FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Realistic Vehicle-to-Vehicle Link Simulation Using Measurement Traces

Vítor Fernandes

WORKING VERSION

Supervisor (FEUP): Ana Aguiar

Second Supervisor: Pedro d'Orey

Supervisor (Veniam) : Daniel Moura

July 26, 2018

Realistic Vehicle-to-Vehicle Link Simulation Using Measurement Traces

Vítor Fernandes

WORKING VERSION

July 26, 2018

Abstract

Vehicular Networks are a class of wireless networks which can provide connectivity between different devices among vehicles and infrastructures, allowing for several applications such as, content distribution, vehicle data collection and safety. Measuring the performance of this kind of networks is becoming an important factor in order to understand the capability to support these applications. Nevertheless, vehicular networks are not widely available and, many times, researchers have to make use of network simulators like NS-3. Network performance can be described through various metrics, including, Delay, Packet Loss and, one of the most important, Throughput, which are heavily affected by high mobility, vehicle and obstacle density. In order to enhance the performance estimation in NS-3, a model was developed to take advantage of geostamped RSSI measurements and obtain a profile of the signal for a given scenario.

These estimates are then compared to real-world values extracted from several benchmarks to validate Throughput estimation. An approach is taken to correct possible errors from these results, by analyzing error patterns. Results show that this modeling approach allows for more accurate experimentation on vehicular networks.

Resumo

As redes veículares são um subtipo de redes sem fio que tem a capacidade de conectar diferentes dispositivos, desde veículos a infra-estruturas, permitindo várias aplicações como: distribuição de conteúdo, recolha de dados de veículos e segurança. A medição do desempenho deste tipo de rede é importante para entender a capacidade das redes veiculares de suportar essas aplicações. No entanto, as redes veiculares não estão disponíveis em todo o lado e os investigadores têm que fazer uso de simuladores de rede como o NS-3. O desempenho da rede pode ser descrito através de várias métricas, incluindo Delay, Packet Loss e, uma das mais importantes, Throughput, que são fortemente afetadas pela alta mobilidade, densidade de veículos e obstáculos. A fim de melhorar a estimativa de desempenho em NS-3, foi desenvolvido um modelo que tira partido das medições de RSSI georeferenciadas, obtendo um perfil do sinal dado um determinado cenário.

Estas estimativas são então comparadas com valores reais extraídos de vários benchmarks, utilizando para validação valores de Throughput. É adotado um método para corrigir possíveis erros desses resultados, partindo da análise dos padrões de erro. Os resultados mostram que o modelo desenvolvido permite efetuar experimentação de redes veículares com maior precisão.

Agradecimentos

Agradeço à minha família, especialmente aos meus pais e aos meus irmãos, por estarem ao meu lado e pelo seu apoio incansável ao longo do meu percurso académico.

Agradeço à a prof. Ana Aguiar e Dr. Pedro d'Orey, pelo apoio incondicional que me prestaram ao longo deste trabalho.

Agradeço à Veniam pela disponibilização do material e suporte necessário à realização deste trabalho, com um agradecimento especial ao Daniel Moura e André Cardote, que me orientaram no ambiente da empresa.

Quero agradecer aos meus amigos, em especial Tiago Dias, Afonso Reis, André Pires, Pedro Rodrigues, Diogo Recharte e José Pintor, por terem contribuído com ideias e com dicas para resolver os problemas que foram surgindo ao longo do trabalho.

Agradeço à Cláudia por todo o apoio e motivação que me deu nas fases mais complicadas.

Por fim, agradeço a todos os que contibuíram para que este período da dissertação fosse bem passado e com boa disposição.

Vítor Hugo Lamela Fernandes

*"What we know is a **drop**,
what we don't know is an **ocean**."*

Isaac Newton

Contents

List of Figures

List of Tables

Abbreviations

4G	Fourth Generation of cellular wireless standards
AP	Access Point
BER	Bit Error Rate
BSS	Basic Service Set
C-ITS	Cooperative Intelligent Transportation Systems
CSMA	Carrier Sense Multiple Access
DSRC	Dedicated Short Range
DTS	Discrete Time Simulator
ETC	Electronic Toll Collection
ETSI	European Telecommunications Standards Institute
GPS	Global Positioning System
IP	Internet Protocol
IVC	Inter-Vehicle Communication
ITS	Intelligent Transportation Systems
LLC	Logical Link Control
LOS	Line of Sight
MANET	Mobile Ad-Hoc Network
MAC	Medium Access Control
MSS	Maximum Segment Size
NC	Network Controller
NLOS	Non Line of Sight
NS-3	Network Simulator 3
OBU	On-Board Unit
OCB	Outside of Context of a BSS
OSI	Open Systems Interconnection
QOS	Quality of Service
PHY	Physical Layer
PDR	Packet Delivery Ratio
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indication
RSU	Road-Side Unit
SAE	Society of Automotive Engineers
SNR	Signal-to-Noise Ratio
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-everything
VANET	Vehicular Ad-Hoc Network
WLAN	Wireless Local Area Network
WAVE	Wireless Access in Vehicular Environment
WSA	Wave Service Advertisement
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Context

The huge increase of vehicles on the road has urged the need to create applications for road safety, information, entertainment and to increase transportation efficiency. Investments are being made in road infrastructure and the automotive industry, which allow for Vehicular Networks to start appearing. A mechanism that alerts the driver whenever there is a possibility of an accident, with vehicles exchanging messages, is a safety application that can be achieved with Vehicular Networks, thus creating an Intelligent Transportation System (ITS)[1].

Veniam is building today what is called 'Internet of Moving Things', by operating vehicular networks, in New York, Singapore and, mainly, in Porto where it owns the largest Mesh Network of interconnected vehicles. Its purpose is to expand wireless coverage and, at the same time, collect a huge amount of data that can be used to obtain statistics of network performance and help improve network configurations. This data can also be used as an input to network simulations, in order to obtain realistic results.

In Porto, their network is installed several vehicles, including buses, municipal vehicles, garbage trucks, currently offering free Wi-Fi to more than 500,000 total active customers, using cellular and vehicular connectivity to allow users to connect to the Internet and at the same time, collect data logs that will be later transferred to the cloud, in an opportunistic fashion. Furthermore, a test platform (testbed) in port of Leixões (HarborNET) [2] exists and is currently being used by various members of the port ecosystem to share critical information, improve coordination and security of essential operations within that system. It is an infrastructure where the company can perform tests remotely and is highly used to collect traffic and availability statistics over time. These networks generate **production network traces** which contain limited information about the physical environment such as node position and signal strength, collected in a non-intrusive way.

Veniam occasionally performs empirical experiments to assess new network protocols, network performance, test new equipment. These experiments results contain information about network traffic, delay, rate of transmission/reception errors, among other network status information. Data from these tests is stored as **benchmark traces**, which can be used to validate network performance predictions.

Therefore, a tool is needed to take advantage of this data to allow testing new communication protocols and scenarios, providing a baseline for choosing network parameters and configurations, in order to improve Quality of Service and coverage of current networks.

1.2 Motivation

Experimentation in the area of vehicular networks is expensive, specially when in large-scale, as it requires skilled labor and time to set up, where measuring methods are intrusive and might hurt network performance, thus needing to be realized in isolated and dedicated environments. Dedicated experiments are always limited to the available testing equipment and changing between different configurations can be time consuming. For these reasons, a simulation environment is essential for studying network protocols and system architectures, as it is easy to quickly change configurations and setup different communication scenarios. However, it is hard to replicate the same real signal propagation conditions within a simulator. It is important for simulators to have channel propagation models well adapted for the environment that is being tested. By setting the physical layer conditions, based on the measurements data collected through the time, it becomes possible to replicate real scenarios conditions on simulation environment. Estimation results can then be validated with real measurements to determine the level of accuracy.

VANET face big challenges that require testing and prototyping such as efficient communications with low complexity, low delay and low number of retransmissions. Simulations are useful to research on these questions. It is important to guarantee Quality of Service (QOS) and to create a scalable and robust network [3].

1.3 Objectives

Veniam currently has data from both **production network traces** and from **benchmark traces**. This data is currently used for network monitoring and it can be useful for simulation research, with a little of processing and analysis. Taking this into account, the main objective will be working on different simulation scenarios by taking advantage of the available data, in order to enhance simulation results.

After organizing this data, a code will be developed in order to perform a trace-based simulation, using for that purpose an already existing simulation tool. In order to validate the results obtained, real world experiments data will be used, so that the estimation error can be obtained for each compared model and scenario.

To sum up, objectives are:

- Develop a model for network performance evaluation that takes advantage of Veniam's available trace data
- Provide a comparison between measurement-based and other channel propagation models
- Understand the limitations of existing models when compared to real-world measurements

1.4 Document Structure

The second chapter will present an introduction to Vehicular Networks and the state-of-the-art in terms of simulation, regarding some related work. Third chapter will present the experiment scenarios, how they were setup and which metrics are obtained. Fourth chapter will present the chosen simulation framework to be used in this work, presenting its architecture and how components are used and changed for the proposed solution. Fifth chapter will contain simulation results with comparison to values obtained from real measurements, in order to perform a validation of the work, analyze the error pattern and attempt a correction. Last but not least, the sixth chapter will contain some conclusions about the work and contributions that have been made for the subject of Vehicular Networks, providing some references for Future Work.

Chapter 2

State-of-the-art

This chapter contains a brief introduction to Vehicular Networks: current challenges, architecture, important components, applications and metrics. After that, a little of background on propagation channel characteristics and modeling approaches is presented, followed by a review on current types of simulators, going through the advantages / disadvantages of using each of them. Lastly, there is a discussion about the considered related work.

2.1 Vehicular Networks

VANET (Vehicular Ad-Hoc Networks) are a type of network which follow a similar structure to MANET, with the difference that in this case, communication is between vehicles, being characterized by a constantly changing network topology, due to the high mobility scenarios and variable vehicle density, leading to frequent disconnections [4]. It can be used to create Intelligent Transportation Systems (ITS), where vehicles can collect and distribute crucial information with a short/medium range of wireless connectivity, enabling several applications and services to drivers and passengers, becoming attractive to the research community as well as to the automotive industry. Vehicular networks are currently expected to contribute on improving road safety, increasing efficiency on communications and providing a good experience for users.

Due to its very dynamic environment, some important challenges arise when doing research in Vehicular Networks [5] :

- *Scalability* - Can be extended to the entire road network, where congestion in communications may arise with the increase of data usage and contents distribution among vehicles.
- *Heterogeneous environments* - different environments, such as highway and urban, which include different density and velocity of vehicles.
- *Partitioned network* - nodes can form several clusters isolated from each other when not in range and might not be able to communicate and disseminate status information.

- *Network topology and connectivity* - due to mobility there are constant connections and disconnections, changing the network topology. It is necessary to guarantee a fast handover process from one topology to a new one.

- *Efficient Routing Protocols* - traditional routing protocols from ad-hoc networks (e.g. MANET) are not suitable for VANET, as they rely on flooding the network to obtain routes, which is a slow method and can cause channel occupation [6]. Thus, it is necessary to optimize this protocols in order to provide smooth multi-hop connections. One solution is proposed on [7], which is a protocol based on spatial connectivity, that is, connectivity based on geographical area by creating a spatial graph of the zone, named LASP (Look-Ahead Spatial Protocol).

- *Security and Reliability* - it is necessary to guarantee that information reaches its destination, otherwise the security of individuals might be compromised, for example, when using safety applications.

- *Quality of Service (QoS)* - The level of quality of service is important most applications, in order to provide a good experience for users. Important metrics are presented near the end of section 2.1.

A Vehicular Network is composed by some essential components: OBU - (on-board unit) that will be operating inside the vehicle; RSU - (road side unit) which are usually installed near the road and provide access to internet or other information related to traffic. Other ITS stations are defined: personal station (hand held devices from people) and central station (where ITS communications are managed)[8]. For the communication types, these are the main two: V2V - vehicle to vehicle communications; V2I - vehicle to infrastructure communication [9]. In Figure 2.1 [3] is shown the most common topology, which is a hybrid of the above mentioned, where usually vehicles obtain relevant information from infrastructures (V2I) or from their neighbour vehicles (V2V) and try to disseminate it to other vehicles nearby, thus reducing the amount of infrastructures necessary. In this hybrid setup, other wireless systems can be used, such as 3G cellular, LTE, IEEE 802.11 b/g/n/ac, being deployed in OBUs and RSUs, in order to establish a connection to other devices and services, through multiple ways. This is known as vehicle-to-everything (V2X).

With this structure there can be single-hop or multi-hop connections, depending if the vehicle is close to an infrastructure or if it can communicate through other vehicles with shared connection, respectively. Multi-hop connections have the advantage of increasing coverage of the network by using several relays that repeat the signal, allowing more vehicles to be reached. However, the connection's performance can degrade when shared with many vehicles and relaying nodes have to handle many traffic, possibly leading to more energy consumption [10].

Communications between vehicles have been under development since the 1990s and a few standards defining its architecture have been released in the past years. Vehicular communication

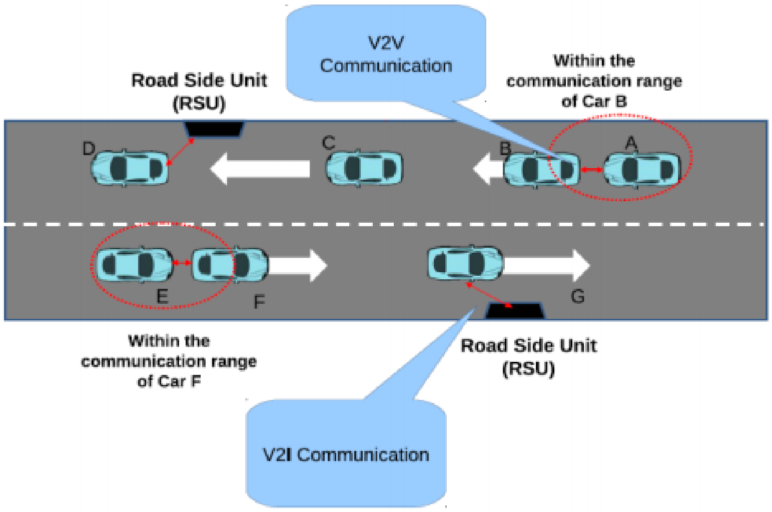


Figure 2.1: Vehicular Network architecture

is made using *IEEE 802.11p*[11], which is a standard for Wireless Access in Vehicular Environment *(WAVE - IEEE 1609.x)*[12], supporting speeds up to a range of 200-300km/h and a coverage up to 1000m. The technology operates on a *"75MHz licensed frequency band of 5.9GHz designed for Dedicated Short-Range Communication (DSRC), which is aimed at reducing latency, improving the bandwidth, providing multi-channel operations and data exchange involving high speed V2V and V2I communications"*[1]. IEEE 802.11p amendment is an enhancement of 802.11a for vehicular communications, where differences can be seen on Table 2.1[13], in order to support the high speed of vehicles, multipath environments and long range of operation. Combined with IEEE 1609.x, works as whole, being referenced as WAVE.

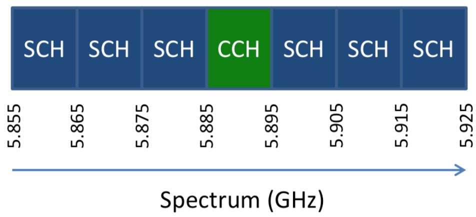


Figure 2.2: WAVE Spectrum allocation

One of the novelties in WAVE operation, when compared to standard 802.11a, is that it can establish a connection without using 802.11 authentication or association services, operating outside of the context of Basic Service Set (OCB). This allows the communication to happen quickly in rapidly varying environments. This is the reason why multiple channels are used: a control channel (CCH), typically for managing connections and one or more service channels (SCH) to send actual data to devices. WAVE devices can organize WAVE basic service sets (WBSSs), defining a specific SCH for their communication. Spectrum allocation for these channels is shown on Figure 2.2 [14].

Parameter	802.11a Value	802.11p Value	Changes
Frequency Range	5.15-5.85Ghz	5.85-5.95 Ghz	Different
Bandwidth	20 Mhz	10 Mhz	Halved
Bit rate (Mbit/s)	6,9,12,18,24,36,48,54	3,4.5,6,9,12,18,24,27	Halved
Guard Interval	0.8μs	1.6μ s	Doubled
Preamble duration	16μs	32μs	Doubled
Symbol Duration	4μs	8μs	Doubled
FFT Period	3.2μs	6.3μ s	Doubled

Table 2.1: Changes between 802.11p and 802.11a Physical Layer

The main differences lie on the frequency ranges used, which are adjacent from each other. Bandwidth and bit rate values are halved and the timing parameters are doubled. Guard Interval is doubled allowing to reduce inter-symbolic interference caused by multipath propagation effect, that is, it can distinguish replicas of the original signal that arrive after and out of sync, by having a larger value than the duration of those interfering signals.

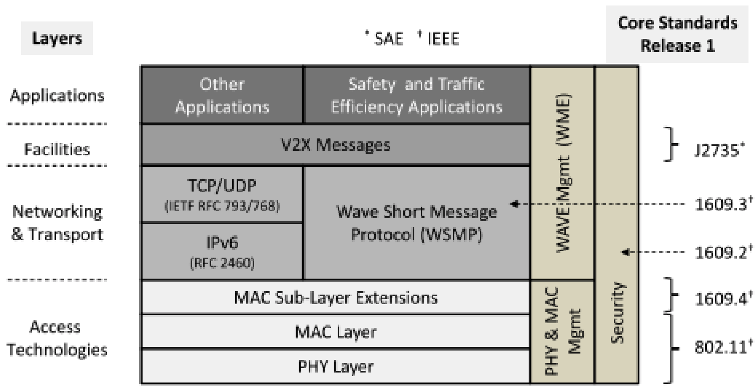
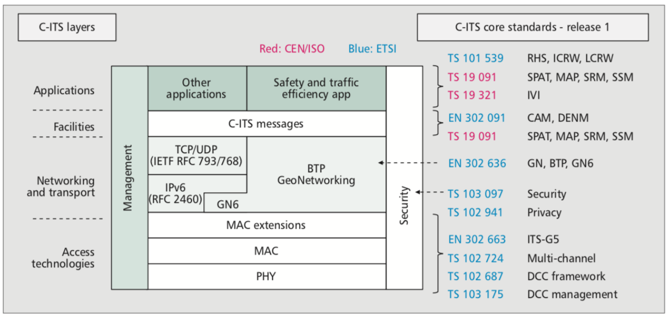


Figure 2.3: Wave communication stack with standard in different colors

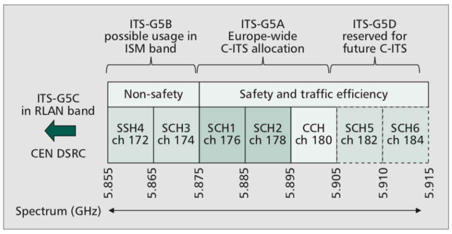
WAVE's Physical and MAC layers comply with 802.11p standard as seen in Figure 2.3 [15]. On top of those, new features are added, like *Multichannel Operation* (one control channel and multiple service channels), *Networking Services* (Data-plane services to carry traffic and Management plane services for system configuration and maintenance), *Resource Manager* (to manage use of each OBU/RSU) and *Security Services* (which are crucial and hard to implement on a distributed network like this). WSMP (WAVE Short-Message Protocol) stack is presented, as an alternative to traditional IPv6 and TCP/UDP, where the first is made for accommodating high-priority and time-sensitive communications and the latter for uses with less demanding exchanges. V2X messages are presented on standard SAE J2735 [16], being BSMs (Basic Safety Message) one of the most important, which are messages exchanged between vehicles with their status information, position and warnings for collision avoidance. The advantage of having common layers with 802.11 is that existing wireless boards can be reused and configured to operate within the 802.11p standard.

In Europe, ETSI (European Telecommunications Standards Institute) has developed ITS-G5 standard [17, 18], with its architecture defined in [8], being a standard for C-ITS which also uses 5Ghz band of the spectrum. It is derived from WAVE and is adapted to European requirements on spectrum allocation, to be shared with Electronic Toll Collection (ETC) services.

Differences lie on top of the MAC Layer, where at the Networking and Transport layer are presented two standards: GeoNetworking, a routing protocol which provides packet deliveries without a coordinating infrastructure and BTP provides a connection-less and unreliable end-to-end packet transport similar to UDP (Figure 2.4a [17]). On top of those are the Facilities layer which provides CAM messages (similar to BSM in WAVE) that are periodic and disseminate status information and DENM messages that convey safety information between vehicles. Last, but not least, Applications layer provides three groups of applications which account for collision warnings and hazard signaling.



(a) Architecture of ITS-G5



(b) Spectrum allocation for ITS in Europe

Figure 2.4: ITS-G5 Specification

Other noticeable differences from WAVE are in the spectrum allocation, where the standard WAVE control channel is 178 and ETSI ITS-G5 uses channel 180 (Figure 2.4b [17]), which may cause some confusion. Spectral power density for each of the channels does also differ and ETSI ITS-G5 divides frequency into three sub-classes: ITS-G5A (safety-road applications), ITS-G5B (non-safety road applications) and ITS-G5D (future applications).

With the increasing number of vehicles on the road, this kind of networks start to emerge with an important role on wireless communication and automobile industries. [19] enumerates applications into 2 basic types:

- *Safety related applications* - Enhancing driver safety by using notifications in real-time through the communication of vehicles, such as Intersection collisions avoidance, public safety messages and vehicle diagnostics/maintenance.

- *Non-safety related applications* - applications that provide the traveler with a set of information about traffic, congestion, navigation, points of interest. It can provide access to internet, leisure and entertainment, usually referred to as Comfort/Entertainment applications.

One important aspect for this applications to work properly is for the car to have a GPS system installed that works along with the OBU, to provide time synchronization and vehicle location. Urban Sensing applications can also be developed, by having sensors in vehicles that can provide useful information for urban monitoring, such as temperature, humidity and traffic density.

These applications have different timing and bandwidth constraints: safety applications usually require low delay, low amounts of data sent with periodic messages (in the order of hundred of bytes); Non-safety applications, depending on the usage, may have information that is delay tolerant and can be sent at any time possible, however, a lot more data is transferred in order to serve the purpose of this applications.

Some relevant metrics to evaluate the performance of Vehicular Networks for different applications are shown in the literature on [20], [21] and [22]. For instance, when evaluating routing protocols, these metrics are used:

Average end-to-end delay (E2E) - sum of times that took to send each packet divided by number of delivered packets

$$Delay = \frac{\sum_{t=0}^{t} packet\ delay}{No.\ of\ packets\ sent} \tag{2.1}$$

Packet Delivery Ratio (PDR) - percentage of packets sent that was received successfully

$$PDR = \frac{No.\ of\ received\ packets}{No.\ of\ sent\ packets} \tag{2.2}$$

Average Throughput (THR) - Results from the sum of delivered packets size divided by time

$$Throughput = \frac{No.\ of\ packets\ sent\ *\ packet\ size}{time\ spent} \tag{2.3}$$

Packet Error Rate (PER) - percentage of packets were received with error

$$PER = \frac{No.\ packets\ received\ with\ error}{No.\ of\ packets\ sent} \tag{2.4}$$

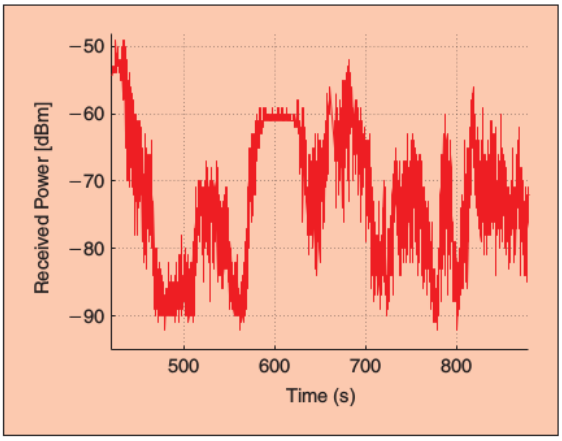
Jitter (JIT) - Delay between two consecutive packet deliveries

$$Jitter = packet[i]\ delay\ -\ packet[i-1]\ delay \tag{2.5}$$

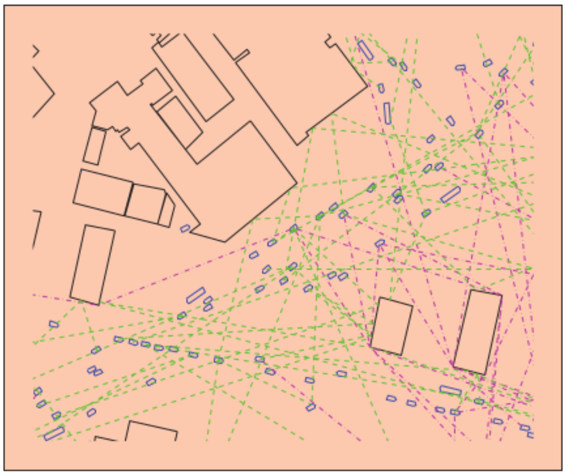
All of the metrics presented above allow to characterize the performance of a vehicular network in several domains, such as time and packet domain.

2.2 Propagation Channel Models and Characteristics

Vehicular Networks propagation channels are constantly changing its conditions, mostly due to vehicle mobility within different types of environment. Figure 2.5a [23] shows that power received on each node has a significant variation with time, due to many factors, such as low antenna heights, environment changes and high mobility of vehicles with different speeds. In Figure 2.5b [23] is represented how the signal sent by each node interacts with its surrounding obstructions, where reflected rays are represented in *green* colour and refracted rays are represented in *magenta*.



(a) Received Power variation with time



(b) Simulation setup and connections

Figure 2.5: Analysis of channel propagation in VANET

So, when doing channel modelling of this type of network, there are some relevant considerations to keep [23]:

Different Environments - buildings, vehicle, vegetation and other objects; *Link Types* - V2V, V2I or V2B, which show different propagation properties; *Vehicles Types* - different shapes of vehicles: bus, truck, car; *Objects* - sources of scattering, reflections and diffractions;

Channel modelling involves estimating all the components that interfere with the signal, either by deterministic models or by real-world measurements. The estimated model is then useful for running simulations allowing to experiment with more control and repeatability, in a cost effective and scalable way.

Usually the process of sending a signal from a transmitter to a receiver can be represented in a few blocks, as seen in Figure 2.6, which can be modeled separately, simplifying the task of modeling the whole channel environment. According to [24, 25], channel effects are composed of 3 essential mechanisms: *path loss*, *shadowing* and *small-scale fading*. The *Path Loss* component results from frequency and distance losses. *Shadowing* represents the loss due to obstructions such as buildings, hills or even other vehicles. *Small scale fading* represents the effect of multipath, which are signal replicas that arrive later may count as constructive or destructive interference.

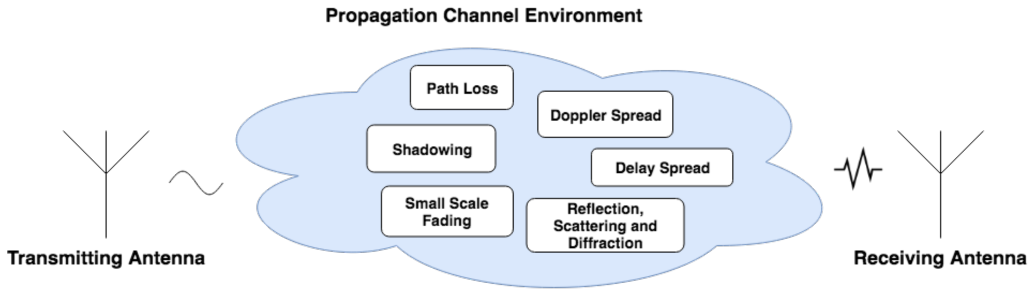


Figure 2.6: Channel Modeling components

Doppler spread is a frequency shift that happens due to the movement of vehicles and *Delay Spread* can be seen as the delay between the first and last multipath components. Along the path, the signal can also suffer from *reflection*, *scattering* and *diffraction*, which will contribute for its attenuation. Only after going through all these effects, the signal strength that the receiver obtains (RSSI) can be degraded more or less depending on the NLOS (with presence of several obstructions) or LOS (in the absence of obstructions on the surrounding environment) conditions of the environment between the nodes, respectively.

Channel Propagation Models are divided into 3 main categories, based on their implementation:

- *Geometry Based Deterministic models* - require detailed description of the propagation environment in order to accurately produce channel statistics
- *Simplified Geometry-Based models* - use maps information in order obtain geometric properties, taking advantages of channel statistics from measurements or simulations
- *Non geometry-based Models* - using measurements of channel in a specific environment to adjust parameters on path-loss, shadowing and small-scale fading.

The presented types of models show some properties that always need to be taken into account when choosing the model that best fits the needs. There can be a spatial and temporal dependency which results from the variable physical world features. The model can be extensible to different environments, for example, if a geometry based approach is used, however, it might not be scalable depending on the computational complexity. The support for different antenna configurations is also an important factor that can enhance signal reception.

Now, focusing on the different components that need to be modeled, the most widely used models for *path-loss* are:

Free-Space Path Loss (Friis) - represents the signal attenuation between two antennas when in line-of-sight conditions.

$$L_{dB}(d) = 10log_{10}\frac{(4\pi df)^2}{c^2} \tag{2.6}$$

Two-Ray Ground Reflection model - This model considers path-loss with two rays from the transmitter to the receiver: one of them goes directly and the other one is reflected on the ground, also for line-of-sight conditions.

$$L_{dB}(d) = 40log_{10}(d_0) - 10log_{10}(Gh_t^2h_r^2) \tag{2.7}$$

Log Distance path loss - This model accounts for both path-loss and shadowing effects on the signal, where $L_{dB}(d)$ is path-loss value at distance d, X is the shadowing component, A_0 is the intercept of equation 2.8 at reference distance d_0 and n is the path loss exponent.

$$L_{dB}(d) = A_0 + 10nlog_{10}(d/d_0) + X \tag{2.8}$$

It is also necessary to model *small-scale fading* (stochastic models)[26, 27], which can be done with the following distributions:

Nakagami-m distribution - a stochastic model used for modeling various signal fading effects in different environments due to its versatility, being able to model several distributions, with specific particular cases: *Rice distribution* - can be used as a particular case of Nakagami-m, when $m > 1$, *Rayleigh distribution* - is a particular case of Nakagami-m when $m = 1$ [26]. The probability density function for this distribution is shown on equation 2.9, where ω denotes the average received power and m is the fading depth parameter.

$$p(x;m;\omega) = \frac{2m^m}{\Gamma(m)\omega^m}x^{2m-1}e^{-\frac{m}{\omega}x^2}, \quad m > 1/2 \quad \omega > 0 \tag{2.9}$$

Gaussian distribution - a stochastic model also used to fit the signal variation between consecutive samples [28]. μ is the mean of the distribution, σ is the standard deviation and σ^2 is the variance.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2 / 2\sigma^2} \tag{2.10}$$

In order to make an appropriate use of the *path-loss* and *small-scale fading* models presented, it is necessary to obtain the parameters that best fit the scenario that is being studied, specially for **LogDistance** and **Nakagami-m**, as they present parameters which depend on the environment.

The most common for geometry-based models are Ray-tracing models, but they require a lot of computation effort. Another approach can be to use measurement data to build a statistical model to use in simulations, which estimates path loss separately for each communication [29].

One example of Simplified Geometry-based model is $GEMV^2$ [30], first distinguishes the path between the vehicles in 3 types: *NLOSb* (obstructed by buildings) , *NLOSv* (obstructed by vehicles) and *LOS* (without any major obstacles in the path). By using a map to obtain geometry of the location, it can use an appropriate propagation model.

In order to obtain measurements, it is possible to use a specific equipment such as channel sounder or vector network analyzer, which is installed on the vehicles to measure the channel characteristics with detail. The process consists of transmitting a known signal and comparing it with the received signal, in order to obtain an estimation of the channel conditions, as shown in [29].

2.3 Vehicular Networks Simulators

Vehicular Networks experimentation is often conducted by simulation as real experiments are sometimes hard to reproduce on the same conditions and in a large scale, which can be easily done in a simulation environment.

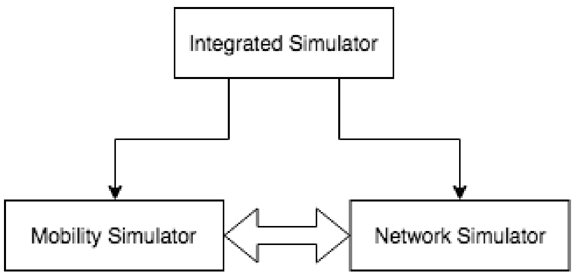


Figure 2.7: Simulator Types

The main components of VANET simulation, are separated in two categories: *Mobility/Traffic simulators*, responsible for simulating the positions of nodes, interactions with traffic lights, vehicle behaviour on intersections and traffic flow within time and space; *Network simulators*, which are responsible for simulating the network behaviour, considering aspects of signal propagation, MAC layer, routing, transport protocols and applications, handling all the communications procedure. *Integrated Simulators* are obtained when both types of simulators interact with each other at runtime, with a communication link between them, as seen in Figure 2.7. This allows to take the best of both worlds, since the two types of simulator complement each other and can make simulations more realistic, combining a complete network simulation stack with realistic vehicle movements, allowing the network to interact with vehicles decisions by exchanging messages between them.

2.3.1 Mobility Simulators and Models

Mobility models play an important role in VANET simulations, because nodes are limited to move following the roads and have to obey specific traffic rules, leading to different movement patterns. Classification of these models is made with respect to their level of detail [31, 32]:

- **Microscopic Models** - simulating at the vehicle level
- **Macroscopic Models** - simulating traffic in an aggregated total
- **Mesoscopic Models** - a combination of both described above

Microscopic models, although being more computationally expensive, can provide a good mobility detail when used in small-medium scale simulation for Vehicular Networks, being the most used for this area. There are two types to consider: *Synthetic* mobility models, which try to reproduce movement of each vehicle with mathematical expressions and random waypoints; *Trace-based* mobility models take advantage of vehicle position logs that are obtained from real environments, in order to abstract all the complexity in exchange for a good data collection with positions of vehicles along time.

Characteristics	Synthetic Mobility Models	Mobility Traces
Scalable	Yes	No
Similarity with real life movement pattern	Low	High
Time Overhead	Small	Large
Complexity	*High*	*Low*
Deployment cost	Low	High
Computation Method	Large	Low

Figure 2.8: Comparison between synthetic and real mobility models

Table on Figure 2.8[33] shows trade-offs between Synthetic Mobility Models and Mobility Traces. Overall, it is possible to understand that Mobility traces are more similar with real-world patterns, taking time and a significant cost to set up along with scalability issues, being limited a certain small region. On the other hand, Synthetic Mobility Models are better for large-scale while being more computationally complex.

There are some different models considered in literature [34]:

- **Random Waypoint** - early model which picks a random destination and velocity on a waypoint
- **Random Waypoint City Model** - combines Random Waypoint with vector street maps

- **Street Random Waypoint (STRAW)** - Limits node movement to streets and vehicular congestion

- **Manhattan Mobility Model** - mobility is generated based on a Map containing horizontal and vertical roads with 2 lanes

- **Stop Sign Model** - Integrates a traffic control mechanism, where vehicles should stop at intersections and other specific locations

Kurkhade et al. presents two more mobility models on article [35]:

- **Markov Model** - Assumes a correlation between current position and future position, calculating future locations using a probability density function, also called, Gauss-Markov Model (GM).

- **Traffic Sign Model** - Vehicles can enter a queue waiting for a traffic light to change

In order to generate more complex mobility scenarios, with data from real-world maps, variable vehicle densities, traffic coordination mechanisms, there are some mobility simulators that replicate those results with a fine grain level of detail. However, these simulators need to be calibrated and have a large set of parameters to tweak. By using a parser, this detailed traffic information can be used on network simulators.

For **Mobility Simulators**, the most popular are[31, 36]:

- **SUMO** - able to model behavior at the vehicle level offering a set of tools, allowing external applications to connect to it.

- **MOVE** - produces quick results, containing real-world databases.

- **VISSIM** - microscopic traffic simulator containing driver models for specific vehicles (bus,cars,trucks,bicycles) and also a dedicated pedestrian model.

- **DIVERT** - microscopic traffic simulator, using a model of pre-defined routes combined with random routes.

One of the most popular mobility simulators is SUMO[37], which has been coupled with NS-3 simulator and produced realistic moving vehicle scenarios [34]. Another mobility simulators of interest are DIVERT, which works both in the microscopic level using vehicles and traffic lights behavior as well as the macroscopic level, using real-world maps, being used in many VANET projects [31]. Others to consider are VanetMobiSim, MOVE, FreeSim and CityMob [38].

2.3.2 Network Simulators and Models

Network Simulators are responsible for reproducing network architecture, allowing users to test new approaches and protocols in a virtual environment. Each network component is defined as a module, so that it is easy to add/remove components and change network topology and configurations. This type of simulations provide an idea of how the network will behave. However, further experiments are always necessary in the field, to verify the behavior is close to what resulted in the simulation.

For **Network Simulators**, the most popular are[31, 36]:

- **JiST/SWANS** - Java in Simulation Time being a discrete event simulator based on Java code.
- **NS-3** - discrete-event network simulator popular among network research community.
- **OMNET++** - discrete event-based simulation with graphical interface.

One of the big problems on Network Simulators is in terms of scalability, where simulators start increasing memory usage and it becomes difficult for a computer to run simulations with a large number of nodes. In [31], this problem is solved by doing a spatial indexing data-structure, including algorithms for quick position updates and neighbor finding. These improvements can be applied on most network simulators.

2.3.3 Integrated Simulators

Integrated simulators combine the best of network simulation and mobility simulation into one framework. Most of these simulators also try to provide specific ITS applications on top of those modules, in order to have a rich environment where many combinations of use cases can be tested. However, this type of simulators require a higher configuration time, due to the complexity that is involved in joining all of these components.

For **Integrated Simulators**, the most popular are[31, 36]:

- **Veins** - uses OMNET++[39] for network simulation and instantiates SUMO for the mobility simulation. Both exchange information in real-time.
- **iTETRIS** - couples NS-3 and SUMO, providing applications for IVC and an implementation of ETSI ITS G5 standard, all connected with a centralized middleware called iCS.
- **VSimRTI** - couples 4 types of simulators: Application Simulator (VSimRTI App), Traffic Simulator (VISSIM), Network Simulator (JiST/SWANS) and an Environemnt Simulator, being optimized for ITS applications.
- **GrooveNET** - support for hybrid simulation and communication between real vehicles and simulated vehicles.

- **NCTUns** - links traffic and network together, independent simulator with some mobility capabilities, providing complete implementation of IEEE 802.11p/1609 standards.

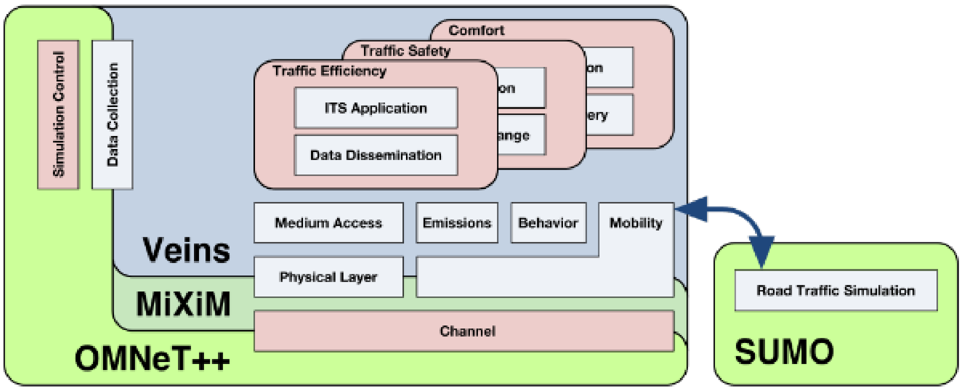


Figure 2.9: Architecture of Veins (VEhicles In Network Simulation)

In terms or architecture, as shown in Figure 2.9, these type of simulators join a lot of components interacting with each other, in order to obtain the best and most realistic combination of simulators. For instance, Veins uses OMNet++ for network simulation, SUMO for mobility simulation and some controllers that simulate ITS applications such as Traffic Efficiency, Traffic Safety and Comfort.

There are some good practices that can be adopted, in order to obtain a better set of results from simulations, presented in [22], where it is explained that there should exist a Warm up period, which means waiting a certain time frame to guarantee that all pairs are transmitting packets, in order to avoid transient effects on the beginning of the simulation. Second aspect is about the number of simulations runs and randomness, which should be high in order to obtain more representative results. Third aspect is on source-destination pairs selection, which means that pairs for communication should be selected according to their characteristics, to obtain consistent results. Last but not least, is the choice of simulation parameters, how and why they are chosen, where the number of nodes is the most common parameter to vary between simulations.

2.4 Related Work

In the area of Vehicular Networks simulation, there has been a lot of work done in order to validate models and protocols against real measurements obtained from experiments. Some other work has been done to test other scenarios with more detailed propagation models, showing some different approaches to model the environment.

Meireles et al. [40] made a study on the impact of connectivity models on large-scale simulations, showing that simulations are a valuable tool and that network connectivity is affected by the choice of parameters.

For channel modeling different approaches are taken: Cardote et al. [28] created a statistical channel model based on RSSI from a real experiment, in order to evaluate LOS propagation models and the statistical model using NS-3, obtaining results with a high concordance level, on the estimation of RSSI vs distance and PDR vs distance. Fontes et al. [41] used a trace-based approach to reproduce UAV experiments in wireless networks, by importing RSSI and GPS traces from the drones, making a comparison of the advantages of using a trace-based approach with significant enhancements when compared to pure simulation models. Cheng et al. [26] made a study based on measurements of V2V propagation channel at 5.9Ghz, using a Nakagami-m distribution fitted to data to describe signal fading process. Fernandez et al. [42] developed a Path Loss Model for LOS and NLOS conditions to use on simulators, in order to provide a more realistic model of propagation that considers the impact of objects and can produce better results when evaluating protocols and system architectures. Carpenter et al. [43] made an obstacle model implementation for NS-3 where it couples a mobility simulator and obstacle data from real world maps, obtaining a low execution time. Nilsson et al. [29] made a study on shadowing that results from vehicles concluding that it is essential to use different propagation models for these situations or even geometry-based models. Boban et al. [30] presents a geometry-based model which obtains data from real world maps and uses different deterministic propagation models for each case of obstruction, validating results with urban measurements.

Simulators have been used for many studies: Celes et al. [44] made a VANET Simulation using Calibrated Vehicular Mobility Traces by filling gaps on traces with clustering algorithms, evaluating the impact of this improvement using network metrics, showing that the existence of gaps does impact the quality of the evaluation results. Yang et al. [45] made a comparison of Vanet routing protocols with network and mobility simulation working together, showing results of various metrics with respect to vehicle speed and density. Liu et al. [46] combined a network simulator with a mobility simulator, creating a feedback loop between both, which allow to shape the simulation scenario according to the given result, in real-time. Fernandes and Ferreira [47] made an improvement of NS-3 simulator in order to better support large-scale simulations, reducing memory usage and time to completion significantly, applying for that Vehicular Networks scenarios.

Many of these previous works, offer complex approaches to model the vehicular network signal propagation characteristics and making validation of the models that were produced. For most simulation works, most of the times, deterministic models for propagation channel are used, such as Two-Ray Ground model in [28] and [34]. The use of available traces (position, RSSI, packets transferred) allows to generate a better simulation of a Vehicular Network, using a simple trace-based approach with a good amount of data.

Chapter 3

Available Trace Data

3.1 First Data Analysis

A first data analysis was made using Veniam's **production network traces** data, starting with a pre-processing phase of this data. After analyzing the given data, it was possible to extract useful information and order it, so that it could go as an input for the model that will be developed on this work.

To begin understanding available data and which useful information could be extracted from it, the first approach was to setup a script that can retrieve and process data from the database. The log files are saved in JSON format, so it is necessary to read the file, filter and extract important data. The first set of data that is analyzed is a little snapshot of network traces data (10 minutes of traces). For each trace, data contains information such as: *Node ID*, *GPS location*, *neighbours* visible for each vehicle with correspondent RSSI and the timestamp associated to the trace.

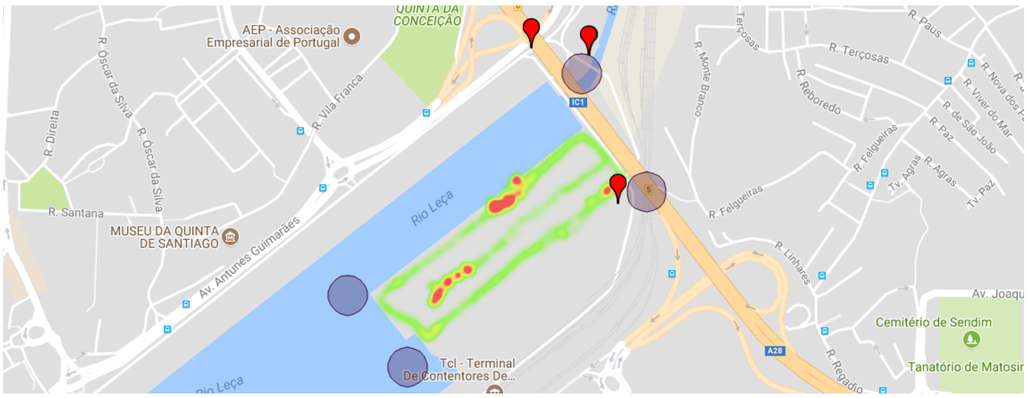


Figure 3.1: Map obtained from snapshot of HarborNet testbed data

Each trace file obtained from database contains traces for a certain region, which in the case of this analysis is the HarborNET testbed [2]. This testbed location is mostly characterized by NLOS communication paths between vehicles, as most the main obstructions are shipping containers and network nodes are composed of trucks and boats. Figure 3.1 shows the heatmap of vehicle movements, ranging from green color up to a red color, depending on the low or high amount of time that vehicles spend at each location, respectively. In this sample, there are some Road Side Units (RSU) that nodes (vehicles) can connect to and access the internet with a optical-fiber internet connection, which are depicted with a *red marker*. These access points are spread in many places and can also be considered in simulation as *DSRC* nodes with a special role of having higher connection bandwidth due to the fiber backhaul.

Several samples from this source of data are missing and there are no network performance metrics available. For that reason, results from these scenarios cannot be use for validation of the model that is being developed in this work. However, this data can be used in the future as an input to obtain predictions of how the network behaves in this scenario with a given configuration. It is important to begin with validating the model with experiments data obtained from **benchmark traces**, so that it can be applied to different scenarios with more confidence.

3.2 Experiments vs Simulation

Experiments are useful to obtain realistic results and understand the behaviour of the system in real physical conditions, thus not requiring any sort of validation and verification. However, these experiments are usually costly to set up in large-scale, requiring time and skilled labour to assure everything is correctly set. Environment characteristics can change from experiment to experiment. On the other hand, simulations are easily reproducible with the same conditions, allowing to set up large scale scenarios where its easier to setup new configurations, deploy different networks and instrument with less constraints. Several non-existing systems can also be implemented and exploited through simulation.

It is useful to use the best of both worlds by using testbed elements to get realism along with a simulator for elements of the network topology to scale. In that matter, NS-3 is able to provide several different elements to create a full network topology while allowing to use measurement traces to provide the realistic elements obtained from experiments. There is also a high-level of abstraction, which allows to easily switch and mix between simulation and testbeds, given the modularity of the components and the independence between, as it is provided by object-oriented programming.

When doing a simulation setup, it is important to configure the environment as close to the real scenario as possible, reproducing the same conditions and data flows between vehicles, to produce comparable results. Currently, in order to validate simulation results, two experiments were conducted:

The first one was made on *Avenida 25 de Abril* in *Porto*, which offers 400m of line of sight in a slope with some trees and lampposts. The second test was made on *Avenida Marginal* in *Aveiro*, which provides 1900m of line of sight coverage on a plain road, with 1 roundabout in between the path. These experiments are described with more detail on the upcoming sections.

3.3 Experimental Platform

These experiments were conducted by Veniam in order to test their platform's DSRC performance, using 2 vehicles (V2V), each one containing one OBU that collects **benchmark traces**. An RSU is also mounted in order to allow testing V2V/V2I multihop scenarios. Veniam's system uses radios that operate on the 5.9Ghz frequency band. The default value for transmit power is set to 23dBm, where vehicle heights are of about 1,75 meters each with two omni-directional DSRC antennas placed on the roof. Wifi and Cellular 4G/LTE antennas are also installed as they are part of the system, however, these technologies are not enabled nor considered for this works evaluation. Across different experiments, vehicles may have a different antenna setup, including cable losses and antenna gains, that result in significant variations of the effective transmit power output at each vehicle, which is expected due to different antenna characteristics. Radios are also transmitting two types of frames: WSA (Wave Service Advertisement) frames, which are compliant with IEEE 1609.3 standard [12] being sent with a frequency of 6Hz and variable size; NSI (*Node Status Information*) frames which are proprietary frames of Veniam's technology that serve the purpose of routing data, depending on the availability of neighbours, being transmitted with a rate of 10Hz and variable size. Positioning information is provided by GPS (Global Positioning System) receivers installed on vehicle's roof. In this work, we will be only focusing on V2V single-hop scenarios since only data from this scenarios can be used for comparison.

Parameter	**Value**
Frequency	5.9 Ghz
Channel Number	180
Bandwidth	10 Mhz
Modulation	OFDM
Channel Rate	6 Mbps
TxPower	23 dBm
Antenna Diversity	Yes (2 antennas)
Receiver Sensitivity	-95 dBm
Application Running	iPerf TCP mode

Table 3.1: Configurations on OBU and RSU for experiments

It is also known that the lowest value of RSSI that the wifi board can capture is -95dBm, which is mentioned as *Receiver Sensitivity*. Besides all the configurations for the physical unit and communication channel, as seen on Table 3.1, there is also an iPerf application setup on OBU and RSU, to allow transmitting data at maximum capacity and measure channel capacity at the given conditions.

3.4 Measurement Test Sites and Data Collected

Our focus is to begin understanding how RSSI varies with distance and then analyze how throughput varies with both aforementioned metrics. In order to understand that and adapt the simulation model to match experiment conditions as well as possible, it is necessary to explore and analyze the available logging data. Only vehicle's OBUs are capable of recording a set of information regarding sent and received data, during the several test runs. Traffic conditions such as vehicle density varied between the test sites and trials, but that kind of information was not reported during the field tests. Some additional considerations taken when conducting these tests were: assuring that the weather was sunny with low wind speeds, making sure that the cables were correctly attached to antennas while paying attention for any defects or damage.

Vehicle movements across the various trials is different, being distinguished in 3 types of tests:

- **Static** - Vehicles are both stopped at a fixed distance from each other.
- **Lap** - One vehicle is stopped and the second vehicle is moving away and back closer many times during the experiment.
- **Max Range** - One vehicle is stopped and the second vehicle is moving away until it loses connection. After that, it returns back closer to the first vehicle.

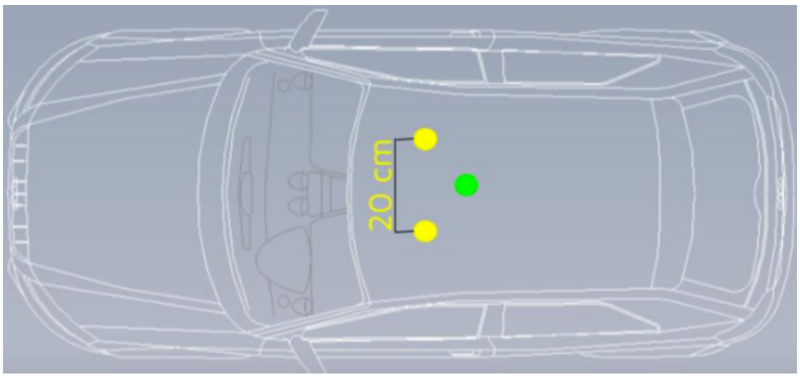


Figure 3.2: Antenna setup for each vehicle. Yellow - DSRC Antenna; Green - GPS Antenna

As seen on Figure 3.2, DSRC antennas are installed vertically (perpendicular to the ground), outside of the vehicle and 20cm apart from each other. The GPS Antenna is installed horizontally on a metallic surface in order to achieve the best signal, also being installed outside on the vehicle's rooftop. There are also Wi-Fi and Cellular (4G/LTE) antennas which are directly attached to the OBU, however, those won't be necessary for this analysis, since we are only interested in evaluating DSRC.

In terms of application setup, there is one OBU running iPerf in Client mode (Data Sender) and another OBU or RSU (depending on the test site) is running iPerf in Server mode (Data Receiver). TCP is being used as the transport protocol, which represents the worst case scenario due to its higher overhead and acknowledgement messages from the receiving vehicle, when compared to

UDP. This way, the communication link is being flooded with data and by measuring the traffic amount within a time frame, it is possible to determine the capability of sending traffic (Channel Capacity), according to this set of configurations and equipment.

Data from experiments is saved on a JSON file which contains the following fields, where a sample of the data is shown in appendix section A.1, from which we can obtain the needed metrics:

- **Neighbours** - *(NodeID|RSSI|GPS|Connection Type)* - This field provides one entry for each neighbour with information of its node ID, GPS location, signal strength and type of connection (DSRC or cellular 4G/LTE). In this case, only values with connection type DSRC will be used and signal strength is the value that the vehicle's OBU observes from its neighbour status packets that were received. RSSI value is an integer value and it is logged with an increment of 95 units (the considered Noise Floor value), that has to be removed to obtain the correct RSSI value, as shown in equation 3.1. RSSI differences are obtained from calculating the difference between the value of a given second i and the value of the previous second $i-1$, described on equation 3.2. GPS coordinates are converted to (x,y) cartesian coordinates and the distance between 2 nodes is calculated as shown in equation 3.3. **This field appears once per second, as long as there is a connection established with another DSRC equipment.**

$$RSSI(dBm) = RSSI_{Neighbours} - 95 \tag{3.1}$$

$$RSSI_{differences} = RSSI_i - RSSI_{i-1} \tag{3.2}$$

$$Distance(m) = \sqrt{(GPS(x)_{node1} - GPS(x)_{node2})^2 + (GPS(y)_{node1} - GPS(y)_{node2})^2} \tag{3.3}$$

- **Traffic** - *DSRC-(TX/RX)* - This field provides a measure of the DSRC traffic sent(TX) and received(RX) from the point of view of the unit. Depending on the direction of the data flow, TX or RX values will be used, when the unit is transmitting data or receiving, respectively. It is **logged once each 2 seconds** and for that reason, to obtain throughput per second, it is necessary to calculate the mean value shown on equation 3.4.

$$Throughput(Mbit/s) = \frac{Traffic(bytes)}{2} \tag{3.4}$$

GPS positions that are provided on data currently show a precision of 5 decimal cases for longitude and 4 decimal cases for latitude, which are equivalent to 10 meters and 1 meter granularity on these directions, respectively. This granularity is visible later on result plots of sections 3.5.2, 3.6.2 and 5.2.

RSSI values are represented as negative integer values in the scale of dBm that correspond to the signal power of the specific status packet that was received, once per second. So, this value represents only 1 sample of RSSI collected at that specific instant, not considering RSSI from other packets in between each second. However, for each packet sent there can be different signal strength values, becoming necessary to add some random process to the simulator that makes the value change within each second, to increase granularity. In order to provide a better visualization on graphs and estimation of parameters for RSSI values, when shown in plots, the values of RSSI will correspond to RSSI + 100, resulting in positive values. Since the boards only capture up to -95dBm, the worst possible RSSI in the new scale will be of 5dBm.

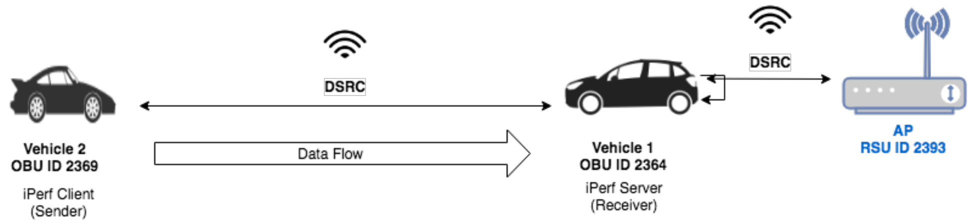
3.5 Scenario No. 1 - Avenida da Liberdade, Leça da Palmeira

Figure 3.3: Map from Leça experiment (Obtained on Google Maps)

This test scenario is composed of several elements as shown on Figure 3.3. The red marker represents the location of Vehicle 1 and the purple dots with blue outline represent the movements of the Vehicle 2 throughout the moving experiments, while the yellow markers represent the locations for the stopped experiments. The green marker shows AP's position. There are currently two iterations of this test that were made, due to some data sample problems in the first iteration.

Experiment was conducted with 2 vehicles exchanging data using full channel capacity. In Figure 3.4 it is presented the setup of the experiment and how the vehicles are positioned. Both vehicles have antennas mounted on the rooftop. Vehicle 1 stays facing the ocean and Vehicle 2 goes moving to different distances through the different experiments. Photos taken at day of the tests, show the LOS between both vehicles. In the case of this experiment, AP is used to create some multihop scenarios and other scenarios where vehicles connect to the internet and the cloud, which are not considered, as we only want to evaluate DSRC performance. Tests were performed on March 20th of 2018 (Tuesday).

There are 3 stopped tests (Static) and 2 moving tests (Lap), as shown in table 3.2. On the Lap tests, Vehicle 2 goes away from Vehicle 1 up to 300 meters and then comes back close, repeating this procedure 5 times, that is, performing 5 laps of moving away and returning.



(a) Setup scheme

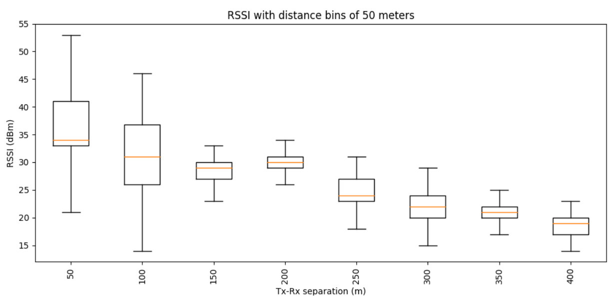
(b) Vehicle 1 - Skoda

(c) Vehicle 2 - Volkswagen

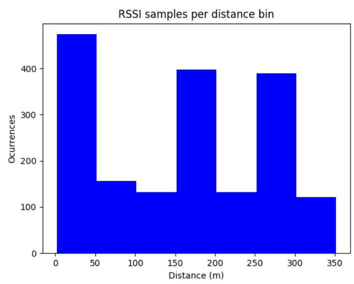
Figure 3.4: Leça da Palmeira Experiment Setup

3.5.1 Environment Analysis

This scenario is located in a parking lot, near the coast as it can be seen on Figure 3.3. The date and hour of the tests was chosen to have the less obstacles possible, that is, vehicles parked in between. Some more tests were made to try and reach higher distances, without success since the highest that is possible to reach is about 500m, where at that distance, there are already some obstacles influencing transmission. In order to avoid any NLOS environment, only distances up to about 350 meters are considered for this analysis. However there are some trees and vegetation, which can influence the signal, at about half of the length of the parking lot (approx. 150m).



(a) RSSI boxplot with 50 meters distance bins



(b) Number of RSSI samples per distance bin

Figure 3.5: Leça da Palmeira RSSI Analysis

The boxplot in Figure 3.5a shows RSSI variation with distance, with bins of 50 meters, which

Exp ID	Duration (s)	Type	Distance (m)	Start Date	End Date	Description
1	120s	Static	50m	18:02:07	18:04:08	Both Vehicles Stopped
2	120s	Static	150m	17:58:56	18:01:01	Both Vehicles Stopped
3	120s	Static	300m	17:54:07	17:56:10	Both Vehicles Stopped
4	389s	Lap	variable	18:35:27	18:41:56	Vehicle 2 Speed 10km/h
5	192s	Lap	variable	18:42:45	18:45:57	Vehicle 2 Speed 50km/h

Table 3.2: Experiments for Leça da Palmeira scenario

were chosen considering the amount of data for each bin (> 40 samples) and the maximum distance, which is roughly 380 meters. It is possible to see that the median value of RSSI on bin 3 (150m) is lower than the previous and the next bin, which can explain the presence of an obstacle. The decrease of signal strength is non-monotonic when node distance increases. These variations are due to the dominating two-ray ground reflection and possibly due to variations in the LOS, including obstructions.

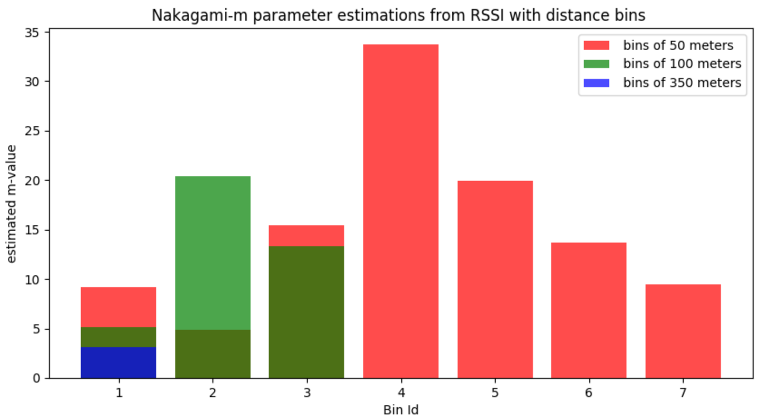
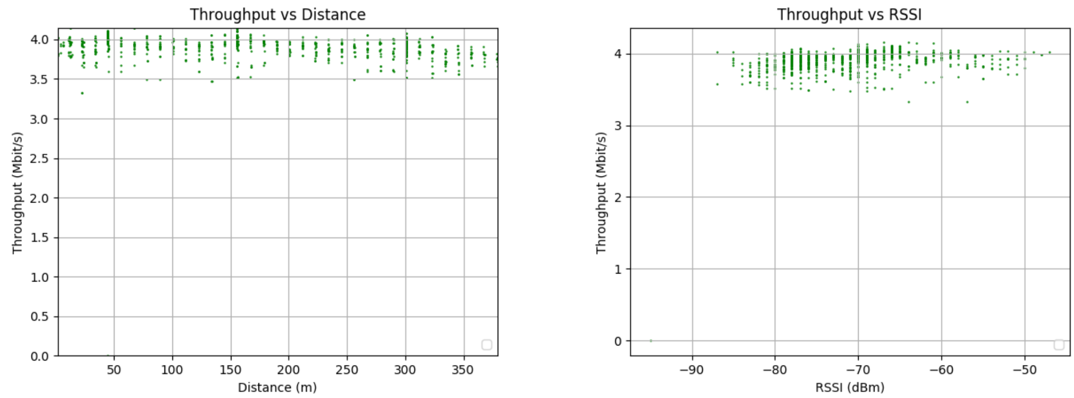


Figure 3.6: Nakagami-m parameter estimations grouped by distance bins

After dividing RSSI values into different distance bins, a Nakagami-m parameter estimation was made using the estimation procedure described in section 4.8.2. Figure 3.6 shows the obtained m-values for the different distance bin sizes, with bins of 50m (in red), bins of 100m (in green) and bins of 350m (in blue). On the x-axis, it is represented the Bin ID, which is the index of the bin with respect to the bin size. For example, Bin 2 with green color means it corresponds to the estimation on the distance range 100-200m. With this in mind, it is possible to see that the high values of m correspond to the bins with the lower RSSI variation, when compared to the box plot on Figure 3.5a, for the case of 50 meters bins. The D-statistic values obtained for these bins range from 0.08 up to 0.24, where Bin number 3 and 6 have the highest values, meaning that the estimated parameter does not fit so well as for other bins. For the case of bins of 100 meters and 350 meters, D-statistic values decreased within the same range. However, it is possible to conclude that with higher bins, a higher fitting accuracy is obtained with the cost of less granularity of the fading process.

3.5.2 Metrics Analysis

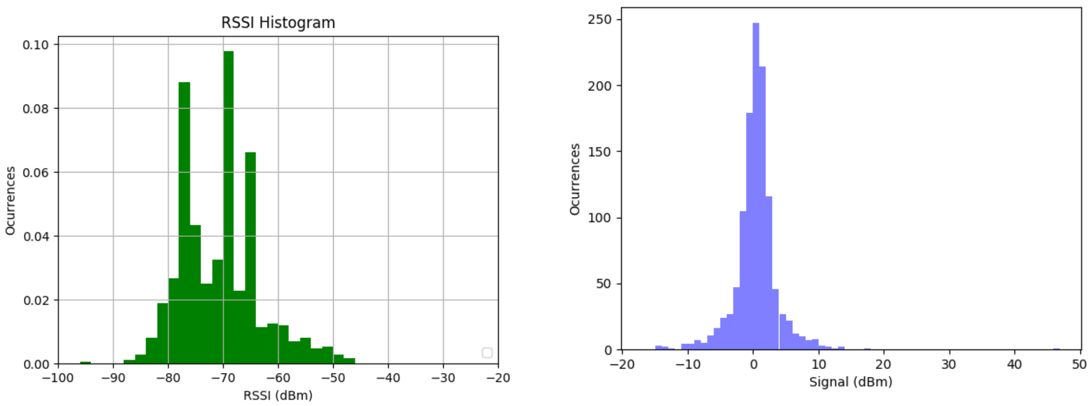
Analyzing the available data that resulted from this experiment, in terms of metrics, there are low variations due to the fact that the highest distance reached was near 400 meters, which is not enough to see some clear impact mostly due to the LOS conditions and only 2 vehicles being in communication. The metrics being evaluated are throughput and RSSI.



(a) Throughput vs Distance

(b) Throughput vs RSSI

Regarding throughput, it shows a very constant behavior if the mean value is considered, both on Figure 3.7a and 3.7b also due to the low distances obtained in this experiment. There is some oscillation in the values since all trials results were merged and different trials have different multipath and fading components. The noticeable granularity is 10m for the distance measures and 1 dBm for the RSSI measures.



(a) RSSI occurrences histogram

(b) RSSI differences histogram

Now focusing on RSSI values, the histogram on figure 3.8a shows three high bars, which are explained by the fact that there are 3 stopped tests that increase the number of occurrences of those RSSI values given the fact that, when stopped, all the multipath components and fading components keep constant and RSSI does not vary much. The remaining bars are due to the moving tests. Overall, there are few weak RSSI values (< -85dBm), producing on a low impact on metrics analyzed before. On figure 3.8b is presented the histogram of RSSI differences, that is, for each second, the different of the current RSSI value with the previous one. This metric helps us to understand how RSSi can vary from one sample to another. It shows to match a Gaussian distribution with average on zero and it is visible how multipath effect on received signal can be constructive (increasing the signal strength) or destructive (decreasing the signal strength).

3.6 Scenario No. 2 - Avenida Marginal, Aveiro

Figure 3.9: Map from Aveiro experiment (Obtained on Google Maps)

This test scenario provides a great amount of distance maintaining LOS for about 1900 meters. The map presented on Figure 3.9 shows the trajectory that Vehicle 1 takes (red dots with blue outline) and it shows the position of the AP, which is always fixed at the red marker, throughout all experiments except for the Lap experiments, where it is placed at the green marker. This scenario allows to take measurements up to the maximum range of DSRC, also providing some information about the impact on the results of using different antenna gains and setups.

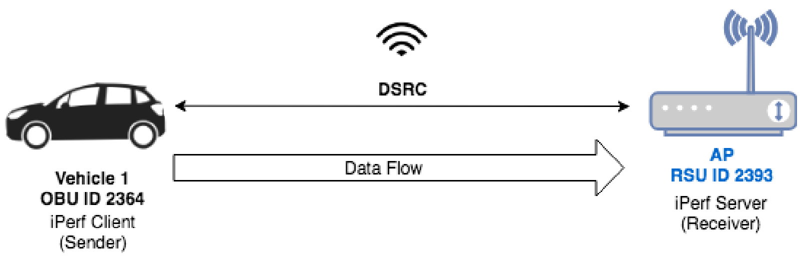


Figure 3.10: Aveiro Experiment Setup

In terms of application setup, this experiment follows the scheme presented on Figure 3.10, where vehicle 1 OBU is running as iPerf Client and AP is receiving data. In this case only data logs from vehicle 1 are available. Vehicle 1 starts moving away from the AP until it reaches the maximum range and after that it goes back near to the AP. Tests were made in 12 April 2018 (Thursday).

Exp ID	Tx/Rx Antenna Gain	Duration (s)	Type	Start Date	End Date	Description
1	5dBi / 6dBi	9m47s	MaxRange	16:28:32	16:28:52	out
2	5dBi / 9dBi	9m33s	MaxRange	16:08:08	16:17:41	out
3	7dBi / 6dBi	9m49s	MaxRange	16:42:33	16:52:22	out
4	7dBi / 9dBi	10m4s	MaxRange	15:55:50	16:05:54	out
5	9dBi / 6dBi	9min37s	MaxRange	17:12:01	17:21:37	out
6	9dBi / 6dBi	10m47s	MaxRange	16:54:29	17:05:16	direct
7	9dBi / 9dBi	10m18s	MaxRange	12:54:57	13:05:15	out
8	9dBi / 9dBi	10m9s	MaxRange	15:40:37	15:50:46	direct
9	5dBi / 6dBi	10m40s	Lap	17:46:06	17:56:46	AP is Receiver
10	5dBi / 6dBi	11m28	Lap Reverse	17:57:53	18:09:21	AP is Sender

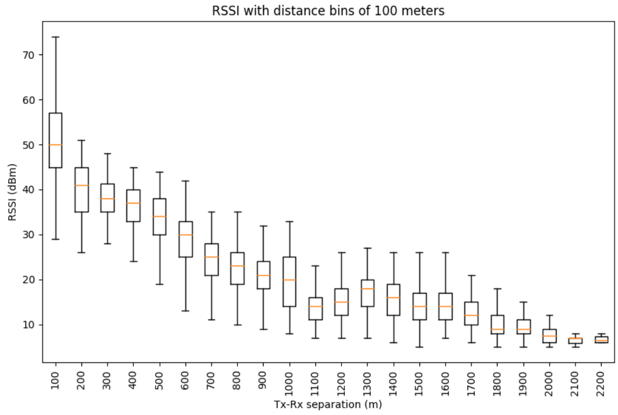
Table 3.3: Experiments for Aveiro scenario

For these tests, several antenna combinations were tested, with different installations, where **direct** means that antenna is connected with only 1 pigtail and **out** means that antenna is connected with 2 pigtails (cable is extended). There are about 10 different experiments with the same trajectory, except for Lap experiments where the AP (receiver) is placed at the middle of the path that the vehicle follows and Lap Reverse which is a special test case where AP is configured as sender (data flow direction changes). These experiments were conducted changing with different combinations of antennas both on the sender and receiver side, with each experiment lasting about 10 minutes, as it can be seen on Table 3.3. Experiments were conducted assuring that the traffic level on the road was low, during several periods of the day.

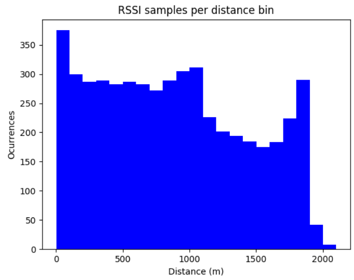
3.6.1 Environment Analysis

This benchmark scenario consists of two vehicles communicating over DSRC, while varying the antenna gains of both sender and receiver. In terms of scenario details, it is possible to see that at about 1 KM away from the starting point there is a roundabout as well as on approximately 1.9 KM, which corresponds to the distance that vehicles stop being on LOS with each other. With the Satellite view on Figure 3.9, it is possible to see that obstacles nearby are mostly houses and vegetation which are elements that can cause a higher multipath effect.

The boxplot presented on Figure 3.11a was made with all available experiments data put together, in order to characterize RSSI versus Distance, using bins of 100m, in this case for 2115 meters, results in 21 bins. It is possible to see that the RSSI median value, represented by the orange line inside each bin, tends to decrease with the increase of distance, as expected. However, between bin 11 and 12 (1100m-1200m) the value drops and raises again. This effect is due to the roundabout which is located at approximately 1100 meters from the vehicle's starting point. The histogram presented on Figure 3.11b shows the number of values available for each distance bin, it is noticeable that the starting distance (near 0m) and the end distance (near 1900m) have more samples, due to a bigger amount of time spent on those places across all experiments. On the Lap



(a) RSSI boxplot with 100 meter distance bins



(b) Number of RSSI samples per distance bin

Figure 3.11: Aveiro RSSI Analysis

experiments, AP is placed in the middle, so the distance only goes up to 1Km, which results in a higher number of samples up to 1000m.

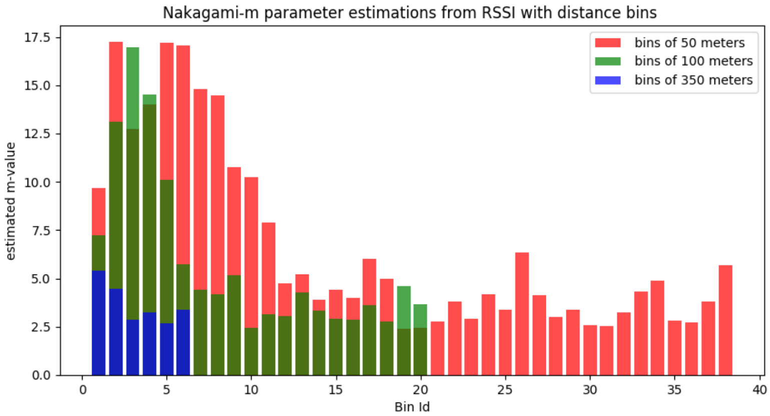
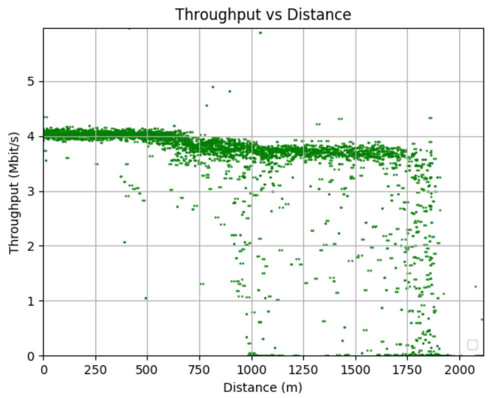


Figure 3.12: Nakagami-m parameter estimations grouped by distance bins

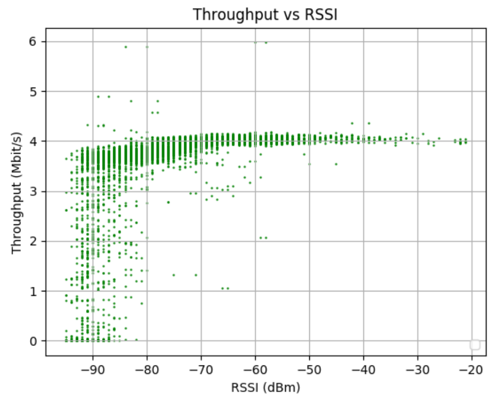
The resulting Nakagami m-parameter estimations for this scenario are presented in Figure 3.12, in a similar way of the figure shown in previous scenario. The estimations tend to result in a higher for lower distances, which means less RSSI variation and in a lower parameter for higher distances which means a higher RSSI variation. In terms of D-statistic, values for 50m bins estimations range from 0.09 to 0.26, where the lower D-statistic values are from bins with lower RSSI variations, due to the different multipath signals that affect each bin. As the bin size increases to 100m and 350m, the D-statistic values decrease, meaning that Nakagami-m fits better for larger bins.

3.6.2 Metrics Analysis

This section presents some graphs that provide a better overview of how collected metrics vary in this experiment with respect to the scenario that was described, regarding throughput and RSSI.

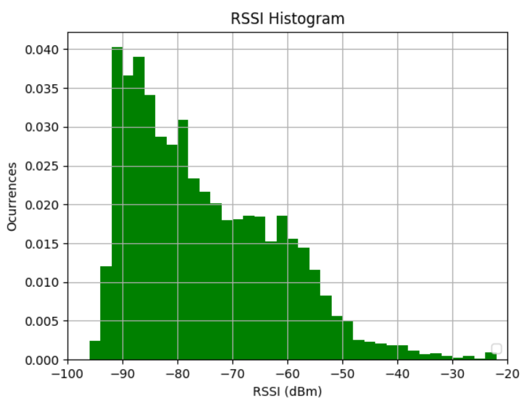


(a) Throughput vs Distance

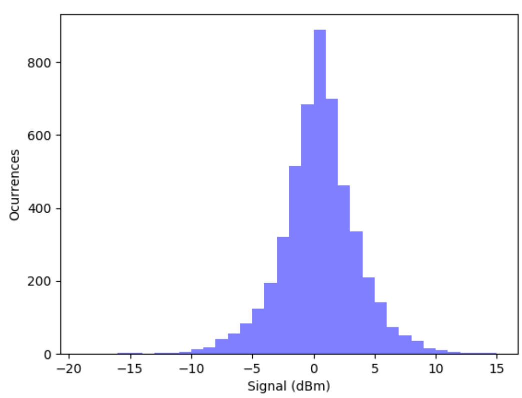


(b) Throughput vs RSSI

Regarding throughput, it is visible on Figure 3.13a that it shows a very stable behavior until about 750 meters where it starts falling, stabilizing again at 1000m, possibly due to weak signal conditions and TCP congestion window adjustment that limits througput. After 1000m, some oscillation on the values starts to appear between different experiment trials. In case of throughput with respect to RSSI on Figure 3.13b, throughput starts having some variation near -80 dBM and it drops when RSSI values reach close to -90dBm, where Noise starts to interfere with packet receptions, resulting in higher oscillations of the throughput measure. The zone near to -90 dBm is where the signal fading and multipath components make a difference and heavily impact throughput performance.



(a) RSSI occurrences histogram



(b) RSSI differences histogram

Regarding the RSSI distribution for this scenario, on Figure 3.14a is shown the relative frequency (all bar heights sum to 1) of each RSSI value with bins of 2 dBm. It is different from Leça scenario, with a lot more data resulting from all experiment trials merged. The tendency shows to have more samples near the lowest RSSI values and it shows a wide range of values from -23 dBm to -95dBm due to the vehicle 1 starting point being near vehicle 2 and the ending point being out of range of vehicle 2. This higher amount of lower values is explained by the environment objects and vegetation that are presented starting on 1km distance, that result in a higher impact on RSSI make it go closer lower values, maintaining them until the maximum range. On Figure 3.14b it is shown the difference between consecutive RSSI values, this time a Gaussian function with more spread values, which show that this environment has stronger components that make the signal vary, such as the elements described in the environment analysis.

Chapter 4

Simulation Framework

This chapter introduces the chosen simulation framework, explaining the reasons why NS-3 was chosen instead of other modeling approaches. NS-3 components are presented, including changes made on those components and how they are used together in order to obtain results that replicate real experiments as close as possible. A global overview of how the whole system works together is also shown, focusing with more detail on each component, on the procedure used to obtain simulation results and how to compare them to real world measurements.

As mentioned before in section 1.3, the main objective is to obtain a model that takes advantage of Veniam's available trace data to produce network performance indicators as output. Several possibilities to create this model have been considered: **Analytic** model, **Machine Learning** model and **Simulation**. The chosen approach is **Simulation** which relies on a set of models, that can take input data, for different aspects such as signal propagation and transmission error, among others. It also contains the full network stack, which allows different network topologies to be set. Furthermore, applications can be configured to generate data flows which are measured to retrieve network performance metrics such as Throughput.

Simulation approach was chosen for the following reasons:

- **Network configuration is changeable** - when in a simulation environment like NS-3, it is possible to easily add or remove protocols (TCP, UDP, IPv4, ..), taking into account the full protocol stack and their interactions. Machine learning and analytic models only work for reproducing the same features of the **measured data**.
- **Testing different data flows** - It is possible to change data flows between nodes running on the network, for example sending a file with a high or low sending rate, that will produce different results, which can be compared.
- **Network congestion** - When several vehicles are communicating at the same time, there are specific routing protocols and congestion mechanisms that are triggered and can be

simulated as well, where machine learning or analytic models can't due to the high level of abstraction of these factors.

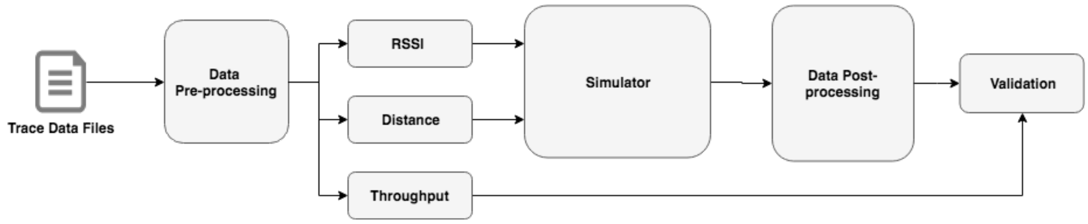


Figure 4.1: High-level system architecture

The architecture on Figure 4.1 presents the overall system architecture, which is composed by various blocks. Trace data files contain data samples which will be processed on the *Data Pre-processing block*, resulting in three outputs: **RSSI, Distance and Throughput**. The first two are inputs that go into the *Simulator*. The output is then forwarded to a *Data Post-processing* block which organizes data into several plots, making it ready for *Validation*, where results from simulation and real measurements are compared, in terms of Throughput, given the available data from **benchmark traces** that allows to perform this step.

4.1 Chosen Simulation Framework

There are several tools that can simulate VANET up to the date. Apart from Mobility Simulation that creates Mobility Models for the simulator, it is necessary to have a Network Simulator along with it to simulate the complete network stack. However, we don't always need to use a mobility simulator, mobility traces are also used when the aim is to replicate a certain mobility pattern, with the disadvantage that mobility cannot be changed nor be influenced by the network messages. Integrated Simulators join the capabilities of these two types of simulation, providing a feedback loop between network simulation and mobility simulation where information is exchanged, bringing some advantages when IVC protocols are likely to influence driver behavior and therefore change the traffic pattern on the fly, as discussed in [48]. An overview of the most popular simulators for each category is presented on Figure 4.2.

Although Veins shows an interesting architecture as an integrated simulator, by coupling OMNet++ and SUMO, it is more difficult to setup, due to the dependencies required to work with all functionalities. This work is more focused on channel modelling and not so much on mobility, since mobility is already given by the traces which provide more realistic vehicle movements. As such, there is no need to use an integrated simulator or a mobility simulator. It is essential to have an easy to configure network simulator that provides a good implementation of the needed standards for Vehicular Network communications.

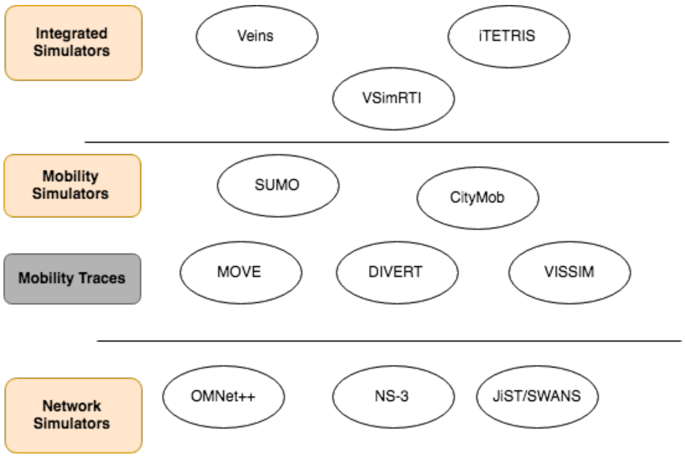


Figure 4.2: Simulators Overview

Regarding network simulators, [49] shows that the most efficient simulators are OMNeT++, NS-3 and JiST for simulating large-scale networks. However, these simulators need to be coupled with a realistic mobility model, which can be provided in run-time by a mobility simulator. Therefore, a feedback loop between a network simulator and a mobility simulator can be a good option when mobility traces are not available.

NS-3 has been widely used as it is one of the most complete network simulators and has the capability of easily changing its code to import real-world position traces, which can be filtered and fill possible gaps before being loaded, as presented in [44].

Taking into account all the characteristics of the presented simulators and the requirements for this work, **NS-3** was chosen among several reasons: it is an **Open Source Simulator**, which allows to easily change code if needed and adapt to new scenarios, it is built with **high modularity**, which means that most protocols and standards belong to a class and have less dependencies on other modules, making it **easy to integrate new models** and use them along existing ones, due to its architecture. It is a **well documented** simulator, where the documentation is frequently updated as new features appear, being **well accepted** by the research community, as a great number of research projects have been made using NS-3. It provides a **decent execution time**, allowing to quickly obtain results and it is **easy to configure a network and debug**, containing several tracing and logging applications that allow to identify possible misconfigurations.

4.2 NS-3 Architecture

NS-3 [50] is a discrete-event network simulator written in C++ and Python, being a free software licensed under GNU GPLv2 license, available for research and development, providing several modules and supporting many standards of 802.11, TCP/IP, along with several physical layer and propagation models as well as integration of mobility simulators. At the time of this work, the current version used is NS-3.27.

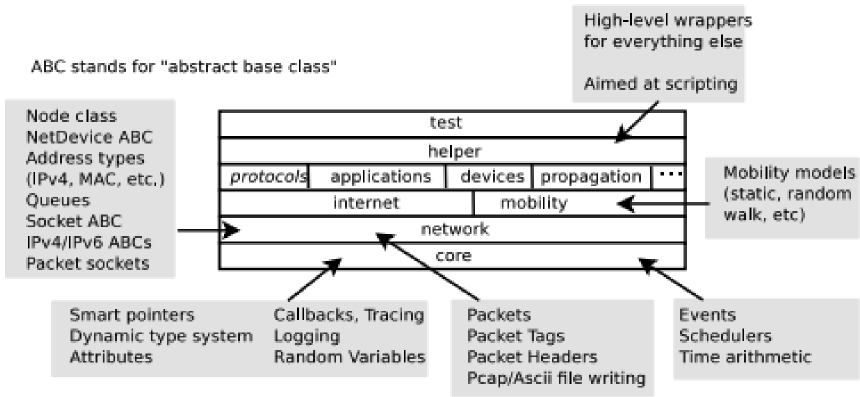


Figure 4.3: NS-3 software organization

The organization of NS-3 software on Figure 4.3 contains several components that are connected within simulation run-time and are accessible with the use of a global referencing system consisting of: smart pointers, attributes and a dynamic *type* system. Considering this fact, it is easy to start building a network topology from scratch. The *Node* instance represents the physical node of a network, such as a computer with input/output interfaces. Each node can contain a *NetDevice*, representing the wireless board that communicates with the *WifiChannel* which is configured with a *PropagationLossModel* to account for signal propagation losses. To configure the node on the network, a MAC and IP *Address* are required. In order to setup the remaining layers, each node can have different applications running, such as TCP or UDP *Sockets* that allow to transfer data packets to other nodes. To account for node's positions, each node is configured with a *Mobility* model. When the whole network is configured with the needed amount of nodes, there are some functions that allow registering the data flows and following the path of each packet to obtain some statistics, which are known as *Callbacks*, *Tracing* and *Logging* components. All of the components mentioned in this text create simulation events with *Simulator::Schedule*, which will make a call to a function happen in a given simulation time. After all these events are scheduled, simulation starts with *Simulator::Run*. During the first seconds of simulation, the network is being properly configured with messages exchanged between the nodes. This means that packet generator and consumer applications must start a few seconds after the simulation started, to guarantee that the network is ready to start sending data.

Packet flow from one node to another within the simulation environment can be seen in Figure 4.4.

NS-3 can currently simulate with a maximum simulation time resolution of one nanosecond.

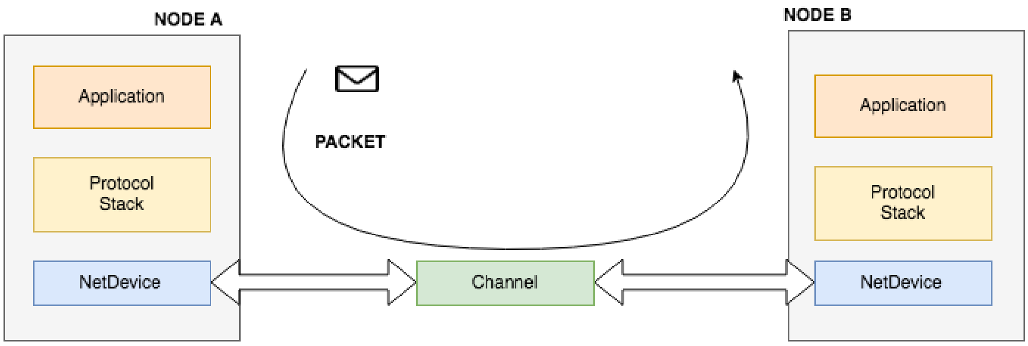


Figure 4.4: Packet Flow on NS-3

When sending packets from Node A to Node B, the trajectory goes across several layers. First, the packet leaves the *Application* layer, then it goes through all the *Protocol Stack* (Transport Layer, Network Layer, Mac Layer) with the respective headers being added to the packet as well. After that, it reaches the *NetDevice* component which corresponds to the physical device that is transmitting data, which is coupled to a channel that is common between for all nodes. The packet gets sent through the *Channel* and once it reaches the other node, the same procedure is made backwards, removing protocol headers and reaching the application layer to complete the process. In this case, the protocol stack of interest is *WAVE/802.11p* amendment, which NS-3 offers support for with a specific module that is being developed by an active community of contributors.

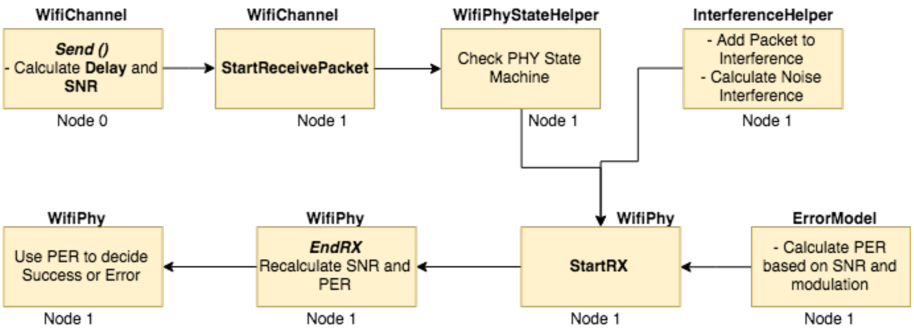


Figure 4.5: Packet Reception at Physical Layer

Figure 4.5 shows how a packet sent from Node 0 to Node 1 is processed at the physical layer. The originator node (Node 0) sends the packet and calculates the SNR value from the *PropagationLossModel* and the transmission delay value from *PropagationDelayModel*. The method *StartReceivePacket* is triggered on the receiver node (node 1), which will add the current packet to the interference list and check for the current state of its Physical Layer, to verify if packet can start being received or, otherwise, reception is cancelled. *StartRX* is called to receive the first chunk of the packet, where a PER value is calculated to decide if this chunk will result in errors. If the first chunk is successful, then *EndRx* method is called at the reception of the last chunk, to evaluate if the complete payload is received, by calculating a new SNR and PER value for this chunk. If the transmission is successful, then the packet will be passed into the MAC Layer.

To make building a network topology simple, NS-3 provides a set of *Helpers* that contain configuration operations and applies them on a *Container* which stores a set of objects. This allows to turn the topology description more high-level, making it easier to read and understand.

In order to reach created objects without having to use a pointer for each, NS-3 also provides an *object namespace*, where each object can be accessed with a specific string (e.g. "/NodeList/1/DeviceList/0/$ns3::WaveNetDevice/PhyEntities/*/$ns3::YansWifiPhy/"). In this example, it is accessed the Physical Layer of device 0 from node 1, that contains a *WaveNetDevice* that is linked to a *YansWifiPhy* object, which represents the physical wireless radio of the device. There are specific operations on this objects such as *Config::Set* which allows to change an attribute of a given object and *Config::Connect* which allows to define a method to be called (*Callback*) on a specific simulation event, such as, a packet drop. This methodology is helpful for tracing events on simulation run-time and obtaining statistics. A higher level of tracing can be obtained by using ascii and pcap trace sinks, which export packet information to an external file that can be viewed after simulation is complete.

On the next sections, each relevant component of this simulator will be explained in more detail, regarding the changes that were made and which features were used to build the desired simulation system. To summarize what follows, the first component to be used was mobility model, on which the position traces were loaded, followed by the physical layer model, on which a new model was developed to import RSSI traces from data and model other signal effects such as multipath fading. WAVE module was used to setup a WAVE compliant stack with the exchange of WSA (Wave Service Advertisement) messages, to mimic the execution of Veniam's system. On top of that, a TCP application is configured to create a data flow which matches the experiments. Last but not least, statistics collection is created on the simulation environment to allow obtaining the needed metrics for further validation with real results.

4.3 Mobility Model

In order to load vehicle mobility, a python script was created to parse GPS data into a NS-3 trace file format. Currently, there is a *MobilityHelper* in NS-3 that allows to import mobility traces that are on NS-2 trace format. GPS data is converted to that format and is directly associated to nodes on NS-3. Each node has a sequential ID, so it is necessary to map the real NodeID of the vehicle with the simulation NodeID, in order to know which real node corresponds to the simulation node. For example, node 0 in simulation can be associated with a real node ID of 2254.

Coordinates are given in x,y,z format along with a given speed, that allow to import the real vehicle movements to the simulation scenario with the granularity of 1 second given by the available data. In order to convert GPS coordinates into a (x,y) cartesian format, the approach taken was to take advantage of a spatial reference system (SRS) given by European Petroleum Survey Group (EPSG), which allows to convert GPS coordinates to a (x,y) format limited to a certain region. Python's pyproj library was used to handle those conversions. First step is to convert to EPSG 4326 which is the reference conversion for all world coverage. Since we are only interested in Portugal and Spain region, there is a further conversion into EPSG 2062, containing a region that is shown on Figure 4.6. In order to further scale down the coordinates value, *xmin* and *ymin* are calculated and subtracted from data, representing the lowest x and y coordinate value of the measurement data, respectively.

Figure 4.6: Region corresponding to EPSG 2062

In order to validate this part of the simulation, there is a useful tool available that can be used along with NS-3 to check the movement of the vehicles. The tool is called NetAnim [51], which at the time of this work is on version 3.10.8. During simulation time, an XML file is generated logging the movement of each vehicle along with packets sent. There is also a maximum packet size which can be changed, preventing the simulation to generate large XML files and avoid the visualization framework to crash. If X and Y coordinate values are too large, nodes cannot be seen easily using this tool, due to the high amount of space generated. For that reason, subtraction of *xmin* and *ymin* are made while parsing data.

Initially, used data was retrieved from a database that corresponds to Sardão - Leixões (HarborNet testbed) which contains about 20 vehicles moving around within time and is shown in Figure 4.7. After parsing the data to NS-3 format and visualizing it on NetAnim, it was possible

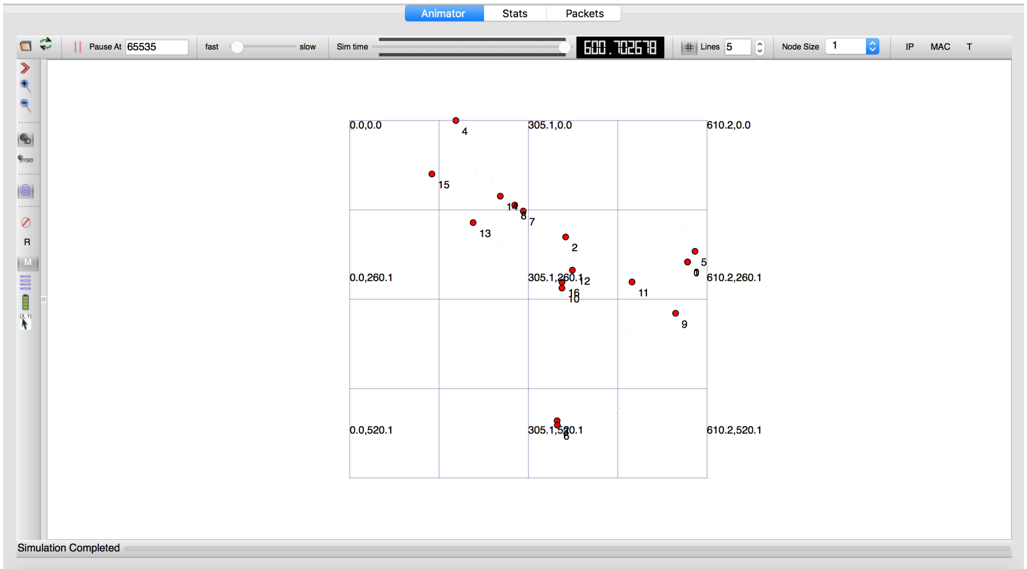


Figure 4.7: NetAnim Simulation Visualization Tool with data from HarborNet

to conclude that all the mapping of positions was correctly done and matched the database values for vehicle positions. Another test that allowed to validate this step was calculating the distance between the 2 vehicles along the time and verifying the same output on the simulation status, for each second.

4.4 Physical Layer Models

In NS-3, there are currently 3 main models that represent the physical layer: *YansWifiPhy*, *SpectrumWifiPhy* and *PhySimWifiPhy*.

The first model, *YansWifiPhy* is a physical layer model based on "Yet Another Network Simulator" [52] on which errors and losses happen at packet level. That is, for every received packet there is a division in chunks, a SNR is calculated along with a PER (packet error rate), drawing a random number at the reception of the first chunk, to decide if the packet will start being received and, at the reception of the last chunk, to decide if the packet is received successfully or with errors.

SpectrumWifiPhy and *PhySimWifi* consider errors at the bit level. These models take more computational power and are particularly useful to model interference between different technologies by considering interference on adjacent frequencies.

In order to investigate possible simulation gains of using packet-level simulation or bit-level simulation, some simulations were made along the way. After importing mobility to the simulation, it was possible to do a mapping of RSSI resulting from the simulation with the RSSI resulting from the real data measures. In order to do that, the simulation script was configured to send 1 packet of 100 bytes for each second of the traces, in order to obtain the value of RSSI for that packet and log it into a file. These values obtained allow us to compare the RSSI output of each physical layer model with real measurement samples, for each given time instant (1 second in this case). However, no major differences were found when using bit-level simulation compared to packet-level simulation, regarding their RSSI output.

This work is only focused on DSRC technology which operates on 5.85-5.95Ghz Band. However, Veniam's system takes advantages of other technologies such as LTE (operating on 2.1GHZ band), 802.11a (operating on 5.15-5.85Ghz) and 802.11b/n/g (operating on 2.4Ghz band) which do not overlap with DSRC and cause no interference. This way, a simple model which only considers interference from other DSRC nodes fits the purpose and there is no need to model cross-technology interference during simulation, saving some computational effort on simulation run-time, especially on large scale simulations. For that reason, **YansWifiPhy** is suitable for the purpose of this work.

Parameter	**Value**
Frequency	5.9 Ghz
Channel Number	180
Bandwidth	10 Mhz
Modulation	OFDM
Channel Rate	6 Mbps
RxGain	5 dBi
TxGain	5 dBi
EnergyDetectionThreshold	-96 dBm
CCAMode1Threshold	-99 dBm
RxNoiseFigure	4 dB

Table 4.1: Physical Layer Settings for current implementation of 802.11p

Values used to configure the physical layer according to the real system are described in Table 4.1. The parameters *CCAMode1Threshold* and *EnergyDetectionThreshold* in *YansWifiPhyHelper* are defining the signal detection levels on the wifi board, which are explained at section 4.4.3. *RxGain* and *TxGain* represent antenna gains for the receiver and transmitter, respectively. *RxNoiseFigure* is a constant value that represents the losses in the receiver due to non-idealities and is different for each receiver. Its default value is 7dB, resulting on experimental data. For this simulation, it was set to 4dB which is the value that fits best for this receiver's data. This change results in a shift of the minimum RSSI value to receive a packet of -95dBm, as imposed by the current wireless boards.

The Physical Layer itself works as a state machine, that is, it can only be at 1 state at each time, with a total of 6 possible states, as seen in Figure 4.8. *RX* and *TX* states represent that the

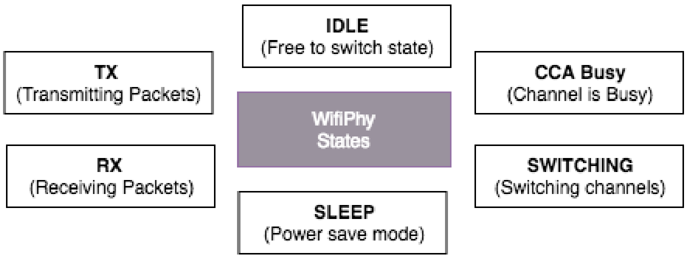


Figure 4.8: Physical layer states

physical unit is currently receiving or sending and this state can be acquired when a packet comes with a signal strength above *EnergyDetectionThreshold* and state is *Idle* or *CCA Busy*. If the signal comes below that threshold, but above *CCAMode1Threshold*, it will count as interference and will raise *CCA Busy* state. Other signals below these two thresholds are just added into Noise. *Switching* state is obtained when Physical layer switches to another wifi channel and *Sleep* state is only reachable if energy options are activated, which are not considered for this work.

4.4.1 Propagation Models

There are currently several propagation models that can be coupled with the channel environment. As seen in the literature, described in section 2.2, there are different types of models of propagation losses, including path-loss models and fast-fading models.

Currently, the implemented propagation loss models in NS-3 are:

- **TwoRayGroundPropagationModel** - This model here uses Friis equation 2.6 for distances up to the cross-over distance and uses TwoRay equation 2.7 for distances above.
- **LogDistancePropagationModel** - It is implemented using the equation described in 2.8.
- **ItuR1411LosPropagationLossModel** - It is a model suitable for frequencies in range of 300Mhz up to 100 Ghz, which uses estimated parameters retrieved from measurements data [53].
- **NakagamiPropagationLossModel** - Models fast-fading effect on signal by using random variables, reproducing the Nakagami-m distribution with a given m parameter. It is implemented as described in equation 2.9.

Each of these models provides initialization functions and can be added together in a chain of propagation models, that is, for example, TwoRayGround model can be used along with Nakagami model. By passing through each of these models, the function *DoCalcRxPower* is called to obtain an RSSI value for each packet given the transmit power (*TxPower*) and the positions of the originator and destination node.

The work on [41] uses a similar approach on loading signal and position traces from nodes. It is a good starting point to make the simulation changes that are necessary.

Contribution: Based on the aforementioned work, a new model was developed, being applied to the context of Vehicular Networks. Since the RSSI value for each link is available, it can be loaded into the simulation environment, being identified by 3 parameters: **origin node**, **destination node** and **second of simulation**. This way, the **position** of both nodes and the **signal strength** value obtained from traces are the values that the simulator will use when *DoCalcRxPower* is called, which is the function that returns the RSSI value to the parent class *PropagationLossModel* that forwards to the channel layer, to decide what happens to each packet sent at a specific second. This new version of the model was named "MBasedPropagationLossModel" where *MBased* stands for *Measurement-Based* and its class diagram is presented on Figure 4.9.

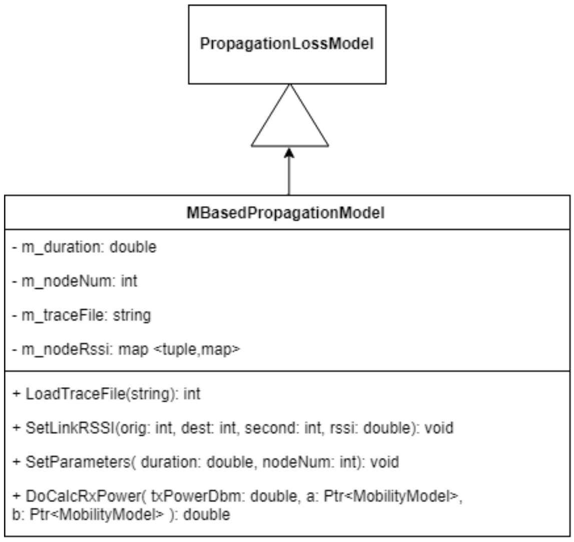


Figure 4.9: MBasedPropagationModel class diagram

This model keeps useful information stored in order to quickly provide to the simulator the needed RSSI values. *m_duration* and *m_nodeNum* parameters are used to validate the usage of the model, that is, if the requested RSSI is within the range of available values and if the node id is also within the nodes that have samples. *m_traceFile* and *m_nodeRssi* store the name of the trace file to load and the RSSI values loaded, respectively. Once the model is created on simulation environment, it will proceed to *LoadTraceFile* and set all the parameters. During simulation, whenever a RSSI value is requested, *DoCalcRxPower* is called.

In order to adapt this model to best fit the Signal fluctuations of a given scenario, it is necessary to introduce a Nakagami-m Fading model, estimating its m-parameter by fitting its distribution to the RSSI distribution of the scenario. It is known that acceptable values for the m parameter satisfy the condition of $m > 0.5$. The higher the m-parameter is, the tighter the shape of the probability density function will be, meaning there can be less variation of the signal.

NakagamiPropagationLossModel supports attaching a different m-parameter, that is, a different fading pattern to different distances separated by bins. There was a modification to allow to introduce more than 3 bins, which was the default from the available code. This way, it is possible to create variable fading behavior with distance instead of using the same parameter for all sets of distances. Considering this, a different parameter for each distance bin is estimated, as it is explained later on section 4.8.2.

4.4.2 Error Model

After knowing the resulting signal strength value due to the propagation channel, the next step is to account for errors. Error modelling is an important part, because it is the module that decides whether a packet is received successfully or with errors, based on calculations that directly involve the SNR for each chunk of the packet. This models the OFDM error by calculating a PER, depending on the used modulation, given the incoming SNR and the Data Rate being used.

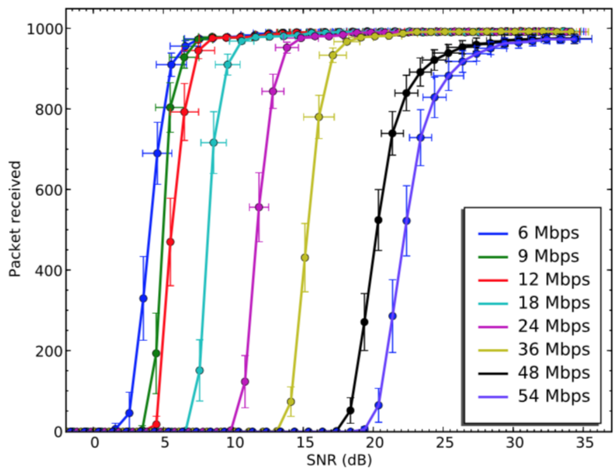


Figure 4.10: Frame Success Ratio for 1000 frames sent with different data rates

There is a validation of the error model *NistErrorRateModel* with real measurements on [54] where Figure 4.10 was obtained from. In this test 1000 frames were sent and the frame success ratio (FSR) was calculated, showing that for a specific SNR range, depending on the data rate and modulation (which is in this case is always OFDM), the probability of receiving a frame decreases quickly, thus increasing PER. A random number is drawn for each packet and with the given PER, a decision is made, that is, a successful reception or a reception with errors.

4.4.3 Interference Helper

This helper is important on NS-3 to account for packets arriving at the same time which can interfere with other packets reception. In order to process them, if the PHY layer is in RX mode, all packets arriving at that time will not be received and will sum into the interference level. It allows also to evaluate if energy rises above CCAmode1threshold, which will tell if the channel is currently Busy.

As shown on figure 4.11, CCAMode1Threshold is the signal detection threshold of other packets or interference in the medium and anything below that threshold is considered noise. EnergyDetectionThreshold represents the detection threshold of the wireless board, that is, the minimum RSSI value to start receiving packets. These 2 thresholds are set to -95dBm and -99dBm, which are the adequate values for the physical wifi radio used on the experiments and on current Veniam's system.

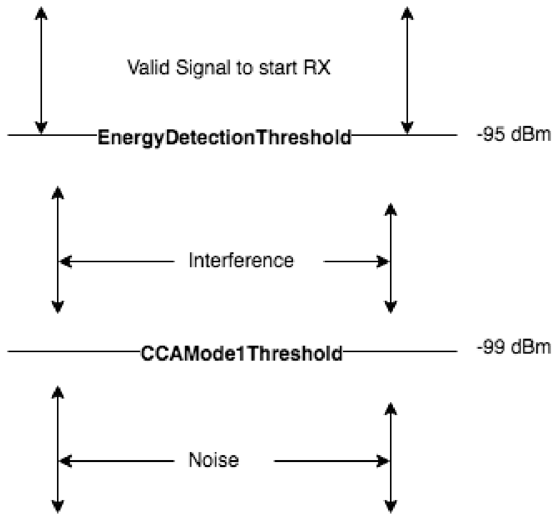


Figure 4.11: Interference mechanism

This helper calculates each SNR value by defining the noise level as shown on equation 4.1.

$$Noise = RxNoiseFigure * Nt + NoiseInterference \tag{4.1}$$

Where *RxNoiseFigure* is noise due to non-idealities, *Nt* is thermal noise and *NoiseInterference* is the resulting noise from packets that are being sent at the same time. This way, in scenarios where there are lots of nodes communicating, the Noise value can increase above those defined thresholds, resulting in the next communications being affected by higher interference.

4.5 WAVE Module

Regarding the WAVE standards implementation, NS-3 currently offers 2 options for using in Vehicular Networks environment:

802.11p MAC/Phy Layer - Simple implementation with only Single-Channel operation, adaptation of 802.11a, which can simulate with 10Mhz of Bandwidth in OFDM at the specific 802.11p Data Rates.

Wave Compliant MAC Layer - Supports Multi-Channel operation and many other features from IEEE 1609.x protocols, such as BSM and WSA, already including the use of 802.11p MAC/-PHY.

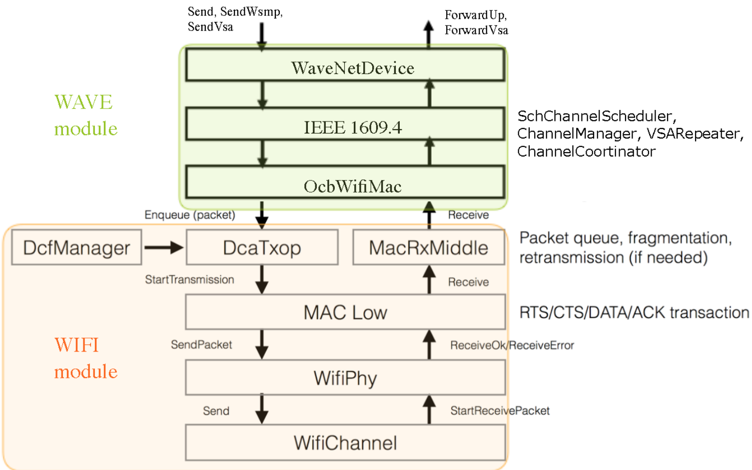


Figure 4.12: WAVE Module architecture

This module adds new feature on top of the existing Wifi Module, as shown in Figure 4.12[55]. The concept of OCB is introduced with *OcbWifiMac*, which allows stations to directly connect to each other when in range, without the typical 802.11 authentication process. On top of that, IEEE 1609.4 stack is implemented, including *ChannelSchedule*r, *ChannelManager* and *ChannelCoordinator*, which handle the operations of alternating channels (between CCH and SCH). At the highest layer of the module is the *WaveNetDevice* which is the device type that is associated to a WAVE node, providing methods for sending IP packets, WSMP packets and WSA frames.

In order to make the implementation as similar as possible to the system running on the OBUs, this module is chosen and configured to send periodic WSA frames. Regarding the channel, no channel switching happens, maintaining a continuous flow on a single channel.

Layers below the WAVE module are already handled by WIFI module, which takes care of forwarding sent and received packets to and from the channel, respectively. *DcfManager*, *DcaTxop* take care of controlling when can data be transmitted by checking the medium and assigning a channel for communication, queuing, fragmenting and retransmitting packets when needed. *MacRxMiddle* handles reception at the same layer. *MAC Low* handles lower-level operations such as Request-to-send/Clear-to-send (RTS/CTS). *WifiPhy* represents the chosen physical layer model and *WifiChannel* represents the propagation channel attached to the physical layer, that contains the propagation models, such as TwoRay, Nakagami and others.

4.6 Transport Layer and Applications

In terms of the application level, several applications were tested, giving estimates of different aspects of the network performance and behavior. Available benchmarks were made using TCP protocol as the Layer 4 protocol, which is considered the worst case scenario due to the headers overhead, connection setup times and re-transmission mechanism. For that reason, all the applications described below are configured to use TCP as well.

BulkSendApplication - This application is similar to FTP content distribution, where data is sent at the maximum possible rate, similar to iPerf, depending on the sending rate configured. It can only be run in TCP.

OnOffApplication - This application allows to send data with a constant Packet Size, which can be defined to be near or equal to the configured MSS. If the value is equal or higher, more retransmissions and packet fragmentation will occur. It sends data using an isochronous timing, that is, sending events occur regularly at equal time intervals, as it is configured on the simulation. It can be used to send Data with a *On-Off* pattern. However, in this case we want the application to be always *On*, continuously sending data.

PingApplication - This application aims to replace the sending of an ICMP packet and printing the RTT of the connection, during simulation time. It can be useful to validate Delay metrics that are being obtained, as this is the correspondent Delay at the application level.

To generate TCP flows with a similar behavior as an iPerf application, **BulkSendApplication** was chosen, being configured with a data payload size of 1460bytes, as defined in iPerf documentation [56]. However, some changes had to be made to the configuration of the *TcpSocket* associated with this application. MSS (Maximum Segment Size) was set to 1460bytes and number of retries and connection attempts were increased to a higher value, in order to avoid problems of TCP not being able to connect to the destination node and transfer data.

4.7 Collecting Statistics

In order to obtain statistics in NS-3, many approaches can be taken, giving results for different parts of the network. The main focus here is to have results being collected with the same granularity and at the same network level as the real-world experiments that were made. For that purpose, 3 different ways of obtaining results are explored in this work, as shown in Figure 4.13.

One useful and customizable way obtaining results is using the *Callbacks* system and the *Simulator::Schedule*, in order to call a specific method, for example, when a packet reception or packet loss event happens. Using the schedule method, it is possible to make a specific piece of code run multiple times in parallel with the simulation steps, collecting, for instance, throughput

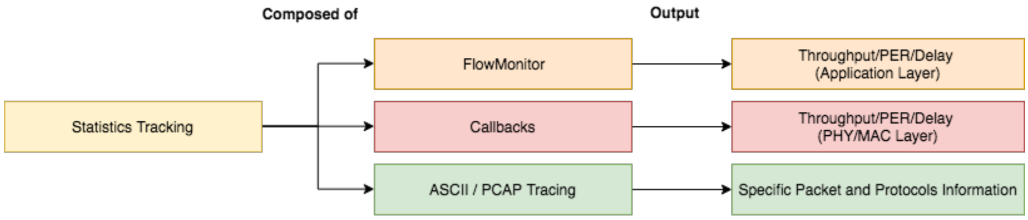


Figure 4.13: Statistics tracking components

per second. These statistics are collected at Physical/MAC Layer of the network stack and can be collected per link or per node. Events being traced during simulation run-time are: packets dropped at Phy/MAC Layer when transmitting or receiving, packets sent and received at Phy/-MAC Layer, packets received successfully or with error and how much time the physical device spends on each state (receiving, sending, idle, ..) to measure medium congestion.

There is also a framework used to collect statistics on NS-3 that was considered, named Flow Monitor [57]. This framework consists of multiple probes placed at different levels of simulation to follow the packets path. Flow Monitor considers that recording data at packet-level will not scale as well as histogram bin data, which is why the framework will collect some histograms for delay and jitter, in order to avoid too much information. The concept of flow here is considering the packets that have the same "five-tupple" which is given by (source-ip,dest-ip,protocol,source-port,dest-port). Statistics for each detected flow are shown at the end of simulation. Althought this framework collects statistics, they work upon IP Layer (IPv4 and IPv6) supporting L4 protocols (TCP and UDP), accounting for retransmitted packets in case of TCP and, as a main limitation, it cannot collect statistics from broadcast traffic. This tool is good for retrieving traffic information from UDP or TCP flows that are generated between 2 vehicles, for example, a BulkSendApplication which represents running an iPerf client and iPerf server on 2 separate vehicles, measuring the throughput at the Transport Layer.

The most high-level way of collecting results is to enable Ascii or PCAP traces on NS-3. It is possible to do so on the *YansWifiPhy* object, which allows to trace each packet sent from one node to others. This option is also enabled on this simulation code, however, it was only used for debugging purposes while configuring network parameters.

4.8 Proposed Solution Overview

4.8.1 Simulation Model Architecture

After obtaining a trace file from the database or from benchmark logs, it is necessary to process that data, in order to generate files with a specific mobility trace file format and translate it into a custom mobility model that can be imported into the simulator. Another important information to take as input is the RSSI information and position of neighbour nodes, in order to create a fading profile and estimate the needed parameters to configure signal propagation loss on the simulation environment, more specifically, Nakagami-m parameters. Input and output simulator components that have been changed are highlighted in red. This information will recreate the physical layer behavior into the simulator taking advantage of the existing models and the newly created model that loads RSSI values from the traces. The remaining layers (Wifi Module, IEEE 802.11p, DSRC/WAVE module, Application) are available on the simulator, with the latest standards. There will then be a simulation where the communications of nodes will be replicated, while measuring them over time. At the end, the output shows some relevant information like Throughput, Packet Loss Ratio, Delay, for each node, which will be saved into two separate files: one that will go to the parsing program to calculate and estimate error corrections, plotting graphics for visualization of the chosen metrics; another file that is used by a tool such as NetAnim [51], to visualize vehicle's mobility and packet exchanges. The whole system overview is seen on Figure 4.14.

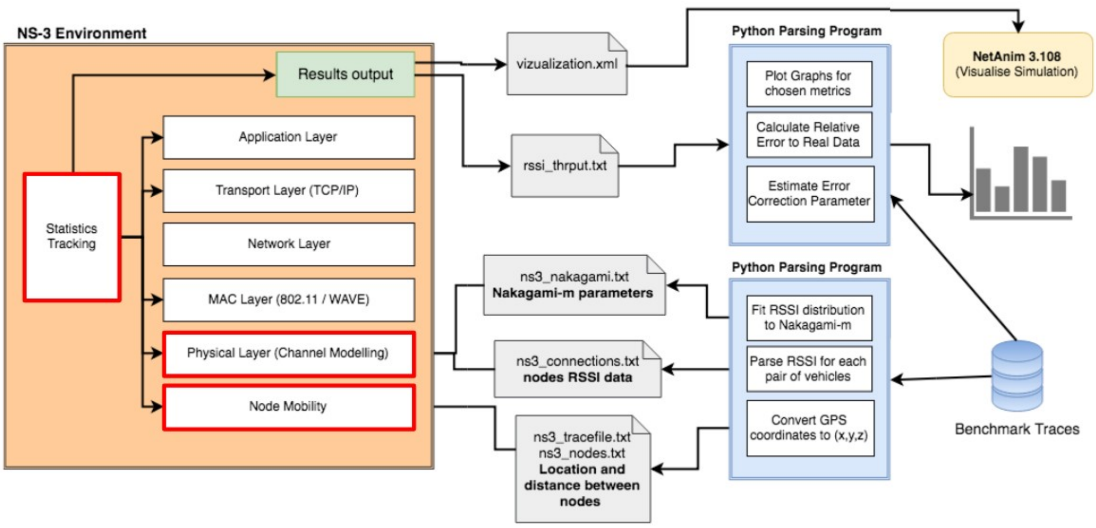


Figure 4.14: Simulation Procedure for Validation

4.8.2 Parameter Estimation and Statistical Tests

As mentioned before, some parameters are extracted from the data, which will be used in the propagation models. When retrieving the parameters for Nakagami-m distribution, which will model signal fading, it is necessary to use an estimation procedure. Two types of estimators are used: **Moment-Based estimators** and **Maximum Likelihood estimators**. After searching the literature, there were a lot of articles talking about the same estimators, all with a similar performance. A total of six different estimators were retrieved from [58, 59, 60, 61]

The approach is to fit a Nakagami-m probability density function to the RSSI samples distribution, either for all data or for data separated within distance bins. Separation with distance bins allows different fading profile for a specific distance within simulation, being studied in [26]. Each estimator produces a m parameter. The resulting parameter is the mean of all estimations, m_{mean}. N denotes the number of samples and x_n is the sample of index n.

$$GM = (\prod_{n=1}^{N} x_n^2)^{\frac{1}{N}} \tag{4.2}$$

$$u_k = \frac{1}{N}\sum_{n=1}^{N} x_n^k \tag{4.3}$$

$$y = \frac{u_2}{GM} \tag{4.4}$$

$$m_{est1} = \frac{u_2^2}{u_4 - u_2^2} \tag{4.5}$$

$$m_{est2} = \frac{8.898919 + 9.059950 * y + 0.9775737 * y^2}{y * (17.79728 + 11.968477 * y + y^2)} \tag{4.6}$$

$$m_{est3} = \frac{0.5}{y} \tag{4.7}$$

$$m_{est4} = \frac{-0.2081}{\frac{u_1^2}{u_2} - 0.99709} \tag{4.8}$$

$$m_{est5} = \frac{0.58799}{y - 5.93 \times 10^{-3}} \tag{4.9}$$

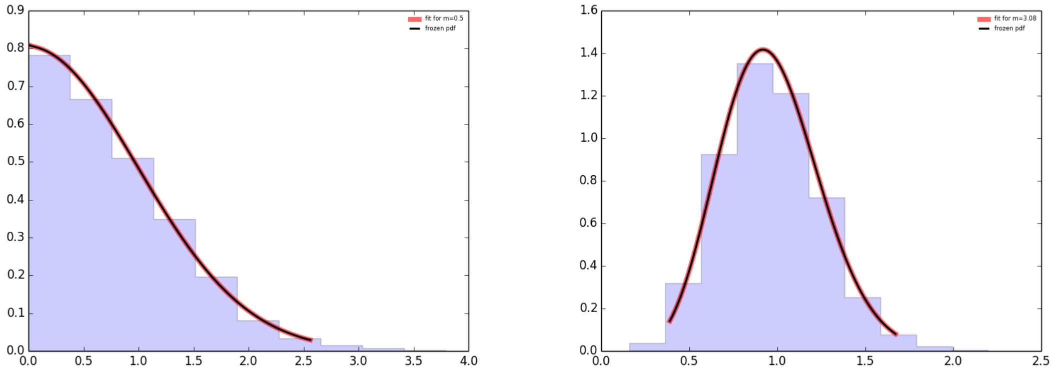
$$m_{est6} = \frac{u_1 u_2}{2u_3 - u_1 u_2} \tag{4.10}$$

$$m_{mean} = \frac{m_{est1} + m_{est2} + m_{est3} + m_{est4} + m_{est5} + m_{est6}}{6} \tag{4.11}$$

Presented equations on 4.2, 4.3 and 4.4 correspond to the Geometrical Mean, k_{th} moment and likelihood function. Equations 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 show the used estimators and how the mean of the estimated values is calculated.

In order to verify the correctness of estimations, some random Nakagami-m populations with 1000 samples were generated, where m is known. This generated data was then passed to the estimators to check if estimations matched the expected m value. Function fit with respect to the data histogram is shown on Figure 4.15.

As seen on Table 4.2, the values obtained from each of the estimators are close to the m parameter that corresponds to the population, resulting in a good estimation mean, where the best estimations are highlighted in bold. Estimators m_{est1}, m_{est6} are purely moment based estimators which have identical performance and work better with respect to the big amount of data. m_{est3} shows the lowest accuracy for low m-values due to the lower order approximation that it assumes and m_{est2} is a maximum likelihood estimator derived by polynomial fitting of the log-likelihood



(a) Fit for m=0.5 population

(b) Fit for m=3 population

Figure 4.15: Nakagami-m estimators fit test

function [59]. m_{est4} and m_{est5} are moment-based and maximum likelihood estimators, respectively, with constant values obtained from fitting and approximations, sometimes resulting in slight lower performance. Overall, all estimators present values closer to the real m-value and m_{est1}, m_{est2} and m_{est6} have the most consistent results.

Population	m_{est1}	m_{est2}	m_{est3}	m_{est4}	m_{est5}	m_{est6}	m_{mean}
m = 0.5	0.514	**0.500**	0.394	0.584	0.465	0.512	0.495
m = 1	1.006	1.015	0.881	**0.999**	1.047	1.014	0.994
m = 3	**3.044**	3.055	2.899	2.759	3.530	3.045	3.055

Table 4.2: Nakagami-m estimations validation

In order to assess how well the Nakagami-m estimated distribution fits the RSSI data when estimating parameters for the experiments, Kolmogorov-Smirnov test is used, as used in [26] for the same purpose, which will compare 2 samples: samples from the experiment $F_{exp}(x)$ and samples from a generated Nakagami distribution with the estimated m-parameter $F_{est}(x)$. This test does not depend on the sample size. This result in a D-statistic value that indicates how close the 2 distributions are, obtained with equation 4.12.

$$D = max_x \|F_{exp}(x) - F_{est}(x)\| \tag{4.12}$$

This test relies on the biggest value of the difference of the measured data and generated data from estimation. The lower this D-statistic value is, the better is the similarity of both distributions.

4.8.3 Error Correction and Validation

Results obtained from the simulation procedure come with an error associated to the real values. The approach used to correct this error is to first obtain the values of deviation and relative error between real and simulated Throughput, by using equations 4.13 and 4.14, respectively.

$$\Delta x = Throughput_{Real} - Throughput_{Simulation} \tag{4.13}$$

$$\varepsilon_r = \frac{|Throughput_{Real} - Throughput_{Simulation}|}{Throughput_{Real}} \tag{4.14}$$

The resulting values from these equations can be plotted and will result into graphs that aim to show how the offset can be characterized. The approach taken to correct this error is to divide it in different zones, divided by RSSI and/or distance thresholds. After the division, a linear regression is done to fit the equation $y = mx + b$ where y represents the real throughput values and x represent the simulated throughput values. The factor m and the constant b are estimated to the best fit for the data points. .

Another way to validate the results obtained, is to make histogram, scatter and box plots where both real values and simulated values are placed on the same graph, allowing to make a direct comparison of both results. This will allow to see if the behaviour of the chosen metrics is equal and if the system can provide a good estimate of Throughput, given the available data to explore in this work.

Once the correction is made, corrected Throughput graphs are presented along with the error information shown in tables. More detailed results about this approach are provided on section 5.4

Chapter 5

Simulation Results

This chapter provides an analysis of the chosen performance metrics with respect to different approaches taken to model the propagation channel. In order to do so, some metrics are presented to compare between the different models with different Nakagami-m fading parameter combinations. The results obtained from the models are then shown for each chosen metric and for each scenario location. After that, in a global view, these results will be compared across Aveiro and Leça da Palmeira locations to understand how different those scenarios are and how it impacts the performance. To perform validation of the simulation setup and show that the proposed methodology has some gain when compared to a regular simulation system, we will be calculating the error of the simulation results vs real results and defining a correction procedure for the obtained errors.

5.1 Comparison Procedure

In order to compare experiment results across the different models and reach conclusions about benefits of using each approach, we will be comparing 2 different propagation models: *TwoRayGroundPropagationModel* and *MBasedPropagationModel*. On top of these models, it is possible to add a Nakagami-m distribution that accounts for the fading effect. For the fading effect there will be 3 combinations: *NoFad* which is the model without any fading added on top; Fad_{global} which is the Nakagami-m with 1 m-parameter fitted for the whole RSSI distribution; Fad_{bin100} corresponding to Nakagami-m with 1 parameter fitted for each distance bin of 100 meters.

Parameter	TwoRayGroundModel	MBasedModel	Nakagami-m
TxPower	14.58dBm/23dBm	-	-
TxGain	2dBi	-	-
RxGain	2dBi	-	-
Frequency	5.9Ghz	5.9Ghz	-
Antenna Number	2	2	-
Propagation Loss Type	Deterministic	Tracebased	Stochastic estimated from RSSI traces
RxNoiseFigure	4 dB	4 dB	-
Bin Size	-	-	None(Fad_{global}) / 100m

Table 5.1: Summary of model configurations and properties

Table above shows configurations needed for the different models to be tested. MBased model does not need to have a transmitter/receiver antenna gain or TxPower attribute, since the values come from the measurements. Nakagami-m is just applied on top of the values coming from either MBasedModel or TwoRayGroundModel, thus not requiring any other parameters, except for the values of the m-parameter for each of the bins, when this option is enabled.

To provide an analysis across the different combinations of models, the following metrics will be used as explained:

- **RSSI** - The evaluation received signal power will be made using a histogram of occurrences along with a scatter plot of its variation with respect to distance. A distinct histogram for $RSSI_{differences}$ showing the differences of consecutive RSSI samples is also presented and the equation for this metric is shown on 3.2.

  This graphs allow to visualize how the signal is being modeled in simulation with respect to the different models in evaluation. It is used as a "sanity check" since we want to assure that the RSSI distribution is as close to reality as possible.

- **Throughput** - This metric will be evaluated with respect to signal strength and distance to the other node. It represents the the amount of data sent between the nodes and it is measured each second. This metric is obtained from data following the formula on equation 3.4.

  Throughput graphs are useful to understand the general network behavior being one of the main performance indicators.

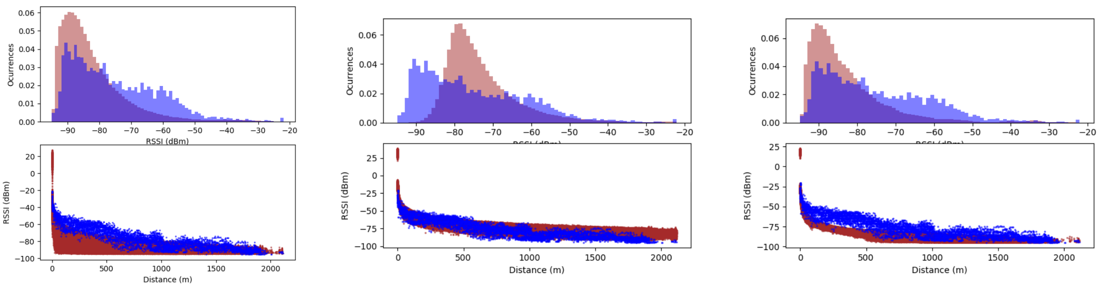
Both of the metrics described above will contain real values and simulated values overlapped on the same plot, in order to allow for a direct comparison. Histogram plots are usually normalized, that is, the sum of all bars is 1 and bin size is 1 unit (for RSSI, 1dBm). This helps to compare distributions by the frequency of their samples instead of the number of occurrences.

After the plots are made and analyzed, we proceed to evaluate the error between simulation and real measurements to fit it to a correction function. The procedure for this step is described on 4.8.3.

5.2 Obtained Results

This section presents the results obtained with respect to metrics shown on section 5.1 for each of the scenarios presented: **Leça da Palmeira** and **Aveiro**. To validate the model, inputs from both of these experiments (RSSI and vehicle positions) are loaded into the simulator that will generate outputs. Those outputs are then compared to the measured values from the dataset, which in this case is throughput. Plots shown in this section are made with all results from all trials merged into one set of data, for each scenario, in order to provide a general overview of the results per scenario location, as shown in figure 5.1. As mentioned before on section 3.4, RSSI data actually contains a granularity of 1 dBm and GPS data contains a granularity of 10 meters on longitude and 1 meter on latitude, which is visible during the plots of this section.

Figure 5.1: Procedure for obtaining results per scenario



(a) RSSI for TwoRay using Fad_{global}

(b) TwoRay model with 23dBm TxPower

(c) TwoRay model with 14.58 dBm TxPower

Figure 5.2: Considerations for parameter exclusions

Before starting to analyze results, it is necessary to talk about why Fad_{global} results will not be analysed. When looking to the RSSI histogram at figure 5.2a, it is shown that RSSI with respect to distance does quickly decrease with a high variation rate, since the parameter is estimated for the whole real RSSI distribution, which has a wide range of values to be fit. This high variations on RSSI have a great impact on throughput throughout the different models and experiments. For that reason, it was decided to only analyze the best case which is Fad_{bin100} with comparison to the worst case $NoFad$.

There is also one issue with TwoRayGround model, where RSSI values are always calculated with respect to distance, so it is likely that RSSI won't match with the measurements. Although, to make a fair comparison, TxPower for this model has been configured to 14.58 dBm which is the measured TxPower, according to [28], for the reason that cable losses and hardware of OBU have limitations and the effective power transmitted is less than 23 dBm. The RSSI output is shown on Figure 5.2c, showing a higher approximation on the histogram behavior. However, with respect to distance there a higher deviation on the first meters. Experiment was made without changing this parameter but the results were more optimistic, as seen on Figure 5.2b, where values don't fall with distance as quick as in the real measurements and most RSSI values lie on a good signal zone. This caused throughput to keep constant on zones where it should start dropping. For that reason, results will be shown considering a 14.58 dBm TxPower.

5.2.1 Scenario 1 - Leça da Palmeira

This scenario was previously described in section 3.5.1 and 3.5.2, containing data up to 300 meters of distance. A total of 5 different trials results were merged both for simulation and real measurements to allow a trial independent comparison. There are 3 trials with stopped vehicles and 2 trials with one vehicle moving away and returning several times.

5.2.1.1 No Fading Results

This sub-section shows the results of both models when no fading is added on top. It is expectable that no considerable RSSI variations occur, specially in this scenario where distances are still within the DSRC range.

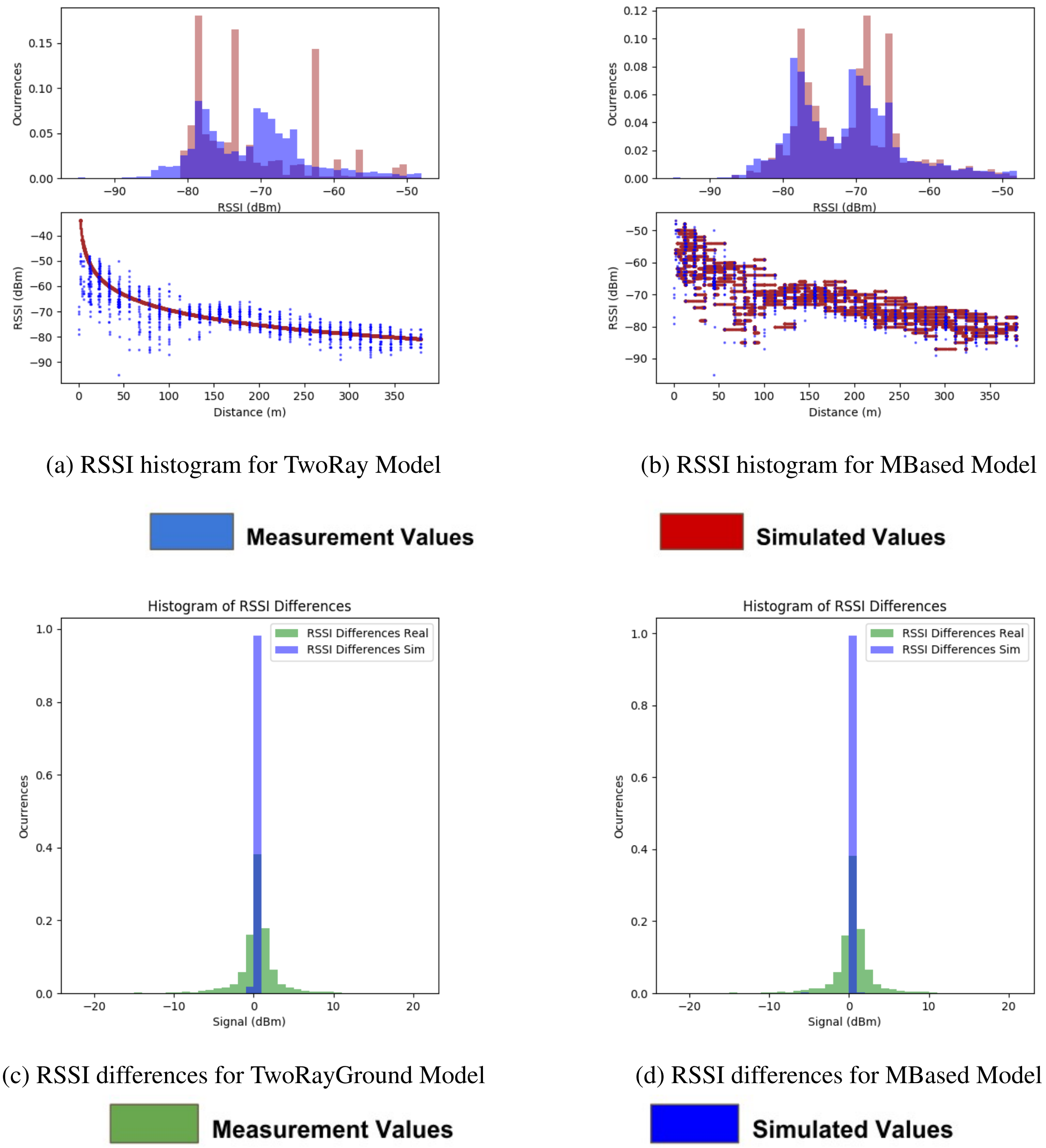
In terms of RSSI, it is visible on figure 5.3a that TwoRay RSSI values with respect to distance are decreasing with a pattern similar to a Log Distance path loss, as expected, and seem to fit well for the average of real RSSI values. In terms of the histogram values, there are 3 higher bars that represent the scenarios where vehicles are stopped, that result in the same RSSI since other signal variation effects are considered and for that reason TwoRay's RSSI distribution (red color) is not as spread as the real one (blue color). For the case of MBased model, on figure 5.3b, it can be seen that with respect to distance, RSSI values have oscillations but with periods where they keep constant due to the 1 second granularity of signal strength trace values. With respect to the histogram, the percentage of occurrences has a similar value, however, since the simulation granularity of RSSI is higher (1 value per packet), the bars don't have an exact match.

With respect to Figures 5.3c and 5.3d, both exhibit the same behaviour, which shows that values resulting from simulation output tend to have 0 variation with an enormous frequency due to the same RSSI value being used for several packets in a row (once again due to RSSI granularity).

Regarding the throughput plots shown on Figure 5.4b and 5.4a , a slight offset between simulation values and real values is still visible and it keeps stable due to the good signal conditions and low distances. In the case of MBased model there are some values close to 2Mbit/s which possibly result on the fact that TCP reduces its contention window to about a half. On Figures 5.4c and 5.4d it is visible that throughput values from TwoRay are more dense towards a lower signal strength value, still being more optimistic that MBased model, that has its throughput with the same spread as the Real values, although showing less fluctuation.

5.2.1.2 Nakagami-m Fading with 100m bins

For this setup, bin values were estimated from this experiment RSSI distribution with the Nakagami-m fitting procedure, resulting in 4 bins of 100 meters.



(a) RSSI histogram for TwoRay Model

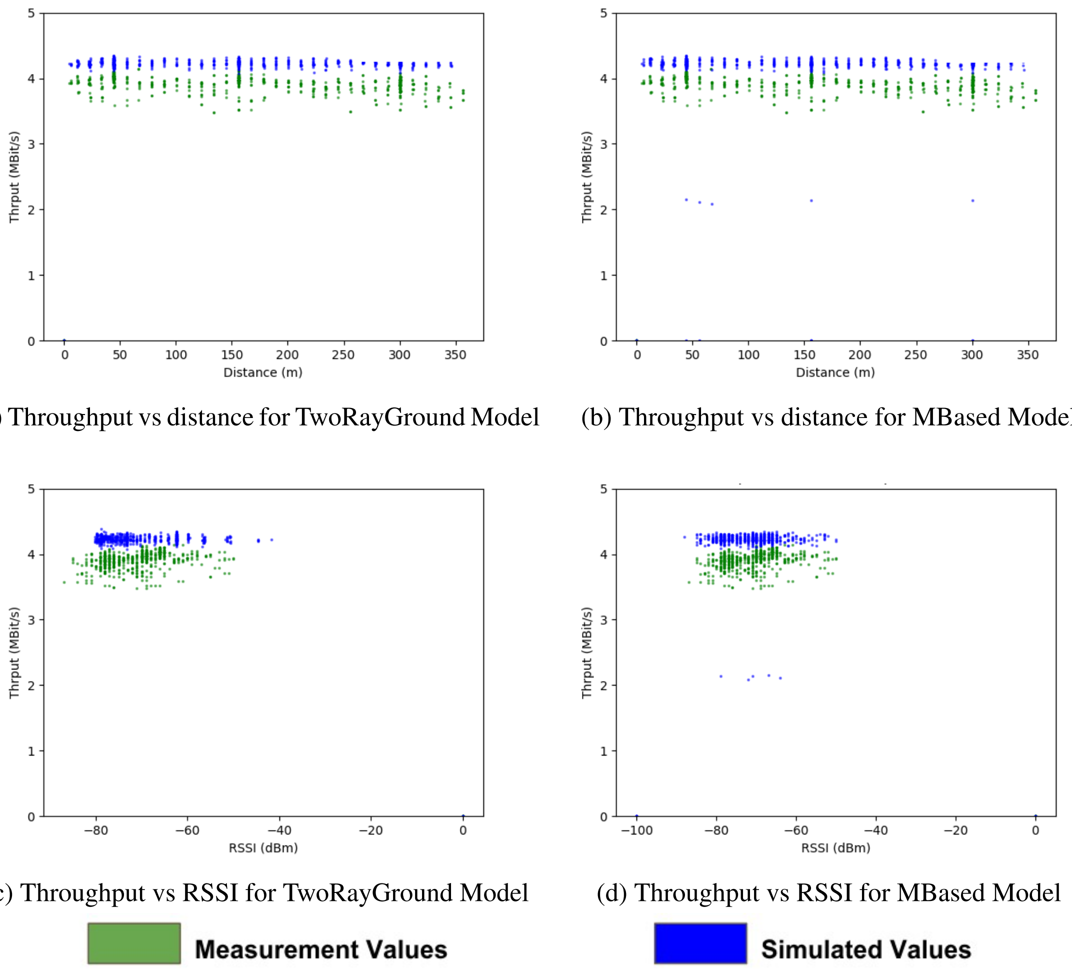
(b) RSSI histogram for MBased Model

(c) RSSI differences for TwoRayGround Model

(d) RSSI differences for MBased Model

Figure 5.3: RSSI analysis (No Fading) for Leça da Palmeira experiment

Considering RSSI samples presented on Figure 5.5a, for the case of TwoRay model, the values show an improvement compared to no fading, in terms of variation within distance, maintaning the same log-distance decrease. In the histogram, it is also visible that TwoRay model has a more similar behaviour compared to the real values, however there is a difference on the peak values, due to other signal effects that the deterministic model cannot reproduce. When analysing figure 5.5b, it is visible that there is a more fine-grained match on the RSSI values with respect to distance, where the simulated values follow the real values behaviour with the fading effect that is missing from the traces. In terms of histogram distribution, it is visible that there is a much better match



(a) Throughput vs distance for TwoRayGround Model

(b) Throughput vs distance for MBased Model

(c) Throughput vs RSSI for TwoRayGround Model

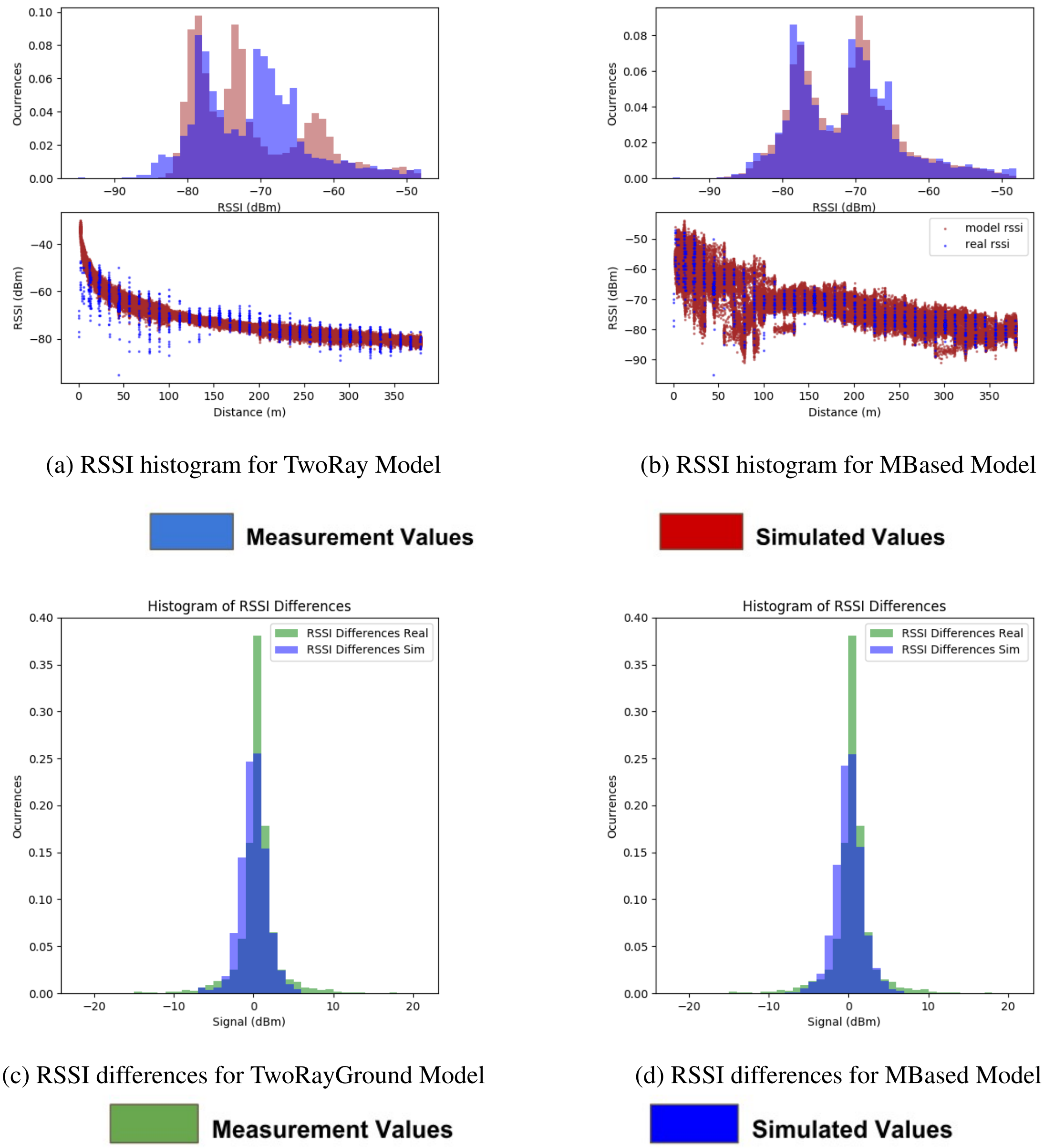
(d) Throughput vs RSSI for MBased Model

Figure 5.4: Throughput analysis (No Fading) for Leça da Palmeira experiment

between simulation values and real measurement values.

Regarding plots on Figures 5.5c and 5.5d, both exhibit a similar behaviour when compared to the real RSSI differences. The noticeable difference is that real values tend to have less variation, with a higher bar on 0 while on simulation the same doesn't happen because of higher fading severity.

Considering throughput plots on figure 5.6a and 5.6b, both exhibit a similar behaviour, with no noticeable differences when compared to no fading, since distance is still within good communication range for this experiment. The plots with respect to RSSI, on figures 5.6c and 5.6d also exhibit a stable values, but this time, with fading on distance bins, throughput values for both Two-Ray and Real-Signal models became more spread due to fading effect that causes more variations on RSSI values.



(a) RSSI histogram for TwoRay Model

(b) RSSI histogram for MBased Model

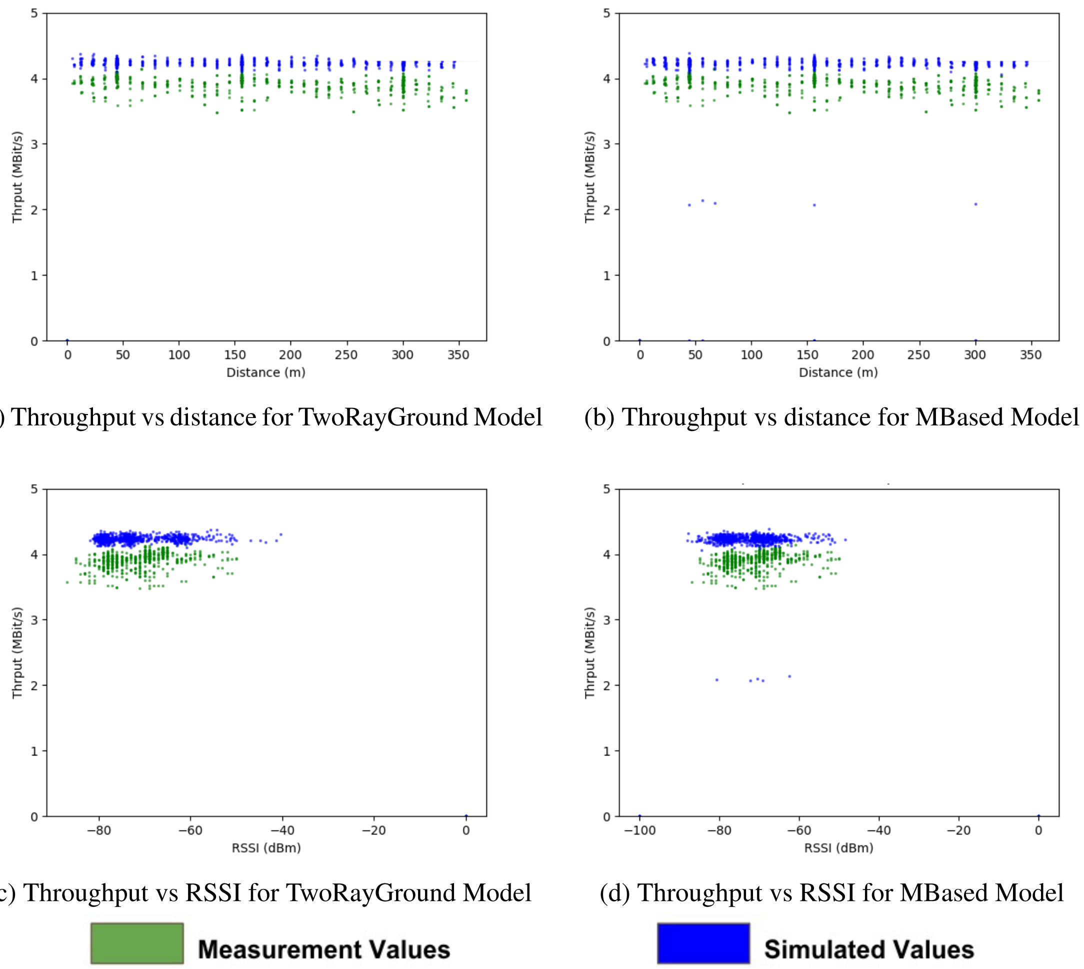
(c) RSSI differences for TwoRayGround Model

(d) RSSI differences for MBased Model

Figure 5.5: RSSI analysis (Fading bin 100m) for Leça da Palmeira experiment

5.2.2 Scenario 2 - Aveiro

This scenario was previously described in section 3.6.1 and 3.6.2 containing data up to 2115 meters of distance. A total of 10 different trials results were merged both for simulation and real measurements to allow a trial independent comparison. There are 8 trials with 1 vehicle moving away and returning and 2 other trials with 1 vehicle moving on laps. Due to higher distances and consequently lower RSSI values, this scenario is expected to provide data about the zone where throughput transitions to 0.



(a) Throughput vs distance for TwoRayGround Model

(b) Throughput vs distance for MBased Model

(c) Throughput vs RSSI for TwoRayGround Model

(d) Throughput vs RSSI for MBased Model

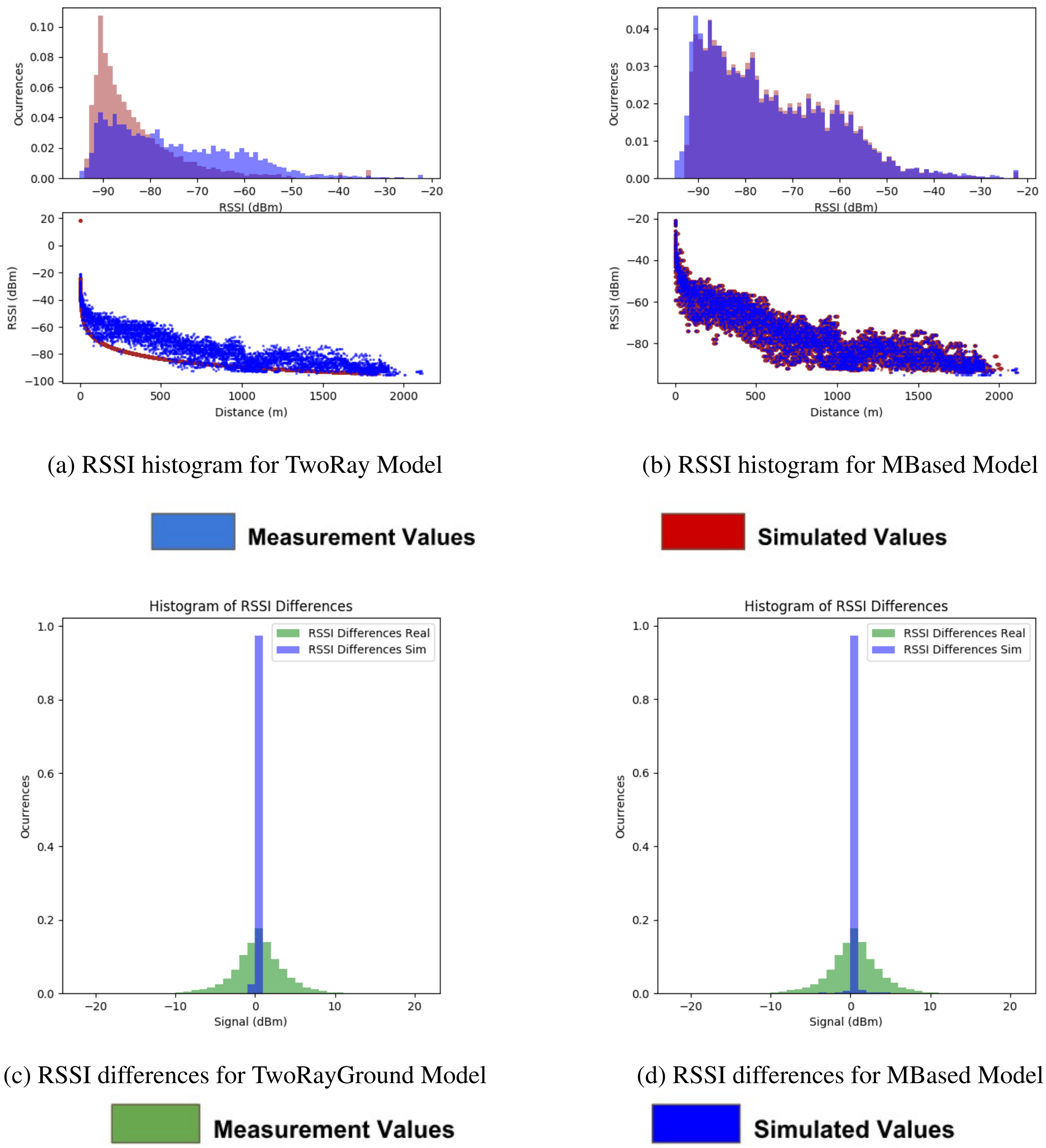
Figure 5.6: Throughput analysis (Fading bin 100m) for Leça da Palmeira experiment

5.2.2.1 No Fading Results

In this case, there won't be any signal variations due to fading. However, due to the higher distances, some impact on the metrics is expected.

In terms of RSSI, figure 5.7a shows that TwoRay has a similar behavior with respect to distance, however with low variations and being less optimistic for lower distances. In terms of the histogram distribution, it is visible that TwoRay distribution is more dense in lower RSSI values while the real one is more spread across other values. Figure 5.7b shows that RSSI values from this model without fading match with high precision the real values, which is expected since there is no fading component added and here more samples are present. Differences on the distribution are due to higher RSSI granularity on simulation which leads to different number of occurrences for each sample and due to applications starting later on simulation than on real experiments,which results in an offset when directly comparing values of the same time instant.

Regarding both graphs on 5.7c and 5.7d, both exhibit a similar behavior with a high bar on RSSI difference of 0, meaning that simulated values are not changing for several packets in a row



(a) RSSI histogram for TwoRay Model

(b) RSSI histogram for MBased Model

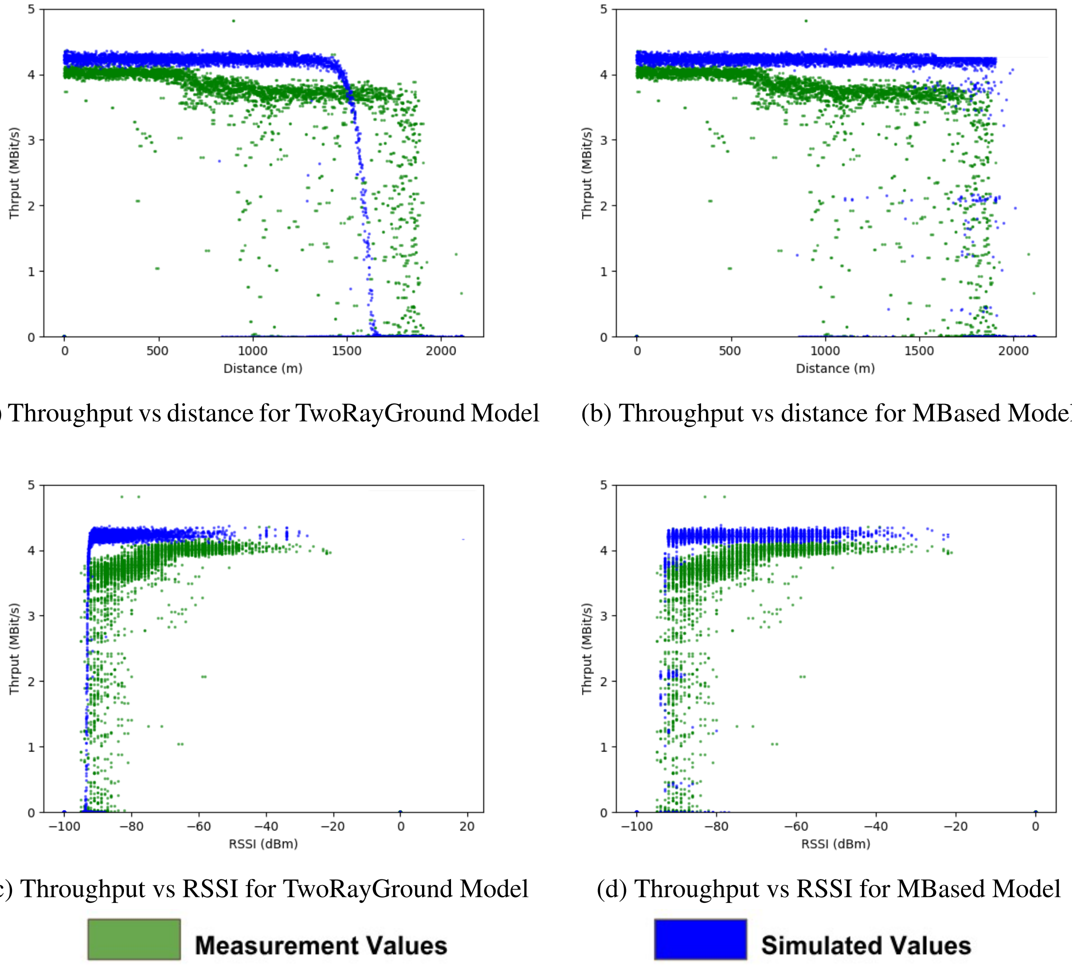
(c) RSSI differences for TwoRayGround Model

(d) RSSI differences for MBased Model

Figure 5.7: RSSI analysis (No Fading) for Aveiro experiment

and only change once per second, since there is no fading component being simulated. However, real values show that signal variations exists and it is more spread from -10dBm to 10dBm and not so dense in 0.

Considering throughput, on figure 5.8a is seen that throughput on TwoRay model has a stable behavior up into about 1500 meters, where it quickly drops to zero, since RSSI values are given for each individual distance. On figure 5.8b it is visible that using real trace-values, throughput keeps with a constant average value until the distance of about 1900m where it falls along the real values and the effect of the throughput drop is not noticeable as it almost goes instantly to



(a) Throughput vs distance for TwoRayGround Model (b) Throughput vs distance for MBased Model

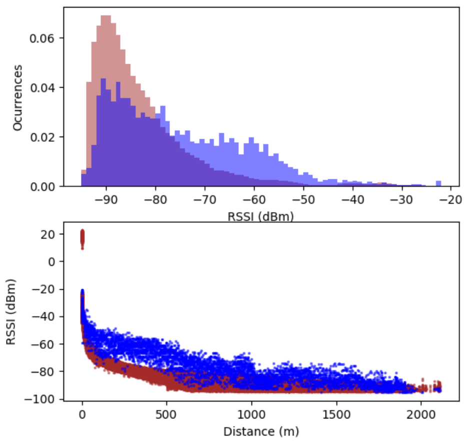
(c) Throughput vs RSSI for TwoRayGround Model (d) Throughput vs RSSI for MBased Model

Figure 5.8: Throughput analysis (No Fading) for Aveiro experiment

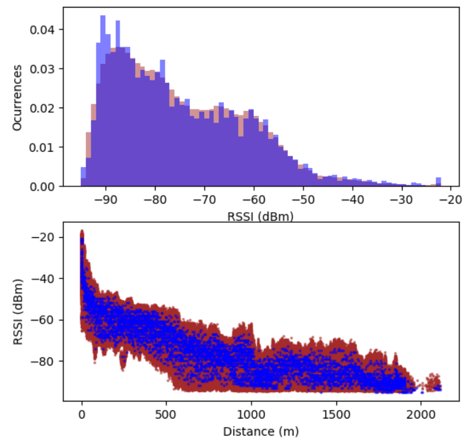
0. Since there are no signal variations, being constant through each second, it has an impact on throughput that does not oscillate as it happens in reality. Since the RSSI value is given from the measurements, communications between the vehicles stop at the same Distances and RSSI values. However, for both models, there is a zone on close to 750 meters of distance where throughput drops and then stabilizes at 1000 meters, with some oscillation starting to appear due to multipath components, one of them being the roundabout at 1100m of distance. Regarding figures 5.8c and 5.8d both exhibit exhibit a similar behaviour. In the case of TwoRay, throughput drop occurs smoothly as for each distance there is a different RSSI value. On the other hand, on MBased model it is noteiceable that RSSI granularity of 1 dBM causes the throughput drop to be almost sudden, with little samples visible on the RSSI < -90dBm area.

5.2.2.2 Nakagami-m Fading with 100m bins

This combination considers a total of 21 distance bins that were obtained by separating RSSI samples within bins of 100 meters of distance. It accounts for several fading processes within the experiment range.

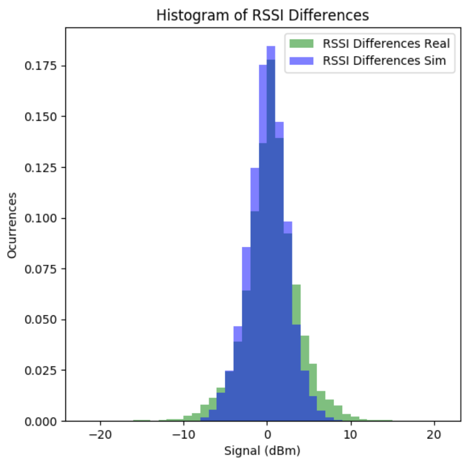


(a) RSSI histogram for TwoRay Model

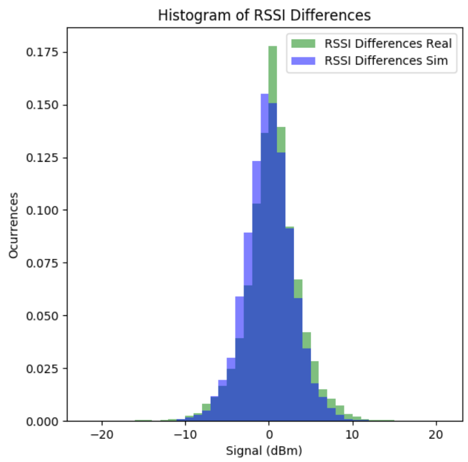


(b) RSSI histogram for MBased Model

Measurement Values Simulated Values



(c) RSSI differences for TwoRayGround Model

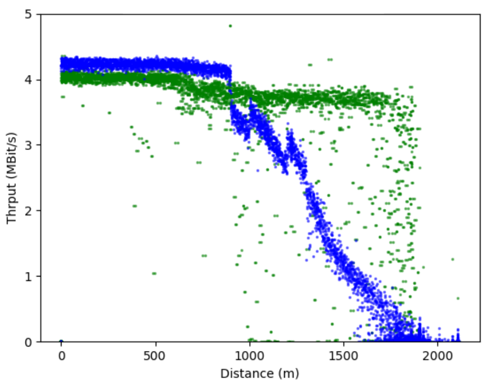


(d) RSSI differences for MBased Model

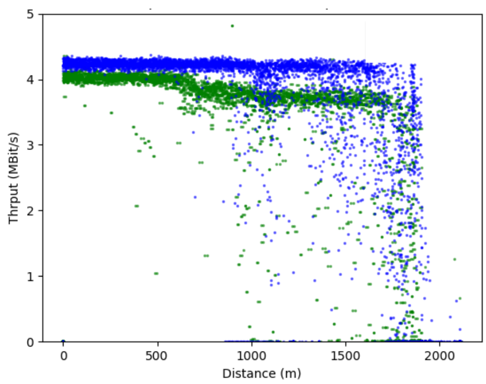
Measurement Values Simulated Values

Figure 5.9: RSSI analysis (Fading bin 100m) for Aveiro experiment

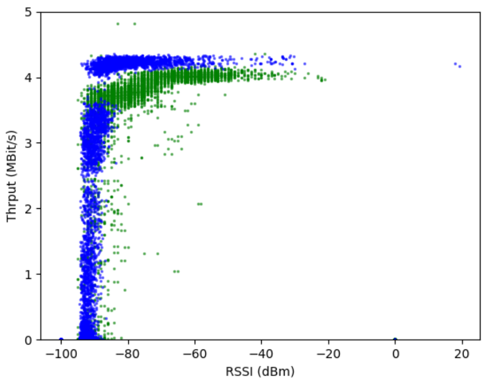
RSSI plots on figure 5.9a show that TwoRay model is now applied with fading variations, including a higher range of RSSI values for each distance, instead of only a single value as in the case of no fading. The histogram distribution is located in the same place with tendency for lower RSSI values but now with a wider range of values as well. However, it shows lower RSSI values on lower distances than what happens in real values due to the configured TxPower that provides the initial offset for the values. On the other hand, on Figure 5.9b is shown that, with respect to distance, simulated values follow the pattern of the real values with the fading variation added on top. In terms of histogram distribution, it maintains a similar pattern to the real values with slightly different occurrences number for each bar that result on the random component of fading. On figures 5.9c and 5.9d , RSSI differences show a similar pattern for both models, indicating that the fading process is creating the needed signal fluctuations from sample to sample. Since all experiments in this scenario are moving experiments, there is no higher peak on 0 value, as it was seen on the previous scenario of Leça da Palmeira.



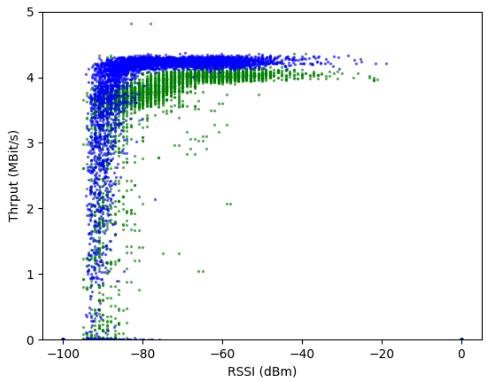
(a) Throughput vs distance for TwoRayGround Model



(b) Throughput vs distance for MBased Model



(c) Throughput vs RSSI for TwoRayGround Model



(d) Throughput vs RSSI for MBased Model

Figure 5.10: Throughput analysis (Fading bin 100m) for Aveiro experiment

Regarding throughput plots, on figure 5.10a throughput values for TwoRay are stable until 1000m where a drop starts to occur. The instability after this zone is due to the different Nakagami fading parameters for each bin of 100 meters, that will have a different impact on the deterministic RSSI that is calculated for each distance. Still, communication does only stop at near 2km, with a high smoother decrease along distance when compared to the throughput drop on real values. In the case of MBased, on figure 5.10b throughput values show some consistency up until 1000m, where from that point fading starts to have a high impact on those values, with higher impact as distances close to 1900m are reached and dropping to 0 with more oscillation than in the case with no fading. Considering the remaining plots, on figure 5.10c is shown that throughput only starts to drop at lower RSSI values than real ones, of near -90 dBm and there is a a gap on the simulated throughput values which can be explained by a sudden fading component change, on the transition to the distance bin of 1000m. For the same conditions, MBased model shown in 5.10d shows a smooth throughput curve in terms of RSSI with lower density of values as throughput reaches close to 0. However, for both models, when compared to real measurement values, on the real scenario throughput starts dropping sooner on about -75 dBm where for both models it only starts dropping on about -85 dBm.

5.3 Results Discussion

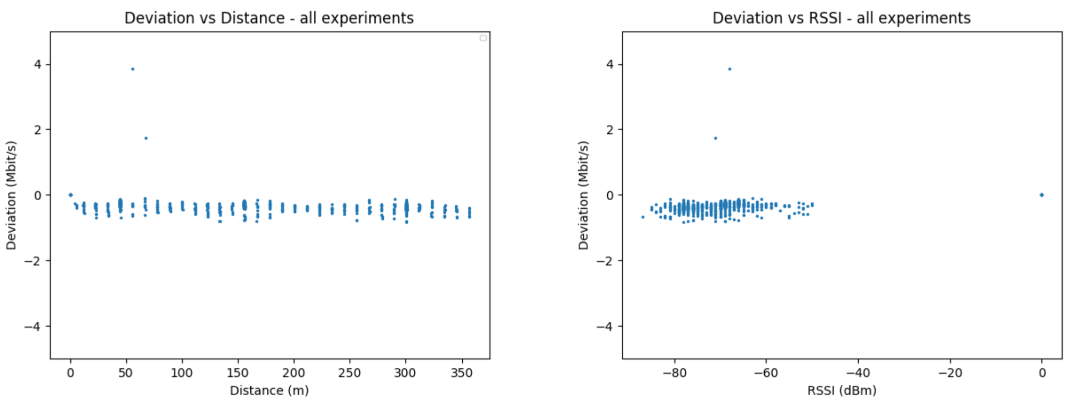
As seen in the previous section 5.2, combinations of models without using a fading distribution on top result in a poor signal modeling into the simulation environment. The signal differences histogram shows how much the RSSI values can vary from one sample to the next one and it can be used to observe the impact of fading. In real world, the signal is always varying due to several effects and distortions, as presented in the state-of-the-art (section 2.2). The impact of the absence of this effect is noticed on the throughput values, when analyzed in terms of distance and signal strength. It is mainly noticeable on the zones where throughput starts decreasing, for RSSI values of under -75dBm and for distance values higher than 750m, which is close to the theoretical range for DSRC communication. A more severe throughput fall occurs near a distance of 1750 meters and RSSI values starting on about -87/-88 dBm. When adding the fading component on top of both models, it can be seen that throughput will more likely have oscillations as soon as the signal becomes weak. Since these experiments used TCP protocol, there are many variations in throughput values due to the fact that the protocol adjusts is congestion window when packets start failing to be transmitted. Overall, TwoRay model shows optimistic throughput estimates since the signal values are deterministic for a given distance. MBased model has the advantage of using real traces to influence the throughput behavior on the simulation results. However, throughput variations on the dropping zone are not easy to model with low error.

5.4 Error Calculation and Corrections

The different models compared in section 5.2 show different error patterns in terms of throughput. The objective of this section will be to obtain a characteristic of that error by essentially calculating the **Deviation** or **Absolute Error** and **Relative Error** as defined in section 4.8.3 and correct it using a regression mechanism. This mechanism is trained using Aveiro's scenario data to make a Linear Regression and used to correct Leça da Palmeira's results, since both experiments, despite their minimal differences on the environment, are made with very similar condition and mostly LOS throughout the different trials.

Considering the approach that is implemented in this work (MBasedModel + Nakagami-m Fading with bins of 100m), we proceed to analyze the error patterns and identify the possibility of correcting results by modeling the error within specific thresholds.

The first analysis was made on Leça experiment. Considering all experiments results from this location merged into one, the deviation of throughput with respect to distance and RSSI was plotted. As it can be seen on graphs present on Figure 5.11, in this case the deviation is constant until aproximately 380meters, which is the maximum range for this test. The same happens for RSSI which keeps a constant deviation until reaching a value of approximately -85dBm, which is the worst value recorded in this test. However, in order to further characterize these results, more data with higher distances and lower RSSI values is needed.

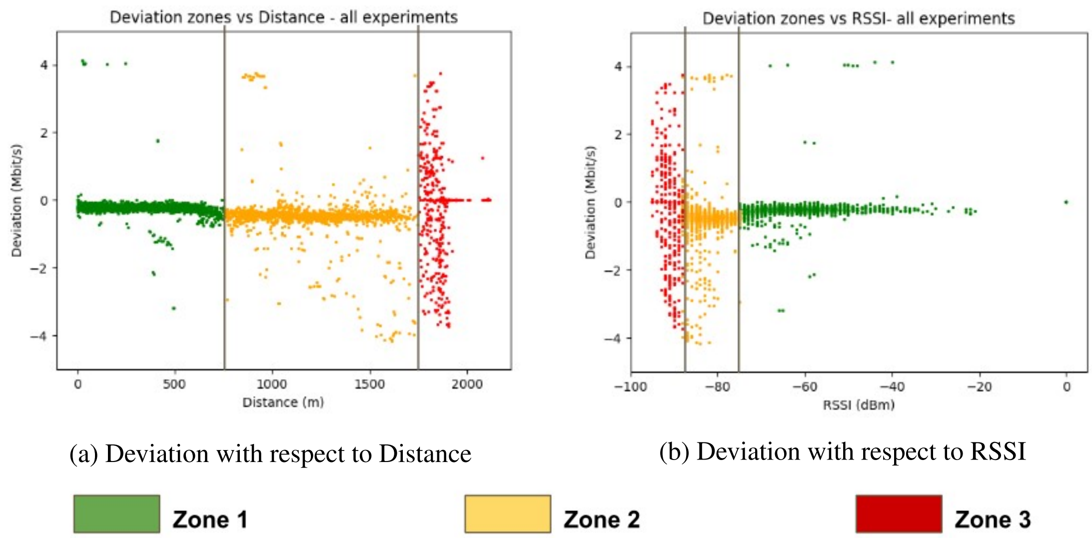


(a) Deviation with respect to Distance

(b) Deviation with respect to RSSI

Figure 5.11: Deviation of simulation results to real measurements for Leça Experiment

In the deviation plots of Aveiro's experiment shown in figure 5.12, there are more data samples for lower RSSI values and higher distances, which allow to characterize error in different zones, for each plot. For this reason, this scenario is chosen to assess the throughput estimation error. As it is possible to see on Figure 5.12b, this plot shows how the error changes with respect to **RSSI** and can be split into three error zones with different behavior: $zone_1$, for values > -75 dBm, $zone_2$ values within the interval of -75 dBm < RSSI < -88 dBm and $zone_3$ for values < -88 dBm. We can extract from this data that on $zone_1$ and $zone_2$, the error is more stable and can be fitted to a linear function. However, for $zone_3$, the error starts to be diffuse and becomes hard to make a regression to correct it. In terms of **distance**, it can also be split into three zones: $zone_1$, for distances lower than 750m, $zone_2$ for distances between the interval 750m < $dist$ < 1750m and $zone_3$ for distances higher than 1750m. It is visible on Figure 5.12a that in $zone_1$, deviation is stable, for $zone_2$ deviation increases a little but still stable and on $zone_3$ even higher deviation starts to appear, becoming higher and random when it is close to 2km. As it was seen on scenario description in section 3.6, for distances higher than 1900m vehicles stop being on LOS with each other which does also influence the quick throughput drop.



(a) Deviation with respect to Distance

(b) Deviation with respect to RSSI

Figure 5.12: Deviation of simulation results to real measurements for Aveiro Experiment

The plots on figure 5.12 is filtered for both distance zones and RSSI zones, due to the fact that there are some outliers found when filtering for only 1 of these metrics alone, either distance or RSSI. Both outliers are seen on the plots from figure 5.13, where points from different zones, specially on $zone_2$ and $zone_3$ are mixed.

In order to perform a clean estimation of the error, these outliers are removed by filtering data for both distance zones and RSSI zones at the same time. For each zones the conditions are as follows, on equations 5.1, 5.2 and 5.3:

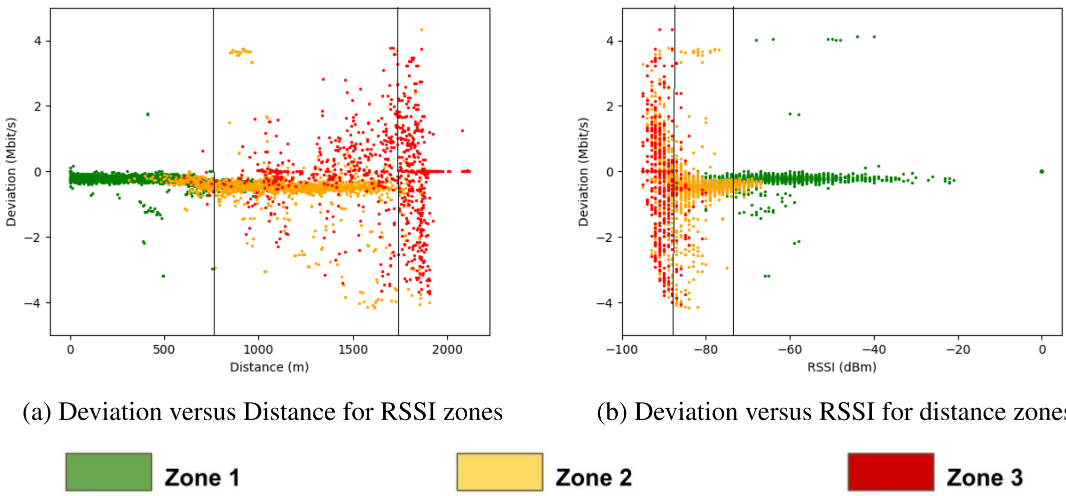
$$zone_1 = dist < 750m \;\; \& \;\; RSSI > -75dBm \tag{5.1}$$

$$zone_2 = 750m < dist < 1750m \;\; \& \;\; -88dBm < RSSI < -75dBm \tag{5.2}$$

$$zone_3 = dist > 1750m \;\; \& \;\; RSSI < -88dBm \tag{5.3}$$

With this filter in error zones already defined, we proceed to calculate the error for each experiment in each error zone, for each scenario with all trial results combined into one data set.

The following figure 5.14 (presented as table on A.1) summarizes the error values obtained for each different combination of tested models, for each scenario considered, with deviation and relative error in **absolute value**. To start off, Leça scenario does not contain information for $zone_2$



(a) Deviation versus Distance for RSSI zones (b) Deviation versus RSSI for distance zones

Figure 5.13: Problems when filtering zones only for 1 metric (RSSI or distance)

and $zone_3$. However, for Aveiro experiment is possible to better understand the error pattern. It is also clear that the mean deviation and median relative error values for zone1 are close to each other for each combination, in a range of 0.2565 - 0.315 for deviation and a range of 5.5% - 7.39% for relative error, indicating this is the difference for both scenarios. This difference is possibly due to some specific routing and advertisement information that is not being sent on the simulation environment and also with respect to overheads in communication which can result from using tunnels for the handover operations that currently run on the OBUs. Another conclusion taken from this figure is that, for both TwoRay and MBased models, using a Fading bin of 100 meters (Fad_{bin100}) reduces the deviation and the relative error. These results are highlighted on the table in bold. For $zone_1$ error values are equal. For $zone_2$, there is better value of deviation and relative error for MBased model. However, for $zone_3$, since it is the most unstable zone, the higher error on MBased model is explained due to the impact of fading when applied on top of real values instead of deterministic values. This way, MBased tends to be more "pessimist" in this zone. Despite that and overall, the model provides a good approximation and throughput graphics shown in the previous section 5.2 show that the concordance of simulation results when compared to real measurements is high.

With respect to this analysis presented on the figure 5.14, we decided to approach to correct error by making a linear regression to fit the error for each zone to the function $y = ax + b$, considering the data set of Aveiro's experiment. Linear regression results are shown in table 5.2.

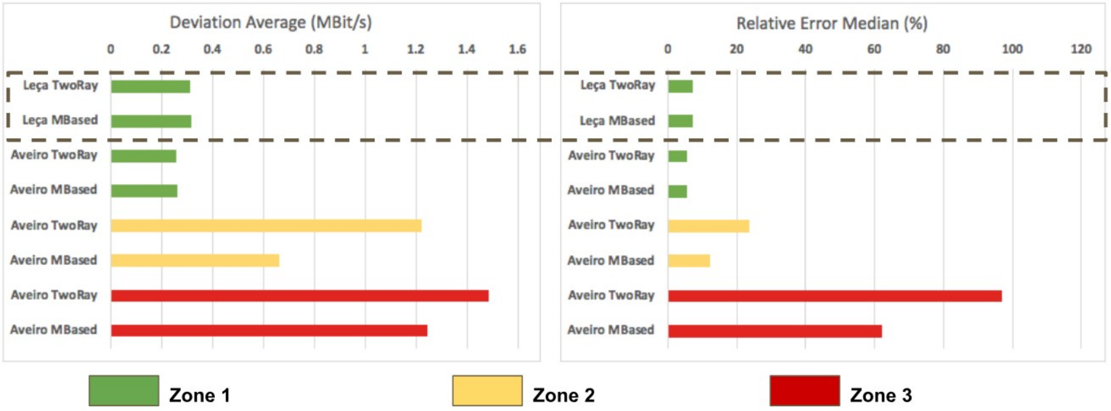
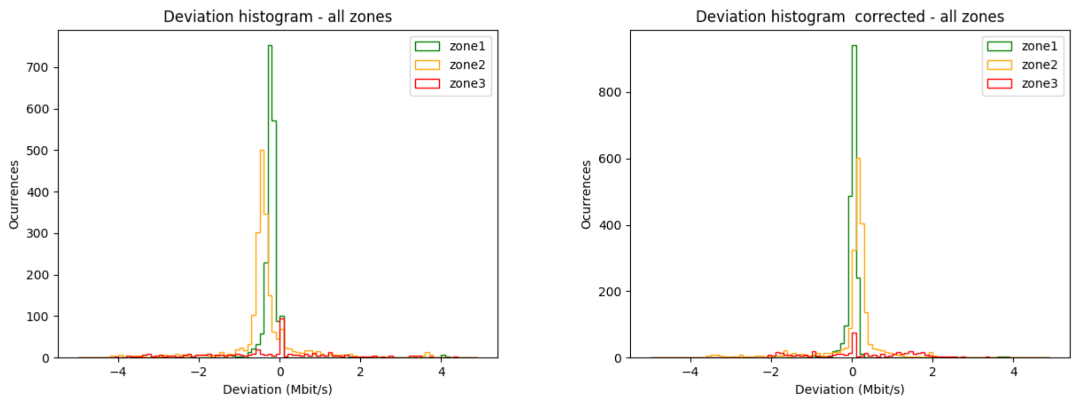


Figure 5.14: Deviation Mean and Relative Error Median for each scenario

zone	**a**	**b**
$zone_1$	0.868	0.309
$zone_2$	0.457	1.681
$zone_3$	0.309	0.936

Table 5.2: Linear regression $y = ax + b$ results for MBased model

This correction was applied to the scenario that we want to evaluate, which is MBased model + Fad_{bin100} for Aveiro experiment. After this correction being applied, a plot of the histogram of error for all zones was made, to compare with the error histogram before the correction, as shown in figure 5.15.

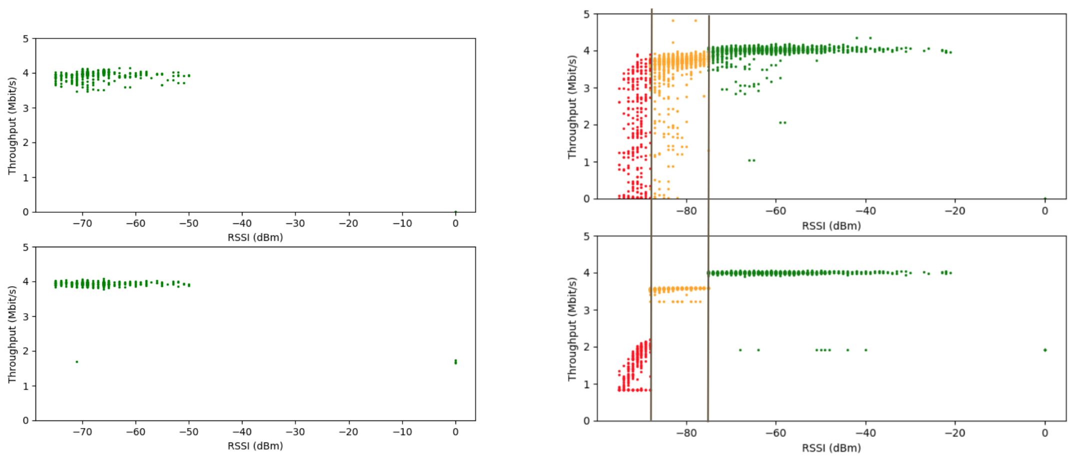


(a) Histogram - without correction (b) Histogram - with correction

Figure 5.15: Histogram of error for each zone for Aveiro experiment MBased Fad_{bin100}

It is possible to see that the deviation values were corrected so distributions from all zones tend to the value of zero. However, $zone_3$ error cannot be corrected due to the difficulties in fitting the diffuse data samples into a simple linear regression. Therefore, the correction is only able to reduce the width of data scattering in that zone.

We now proceed to analyze the throughput graphs obtained when this correction is applied, to see check how well the graphs will be fitted. The linear regression of $zone_1$ obtained from Aveiro experiment on table 5.2, is first applied to Leça experiment, to check that if the error on that zone can be corrected across scenarios. The whole correction for all zones is also applied for the whole Aveiro data set, regarding the simulation results of MBased Fad_{bin100}. Graphs on figures 5.16 and 5.17 show **empiric measurements on the top graph** and **corrected simulation results on the botton graph**, each divided and colored by the 3 corresponding zones. In the case of Leça da Palmeira, there is only data that fits into the first zone.



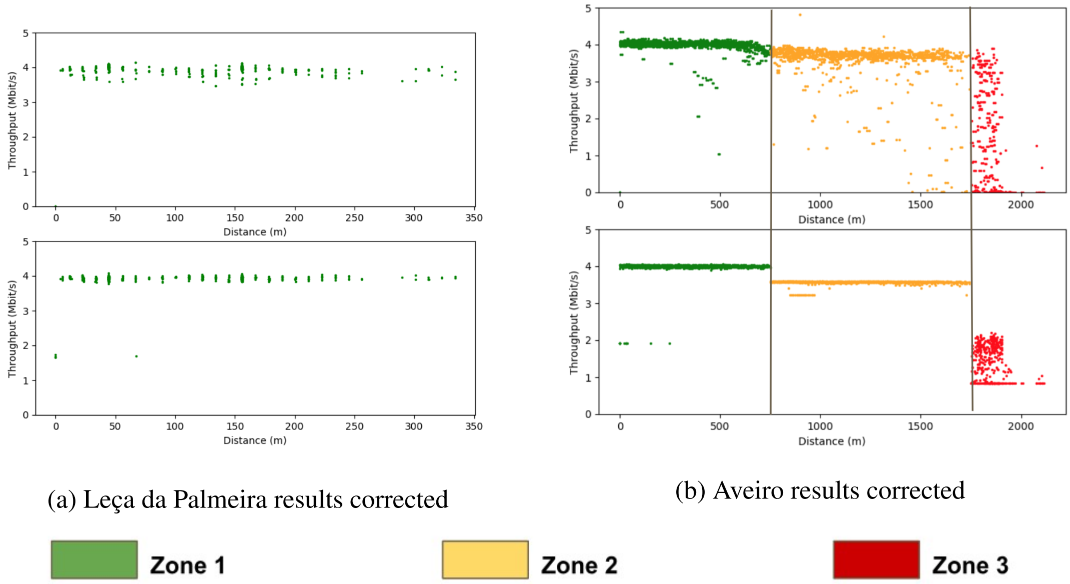
(a) Leça da Palmeira results corrected

(b) Aveiro results corrected

Zone 1 Zone 2 Zone 3

Figure 5.16: Measurement (top graph) vs Corrected (bottom graph) throughput data with respect to RSSI

In terms of RSSI the correction does decrease the maximum throughput value, becoming a bit lower than the real value for Aveiro experiment, shown in figure 5.16b, which is explained by the spread of the throughput points that result from this experiment. For Leça experiment, the correction does places the maximum throughput value on top of the real values which is about 4 Mbit/s, as shown in 5.16a, in separate graphs.



(a) Leça da Palmeira results corrected

(b) Aveiro results corrected

Figure 5.17: Measurement (top graph) vs Corrected (bottom graph) throughput data with respect to distance

Regarding throughput with respect to distance, the Leça experiment shows a good correction result since throughput shows a stable behaviour and a linear correction works well, as shown in figure 5.17a. In the case of Aveiro, the maximum throughput value tends to the level of throughput that real measurements show from 1km until near 2km, as expected with the division in several zones.

A summary of the obtained error values regarding this last correction is shown in table 5.3.

Scenario	**Model**	**zone**	**Average Deviation (MBit/s) [Before → After]**	**Relative Error Median (%) [Before → After]**
Leça	MBased	$zone_1$	0.315 → 0.104	7.3 → 1.79
Aveiro	MBased	$zone_1$	0.261 → 0.105	5.5 → 1.4
Aveiro	MBased	$zone_2$	0.664 → 0.421	12.4 → 5.3
Aveiro	MBased	$zone_3$	1.242 → 0.995	62.1 → 43.5

Table 5.3: Error values obtained after correction

Comparing these values of table 5.3, it is possible to conclude that this corrections have reduced even more the error for both Aveiro and Leça experiments. We expect that this error correction can be used to scenarios with similar conditions: LOS, low vehicle density and low density of shadowing objects. Across different scenarios, it is possible to see that throughput relative error median on $zone1$ is similar and below 10%.

Chapter 6

Conclusions

The results of this work show that a trace-based approach for modeling the environment of Vehicular Networks can be adequate, although it requires higher amount and higher granularity of data to become better. Due to a low granularity of signal measurements and GPS position measurements, it can be hard to reproduce the conditions accurately. However, there is an improvement when compared to Two-Ray model in scenarios of LOS with nearly ideal conditions. This outlines that deterministic models may not be the best approach to take when simulating this kind of networks. Regarding this proposed trace-based approach, it has up to 5.3% median error when compared to real measurements results and gives a better estimation of when metrics like Throughput will start decreasing, due to the signal information that is being used.

6.1 Contributions of this Thesis

The main focus of this thesis and its first contribution was to develop an enhanced model for channel/link layer modeling which is based on real signal measurements that come from benchmarks/tests or from a database of collected samples within production networks, applied to Vehicular Networks. The second contribution of this thesis is a study on how Throughput affects the performance of a network, when compared to real data that is obtained with dedicated experiments, validating as well some simulation aspects. An analysis of the resulting errors and deviation from the simulation results when compared to real results, with an approach to correct them is explored as well, for scenarious in the same conditions. These are all made considering LOS scenarios avoiding interference from other sources and only considering single-hop communications within the DSRC technology, since those are the scenarios that could be validated with the available data.

6.2 Future Work

There are a few aspects that could be done in order to enhance these throughput estimations in a reasonable level and reach further conclusions for specific VANET applications. Here are listed some that were relevant during this work:

- RSSI measurements can be retrieved with a higher sampling rate in order to better reconstruct the signal characteristics on the simulation environment.
- Some routing protocols can be applied and compared in order to see the impact of those additions in terms of throughput, delay and packet delivery rate.
- In order to measure congestion, which was not part of this work, some more experiments considering 2 sender vehicles and 1 receiver can be done, which will allow to better understand and build a prototype of a connection manager that can decide where to connect next.
- Following the congestion part and along with the routing protocols, some multihop scenarios can be tested, for example, to develop a content distribution framework in NS-3, by using this measurement-based channel model as the base propagation model.

Appendix A

Appendix

This section provides some samples of the data that was available from experiments, in order to understand which fields can be obtained.

A.1 Trace Data Sample


```
{
    "netrider_id": 2369,
    "latency": {
      "ap": {
        "ts": 1521568737409,
        "dsrc": 43
      }
    },
    "sys_time": 1521568739184,
    "connection": {
      "rssi": 0,
      "type": 2,
      "start": 1521567169138,
      "hops": 2,
      "endpoint": 2393,
      "next_hop": 2364
    },
    "handover": {
      "to": {
        "rssi": 0,
        "type": 2,
        "start": 1521567169138,
        "hops": 2,
        "endpoint": 2393,
        "next_hop": 2364
      },
      "from": {
        "rssi": 0,
        "type": 0,
        "start": 0,
        "hops": 0,
        "endpoint": 0,
        "next_hop": 0
      },
      "time": 5
    },
    "opportunistic": {
      "total": 0,
      "pending": 0,
      "free": 14701760512
    },
    "gps": {
      "lat": 41.2167,
      "lon": -8.71308,
      "heading": 187.01,
```

```
      "alt": 18.7,
      "speed": 0.03704
    },
    "neighbours": [
      {
        "netrider_id": 2364,
        "gps": {
          "lat": 41.2181,
          "lon": -8.71293,
          "heading": 122.3,
          "alt": 17.7,
          "speed": 0.0
        },
        "connection": {
          "rssi": 24,
          "type": 2,
          "start": 0,
          "hops": 1,
          "endpoint": 2393,
          "next_hop": 2393
        }
      },
      {
        "netrider_id": 2393,
        "gps": {
          "lat": 41.2186,
          "lon": -8.71287,
          "heading": 224.0,
          "alt": 21.8,
          "speed": 0.0
        },
        "connection": {
          "rssi": 17,
          "type": 0,
          "start": 0,
          "hops": 0,
          "endpoint": 0,
          "next_hop": 0
        }
      }
    ],
    "per": {
      "dsrc": {
        "rx": {
          "perc": 0.141001,
          "total": 502123,
          "err": 708
        },
        "tx": {
          "perc": 0.000337548,
          "total": 296254,
          "err": 1
        }
      }
    },
    "diagnose": {
      "load": [
        {
          "cpu0": 0.191011
        }
      ],
      "mem_available": 0.842841
    },
    "seq": 5
  },
```

A.2 Error results table

Scenario	**Model**	**zone**	**Deviation Average (MBit/s)**	**Relative Error Median (%)**
Leça	TwoRay	$zone_1$	0.311	7.39
Leça	MBased	$zone_1$	0.315	7.3
Aveiro	TwoRay	$zone_1$	**0.2565**	**5.5**
Aveiro	MBased	$zone_1$	**0.2606**	**5.5**
Aveiro	TwoRay	$zone_2$	**1.2229**	**23.6**
Aveiro	MBased	$zone_2$	**0.6643**	**12.4**
Aveiro	TwoRay	$zone_3$	**1.487**	**96.9**
Aveiro	MBased	$zone_3$	**1.2418**	**62.1**

Table A.1: Error values obtained for both models using fading Fad_{bin100}

Bibliography

[1] C. Jeremiah and A. J. Nneka, "Issues and possibilities in Vehicular Ad-hoc Networks (VANETs)," in *Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), 2015 International Conference on*. IEEE, 2015, pp. 254–259.

[2] C. Ameixieira, A. Cardote, F. Neves, R. Meireles, S. Sargento, L. Coelho, J. Afonso, B. Areias, E. Mota, and R. Costa, "Harbornet: A real-world testbed for vehicular networks," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 108–114, 2014.

[3] S. Rehman, M. A. Khan, T. Zia, and L. Zheng, "Vehicular ad-hoc networks (vanets) — an overview and challenges," vol. 3, pp. 29–38, 01 2013.

[4] M. Noussaiba and R. Rahal, "State of the art: VANETs applications and their RFID-based systems," in *Control, Decision and Information Technologies (CoDIT), 2017 4th International Conference on*. IEEE, 2017, pp. 0516–0520.

[5] H. Moustafa and Y. Zhang, Eds., *Vehicular Networks: Techniques, Standards, and Applications*. Auerbach Publications, Apr. 2009. [Online]. Available: http://www.crcnetbase.com/doi/book/10.1201/9781420085723

[6] E. Spaho, L. Barolli, G. Mino, F. Xhafa, and V. Kolici, "VANET Simulators: A Survey on Mobility and Routing Protocols." IEEE, Oct. 2011, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/document/6103008/

[7] R. Meireles, P. Steenkiste, J. Barros, and D. C. Moura, "LASP: Look-ahead spatial protocol for vehicular multi-hop communication," in *Vehicular Networking Conference (VNC), 2016 IEEE*. IEEE, 2016, pp. 1–8.

[8] ETSI EN 302 665, "Intelligent transport systems (ITS); communications architecture," Version 1.1.1, ETSI ITS WG2, Sophia Antipolis, France, Sep. 2010. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf

[9] W. Liang, Z. Li, H. Zhang, S. Wang, and R. Bie, "Vehicular Ad Hoc Networks: Architectures, Research Issues, Methodologies, Challenges, and Trends," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 745303, Aug. 2015. [Online]. Available: http://journals.sagepub.com/doi/10.1155/2015/745303

[10] K. Schwieger and G. Fettweis, "Multi-Hop Transmission: Benefits and Deficits," p. 3.

[11] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for Information technology– telecommunications and information exchange between systems– local and metropolitan area networks– specific requirements: Part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications : Amendment 6: Wireless access in vehicular environments*. New York: Institute of Electrical and Electronics Engineers, 2010, oCLC: 670422348. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=5514473

[12] R. A. Uzcátegui, A. J. De Sucre, and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications magazine*, vol. 47, no. 5, 2009.

[13] W.-Y. Lin, M.-W. Li, K.-C. Lan, and C.-H. Hsu, "A Comparison of 802.11a and 802.11p for V-to-I Communication: A Measurement Study," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, X. Zhang and D. Qiao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 74, pp. 559–570. [Online]. Available: http://link.springer.com/10.1007/978-3-642-29222-4_39

[14] K. Sjöberg, "Standardization of Wireless Vehicular Communications within IEEE and ETSI," p. 25.

[15] A. Festag, "Standards for vehicular communication—from IEEE 802.11p to 5g," *e & i Elektrotechnik und Informationstechnik*, vol. 132, no. 7, pp. 409–416, Nov. 2015. [Online]. Available: http://link.springer.com/10.1007/s00502-015-0343-0

[16] C. Hedges and F. Perry, "Overview and use of sae j2735 message sets for commercial vehicles," in *SAE Technical Paper*. SAE International, 10 2008. [Online]. Available: https://doi.org/10.4271/2008-01-2650

[17] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 166–172, Dec. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6979970/

[18] E. G. Strom, "On Medium Access and Physical Layer Standards for Cooperative Intelligent Transport Systems in Europe," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1183–1188, Jul. 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5771027/

[19] E. Eze, S. Zhang, and E. Liu, "Vehicular ad hoc networks (vanets): Current state, challenges, potentials and way forward," pp. 176–181, 09 2014.

[20] S. Singh, P. Kumari, and S. Agrawal, "Comparative Analysis of Various Routing Protocols in VANET." IEEE, Feb. 2015, pp. 315–319. [Online]. Available: http://ieeexplore.ieee.org/document/7079099/

[21] B. V. Coutinho, E. C. Wille, and H. I. Del Monego, “Performance of routing protocols for VANETs: A realistic analysis format,” in *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*. ACM, 2015, p. 1.

[22] J. Garcia-Campos, D. Reina, S. Toral, N. Bessis, F. Barrero, E. Asimakopoulou, and R. Hill, “Performance Evaluation of Reactive Routing Protocols for VANETs in Urban Scenarios Following Good Simulation Practices.” IEEE, Jul. 2015, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/7284919/

[23] W. Viriyasitavat, M. Boban, Hsin-Mu Tsai, and A. Vasilakos, “Vehicular Communications: Survey and Challenges of Channel and Propagation Models,” *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 55–66, Jun. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7108160/

[24] D. W. Matolak, “Modeling the vehicle-to-vehicle propagation channel: A review: MODELING V2v PROPAGATION CHANNEL,” *Radio Science*, vol. 49, no. 9, pp. 721–736, Sep. 2014. [Online]. Available: http://doi.wiley.com/10.1002/2013RS005363

[25] A. Molisch, F. Tufvesson, J. Karedal, and C. Mecklenbrauker, “A survey on vehicle-to-vehicle propagation channels,” *IEEE Wireless Communications*, vol. 16, no. 6, pp. 12–22, Dec. 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5361174/

[26] L. Cheng, B. Henty, D. Stancil, F. Bai, and P. Mudalige, “Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frequency Band,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4346439/

[27] N. Akhtar, S. C. Ergen, and O. Ozkasap, “Vehicle Mobility and Communication Channel Models for Realistic and Efficient Highway VANET Simulation,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 248–262, Jan. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/6803095/

[28] A. Cardote, F. Neves, S. Sargento, and P. Steenkiste, “A statistical channel model for realistic simulation in VANET,” in *Vehicular Networking Conference (VNC), 2012 IEEE*. IEEE, 2012, pp. 48–55.

[29] M. G. Nilsson, C. Gustafson, T. Abbas, and F. Tufvesson, “A Measurement-Based Multilink Shadowing Model for V2v Network Simulations of Highway Scenarios,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8632–8643, Oct. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7934405/

[30] M. Boban, J. Barros, and O. K. Tonguz, “Geometry-Based Vehicle-to-Vehicle Channel Modeling for Large-Scale Simulation,” *IEEE Transactions on Vehicular*

Technology, vol. 63, no. 9, pp. 4146–4164, Nov. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6799304/

[31] R. J. Fernandes, "Large-scale Simulation of Vehicular Ad Hoc Networks," PhD Thesis, Universidade do Porto (Portugal), 2014.

[32] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 19–41, 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5343061/

[33] D. Mohan, "A SURVEY ON VANET ARCHITECTURE, CHALLENGES AND MOBILITY MODELS," 2016.

[34] V. Kolici, T. Oda, E. Spaho, L. Barolli, M. Ikeda, and K. Uchida, "Performance Evaluation of a VANET Simulation System Using NS-3 and SUMO." IEEE, Mar. 2015, pp. 348–353. [Online]. Available: http://ieeexplore.ieee.org/document/7096199/

[35] V. W. Kurhade, A. R. Deshmukh, and S. S. Dorle, "COMPARATIVE ANALYSIS OF MOBILITY MODELS USING ROUTING PROTOCOL FOR VEHICULAR AD-HOC NETWORK (VANET)," 2017.

[36] C. Sommer, J. Härri, F. Hrizi, B. Schünemann, and F. Dressler, "Simulation Tools and Techniques for Vehicular Communications and Applications," in *Vehicular ad hoc Networks*, C. Campolo, A. Molinaro, and R. Scopigno, Eds. Cham: Springer International Publishing, 2015, pp. 365–392. [Online]. Available: http://link.springer.com/10.1007/978-3-319-15497-8_13

[37] Simulation of Urban Mobility - SUMO. [Online]. Available: http://sumo.dlr.de/

[38] N. M. Mittal and S. Choudhary, "Comparative study of simulators for vehicular ad-hoc networks (vanets)," *International Journal of Emerging Technology and Advanced Engineering Website: www. ijetae. com (ISSN 2250-2459, ISO 9001: 2008 Certified Journal, Volume 4, Issue 4*, 2014.

[39] Omnet++ - discrete event simulator. [Online]. Available: https://www.omnetpp.org/

[40] R. Meireles, M. Ferreira, and J. Barros, "Vehicular connectivity models: From single-hop links to large-scale behavior," in *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*. IEEE, 2009, pp. 1–5.

[41] H. Fontes, R. Campos, and M. Ricardo, "A Trace-based ns-3 Simulation Approach for Perpetuating Real-World Experiments." ACM Press, 2017, pp. 118–124. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3067665.3067681

[42] H. Fernandez, L. Rubio, V. M. Rodrigo-Penarrocha, and J. Reig, "Path Loss Characterization for Vehicular Communications at 700 MHz and 5.9 GHz Under LOS and NLOS Conditions,"

IEEE Antennas and Wireless Propagation Letters, vol. 13, pp. 931–934, 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6810849/

[43] S. E. Carpenter and M. L. Sichitiu, "An obstacle model implementation for evaluating radio shadowing with ns-3." ACM Press, 2015, pp. 17–24. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2756509.2756512

[44] C. Celes, F. A. Silva, A. Boukerche, R. M. d. C. Andrade, and A. A. F. Loureiro, "Improving VANET Simulation with Calibrated Vehicular Mobility Traces," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3376–3389, Dec. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7895148/

[45] Y. Yang, S.-Y. Hao, and H.-B. Cai, "Comparison and Evaluation of Routing Protocols Based on a Collaborative Simulation Using SUMO and NS3 with TraCI," in *Information System and Artificial Intelligence (ISAI), 2016 International Conference on*. IEEE, 2016, pp. 253–257.

[46] W. Liu, X. Wang, W. Zhang, L. Yang, and C. Peng, "Coordinative simulation with SUMO and NS3 for vehicular ad hoc networks," in *Communications (APCC), 2016 22nd Asia-Pacific Conference on*. IEEE, 2016, pp. 337–341.

[47] R. Fernandes and M. Ferreira, "Scalable vanet simulations with ns-3," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*. IEEE, 2012, pp. 1–5.

[48] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5510240/

[49] S. A. B. Mussa, M. Manaf, K. Z. Ghafoor, and Z. Doukha, "Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives," in *Wireless Networks and Mobile Communications (WINCOM), 2015 International Conference on*. IEEE, 2015, pp. 1–8.

[50] Network simulator - ns-3. [Online]. Available: http://www.nsnam.org/

[51] Netanim 3.108 - nsnam. [Online]. Available: https://www.nsnam.org/wiki/NetAnim_3.108

[52] M. Lacage and T. R. Henderson, "Yet another network simulator." ACM Press, 2006, p. 12. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1190455.1190467

[53] P.1411 : Propagation data and prediction methods for the planning of short-range outdoor radiocommunication systems and radio local area networks in the frequency range 300 mhz to 100 ghz. [Online]. Available: https://www.itu.int/rec/R-REC-P.1411/en

[54] G. Pei and T. R. Henderson, "Validation of OFDM error rate model in ns-3," p. 15.

[55] K. Katsaros. Vehicular communication simulations with NS-3. [Online]. Available: https://www.nsnam.org/tutorials/consortium15/Vehicular-Comms.pptx

[56] iperf - iperf3 and iperf2 user documentation. [Online]. Available: https://iperf.fr/iperf-doc.php

[57] G. Carneiro, P. Fortuna, and M. Ricardo, "FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3)." ICST, 2009. [Online]. Available: http://eudl.eu/doi/10.4108/ICST.VALUETOOLS2009.7493

[58] K. Noga and R. Studański, "Estimation of Nakagami distribution parameters in describing a fading radio-communication channel," *Zeszyty Naukowe Akademii Marynarki Wojennej*, vol. 204, no. 1, pp. 69–81, Mar. 2016. [Online]. Available: http://9824.indexcopernicus.com/abstracted.php?level=5&ICID=1202437

[59] Q. T. Zhang, "m A Note on the Estimation of Nakagami- Fading Parameter," p. 2.

[60] J. Gaeddert and A. Annamalai, "Further results on Nakagami-m parameter estimation," *IEEE Communications Letters*, vol. 9, no. 1, pp. 22–24, 2005. [Online]. Available: http://ieeexplore.ieee.org/document/1375229/

[61] R. Mitra, A. K. Mishra, and T. Choubisa, "Maximum likelihood estimate of parameters of Nakagami-m distribution." IEEE, Dec. 2012, pp. 9–12. [Online]. Available: http://ieeexplore.ieee.org/document/6422123/