POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Simulation of Vehicle-to-Everything communications based on Geometrical Channel Modelling using GEMV^2

TESI DI LAUREA MAGISTRALE IN
TELECOMMUNICATION ENGINEERING in DATA
COMMUNICATIONS

Author: **Anirban Das**

Student ID:	914638
Advisor:	Maurizio Magarini
Co-advisor:	Francesco Linsalata
Academic Year:	2020-21

Contents

List of Abbreviations

- LOS - Line of Sight
- NLOSv - Non-Line of Sight due to vehicles
- NLOSb - Non-Line of Sight due to buildings
- NLOSf -Non-Line of Sight due to foliage
- DSRC - Dedicated Short-Range Communications
- ITS - Intelligent Transportation System
- VANET - Vehicular Ad-hoc Network
- V2V - Vehicle-to-Vehicle
- V2I - Vehicle-to-Infrastructure
- V2X - Vehicle-to-Everything
- RSU - Roadside Unit
- SUMO – Simulation of Urban Mobility
- 3GPP – 3rd Generation Partnership Project
- SC-FDMA – Single Carrier Frequency Division Multiple Access
- OFDMA – Orthogonal Frequency Division Multiple Access
- DMRS – Demodulation Reference Signal
- CSMA/CA – Carrier Sensing Multiple Access with Collision Avoidance
- DCF – Distributed Coordination Function

- PCF – Point Coordination Function
- LAN – Local Area Network
- WAN – Wide Area Network
- WPAN – Wireless Personal Area Network

List of Figures

Acknowledgements

Having navigated to the completion of my Masters Degree, a journey which had started three years prior, it is the biggest sense of relief and purest form of joy to have arrived at this point in my career, and I would like to extend gratitude and thanks to my supervisor and teacher, Professor Maurizio Magarini who was patient with me throughout my thesis period. I am grateful for his suggestions, knowledge, and inputs not only on my thesis but on a personal level as well, during difficult times.

I would also like to thank the PhD students, Dr. Francesco Linsalata, Dr. Mehdi Haghshenas and Dr. Eugenio Moro, who provided their time, suggestions, and valuable inputs whenever I was unable to go on and needed guidance. I would also like to thank all the other professors of my course at Politecnico under whom I was fortunate to learn and take classes. Finally, I am grateful to have the opportunity to have been a student of Politecnico di Milano and I hope to become an ambassador for this institution.

I would also like to thank my parents who support and encourage me right from my beginning of my master’s to this point and continue to do so. Sincere gratitude and love to all my friends for supporting me in this whole career and always motivating me to do better and being always available for any help.

I will remember all these memories of the past three years with fondness and carry them with me for the future that awaits. So much more to come, hopefully!!!

Abstract

There has been a lot of interest in the past decade on technologies which involve the automotive industry in terms of Vehicle-to-Vehicle Communications. With respect to the vehicular industry, there has been a development of IoV (Internet of Vehicles), which is another term given for a distributed network that supports the use of data created by connected cars and vehicular ad hoc networks (VANETs). The use cases of such applications include road safety, dynamic navigation and traffic information, autonomous driving, fleet management, as well as providing mobile and location-based services in the infotainment domain.

It started with the IEEE having a standard for such communications called DSRC (Dedicated Short-Range Communication), which is also known as IEEE 802.11p, which supported the concept of VANET. These DSRC devices operate in the sub-6GHz range (at the 5.85-5.925 GHz band), and the standard was specifically developed for vehicular communications. Consequently, in the past few years another vehicular communication technology was developed called C-V2X (Cellular Vehicle-to-Everything), which considered the drawbacks of DSRC and made improvements to better the communication.

The primary work of the thesis is to understand the various real-world conditions that might occur in such communications and whether it is feasible to deploy such in large scale scenarios. Typically, in such communications, we understand that the nodes will be dynamic and there are scenarios where the vehicles go through in terms of Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) conditions, and thus affect the signal levels in such cases. Reliability, latency are the keywords in such critical transfer of messages. Hence, we also make a note of the SINR (Signal-to-Noise and Interference Ratio) and the PER (Packet-Error-Rates) while making the simulations.

Keywords: *V2X; Vehicular Communications; VANET.*

Abstract in lingua italiana

L’interesse nell'ultimo decennio sulle tecnologie che coinvolgono l'industria automobilistica è aumentato in termini di comunicazioni da veicolo a veicolo. Per quanto riguarda l'industria automobilstica, vi è stato anche uno sviluppo di IoV (Internet of Vehicles), che è un altro termine utilizzato per descrivere una rete di distribuzione che supporta l'uso di dati creati da auto connesse e VANET (Vehicular Ad hoc Network). I casi d'uso di tali applicazioni includono la sicurezza stradale, la navigazione dinamica e le informazioni sul traffico, la guida autonoma, la gestione della flotta, nonché la fornitura di servizi mobili e basati sulla posizione nel dominio dell'infotainment.

L'IEEE ha sviluppato un primo standard per tali comunicazioni chiamato DSRC (Dedicated Short-Range Communication), noto anche come IEEE 802.11p, che supportava il concetto di VANET. Questi dispositivi DSRC operano nella gamma di frequenza sub-6GHz (alla banda 5,85-5,925 GHz) e lo standard è stato sviluppato specificamente per le comunicazioni automobilistiche. Negli ultimi anni è stata sviluppata anche un'altra tecnologia di comunicazione automobilistica chiamata C-V2X (Cellular Vehicle-to-Everything) che, a partire da gli svantaggi del DSRC ha apportato miglioramenti per migliorare la comunicazione.

Il contributo principale della tesi riguarda la comprensione dei differenti scenari propri del mondo reale che potrebbero verificarsi in tali comunicazioni e se possa essere possibile implementarle in scenari su larga scala. In genere, in tali comunicazioni, comprendiamo che i nodi saranno dinamici dove i veicoli attraverseranno in termini di condizioni di LOS(Line-of-Sight) e NLOS (Non-Line-of-Site) che potrebbero influenzare i livelli del segnale in questi casi. Affidabilità, latenza sono le parole chiave in una trasmissione così critica di messaggi e quindi si valuteranno le prestazioni in termini di SINR (Signal-to-Noise and Interference Ratio) e PER (Packet-Error-Rates) mediante le simulazioni. Il simulatore utilizzato per eseguire queste analisi è OMNET++. Insieme ad esso, utilizziamo diversi moduli aggiuntivi tra cui INET e Artery che sono stati spiegati molto più avanti nel progetto.

Parole chiave: *V2X; Vehicular Communications; VANET*.

1. Introduction

1.1 Background

Vehicular communications have become more and more useful in the present days since their advent with DSRC, which was the only present standard earlier. But 3GPP has been considering a newer version of the technology named Vehicle-to-Everything (V2X) designed specifically for vehicular communications. This technology is more commonly known as LTE-V2V, which had few modifications in the subframe and PHY and MAC layers to support better mobility scenarios as should be considering the use cases of this. What is most considered is a scenario of cooperative awareness in which vehicles periodically send broadcast messages, or basic safety messages and are generically called beacons to inform other vehicles around about their position and movements. This information can be used for a variety of scenarios regarding accident prevention, cooperative driving or perform geographical routing towards intended destinations.

1.2 Overview

To better understand how vehicular communication can be put into practice, the concept of resource allocation is one of the primary issues which is being investigated. Since resources are limited, we need to make sure resources are utilized maximally and fairly by all vehicles present in the scenario without interference between the signals. In this work, we use the simulator GEMV^2 which uses messages to communicate between modules and focusses on resource allocation for cooperative awareness service, and we have several modifiable input parameters for example the propagation settings, the MCS and vehicular traffic. So, what we can do is either to adopt simple mobility models of traffic or create traffic traces (i.e., having input files describing the realistic position of vehicles in time). From all these inputs, the simulator calculates the blocking rate, packet error rate and rate of radio resource re-assignment.

2. Resource Allocation

2.1 LTE-V2V

V2V communications was first standardized in LTE Release 14. It is based on its uplink PHY and MAC and uses OFDM at the PHY layer and SC-FDMA at MAC layer. Typically, a dedicated carrier is used with GNSS for time synchronization. The frame structure is as follows – in the time domain the signal into frames of 10ms with each frame consisting of 10 subframes of 1ms and they in turn are composed of 2 slots. In the frequency domain, the signal is divided into several sub carriers – 12 subcarriers in frequency (180 kHz), which are spaced at 15 KHz, and one slot in time (0.5ms) and they form the resource block (RB), and the number of RB depends on the bandwidth and configuration of the network.

Moreover, to adapt to highly dynamic scenarios, LTE-V2V modifies the LTE subframe. Specifically, in addition to the last one of 14 OFDM symbols in the subframe left unused to allow for timing adjustment and Tx-Rx turnaround, and the 2 DMRS symbols, an additional 2 DMRS symbols have been added to handle the high Doppler with higher relative speeds. So, in the newer frame structure only 9 OFDM symbols per subframe are left for the data.

LTE-V2V communications are possible both under centralized control of the network and in a distributed way. In centralized control called Sidelink Mode 3, vehicles should be under cellular coverage and resource allocation is decided and communicated through a centralized entity through the eNodeB. In Sidelink mode 4 allocation happens in an autonomous manner and V2V links are given dedicated resources and vehicles autonomously select those using local information and which will provide the best performance.

However, in the simulator that we have, we investigate the resource allocation for cooperative awareness service. So, it is assumed that vehicles periodically broadcast beacon messages that address all the neighbours that are within a selected distance called the awareness range that are denoted as raw. So, whenever a group of RB's is obtained or selected for the transmission of beacons from vehicles, the same RBs are periodically reserved for the same transmission until a collision is detected.

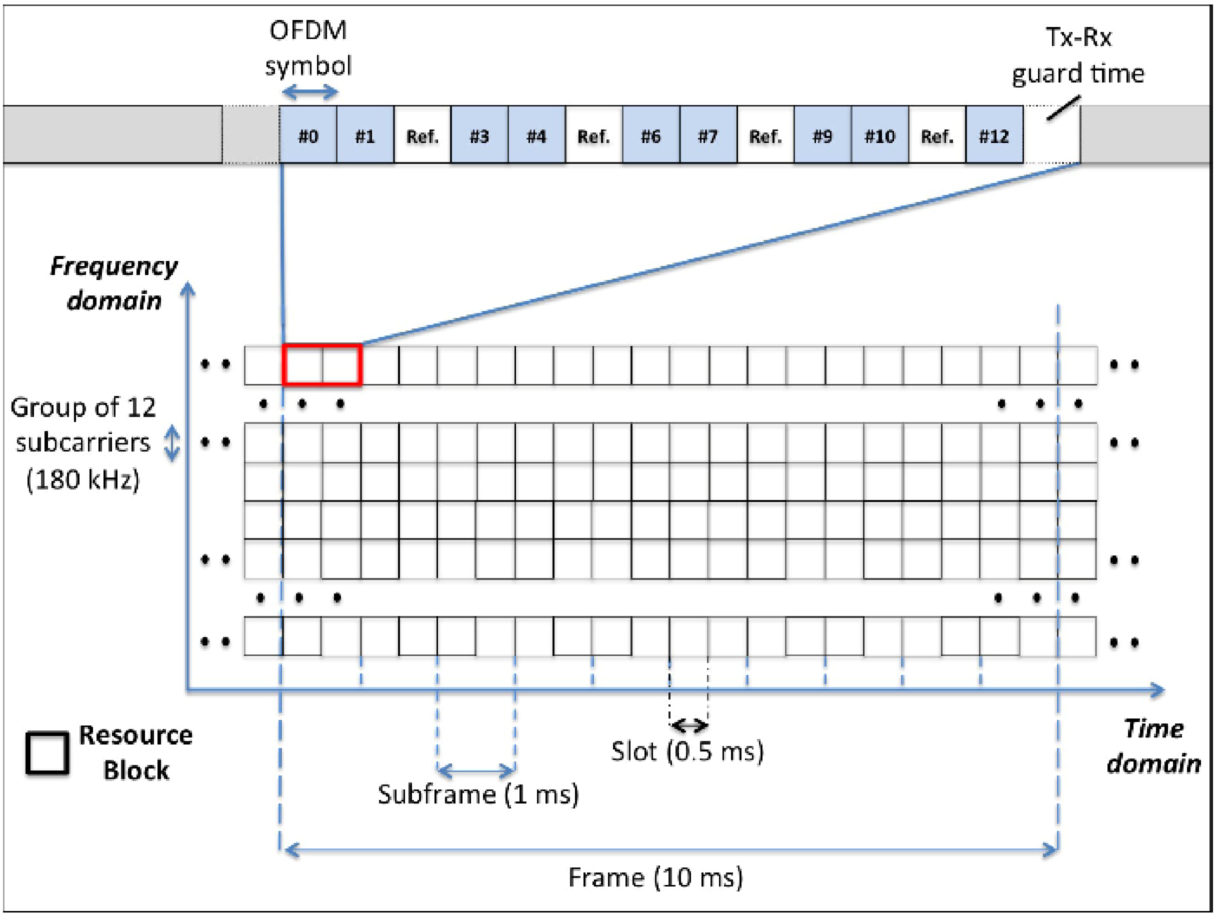


Figure 2.1 : Frame Structure of LTE-V2V communication

2.2 802.11p

For DSRC, the resource allocation has mostly been on the MAC layer and sometimes on the channel allocation or rate allocation techniques. All vehicles have similar MAC parameters in a scenario in DSRC by default and hence should have equal opportunity for network access. Now, we understand that the speed of a vehicle maybe a significant parameter in terms of network performance of the vehicle in a specific scenario (for e.g., a fast-moving vehicle will have poorer performance than a slow-moving one, as the slow vehicle can better communicate with nodes around it). So, several researchers have proposed that the speed of the vehicle be taken into consideration when setting the contention window and such dynamic contention windows can improve the fairness between these two categories of vehicles.[10]

DSRC also uses orthogonal frequency bands to support multi-channel operation to make emergency messages to be processed with high priority, ultra-reliability and with minimal latency. It also supports the usage of multiple MCS to support a wide range of data rates and this is good because a constant MCS might not be proper for dynamic and diverse environments.

2.3 Research Problem

For the past few years, vehicular communications being looked at as the concept of smart vehicles are coming into plays. Most automobile manufacturers have started producing several lines of vehicles capable of smart communication. However, they have not been done on a large scale on concerns of practical implementation, cost and impracticality with the sites chosen for analysis.

The main goal of the thesis is to simulate different scenarios and traffic conditions, that can give a fair account of what might happen in a real-world scenario in terms of how messages might be transmitted from one node to another to facilitate vehicular communications and analyze the performance of this with respect to SNIR (Signal to Noise and Interference Ratio) and PER (Packet Error Rates).

2.4 Research Methodology

We start investigating this topic by understanding the theoretical concepts involved in the development of such technology and then we resort to the use of a ray-tracing software called GEMV^2 to understand how the dynamic vehicular scenarios are analyzed in terms of performance parameters like SNIR and PER, as mentioned earlier . GEMV^2 is a geometry-based, efficient propagation model for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, and it models outline of objects and use them to differentiate traffic conditions (explained more in a separate chapter, later). We will see how the tool works, what the user sees on the screen when the simulation is running, the modules and header files that are part of the simulation process and we will have a peek into the code. We will also analyze the results and make conclusions of them and talk about the advantages and disadvantages of such a tool.

The tool is run on a discrete event simulator called OMNeT++, which provides all the tools and modules necessary to design and analyze real world scenario. It is coded in C++ and uses a message system to communicate with different modules. It has several use cases which include network protocols, queuing systems, multiprocessor systems with parallel tasks etc. OMNeT++ will be investigated in more detail later in this work and can be used to analyze complex scenarios such as that of vehicular communications where different agents and objects are involved.

3. Technologies for Vehicular Communications

After having a brief introduction on the technologies in the previous chapter we investigate more in depth some of the technologies that were developed for vehicular communications. Typically, we will see that IEEE has made a standard specifically to support such communications. We start by illustrating IEEE protocols designed for vehicular communications.

3.1 IEEE 802.11p

From the very onset of the concept of Wireless Networking/Communication the IEEE began naming the protocol with '802' irrespective of the kind of network it was, be it a LAN, WPAN, WAN etc.

To that respect, we initially focus on the '.11p' implementation of the Wireless Communication that is part of the WLAN standards (802.11xx). Then we go deeper to all the specifics that enhance Vehicular Communication, which is the concept of intelligent vehicles(standard IEEE 802.11p). This is also called Wireless Access in Vehicular Environment(WAVE), which is more of a standardization process originating from the DSRC (Dedicated Short-Range Communication) spectrum band in the US and adapted to European standards. IEEE 802.11p (also known as IEEE 802.11p WAVE) form the basis of the DSRC communication, which not only includes vehicle to vehicle transmissions but also vehicle to infrastructure/pedestrians etc. and include other roadside units.

3.1.1 Technical Specifications

We will investigate the IEEE 802.11p standard in a little detail to understand how the protocol is modelled in layers. The 802.11p physical layer is based on Orthogonal Frequency Division Multiplexing (OFDM).

The IEEE 802.11p MAC layer lacks some typical features of the stack which include authentication and association mechanisms. It uses EDCA to have different priorities to different classes of data according to QoS requested. The frequency range of operation of the standard is in the 5.850 to 5.925 GHz which is part of the ISM band and is divided into channels of 10 MHz each. The division is based on Distributed Coordination Function (DCF) which makes use of the CSMA/CA algorithm with interframe spacings. These spacings are used to sense the channel before initiating a transmission and the transmission goes through if the channel is free. If the channel is busy, we wait for certain intervals of time before trying a transmission again and this process continues till the channel is free or until a certain number of transmission attempts have been made.

The spectrum of DSRC is allocated in the upper 5 GHz range, and it is divided into sub-slots: two slots for service channels, one for accident avoidance, one for Controlling and the other for high power and long-range transmission. However, it is not a duplex channel and hence cannot be used simultaneously, so they switch between the Control channel and one of the Service or Safety channels.

Depending on the kind of messages transmitted, there are two – Cooperative Awareness Message (CAM) of Decentralized Event Notification Message (DENM). 802.11p is limited to the two lower layers of the stack i.e., the PHY and MAC layer and it adds features relevant to the V2X communications to the stack and integrates with it.

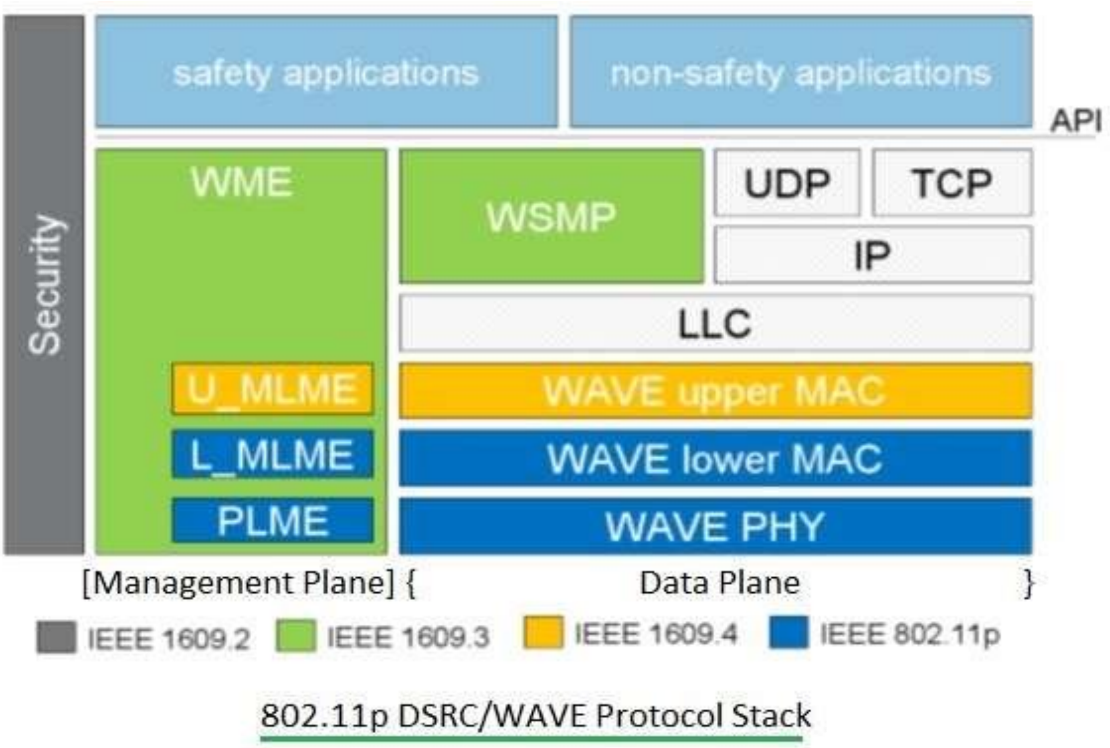


Figure 3.1 : Communication Stack for vehicular communications (Source IEEE 802.11p : Towards an International Standard for Wireless Access in Vehicular Environments by Daniel Jiang and Luca Delgrossi) [16]

3.2 Intelligent Transportation System

This is used to refer to the Intelligent Transport System, which is coming more into play along with the concept of smart cities and smart vehicles as such. It can be considered as a part of IoT or Internet of Vehicles which is being implemented for road safety purposes along with fleet management, assisted driving etc. to make our lives easier and enabling digital transformation. The goal of such a project is to achieve efficiency in traffic systems by broadcasting information through the whole traffic network enabling people to make the best choice on the information received.

This is however not limited to traffic control applications but can be extended to road safety and efficient infrastructure usage. The European Telecommunication Standards Institute group for Intelligent Transport Systems are trying to bring a unified intelligent system of transport to better manage resources on the road. Depending on the technologies applied from satellite navigation, traffic light infrastructures, cameras up to real-time data from various external sources such as information from infrastructures around the region where the vehicle is operating, there could be a wide range of use cases that can be supported.

Some of the important applications could be:

- **Traffic Management and Control:** which leverage data on vehicle stops, traffic viability, traffic signals monitoring, payments.
- **Vehicle Safety and Control:** which includes collision avoidance systems, intelligent cruise control systems, vehicle monitoring, automated driving and autonomous or assisted driving systems.
- **Fleet Management:** logistics support, cargo management systems, vehicle management.
- **Public Transport Management:** Localization, Triangulation, Integrated payment systems.

ITS is becoming an important discipline and organizations are looking into investing heavily in it be it smart vehicles or infrastructure or modules which assist in vehicular communications and such. A centralized traffic management centre is important in this architecture where data is collected and analysed for predictions related to traffic density based on times of the day, location-wise etc. There are a lot of things that goes into this process which includes:

- **Data Collection:** A lot of data is collated via sensors in vehicles and other RSU's, cameras, GPS and location data that might be analysed in the centralized traffic management centre. The data might include traffic count, surveillance, density of vehicles, locations.
- **Data Analysis:** The collected data is analysed which might reveal traffic patterns and may be used for predictive analysis too. This can be sent as an on-demand output to resources that ask for it or be sent periodically.
- **Data Transmission:** As mentioned above the output of any analysed data is sent in real time either on demand or periodically to units requesting for such data which might include vehicles, pedestrians etc.
- **Traveller Information:** Travel Advisory Systems (TAS) is used to inform about transportation updates to the user like traffic conditions, diversions and roadblocks, delays, accidents on the road etc and other such information using modes of delivery that might be through the internet or other modes.

3.3 Messages for Communication

The communication between vehicles or infrastructure takes place through messages that are transferred from one node to another. These messages can be about vehicular positions, speed, acceleration, general traffic conditions, or of the surroundings etc.

We can divide these into 2 types of messages:

- **Cooperative Awareness Message (CAM):** which is a periodic message broadcasted by a node to all other nearby nodes be it single/multi-hop which transmits a message which might be when there is some change in the vehicle or traffic characteristics, say a vehicle has changed lanes or changed speed. So, the receiving nodes are made aware of such changes, and they may use this information to make any changes to themselves, if necessary. The message will contain some information like: ID of the transmitting node, position of the node sending the message, and other data as it comes from the standard.

- **Decentralized Environmental Notification Message (DENM):** which is generated every time the node needs to notify of a relevant event, say a security situation or say the transit of an emergency vehicle. The message is valid till the event is valid. Consequently, this message could be stored for notifying nodes which enter the area later where the event had occurred. The event may be updated, and the new updates maybe handled and disseminated by a new node, and the new node decides the location and the frequency at which the DENM message should be forwarded.

We will see that the CAM is associated with a cellular network i.e., it uses a cellular network to be broadcasted while the DENM message is a D2D broadcast. Being a D2D broadcast, it is understood that these have a higher priority w.r.t CAM messages due to their safety applications in the scenario.

3.4 The ETSI ITS – G5 standard

The ETSI ITS-G5 standard was crucial to the development of the ITS communication-the backbone of which was based upon that every device in range can communicate with each other which might be a mobile device, or it might be an infrastructure. There are primarily two technologies developed for the Intelligent Transport systems: DSRC (Dedicated Short-Range Communication) developed in the US while the same concept for the European version was called ETSI (European Telecommunication Standard Institute). Both the specifications are leveraged from the 802.11p protocol and other technologies can augment the stack with their own technologies say, Cellular 5G and LTE technologies.

ETSI ITS-G5 is an extension of the Wi-Fi standard 802.11p specifically for vehicular environments in terms of the optimization made for dynamic automotive environments some of which includes the safety messages delivered in the 5.9 GHz channel which is specifically allocated for vehicular communication applications. The allocation was initially done by the FCC and the European Commission later allocated the same for transport safety applications. The protocol is built on existing protocol stack allocated for vehicular communications.

Cooperative-ITS(C-ITS) is a variation of ETSI ITS-G5 that has a wide range of tools to support the connectivity of vehicles, roadside infrastructure, traffic signals as well as with other road users which enables the reliable and efficient inter-vehicle communications. We also have other specifications that we find in the Control Channel and Service Channel inside the device enabled for ITS-G5 technology and that the device operates outside the context of a Basic Service Set (BSS), unlike that of legacy devices.

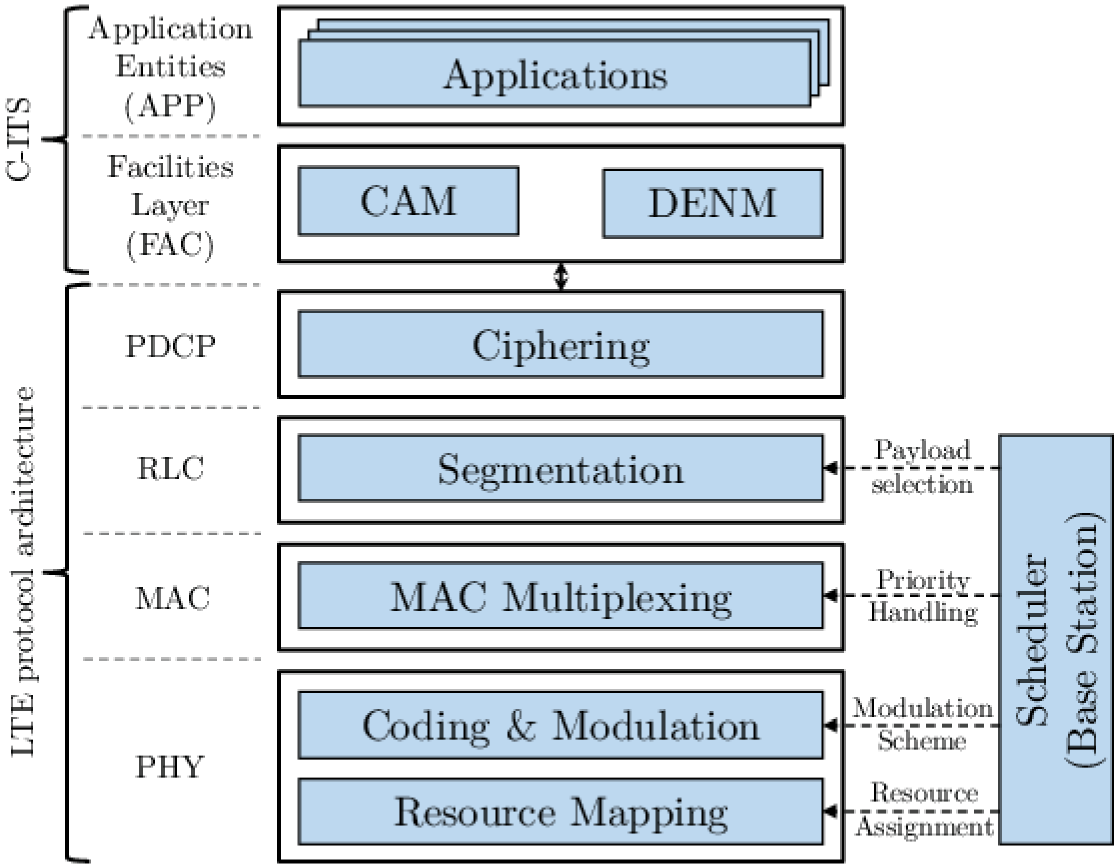


Figure 3.4 : C-ITS communication stack

3.5 Device-to-Device Technology

Device to Device communication signifies that it is not part of a PCF (Point Coordination Function) in the sense that there is not a centralized entity which is responsible for managing the communication infrastructure. Instead, it uses a DCF (Distributed Coordination Function) as there is no network infrastructure involved in the communication, such as an access point or a base station. What occurs is the devices in proximity communicate with each other over a direct link. This enables reduction in time compared to conventional methods thus enabling ultra-reliable low latency communication with increased throughput, low latency and is energy efficient.

With regards to our discussion, the devices are vehicles which are connected to a network which can be the 5G wireless network which has specific use-cases for such kind of communication with short-range wireless technologies (like Wi-Fi Direct, LTE Direct or Bluetooth) and vehicle-to-everything protocols can be used to enable D2D communications. Like the boom of the IoE market, we can consider a vehicle to be a node in IoT communications. There is also an evolving term called IoV (Internet of Vehicles) which is when a vehicle when connected to a network can advertise nearby infrastructure, vehicles, and humans before it changes any of its parameters like acceleration, or before a crossing etc.

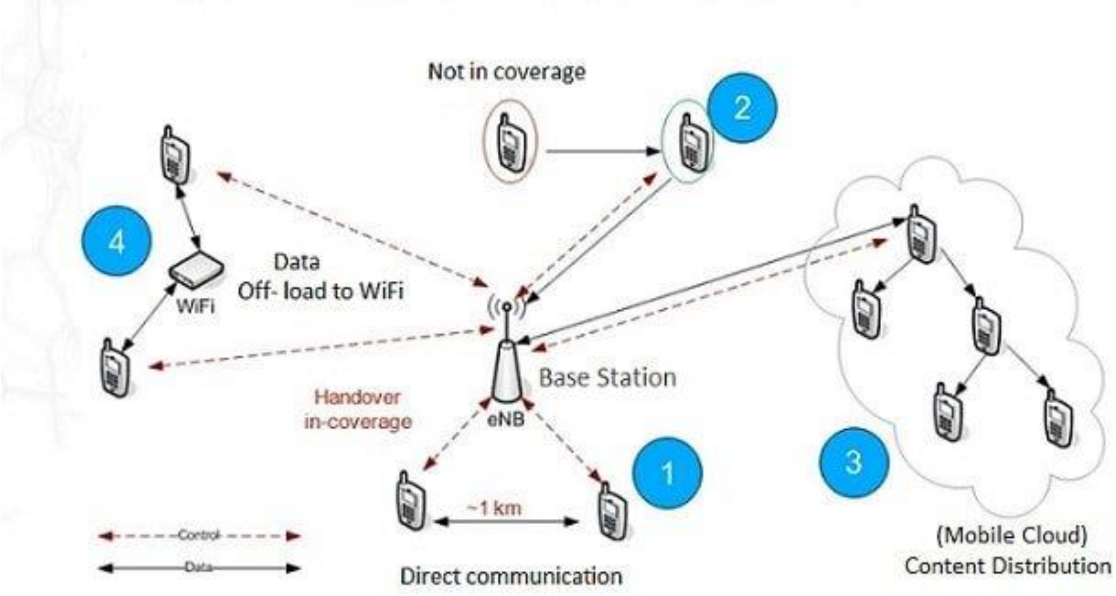


Figure 3.5 : Cellular communication and D2D communication in both single and multi-hop scenarios

The 5G network is a major gamechanger with respect to supporting D2D communications as they support MIMO antennas, heterogenous networks, beamforming which will increase the spectral efficiency and energy efficiency and reduce end-to-end latency. D2D is a part of Dedicated Short-Range Communications (DSRC) which is a Wi-Fi implementation of standard Wi-Fi specification 802.11p, which was the traditional and first standard used in D2D communications. However, to mitigate the issues prevalent in DSRC we have newer standards one of which is Cellular-Vehicle-to-Everything communications (C-V2X). C-V2X combines D2D communication with 5G and is also referred to as LTE-V2X.

The advantages of implementing D2D communications is:

(a) **Resource allocation:** to maintain the direct link in the D2D pairs as the protocol is provided by D2D

(b) **Synchronization:** time and frequency slot are chosen based on the UE that is most efficient for the user in question thus enabling energy-efficient communication with other peer whether it is directly in range or not

(c) **Interference management:** interference is reduced by accurate scheduling when the channel is shared between cellular and D2D. D2D standards balance transmitted powers and required QoS for transmission on such a shared channel

(d) **Mode selection:** D2D offers far better performance in terms of spectral efficiency, low latency, QoS etc with respect to cellular communications

(e) **Peer Discovery:** D2D implements and efficient way to discover nearby users typically using random access methods or getting assisted from e-NB.

(f) **Mobility:** D2D supports dynamic users as it is primarily focused on communications without a centralized signalling entity in between.

So, we can consider that D2D works well as a part of 5G. When two devices are connected to the same cell, it means they are in proximity. Hence, they release resources for the entire network and develop direct communications between them. This is very important with respect to vehicular communications due to the nature of such communications.

3.5.1 D2D Applications

The applications of D2D include systems where a large amount of data needs to be transferred with a short-range connection and where devices are in proximity. We can consider the following applications:

a) **Local Service:** Data is directly transmitted between the terminals and does not route through the network side. Any user can find ither nearby users and share data with them. Another service is local data transmission which include advertisements based on proximity to certain locations. Even with the heavy nature of cellular traffic, by keeping media servers to deliver media services to users.

b) **Emergency Communications:** During emergencies, traditional networks might get damaged and can hamper communication. That is where wireless networks can still be set up based on D2D connection in an ad hoc fashion in single or multi-hop fashion so that a network can be established that spans from the device to the working BS.

c) **IoT Enhancement:** D2D communications is supposed to bring about the enhancement in the use of IoT devices which not only include traditional smart devices that are used but also includes scope for V2V communications and Internet of Vehicles (IoV) which enables sending messages among vehicles and infrastructure based on the dynamic scenario of a real-time road.

4. Vehicular Ad hoc Networks

From the era of wired networking, we have shifted to wireless infrastructure. This brings us to the concept of decentralized networks where a centralized entity to control such communications. Instead, every node participating in such communications forwards information to the other nodes in a dynamic way. This is referred to as a Mobile Ad-hoc NETwork or MANET. The key parameter of such a communication is the mobility of such a device which will change cells continuously and hence there are several challenges which include that each device must keep a constant update of the network topology that it is connected in.

Continuing from this we come to the specialized case where these devices can be vehicles and that is how the concept of **Vehicular Ad-hoc NETwork (VANET)** comes into play. VANET is a dynamic, wireless multi-hop network with a constraint of rapid topology change due to the non-static nature of vehicles. It can be implemented for a variety of applications including collision prevention, real time traffic monitoring, dynamic route scheduling, communication with pedestrians or infrastructure.[5] This is managed through the development of smart vehicles which contain modules to enable such communications. VANET supports Intelligent Transport Systems standards. ITS is responsible for offering the services while VANET provides the tools to implement ITS's purpose to further the concept of smart vehicles and smart cities and that is what is considered plausible for the future of IoV (Internet of Vehicles).

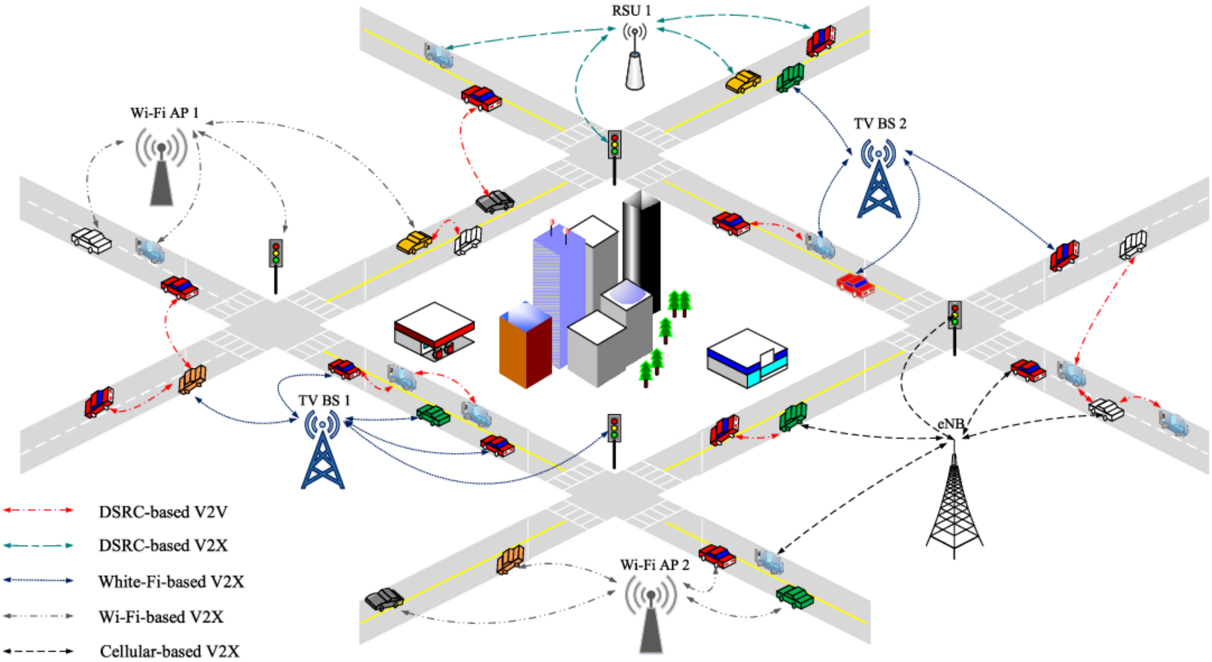


Figure 4: Representation of a VANET environment

4.1 Applications of VANET

The dynamic nature of these communications would mean that latency and reliability are two key factors that needs to be enforced. Additionally, what needs to be checked is the scope of message validity, which means that since the communication is delay sensitive, outdated messages needs to be filtered out. The transactions carried out by vehicles has local and limited relevance with respect to time (lifetime and limited temporal scope), space (local validity, limited spatial scope and local interest), and agents involved (important to agents in a limited area around the vehicle)

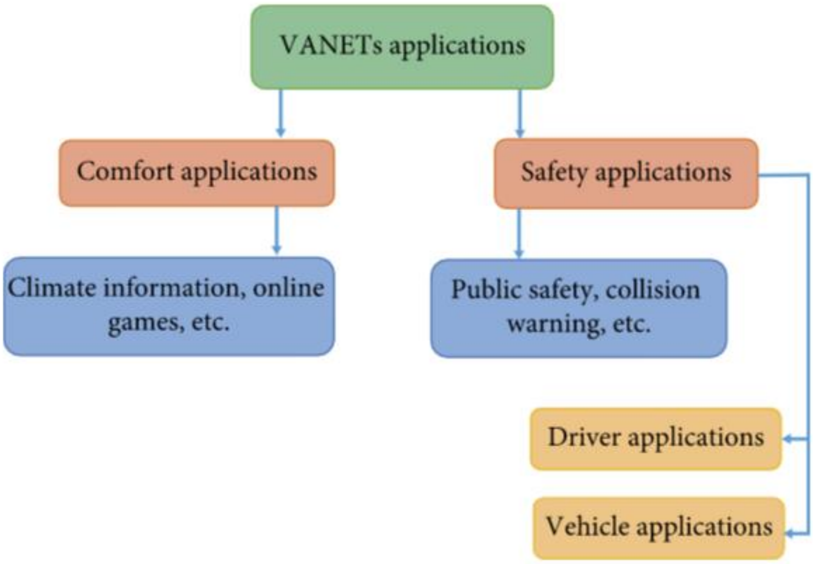


Figure 4.1 : Applications of VANET

VANET applications can be divided between Safety (safety messaging applications) and non-Safety applications (assisted driving, best path to destination) etc. Some of the applications of VANET would be:

a) General Information Services
b) Vehicle Safety Information Services
c) Critical and time-sensitive information
d) Fleet management applications
e) Active Road Safety applications
f) Commercial applications

Most of the applications require the fundamental conditions of latency and reliability. However, there are many challenges additionally which include:

(a) **Scalability:** The model needs to be implementable in a scale that matches the needs of such an infrastructure. So, we need to understand the environment of implementations, the traffic parameters, as well as respect the criticality of the communications and make it scalable.
(b) **Security and Trust:** During message transmission or receiving the data travels on a wireless medium and is equally prone to security issues that other wireless transmission faces. So, users might need the presence of a security blanket on such critical messages. However, security mechanisms if

implemented could place a constraint on the latency-criticality of such communications.

(c) **QoS and Traffic characterization:** QoS can be another factor as other than latency-critical information, there will be other kind of messages from different applications

(d) **Node Cooperation:** Most times, we will have conditions where the vehicles will not be in direct line of contact with each other but there might be other vehicles in-between them. So, we understand that there might be scenarios when we will be having multi-hop communications. So, nodes need to make available their resources to other nodes to facilitate such communication.

Therefore, we need to find a trade-off between such issues without degrading the requirements of such a communication. These need to be simulated to understand the complexities involved and that is where we GEMV^2 will be used.

4.2 Routing Protocols

A routing protocol is required in any communication network to forward data on routes, receiving data, route selection and management. VANET protocols can be defined into proactive, reactive and hybrid, each with its specific use cases which are (but not limited to) Proactive protocols maintain and update information on routing between nodes while reactive routing protocols to route requests and keep track of changes in network topology.

Some of the routing protocols include:

(a) **AODV** (Ad-hoc On Demand Distance Protocol): In AODV, nodes discover routes in request-response cycles. A node requests a route to a destination by broadcasting messages to all its neighbours. When a node receives a message but does not have a route to the requested destination, it in turn broadcasts the message. Also, it remembers a reverse-route to the requesting node which can be used to forward subsequent responses to this node. This process is continued till the destination node is found and a response is sent back to the source node through the path maintained. The protocol will also keep the discovered route in its memory.

(b) **OLSR** (Optimized Link State Routing Protocol): OLSR is a protocol used in mobile networks which can be extended to use in other wireless ad hoc networks. It is a proactive protocol which sends messages to discover and disseminate link state information to compute next hop destinations using shortest hop paths as a metric. Control messages are also forwarded by these nodes between themselves to maintain and select proper routing to any destination nodes. These are called multipoint relays. Being a proactive protocol routes to all other destinations within the network are maintained before use and no overhead is used in route discovery.

(c) **DSR** (Dynamic Source Routing): As the name suggests it is a reactive protocol and sends data when needed by discovering routes first and send route replies using unique IDs and notifies of Route Error if needed. VANET uses network information to be sent to roadside unit regarding density and distribution of traffic.

4.3 RSU Deployment

In a VANET or a mobile environment to maintain use-cases for vehicle-to-everything communications, we need to consider roadside infrastructure that needs to be considered which include traffic lights, streetlights, speed braking systems etc. These also need to have sensors to act as transceivers in the ecosystem of vehicular communications. So, these are called **Roadside Unit** (RSU), which is a DSRC transceiver to collect real-time data from vehicles and transfer it to a centralized database system for analysis. These RSUs also disseminate information which include safety messages, weather conditions, traffic conditions, dynamic incidents etc.

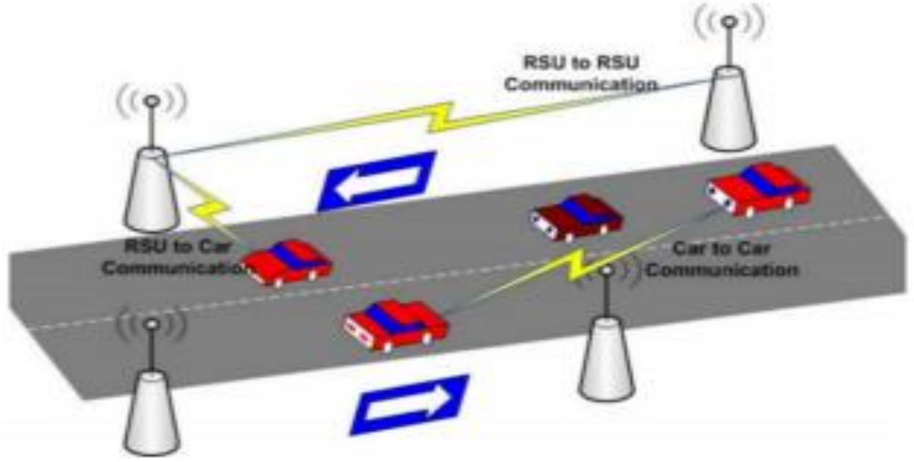


Figure 4.3 : RSUs in a VANET environment

4.4 Vehicle-to-Everything Protocol

V2X was a standard developed keeping in mind specifically vehicular communications so all devices which are part of this technology need to follow the same paradigm to communicate with other devices. There is a specific standard from IEEE which specifies vehicular operations which is IEEE 802.11p operating in the 5.9 GHz frequency band and enables the direct exchange of information among vehicles in proximity including the communication between vehicles and roadside infrastructure. There are several sectors of Vehicle-to-Everything communications:

(a) **Vehicle-to-Vehicle (V2V)**
(b) **Vehicle-to-Infrastructure (V2I)**
(c) **Vehicle-to-Pedestrian (V2P)**
(d) **Vehicle-to-Network (V2N)**

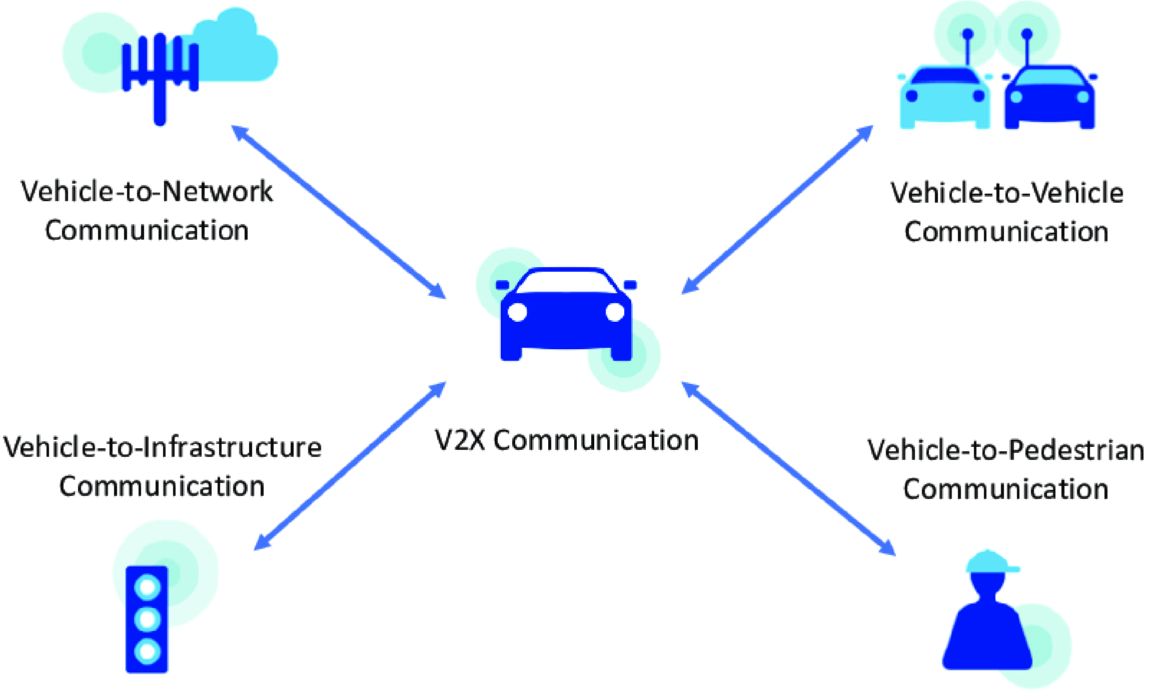


Figure 4.4 : Facets of Vehicular Communications

The uses cases for such sectors were divided into several categories:

(a) Active road safety
(b) Traffic and Fleet management
(c) Co-operative services
(d) Global Internet services
(e) Driving assistance and smart driving
(f) Cooperative manoeuvring with safety margin
(g) Sharing of trajectory of vehicles in proximity

V2X works in a distributed manner so there need not be any synchronization with a centralized infrastructure as Device-to-Device paradigm is maintained with high throughput and minimal latency. However, V2V communications are quite complex to implement because of the mobile nature of both nodes. So, most cases there is not a direct communication between the vehicles involved but what occurs is a multi-hop communication in which vehicles between the source and destination nodes act as relays. In that case there is also the need of an efficient routing protocol.

V2X implementation always incorporates the challenges of:

(a) Mobile nodes
(b) Non-static nature of the connection
(c) Low duration of proximity for communicating nodes
(d) Condition of the channel with respect to Fading
(e) Simultaneous transmissions from multiple devices in proximity resulting in collision
(f) Power transmission of the nodes

4.5 Internet-of-Vehicles (IOV)

We understand that with the evolving concept of smart vehicles we go towards the concept of Internet of Vehicles (IoV). This can help maintain traffic security as each vehicle is a node emitting real-time data on its parameters to other entities around it which bring the concept of Vehicle-to-Everything (V2X) communications. So, this will improve the efficiency of transportation, with real-time traffic information being exchanged among various nodes and with RSUs, which in turn send this huge volume of information to a centralized Traffic Control systems for processing.

In short, we understand that the implementation of IoV will be a good way to involve vehicular communications by leveraging the 5G infrastructure in place for optimal traffic control, smart driving, fleet control etc. 5G with its superior throughput will enable such huge volumes of data to be sent with minimal latency and high reliability to effectively maintain the services associated with such dynamic scenarios. Working together with IoT will be key to support the automotive industry in implementing this.

4.6 Impact of 5G on V2X

The 5G network is supposed to help implement use cases for V2X communications because of the nature of the 5G communications itself involving MIMO and beamforming antennas which can help the Device-to-Device scenarios to be supported. The automotive industry led the foundations to work with the 5G New Radio solutions creating a different branch of V2X based solutions called NR-V2X. This led to the advancement of vehicle platooning applications and for automated driving applications, better application support and services, fully automated driving applications and better connection to mobility and road safety applications and incorporates the idea of smart cities and smart navigation. In addition to it, it supports the needs of reliability, latency, and throughput even when including rapidly changing channels due to vehicle mobility or fading conditions.

5. Geometry based propagation models for V2V communication

5.1 Propagation Models

The environment, or the channel conditions, needs to be taken into consideration where the vehicles are running in the real world, there are streets which are laid with all kinds of obstacles around them including vehicles, foliage, buildings, streetlights, and traffic lights. So, a lot of the channel is obstructive, when considered from the point of view of the communicating vehicles. We also need to take into consideration the kind of radios/sensors/transceivers which are located on the vehicles or devices controlling the transmission and receiving capabilities of the nodes. But we can safely say that since the height of most vehicles are lower than buildings, foliage, and other obstructions, most of the transmitted signal will be attenuated and lost. [1]

These radio waves will have a combination of a lot of obstacles while travelling from transmitter to receiver. This might cause a multipath effect at the receiver wherein the transmitted signal will reach the receiver on more than one path and attenuated. There might be both large and small-scale propagation effects.

5.1.1 Free Space propagation model

The free space propagation model is used to predict received signals when the transmitter and receiver have no obstruction/minimal obstruction between them. This is what is called Line-of-Sight (LOS) condition. In such conditions, the free space propagation model tells us that the received power decays as a function of the transmitted power which depended on the receiver separation, raised to a particular power. Friis free space equation helps us understand it,

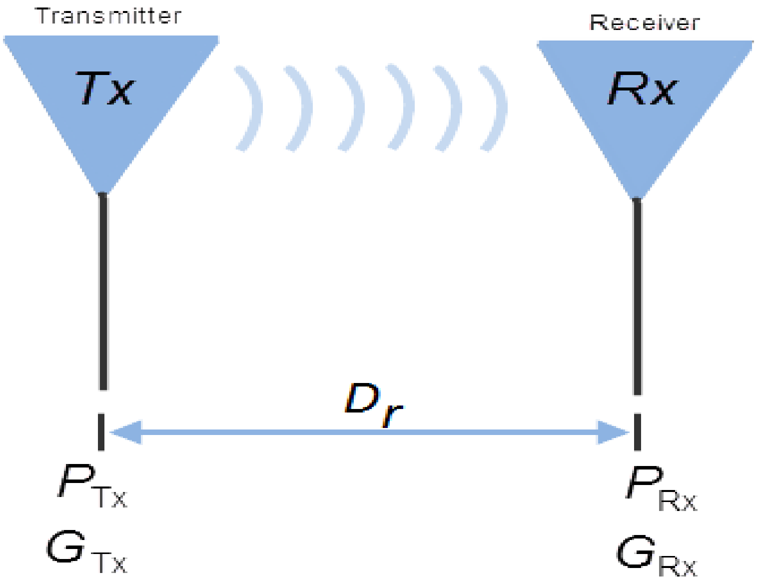


Figure 5.1.1: Free space propagation model

$$P_{Rx}(d) = P_{Tx} G_{Tx} G_{Rx} \lambda^2 / (4\pi)^2 D_r^2 \qquad (5.1.1)$$

(a) P_{Tx} is the transmitted power

(b) P_{Rx} is the received power which is a function of transmitted power, distance between the receiver and transmitter, and raised to some power

(c) G_{Tx} and G_{Rx} are the gain of the transmitter and receiver respectively

(d) D_r is the distance between the transmitting and receiving antenna

The gain of an antenna is related to its effective aperture, A_e as

$$G = 4\pi A_e / \lambda^2 \qquad (5.1.2)$$

Aperture A_e depends on the dimensions of the antenna and $\boldsymbol{\lambda}$ is related to carrier frequency by:

$$\lambda = c/f = 2\pi c/\omega_c \qquad (5.1.3)$$

The path loss in dB is calculated as:

$$PL = 10 \log_{10} (P_t/P_r) \tag{5.1.4}$$

In a rural or highway environment we can have this kind of propagation model.

5.1.2 Two-ray ground reflection model

The two-ray ground reflection model indicates that we have a direct path (LOS condition) and a non-direct path (Non-Line-of-Sight) condition. The total signal at the receiver is the sum of the two signals: the direct path and the non-direct path. The non-direct path might include reflections, scattering, diffractions. In V2X communications, there is always the existence of such components.

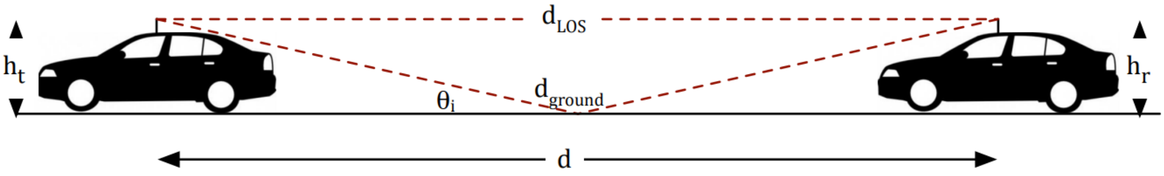


Figure 5.1.2: Two-ray ground reflection model

So mathematically we have,

$$E_{receiver} = E_{LOS} + E_{NLOS}$$

where,

$E_{receiver}$ is the energy received at the destination of the signal

E_{LOS} and E_{NLOS} are the direct and non-direct components.

5.1.3 Diffraction model

When there is an obstacle between the two nodes which might match the dimensions of the travelling wave travelling between the source and destination. In our cases what might cause these effects is due to the obstructions possible between the source and receiver. There are multiple computations which show the total energy received at the receiver. However, for reasons of simplicity we do not go to such details.

5.2 Geometry based models (GEMV^2)

Geometric modelling refers to the mathematical description of elements for computations using software. Mostly, two-dimensional, or three-dimensional shapes are used to describe the model as best as possible. In terms of vehicular communications, the geometry-based model can be used to define and model vehicles, their field of view, the other vehicles, buildings, or foliage around it. This is where we use the tool **GEMV^2** (Geometry-based Efficient propagation model for Vehicle-to-Vehicle communication).

GEMV^2 is used to model the scenarios possible in a typical real-time vehicular environment. These kinds of simulations are indeed important before deploying any such scenario to understand issues regarding it and working on a roundabout for these issues as the cost involved in deployment is too high to be remediated after a deployment has been made of such a technology. We also need to see the impact or the scale at which this can be deployed because in a realistic traffic scenario there are massive number of vehicles and objects and depends on traffic density, geographical location etc. So, we first need to perform a simulation of a real-world traffic scenario and understand and analyse the results before actual deployments.

5.2.1 Tool features

GEMV^2 considers several features which include:

(a) Classifies how the channel or link is between the objects or vehicles which include Line-of-sight (LOS) and Non-LOS scenarios.
(b) Considers open spaces, urban and semi-urban scenarios.
(c) Due to dynamic nature of nodes, it uses the wave reflection and diffraction models to identify the signal path between nodes and considers traditional Path-loss models and Fading conditions.
(d) Uses external tools like SUMO to simulate real world dynamic vehicle scenarios and uses OpenStreetMap for the GPS location of the vehicles.

(e) It uses the R-Tree structure to store the outlines of vehicles, buildings, and foliage.
(f) Output of all the simulations is stored in a CSV format.

5.3 Conclusions

So, we need to consider the LOS and NLOS conditions. Line-of-Sight (LOS) conditions are when there is a direct communication between the two participating devices without any obstruction or help from any other device in between. It is very rare in real world or real time scenarios for such events, but it could take place in a highway or semi-urban environment. Non-Line-of-Sight (NLOS) conditions is the prevalent condition in most cases, and it is when there is an obstruction between the sending and receiving device. In such conditions we have a single-hop or multi-hop communication with the obstructing vehicle or infrastructure acting as a relay.

Going forward with this point we can categorize NLOS conditions can be divided broadly into the following kinds of obstruction:

(a) **NLOSv:** Non-LOS due to vehicle/s being an obstruction
(b) **NLOSb:** Non-LOS due to buildings being a partial/complete obstruction
(c) **NLOSf:** Non-LOS due to foliage (like trees, bushes, and shrubbery)

It might be possible that there is a combination of all these scenarios as a whole or there might be times when there is a higher vehicle count/density. There might also be cases where we have multiple reflections/diffractions between source-destination pair, and we need to take all of this into account when running a simulation with GEMV^2.

6. Tools used in conjunction to run GEMV^2

To run GEMV^2, we need certain tools and IDEs to run since GEMV^2 is written primarily in C++ and would need an execution environment. Additionally, there are other modules plugged in with the core modules of GEMV^2 which also enable the simulation. We can go through the modules one by one briefly.

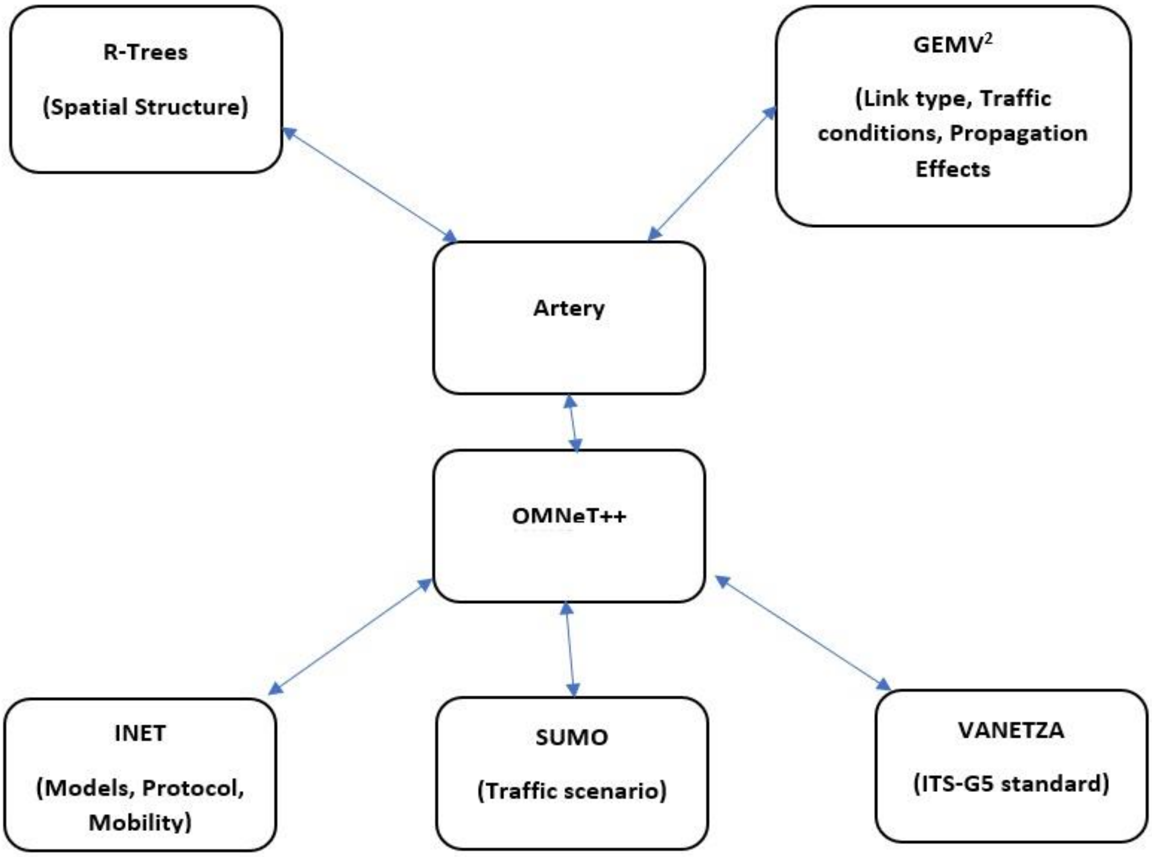


Figure 6 : Tools and Frameworks used with GEMV^2 (details of these explained further in this chapter)

6.1 OMNeT++

OMNeT++ is an extensible, modular, component-based C++ simulation library used for building network simulators. There are already existing modules which a user can add based on need. The simulator works on the concept of exchanging messages. It is a simulation framework without network protocols models like IP or HTTP. There are several external frameworks too among which INET is the most important and INET models are part of GPL for the usage of everyone. Network Description (NED) is the topology description language of OMNeT++.

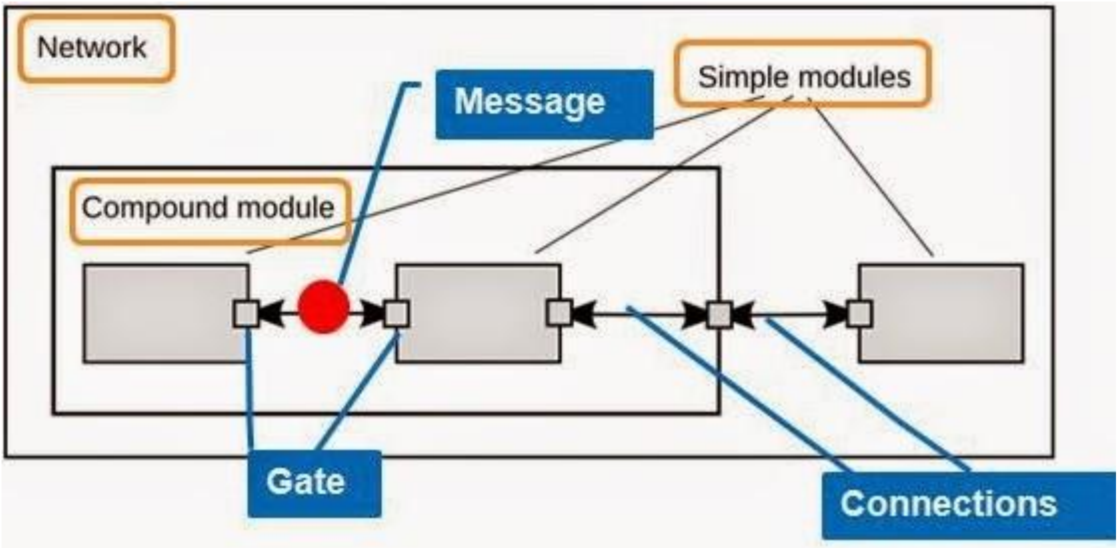


Figure 6.1 : OMNeT++ structure

6.1.1 Model

An OMNeT++ project consists of basic few parts which are common to all works of OMNeT++ which are the building blocks of any design and includes the following components:

(a) **Modules:** are standalone 'ready-made' components which can be used to perform a specific task or can be extended and mixed with other modules which can perform more complex tasks.
   i. **Simple module:** a component designed to achieve a specific task
   ii. **Complex module:** Combination of several simple modules which might be mixed with other modules to achieve more complicated tasks.

(b) **Messages:** modules communicate with each other using messages

(c) **Connections:** linking of modules, which might be done as required, to form a gateway for message passing

(d) **Gates:** gateways to establish connection between two modules/nodes

(e) **Network:** combination of one or more of the above components which help design a project to achieve a specific task

6.1.2 Structure

For a simulation to be done, we need to have a fixed structure which will enable to run the simulation. The following components must be present:

(a) **Network Description file (NED file):** The NED defines the topology of the network that needs to be present to build a working model. OMNeT++ offers a graphical view and a programmable view to change and update it according to one's needs.

(b) **Source file (.cc file):** It is the file composed of the main functions of the project. It contains the logic of the project – i.e., the class, functions/methods that need to be called to implement a specific action. The source file defines the 'initialize' method to create the message and perform actions on it and the 'handle' method to manage the message during its lifecycle.

(c) **Network Configuration file (.INI file):** The .INI file editor lets the user configure the simulation models for execution. It tells the simulation program which network we want to simulate (as the NED file may contain several networks), we pass parameters to the model, specific seeds for the random number generator, number of repetitions of the event etc. It has two modes – like the NED editor wherein we can use C++ code or a graphical interface to change the characteristics.

6.1.3 Outputs

Outputs are stored in the form of record vector or scalar values by default. We can also record histograms by putting a parameter in the .INI file and hence we get three kinds of data:

(a) **Output Vectors:** data that contain time series recorded as output after the modules run. They might record statistics about packet delay from a particular layer to the next layer, packet lost count etc.

(b) **Output Scalars:** data that contain summary statistics like packet count, maximum, minimum, mean, and standard deviation, sums etc.

(c) **Output histograms:** they are recorded as a collection of scalar results mostly which record counts of something with ranges and bins.

We can record results in two ways:

(a) **Declared statistics:** This method combines the signal mechanism and NED properties. Statistics of what needs to be recorded are declared in the NED file with @static attribute and then the correct module emits a signal which is captured with @signal attribute and then it is uploaded with the value.
(b) **Direct result recording:** In this method we use C++ to capture results from the simulation. Statistics are collected in class variables inside the modules and then a call to the functions is made using recordScalar() or recordVector() functions and using classes it is possible to store statistics like mean, standard deviation,min/max etc. We can also record distributions of data with histograms for which we need classes like (cHistogram, cKSplit, etc.) to record the values.

All these values are recorded and stored; however, they are not necessarily displayed in the output. We need to manually enable/disable from the .INI file.

(a) All recording from a @statistic can be enabled/disabled using the statistic-recording option.
(b) Recording of a scalar or a statistic object can be controlled with the scalar-recording option.
(c) Recording of a vector can be controlled with the vector-recording option.
(d) Recording of the bins of a histogram can be controlled with the bin-recording option.

A 'true/false' attribute value enables/disables the results from the user. We can also record other values like count, mean, max, vector and we can use the + and – sign to record a combination or none of if where a + (signifies a combination) and – (signifies none of that). All the collected results are stored in the Results folder. We can view them graphically from OMNeT++.

6.1.4 Result Analysis File

Default results are stored in scalar (.sca) and vector (.vec) files, but with some C++ command lines it is possible to export them in a tabular file (.csv). There are also log files created that document the significant events during the simulation. The .ANF file is not created by itself after the simulation; rather only the scalar and vector file is created containing data recorded from the simulation. Opening the scalar and vector files simultaneously creates the .ANF file. Using this file, we can view the scalar and vector data recorded about various characteristics which include count of packets passed to a particular layer, transmissions count etc. It also records histogram data from which we plot the results on a graph.

6.2 Artery

Artery enables V2X simulations based on ETSI ITS-G5 protocol and primarily focusses on the European standards. Artery provides such services through its middleware and integrates with another framework called INET.

Artery enables the V2X mechanisms providing the communication links, the interfaces for the nodes, the message communication mechanisms, the set of features to work with 5G mobile networks. It also provides mechanisms to support different part of a vehicular transmission like mobility of vehicles, environment in which the vehicles are running, the radio environment for the communications etc. It also integrates and supports features coming from INET.

Artery supports three modes and in a dynamic way :

(a) Uplink/Downlink
(b) Network-assisted sidelink Mode 3
(c) Out-of-Coverage sidelink Mode 4

All these modes are dynamic, and it may happen that all or some of the modes are encountered in each scenario.

Artery is modular and uses messages to communicate with each other with a message passing mechanism [7]:

(a) The control-plane is independent and acts through messages
(b) The user-plane is implemented as an amalgamation of simple modules and some compound modules
(c) The link path of a vehicle is given by GEMV^2 and is implemented in Artery autonomously

(d) The type of messages exchanged are CAM and DENM messages which have already been explained previously in detail
(e) It supports both IP-based and non-IP based transmissions

6.2.1 Architecture

GEMV^2 requires Artery to run, and both work together when running a simulation. The visual representation of a scenario is given by SUMO, which reads a configuration file which refers to XML files defining the scenario.

OMNeT++ is the event simulator where the simulation is runs. This starts by executing a target file (run-gemv^2) which launches the TraCI manager responsible for launching the SUMO process and connecting the command interface via TCP.

6.2.2 Environment

We consider the vehicles in a simulation to have sensors (as is expected of smart vehicles) and each such sensor in the vehicles have a transmission and reception range. In GEMV^2 , these ranges are visualized in the form of a cone which correspond to the maximum reception or transmission range of a sensor. We consider that within this cone we have perfect transmissions and receptions. Around the vehicles we have the environment as appropriate for a scenario with foliage, buildings etc.

6.3 INET

INET framework is specifically designed for communication networks which is used to test an existing network in place or try new paradigms and scenarios. It supports a wide range of communication protocols and standards, both wireless and wired, including TCP, UDP, Ipv4/Ipv6. It also supports mobility standards like VANET protocols. The baseline concept is the same as that of OMNeT++, of communicating with messages between different modules and like OMNeT++, the simple modules can be collated together to form complex modules. It is open source and can be extended or used with OMNeT++ and is hence used in this project.

Some of the features of INET are:

(a) OSI layers implemented from physical to application layer
(b) Pluggable protocol implementations
(c) Ipv4/Ipv6 network stack (we can build our own network layer)

(d) Transport layer protocols supported like TCP, UDP
(e) Routing protocols present (ad-hoc and wired)
(f) Wired/Wireless interfaces (Ethernet, 802.11, PPP etc.)
(g) Network emulation support
(h) Mobility support
(i) Wide range of application models
(j) Supports the modelling of physical environment (obstacles for radio propagation etc.)

The list is not exhaustive, and these features can be implemented with several packages or modules in place to achieve it.

6.3.1 Layered Protocol Base

Here, the signals are defined and the mechanism of packet 'sent-to-upper' and 'sent-to-lower' is defined as well as 'packet received' and 'packet drops.' So, we understand that signals are recorded in this layer to be analysed later. We can also consider the Layered Protocol Base to be the OSI model with a seven-layer protocol in place where we have the concept of 'sent-to-upper' and 'sent-to-lower'.

6.3.2 Physical Layer Base

In the Physical Layer Base, we have the implementation of the Radio module. This description of the radio module defines the transceiver that transmits and receives signals on the medium. The radio mode can be changed between mode 3 and mode 4. It also contains sub-modules for antenna, transmitter, receiver, energy models, transmission power etc.

There is also the Shortcut-Radio module which is a peer radio protocol that bypasses the physical medium and sends packet to other radio modules without any physical layer processing in the radio medium.

The IRadio module is useful for transmission of frames over a wireless medium. In transmissions, upper layers send frames to the radio module which in turn are sent out via the physical layer in the form of signals. In reception, the received signals are received by the Physical layer and sent to upper layers after a check has been made that the signals have been received correctly.

6.3.3 Mobility Base

This is the module used to set up mobility parameters and areas in which the mobility is constrained, i.e., in which the vehicle parameters which affect the whole computation, and which are:

(a) Position
(b) Velocity
(c) Acceleration
(d) Angular position

6.3.4 Antenna Base

This defines the kind of antenna modules the user can use in the simulation. We have used only Isotropic and Parabolic antennas.

6.3.5 Physical Environment

This defines the channel through which our signals are going to travel, and it contains all the parameters which define an actual channel in a real-world scenario and there are several objects and surfaces that influence the signal traversal which includes absorption in the medium, reflection, refraction, and other multi-path phenomena. Most of the objects are predefined in the XML configuration file (XML files are a good way of assigning parameters via external sources) and here it is a practice to use XML files to assign such data.

The objects have the following properties:

(a) **Object:** id, name, position, orientation, colour, outline width, opacity, texture etc.
(b) **Shape:** id, type, size, radius, height etc.
(c) **Material:** id, name, resistivity, permittivity, permeability etc.

6.3.6 Scenario Manager

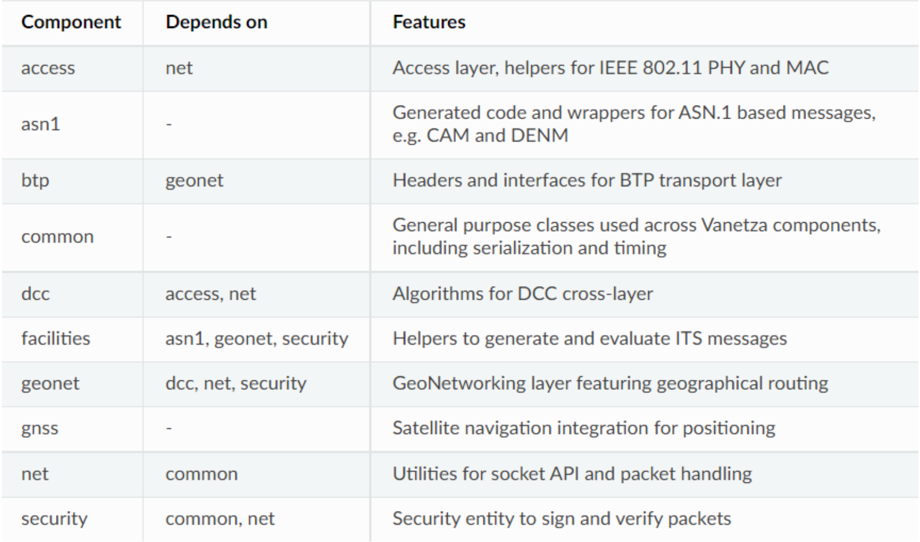
The scenario manager is used for setting up and controlling simulation experiments, scheduling events and their time, changing parameter values and/or rates, removing or adding connections, removing, or adding routes etc. This executes and XML script.

6.4 VANETZA

Vanetza is an open-source implementation of ETSI C-ITS protocol suite. It comprises of the following protocols and features:

(a) Geo Networking (GN)
(b) Basic Transport Protocol (BTP)
(c) Decentralized Congestion Control (DCC)
(d) Security
(e) Support for ASN.1 messages (Facilities) such as CAM and DENM

The major features are listed below:



Component	Depends on	Features
access	net	Access layer, helpers for IEEE 802.11 PHY and MAC
asn1	-	Generated code and wrappers for ASN.1 based messages, e.g. CAM and DENM
btp	geonet	Headers and interfaces for BTP transport layer
common	-	General purpose classes used across Vanetza components, including serialization and timing
dcc	access, net	Algorithms for DCC cross-layer
facilities	asn1, geonet, security	Helpers to generate and evaluate ITS messages
geonet	dcc, net, security	GeoNetworking layer featuring geographical routing
gnss	-	Satellite navigation integration for positioning
net	common	Utilities for socket API and packet handling
security	common, net	Security entity to sign and verify packets

Figure 6.4 : Components of VANETZAs tools

6.5 SUMO and TraCI

Simulation of Urban Mobility (SUMO) is an open source, portable, microscopic, and continuous multi-modal traffic simulation package designed to handle large networks. It is an open-source software used for simulation and analysis of road traffic and traffic management systems such as route choice, traffic light algorithms, vehicular communications, fleet management system etc.

SUMO has several modules among which **netedit** is one of the most important. It is a visual network editor to create networks from scratch and modify all aspects of existing networks. It is used to make a visual representation of the roads and other infrastructure on roads to simulate the real world. Sometimes, we can have a more realistic and accurate description of the roads if we use external sources to upload more realistic maps, like **OpenStreetMap.**

When we are running a simulation in SUMO, each vehicle has a particular time in which it enters and leave the scenario during which its parameters change like velocity, position, a pre-defined route to follow during the simulation. These vehicles are supposed to receive data dynamically and other messages and information from RSUs in real-world scenario. To simulate that we have the **TraCI** (Traffic Control Interface). TraCI uses a TCP based client-server architecture to provide access to SUMO. TraCI couples two simulators: traffic simulator (SUMO) and a network (OMNeT++) simulator. TraCI connects both in real time thus controlling the mobility attribute of each vehicle.

7. GEMV^2 in SUMO

All the traffic scenarios that need to be uploaded to SUMO are present in GEMV^2 There are two or three scenarios simulated consisting of LOS and NLOS conditions and we have configuration files of each scenario along with .xml files which define the scenarios. So, we have the following files,

(a) **LinkType.sumo.cfg:** permits SUMO to load the network and road files for the traffic scenario
(b) **LinkType.net.xml:** this file contains the parameters of the vehicles (length, width, height, speed, direction, etc.)
(c) **LinkType.rou.xml:** it contains the parameters regarding the route edge followed by vehicles during their mobility in the scenario and it sets up the departure speed and time for each street
(d) **LinkType.poly.xml:** this file defines parameters regarding buildings and foliage

7.1 R-Trees Structure

R-trees are tree data structures used for spatial access methods i.e., for indexing multi-dimensional information such as geographical coordinates, rectangles, or polygons. A common real-world use of R-trees would be to store spatial objects are the polygons that typical maps are made of streets, buildings, coastlines etc. and algorithms can be used to find certain structures within a certain distance or radius. The R-tree structure accelerate nearest neighbour search.

The R-trees structure is used to manipulate traffic objects using a hierarchical view based on their location in the map space. It can also distinguish moving objects with respect to static objects (buildings and foliage and other static infrastructure we may find in the streets). We traverse the tree in a top-down fashion. It is also a balanced search tree (so all leaf nodes are at the same depth). The searching algorithm is to use bounding boxes to decide whether to search inside a subtree or not. These trees are suited for large data sets and databases

They do not guarantee good worst-case performance, but generally perform well with real-world data. When data is organized in an R-tree, the neighbours within a given distance r and the k nearest neighbours of all points can be efficiently computed using a spatial join. The R-tree structure was formulated by Antonin Guttman and details of operations that can be done in the tree is explained well in his work.

7.2 Simulations

To simulate the scenarios we have envisaged, we follow a step-by-step procedure that needs to be performed after the launch of the program. Some of them are done automatically by OMNeT++ and the user also needs to take some choices. The complete flow of the application is shown in the form of a flowchart.

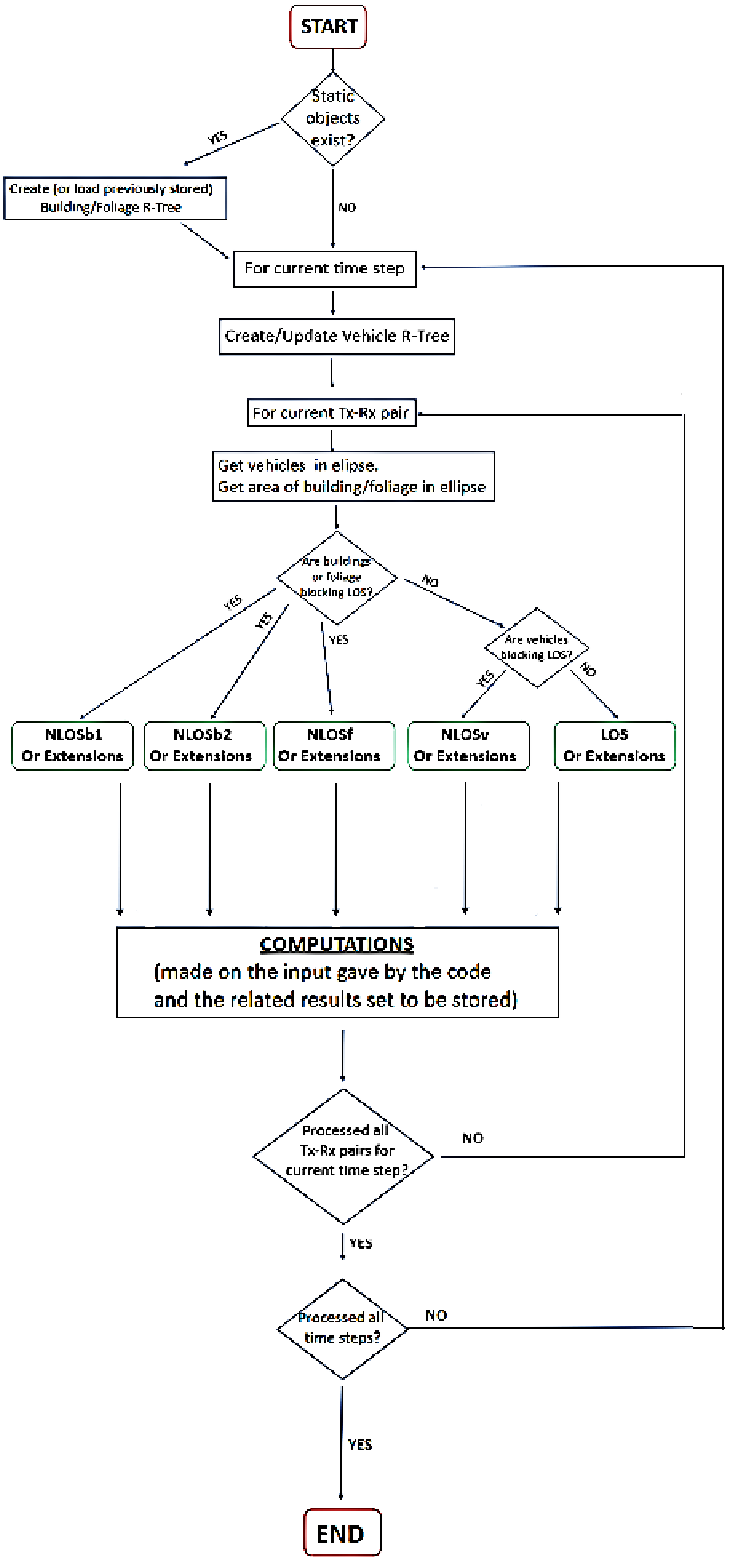


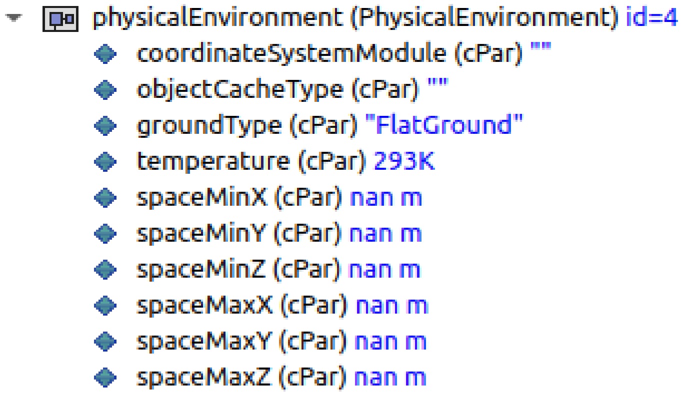
Figure 7.2 : Simulation flow

7.3 INI File, NED File and SUMO File

The INI, NED and SUMO files are the primary necessities for the simulation to start. The .INI file is used to set the initial parameters, load a NED file (which may call or load other NED files, if required in the scenario)

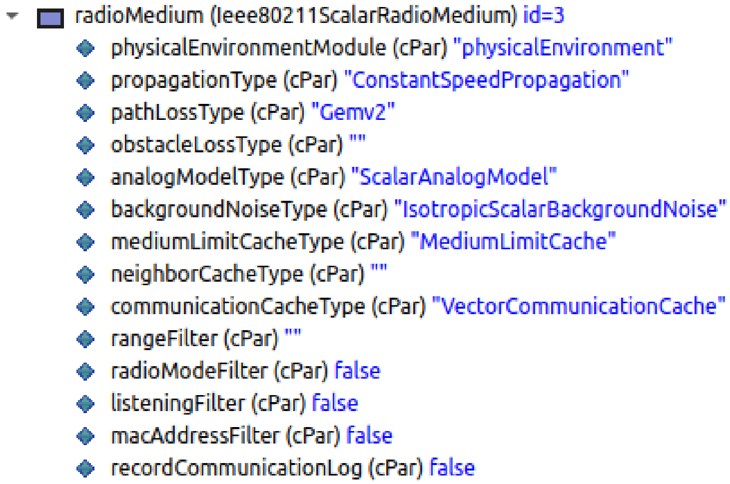
We also consider certain assumptions while making the simulations

(a) The buildings are tall with respect to the vehicles and very little power is received over them
(b) The primary obstructions considered are building, foliage and vehicles. All other obstacles present in the streets besides the above are not considered
(c) The simulation is assumed to be on a flat-ground surface
(d) Scattering of signals is not considered



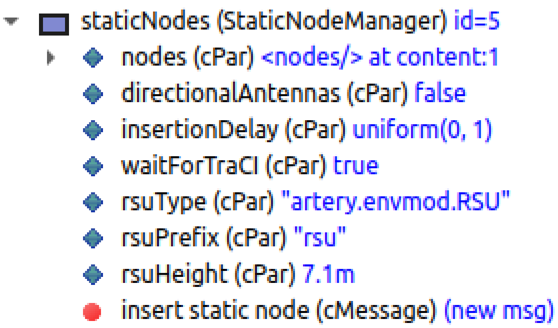
```
physicalEnvironment (PhysicalEnvironment) id=4
    coordinateSystemModule (cPar) ""
    objectCacheType (cPar) ""
    groundType (cPar) "FlatGround"
    temperature (cPar) 293K
    spaceMinX (cPar) nan m
    spaceMinY (cPar) nan m
    spaceMinZ (cPar) nan m
    spaceMaxX (cPar) nan m
    spaceMaxY (cPar) nan m
    spaceMaxZ (cPar) nan m
```

Figure 7.3 (a) : Physical environment details



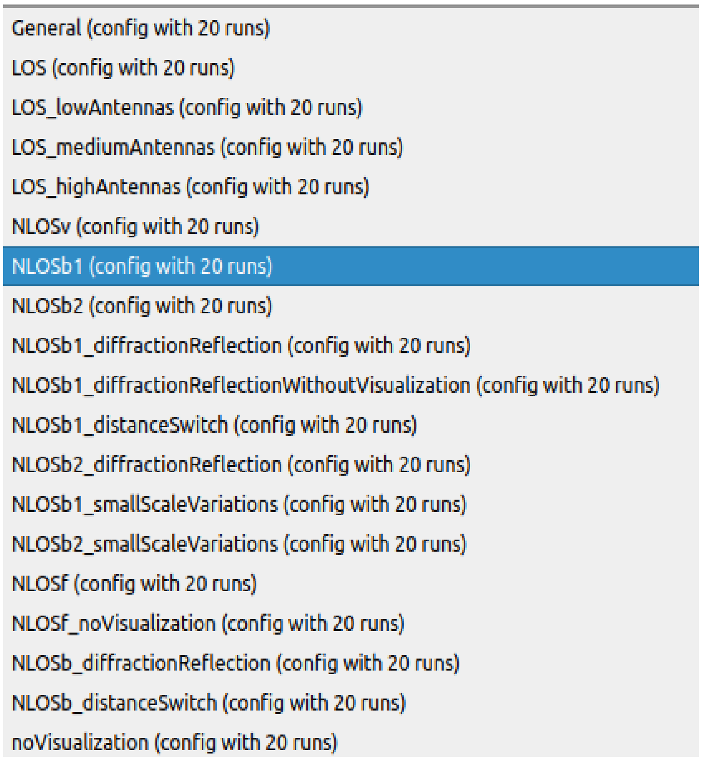
```
radioMedium (Ieee80211ScalarRadioMedium) id=3
    physicalEnvironmentModule (cPar) "physicalEnvironment"
    propagationType (cPar) "ConstantSpeedPropagation"
    pathLossType (cPar) "Gemv2"
    obstacleLossType (cPar) ""
    analogModelType (cPar) "ScalarAnalogModel"
    backgroundNoiseType (cPar) "IsotropicScalarBackgroundNoise"
    mediumLimitCacheType (cPar) "MediumLimitCache"
    neighborCacheType (cPar) ""
    communicationCacheType (cPar) "VectorCommunicationCache"
    rangeFilter (cPar) ""
    radioModeFilter (cPar) false
    listeningFilter (cPar) false
    macAddressFilter (cPar) false
    recordCommunicationLog (cPar) false
```

Figure 7.3 (b) : Radio medium details



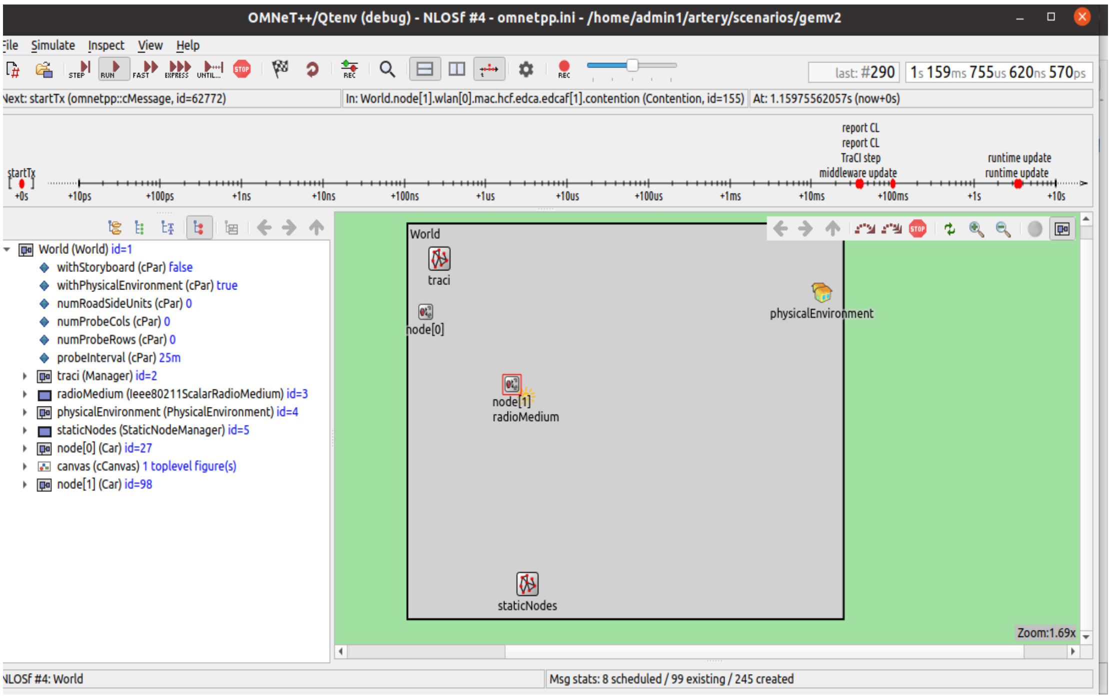
```
▾ staticNodes (StaticNodeManager) id=5
  ▸ ◆ nodes (cPar) <nodes/> at content:1
    ◆ directionalAntennas (cPar) false
    ◆ insertionDelay (cPar) uniform(0, 1)
    ◆ waitForTraCI (cPar) true
    ◆ rsuType (cPar) "artery.envmod.RSU"
    ◆ rsuPrefix (cPar) "rsu"
    ◆ rsuHeight (cPar) 7.1m
    ● insert static node (cMessage) (new msg)
```

Figure 7.3 (c) : Node details



```
General (config with 20 runs)
LOS (config with 20 runs)
LOS_lowAntennas (config with 20 runs)
LOS_mediumAntennas (config with 20 runs)
LOS_highAntennas (config with 20 runs)
NLOSv (config with 20 runs)
NLOSb1 (config with 20 runs)
NLOSb2 (config with 20 runs)
NLOSb1_diffractionReflection (config with 20 runs)
NLOSb1_diffractionReflectionWithoutVisualization (config with 20 runs)
NLOSb1_distanceSwitch (config with 20 runs)
NLOSb2_diffractionReflection (config with 20 runs)
NLOSb1_smallScaleVariations (config with 20 runs)
NLOSb2_smallScaleVariations (config with 20 runs)
NLOSf (config with 20 runs)
NLOSf_noVisualization (config with 20 runs)
NLOSb_diffractionReflection (config with 20 runs)
NLOSb_distanceSwitch (config with 20 runs)
noVisualization (config with 20 runs)
```

Figure 7.3 (d) : Scenario options



OMNeT++/Qtenv (debug) - NLOSf #4 - omnetpp.ini - /home/admin1/artery/scenarios/gemv2

last: #290 1s 159ms 755us 620ns 570ps

Next: startTx (omnetpp::cMessage, id=62772)

In: World.node[1].wlan[0].mac.hcf.edca.edcaf[1].contention (Contention, id=155)

At: 1.15975562057s (now+0s)

- World (World) id=1
  - withStoryboard (cPar) false
  - withPhysicalEnvironment (cPar) true
  - numRoadSideUnits (cPar) 0
  - numProbeCols (cPar) 0
  - numProbeRows (cPar) 0
  - probeInterval (cPar) 25m
  - traci (Manager) id=2
  - radioMedium (Ieee80211ScalarRadioMedium) id=3
  - physicalEnvironment (PhysicalEnvironment) id=4
  - staticNodes (StaticNodeManager) id=5
  - node[0] (Car) id=27
  - canvas (cCanvas) 1 toplevel figure(s)
  - node[1] (Car) id=98

NLOSf #4: World

Msg stats: 8 scheduled / 99 existing / 245 created

Figure 7.3 (e) : OMNeT++ graphical view

8. Results

8.1 Simulation and Analysis of Results

The results of the simulation are stored by the tool because it is inherent of the way it was designed. We can also store some results ourselves by making some modifications in the .INI file.

8.1.1 Constant Parameters

(a) **Flat ground:** We consider in all the scenarios that it is a flat ground surface and not consider the curvature of the earth as it would make the scenarios complicated.

(b) **Frequency:** As applicable for DSRC, the carrier radio frequency on which GEMV^2 works is 5.9GHz.

(c) **Bandwidth:** The bandwidth used is 10MHz

(d) **Permittivity:** The relative permittivity is different for different scenarios, for LOS=1, for vehicles in NLOS=6, for buildings in NLOSb and NLOSf=4.5

(e) **Path loss exponent:** 2.9 dB

(f) **Polarization:** vertical

(g) **Antenna height:** 1.5m from the ground. For LOS we have small, medium, and high antennas

(h) **RSU height:** 7.1m

8.1.2 Variable Parameters

We can make a change in some parameters. For example, we can vary transmitter power from 0.01 mW (-20 dBm) to 1mW (0 dBm) and to a maximum of 200mW (23 dBm). We also have an option to vary the LOS/NLOS ranges from 100m to a different value. For some NLOSb conditions we can customize the values of the maximum vehicle density and obstacle density.

8.2 Simulation Results

The simulation automatically stores the scalars and vector files in a repository. From these files, we open the scalar and vector files together to form the .ANF file or the analysis file which will contain the results of the simulation. We will investigate the .ANF file in more details to understand the kind of results it shows. Other than the .ANF file, we can also access the results from the Artery path or export them in .CSV format.

The .ANF file has three views we can choose from:

(a) **Inputs:** The inputs are the .sca and .vec files from which the .anf file is created
(b) **Browse Data:** these are the main outputs of the simulation and contain scalars, vectors as week as histogram data.
(c) **Datasets:** where we can perform automatic computations at the end of the simulation.

Let us list some of the types of values of each type that are recorded:

8.2.1 Browse Data : Vector Values

The values recorded are an aggregation of values and there is no concept that the data can be accessed individually. Some of the values are:

(a) **radioChannel:** it is the number of the channel used in transmission and channel number 180 is used by default
(b) **transmissionState:** number of transmissions going through each node. It may be a source node or an intermediate node, helping in multi-hop communication.
(c) **receptionState:** number of receptions going through a node. It may be a terminal node or an intermediate node, in case of multi-hop communications.
(d) **passedUpPk(in bytes):** no of bytes the node forwards to upper layers
(e) **sentDownPk(in bytes):** no of bytes the node sends down to lower layers
(f) **rcvdPkFromHL(in bytes):** no of bytes the node receives from the higher layers
(g) **rcvdPkFromLL(in bytes):** no of bytes the node receives from the lower layers

From the OSI model, we understand that the **Link Layer** is taken into consideration here. The Link Layer receives packets from the lower layer (rcvdPkFromLL) when it has received a new packet from any other node and passes it up to higher layer. Conversely, when the node needs to send a packet out it receives packets from the higher layer (rcvdPkFromHL) and sends it down to the Physical Layer for it to be transmitted out.

8.2.2 Browse Data : Scalar Values

Some of the scalar values recorded are:

(a) **Arrival computation count:** total number of occurrences propagated by the radio medium
(b) **Bitrate:** GEMV^2 operates in the 6 GHz band
(c) **Interference computation count:** number of times the radio medium has experienced any kind of interference
(d) **Radio frame sent count:** number of occurrences the radio medium has forwarded a frame on a transmission
(e) **passedUpPk(in bytes and count):** no of bytes and count of packets the node forwards to upper layers
(f) **sentDown (in bytes and count):** no. of bytes and count of packets sent to lower layers
(g) **rcvdPkFromHL (in bytes and count):** similar to what was described in vector values
(h) **rcvdPkFromLL (in bytes and count):** similar to what was described in vector values

8.2.3 Browse Data : Histogram Values

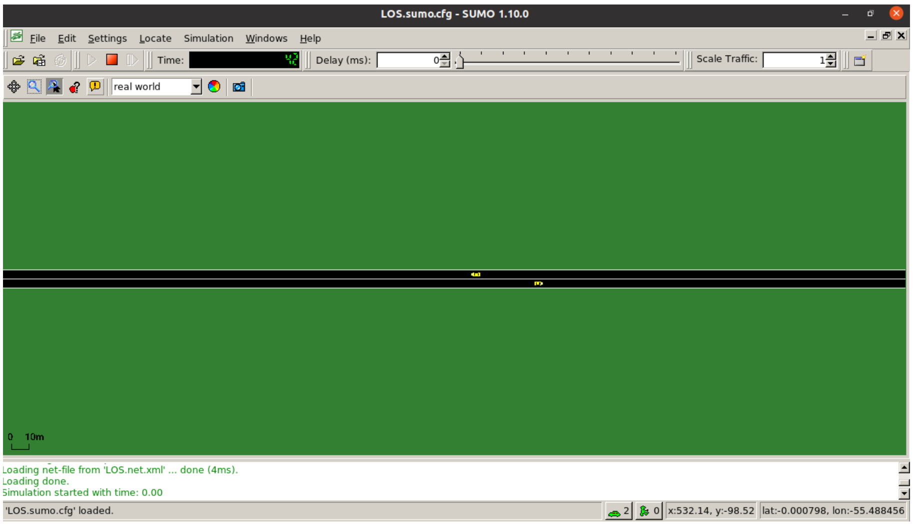
This section computes certain critical values like Packet Error Rates and Signal to Noise ratio. This helps us understand what phase our transmission is in. The two prime data that are recorded here are the following:

(a) **packetErrorRate:** count of the number of packets transmitted wrongly
(b) **minSNIR:** total number of occurrences involved in the computation of the minimum signal-to-noise plus interference ratio of all transmitted packets

8.3 Data Recorded

8.3.1 Line-of-Sight(LOS) condition

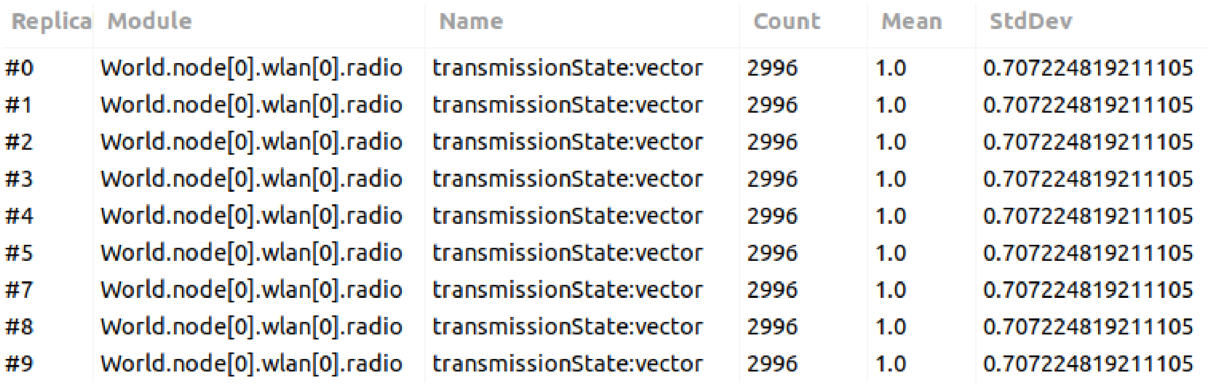
When there is a direct communication between the two participating devices without any obstruction or help from any other device in between



```
Loading net-file from 'LOS.net.xml' ... done (4ms).
Loading done.
Simulation started with time: 0.00
'LOS.sumo.cfg' loaded.
```

Figure 8.3.1 : LOS condition as visualized in SUMO

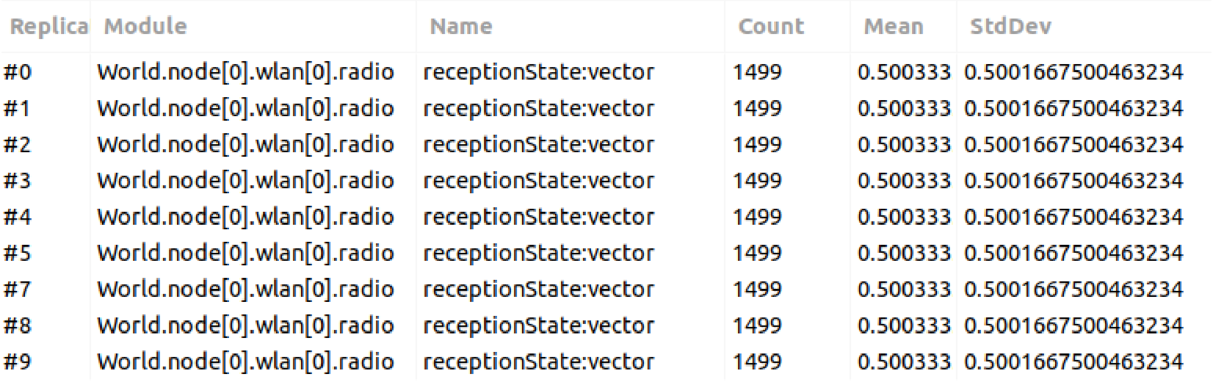
- **Transmission state:** The number of packets sent by each node is indicated here. We see no change in the number of packets transmitted and hence we conclude that the node forwards the same number of packets every time.



Replica	Module	Name	Count	Mean	StdDev
#0	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#1	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#2	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#3	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#4	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#5	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#7	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#8	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105
#9	World.node[0].wlan[0].radio	transmissionState:vector	2996	1.0	0.707224819211105

Figure 8.3.1(a) : Transmission state with count

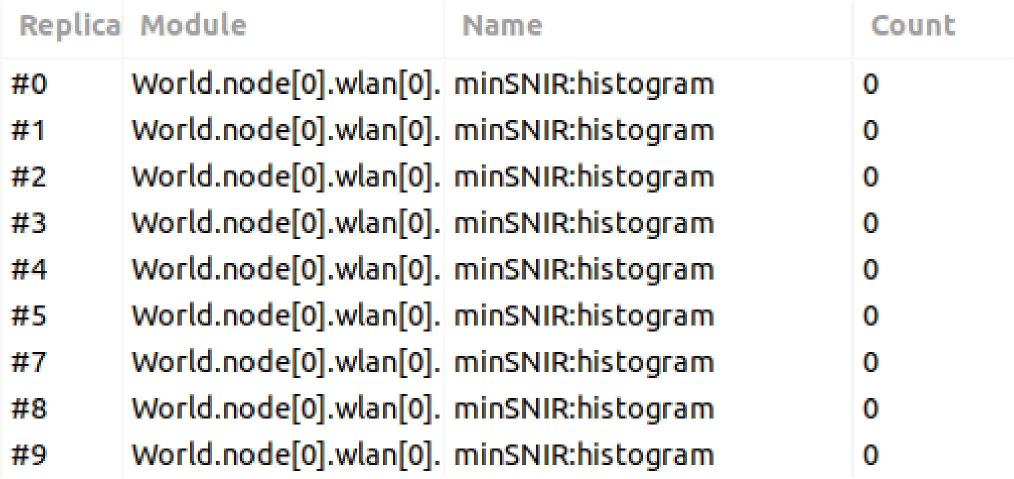
- **Reception state:** The number of packets received by the node is indicated here. We see that there is lower number of packets received as is expected since some are lost due to channel conditions or fading.



Replica	Module	Name	Count	Mean	StdDev
#0	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#1	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#2	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#3	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#4	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#5	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#7	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#8	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234
#9	World.node[0].wlan[0].radio	receptionState:vector	1499	0.500333	0.5001667500463234

Figure 8.3.1(b) : Reception state with count

- **Packet Error Rate and minSNIR:** they are related to the reception of signals or packets at the receiving nodes. At low power rates, the transmission as well as reception is not possible hence, we have low SNIR and high packet error rates. As we go towards increasing power levels, we have lower packet error rates and better SNIR.



Replica	Module	Name	Count
#0	World.node[0].wlan[0].	minSNIR:histogram	0
#1	World.node[0].wlan[0].	minSNIR:histogram	0
#2	World.node[0].wlan[0].	minSNIR:histogram	0
#3	World.node[0].wlan[0].	minSNIR:histogram	0
#4	World.node[0].wlan[0].	minSNIR:histogram	0
#5	World.node[0].wlan[0].	minSNIR:histogram	0
#7	World.node[0].wlan[0].	minSNIR:histogram	0
#8	World.node[0].wlan[0].	minSNIR:histogram	0
#9	World.node[0].wlan[0].	minSNIR:histogram	0

Figure 8.3.1(c) : Packet loss count

8.3.2 Non-Line of Sight foliage(NLOSf) condition

When we do not have direct communication due to foliage (like trees, bushes, and shrubbery) which might result in some of the transmission being attenuated.

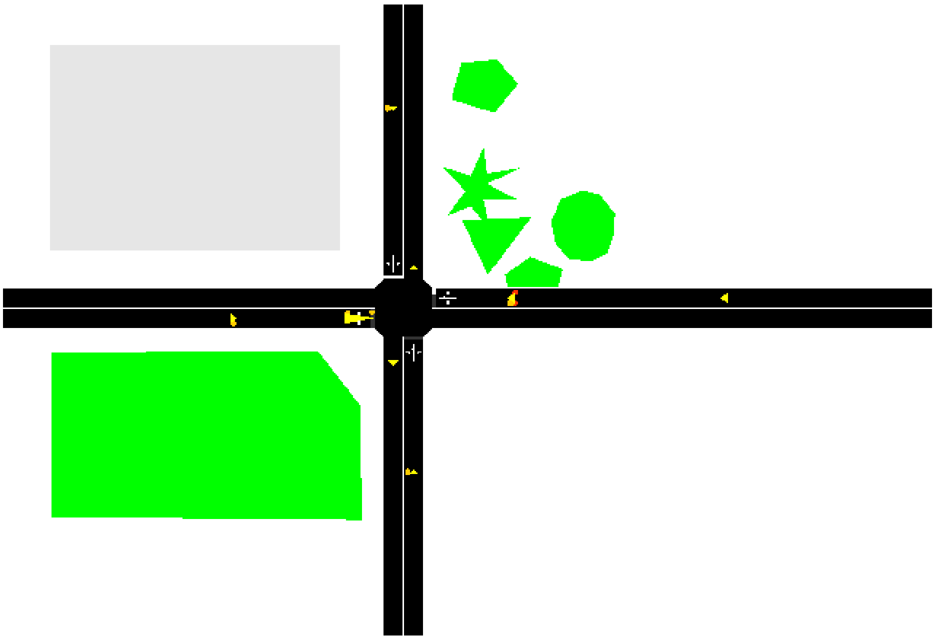
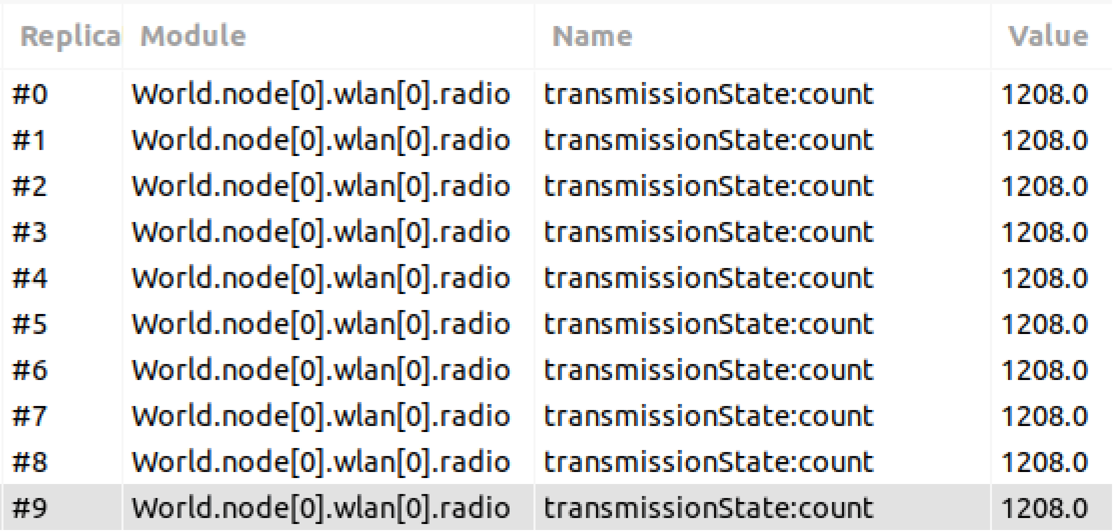


Figure 8.3.2 : NLOSf condition as visualized in SUMO

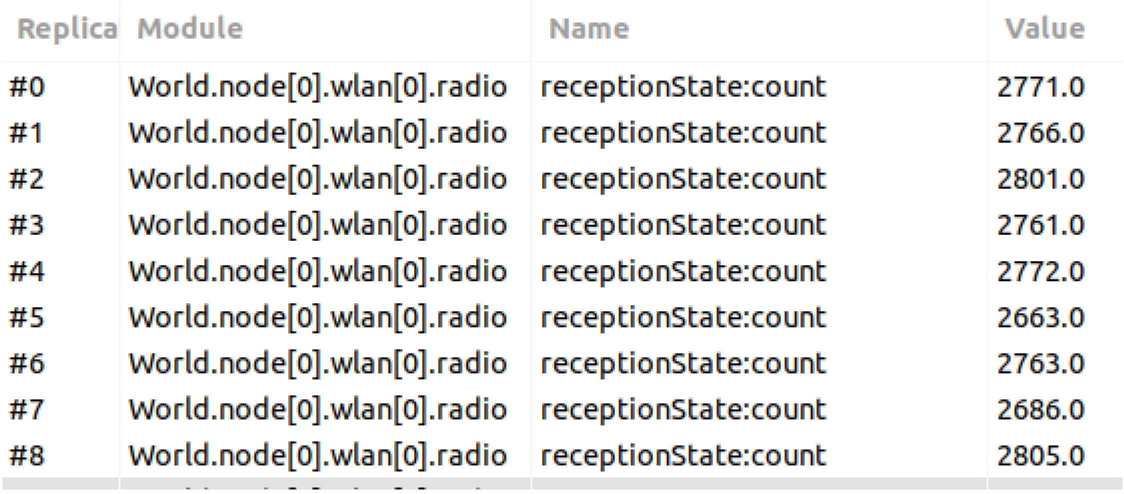
- **Transmission State:** The number of packets sent by each node is indicated here. We see no change in the number of packets transmitted and hence we conclude that the node forwards the same number of packets every time.



Replica	Module	Name	Value
#0	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#1	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#2	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#3	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#4	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#5	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#6	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#7	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#8	World.node[0].wlan[0].radio	transmissionState:count	1208.0
#9	World.node[0].wlan[0].radio	transmissionState:count	1208.0

Figure 8.3.2(a) : Transmission count of packets

- **Reception State:** The number of packets received by the node is indicated here. We see that there is lower number of packets received as is expected since some are lost due to channel conditions or fading.



Replica	Module	Name	Value
#0	World.node[0].wlan[0].radio	receptionState:count	2771.0
#1	World.node[0].wlan[0].radio	receptionState:count	2766.0
#2	World.node[0].wlan[0].radio	receptionState:count	2801.0
#3	World.node[0].wlan[0].radio	receptionState:count	2761.0
#4	World.node[0].wlan[0].radio	receptionState:count	2772.0
#5	World.node[0].wlan[0].radio	receptionState:count	2663.0
#6	World.node[0].wlan[0].radio	receptionState:count	2763.0
#7	World.node[0].wlan[0].radio	receptionState:count	2686.0
#8	World.node[0].wlan[0].radio	receptionState:count	2805.0

Figure 8.3.2(b) : Count of number of packets received (raw data)

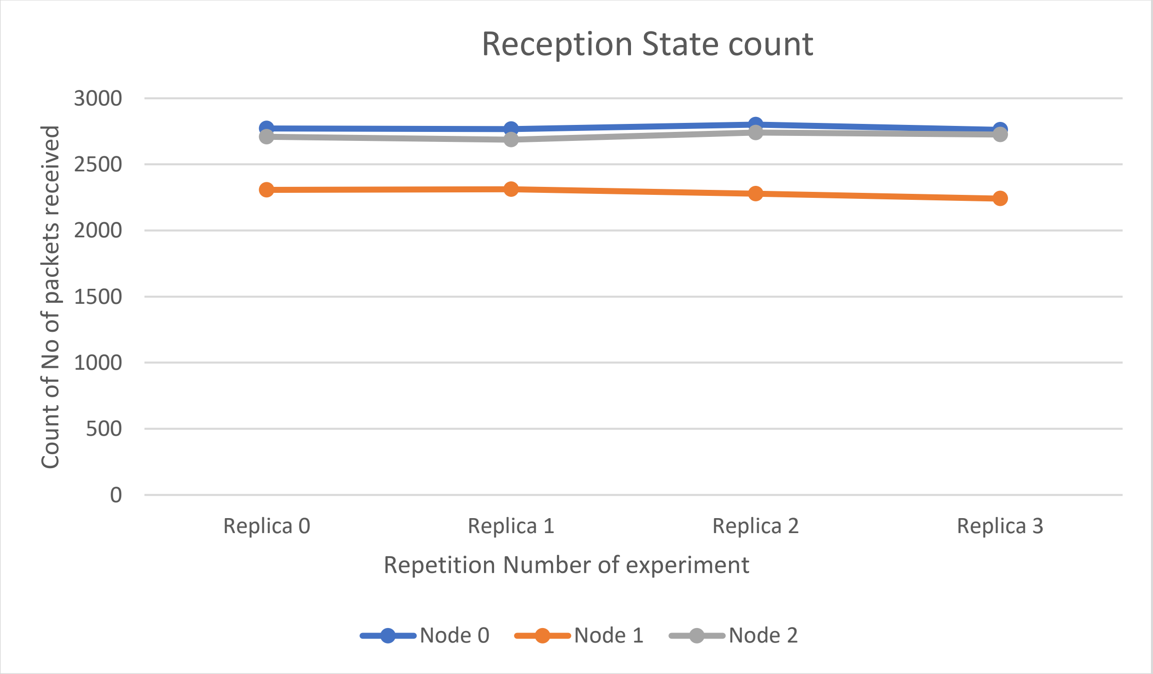
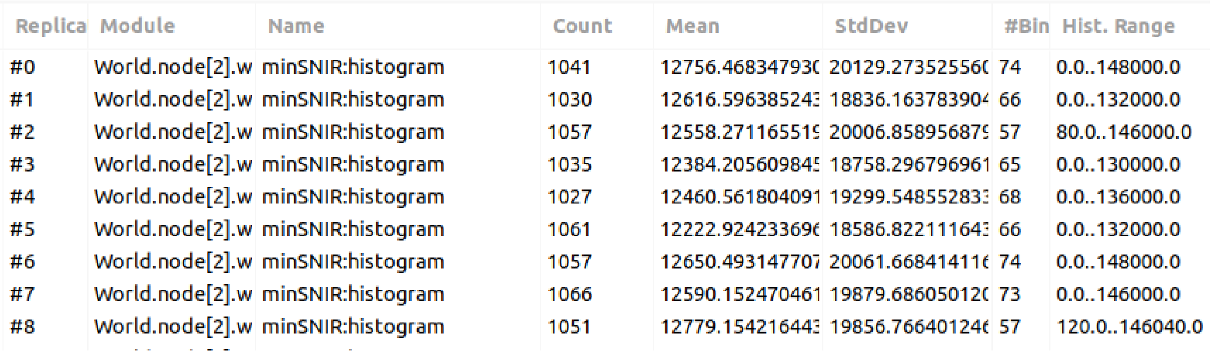


Figure 8.3.2(c) : Reception count of packets (few points plotted)

- **Packet Error Rate and minSNIR:** they are related to the reception of signals or packets at the receiving nodes. At low power rates, the transmission as well as reception is not possible hence, we have low SNIR and high packet error rates.

As we go towards increasing power levels, we have lower packet error rates and better SNIR.



Replica	Module	Name	Count	Mean	StdDev	#Bin	Hist. Range
#0	World.node[2].w	minSNIR:histogram	1041	12756.46834793	20129.27352556	74	0.0..148000.0
#1	World.node[2].w	minSNIR:histogram	1030	12616.59638524	18836.16378390	66	0.0..132000.0
#2	World.node[2].w	minSNIR:histogram	1057	12558.27116551	20006.85895687	57	80.0..146000.0
#3	World.node[2].w	minSNIR:histogram	1035	12384.20560984	18758.296796961	65	0.0..130000.0
#4	World.node[2].w	minSNIR:histogram	1027	12460.561804091	19299.54855283	68	0.0..136000.0
#5	World.node[2].w	minSNIR:histogram	1061	12222.92423369	18586.82211164	66	0.0..132000.0
#6	World.node[2].w	minSNIR:histogram	1057	12650.49314770	20061.66841411	74	0.0..148000.0
#7	World.node[2].w	minSNIR:histogram	1066	12590.152470461	19879.68605012	73	0.0..146000.0
#8	World.node[2].w	minSNIR:histogram	1051	12779.15421644	19856.76640124	57	120.0..146040.0

Figure 8.3.2(d) : Count of the number of packets lost from SNIR data (raw data)

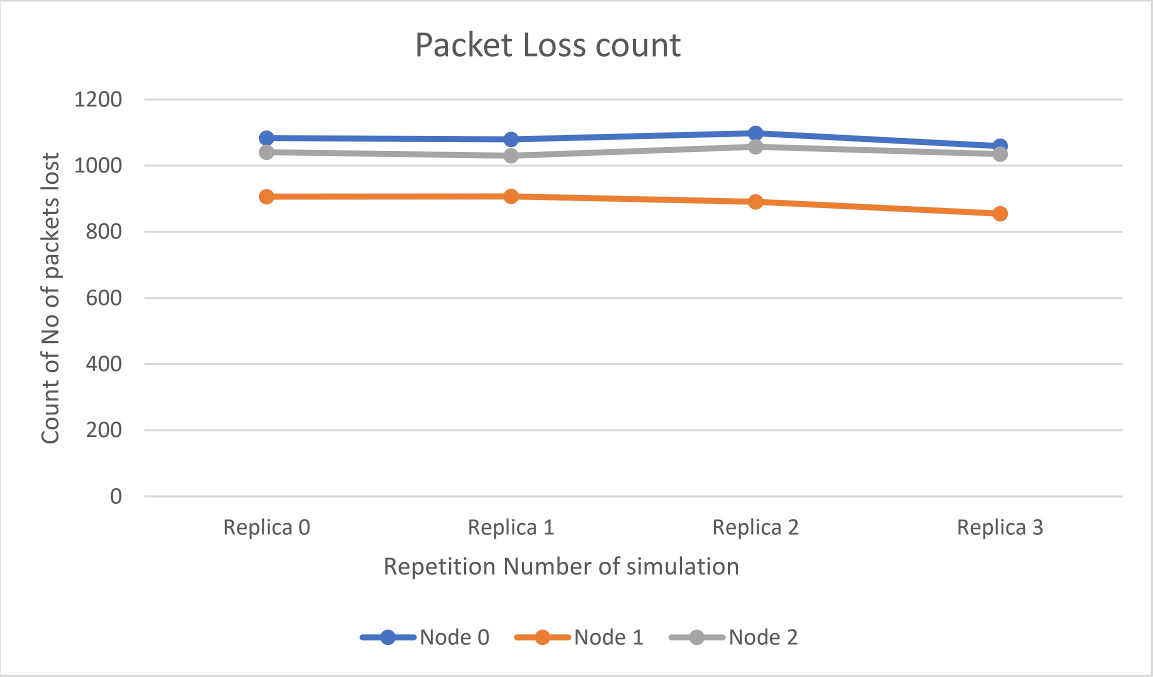
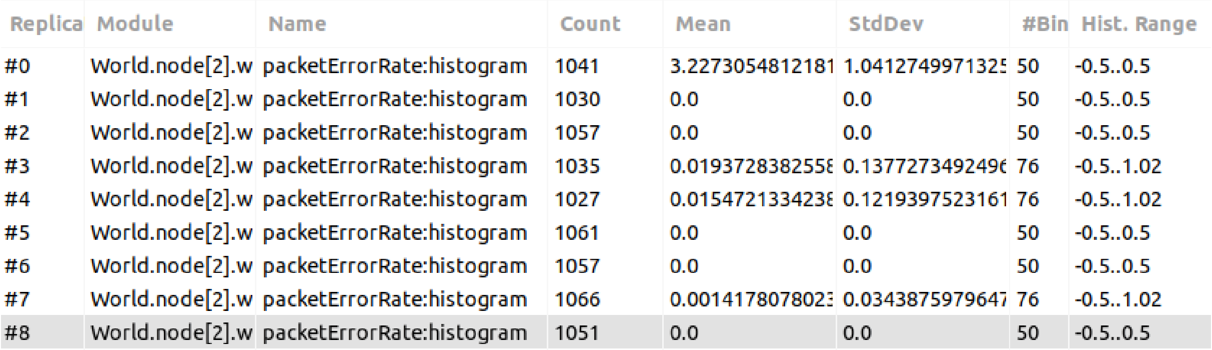


Figure 8.3.2(e) : Count of the number of packets lost from SNIR data (few points plotted)



Replica	Module	Name	Count	Mean	StdDev	#Bin	Hist. Range
#0	World.node[2].w	packetErrorRate:histogram	1041	3.2273054812181	1.041274997132	50	-0.5..0.5
#1	World.node[2].w	packetErrorRate:histogram	1030	0.0	0.0	50	-0.5..0.5
#2	World.node[2].w	packetErrorRate:histogram	1057	0.0	0.0	50	-0.5..0.5
#3	World.node[2].w	packetErrorRate:histogram	1035	0.019372838255	0.137727349249	76	-0.5..1.02
#4	World.node[2].w	packetErrorRate:histogram	1027	0.015472133423	0.1219397523161	76	-0.5..1.02
#5	World.node[2].w	packetErrorRate:histogram	1061	0.0	0.0	50	-0.5..0.5
#6	World.node[2].w	packetErrorRate:histogram	1057	0.0	0.0	50	-0.5..0.5
#7	World.node[2].w	packetErrorRate:histogram	1066	0.001417807802	0.034387597964	76	-0.5..1.02
#8	World.node[2].w	packetErrorRate:histogram	1051	0.0	0.0	50	-0.5..0.5

Figure 8.3.2(f) : Count of the number of packets lost from PER data

8.3.3 Non-Line of Sight Vehicle(NLOSv) condition

When we do not have direct communication due to vehicle/s being an obstruction. Hence, some of the transmission is affected and even if some of them are successful transmissions, they will happen in multiple hops.

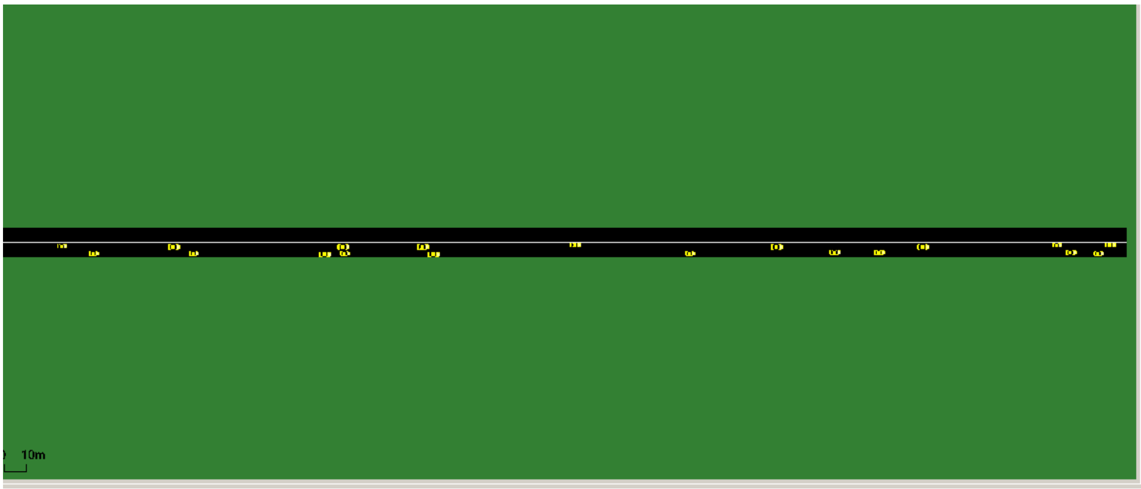
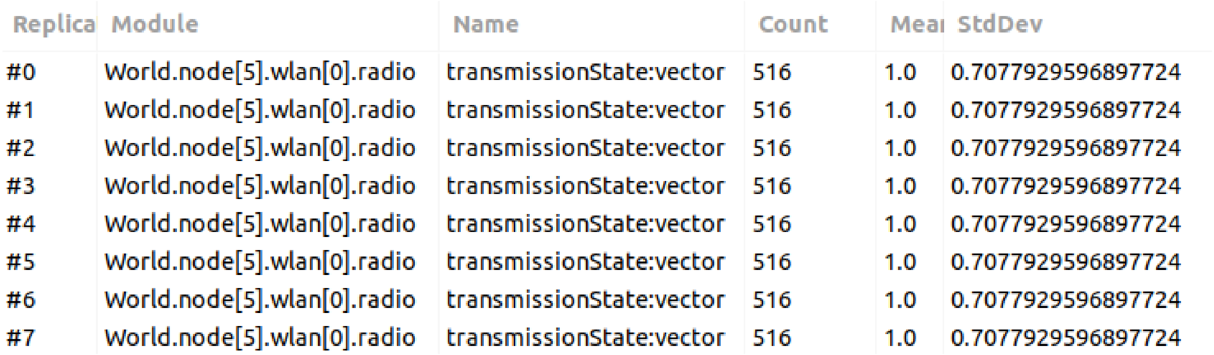


Figure 8.3.3 : NLOSv condition as visualized in SUMO

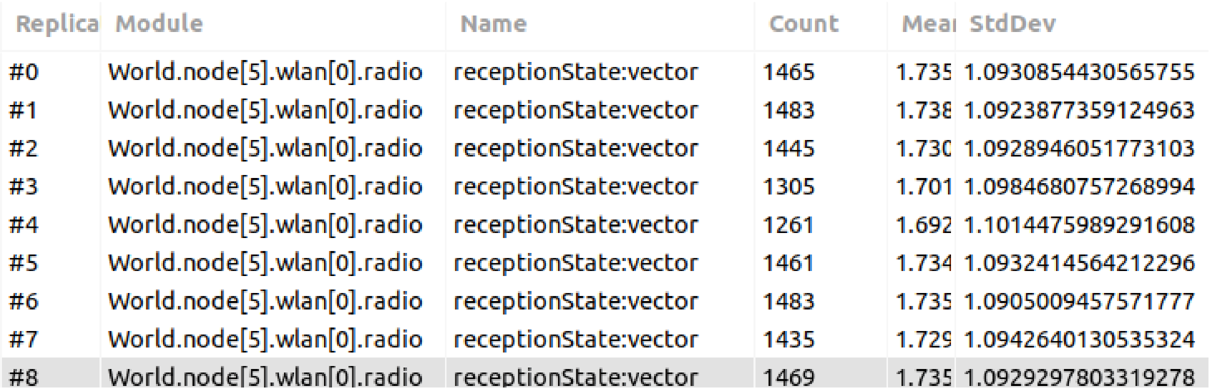
- **Transmission State:** The number of packets sent by each node is indicated here. We see no change in the number of packets transmitted and hence we conclude that the node forwards the same number of packets every time.



Replica	Module	Name	Count	Mea	StdDev
#0	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724
#1	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724
#2	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724
#3	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724
#4	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724
#5	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724
#6	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724
#7	World.node[5].wlan[0].radio	transmissionState:vector	516	1.0	0.7077929596897724

Figure 8.3.3(a) : Transmission count of packets

- **Reception Count:** The number of packets received by the node is indicated here. We see that there is lower number of packets received as is expected since some are lost due to channel conditions or fading.



Replica	Module	Name	Count	Mea	StdDev
#0	World.node[5].wlan[0].radio	receptionState:vector	1465	1.735	1.0930854430565755
#1	World.node[5].wlan[0].radio	receptionState:vector	1483	1.738	1.0923877359124963
#2	World.node[5].wlan[0].radio	receptionState:vector	1445	1.730	1.0928946051773103
#3	World.node[5].wlan[0].radio	receptionState:vector	1305	1.701	1.0984680757268994
#4	World.node[5].wlan[0].radio	receptionState:vector	1261	1.692	1.1014475989291608
#5	World.node[5].wlan[0].radio	receptionState:vector	1461	1.734	1.0932414564212296
#6	World.node[5].wlan[0].radio	receptionState:vector	1483	1.735	1.0905009457571777
#7	World.node[5].wlan[0].radio	receptionState:vector	1435	1.729	1.0942640130535324
#8	World.node[5].wlan[0].radio	receptionState:vector	1469	1.735	1.0929297803319278

Figure 8.3.3(b) : Reception count of packets(raw data)

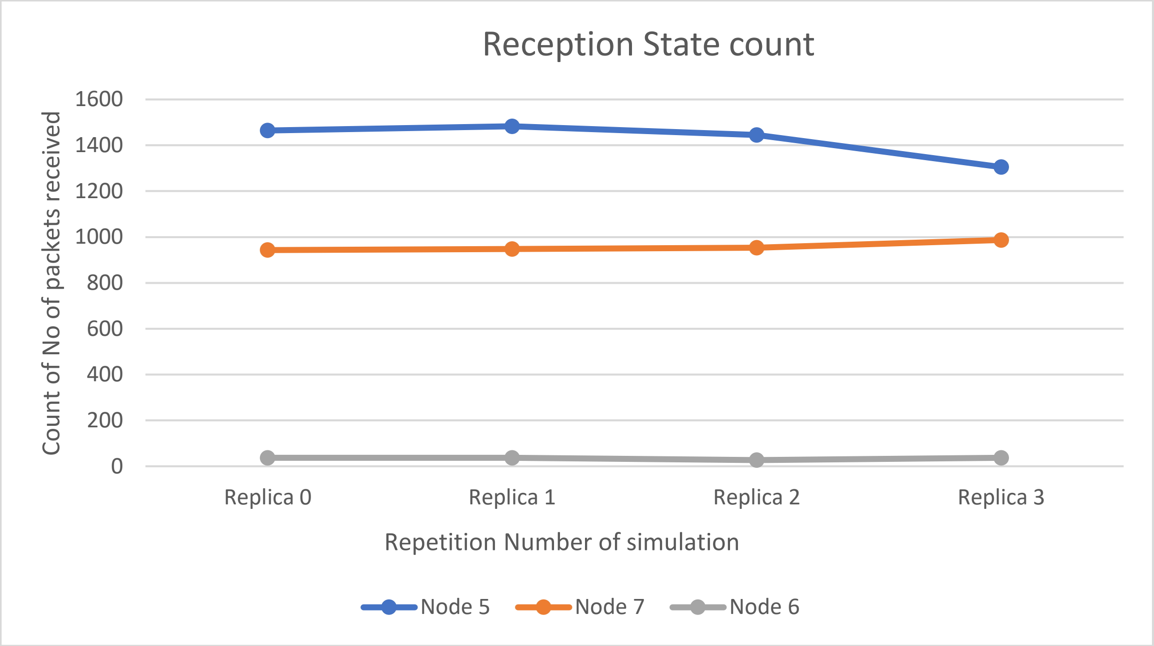
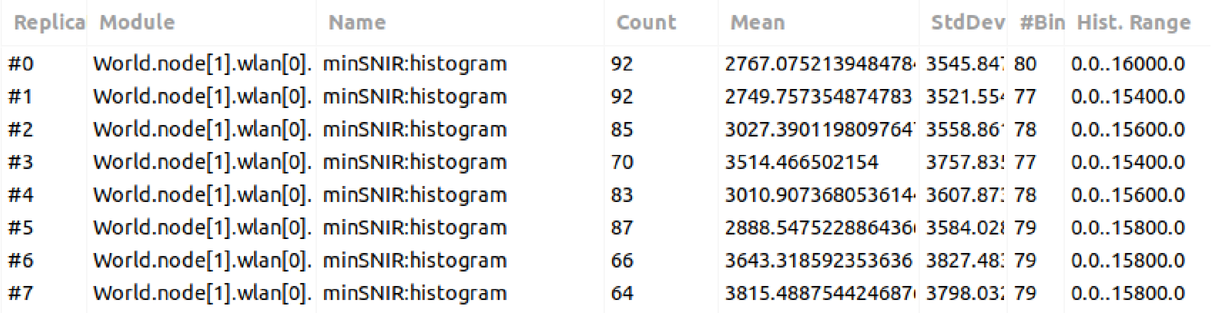


Figure 8.3.3(c) : Reception count of packets (few points plotted)

- **Packet Error Rate and minSNIR:** they are related to the reception of signals or packets at the receiving nodes. At low power rates, the transmission as well as reception is not possible hence, we have low SNIR and high packet error rates. As we go towards increasing power levels, we have lower packet error rates and better SNIR.



Replica	Module	Name	Count	Mean	StdDev	#Bin	Hist. Range
#0	World.node[1].wlan[0].	minSNIR:histogram	92	2767.075213948478	3545.84	80	0.0..16000.0
#1	World.node[1].wlan[0].	minSNIR:histogram	92	2749.757354874783	3521.55	77	0.0..15400.0
#2	World.node[1].wlan[0].	minSNIR:histogram	85	3027.390119809764	3558.86	78	0.0..15600.0
#3	World.node[1].wlan[0].	minSNIR:histogram	70	3514.466502154	3757.83	77	0.0..15400.0
#4	World.node[1].wlan[0].	minSNIR:histogram	83	3010.907368053614	3607.87	78	0.0..15600.0
#5	World.node[1].wlan[0].	minSNIR:histogram	87	2888.547522886436	3584.02	79	0.0..15800.0
#6	World.node[1].wlan[0].	minSNIR:histogram	66	3643.318592353636	3827.48	79	0.0..15800.0
#7	World.node[1].wlan[0].	minSNIR:histogram	64	3815.488754424687	3798.03	79	0.0..15800.0

Figure 8.3.3(d) : Count of the number of packets lost from SNIR data (raw data)

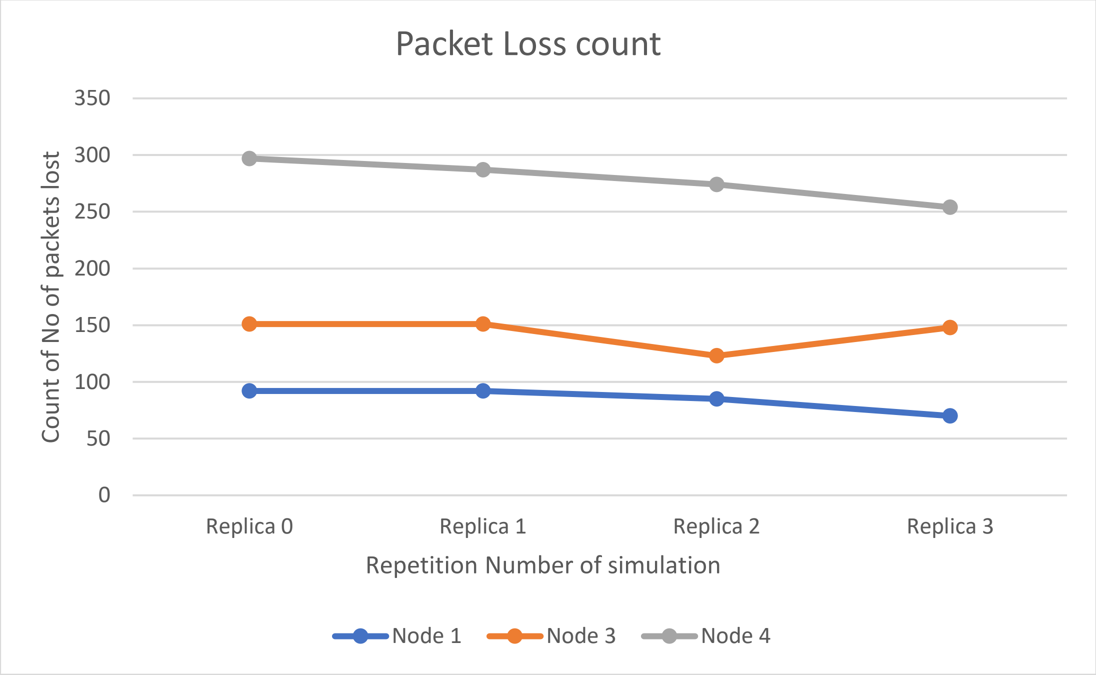
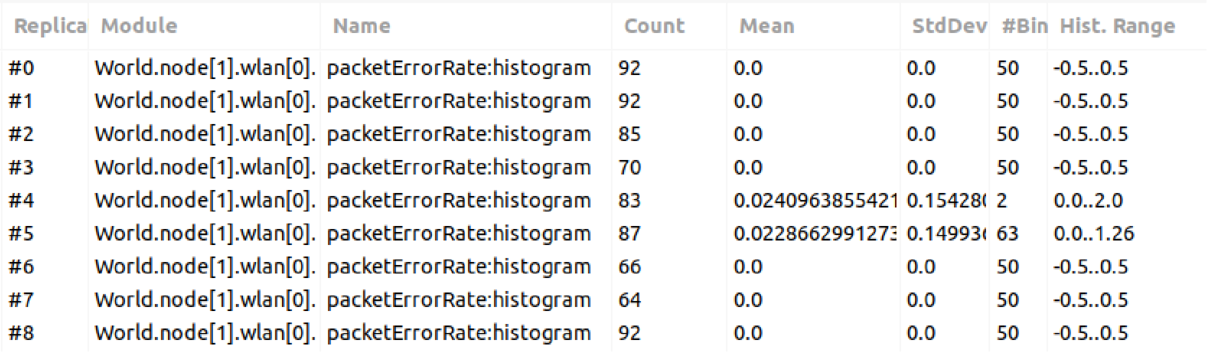


Figure 8.3.3(e) : Count of the number of packets lost from SNIR data (few points plotted)



Replica	Module	Name	Count	Mean	StdDev	#Bin	Hist. Range
#0	World.node[1].wlan[0].	packetErrorRate:histogram	92	0.0	0.0	50	-0.5..0.5
#1	World.node[1].wlan[0].	packetErrorRate:histogram	92	0.0	0.0	50	-0.5..0.5
#2	World.node[1].wlan[0].	packetErrorRate:histogram	85	0.0	0.0	50	-0.5..0.5
#3	World.node[1].wlan[0].	packetErrorRate:histogram	70	0.0	0.0	50	-0.5..0.5
#4	World.node[1].wlan[0].	packetErrorRate:histogram	83	0.0240963855421	0.15428	2	0.0..2.0
#5	World.node[1].wlan[0].	packetErrorRate:histogram	87	0.022866299127	0.14993	63	0.0..1.26
#6	World.node[1].wlan[0].	packetErrorRate:histogram	66	0.0	0.0	50	-0.5..0.5
#7	World.node[1].wlan[0].	packetErrorRate:histogram	64	0.0	0.0	50	-0.5..0.5
#8	World.node[1].wlan[0].	packetErrorRate:histogram	92	0.0	0.0	50	-0.5..0.5

Figure 8.3.3(f) : Count of the number of packets lost from PER data

8.3.4 Non-Line of Sight Buildings (NLOSb)

When we do not have direct communication due to buildings/structures being a partial/complete obstruction. Hence, some of the transmission is affected and even if some of them are successful transmissions, they will happen in multiple hops.

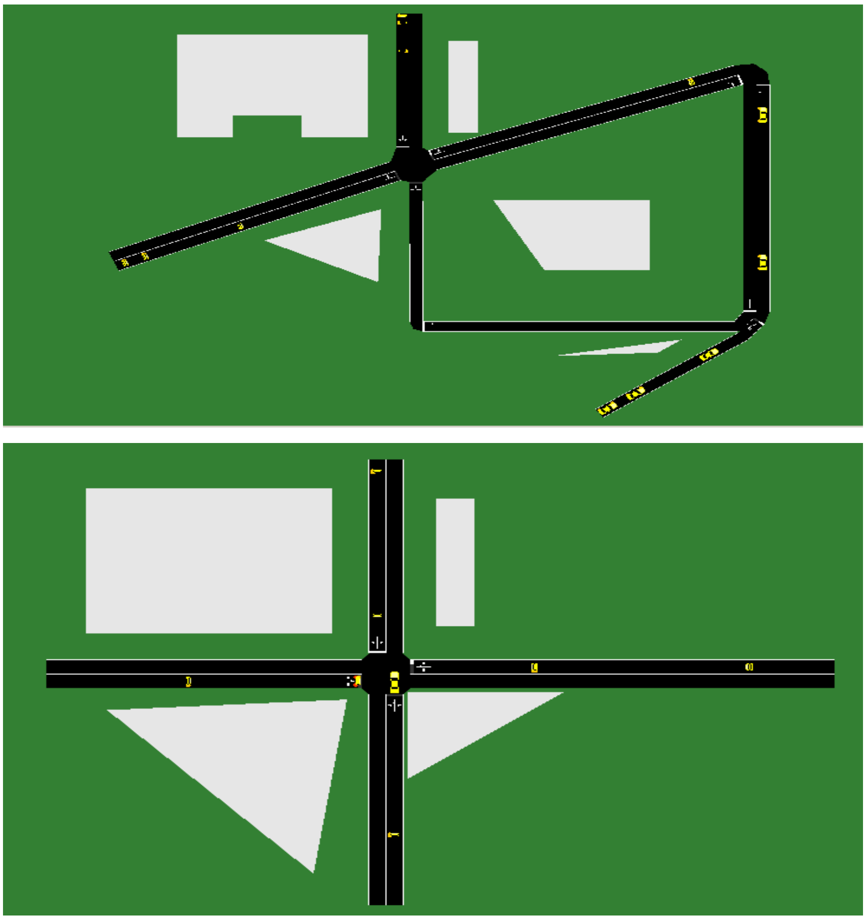
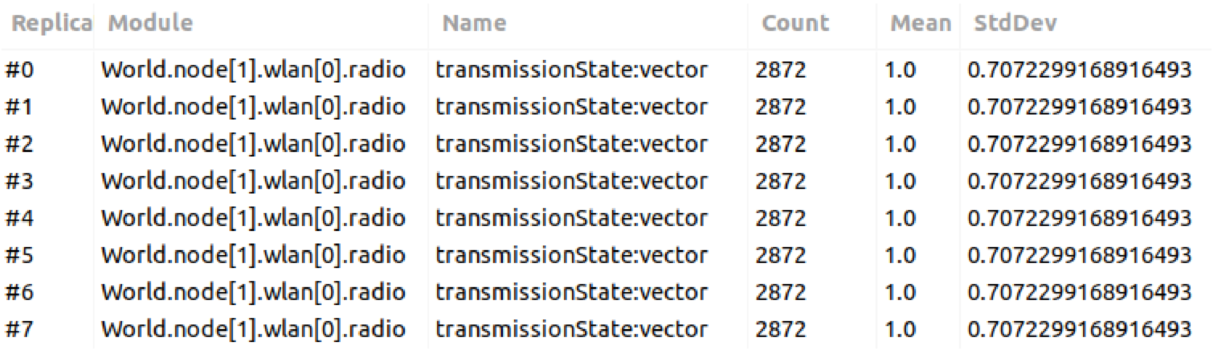


Figure 8.3.4 (a) and (b) : NLOSb conditions as visualized in SUMO

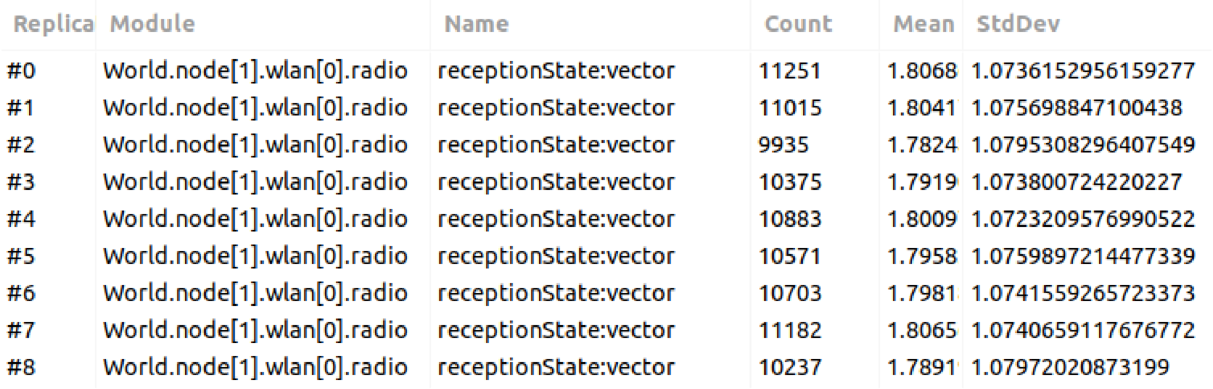
- **Transmission State:** The number of packets sent by each node is indicated here. We see no change in the number of packets transmitted and hence we conclude that the node forwards the same number of packets every time.



Replica	Module	Name	Count	Mean	StdDev
#0	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493
#1	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493
#2	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493
#3	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493
#4	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493
#5	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493
#6	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493
#7	World.node[1].wlan[0].radio	transmissionState:vector	2872	1.0	0.7072299168916493

Figure 8.3.4(c) : Transmission count of packets

- **Reception State:** The number of packets received by the node is indicated here. We see that there is lower number of packets received as is expected since some are lost due to channel conditions or fading.



Replica	Module	Name	Count	Mean	StdDev
#0	World.node[1].wlan[0].radio	receptionState:vector	11251	1.8068	1.0736152956159277
#1	World.node[1].wlan[0].radio	receptionState:vector	11015	1.8041	1.075698847100438
#2	World.node[1].wlan[0].radio	receptionState:vector	9935	1.7824	1.0795308296407549
#3	World.node[1].wlan[0].radio	receptionState:vector	10375	1.7919	1.073800724220227
#4	World.node[1].wlan[0].radio	receptionState:vector	10883	1.8009	1.0723209576990522
#5	World.node[1].wlan[0].radio	receptionState:vector	10571	1.7958	1.0759897214477339
#6	World.node[1].wlan[0].radio	receptionState:vector	10703	1.7981	1.0741559265723373
#7	World.node[1].wlan[0].radio	receptionState:vector	11182	1.8065	1.0740659117676772
#8	World.node[1].wlan[0].radio	receptionState:vector	10237	1.7891	1.07972020873199

Figure 8.3.4(d) : Reception count of packets (raw data)

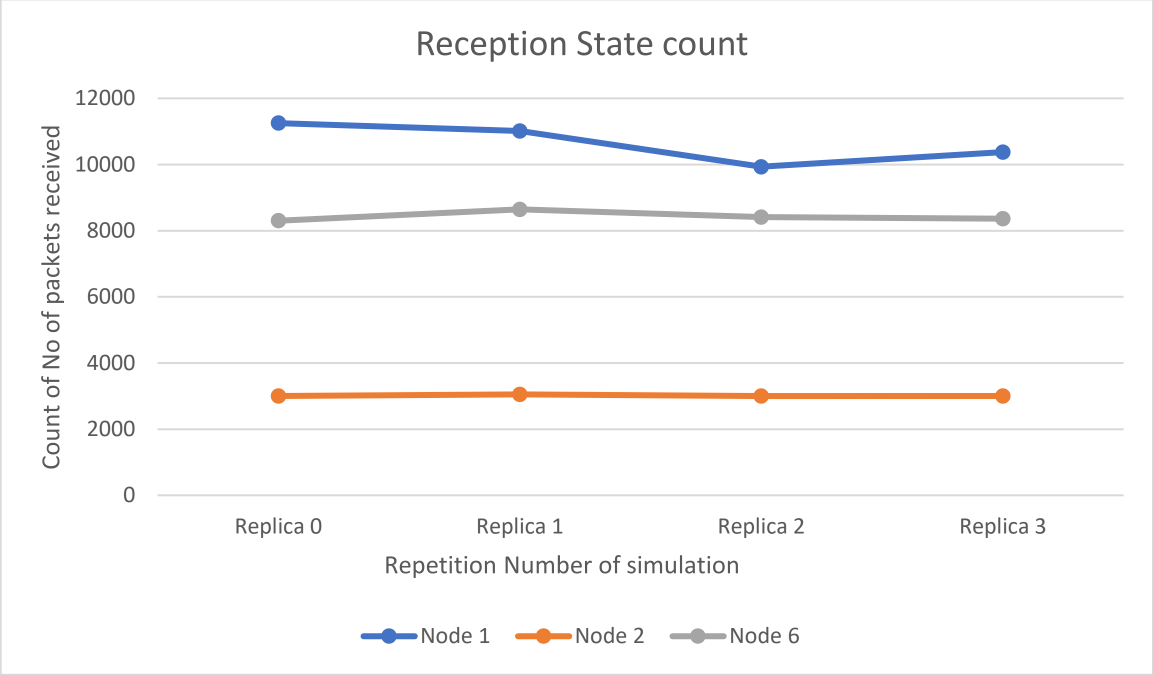
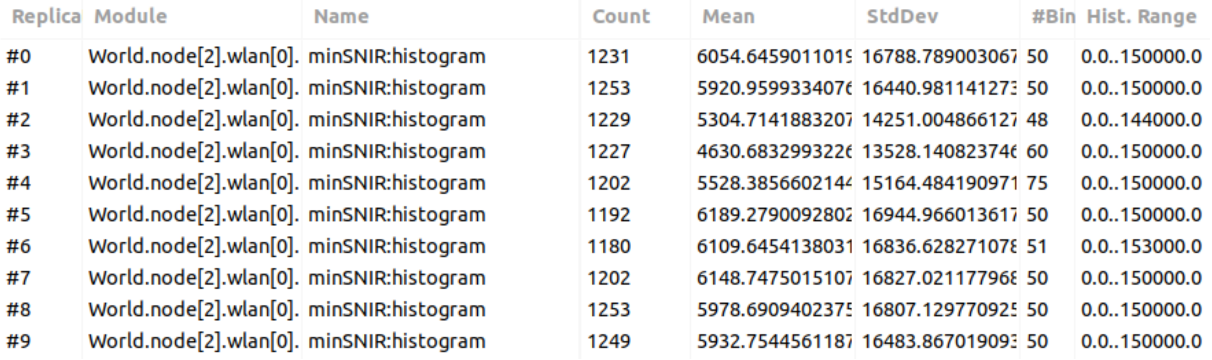


Figure 8.3.4(e) : Reception count of packets (few points plotted)

- **Packet Error Rate and minSNIR:** they are related to the reception of signals or packets at the receiving nodes. At low power rates, the transmission as well as reception is not possible hence, we have low SNIR and high packet error rates. As we go towards increasing power levels, we have lower packet error rates and better SNIR.



Replica	Module	Name	Count	Mean	StdDev	#Bin	Hist. Range
#0	World.node[2].wlan[0].	minSNIR:histogram	1231	6054.645901101	16788.789003067	50	0.0..150000.0
#1	World.node[2].wlan[0].	minSNIR:histogram	1253	5920.959933407	16440.98114127	50	0.0..150000.0
#2	World.node[2].wlan[0].	minSNIR:histogram	1229	5304.7141883207	14251.004866127	48	0.0..144000.0
#3	World.node[2].wlan[0].	minSNIR:histogram	1227	4630.683299322	13528.14082374	60	0.0..150000.0
#4	World.node[2].wlan[0].	minSNIR:histogram	1202	5528.385660214	15164.484190971	75	0.0..150000.0
#5	World.node[2].wlan[0].	minSNIR:histogram	1192	6189.279009280	16944.966013617	50	0.0..150000.0
#6	World.node[2].wlan[0].	minSNIR:histogram	1180	6109.6454138031	16836.62827107	51	0.0..153000.0
#7	World.node[2].wlan[0].	minSNIR:histogram	1202	6148.7475015107	16827.02117796	50	0.0..150000.0
#8	World.node[2].wlan[0].	minSNIR:histogram	1253	5978.690940237	16807.12977092	50	0.0..150000.0
#9	World.node[2].wlan[0].	minSNIR:histogram	1249	5932.7544561187	16483.86701909	50	0.0..150000.0

Figure 8.3.4(f) : Count of the number of packets lost from SNIR data (raw data)

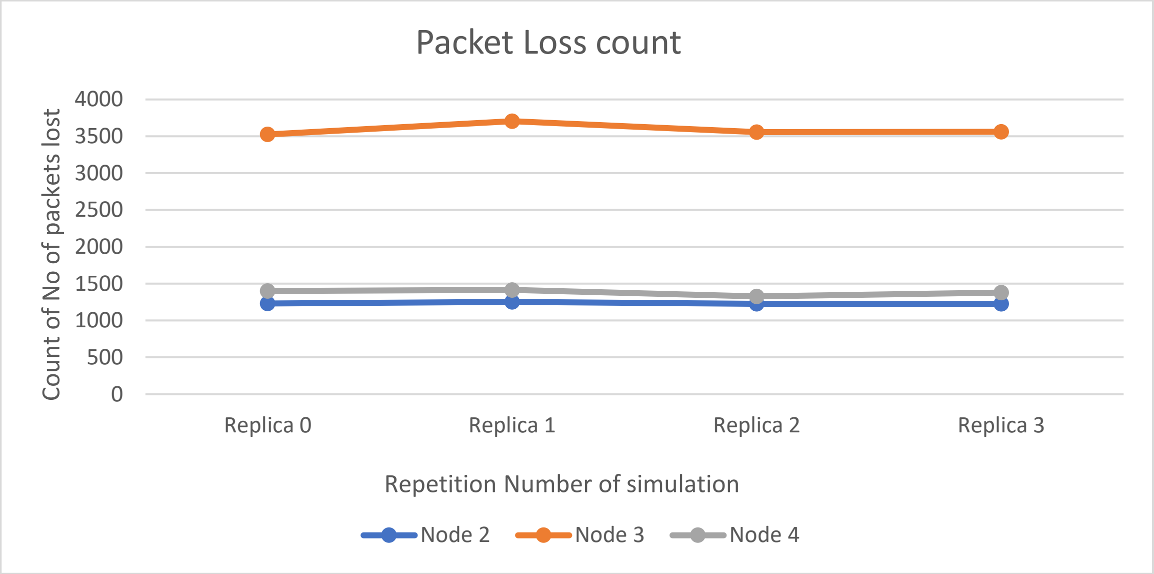
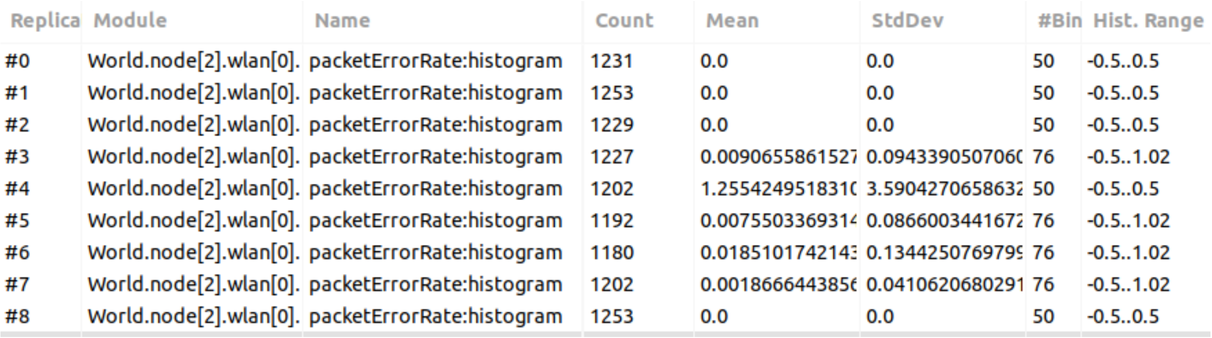


Figure 8.3.4(g) : Count of the number of packets lost from SNIR data (few points plotted)



Replica	Module	Name	Count	Mean	StdDev	#Bin	Hist. Range
#0	World.node[2].wlan[0].	packetErrorRate:histogram	1231	0.0	0.0	50	-0.5..0.5
#1	World.node[2].wlan[0].	packetErrorRate:histogram	1253	0.0	0.0	50	-0.5..0.5
#2	World.node[2].wlan[0].	packetErrorRate:histogram	1229	0.0	0.0	50	-0.5..0.5
#3	World.node[2].wlan[0].	packetErrorRate:histogram	1227	0.009065586152	0.094339050706	76	-0.5..1.02
#4	World.node[2].wlan[0].	packetErrorRate:histogram	1202	1.255424951831	3.590427065863	50	-0.5..0.5
#5	World.node[2].wlan[0].	packetErrorRate:histogram	1192	0.007550336931	0.086600344167	76	-0.5..1.02
#6	World.node[2].wlan[0].	packetErrorRate:histogram	1180	0.018510174214	0.134425076979	76	-0.5..1.02
#7	World.node[2].wlan[0].	packetErrorRate:histogram	1202	0.001866644385	0.0410620680291	76	-0.5..1.02
#8	World.node[2].wlan[0].	packetErrorRate:histogram	1253	0.0	0.0	50	-0.5..0.5

Figure 8.3.4(h) : Count of the number of packets lost from PER data (raw data)

As a final comment, we plot a graph of path loss percentage with all kinds of link types possible. What we find is the following graph :

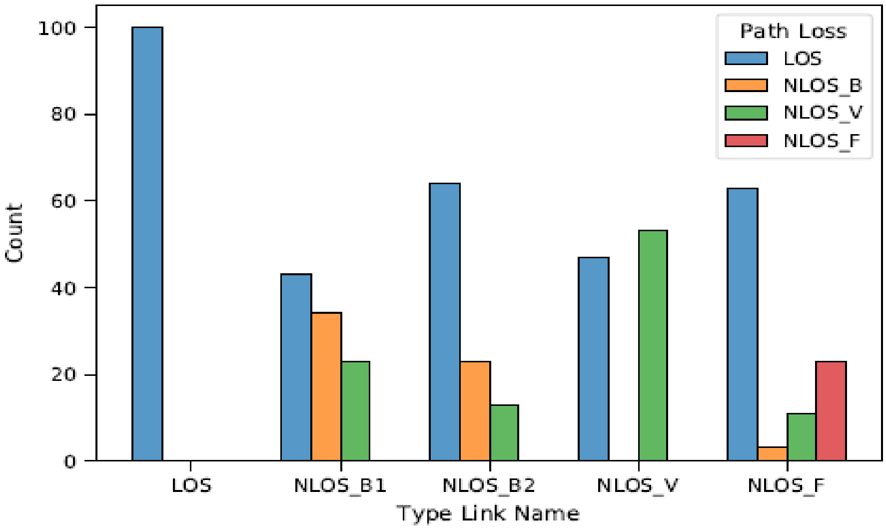


Figure 8.3.4(i) : Packet loss percentage vs link type

We see the link types on the X-axis and the percentage of LOS /NLOSx that has influenced the transmissions. We see that in LOS condition, the type of path loss possible is due to LOS only while loss due to foliage is only introduced in NLOSf type. In an NLOSv scenario, we find that the loss is none due to buildings or foliage since the design of the scenario is such that there are no such obstacles in that scenario.

9. Conclusions

The simulations are considered to mirror real-world scenarios to have an understanding the difficulties that might arise when implementing it in the real-world. It also gives us an understanding of the performance of such a system being implemented, in the sense that we can make an analysis whether we can use such a technology to implement smart vehicular technology.

GEMV^2 is a tool that can be used to test out such scenarios. Though the tool is incomplete in nature and do not capture some important parameters like Bit Error Rates, to name one, it gives a fair assessment of different scenarios and traffic conditions. The integration with SUMO gives a more visual representation of the dynamic conditions happening while an integration with an external map like OpenStreetMap tool gives a realistic impression of how it could work in actual streets. This could also open scope for future work where the scenarios can be modified in terms of its design and landscape to simulate more complicated scenarios.

The work done here gives an understanding of how dynamic conditions affect the Packet Error Rate (PER) and Signal to Noise and Interference Ratio (SNIR). We can additionally find the number of packets transferred up or down each layer and hence can understand the packet flow through the OSI model and additionally we can view what is effectively transmitted from one node to another. However, there is a scope of improvement here. When transmitting packets, it could be wise to add a flag in it to report errors, before it is passed up to the upper layers, which could enable the user to understand faulty transmissions.

We also find that there are results which include the mean, standard deviation and range in histograms which could be better analyzed to understand finer performance scenarios as well as the quality of signal. Another suggestion I would provide would be to make use of the datasets feature of OMNeT++ to record parameters like throughput or other performance parameters as required by the user.

Since GEMV^2 is open source and gives a scope to add newer functionalities for the benefit of everyone, it is of utmost importance to have a good knowledge of C++ to implement further modules to store more values and manipulate them in the results file to understand them.

The field of this study is vast and there is a lot of scope in the future as the autonomous driving and traffic management industry is becoming a key research topic that a lot of companies are looking into. With the advent of newer generations of communication technologies, it is becoming more and more easier to develop Intelligent Transportation Systems as the newer generations of technologies have capabilities in the design which enable them to be used in such scenarios.

Bibliography and Citations

1. M. Boban, J. Barros, and O. Tonguz: 'Geometry-Based Vehicle-to-Vehicle Channel Modeling for Large-Scale Simulation,' IEEE Transactions on Vehicular Technology, Vol. 63, No. 9, November 2014
2. OMNeT++ simulation manual : OMNeT++ version 5.6.1
3. OMNeT++ user guide : OMNeT++ version 5.6.1
4. INET framework page
5. R. Meireles, M. Boban, P. Steenkiste, O. Tonguz, and J. Barros, "Experimental study on the impact of vehicular obstructions in VANETs," 2010 IEEE Vehicular Networking Conference, VNC 2010, 2010.
6. Veins website
7. Artery architecture in Artery website
8. R. Meireles, M. Boban, P. Steenkiste, O. K. Tonguz, and J. Barros,"Experimental study on the impact of vehicular obstructions in VANETs," in Proc. IEEE VNC, Jersey City, NJ, USA, Dec.2010, pp. 338-345.
9. R-Trees definition and analysis in Wikipedia
10. Md. Noor-A-Rahim, Zilong Liu, Haeyoung Lee, G.G. Md. Nawaz Ali, Dirk Pesch, Pei Xiao: 'A survey on Resource Allocation in Vehicular Networks". arXiv 24th August 2020,pp 4-7
11. T. Yeferny and S. Hamad, "Vehicular Ad-hoc Networks : Architecture , Applications and Challenges," vol. 20, no. 2, pp. 1-7, 2020.
12. VANETZA website
13. W. Viriyasitavat, M. Boban, H.-M. Tsai, A. Vasilakos: 'Vehicular Communications: Survey and Challenges of Channel and Propagation Models,' IEEE Vehicular Technology Magazine, Vol. 10, No. 2, June 2015
14. B. Aygun, M. Boban, J. P. Vilela, A. M. Wyglinski: 'Geometry-Based Propagation Modeling and Simulation of Vehicle-to-Infrastructure Links', IEEE Vehicular Technology Conference (VTC-Spring), 2016
15. 'Incorporating the geometry-based vehicle-to-vehicle radio propagation channel model into the artery simulation framework for vanet applications' : Thiago Camargo Vieira, Universidade Federal do Paraná.

16. Daniel Jiang, L. D. (2008). IEEE 802.11p:Towards an International Standard for Wireless Access in Vehicular Environments. Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE Xplore.