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 13.1 (V), Dec 2021, 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** .................................................................................................................. 4

2 **Simulation GUI** .............................................................................................................. 6  
   2.1 Create Scenario .................................................................................................... 6  
   2.2 Devices specific to NetSim VANETs Library .......................................................... 7  
   2.3 Set Node, Link and Application Properties ............................................................. 7  
   2.4 Enable Packet Trace, Event Trace & Plots (Optional) .......................................... 10  
   2.5 Run Simulation .................................................................................................... 10

3 **Model Features** ........................................................................................................... 13  
   3.1 Implementation of the 802.11p in NetSim ............................................................ 13  
   3.2 DSRC Channels: CCH and SCH ......................................................................... 13  
   3.3 BSM Application .................................................................................................. 14  
   3.4 NetSim – SUMO interfacing ................................................................................. 14  
   3.5 How to create a VANET using SUMO and simulate with NetSim ......................... 14  
       3.5.1 Using SUMO NetEdit utility and randomtrips.py to configure road traffic models ................................................................. 15  
       3.5.2 Creating your own network in SUMO manually ........................................ 22  
   3.6 How to include Roadside Units (RSU's) in a VANET network? ......................... 25

4 **Featured Examples** ..................................................................................................... 28  
   4.1 Simple Scenario .................................................................................................. 28  
   4.2 Vehicles and RSUs ............................................................................................. 30  
   4.3 CCH Time Analysis ............................................................................................. 33  
   4.4 SUMO Manhattan Mobility with Single and Multi-hop Communication .......... 36  
   4.5 SUMO Interfacing with vehicles moving in a closed loop ................................. 44

5 **Reference Documents** ................................................................................................. 48

6 **Latest FAQs** ................................................................................................................ 48
1 Introduction

Note: NetSim VANET component is available only in standard and pro versions

The vehicular communication architecture in NetSim is based on a combination of the IEEE 1609 standard and IEEE 802.11p standard. 802.11p standard defines PHY and MAC layers while upper layers are defined by IEEE1609.

The following are the important features of the NetSim VANET library that are integrated to operate together in a harmonious fashion:

- IEEE 802.11p PHY operating in the 5.9 GHz band with a channel bandwidth of 10 MHz.
- Radio propagation modeling in the PHY layer covering pathloss, shadowing and fading.
- 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 sits on top of 802.11p layers. It enables upper layer operational aspects across multiple channels without knowledge of Physical layer parameters.
- DSRC SAE J2735
- BSM packets are transmitted using WSMP.
- L3 IP routing via DSR, AODV, OLSR or ZRP for non-BSM packets
- Vehicular mobility through in-built mobility models or through 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 information, such as safety warnings and traffic information. 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 between vehicles V2V and between vehicle to infrastructure V2I.
NetSim SUMO interfacing for V2V simulation

Network Communication
- IEEE 802.11p WAVE, ITS, DSRC, BSM
- V2V, V2X Routing
- Channel, RF Propagation
- Applications: Vehicule, V2V, TTP Database, HTTP and more
2 Simulation GUI

2.1 Create Scenario

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

![NetSim Home Screen](image1)

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

![Sumo Configuration File selection](image2)

- 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:** A Vehicle is a 5-layer device that can be connected to an RSU through an Adhoc link. It supports only 1 Wireless interface and has its own IP and MAC Addresses.
- **Roadside Unit (RSU):** A devices which is used for V2I Communication. It is a 5-layer device that can be connected to a Vehicle or to a Router. It supports 1 wireless interface and 1 Serial interface and 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 WiFi protocol. It can be connected to wireless nodes via wireless links and to a router or a switch via a wired link.

![VANET Library Device Palette in GUI](image)

**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](image-url)
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.

![Application Configuration window](image)

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.

![Enable Packet Trace, Event Trace & Plots options](image)

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

  ![Record animation](image)

  **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.

![Vehicle moments](image)

**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

- Adhoc Wi-fi MAC allowing for STA transmission of data frames outside the context of a BSS.
- Immediate communication with no establishment of a BSS. No authentication or 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 - OFDM
- Slot time - 9µs, SIFS - 16µs
- Uses a Medium Access Control (MAC) protocol based on the Carrier Sense Multiple Access Collision Avoidance (CSMA/CA) protocol
  - This means that 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

- Control channel (CCH): A single radio channel, not a service channel, intended for the exchange of management information, including Wireless Access in Vehicular Environments (WAVE), Service Advertisements, and WAVE Short Messages.
- Service channel (SCH): Any channel that is not the control channel, intended for management frames and higher layer information exchanges (Wireless Access in Vehicular Environments [WAVE] Short Message [WSMs]).
- Guard interval: A time interval at the start of each control channel (CCH) interval and service channel (SCH) interval during which devices that are switching channels do not transmit.
3.3 BSM Application

- DSRC protocol runs with BSM (Basic Safety Message) applications
- BSM is a broadcast packet transmitted at a regular interval, and it can be classified as a beacon style transmission.
- 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.
- 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 3-1: SUMO NetEdit Screen](image)

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

**Step 7:** Select "(c) Connect" icon. Select the lanes and ensure connectivity between them.
Figure 3-5: 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 3-6: 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 3-7: 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](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:

```xml
<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.
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.

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 3-10: Importing a sumo network configuration into NetSim
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](https://sumo.dlr.de/wiki/Networks/PlainXML#Node_Descriptions)).
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).

![Figure 3-13: Edge file](image)

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 3-14: Open command prompt in installation directory](image)
**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 3-15: Generating Network file by using NETCONVERT command](image)

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

![Figure 3-16: Direction of the vehicle’s movement](image)
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.

![Figure 3-17: Sumo configuration file](image)

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

RSU’s should be connected using Adhoc links manually.
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 Simple Scenario

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

This scenario involves a simple road traffic network scenario as shown below:
An equivalent network is created in NetSim by importing the SUMO configuration file as shown below:

In NetSim the TCP/IP stack parameters of the devices are configured along with network traffic between vehicles.

![Network setup for studying the Simple Scenario](image1)

After running the simulation, NetSim Animation can be used to visualize packet flow and vehicle mobility along with packet information and throughput plots.

![Packet animation window and link throughput plot](image2)

Simulation results dashboard provides the performance metrics of protocols running in different layers of the network stack of the devices.
4.2 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.

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:

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

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 CCH Time Analysis

Open NetSim and Select Examples->VANETs->CCH Time Analysis then click on the tile in the middle panel to load the example as shown in below screenshot.

Figure 4-11: List of scenarios for the example of CCH Time Analysis

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

Figure 4-12: Network topology in Sumo

An equivalent network is created in NetSim by importing the SUMO configuration file as shown below:
Settings done for this sample experiment:

- In INTERFACE_1(WIRELESS) → Data Link Layer, CCH_TIME = 30000(Microsec)
  
- Set Application BSM (Basic_Safety_Message) with 3Mbps generation rate.
  - Packet size: 100 Bytes
  - IAT: 266 (Microsec)
- 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_v13_1\Docs\Sample_Configuration\VANET\CCH

![General Properties window for vehicle](image)

**Figure 4-15: General Properties window for vehicle**

![CCH Time Analysis Example folder](image)

**Figure 4-16: CCH Time Analysis Example folder**

- Enable Packet trace and plot option.
- Run simulation and note down the number of data packets transmitted (BSM Packets) from Packet Trace by filter PACKET_TYPE to Basic_Safety_Message and throughput.
- For Same Scenario User need to Modify the CCH_TIME to 50000, 70000 microseconds respectively.
- Run the Simulation and Note down the Throughput.
### 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.

#### Table 4-1: Different CCH time vs. Throughputs

<table>
<thead>
<tr>
<th>CCH Time (micro seconds)</th>
<th>Throughput (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30000</td>
<td>0.64</td>
</tr>
<tr>
<td>50000</td>
<td>0.93</td>
</tr>
<tr>
<td>70000</td>
<td>1.15</td>
</tr>
</tbody>
</table>

*Remarks:* The BSM application is transmitted on CCH. Hence as we increase the CCH time, the throughput of the BSM application increases.
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 continuously. 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 4-18: List of scenarios for the example of Single hop communication](image)

The NetSim UI would display as shown below.
Figure 4-19: Network set up for studying the Single hop communication

Settings done for this sample experiment.

1. Applications set as CBR (Broadcast application)

<table>
<thead>
<tr>
<th>Application Method</th>
<th>Application Type</th>
<th>APP_Name</th>
<th>Source_ID</th>
<th>Destination_ID</th>
<th>Packet_Size (Bytes)</th>
<th>Inter-Arrival Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast</td>
<td>CBR</td>
<td>APP_1_CBR_Broadcast</td>
<td>21</td>
<td>Broadcast to all vehicles</td>
<td>300</td>
<td>2,000,000</td>
</tr>
</tbody>
</table>

   Table 4-2: CBR Applications Settings

2. Transport protocol set as UDP in application Configuring window.

3. Adhoc link/Wireless link properties were set as follows:

<table>
<thead>
<tr>
<th>Channel characteristics</th>
<th>Pathloss Model</th>
<th>Pathloss Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pathloss Only</td>
<td>Log Distance</td>
<td>2.5</td>
</tr>
</tbody>
</table>

   Table 4-3: Wireless link properties

4. Plots and packet trace are enabled and run simulation and observe the movement of the vehicles in the packet animation window.

5. In Netsim packet animation window, you can see that vehicles choose random directions when they reach a junction in the Manhattan grid network.

6. 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.
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)

<table>
<thead>
<tr>
<th>Environment</th>
<th>Path-loss Exponent</th>
<th>Packets Received (Aggregate)*</th>
<th>Throughput (Kbps) (Aggregate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Rural Area</td>
<td>2.5</td>
<td>1000</td>
<td>24</td>
</tr>
<tr>
<td>Urban Area</td>
<td>3</td>
<td>232</td>
<td>5.59</td>
</tr>
<tr>
<td>Dense Urban Area</td>
<td>3.5</td>
<td>100</td>
<td>2.4</td>
</tr>
</tbody>
</table>
### Table 4-4: 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)

<table>
<thead>
<tr>
<th>Environment</th>
<th>Path-loss Exponent</th>
<th>Packets Received (Aggregate)</th>
<th>Throughput (Kbps) (Aggregate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Rural Area</td>
<td>2.5</td>
<td>1980</td>
<td>47.52</td>
</tr>
<tr>
<td>Urban Area</td>
<td>3</td>
<td>463</td>
<td>11.11</td>
</tr>
<tr>
<td>Dense Urban Area</td>
<td>3.5</td>
<td>197</td>
<td>4.72</td>
</tr>
<tr>
<td>Very Dense Urban Area with Shadowing</td>
<td>4</td>
<td>79</td>
<td>1.89</td>
</tr>
</tbody>
</table>

### Table 4-5: Results Comparison for RSU Broadcast Data Rate = 2.4 Kbps

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

<table>
<thead>
<tr>
<th>Environment</th>
<th>Path-loss Exponent</th>
<th>Packets Received (Aggregate)</th>
<th>Throughput (Kbps) (Aggregate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Rural Area</td>
<td>2.5</td>
<td>7980</td>
<td>191.52</td>
</tr>
<tr>
<td>Urban Area</td>
<td>3</td>
<td>1857</td>
<td>44.56</td>
</tr>
<tr>
<td>Dense Urban Area</td>
<td>3.5</td>
<td>791</td>
<td>18.98</td>
</tr>
<tr>
<td>Very Dense Urban Area with Shadowing</td>
<td>4</td>
<td>316</td>
<td>7.58</td>
</tr>
</tbody>
</table>

### Figure 4-22: 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

![Image](image_url)

Figure 4-23: List of scenarios for the example of Multi hop communication

The NetSim UI would display as shown below.
**Figure 4-24: Network set up for studying the Multi hop communication**

**Settings done for this sample experiment.**

1. Applications set as CBR.

<table>
<thead>
<tr>
<th>Application Method</th>
<th>Application Type</th>
<th>Source_Id</th>
<th>Destination_Id</th>
<th>Packet Size (Bytes)</th>
<th>Inter-Arrival Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CBR</td>
<td>(All vehicles)</td>
<td>RSU</td>
<td>1460</td>
<td>20,000</td>
</tr>
</tbody>
</table>

Table 4-7: 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>
3. Transport protocol set as UDP in application Configuration window.

4. Adhoc link/ Wireless link properties set as follows:

<table>
<thead>
<tr>
<th>Channel Characteristics</th>
<th>Pathloss Model</th>
<th>Pathloss Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pathloss_Only</td>
<td>Log_Distance</td>
<td>2.5</td>
</tr>
</tbody>
</table>

5. Co-ordinates of RSU are set as X = 450, and Y = 450

6. Network layer routing protocol is set as DSR.

7. Plots are enabled and run the simulation.

8. Increase the number of vehicles in the order 10, 20, 30 etc. and note down the aggregate throughput.

Result:

<table>
<thead>
<tr>
<th>Number of vehicles</th>
<th>Throughput (Kbps) (Aggregate)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4098.74</td>
</tr>
<tr>
<td>20</td>
<td>4037.54</td>
</tr>
<tr>
<td>30</td>
<td>4160.18</td>
</tr>
<tr>
<td>40</td>
<td>4454.28</td>
</tr>
<tr>
<td>50</td>
<td>5154.85</td>
</tr>
<tr>
<td>60</td>
<td>3988.72</td>
</tr>
<tr>
<td>70</td>
<td>4403.94</td>
</tr>
</tbody>
</table>

Table 4-8: Results Comparison
*Aggregate is the sum of the packet_throughputs obtained by all applications.

**Figure 4-27: 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 4-28: List of scenarios for the example of SUMO Vehicles in closed loop**

The NetSim UI would display as shown below.
Settings done for this sample experiment:

1. Applications set as BSM (Basic_Safety_Message)

<table>
<thead>
<tr>
<th>APP_ID</th>
<th>Source_ID</th>
<th>Destination_ID</th>
<th>Packet_Size (Bytes)</th>
<th>Inter-Arrival Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>APP_1_BSM</td>
<td>1</td>
<td>6 (RSU)</td>
<td>100</td>
<td>2,000,000</td>
</tr>
<tr>
<td>APP_2_BSM</td>
<td>2</td>
<td>6 (RSU)</td>
<td>100</td>
<td>2,000,000</td>
</tr>
<tr>
<td>APP_3_BSM</td>
<td>3</td>
<td>6 (RSU)</td>
<td>100</td>
<td>2,000,000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Channel Characteristics</th>
<th>Pathloss Model</th>
<th>Pathloss Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pathloss Only</td>
<td>Log_Distance</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4-10: Wireless link properties

5. Medium access protocol set as DCF in all vehicles and RSU.

6. Enable Plot and Run simulation and observe the movement of the vehicles in the packet animation window.

7. 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 4-31: Packet animation window for NetSim

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

Figure 4-32: 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

- 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