH and 1 guard interval. While the sync period is generally equal to 100 ms. NetSim allows users to modify the CCH and SCH interval, and in turn the total Sync period.

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

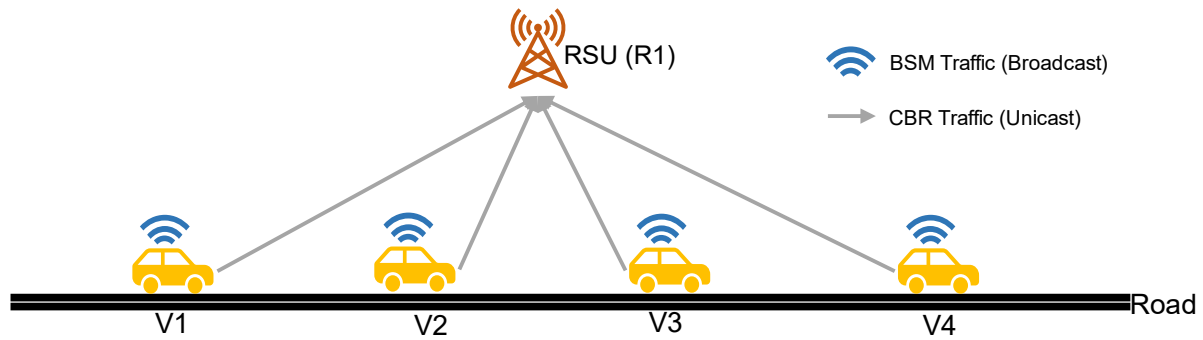


Figure 4-12: Illustration of the VANET scenario under study. The network comprises of 4 vehicles and 1 roadside unit. Each vehicle transmits two applications: (i) a BSM broadcast application that is sent to all other devices (vehicles plus RSU) within range and (ii) a CBR application transmitted to the 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 4-11, 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

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

Application	Application Type	Gen. Rate (Mbps)	CCH 20 ms SCH 80 ms	CCH 25 ms SCH 75 ms	CCH 30 ms SCH 70 ms	CCH 50 ms SCH 50 ms
			Throughput (Mbps)	Throughput (Mbps)	Throughput (Mbps)	Throughput (Mbps)
BSM 1	Broadcast	0.5	0.090	0.112	0.134	0.226
BSM 2	Broadcast	0.5	0.099	0.124	0.149	0.248
BSM 3	Broadcast	0.5	0.105	0.131	0.158	0.264
BSM 4	Broadcast	0.5	0.108	0.136	0.164	0.275
Sum Throughput (Mbps)			0.402	0.504	0.605	1.014
Sum Throughput * (SCH+CCH)/CCH			2.011	2.015	2.017	2.027
CBR 1	Unicast	2	1.020	1.241	1.146	0.456
CBR 2	Unicast	2	0.695	0.667	0.903	0.868
CBR 3	Unicast	2	1.379	0.972	0.594	0.612
CBR 4	Unicast	2	1.808	1.626	1.553	1.109
Sum Throughput (Mbps)			4.902	4.507	4.195	3.045
Sum Throughput * (SCH+CCH)/SCH			6.127	6.009	5.993	6.090

Table 4-1: We see that the as the CCH interval increases, BSM application has higher throughput rate. Similarly, as the SCH Interval decreases there is decrease in throughput rate.

4.3.4.1 Observations

1. BSM is 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

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.

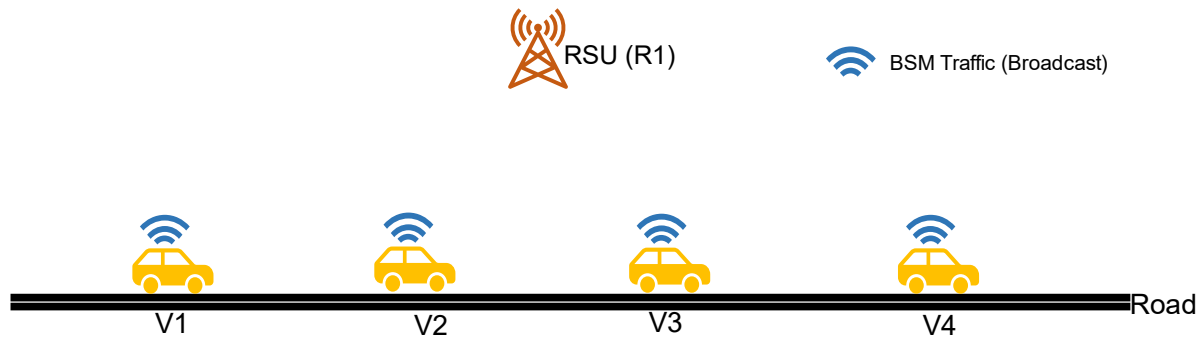
Application	Application Type	Gen. Rate (Mbps)	CCH 20 ms SCH 80 ms	CCH 25 ms SCH 75 ms	CCH 30 ms SCH 70 ms	CCH 50 ms SCH 50 ms
			Delay (Micro sec)	Delay (Micro sec)	Delay (Micro sec)	Delay (Micro sec)
BSM 1	Broadcast	0.025	38774.178	33711.174	29184.264	15302.952
BSM 2	Broadcast	0.025	39085.500	33999.099	29580.397	15452.973
BSM 3	Broadcast	0.025	38871.482	34065.658	29612.045	15592.555
BSM 4	Broadcast	0.025	38971.652	34244.557	29722.204	15687.207
Average Delay			38925.703	34005.122	29524.728	15508.922
CBR 1	Unicast	1	970820.956	623360.782	1070605.026	3231622.594
CBR 2	Unicast	1	554478.563	1711244.232	938543.697	3167638.761
CBR 3	Unicast	1	660493.934	1021442.963	1682596.057	3767856.679
CBR 4	Unicast	1	827215.500	694044.413	1435345.051	4040215.389
Average Delay			753252.238	1012523.097	1281772.458	3551833.356

Table 4-2: 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.

4.3.5.1 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



The scenario layout remains the same, however we change the application settings. In this example we only have the 4 BSM applications. There are no CBR applications. The application generation rates are mentioned in Row 1 (shaded grey).

Application	Application Type	Gen Rate 0.005 Mbps	Gen Rate 0.010 Mbps	Gen Rate 0.015 Mbps	Gen Rate 0.020 Mbps
		Collision Count	Collision Count	Collision Count	Collision Count
BSM 1	Broadcast	1148	2686	4347	6157
BSM 2	Broadcast	826	1897	3049	4337
BSM 3	Broadcast	617	1298	2223	3014
BSM 4	Broadcast	474	1034	1670	2370
Total collisions		3065	6915	11289	15878
Total pkts transmitted		28934	57818	86720	115578
Collision Probability		0.106	0.120	0.130	0.137

Table 4-3: Comparison of Collision count of BSM applications with changing generation rate

4.3.6.1 Observations

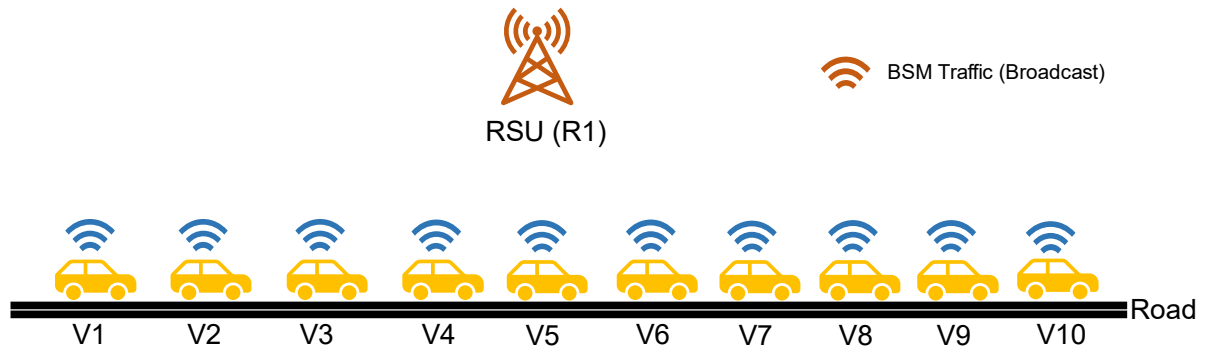
- Saturation throughput is about 0.25 Mbps per app or 1 Mbps total. Note the generation rate is below the saturation capacity of the network
- We see collision probability increases as generation rate increases
- To the best of our knowledge the mathematical modelling of collisions with non-saturated 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

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

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

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



4.3.7.1 Results

Number of Tx nodes	Collision Count	Pkts transmitted	Collision Probability
1	0	24786	0.000
2	5297	69515	0.076
3	15080	127398	0.118
4	28723	179751	0.160
5	45665	244391	0.187
6	66506	316725	0.210
7	89357	396539	0.225
8	127606	490200	0.260
9	166575	586842	0.284
10	216332	694672	0.311

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

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

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

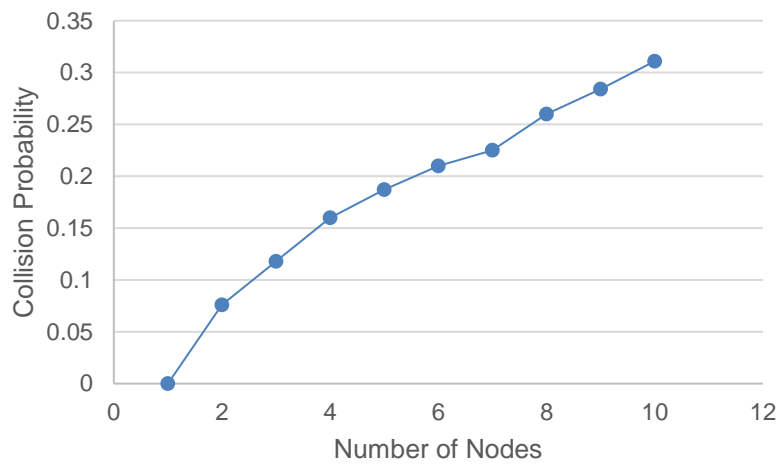


Figure 4-13: Collision probability vs. number of transmitting nodes

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.

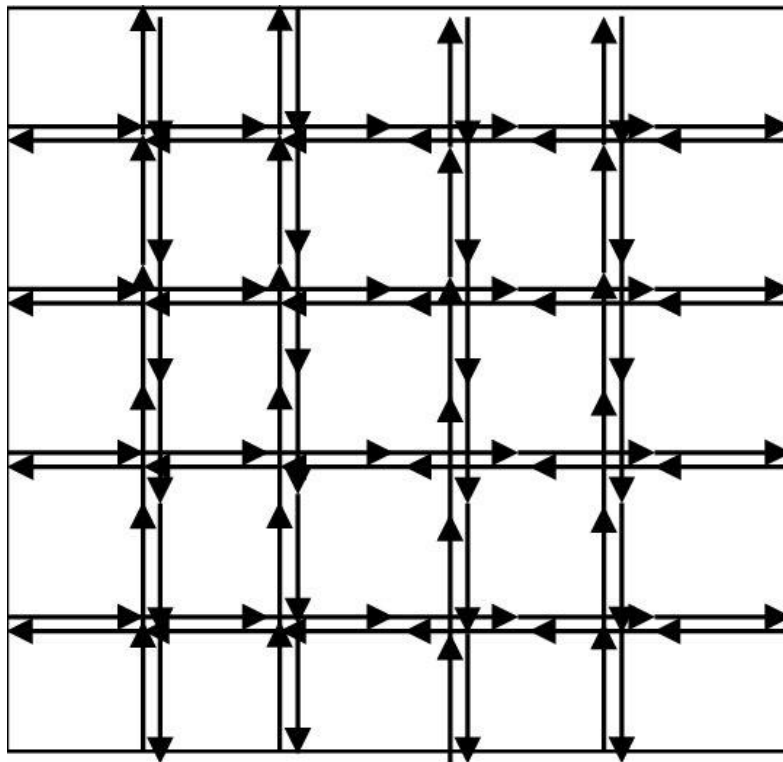


Figure 4-14: Manhattan mobility in SUMO features a grid topology

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

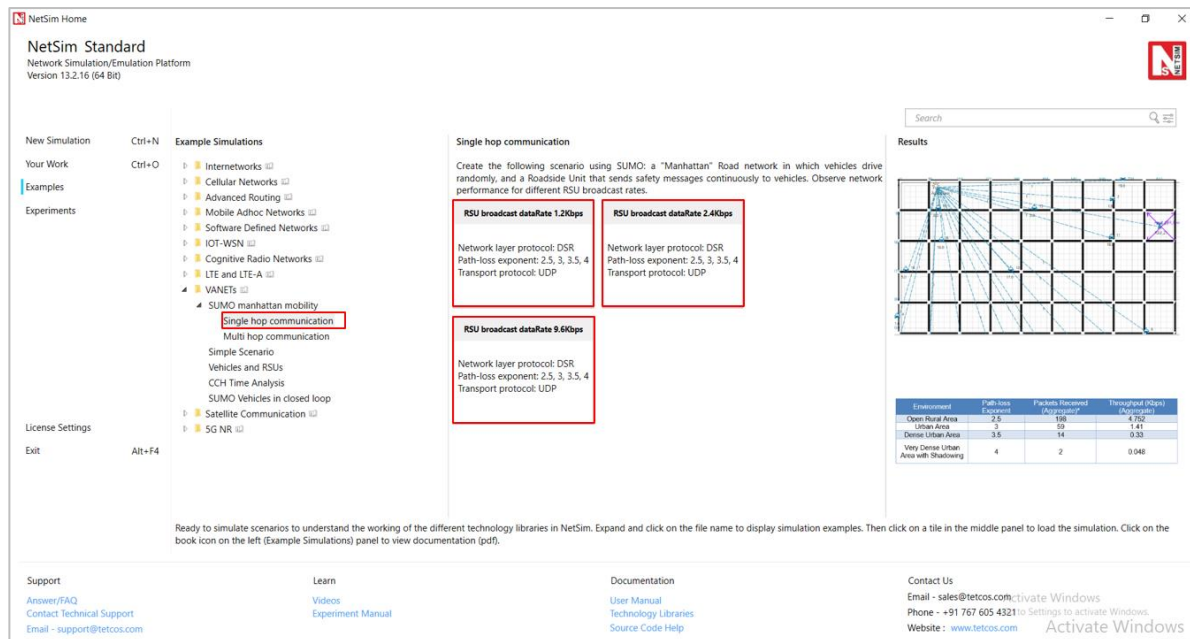


Figure 4-15: List of scenarios for the example of Single hop communication

The NetSim UI would display as shown below.

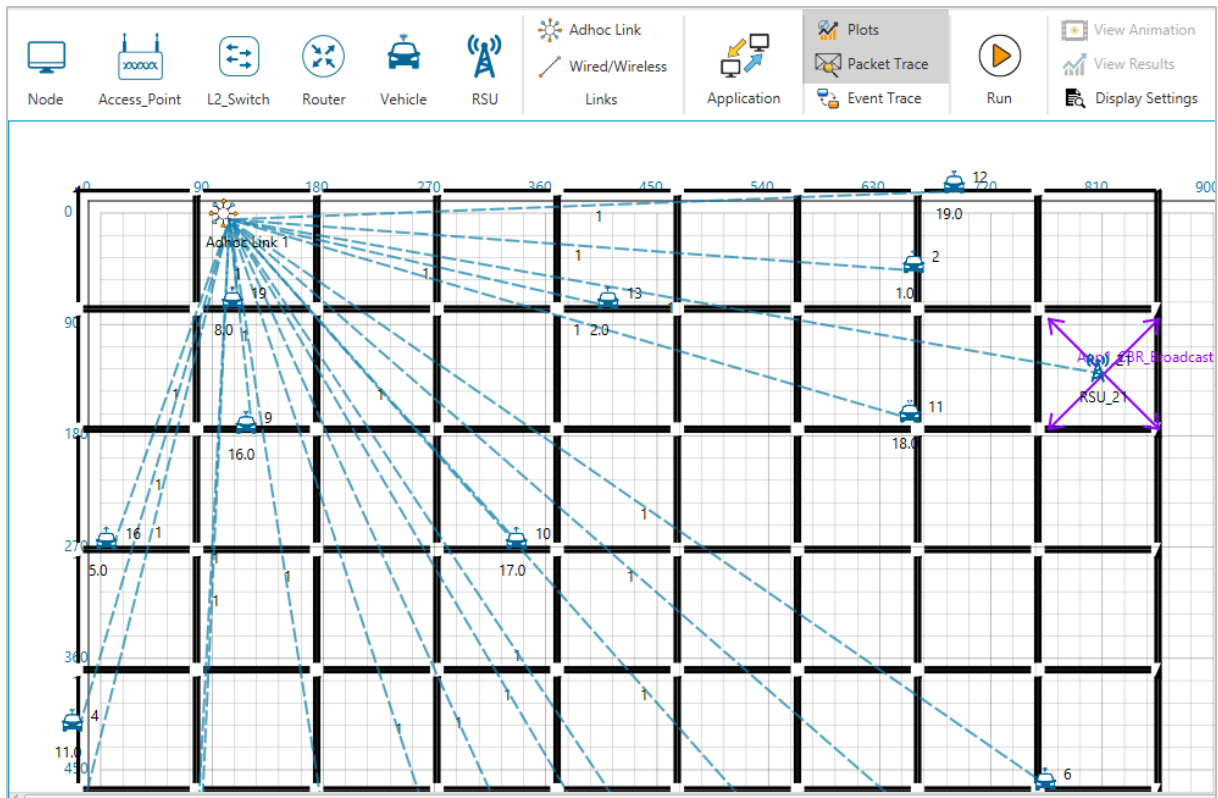


Figure 4-16: Network set up for studying the Single hop communication

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 (μ s)
Broadcast	CBR	APP_1_CBR_Broadcast	21	Broadcast to all vehicles	300	2,000,000

Table 4-5: 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 4-6: 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 and observe the movement of the vehicles in the packet animation window.
7. In NetSim packet animation window, you can see that vehicles choose random directions when they reach a junction in the Manhattan grid network.

8. 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.

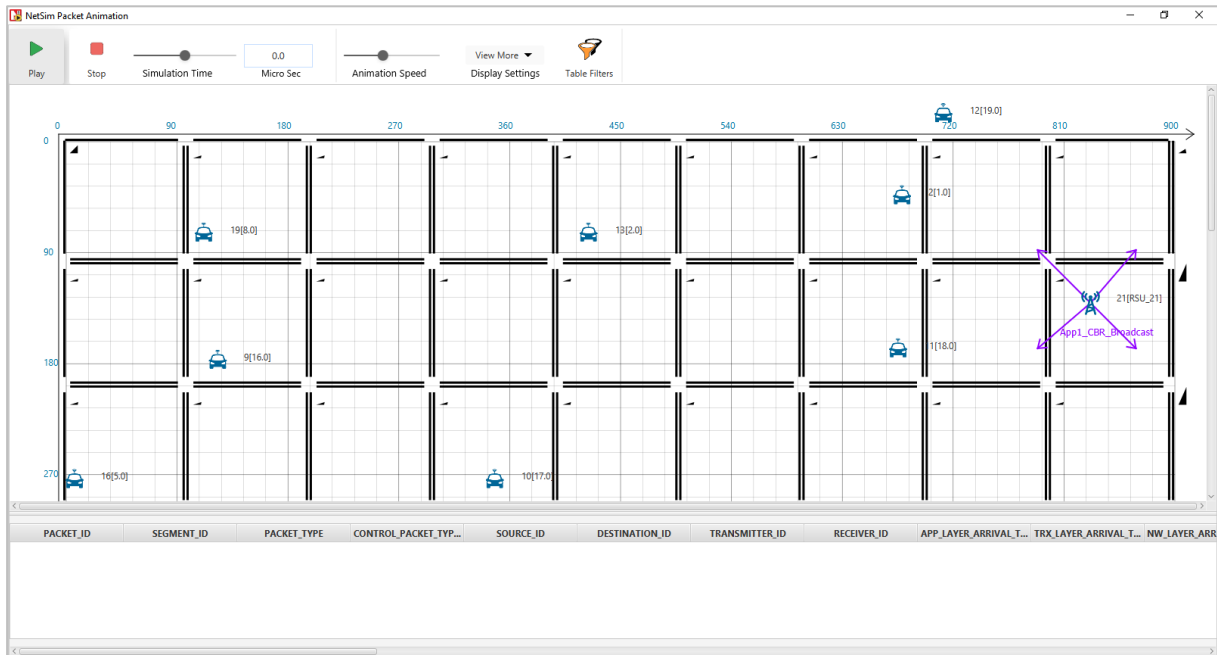


Figure 4-17: Animation Window for NetSim

With play and record animation enabled, same can be observed in SUMO as follows:

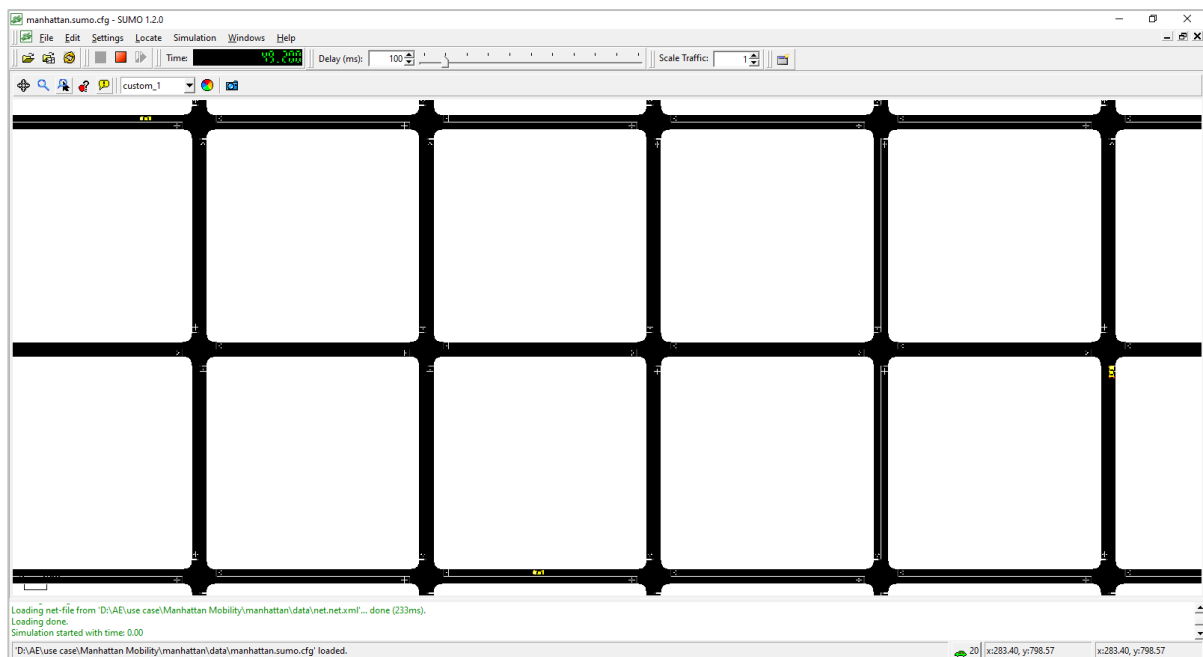


Figure 4-18: Animation Window for Sumo

Results and Observations

For sample **RSU Broadcast Data Rate = 1.2 Kbps** (Packet size = 300 bytes, IAT = 2,000,000 μ s. 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	198	4.752
Urban Area	3	59	1.41
Dense Urban Area	3.5	14	0.33
Very Dense Urban Area with Shadowing	4	2	0.048

Table 4-7: 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 = 1,000,000 μ s 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	397	9.528
Urban Area	3	119	2.856
Dense Urban Area	3.5	26	0.624
Very Dense Urban Area with Shadowing	4	4	0.096

Table 4-8: Results Comparison for RSU Broadcast Data Rate = 2.4 Kbps

For sample RSU Broadcast Data Rate = 9.6 Kbps (Packet size =300Bytes, IAT =250,000 μ s 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	1593	38.232
Urban Area	3	482	11.56
Dense Urban Area	3.5	102	2.448
Very Dense Urban Area with Shadowing	4	16	0.384

Table 4-9: Results Comparison for RSU Broadcast Data Rate = 9.6 Kbps

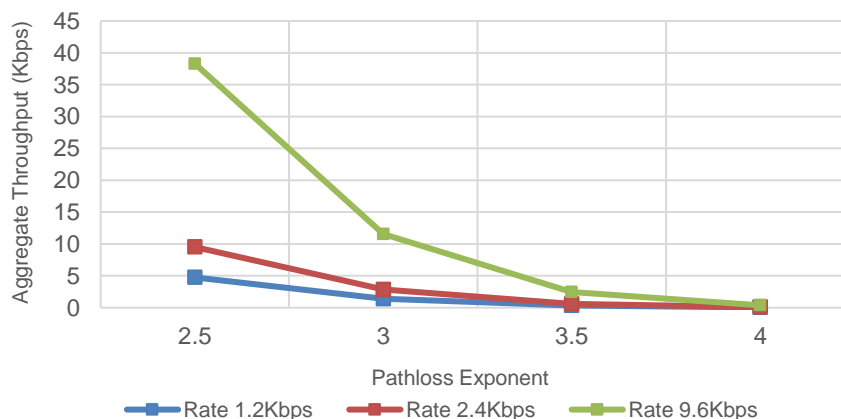


Figure 4-19: Plot of Throughput vs. Pathloss Exponent for different RSU broadcast for different DR (Data Rates)

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

NetSim Standard
Network Simulation/Emulation Platform
Version 13.2.20 (64 Bit)

New Simulation Ctrl+N
Your Work Ctrl+O
Examples
Experiments

Example Simulations
Internetworks
Cellular Networks
Advanced Routing
Mobile Adhoc Networks
Software Defined Networks
IOT-WSN
Cognitive Radio Networks
LTE and LTE-A
VANETs
SUMO manhattan mobility
Single hop communication
Multi hop communication
Simple Scenario
Vehicles and RSUs
CCH Time Analysis
SUMO Vehicles in closed loop
Satellite Communication
5G NR

License Settings
Exit Alt+F4

Multi hop communication
Create the following scenario using SUMO: a "Manhattan" Road network in which vehicles drive randomly, and a Roadside Unit that sends safety messages continuously to vehicles. Observe network performance for different vehicle counts.

10 Vehicles	20 Vehicles	30 Vehicles
Network layer protocol: DSR No. of Vehicle: 10	Network layer protocol: DSR No. of Vehicle: 20	Network layer protocol: DSR No. of Vehicle: 30

40 Vehicles	50 Vehicles	60 Vehicles
Network layer protocol: DSR No. of Vehicle: 40	Network layer protocol: DSR No. of Vehicle: 50	Network layer protocol: DSR No. of Vehicle: 60

70 Vehicles
Network layer protocol: DSR No. of Vehicle: 70

Results

Number of vehicles	Throughput (Kbps) (Aggregate)*
10	1.1826
20	0.5405
30	0.6668
40	1.1857
50	1.0780
60	0.9119
70	0.5720

Ready to simulate scenarios to understand the working of the different technology libraries in NetSim. Expand and click on the file name to display simulation examples. Then click on a tile in the middle panel to load the simulation. Click on the book icon on the left (Example Simulations) panel to view documentation (pdf).

Support
Answers/FAQ
Contact Technical Support
Email - support@tetcos.com

Learn
Videos
Experiment Manual

Documentation
User Manual
Technology Libraries
Source Code Help

Contact Us
Email - sales@tetcos.com
Phone - +91 767 605 4321
Website : www.tetcos.com

Figure 4-20: List of scenarios for the example of Multi hop communication

The NetSim UI would display as shown below.

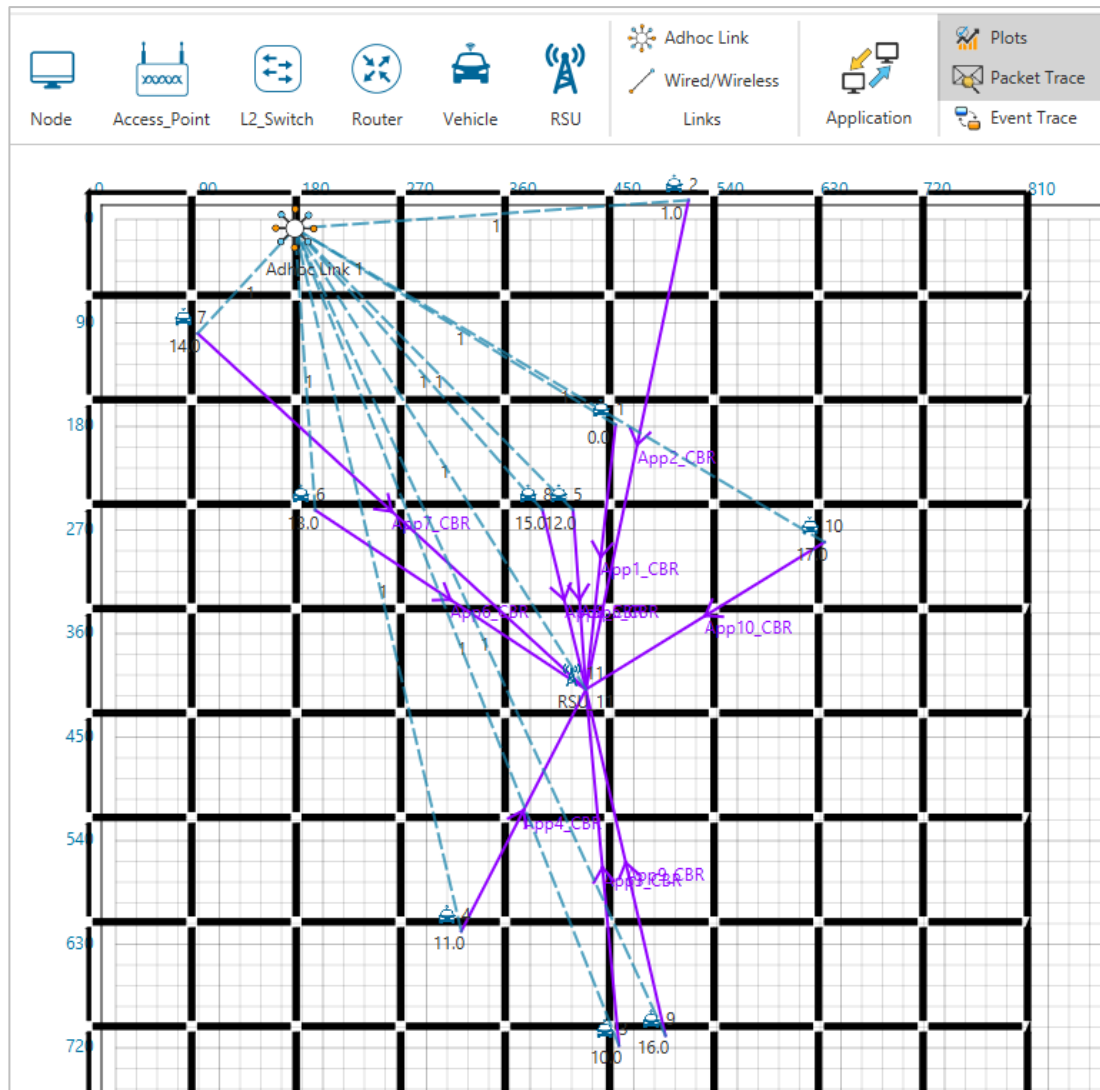


Figure 4-21: Network set up for studying the Multi hop communication

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 4-10: 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>

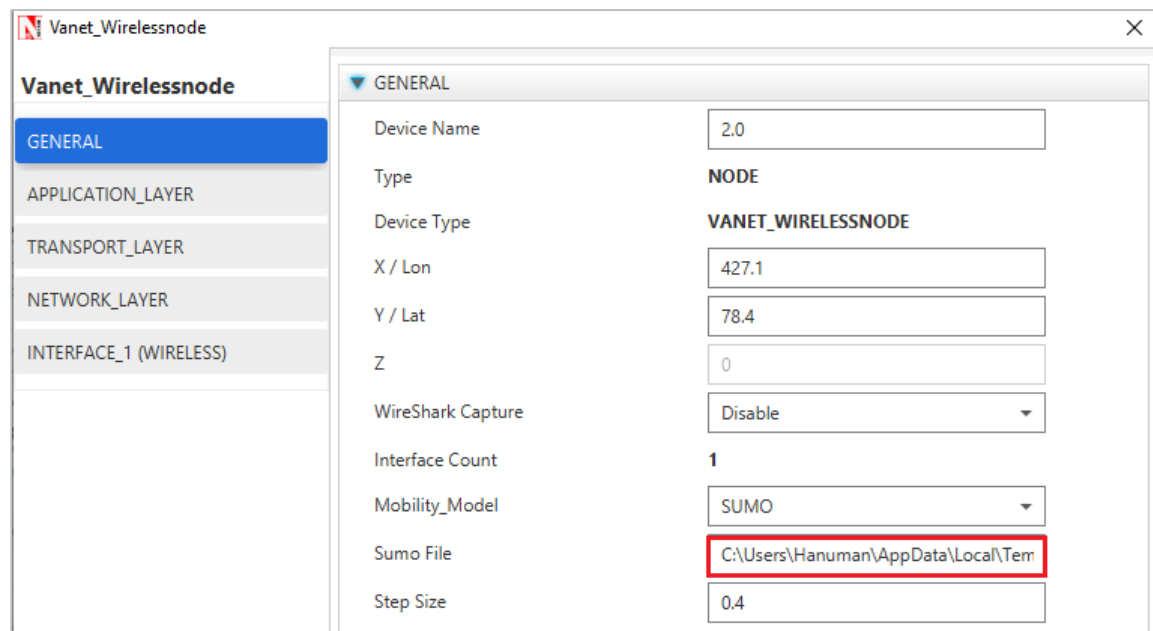


Figure 4-22: General Properties window

3. Transport protocol set as UDP in application Configuration window.
4. Adhoc link/ Wireless link properties set as follows:

Channel Characteristics	Pathloss Model	Pathloss Exponent
Pathloss Only	Log Distance	2.5

Figure 4-23: Wireless link properties

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	1.1826
20	0.5405
30	0.6668
40	1.1857
50	1.0780
60	0.9119
70	0.5720

Table 4-11: Results Comparison

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

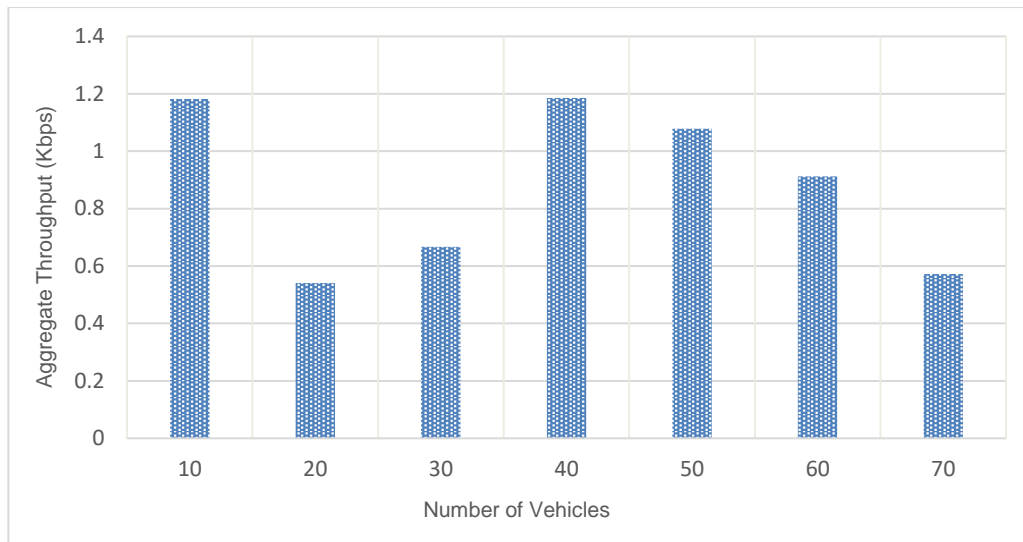


Figure 4-24: Aggregate Throughput vs. Number of Vehicles

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 screenshot

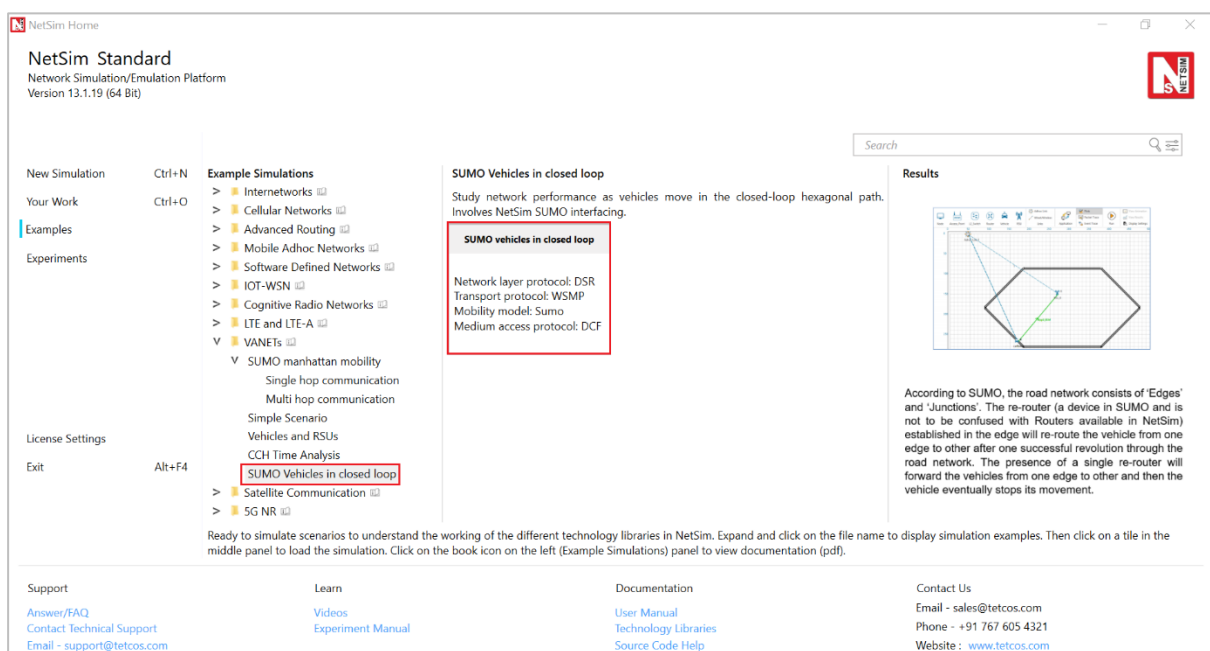


Figure 4-25: List of scenarios for the example of SUMO Vehicles in closed loop

The NetSim UI would display as shown below.

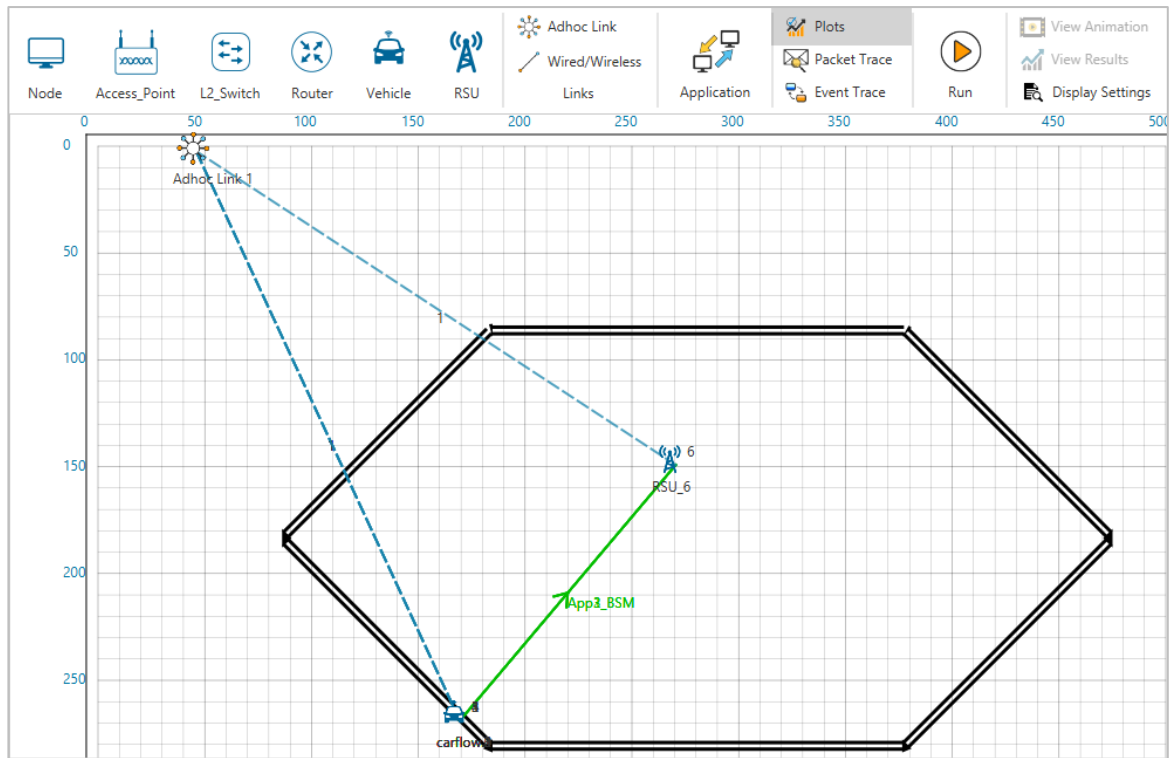


Figure 4-26: Network set up for studying the SUMO Vehicles in closed loop

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 (μ s)
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 4-12: CBR Applications settings

2. Transport protocol set as WSMP for all applications in Application window.
3. In Vehicle General 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 >

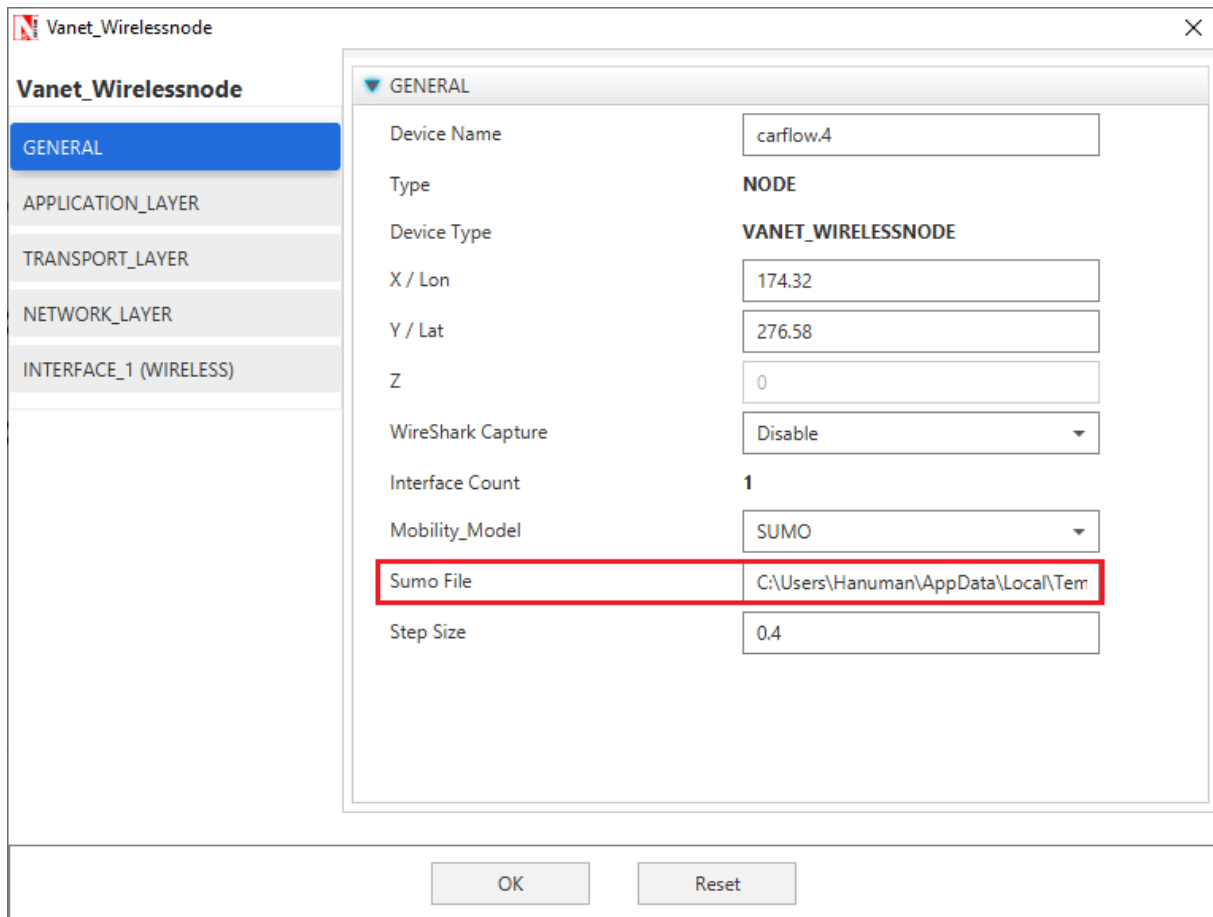


Figure 4-27: General Properties window

4. Adhock link/Wireless link properties were set as follows:

Channel Characteristics	Pathloss Model	Pathloss Exponent
Pathloss_Only	Log_Distance	2

Table 4-13: Wireless link properties

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 and observe the movement of the vehicles in the packet animation window.
9. After the simulation, in NetSim Packet Animation window, we can see that vehicles are moving continuously through the closed-loop hexagonal path till the given end time.

Result:

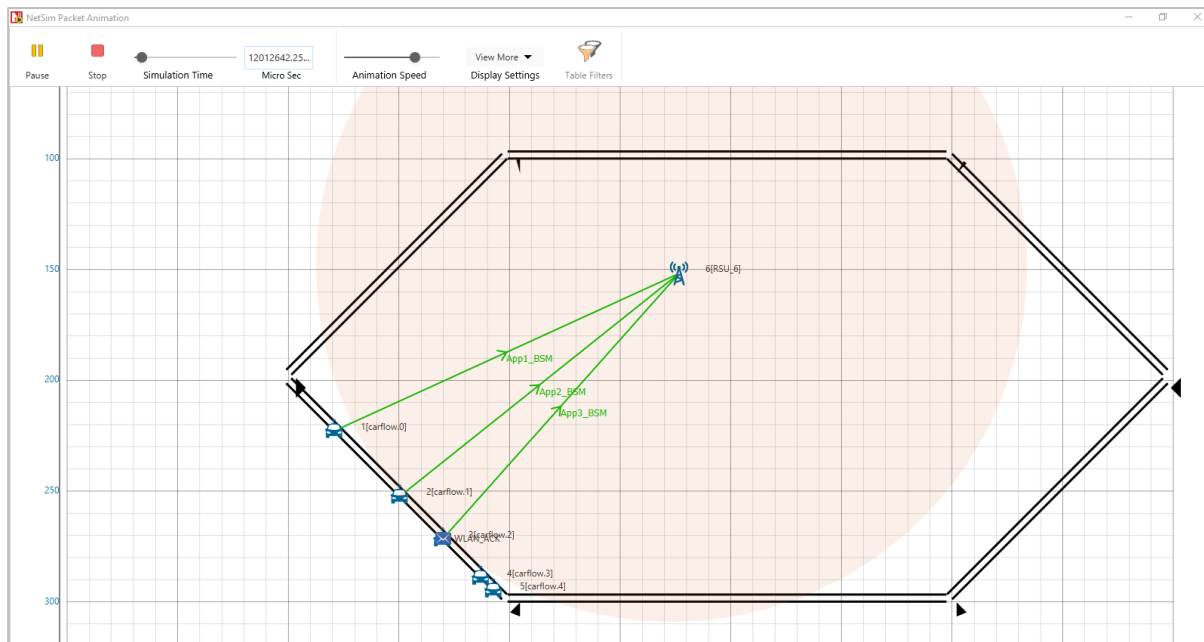


Figure 4-28: Packet animation window for NetSim

With play and record animation enabled, same can be observed in SUMO as well

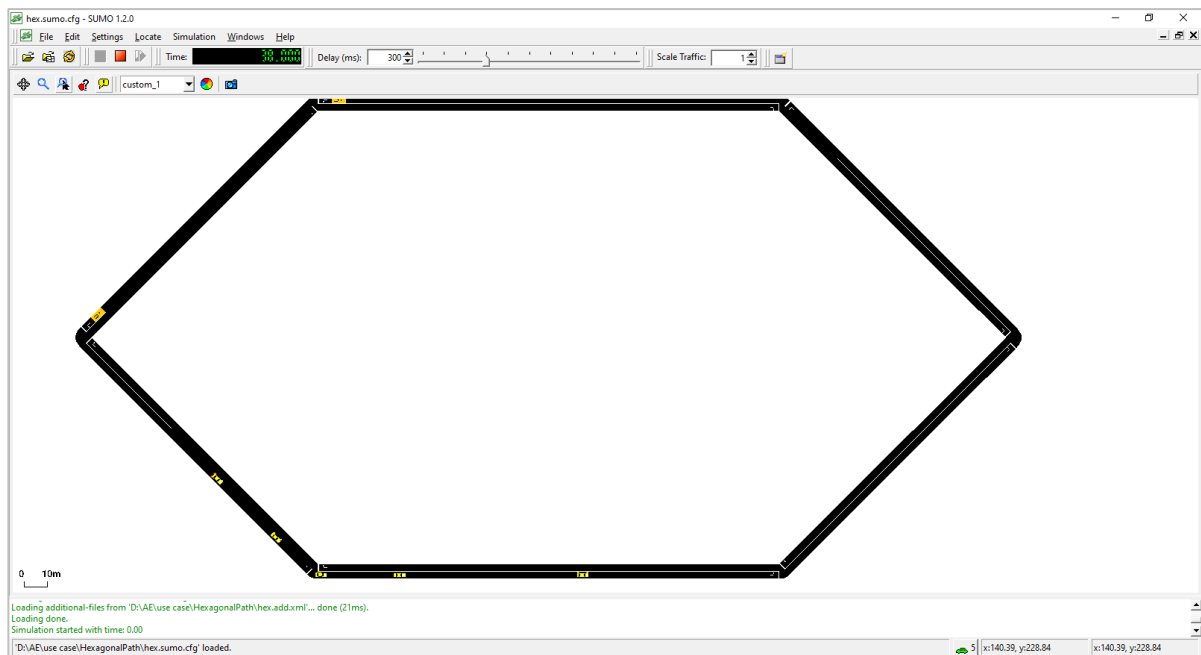


Figure 4-29: Animation window for Sumo

According to SUMO, the road network consist 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-routes 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.