UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

PERFORMANCE OF SENSING-BASED SEMI-PERSISTENT SCHEDULING (SPS) IN LTE-V2X RELEASE 14 DISTRIBUTED MODE

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Leandro Miguel Wong Lopez

In partial fulfilment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisors: Jordi Casademont-Serra, PhD

Daniel Camps-Mur, PhD

Barcelona, May 2019

Title of the thesis: Performance of Sensing-Based Semi-Persistent Scheduling in LTE-V2X Release 14 Distributed Mode

Author: Leandro Miguel Wong Lopez

Advisors: Jordi Casademont-Serra, PhD and Daniel Camps-Mur, PhD

Abstract

The initial standard for cellular-based Vehicle-to-everything (V2X) communications was introduced in 2017 by 3GPP in Long Term Evolution (LTE) Release 14 to serve as a viable alternative to the mature yet dated WLAN-based 802.11p technology. LTE-V2X Release 14 introduced a new arrangement of the resource grid as well as a sensing-based semi-persistent scheduling (SPS) algorithm for the distributed mode in order to reduce latency and increase capacity. A simulator based on open-source software frameworks was developed to evaluate the performance of the Release 14 sensing-based SPS and random allocation in scenarios with varying traffic loads, message sizes, resource keep probabilities *P*, and collision power thresholds. The performance was then evaluated in terms of Packet Reception Ratio (PRR), occupancy, and goodput, Neighborhood Awareness Ratio (NAR), position error, and latency. Simulation results showed that sensing-based SPS generally performed better than random allocation in terms of PRR in short to medium distances. Sensing-based SPS configured with *P*=0 performed only slightly better than random allocation in terms of NAR but slightly worse in terms of position error. However, with sufficiently high message traffic, sensing-based SPS performed similar to, or even worse than random allocation.

Acknowledgements

My sincere thanks to my advisors, Prof. Jordi Casademont-Serra and Dr. Daniel Camps-Mur for their valuable guidance and expertise. Thanks to Joan Josep Aleixendri Cruelles of i2CAT for the technical support, especially during the results gathering phase of the project. Finally, thanks to my colleague and fellow thesis advisee Charmae Franchesca Mendoza for her cooperation and assistance.

Revision history and approval record

Revision	Date	Purpose
0	31/03/2019	Document creation
1	09/05/2019	Document revision

Written by:		Reviewed and approved by:	
Date	07/05/2019	Date	09/05/2019
Name	Leandro Miguel Wong Lopez	Name	Jordi Casademont-Serra
Position	Project Author	Position	Project Supervisor

Table of Contents

List of Figures

List of Tables

1. Introduction

Vehicle-to-Everything (V2X) is the technology that enables any vehicle to communicate with any other entity. V2X itself is an umbrella term, but it can be classified into more specific categories depending on the type of entity the vehicle communicates with, namely: Vehicle-to-Vehicle (V2V), Vehicle-to-Pedestrian (V2P), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Network (V2N) Figure 1. V2X is considered to be a revolutionary technology that will tremendously improve road safety, traffic efficiency, and overall transport experience as it unlocks a myriad of new services, ranging from the seemingly mundane traffic alerts to the more futuristic autonomous driving.

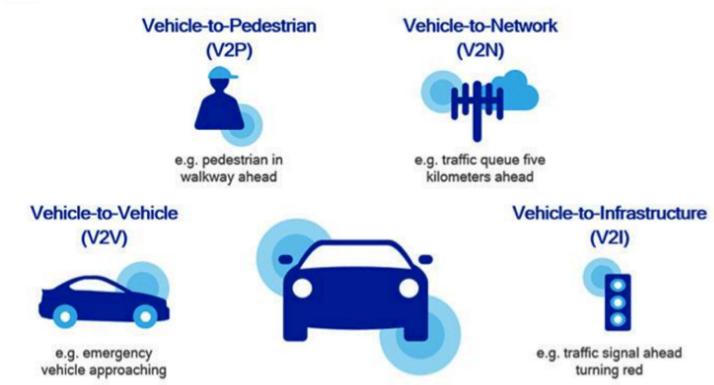

Figure 1. Types of V2X (Source: Qualcomm)

There are two main V2X standard suites that have been developed as of the time of this writing, namely Dedicated Short-Range Communications (DSRC), which was developed in the USA, and the Intelligent Transportation System (ITS)-G5 standards, which was developed by the European Telecommunication Standards Institute (ETSI). Both of these standard suites are based on communication using Wireless Local Area Network (WLAN) technology, specifically, using the IEEE 802.11p standard.

WLAN-based V2X is widely regarded as the most mature V2X standard with over 10 years of field trials. In spite of this, IEEE 802.11p is facing several challenges. For instance, IEEE 802.11p makes use of Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), wherein nodes only transmit when they sense that the channel is idle in order to avoid collisions. The well-known problem with relying on CSMA/CA is that under heavy traffic conditions, such as a dense urban scenario, CSMA/CA suffers from a high level of collisions, mainly due to hidden terminal situations. This problem will only become worse with the growing demand for autonomous vehicles, yet in spite of this, there are no clear plans to enable improvements in the technology. Another challenge from a business perspective is that IEEE 802.11p faces is that significant investment is required to deploy specialized Roadside Units (RSU) in order to provide V2I-related services such as route recommendation and curve speed warnings.

The first real alternative to WLAN-based V2X came out in 2017 when the 3rd Generation Partnership Project (3GPP) presented cellular-based V2X (C-V2X) through the introduction of V2X support in 3GPP Long Term Evolution (LTE) Advanced Release 14 [1]. LTE-based V2X (LTE-V2X), enjoys numerous technological advantages over

WLAN-based V2X. For instance, LTE-V2X makes use of Single-Carrier Frequency Division Multiple Access (SC-FDMA) instead of Orthogonal Frequency Division Multiplex (OFDM), which permits more transmit power given the same power amplifier. Another technological advantage is that the LTE-V2X resource selection has no contention overhead, leading to a more deterministic latency even in congested scenarios. C-V2X in general also has the business advantage of being able to leverage the existing cellular infrastructure unlike IEEE 802.11p, which requires the deployment of entirely new equipment. This would greatly reducing deployment costs. Finally, C-V2X has a clear evolutionary path from LTE-V2X Release 14 towards future releases in 5G, unlike IEEE 802.11p, which was not evolved and improved [2]. Nevertheless, as of the time of writing this thesis, there is still no clear consensus whether to select WLAN or cellular networks (or a hybrid of both) as the superior means of implementing V2X communications.

1.1. Objectives and Scope

Being a relatively young technology, there is a lot of ongoing of research activity on understanding better the true potential of 3GPP LTE-V2X Release 14. This will not only help in ultimately deciding *which* communications technology to use, but also *how* to use the technology in an optimal manner. As such, this project aims to contribute to existing research by studying the performance of the 3GPP LTE-V2X Release 14 standard sensing-based Semi-Persistent Scheduling (SPS) algorithm in Mode 4 communications under different configurations, and by comparing it with a random resource allocation algorithm through an open-source network simulation platform.

There are numerous possible combinations of simulation variables that could be experimented with in order to test their impact on the performance, but the study limited the scope to the following parameters: traffic load (by varying the number of vehicles through the road traffic scenario size and/or vehicle speeds), message size (in terms of *subchannels* per message), keep probability (in case sensing-based SPS was used), and collision power threshold. Most of the simulations were performed on a bidirectional highway environment, although a unidirectional highway and Manhattan grid was also used in a few cases.

The performance was then viewed from the lens of the following performance metrics: the Medium Access Control (MAC)-level metrics of Packet Reception Ratio (PRR), resource grid occupancy and goodput; and the application-level metrics of Neighborhood Awareness Ratio (NAR), position error, and latency.

1.2. Workflow

The project was undertaken for a total duration of 8 ½ months, from the start of September 2018 until mid-May 2019. The project was divided into three main phases: the *study* phase, the *development* phase, and the *results and analysis* phase. A Gantt chart depicting the overall workflow and schedule of the project is shown in Figure 2.

The first 1 ½ months was allocated to the study phase, which involved understanding the fundamental concepts necessary for the project, deciding the simulation frameworks to use, familiarization with the chosen simulation frameworks and related programming languages, as well as reviewing related research projects. The study phase of the project was allocated a relatively short time because several of the concepts necessary for the

project had already been covered from *Introduction to Research* subjects taken in semesters prior to the master's thesis project.

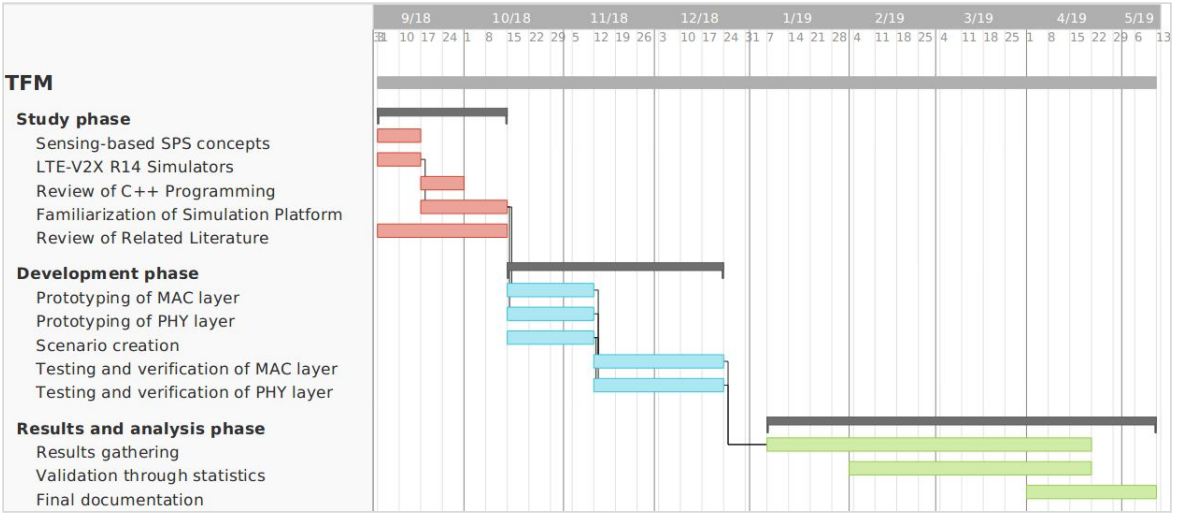

Figure 2. Project Schedule

The next 2 ½ months was devoted to the development phase, which focused on the actual implementation of the LTE-V2X Mode 4 simulation. Most of the work was on developing a model of the Network Interface Card (NIC) that contains the Medium Access Control (MAC) layer and the physical layer (PHY) functionalities, but some work was also necessary in designing road network scenarios for the simulations.

The final 4 ½ months was for the results and analysis phase, which involved running simulations, gathering the results, and then checking if the results matched the theory. This phase was the most time consuming, mostly because of the long periods of time that are necessary to run the simulations and present the data in intuitive graphs. Another reason is that there were several instances when the resulting statistics were far from expected or were unreasonable, thereby entailing the need to tweak the previously developed models and do a repetition of the simulations to gather a new set of results.

1.3. Outline

Chapter 2 explains the basics of LTE-V2X, focusing on the concepts necessary to understanding this dissertation, as well as a review of related literature. Chapter 3 describes the methodology used to develop the simulation platform. Chapter 4 discusses the simulation results. Chapter 5 briefly explains the budget allocated for the project. Chapter 6 concludes the report and gives recommendations on further analyses that could be taken. Finally, the Appendix section provides quick user manuals to anyone who is interested in replicating the simulation framework used in this project, as well as a copy of the paper written by the thesis author regarding the impact of LTE-V2X resource grid design on its capacity and efficiency (under revision) [33].

2. State of the Art

This chapter provides an overview of some concepts in LTE-V2X Release 14 that may aid in not only understanding the technology, but also in appreciating the context of the thesis project (Section 2.1). The following section gives a glimpse of potential use cases and possible enhancements in future versions of C-V2X (Section 2.2). Finally, the chapter also presents related research studies involving the study of the performance of LTE-V2X Release 14 Mode 4 (Section 2.3).

2.1. LTE-V2X Release 14 Concepts

With the primary goal of providing support for public safety applications, 3GPP introduced in 2016 through Release 12 initial support for Device-to-Device (D2D) communication, which allows LTE devices to directly communicate with each other without the need for a base station. This makes LTE a competitive technology for backup public safety communications since LTE devices could remain functional even if the eNB is not accessible. On top of that, LTE-D2D may also be useful in commercial location-based applications as well, such as social networking, gaming, video sharing, and so on. Direct communication between devices with each other has the advantage of improving spectrum utilization, as well as energy efficiency and throughput. However, LTE-D2D was not developed with vehicular communications in mind, and as a result, the D2D specifications in Release 12 was not enough to support the use case of vehicular communications, which requires support for higher channel variability, lower latency, and increased density of devices. Building upon the D2D standard to provide enhancements specifically for V2X, 3GPP introduced support for V2X communications in Release 14 in 2016.

In the following sections, the basic LTE-V2X Release 14 use cases (Section 2.1.1) as well as the service requirements based on these use cases (Section 2.1.2) are presented. The different communication options (Section 2.1.3), channels (Section 2.1.4), and modes (Section 2.1.5) are then discussed, followed by additional technical features that were included in order to support high speed and high density scenarios in vehicular communications (Section 2.1.6).

2.1.1. Use Cases

There is a multitude of potential use cases in which LTE-V2X can be applied. Shown in the table below are examples of use cases identified by 3GPP for LTE-V2X Release 14.

Use Case	Description
Forward Collision Warning	A vehicle driver is warned of an impending rear-end collision with another vehicle ahead in traffic (rear-end collision).
Control Loss Warning	A vehicle self-generates a control loss event to surrounding vehicles.
Emergency Vehicle Warning	A vehicle can obtain information (location, speed, direction) of a nearby emergency vehicle (ambulance, fire trucks, etc.) in order to help free the path.
Emergency Stop	A vehicle warns other vehicles that it is suddenly stopping (e.g. due to engine breakdown).

Queue Warning	A vehicle is informed beforehand of queues of vehicles that can cause traffic delays.
Wrong Way Driving Warning	A vehicle is warned that it is driving the wrong way on a road.
Pre-Crash Sensing Warning	A vehicle is warned that an imminent and unavoidable collision is about to happen.
Curve Speed Warning	A vehicle is warned when it is approaching a curve or exit on the road too quickly.
Warning against Pedestrian Collision	A pedestrian is warned of a moving vehicle in case a dangerous situation is detected.
Vulnerable Road User (VRU) Safety	A vehicle is warned of the presence of a VRU (e.g. a wheelchair user that is crossing the road) and alerts the driver of any imminent threats.
Remote diagnosis and Just In Time Repair Notification	A vehicle may report its current functional state to a local or remote diagnosis center through an RSU in order to receive any suggested repair or maintenance information.

Table 1. Examples of V2X use cases

The use cases presented above is an incomplete list of those identified by 3GPP for LTE-V2X Release 14, and the rest may be found in [3]. There are more potential use cases for LTE-V2X release 14 that have been identified by other parties such as communications equipment vendors and national transportation agencies. For instance, in Figure 3 below, Qualcomm identifies a Intersection Movement Assist (IMA) use case, wherein vehicles indicate when there is a high collision probability when entering an intersection, and a Do Not Pass Warning (DNPW) use case, wherein vehicles are warned when it is unsafe to pass a slower-moving vehicle ahead.

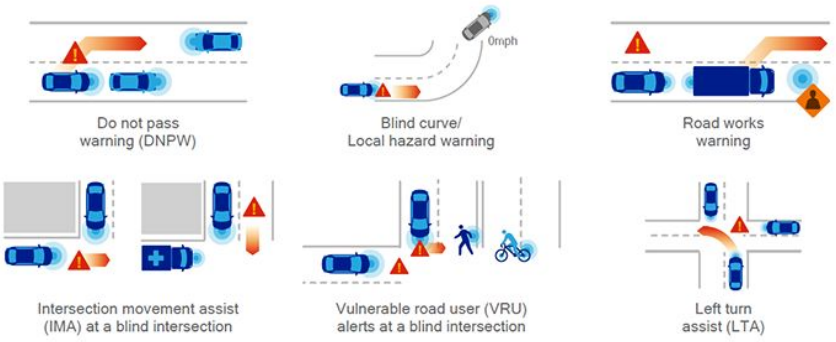

Figure 3. Illustration of LTE-V2X use cases (Source: Qualcomm)

2.1.2. Service Requirements

LTE-V2X must be able to have minimum specifications so that the technology may be able to satisfactorily provide the services set forth in the previous Section 2.1.1. For instance, in the Forward Collision Warning use case, it would be necessary that LTE-V2X should be able to support a communication range that is large enough to give the driver enough time to respond. The service requirements for LTE V2X release 14 are described in the 3GPP technical specifications [4]. Some of these requirements are listed on the table below.

Requirement Type	Details
General	Support for pre-configuration of a UE with the parameters to be used when not served by E-UTRAN supporting V2X communications. Variable transmission rate and range of the V2X communication based on service conditions (e.g., UE speed, UE density). Ability of the E-UTRAN to support a high density of UEs supporting V2X application. Prioritization of transmission of messages according to their type (e.g. safety vs. non-safety). Minimization of impact on UEs supporting V2X application with limited resources (e.g., battery), due to message transfer.
Latency and Reliability	Maximum latency of 100ms. For particular cases, such as pre-crash sensing, the maximum latency is 20ms. Support for high reliability without requiring application-layer message retransmissions.
Message Size	Periodic broadcast message (CAM): payload of 50-300 bytes without security-related content. Event-triggered message (DENM): payload of up to 1200 bytes without security-related content.
Frequency	Maximum frequency of 10 messages per second.
Range	Sufficient range to give the driver(s) ample time to respond (e.g. 4 seconds).
Speed	Entities both supporting V2V: maximum relative velocity of 500 km/h. Entities supporting V2V and V2P respectively: maximum absolute velocity of 250 km/h. Entities both supporting V2I: maximum absolute velocity of 250 km/h.

Table 2. Examples of LTE V2X Release 14 service requirements

2.1.3. Communication Options

LTE-V2X release 14 has two options for communication, namely communication (1) over the *sidelink*, or more technically, the *PC5* interface between the vehicles, and (2) over the *LTE-Uu* interface, i.e. the interface between the vehicle and the EUTRAN. V2X communication over the sidelink is referred to as direct communication because the communication between vehicles takes place independent of the cellular network. In contrast, V2X communication over the LTE-Uu interface is referred to as network communication because assistance from the network is required in order to enable V2X communication. It is possible to use both of these two communication options independently for transmission and reception. The two communication options are illustrated in Figure 4.

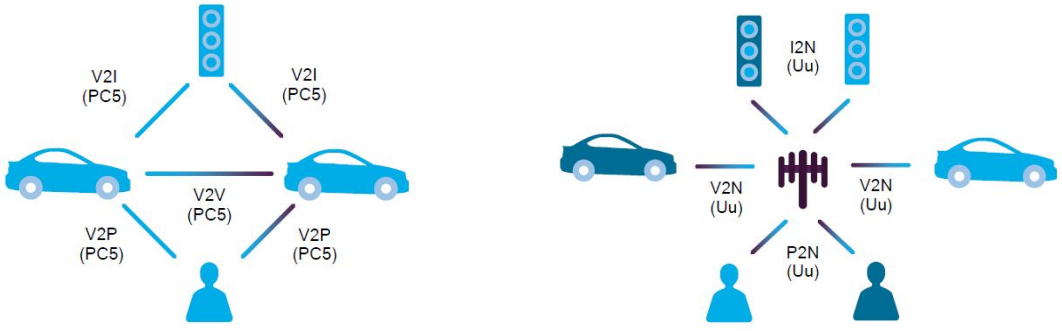

Figure 4. V2X communication using PC5 (left) versus LTE-Uu (right) interface (Source: 5GAA)

2.1.4. Communication Channels

LTE makes use of different communication channels to define the information flows between the different protocol layers for efficient data transport: physical channels for physical layer services, transport channels for services between the physical and the Medium Access Control (MAC) layer, and logical channels for services between the MAC and RLC layer. Basically, the RLC layer passes data to the MAC layer through the logical channels, the MAC layer formats and sends the logical channel data through the transport channel, and the physical layer encodes the transport channel data to the physical channels.

To add to the existing channels in the LTE downlink and uplink, LTE-D2D in release 12 introduces new logical, transport and physical channels for the sidelink. The same channels used in LTE-D2D are also used in LTE-V2X in release 14. A mapping of the channels is shown in Figure 5.

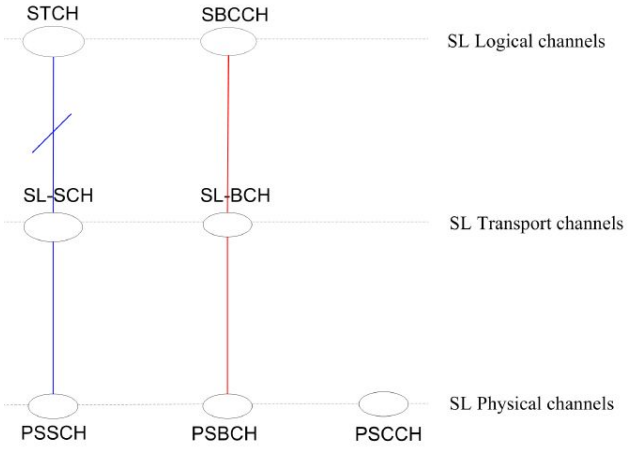

Figure 5. Sidelink Channels (Source: Rohde & Schwarz)

Looking first at the data path indicated in blue, the Sidelink Traffic CHannel (STCH) is a point-to-multipoint logical channel meant for carrying user information from a UE to other UEs. STCH is mapped to the Sidelink Shared Channel (SL-SCH), a transport channel that supports both scheduled and autonomous resource selection (see Section 2.1.5 for more details) though it has a collision risk if autonomous resource selection is used. SL-SCH is then mapped to the Physical Sidelink Shared Channel (PSSCH), which basically carries the sidelink data over the air. Its applicable modulation schemes are QPSK and 16QAM.

Now, looking at the control path indicated in red, the Sidelink Broadcast Control CHannel (SBCCH) is the logical channel in charge of broadcasting sidelink system information from a UE to other UEs. SBCCH is mapped to the Sidelink Broadcast Channel (SL-BCH), which carries the MIB-SL-V2X (Master Information Block-SL-V2X) Radio Resource Control (RRC) message for indicating timing information and configuration parameters. SL-BCH is in turn mapped to the Physical Sidelink Broadcast Channel (PSBCH). PSBCH

carries the system information and synchronization signals, and its application modulation scheme is QPSK only.

The Physical Sidelink Control Channel (PSCCH) carries the Sidelink Control Information (SCI) which includes the necessary information for the receiving UE to be able to receive and demodulate the PSSCH. Its application modulation scheme is QPSK only.

2.1.5. Communication Modes

In LTE-D2D communication, there are two modes of communication: Mode 1 and Mode 2. Mode 1 utilizes scheduled resource allocation, that is, the resources are selected and assigned by the network. On the contrary, Mode 2 utilizes autonomous resource selection, that is, the vehicle selects resources on its own, regardless of its connection status to the network. However, in order to meet the strict latency and reliability requirements of V2X, two new modes specifically meant for LTE-V2X have been introduced in Release 14, namely Mode 3 and Mode 4, which are analogous to Mode 1 and Mode 2, respectively. Table 3 shows the modes of communication used in LTE-D2D and LTE-V2X.

	Scheduled	Autonomous
D2D Communication	Mode 1	Mode 2
V2X Communication	Mode 3	Mode 4

Table 3. Summary of sidelink resource allocation modes

2.1.5.1. Mode 3 Communication

Mode 3 communication works as follows. The eNB periodically broadcasts messages called System Information Blocks (SIB) on the Physical Downlink Shared Channel (PDSCH). SIBs contain useful information for the vehicles, such as procedures to let the UE know how to access the cell or how to do cell (re-)selection. There are different types of SIBs in LTE depending on the purpose, but in the case of LTE-V2X, the eNB broadcasts SIB Type 21, which includes V2X sidelink communication configuration and other information relevant to resource allocation (including parameters and available resources) [5][6]. The UE then requests assignment of dedicated transmission resources for V2X sidelink communication by sending a `SidelinkUEInformation` message to the eNB. The UE may also inform the EUTRAN that it is no longer interested to receive V2X sidelink communication or that it wants to release transmission resources by sending the `SidelinkUEInformation` message.

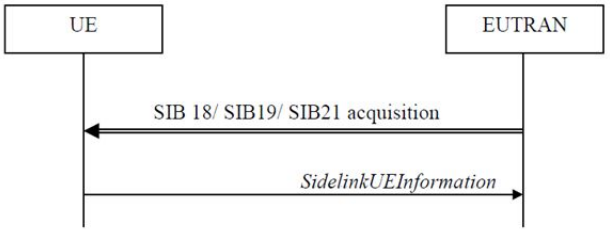

Figure 6. SIB21 and SidelinkUEInformation (Source: 3GPP)

The eNB then sends the Downlink Control Indicator (DCI) message in Format 5A in the Physical Downlink Control Channel (PDCCH). DCI 5A itself contains the fields of the Sidelink Control Indicator (SCI) Format 1 used for the scheduling of PSCCH, namely the the frequency resource location of the initial transmission and retransmission, and the time gap between the initial transmission and retransmission.

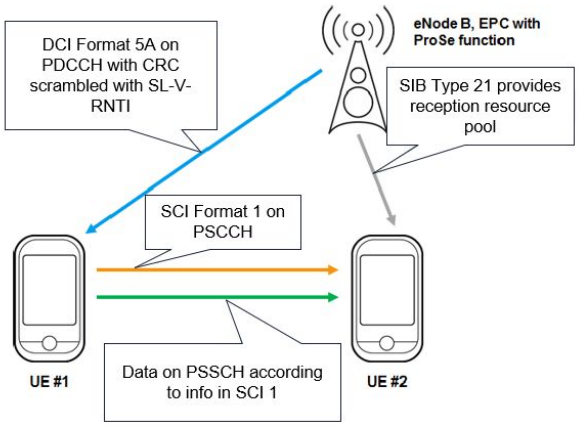

Figure 7. Mode 3 allocation of dedicated resources

2.1.5.2. Mode 4 Communication

In Mode 3, the UE needs to be connected the network, i.e. the UE has to be in the `RRC_CONNECTED` state in order to transmit data. On the other hand, in Mode 4, the UEs select resources autonomously regardless if it is connected to the network or not, i.e. the UE may transmit even when the UEs are in `RRC_IDLE` status, or even when they are out of coverage. If the UE is in-coverage, the network informs the vehicle about the V2X configuration through the Sidelink V2X Configurable parameters [7]. This includes the carrier frequency, resource pool, synchronization references, subchannelization scheme, number of subchannels per subframe, and the number of RBs per subchannel. Otherwise, if the vehicle is out-of-coverage, it uses pre-configured parameters instead.

2.1.6. Technical Features

In order to support high speed and high density vehicular use cases, there are some fundamental modifications to the PC5 interface that have been introduced: a new arrangement for the sidelink or PC5 interface resource grid; a new mechanism for scheduling in the distributed mode, called sensing-based Semi-Persistent Scheduling (SPS); a zone-based resource allocation option, and; additional DeModulation Reference Signal (DMRS) symbols.

2.1.6.1. New Resource Grid Arrangement

For downlink communications, LTE makes use of Orthogonal Frequency-Division Multiple Access (OFDMA), a multi-user variation of OFDM. For uplink communications, LTE makes use of Single-Carrier Frequency-Division Multiple Access (SC-FDMA), a variation of OFDMA with better transmit power efficiency for the benefit of UEs. Because the

physical layers of OFDMA and SC-FDMA both have a time dimension and a frequency dimension, they can be better viewed using a two-dimensional resource grid.

The resource grid is divided in time by radio frames that are 10 ms long. Each radio frame is then divided into 10 subframes of duration 1 ms. Each subframe is further divided into two slots, where each slot can carry 7 OFDM symbols. In frequency, the resource grid is divided into Resource Blocks, which is the smallest resource unit that can be allocated to a user. Each RB has 12 subcarriers of 15 kHz, totalling to a width of 180 kHz.

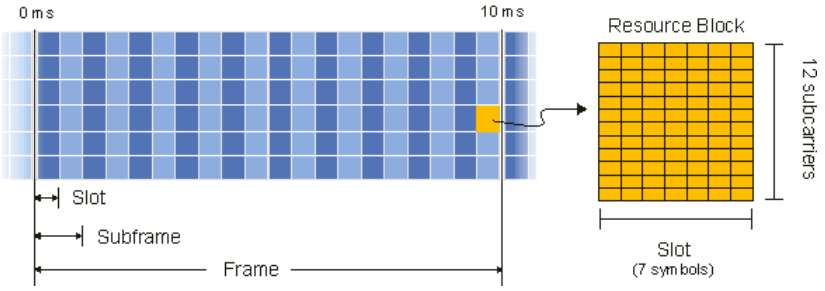

Figure 8. Structure of an LTE frame (Source: Keysight Technologies)

The sidelink, similar to the uplink, uses SC-FDMA, and thus both have almost the exact resource grid structure, except that LTE release 14 introduced a change in the arrangement of the sidelink resource grid. In the sidelink, resource allocation is in *subchannel* granularity rather than in RB granularity. A subchannel is basically a set of contiguous RBs grouped together in the same subframe. Figure 2.7 illustrates how the RBs are grouped together to form subchannels.

V2X data is transmitted in Transport Blocks (TB) through the PSSCH. A TB occupies at least one subchannel, but when the data is large enough, TBs may occupy the succeeding subchannels as long as they are free. Meanwhile, V2X control information is transmitted in the SCI through the PSCCH. An SCI always occupies 2 RBs.

Transmitting data necessitates the transmission of its associated SCI because the SCI is needed to correctly receive and the decode the data. However, in LTE-V2X Release 14, the fact that the PSCCH and the PSSCH are transmitted together in a single subframe means that the PSSCH can be decoded immediately upon recovery of the PSSCH. This enhanced resource grid structure therefore reduces latency.

There are two subchannelization schemes, namely the adjacent SCI-TB (or adjacent PSCCH-PSSCH) scheme and the non-adjacent SCI-TB scheme, as shown in Figure 9. In the adjacent scheme (left), the SCI and TBs are transmitted in contiguous RBs, whereas in the non-adjacent scheme (right), the SCIs and TBs are separated into two resource pools − the upper pool is meant only for SCI transmission while the lower pool is meant only for TB transmission.

A cellular network operator needs to select the value of several design parameters in order to operate a PC5 resource grid, such as the bandwidth, the number of subchannels per subframe, and the number of RBs per subchannel. Conventional LTE supports channel bandwidths of 1.4, 3, 5, 10, 15, and 20 MHz, but LTE-V2X allows only 10 and 20 MHz of channel bandwidth at the 5.9 GHz frequency band. Consequently, a resource grid with a bandwidth of 10 MHz has 50 RBs, while a bandwidth of 20 MHz has 100 RBs. The number of subchannels per subframe is announced through the `numSubchannel` field in

`sl-commResourcePoolV2X-r14` Information Element (IE). `numSubchannel` is only limited to values of {1, 3, 5, 8, 10, 15, 20}. The size of the subchannels in terms of RBs is announced through the `sizeSubchannel`, which is also found in the same IE. `sizeSubchannel` is limited to values of {5, 6, 10, 15, 20, 25, 50, 75, 100} for the adjacent scheme, and {4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 30, 48, 72, 96} for the non-adjacent scheme [6]. The selection of these values may significantly affect the performance of the V2X system based on LTE Release 14, as discussed in study written by the same author of this thesis [33].

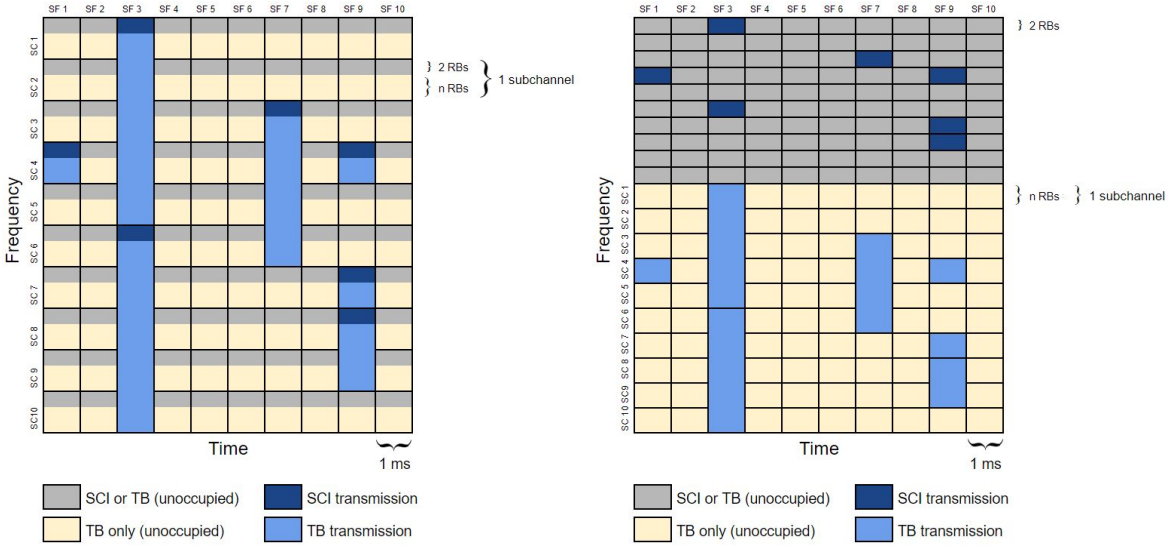

Figure 9. Resource grids with adjacent (left) and non-adjacent (right) subchannelization schemes.

2.1.6.2. Sensing-based Semi-Persistent Scheduling (SPS)

Resource scheduling in LTE may be done as dynamic or semi-persistent scheduling (SPS). In dynamic scheduling, the UE can get scheduling assignments in every subframe, thereby giving the network a large amount of flexibility in resource assignment, especially in varying channel conditions. However, there is additional overhead from control signaling because it is would be necessary transmit resource allocation information on the PDCCH in every subframe. SPS on the other hand allows the same resource to be pre-allocated for a certain amount of time and for a preconfigured periodicity without having to allocate resources every time.

SPS is suitable for services that require persistent radio resource allocations at regular intervals, such as Voice over LTE (VoLTE), which sends one packet every 20ms from the Adaptive Multi-Rate (AMR) speech codec. SPS can therefore support a large number of simultaneous VoLTE users without overwhelming the EUTRAN with control signaling.

As in VoLTE, SPS is also suitable for V2X because a large part of V2X communication traffic consists of Cooperative Awareness Messages (CAMs) that are periodically transmitted by vehicles to provide information of its current status (position, movement, basic attributes, etc.). Nevertheless, LTE-V2X mode 3 permits either dynamic or SPS. Mode 4 uses an enhanced version of SPS called sensing-based SPS.

Sensing-based SPS combined semi-persistent transmission with relative energy-based selection, which theoretically optimizes resource selection by sensing and estimating congestion and choosing the closest to the best resource without no contention overhead. Sensing-based SPS is theoretically better than the CSMA/CA MAC protocol used by IEEE 802.11p, which selects the first available satisfactory resource regardless if it is optimal or not, thereby leading to contention overhead.

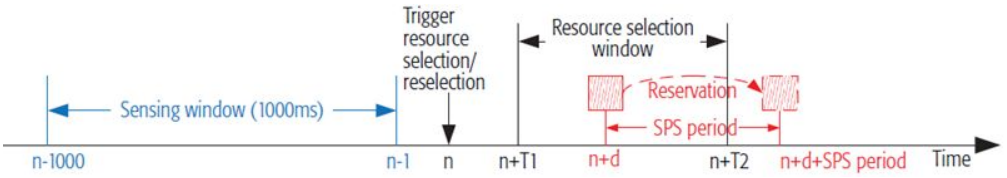

Figure 10. Sensing-based Semi-persistent Scheduling (Source: Chen, et al. [8])

The sensing-based SPS algorithm is summarized in the following steps [9][10]:

1. The *resource sensing window* is established, typically with a length of 1000 subframes (1000ms). The UE continuously monitors the subframes for the duration of the sensing window. The UE checks the decoded SCIs in order to check the remaining transmissions. The UE also checks the sidelink Received Signal Strength Indicator (S-RSSI) measurements.
2. When a resource (re-)selection is triggered at time *n* (mostly due to an incoming packet from the upper layer), all potential resources or *candidate resources* from time *n+T1* until *n+T2* are checked. The time period from *n+T1* to *n+T2* is the *resource selection window*, and the duration *T2-T1* may not be more than 100 ms (in order to meet the maximum LTE-V2X Release 14 latency).
3. A two-step exclusion process is performed over the candidate resources:
    a. If the candidate resource was not monitored, it is discarded.
    b. If the candidate resource is reserved or if the candidate resource S-RSSI is lower than the power threshold, it is discarded.
4. If less than 20% of the total resources in the selection window remain, then the power threshold is increased by 3dB, and the above exclusion process is repeated. Otherwise, one of the candidate resources is randomly selected from those remaining.
5. The selected resource is maintained for a number of *Resource Counter* (RC) message transmissions. RC is randomly selected between:
    - [5, 15] if the resource reservation interval ≥ 100 ms,
    - [10, 30] if the resource reservation interval = 50 ms, or,
    - [25, 75] if the resource reservation interval = 20 ms.
6. When RC goes down to zero, there is a *keep probability P* = [0, 0.8] to maintain the previously selected resources and reset RC. In other words, if *P* = 0, the UE always selects new resources when RC goes down to zero; if *P* = 0.8, there is an 80% chance of keeping the same resources and a 20% chance of reselection when RC goes down to zero.

LTE-V2X offers the option to perform Hybrid Automatic Repeat reQuest (HARQ) retransmissions in order to theoretically increase the reliability of communications. In case

HARQ is to be done, the sensing-based SPS algorithm has to create a second list of candidate resources for the retransmission. Suppose that the reservation of the original transmission occurs on time *n+D*, then the algorithm has to choose randomly from a new set of candidate resources within a selection window of [*n+D*-15ms *n+D*+15ms] for the HARQ retransmission.

It is also worthwhile to note that in the case of pedestrian devices, LTE-V2X allows a *partial* sensing scheme for the purpose of saving battery power.

Instead of using sensing-based SPS, it is possible to do random allocation instead. In this case, a resource is simply randomly selected regardless if it will collide or not. The selected resource is maintained for a certain number of RCs, as in the sensing-based SPS algorithm, and when RC goes down to zero, a new resource is randomly selected.

2.1.6.3. Zone-based Resource Allocation

Zone-based resource allocation is the concept of mapping a certain set of resources into geographical areas or zones. When the vehicle selects resources, it chooses from the resource pool corresponding to the zone where it is currently located [5].

The UE determines the identity of the zone (a.k.a. a zone ID) in which it is located using the following equations:

$$x_1 = floor\,(x / L)\ mod\ N_x$$
$$y_1 = floor\,(x / W)\ mod\ N_y$$
$$Zone\_id = y_1 \times N_x + x_1$$

Where L is the zone length in `zoneConfig` (found in SIB21 in the in-coverage scenario, or SL-V2X-Preconfiguration UE parameter in the out-of-coverage scenario), W is the value of zone width in `zoneConfig`, N_x and N_y are respectively the longitude and latitude parameters in `zoneConfig`, x_1 and y_1 are the distance between the UE current location and geographical coordinates (0, 0) in longitude and latitude, respectively. The UE selects a pool of resources which includes a zone ID equal to the $Zone\_id$ calculated according to above mentioned formulas [6].

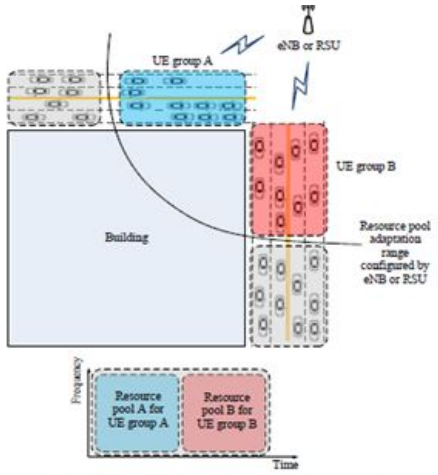

Figure 11. Illustration of zone-based resource allocation in an urban area (Source: 3GPP)

2.1.6.4. Additional DMRS symbols

DMRS is a physical signal in the uplink that is used to estimate the uplink channel impulse response and enable coherent signal demodulation at the eNB. As previously discussed in Section 2.1.6.1 above, a single subframe is composed of two slots of 7 OFDM symbols each. In the uplink, two of those symbols are DMRS symbols, but for LTE-V2X sidelink, additional DMRS symbols have been added in order to handle the high Doppler effects that are associated with relative speeds of up to 500 km/h [9]. Specifically, three DMRS symbols per RB for the PSBCH and four for the PSCCH and the PSSCH have been added. Figure 12 below shows an illustration of the subframe for the PC5 interface that has 4 DMRS symbols, in addition to the Tx-Rx turnaround symbol at the end, meant for better tracking of the channel at high speed [1].

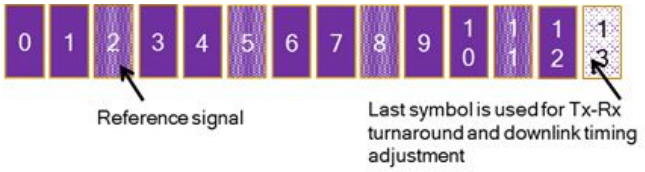

Figure 12. DMRS symbols (highlighted) in PC5 interface (Source: 3GPP)

2.2. V2X in 5G

Even though full 5G V2X specifications are yet to be completed as of the time of writing this thesis, 3GPP has already completed the analysis of new use cases and requirements for V2X in 5G under Release 15 (Section 2.2.1). In addition, several enhancements such as carrier aggregation and higher order modulations for V2X are expected to be introduced in 3GPP Release 15 (Section 2.2.2).

2.2.1. New Use Cases

As discussed in Section 2.1.1, LTE-V2X release 14 supports use cases such as forward collision warning and wrong way-driving warning. 5G V2X takes many steps further by supporting more technologically sophisticated use cases, notably on autonomous driving (platooning, sensor and state map sharing, etc.) and virtual and augmented reality (see-through capabilities from video data sharing, 3D video decomposition, etc.). 5G V2X also supports more use cases that generally enhance safety and convenience (secure software updates, ride sharing, etc.) as well as enhanced support for V2X services in multiple 3GPP RATs (Radio Access Technologies).

Shown below is a table describing some use cases supported by 5G V2X:

Use Case	Description
Vehicle platooning	V2X in 5G will enable platooning, or the operation of a group of vehicles in a closely linked manner such that the vehicles move like a train. Its benefits include reduced distance between vehicles, reduced overall fuel consumption, and reduced number of needed drivers. In order to support platooning, vehicles need to share status information such as speed, heading and intentions such as braking, acceleration, etc.

Sensor and state map sharing	V2X in 5G will provide support for Sensor and State Map Sharing (SSMS), or the sharing of raw or processed sensor data (lidar, cameras, etc) in order to build collective situational awareness. SSMS enables low latency communication for precision positioning and control necessary for applications such as platooning and emergency vehicle communication.
Remote driving	V2X in 5G will also provide support for remote driving. As opposed to autonomous driving, which requires sophisticated sensing and algorithms, remote driving only requires that a human operator or cloud computing control a vehicle remotely. An example of a remote driving scenario could be a bus with an on-board camera that feeds the live video to a remote human operator, who then sends commands to the vehicle based on the video feed without the need of sophisticated computing.
Dynamic ride sharing	V2X in 5G will allow vehicles to advertise willingness to share capacity with other road users and for a pedestrian to indicate intent to travel in a rideshare. The vehicle may share information about itself such as available capacity, destination, and estimated time of arrival, while the pedestrian may share information about itself such as destination, some personal information and credentials, etc.
Secure software updates	An Electronic Control Unit (ECU) is a software module that controls the electronics within a car system, including the automated car driving functions. V2X in 5G will provide a secure mechanism to update the ECU.
Video data sharing	5G V2X will help extend the visual range of the driver, especially in some road traffic situations with obstructions, through video data sharing. Sharing high resolution video data will support drivers in their driving decisions, in accordance with their safety preferences.
3D video composition for the V2X scenario	3D video composition is when multiple UEs with cameras take a video of the environment, and then send this video to a server, which combines the videos to create a single 3D video of the environment. The 3D video can then be used in various ways, such as sharing the video with end-users in a car race, evaluation of possible accident by law enforcement, etc. In order to support this use case, V2X in 5G will provide a mechanism so that a server outside the 3GPP domain is able to time-synchronize different videos received from different UEs, where each UE having a maximum absolute speed of 250 Km/h. In addition, a data rate of 10 Mbps in the uplink per UE (in order to support 4K/UHD video).

Table 4. 5G V2X use cases

For additional use cases and a more in-depth discussion on the V2X use cases for 5G, the reader is asked to refer to the technical report from 3GPP [11].

2.2.2. Envisaged Enhancements

In order to support the use cases for 5G V2X, a transmission of up to 50 pps, with a maximum latency of 3-10 ms and up to a 99.99% reliability level would be required. To achieve this, the following enhancements are currently being under discussed for Release 15 [12]:

Enhancement	Description
Carrier Aggregation	LTE supports Carrier Aggregation of up to 32 carriers. Now, 3GPP is considering the aggregation of up to 8 carriers for LTE-V sidelink as well.
Higher-order modulation	Compared with Release 14, which supports QPSK and 16-QAM, Release 15 adds support for 64-QAM. This increases the data rate and can reduce the channel occupancy. 3GPP is currently analyzing the need for a new DMRS scheme when introducing 64-QAM.
Latency reduction	Release 15 aims to have a maximum latency between 3 and 10ms for V2X. In addition, Release 15 aims to reduce the maximum time between when the packet arrives at layer 1 and the start of the selected sub-channel for transmission from 20ms in Release 14 to less than 10ms.

Shared resources between Modes 3 and 4	As mentioned in Section 3.4.1, both modes 3 and 4 could independently operate using a different pool of RBs. Now, Release 15 is analyzing the possibility of both modes coexisting to optimize the usage of resources. As a consequence, this may require changes for both modes. Some proposals include giving higher priority to Mode 3 reservations (indicated through the SCI), or including the "resource reservation" field in the SCI of Mode 3 transmissions so that Mode 4 can take them into account [13].
Transmit diversity	The feasibility and gains from using transmit diversity schemes such as space time and frequency block coding, as well as small delay cyclic delay diversity are being investigated for Release 15.
Transmission Time Interval (TTI) reduction	The possibility of reducing the TTI for V2X from 1ms to 0.5ms or 2 symbols is being investigated for Release 15. A possible impact of this enhancement would be that Release 14 vehicles with a higher TTI will not be able to overhear Release 15 vehicles using lower TTIs. Also, Release 15 transmissions should not interfere with those from Release 14.

Table 5. 5G V2X possible enhancements

Generally, the enhancements of 5G V2X addresses the technological limitations of LTE-V2X. Unfortunately, a huge downside of the Release 15 enhancements is that it would make it incompatible with LTE Release 14. In fact, some critics even argue that this will effectively render LTE-V2X Release 14 obsolete even before Release 14-compatible devices are deployed [14].

2.3. Review of Related Literature on LTE-V2X Evaluations

Since its 2017 release, there is already a number of publications on the evaluation of LTE-V2X Release 14. This section will briefly discuss the state of the art of the research on the performance of LTE-V2X Release 14.

The paper of [7] presents the first evaluation of the performance of sensing-based SPS under more realistic traffic conditions in a Manhattan grid scenario using OMNeT++, SUMO, and Veins as the simulation platforms. Their results showed that the sensing-based SPS has a slightly better performance than random resource allocation in terms of Packet Delivery Ratio (PDR) over short to medium distances, but the gains of the sensing-based SPS decreases with distances. They also showed that including HARQ functionality virtually negates the gains of sensing-based SPS over random allocation. Finally, they found that collisions produce the most errors at short to medium distance, but at higher distances, propagation errors prevail.

The same authors from above published a subsequent paper [15] that analyzes different configurations of the LTE-V2X Release 14 Mode 4 using the same simulation platform and traffic conditions as above, by varying the keep probability *P*, the sensing window, the power threshold, and the transmit power, and the size of the selection window list and then evaluating the PDR over the Tx-Rx distance. Their conclusions are as follows: increasing *P* reduces the PDR in situations with channel load, although the differences between different values of *P* are not profound regardless of the channel load; changing the sensing window from the standard 1000ms to a non-standard *exponential* window (where more recent sensing measurements are given higher weights in the selection process) produces some improvement, although the gains are unfortunately very little; changing the power threshold has no effect on low channel loads, but a low power threshold improves the PDR in higher channel loads; the transmit power reduces PDR in low loads, but has negligible effect in high channel loads; and finally, the size of the

selection list (which is 20% of the selection window, according to the 3GPP standard) has no significant impact as well.

The researchers in [16] do a performance analysis of LTE-V2X Release 14 Mode 4 using a custom MATLAB-based resource allocation simulator named *LTEV2Vsim* [17], under two urban scenarios (medium density and congested) and a highway scenario (high density). Based on the PRR and the *update delay* (i.e. the difference between the time instant when a message is correctly received and the time instant that the last message was correctly received) they conclude that varying the parameters seemed irrelevant in scenarios with low and medium vehicle load, although they might become significant in congested scenarios. They also concluded that a higher keep probability *P* improves PRR at the cost of higher update delays.

The paper of [18] shows another performance analysis of LTE-V2X Release 14 Mode 4 sensing-based SPS using the ns-3 simulator under a highway scenario with two different UE speeds (UE speeds of 70 and 140 km/h). Their results showed that: the PDR improved when the number of available subchannels increase; reducing the SPS resource reservation interval increases PDR; and the keep probability has insignificant effect due to the high density of the vehicular network. They recommended that more suitable congestion control mechanisms be investigated in order to enhance the sensing-based SPS performance especially in congested networks.

An example of a research that tried to go one step further and propose an enhancement for the standard sensing-based SPS algorithm is [19]. The researchers in emphasized the issue in the Mode 4 sensing-based SPS algorithm where two (or more) vehicles happen to select the same (or overlapping) resource, leading to a continuation of the collisions for multiple messaging intervals. This sustained collision problem could arise when two (or more) vehicles perform resource reselection more or less at the same time and they coincidentally select the same or an overlapping resource. Another possible situation due to the topology changes in the vehicular network, such as for instance, when a new vehicle suddenly joins a group of vehicles that have already coordinated their resource use through sensing-based SPS, and the new vehicle happens to be using the same or an overlapping resource selected by a vehicle in the group. Thus, they proposed to solve the sustained collision problem by letting vehicles share the information about their intended resource reselections *earlier* than their actual reselection instance in order to let other vehicles plan in advance. According to their results, their proposed algorithm performs better compared to the standard sensing-based SPS algorithm due to lower message collision probabilities.

Summing up the findings of the research papers that have been discussed in this section so far, the general consensus seems to be that the standard sensing-based SPS algorithm in LTE-V2X Release 14 Mode 4, as it currently stands, is in need of improvements in order to enhance its performance, thereby compelling some researchers to start investigating on possible enhancements. Many of the other sensing-based SPS parameters generally seem to have negligible effects on the performance, although some of the research present conflicting results, such as [16] claiming that keep probability has a significant impact on the PDR or PRR performance, while the results of [15] and [18] show otherwise. Regardless, most of the current research focus on looking at the performance only from the viewpoint of the PDR or the PRR, so by looking not only at the

PRR but at alternative perspectives such as resource grid occupancy, goodput, NAR, and distance error as well, this thesis might be able to contribute in the better understanding of the different aspects of the performance of LTE-V2X Release 14 Mode 4. Moreover, this thesis also looks at other parameters that have not yet been thoroughly investigated in the current research, such as the message size and the collision threshold.

3. Methodology and Project Development

This chapter provides basic information about the simulation frameworks used in this project (Section 3.1) and discusses the development and configuration of the LTE-V2X Release 14 Mode 4 simulator (Section 3.2).

3.1. Project Simulator Framework

The LTE-V2X Release 14 Mode 4 simulator was developed by making use of the following software frameworks and simulators: OMNeT++, INET framework, SUMO, and Veins. Figure 3.1 depicts the relationship between the frameworks. The LTE-V2X Release 14 Mode 4 simulator is based on modified INET components (which lies on the same hierarchy as Veins). INET and Veins are built on top of the OMNeT++ network simulation framework, while Veins obtains road traffic simulation data from SUMO. The following sections explain the simulation frameworks in more detail, as well as why these were chosen for the project.

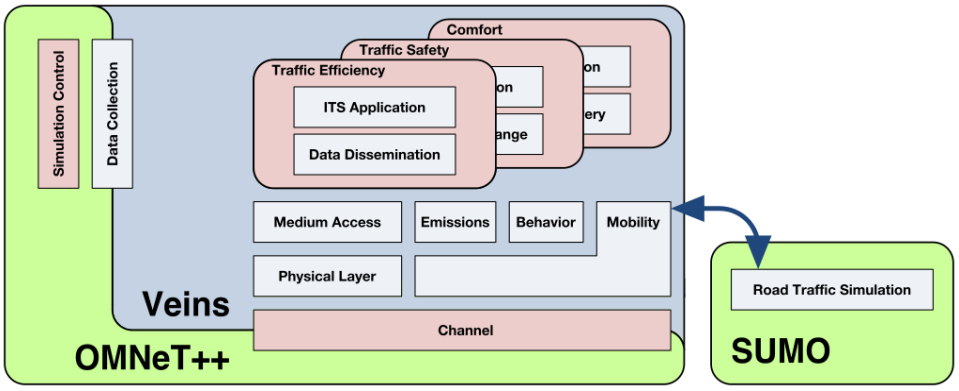

Figure 13. Project simulator framework (Source: Veins)

3.1.1. OMNeT++

OMNeT++ is a C++-based extensible, modular, component-based simulation library and framework. OMNeT++ is not a network simulator itself, but rather, it is used to build network simulations such as wired and wireless communication networks, queuing networks, etc. [20]. OMNeT++ is public-source and can be used under the Academic Public License, thus it is free for non-profit use. It is based on the Eclipse Integrated Development Environment (IDE), and it can be installed on common platforms including Linux, Mac OS/X and Microsoft Windows [21].

OMNeT++ modules may either be simple or compound, and they communicate with each other by message passing through a connection between gates, i.e. the input and output interfaces of the modules. Messages can have any arbitrary data, and can be defined by specifying their contents in a `*.msg` file. Connections can be assigned with properties such as propagation delay, data rate, and bit error rate.

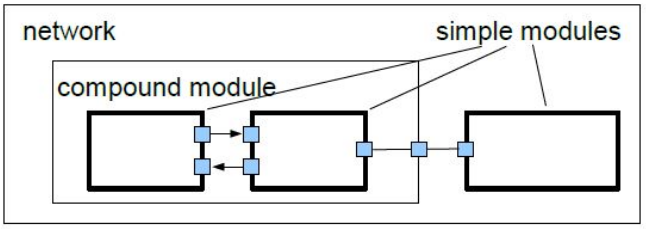

Figure 14. Module structure in OMNeT++ [21]

Module behavior is defined by C++ classes. Modules may have parameters, which can take string, numeric, or boolean values. Configuration files or `*.ini` files enable the easy modification of module parameters, as well as other more general simulation-related parameters Module topology, i.e. its structure and their interconnections, is defined in `*.ned` files and is written using the high-level NEtwork Description (NED) language. Simple module declarations in `*.ned` files include the gates and parameters of the module; compound modules declarations include the gates and parameters, as well as the definition of internal submodules and their interconnections; finally, network declarations include compound modules that are self-contained and do not have any external connections [21].

OMNeT++ supports *Qtenv* and *Tkenv*, which are graphical runtime environments for running simulations. Both environments support interactive simulation execution, animation, inspection, tracing, as well as debugging. OMNeT++ also supports *Cmdenv*, a command-line environment for running simulations. Cmdenv is able to perform simulations much faster than Qtenv and Tkenv not only due to less overhead from having no graphical output, but also because batch execution and progress feedback during simulation are only supported when using Cmdenv [22].

OMNeT++ has several prebuilt statistical classes that allow the collection of scalar statistics and vector statistics. OMNeT++ stores simulation results in `*.sca` files for scalar statistics and `*.vec` files for vector statistics, which can then be viewed through analysis files or `*.anf` files. OMNeT++ has the ability to show visual representations of these statistics through line charts and bar charts, but modules may be programmed to output custom statistic files such as Comma-Separated Value (CSV) files that can then be analyzed using external tools such as spreadsheet software or MATLAB.

For this project, OMNeT++ was chosen as the simulation platform of choice because it is free, relatively easy to build upon, and it is well-known in academic circles for network simulations. In fact, OMNeT++ was also the simulation platform for some of the related research in LTE-V2X Release 14 Mode 4 evaluation in combination with SUMO and Veins [7][15], although the modules used in this project were developed independently. The version of OMNeT++ used in this project was version 5.1.1.

3.1.2. INET Framework

The INET framework is an OMNeT++-based open-source model library providing a wide range of models for communication networks. INET provides ready-to-use models for the Internet stack (IPv4, IPv6, TCP, UDP, OSPF, BGP, etc.), link layer (Ethernet, IEEE 802.11, PPP, etc.), and the physical layer (propagation model, obstacles,

frame/bit/symbol level representation, etc.). Regardless, as it is open source, new components may be programmed or existing components may be modified by the user. And since it is based in OMNeT++, it utilizes the same concepts described in the previous section, such as simple and compound module composition, message passing, etc. [23].

The version of INET used in this project was 3.6.0.

3.1.2.1. Radio Transmitter and Receiver Models

For the physical layer, INET provides a radio model that relies on several submodels, namely the antenna, transmitter, receiver, and error models. There are some readily-available generic radio models such as the Unit Disk Radio model, which provides a very simple and fast physical layer model, and the Amplitude Phase Shift Keying (APSK) radio model, which simulates modulations such as Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM) modulations. INET also provides application-specific radio models such as an IEEE 802.11 and a IEEE 802.15.4 radio model.

Antenna models describe how the electrical signals are converted into radio waves and vice-versa. Antenna models are shared between the transmitter model and the receiver model. Readily-available antenna models in INET include isotropic, constant gain, dipole, and interpolating antenna models.

Transmitter models describe how the packets are converted into electrical signals. Transmitter models may either be flat or layered. Flat transmitter models exclude the symbol and the sample domain representations, while the layered model includes more detailed processing steps such as packet serialization, forward error correction encoding, modulation, etc. As such, the layered model is much more computationally intensive.

Receiver models may also be flat or layered. Receiver models check if the reception is *possible* (based on received power, bandwidth, modulation, etc.) and if the reception is *attempted* (synchronization was achieved). If these two conditions are satisfied, the receiver will send the packet to the MAC layer. However, the receiver performs another check if the reception is *successful* (based on the error model). If this final condition is met, then the receiver passes the packet to the MAC layer with a tag indicating a successful reception; otherwise the receiver passes the packet with a tag indicating an unsuccessful reception.

The receiver model also has an error model that determines whether the received packet has errors or not based on the SNIR [23].

3.1.2.2. Radio Medium Models

The INET radio medium models describe the shared physical medium where wireless communication takes place. It makes use of several submodels to perform its tasks, such as the signal propagation, path loss, obstacle loss, background noise, and signal analog model.

The signal propagation model describes how the signal is transmitted through the medium. INET provides a constant time propagation model, which is a simple model where the propagation time is independent of the distance, and a constant speed

propagation model, which is a relatively more realistic model where the propagation time is proportional to the distance travelled.

The path loss model describes how the power density goes down as the signal propagates. INET has built-in free space, log normal, two-ray, Rayleigh, Nakagami, and other popular models, each of which having its own set of parameters.

The obstacle loss model describes how the signal power is lost as it interacts with physical objects. INET has an ideal obstacle loss model, which decides complete or no power loss depending on the presence of a physical object, and dielectric obstacle loss model, which computes power loss based on dielectric and reflection loss taking into account the physical properties and shape of the object.

The background noise model describes the effects of different sources of noise, such as thermal noise and cosmic noise. INET provides isotropic scalar background noise model, which is a simple noise model for computing noise independent of space and time [23].

3.1.2.3. Medium Access Control Models

INET offers a variety of MAC models depending on the application. For instance, the IEEE 802.11 MAC model works in accordance with the IEEE 802.11 MAC procedures, including acknowledgement (ACK), Request to Send/Clear to Send (RTS/CTS), fragmentation, and MAC Service Data Unit (MSDU) policies to name a few. It is necessary to use the provided IEEE 802.11 physical layer model as well in order to properly use the IEEE 802.11 MAC model. For less application-specific simulations, INET does offer a generic ideal MAC model that performs simple tasks such as checking if the received packet has the correct MAC address, or dropping erroneous frames [23].

3.1.2.4. Application Layer Models

INET has simple application models that can generate message traffic based in TCP or UDP. For TCP, INET provides built in basic, echo, and even Telnet application models, while for UDP, INET provides basic, burst, echo, video stream and other application models. The basic UDP application is the most straightforward model, as it models an application that just sends UDP packets to a given IP address every given interval [23].

3.1.3. SUMO

SUMO is an open-source road traffic simulation package. It enables the creation and modification of road networks through a visual editor called NETEDIT [24]. SUMO traffic flow is simulated microscopically, i.e. each vehicle is modelled individually and has a certain speed and position, which are updated every time step. Simulations are multimodal, meaning SUMO may also have other modes of transportation such as public transportation systems and even pedestrians in addition to cars. Simulations are also deterministic, with the option of adding randomness to the simulation.

The version of SUMO used in this project is 0.30.0.

3.1.4. Veins

Veins is an open-source framework based on OMNeT++ (and INET to a certain extent) and works with SUMO (see Section 3.1.4) to perform vehicular network simulations. While OMNeT++ takes care of the network simulation part and SUMO takes care of the road traffic simulation part, Veins is in charge of coordinating these two and sets up, runs, and monitors the vehicular network simulation. The interaction between Veins and SUMO is managed through a TCP socket and through the Traffic Control Interface (TraCI) protocol [25].

The version of Veins used in this project is 4.6.

3.1.5. Alternative Software

The following subsections introduce some software or simulation platforms that were considered for use in this thesis project, but were ultimately decided not to be used due to reasons discussed below.

3.1.5.1. SimuLTE

SimuLTE is an open-source system-level simulator for LTE (Release 8 onwards) networks. It is based on OMNeT++, INET, and Veins, and it works with SUMO as well. SimuLTE aims to be widely-used simulation framework for LTE for the purpose of creating a common simulation framework that is easily verifiable by the academia, instead of creating single-purpose homebrew simulators that are never released to the public [26].

SimuLTE has support for D2D communication, both one-to-one and one-to-many. However, SimuLTE remains in beta status, with many features still unimplemented, some statistics unavailable, and several bugs unresolved. Its current D2D implementation is also limited in that it only models mode 1 communication, which is always supervised by the eNB. It does not model the PC5 resource grid, and it lacks other LTE-V2X release 14 functionalities. Moreover, SimuLTE seems to have halted, with no further updates since March 2018. Due to the aforementioned problems, it was decided that developing on top of SimuLTE was not worth the effort.

3.1.5.2. MATLAB-based Simulators

MATLAB provides *LTE Toolbox*, which enables the simulation of the physical layer of LTE-V2X sidelink as well [27]. With the functionalities provided by the toolbox, it is possible to have a model of the PC5 resource grid down to the waveform level. In addition, with the *TraCI4Matlab* Application Programming Interface (API), it is possible to interact with SUMO in order to perform vehicular simulations [28]. Nonetheless, the toolbox is not well-suited for larger-scale simulations, as it focuses on end-to-end communication links only.

There are a few other MATLAB-based simulators built by the academia for simulating LTE-V2X, such as *LTEV2Vsim* [17], which is freely available to interested researchers,

but unfortunately does not have an active development community. Moreover, it was developed with a singular purpose of investigating resource allocation for cooperative awareness, and it does not attempt to model the 3GPP standards. Another publication described the simulation of the distributed resource allocation for integrating LTE-V2V and Visible Light Communication (VLC) [29]. However, the simulator in this paper is neither open nor applicable for other purposes.

3.1.5.3. NS3-based simulators

LTE-EPC Network simulAtor (LENA) is an open-source LTE simulator based on the ns-3 network simulator. In terms of maturity, LENA appears to be one of the most well-developed open-source simulators for LTE systems with a wide array of features and thorough documentation, and is still in active development. Similar to SimuLTE, LENA does not attempt to simulate down to the symbol level in order to reduce computational complexity. LENA, however, has an advantage of being able to represent the resource grid with the physical channels, unlike SimuLTE, which models channels through messages. That said, there are no available functionalities available for LTE-V2X [30].

3.1.5.4. Other OMNeT++-based simulators

Artery-LTE is a simulation framework combining Artery and VeinsLTE. It implements the ETSI ITS-G5 and the LTE protocol stacks, thereby enabling the analysis of different communication strategies on heterogeneous vehicular networks. However, Artery-LTE is no longer in active development, with its last version released on April 2016 [31].

A notable example of a simulator developed for studying the performance of Release 14 LTE-V2X is the one developed in 2017 by the researchers in reference [7]. They made use of MiXiM, an OMNeT++ framework for mobile and fixed wireless networks, to implement the MAC and the physical layers, including the resource pool divided into subchannels, sensing-based semi-persistency algorithm, separate SCI and RB transmissions, transmission error calculation by lookup tables, etc. They did not attempt to model the interaction between the eNB, instead, they simply added a global module in OMNeT++ that emulates the eNB tasks such as providing the preconfigured transmission parameters. This simulator however, is not open to the public.

3.2. Development of the LTE-V2X Release 14 Mode 4 Simulator

This section provides an explanation of the architecture, functions, and configuration of the LTE-V2X Release 14 Mode 4 Simulator that was developed for this thesis project. The top-level network (Section 3.2.1), the physical layer (Section 3.2.2), the MAC layer (Section 3.2.3), and the application layer (Section 3.2.4) models, as well as the road and traffic models (Section 3.2.5) are discussed.

3.2.1. Vehicular Network Model

At the top of the simulation module hierarchy is the so-called `Highway` network module, which, although named as such, may run both highway and grid road topologies (see Section 3.2.5). The `Highway` module is described in the file `Highway.ned` and it consists of the `radioMedium`, `physicalEnvironment`, the `manager`, the `systemMonitor`, and the `MyCar` modules.

The `radioMedium` comprises the modules that model the radio medium such as the signal propagation, path loss and background noise. For the simulations, the signal propagation was set to have a constant time propagation since propagation time effects were unimportant for the purpose of this project. The path loss model used was the Rayleigh fading model because it is commonly used to represent densely urbanized environments without LOS, whereas an alternative Rician fading model is more appropriate in suburban areas with LOS propagation. Since the highway and grid scenarios in this project are imagined to be in urban locations, it was decided that the Rayleigh fading model is appropriate. It should be noted that many of the related research works [7][15][18] typically make use of the so-called *WINNER+ B1* model for the path loss, however it was not readily available in INET. The Rayleigh fading model has a shape parameter *alpha* that has typical values of 2.7 to 3.5 for urban areas, thus an alpha value of 3.0 was arbitrarily chosen to represent the urban scenarios. The background noise was set to be of the isotropic scalar type with a power of -110dBm.

The `physicalEnvironment` comprises the modules that model the buildings, walls, terrain, weather, and other physical objects and conditions that affect on radio signal propagation. In the Manhattan grid scenario, it is possible to configure the physical environment to have walls that represent buildings and other obstacles.

The `manager` module handles Veins-related tasks. The `firstStepAt` parameter of the manager module indicates what time the simulator should connect to the TraCI server of SUMO in seconds. For the grid scenarios, `firstStepAt` was set to 300 in order to allow SUMO sufficient time to fill the entire grid with vehicles. For the highway scenarios, `firstStepAt` was set to the default value of -1 since SUMO is able to fill the highway up with vehicles much more quickly by using a random depart position and small vehicle insertion period such as 0.01s (Section 3.2.5). The `updateInterval` parameter, which indicates the time interval to update the positions of the vehicles, was set to 0.01s.

The `SystemMonitor` module is a custom-made module that was developed in order to manage statistics gathering and random number generation. The car modules pass some of their statistics information to the `SystemMonitor` module. `SystemMonitor` saves the gathered statistics as either OMNeT++ scalar and vector files, or as CSVs.

The `MyCar` modules represent the vehicles and its communication components. As shown in Figure 15, every vehicle module consists of components such as the UDP application module (or `udpApp`), the mobility module, the Network Interface Card (NIC) or the `wlan` module, and so on. The component that was most heavily modified for this project was the NIC, and its physical and MAC layer features are discussed in the following sections.

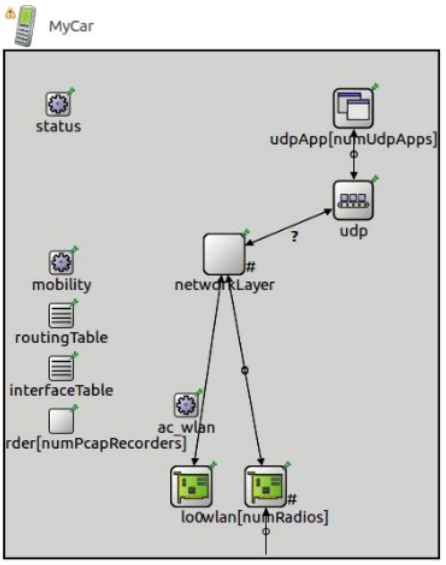

Figure 15. The `MyCar` module and its components

3.2.2. Physical Layer Model and Parameters

A major limitation of INET is that it does not have any available module that has capabilities to simulate SC-FDMA communications in the physical layer. Since developing an entirely new physical layer model would be too arduous to fit within the timeline of this project, and since the physical layer performance of LTE-V2X Release 14 is not the primary focus of this project, it was decided that the INET APSK model be adopted for the physical layer. Also, due to the complexity of the physical layer, the resource grid and the sensing functionalities of the sensing-based SPS algorithm was modelled entirely in the MAC layer instead of the physical layer. This necessitated a minor modification to the PHY so that the PHY layer included the S-RSSI information every time it a packet to the MAC layer.

Even though developing a highly accurate PHY layer model was not a priority for the project, some workarounds still had to be done so that some SC-FDMA capabilities are somehow emulated. Transmissions were made to be completed near-instantaneously, and transmission times were slightly staggered in time in order to reduce interference between nodes. All of the transmissions that were successfully received within the duration of a single TTI (1ms) are passed to the MAC layer, where they are temporarily stored in a buffer for processing.

The transmitter was configured to transmit power at 200 mW, while the receiver was configured to have a sensitivity of -95 dBm, an energy detection of -95 dBm, and an SNIR threshold of 10 dB.

3.2.3. MAC Layer Model and Parameters

In order to model the LTE-V2X Mode 4 MAC layer, the basic *ideal* INET MAC module was adopted and heavily modified to include the resource grid as perceived by the node, sensing-based SPS functionalities, and an application layer-level memory data structure. Some optional capabilities of the Mode 4 communications and sensing-based SPS such

as HARQ and zone-based resource grid separation were considered to be out of the scope of this project, and were excluded from development.

The resource grid was modelled only up to the subchannel level, thus resource blocks and symbols such as the DMRS symbols were not modelled. Thus, in configuring the resource grid, it is only necessary to specify the number of subchannels per subframe. For the simulations, the resource grid was set to have 10 subchannels per subframe.

As prescribed by the sensing-based SPS algorithm, the MAC layer constantly monitors the state of the resource grid. When a node wishes to transmit without a prior reservation, the MAC layer first looks at the past 1000 subframes of the resource grid, checks all the candidate resources within the duration of a selection window, filters the candidates based on the two-step exclusion process, until it finally selects and reserves a resource taken randomly from the top 20% of the choices for a number of transmissions equal to the randomly-chosen RC (refer to Section 2.1.6.2 for a more detailed description of the sensing-based SPS algorithm). For the simulations, the candidate resource ratio is retained at 20%; the selection window is set to 100 subframes; the maximum and minimum RC values are 15 and 5 respectively; and the power threshold is -95dBm. The sensing-based SPS keep probabilities are varied between 0, 0.8 and 1.0.

Prior to passing the frame to the PHY, the MAC layer of a transmitting node attaches metadata to the frame. The metadata contains information that is supposed to be found in the SCI, such as the location of the resource that the vehicle intends to reserve, and for how many more message intervals the vehicle intends to reserve the same resource. The metadata also includes the current location of the vehicle upon generating the MAC layer frame.

A temporary buffer at the MAC layer of a receiving node collects all of the successfully received packets from the PHY layer that have been transmitted within the duration of a single TTI. At the instant that the next TTI begins, all of the packets in the buffer are checked to see if any of those have colliding resource reservations. If there are any collisions, then the packets are compared against the combined S-RSSI values of the other colliding packets. If the packet has an S-RSSI value that is greater than or equal to the combined S-RSSI values of the other packets plus the collision threshold value, then the packet is considered to be a good and the others are considered to be corrupt; otherwise, all packets are considered to be corrupt. For the simulations, the collision thresholds are varied between 3 dB and 20 dB.

After going through the buffer, good packets are passed on to the upper layers, while packets marked as corrupt are dropped by the MAC layer. The appropriate subchannels of the resource grid are marked as corrupt or good depending on the state of the packets that intended to reserve those subchannels. The metadata of the good packets are then added to the resource grid.

Whenever a good packet is successfully received, an application layer-level memory data structure also temporarily records information about the transmitting vehicle, including its position at the time it transmitted the packet. This information is used to determine the distance error. However, if the receiver did not receive any updates from a transmitting node for a certain time, it removes the memory entry related to that particular transmitter. For the simulation, the time limit of the memory before entries are removed is set to the duration of 20 transmissions.

3.2.4. Application Layer Model and Parameters

A basic UDP application model was used for the LTE-V2X Release 14 Mode 4 simulator. The basic UDP application simply transmits packets at every interval representing CAMs, starting from a specified point in time. ETSI specifies that the minimum and maximum time intervals between consecutive CAMs must be 100ms and 1000ms, respectively. For all simulations this project, a transmission interval of 100ms was chosen in order to have an increased message traffic. The starting time of transmission was set to be randomly chosen from a uniform distribution between 1 and 2 seconds from the creation of the car module in the simulation.

The destination addresses for the UDP messages are sent to `224.0.0.1`, which is an IPv4 address designated for multicasting on the local subnetwork. The UDP applications are also set to join the local multicast groups.

3.2.5. Road and Traffic Models and Parameters

The default road topology used in this project was a straight bidirectional highway. In order to provide a few comparisons with the default case, two other road topologies were also used in this project, namely a unidirectional highway and a Manhattan grid.

3.2.5.1. Highway Scenarios

The length of the straight highway was set to 1000 m, where each lane has a width of 3.2 m.

A *statistical region* was defined so that statistics from vehicles outside the region were not recorded. The aim of defining the statistical region was to eliminate border effects caused by having fewer less vehicles at either end of the highway scenario compared to its central region. For the highway scenarios, the statistical region was set to the central 400 m (i.e., 300 m < x < 700 m), as depicted in Figure 16.

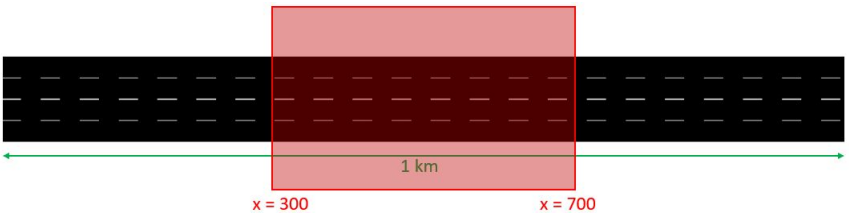

Figure 16. Statistical region in the 1km highway scenario

For the simulations, three types of highway sizes were created by varying the number of lanes, namely 4, 8, or 16 lanes. In the default highway topology, which is bidirectional, this means that the 4, 8, and 16 lane scenarios have 2, 4, or 8 lanes in each direction, respectively. Figure 17 shows the bidirectional highway having each of the different lane size scenarios.

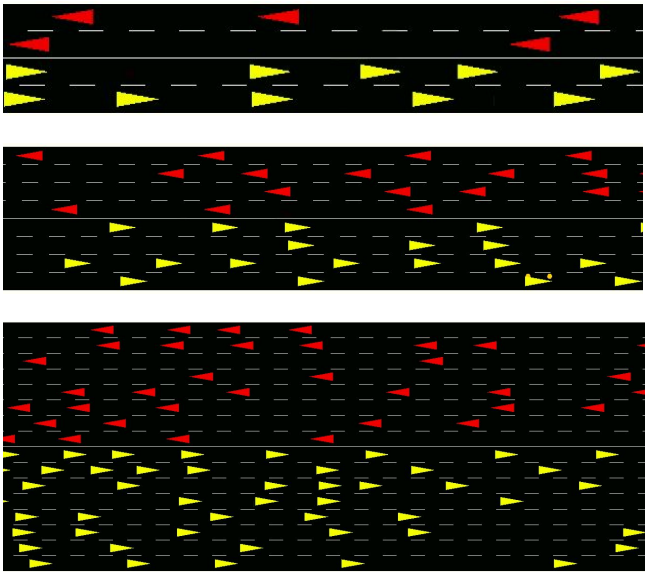

Figure 17. Bidirectional highway with 4 (top), 8 (middle), and 16 (bottom) lanes

3.2.5.2. Manhattan Grid Scenarios

The Manhattan grid scenario was set to a size of 445 m × 445 m, with a statistical region bounded at 30 m < *x* < 375 m and 111.25 m < *y* < 333.75 m, where *x* and *y* are the horizontal and vertical axes, respectively. The statistical area dimensions represents an approximation of the central region of the grid.

A simplistic model of buildings, walls, and other physical obstacles was also included in order to have a more realistic model of the propagation effects in such a dense urban environment.

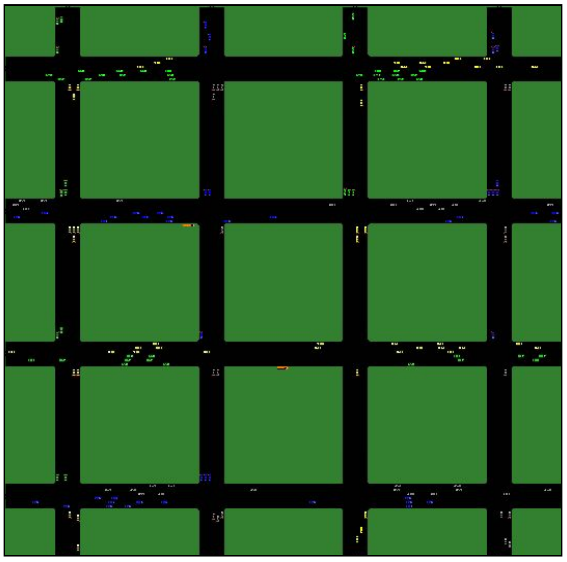

Figure 18. Manhattan grid scenario

3.2.5.3. Traffic Characteristics

The vehicular simulations used the default *Krauss* car following model, a collision-free model that allows cars to drive up to the maximum safe speeds while maintaining sufficient braking distance with each other. This means that a scenario where all vehicles run fast have more separation from each other compared to a scenario where all vehicles run slow.

For the simulations, a fast and slow vehicle scenarios where vehicles run with a maximum speed of 33.33 m/s and 3 m/s respectively were defined. Figure 19 shows a snapshot of the bidirectional 4-lane highway under both vehicle speed scenarios. Notice that the slow scenario is remarkably denser due to smaller vehicle separations. In some parts, the gaps between the cars in the slow scenario are even less than the car length, of 5 m. Regardless of the vehicle speed, the vehicle gap was configured so that it may never go lower than 2 m.

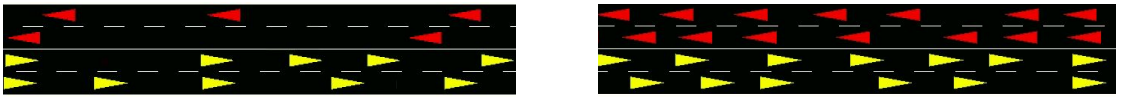

Figure 19. Bidirectional 4-lane highway scenario with fast (left) and slow (right) vehicle speeds

The approximate number of vehicles/km in the highway for the 4, 8, and 16-lane scenarios are shown in Table 6.

Scenario	Maximum Vehicle Speed [m/s]	Approximate number of vehicles per km		
		4-lane highway	8-lane highway	16-lane highway
Fast	33.33	163	321	642
Slow	3.00	351	700	1382

Table 6. Approximate number of vehicles per km per scenario

The departure position of the highway scenarios was set to random, which means that vehicles enter the highway scenario from any position. This is in contrast to having all the vehicles enter the scenario from the same entry point, such as the leftmost side of the highway. The reason for setting the departure position to random is because SUMO creates groups of vehicles at the same time, and if this group of vehicles enter the scenario from the same entry point, and move at more or less the same speed, they tend to stick together as a bunch. The bunching effect is alleviated in with random departure position. The vehicle insertion period was set to 1 ms so that the entire highway is populated immediately with vehicles. Regardless, a warmup time wherein no statistics are collected was set to 5 s in order to let the highway reach a steady state of traffic flow first. Even if the departure position was set to random, SUMO is no longer able to insert new vehicles into any position at the steady state because at that time, all the vehicles are already moving at high speeds while maintaining safe distances, thus simply inserting

new cars randomly will violate the car following model. Consequently, at steady state, all the vehicles now enter from the end of the highway.

For the Manhattan grid scenario, it was not as simple as the highway scenarios to reach the steady state of traffic because vehicles enter the scenario only on or near the ends of the roads regardless if the departure position was set to random. This may be due to the fact that the Manhattan grid is more complicated and needs to obey traffic light rules, avoid collisions at intersections, and so on. Thus, instead of tinkering with the departure position, SUMO was just left to run the road traffic simulation on its own for 300 seconds until it reaches more or less a steady traffic state, after which the Veins manager module connects to SUMO. Statistics are then gathered from a simulation time of 305 seconds onwards.

3.3. Summary of Parameters and Models Used

Table 7 summarizes the simulation parameters and models used in the simulations.

Category	Parameter	Value
Medium	**Propagation type**	Constant time
	Path loss type	Rayleigh fading
	Alpha	3.0
	Background noise power	-110dBm
	Background noise type	Isotropic scalar
Application	**UDP transmission interval**	0.1s
	Destination IP address	224.0.0.1 (multicast)
	Transmission start time	uniform distribution between 1.0s, 2.0s from node creation.
Radio	**Carrier frequency**	5.9GHz
	Radio type	APSK Scalar
	Modulation	BPSK
	Transmitter power	200mW
	Receiver sensitivity	-95dBm
	Energy detection	-95dBm
	SNIR Threshold	10dB
MAC (Sensing-based SPS)	**Resource allocation method**	{sensing-based SPS, random allocation}
	Candidate Ratio	0.2
	Selection window	100 subframes
	Maximum RC value	15

	Minimum RC value	5
	Memory limit	20 transmissions
	Power threshold	-95dBm
	Collision threshold	{3dB, 20dB}
	Resource grid subframes	1000
	Keep probability	{0.0, 0.8, 1.0}
	Subchannels per subframe	10
	Subchannels per message	{3, 5, 10}
Environment	**Scenarios**	{unidirectional highway, bidirectional highway, Manhattan Grid*}
	Maximum vehicle speeds	{fast (33.33m/s), slow (3.00m/s)}
	Mobility model	Krauss car following model
Others	**Simulation time limit**	35 seconds (highway), 360 seconds (Manhattan Grid)
	Warmup time	5 seconds (highway), 305 seconds (Manhattan Grid)
	Manager first step time	-1 (highway), 300 seconds (Manhattan Grid)

Table 7. Summary of parameters and models

4. Results and Analysis

This chapter presents and discusses the results of running simulations using the models and configurations previously covered in Chapter 3. The results are evaluated from the perspective of the MAC layer and the application layer through certain performance metrics. The MAC-level metrics used were the Packet Reception Ratio (PRR) (Section 4.1) and the resource grid occupancy and goodput (Section 4.2). The application layer-level metrics used were the Neighborhood Awareness Ratio (NAR) (Section 4.3), the position error (Section 4.4), and latency (Section 4.5).

For the purpose of simplifying the discussion of the results, the *default configuration* is hereafter defined as the 4-lane bidirectional highway scenario running fast-moving vehicles (33.33 m/s maximum) that transmit messages occupying 3 subchannels, utilize sensing-based SPS with a keep probability *P*=0 and a collision threshold of 3 dB.

4.1. Packet Reception Ratio (PRR)

PRR is a commonly used performance indicator in communication networks. In fact, it is also the main measure in a majority of the related research in LTE-V2X Release 14 Mode 4 [7][15][16][18][17][19]. For the simulations, the PRR is calculated as follows:

$$PRR = \frac{correctly\ received\ packets}{total\ received\ packets\ above\ SNIR\ threshold} \times 100$$

A packet is considered to be correctly received if it *successfully* arrived at the MAC layer (Section 3.1.2.1), and has exceeded the collision power threshold in the case that a collision occurred (Section 3.2.3). The PRR results in this paper are represented as functions of the distances between the actual positions of the transmitter and the receiver.

The effects of varying the vehicle traffic load (Section 4.1.1), message size (Section 4.1.2), and collision power threshold (Section 4.1.3) on the PRR when using sensing-based SPS are henceforth discussed. Comparisons between the PRR performances of sensing-based SPS and random allocation (Section 4.1.4) are also presented.

4.1.1. Impact of Traffic Load

As discussed in Section 3.2.5, the vehicle load of the scenario is varied by changing the number of lanes of the highway or by changing the maximum vehicle speeds. Shown on Figure 20 is the effect on the PRR of changing the vehicle speeds (fast or slow) along with the number of lanes (4, 8, or 16) of the default configuration. Recall that the average densities of vehicles in the highway when the vehicle speed is fast are approximately 163, 321, and 642 vehicles/km for the 4, 8, and 16-lane scenarios respectively; in contrast, the average densities when the vehicle speed is slow are approximately 351, 700, and 1382 vehicles/km (Table 6).

Looking first on the impact of the number of lanes (blue vs. red vs. yellow lines), it is unsurprising that the PRR becomes worse with more lanes due to the increasing interference and collisions that comes with having more transmitting nodes within the range of each vehicle. For example, looking at the PRRs of the fast scenarios at 100m, the 4, 8, and 16-lane scenarios have values of about 75%, 64%, and 43% respectively.

Now focusing on the impact of the vehicle speeds (solid vs. dashed lines), since going from the fast vehicle scenario to the slow vehicle scenario causes the traffic density of vehicles to go slightly more than double, the number of potentially interfering nodes also doubles, thereby decreasing the PRR as well. However, notice that the 8-lane fast vehicle scenario has a slightly better PRR because it has actually less average number of vehicles (321 vehicles/km) compared to the 4-lane slow vehicle scenario (351 vehicles/km). The same reason can be used to explain why the 16-lane fast scenario (with 642 vehicles/km) has a slightly better PRR than the 8-lane slow scenario (with 700 vehicles/km).

Regardless of the scenario (4, 8, or 16 lanes; fast or slow), the PRR always tends to approach zero from 400m onwards because the signal is already too weak to cope for the effects of additional interferences at those distances. This behavior was observed in other configurations as well (varying message size, collision power threshold, etc.).

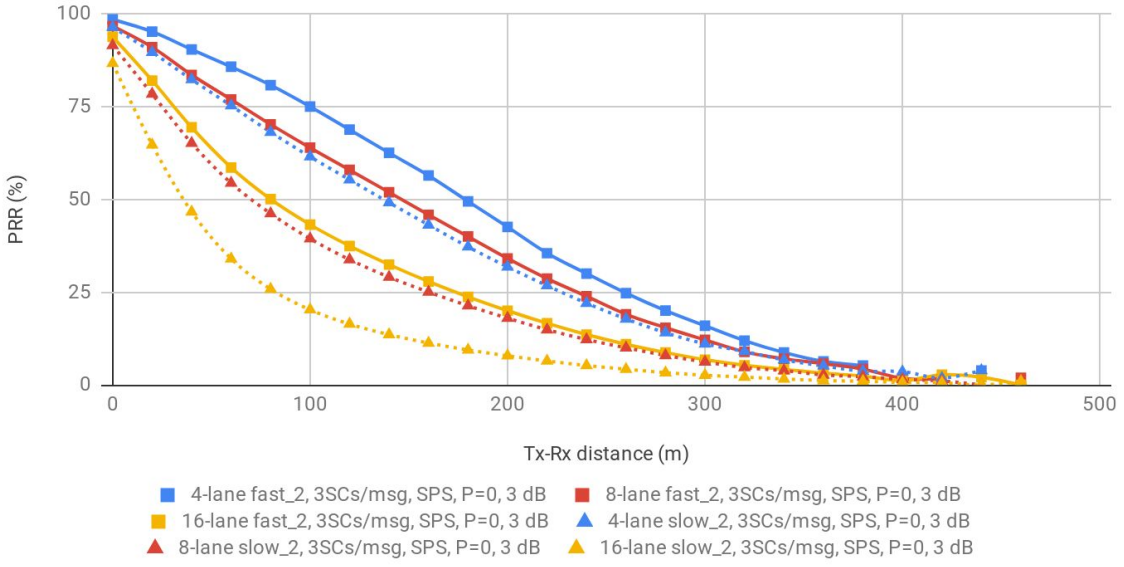

Figure 20. PRR when number of lanes and car speed are varied (default configuration)

4.1.2. Impact of Message size

Now, the effect on the PRR of the message size when the messages occupy 3, 5 and 10 subchannels each is investigated. Recall that for all scenarios, the PC5 resource grid was set to have 10 subchannels available per subframe, thus a message occupying 10SCs/msg takes up the entire subframe.

Shown on Figure 21 is the effect on the PRR of changing the message sizes (3, 5, or 10 SCs/msg) along with the numbers of lanes (4, 8, or 16) of the default configuration.

The scenarios where all vehicles transmit a smaller message size of 3 SCs/msg (solid lines) have significantly better PRRs than the other scenarios with larger message sizes. This is simply because when all resource allocations are smaller, then the probability of colliding with another resource allocation of the same size would undoubtedly be smaller than when all resource allocations are larger. Unfortunately, it can not be simply concluded that smaller messages necessarily translate into better PRRs, because comparing the results between the 5 SCs/msg (dotted lines) and the 10 SCs/msg (dashed lines) scenarios actually produces a counterintuitive outcome – the scenarios where all vehicle transmissions occupy 5 SCs/msg have PRRs that may be considered to be just as poor as that of the 10 SCs/msg scenario. The main reason why this is so may be explained by the fact that suboptimal resource allocations in the 5 SCs/msg scenario cause it to have a performance similar to that of the 10 SCs/msg scenario.

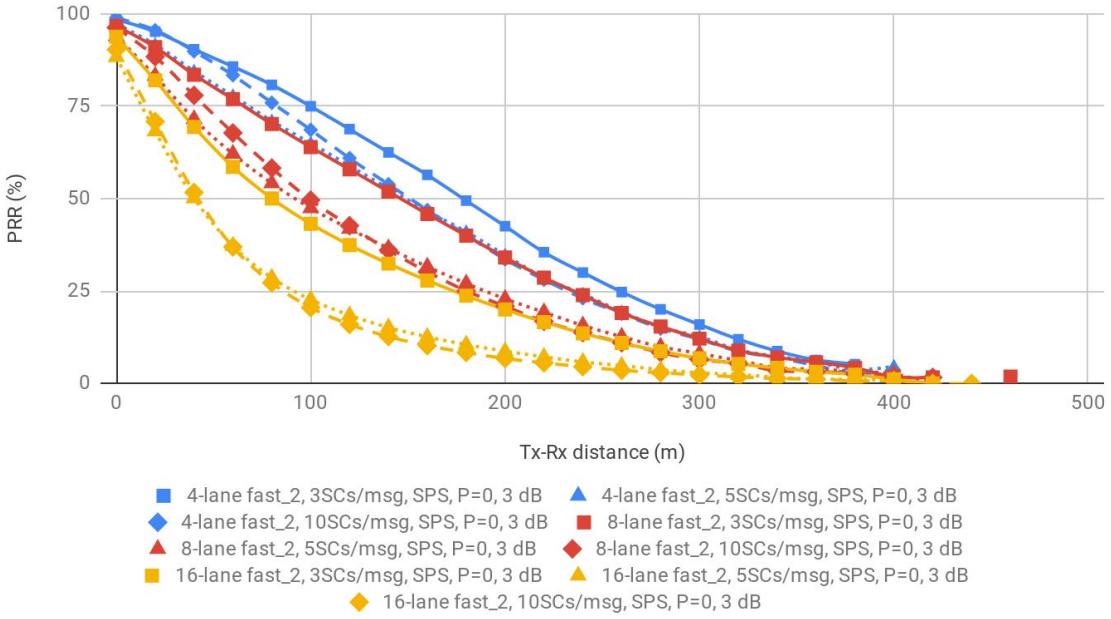

Figure 21. PRR when message size is varied along with number of lanes and car speed (default configuration)

The suboptimal allocation problem may be better understood by looking at the illustration on Figure 22. There are six possible configurations for allocating 5 subchannels onto a subframe with 10 subchannels, i.e. by allocating subchannels 1-5, 2-6, 3-7, 4-8, 5-9, or 6-10. Case (a) on the left shows that when subchannels 1-5 of a certain subframe have already been allocated with a resource, then another message of the same size that would be allocated to subchannels 1-5, 2-6, 3-7, 4-8, or 5-9 could cause a potentially disruptive collision. However, it is still possible to fit another message of the same size in subframes 6-10. In contrast, case (b) the right shows that when a subchannels 4-8 have already been allocated, then there is no way to fit another message of the same size without causing a potentially disruptive collision. In this sense, the resource allocation in case (b) is suboptimal compared to the resource allocation in case (a). Consequently, in the 5 SCs/msg scenario, it could be said that the allocation of subchannels 1-5 and 6-10 are optimal, whereas the allocation of subchannels 2-6, 3-7, 4-8, 5-9 are suboptimal.

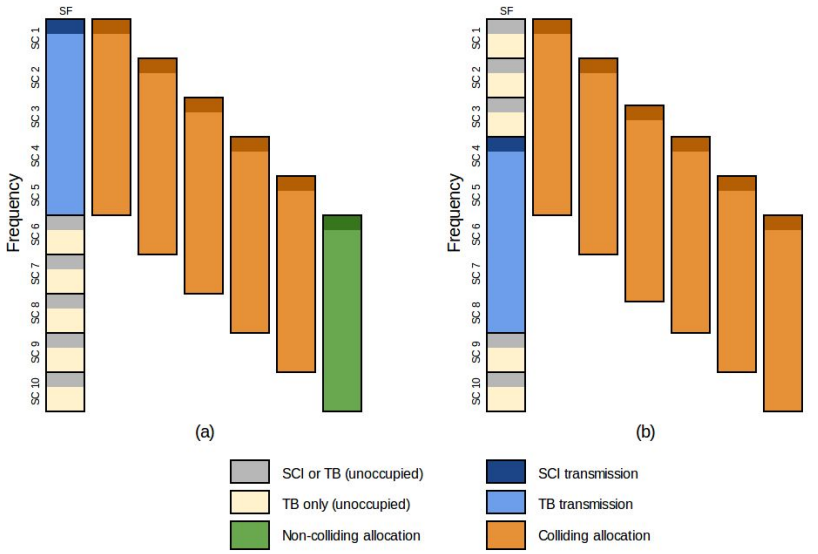

Figure 22 Illustration of relatively (a) optimal and (b) suboptimal resource allocations in the 5 SCs/msg scenario

Given a 10 SCs/subframe resource grid, suboptimal resource allocations also appear in scenarios where the message sizes are smaller than 5 SCs/msg. For instance, cases (a) and (b) in Figure 23 shows possible resource allocations in a scenario where all messages are of size 3 SCs. Case (b) may be regarded as suboptimal compared to case (a) in a sense that in case (b), 4 out of 8 possible allocations would cause a collision, whereas in case (a), only 3 out of 8 would. However, the advantage of the 3 SCs/msg over the 5 SCs/msg scenario is that, since the message sizes are smaller, it is still possible to fit in one or even two more messages of the same size in the same subframe even with a suboptimal allocation as in case (b). On the other hand, a suboptimal allocation in the 5 SCs/msg scenario, would always lead to a potentially disruptive collision if another node insists on allocating a resource in that subframe.

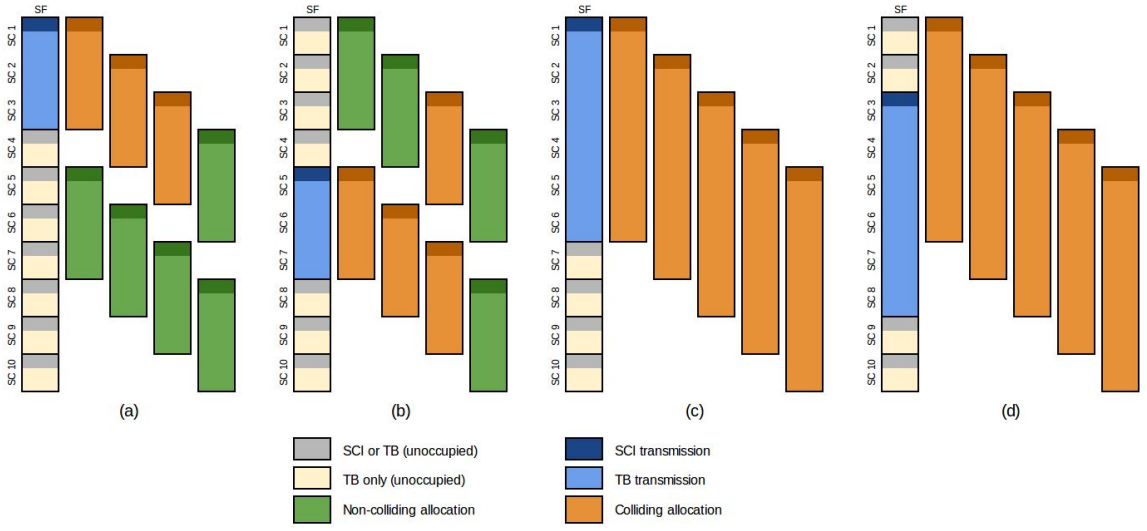

Figure 23 Relatively (a) optimal and (b) suboptimal allocations in the 3 SCs/msg scenario, and suboptimal (c) (d) allocations in the 6 SCs/msg scenario

In scenarios where the message sizes are smaller than 5 SCs/msg however, it could be considered that all possible resource allocations are suboptimal in a sense that no matter how a resource is allocated, it would no longer be able to fit any other messages of the same size, and that resource allocations by other nodes in the same subframe would always lead to a potentially disruptive collision. This situation is illustrated in cases (c) and (d) of Figure 23.

As we have seen so far, the very nature of subchannel-based allocation creates resource allocations that are suboptimal. Unfortunately, the design of the sensing-based SPS algorithm exacerbates the problem because, assuming that all candidate resource allocations in a subframe are equally viable, the probability that the sensing-based SPS algorithm chooses a suboptimal location for the resource is actually higher. Recall that for the 5 SCs/msg scenario, there are two optimal and four suboptimal configurations. Assuming that all of the above configurations are equally viable, and the sensing-based SPS would allocate resources on a free subframe, then the sensing-based SPS chooses the suboptimal resource allocations four out of six times (or 67% of the time). The suboptimal resource allocations in the 5 SCs/msg are such that it is no longer too different from the 10 SCs/msg scenario, because in both cases, only one message could properly fit in the subframe, and any other resource allocations in the same subframe would cause a collision. And since suboptimal resource allocations are more probable due to the random selection part in sensing-based SPS, then this makes the 5 SCs/msg scenarios perform just as badly as the 10 SCs/msg.

Digressing a bit from the above discussion, a study [18] concluded that simply increasing the number of subchannels increases the PDR, as discussed in the review of related literature (Section 2.3). It could be argued that their conclusion is valid when the message sizes are smaller than the potential capacity of the TB, such that even if the resource grid were divided into more but smaller subchannels, it would not change the number of subchannels occupied by each message. For example, suppose that vehicles all transmit message with sizes that are sufficiently small, such that even if the resource grid was configured to have 2 subchannels/subframe or 10 subchannels/subframe, the message would only still occupy one subchannel. The scenario where resource grid was configured to have only 2 subchannels/subframe would, surely experience a higher rate of collisions because of the much fewer choices for resource allocation.

However, suppose that the message sizes are large enough such that the messages need to occupy multiple subchannels. In this case, it could be argued that dividing the resource grid into more but smaller subchannels may actually be detrimental. Looking at the left illustration of Figure 24, it can be seen that having a resource grid configured to have 10 subchannels/subframe in 5 SCs/msg scenario creates suboptimal resource allocations, as in subframes 6 to 10. On the other hand, if the resource grid were configured to have only 2 subchannels/subframe as in the right illustration of Figure 24, then the suboptimal allocations may be avoided altogether.

With the above findings, we have thus seen that the relationship of message size with PRR performance is not as straightforward. It is imperative that the characteristics of the messages sent by the vehicles be properly studied so that V2X system designers would be able to design a resource grid configuration that is optimized.

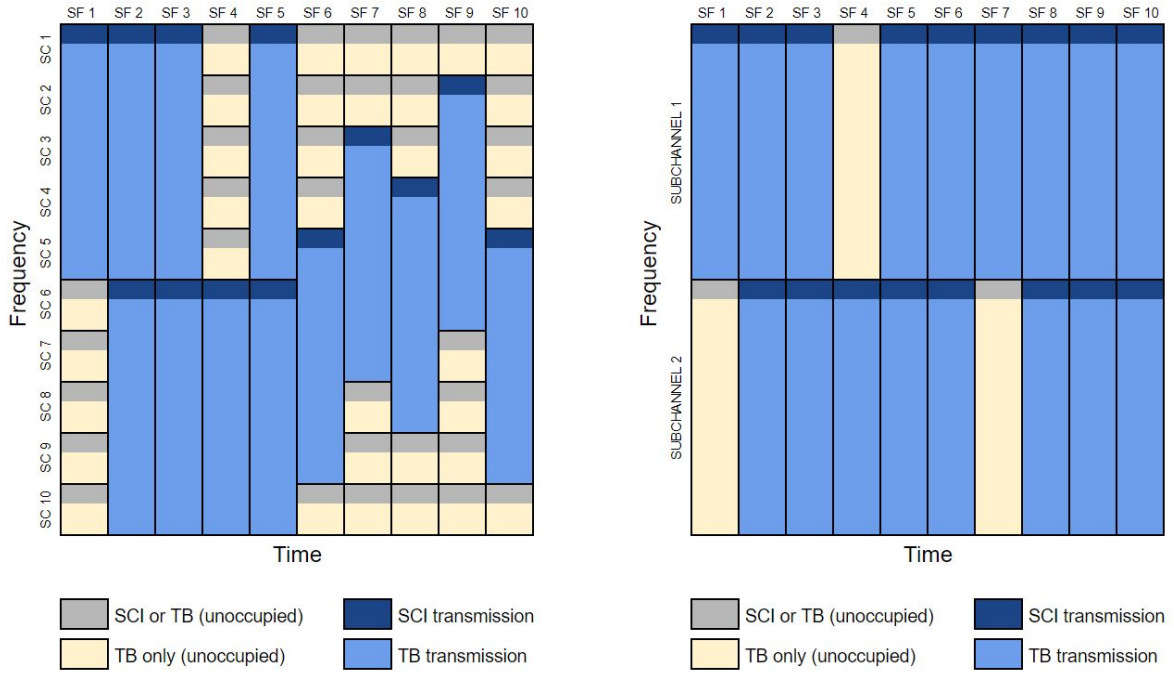

Figure 24. Illustration of resource grid packing when there are 10 SCs/subframe (left) vs. 2 SCs/subframe (right)

4.1.3. Impact of Collision Power Threshold

Shown on Figure 25 is the effect on the PRR of varying the collision power threshold (3 or 20 dB) along with the number of lanes (4, 8, or 16) of the default configuration.

First looking at the effect of the collision power threshold change regardless of the vehicle traffic load, it appears that increasing the collision threshold from 3dB to 20dB pushes the PRR performance down particularly in short distances. The fact that the PRR of the 20 dB scenario is lower is simply because it is more difficult to satisfy the 20 dB threshold condition; it requires that the S-RSSI of a received packet is at least 20 dB higher (or 100 times higher, in linear) than the combined S-RSSI of other colliding packets for the packet to be considered as a good packet. However, at Tx-Rx distances beyond 200 meters, there is hardly no more difference between the 3dB and the 20dB cases. This phenomenon can be attributed to the fact that, as uncovered by [7], collisions are problematic at short to medium distances, whereas from medium to long distances, the signal is already too weak to cope for interference regardless of collisions. This is why from at Tx-Rx distances of 200m and beyond, the collision threshold value does not matter much anymore since collision errors are already overshadowed at long distances.

Now looking at the effect of the collision power threshold on the different vehicle traffic loads, it appears that the effect of increasing the collision power threshold becomes more exaggerated with higher vehicle traffic. In fact, looking at the Tx-Rx distance of 20m, the PRR gap between the 3dB and 20dB scenarios for the 16 lane scenario even goes beyond 30%, whereas the gap only goes to a little more than 10% for the 4-lane scenario. This is because scenarios with higher vehicle traffic also have higher chances of collisions, and since a higher collision power threshold makes it harder for a packet to be considered as a good packet when it collides with other packets, the destructive effect of

collisions are compounded. Therefore, the effects of increased collision power threshold grow higher in scenarios with higher collision probabilities.

Figure 26 depicts the effect on the PRR of changing the collision power threshold (3 or 20 dB) along with the number of lanes (4, 8, or 16), but using the 10 SCs/msg variation of the default configuration. As it could be seen, the impact of increased collision power threshold is more profound compared to the previous 3 SCs/msg scenario. Again, this could be attributed to the fact that the 10 SCs/msg scenarios have higher collision probabilities compared to the 3 SCs/msg scenarios, and since increasing the collision power threshold compounds the destructive effect of collisions, higher collision power thresholds would have a greater impact on the PRRs of the 10 SCs/msg scenarios than those of the 3 SCs/msg scenarios.

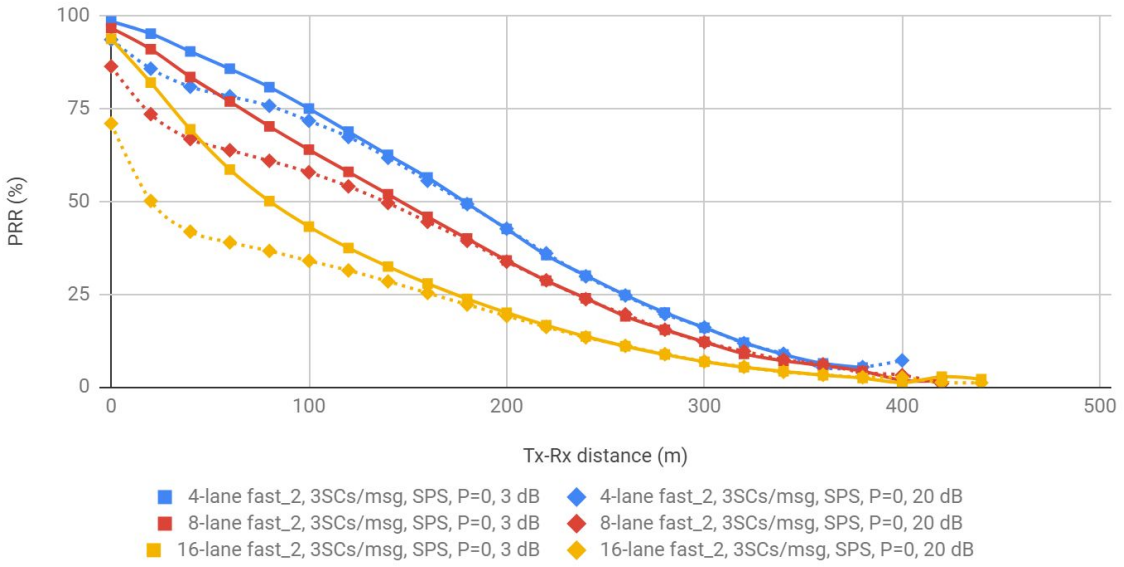

Figure 25. PRR when threshold is varied along with number of lanes (default configuration)

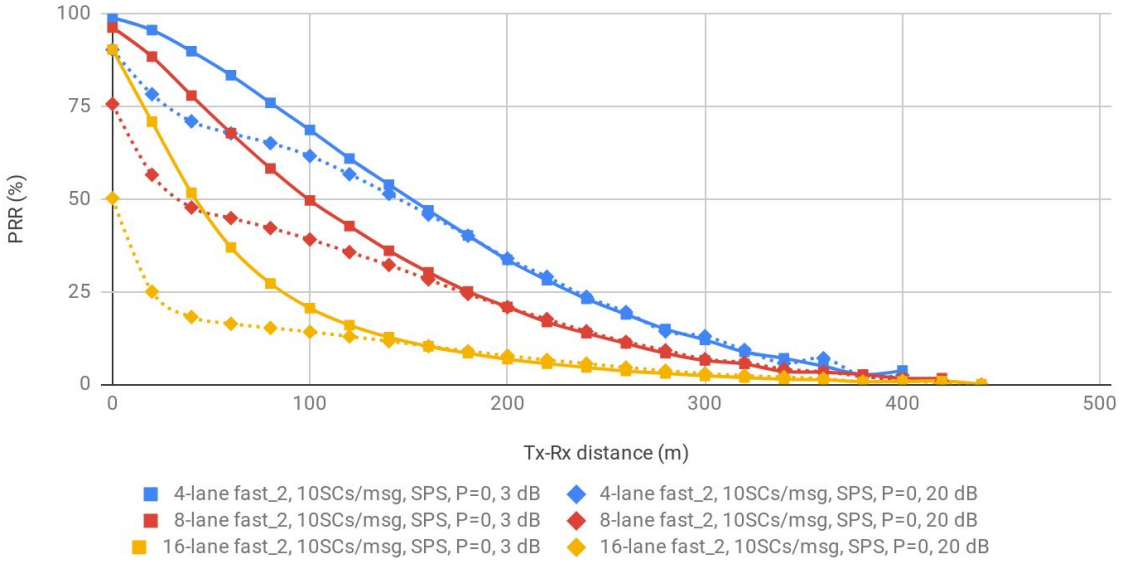

Figure 26. PRR when threshold is varied along with number of lanes (10 SCs/msg variant of default configuration)

4.1.4. Sensing-based SPS versus Random Allocation

Now, comparisons between the sensing-based SPS under different keep probabilities (*P*=0, 0.8, or 1.0) and random allocation scenario are discussed below. Note that the 3GPP standards only allow a maximum *P* of 0.8, but a keep probability of 1 was nevertheless included to see what the effect would be when the sensing-based SPS algorithm always keeps the same resource that it has first reserved.

Figure 27 shows the effect on the PRR of changing the resource allocation method, i.e. using sensing-based SPS (*P*=0, 0.8, or 1.0) or random allocation, along with the number of lanes (4, 8, or 16) of the default configuration. Figure 28 and Figure 29 show the same, but with message sizes of 5 SCs/msg and 10 SCs/msg respectively.

It could be observed that regardless of the message size and the vehicle traffic, there is hardly no remarkable difference between any of the keep probabilities of 0 and 0.8. It could also be observed that *P*=1.0 generally leads to slightly worse PRR than the other two variations of *P*, except in the 16-lane cases of the 5 SCs/msg and the 10 SCs/msg scenarios, where

Also, random allocation is always significantly worse when compared to the PRRs of those using sensing-based SPS. The differences between sensing-based SPS and random allocation grows more significant with increasing message sizes.

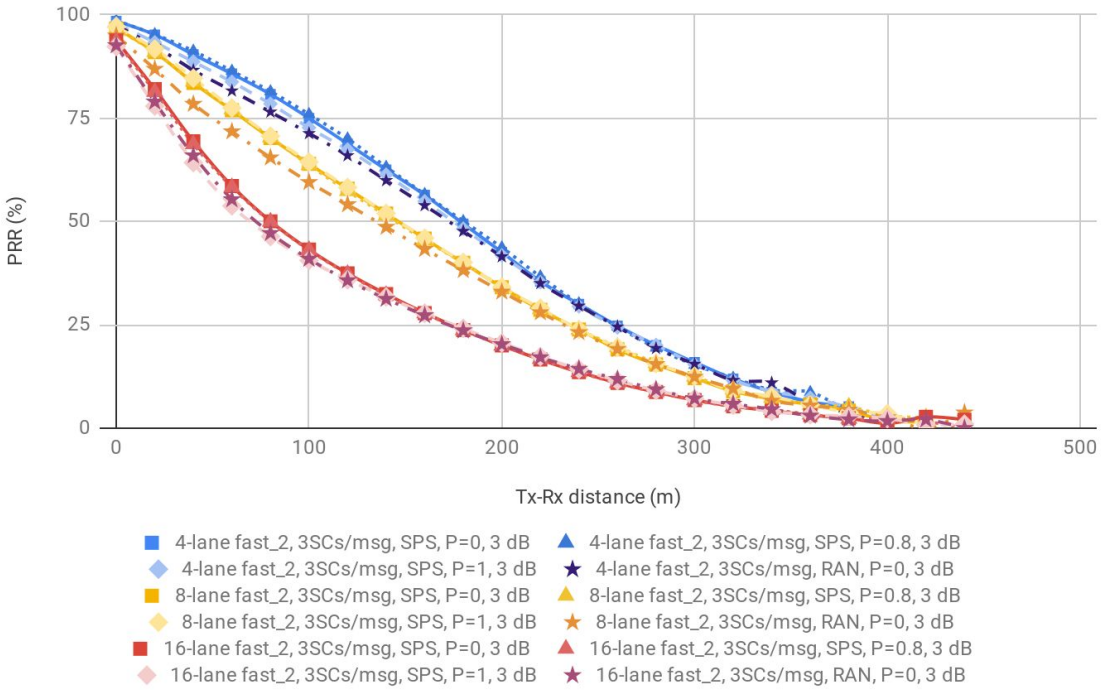

Figure 27. PRR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (default configuration)

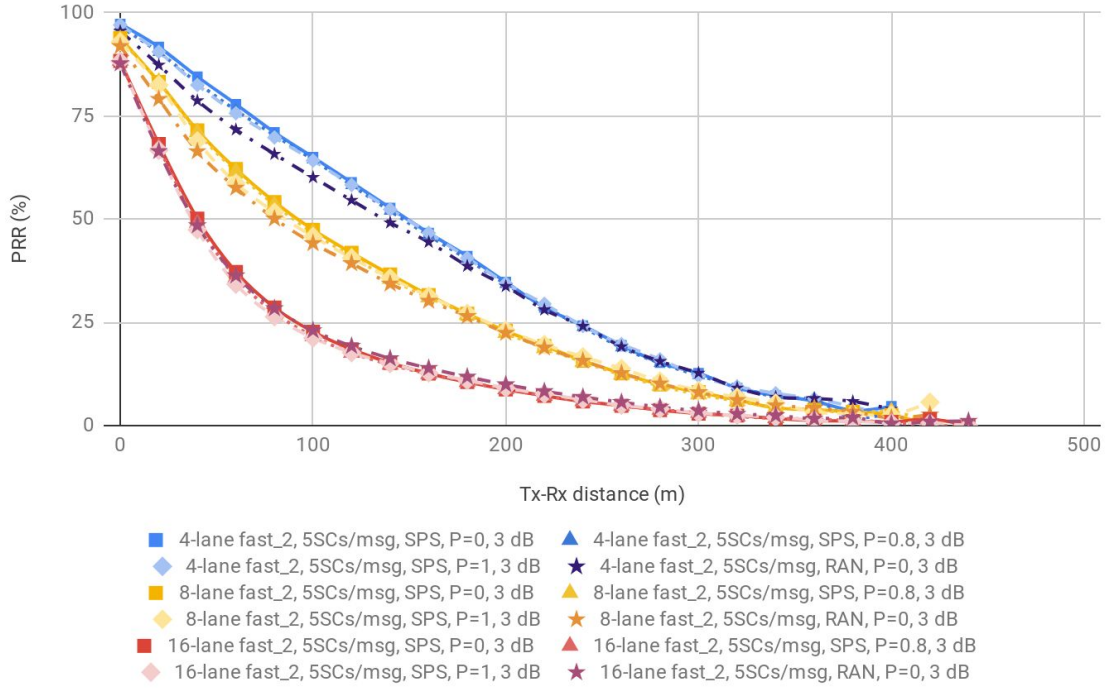

Figure 28. PRR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (5 SCs/msg variant of default scenario)

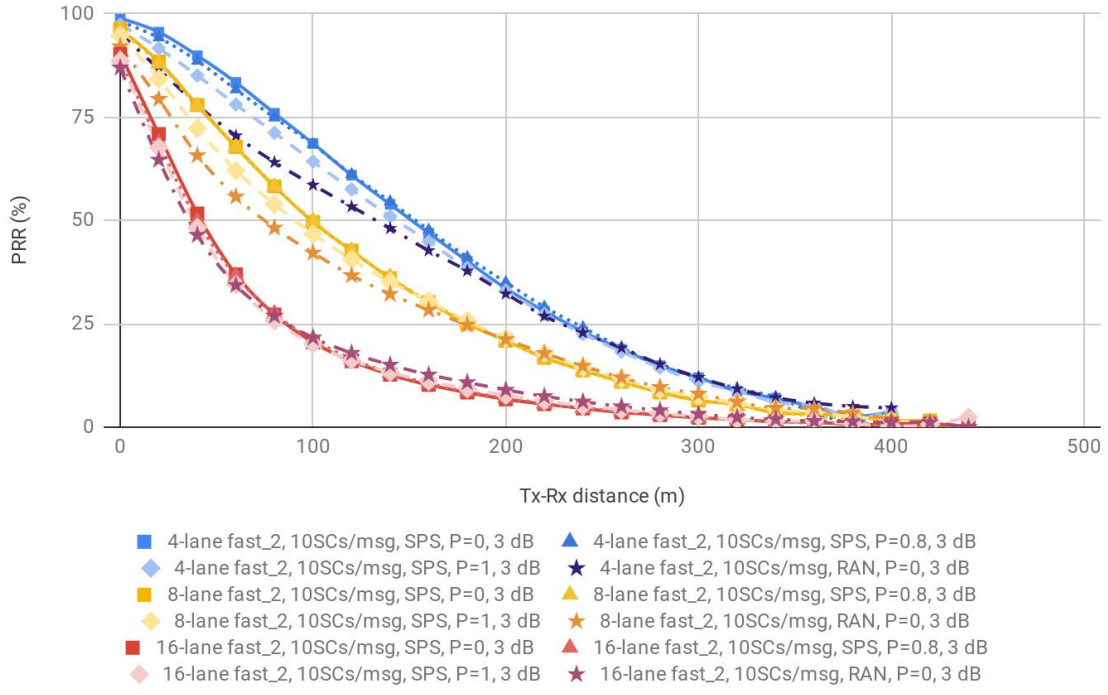

Figure 29. PRR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (10 Scs/msg variant of default scenario)

Looking briefly at a unidirectional highway case, Figure 30 shows the effect on the PRR of using sensing-based SPS (*P*=0, 0.8, or 1.0) or random allocation, along with the message sizes (3, 5, or 10 SCs/msg) of the default configuration. It appears that the trends are more or less the same as in the bidirectional case, i.e. *P*=0 and 0.8 are nearly the same, *P*=1 is worse than both, and random allocation performs the poorest.

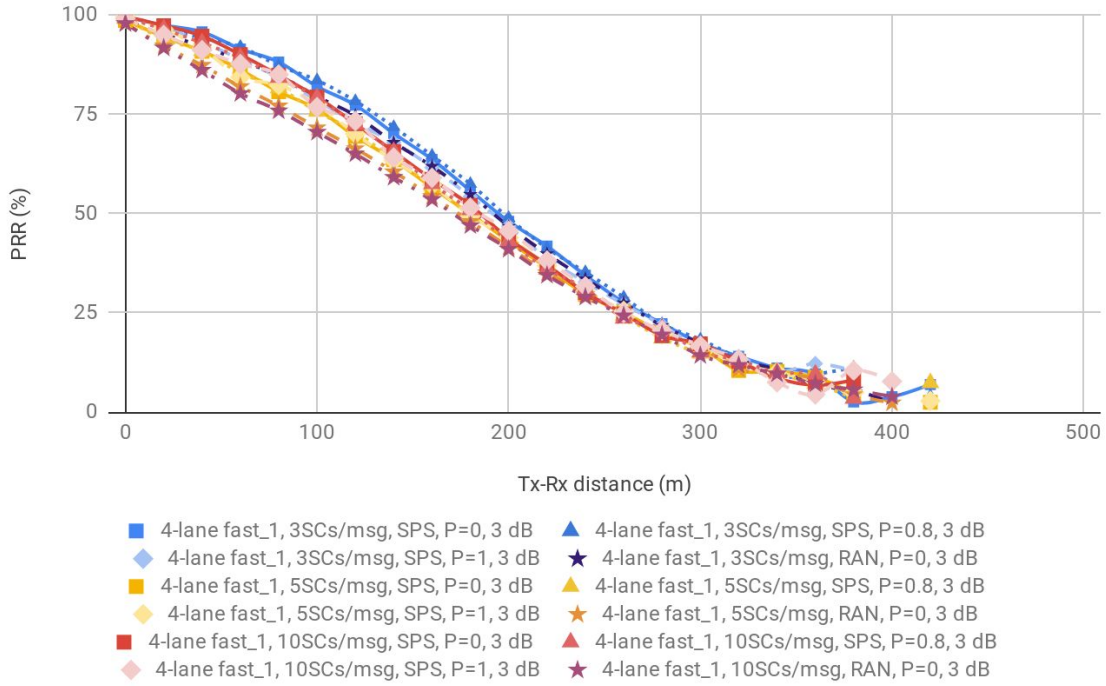

Figure 30. PRR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation with varying message size (unidirectional version of default configuration)

Looking briefly at a Manhattan grid case as well, Figure 31 shows the effect on the PRR of using sensing-based SPS (*P*=0, 0.8, or 1.0) or random allocation in a lightly-loaded Manhattan grid scenario with a constant 200 vehicles throughout the entire 445m×445m area. It could be seen that its PRR performance is again similar to those of the highway cases, except that the performance of *P*=1 is the same as the other keep probabilities.

It could be imagined that the dynamicity of the traffic in the Manhattan grid road traffic also creates a dynamic resource grid utilization similar to that of the bidirectional highway scenario, wherein there is an opposing flow of vehicles that may have different resource allocations. Thus it was expected that having a lower *P* would make the resource grid allocation more adaptable, and hence perform better, at least in the bidirectional highway and Manhattan grid scenarios. On the other hand, in the unidirectional highway, cars would tend to travel with the same group of vehicles, thus its resource allocations would tend to be more static, and consequently having a higher *P* would be expected to perform better. However, the results showed again that *P*=0 and 0.8 had no differences, and that *P*=1 performed the worst even in the unidirectional case. Even though this thesis project did not focus on running simulations using the unidirectional highway and Manhattan grid scenarios, it is nevertheless recommended that more investigations be done using those road traffic topologies in future studies.

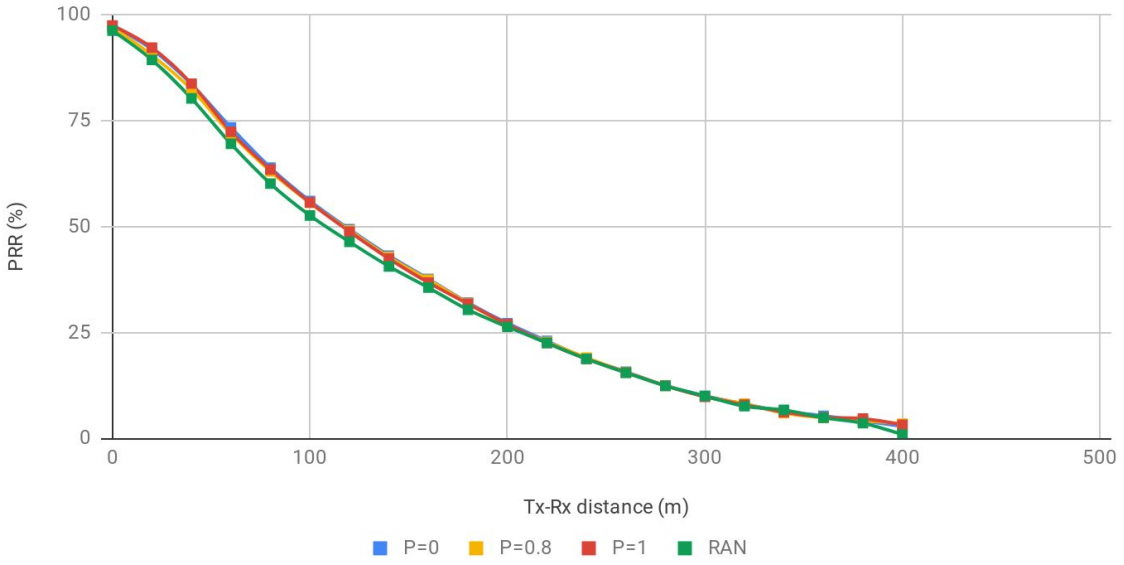

Figure 31. PRR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (Manhattan grid, 3SCs/msg version of default configuration)

Overall, the results suggest that regardless of the scenario, the distinctions in the PRRs of the standard keep probabilities (i.e. between 0 and 0.8) are minute. These results seem to be consistent with the results in [15] and [18], who also found that the keep probability had an unremarkable impact on the PRR. Nevertheless, sensing-based SPS still performs generally better than random allocation, though its advantages over random allocation becomes less significant with higher message traffic.

4.2. Resource Grid Occupancy and Goodput

The resource grid occupancy and goodput statistics are meant to provide insights on how the LTE-V2X resource grid is being efficiently utilized. Occupancy shows how much of the resource grid has been occupied, while goodput shows how much of the resource grid has been occupied by correctly-received packets. The occupancy and goodput statistics are calculated by checking the subchannel usage in the past 1000 subframes of the resource grid of the vehicle, and then plugging the values into the formulas below:

$$occupancy = \frac{occupied\ subchannels}{total\ subchannels} \times 100$$

$$goodput = \frac{good\ subchannels}{total\ subchannels} \times 100$$

$$occupied\ subchannels = good\ subchannels + corrupt\ subchannels$$

$$total\ subchannels = occupied\ subchannels + unused\ subchannels$$

As an example, suppose that the resource grid is represented by Figure 32. There are 100 total subchannels, 61 of which are occupied. Among those occupied, 40 of those are good (blue), and 21 of those are corrupt (red) subchannels. In this case, the occupancy would be 61% and the goodput would be 40%.

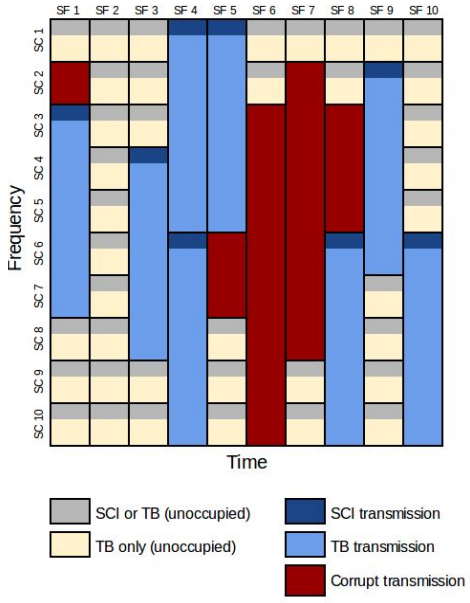

Figure 32. Illustration of resource grid with occupied (blue) and corrupt (red) subchannels

The occupancy and the goodput statistics are gathered every 100 ms for all the vehicles that are present in the scenario at the time of measurement. Both statistics are depicted as functions of the message traffic density, which is represented as the number of CAMs per second per kilometer. This is calculated by multiplying the vehicle density (vehicles/km) by the CAM transmission rate of the vehicle (cams/sec/vehicle).

The effects of varying the message size (Section 4.2.1) and collision threshold (Section 4.2.2) on the resource grid occupancy and goodput when using sensing-based SPS are henceforth discussed. Comparisons between the occupancy and goodput performances of sensing-based SPS and random allocation (Section 4.2.3) are also presented. Note that the effects of the traffic load are already implicit in the occupancy and goodput graphs, therefore, its impact on occupancy and goodput is not discussed.

4.2.1. Impact of Message Size

Shown on Figure 33 is the effect on the resource grid occupancy and goodput of changing the message size (3, 5, or 10 SCs/msg) of the default configuration.

As expected, the scenarios with larger message sizes have larger occupancies due to the fact that the message sizes take up more resources in the grid. However, unlike in the PRR statistics, wherein the 5 SCS/msg and the 10 SCs/msg scenarios had very similar PRR performance, the occupancy and goodput of the 5 SCs/msg and the 10 SCs/msg scenarios are actually much more separated from each other. In line with the above

reasoning, this gap is because the 10 SCs/msg scenarios simply occupy more resources than the 5 SCs/msg scenarios, regardless of the detrimental effect of suboptimal resource allocations on the PRRs of the 5 SCs/msg scenarios (Section 4.1.2).

When the message traffic is low, there are hardly any collisions that occur, hence the occupancy is more or less the same as the goodput value. But as the message traffic increases, the occupancy and the goodput begin to diverge more, and at a certain point, the goodput values seem to plateau, and even start to fall down. This is expected behavior, because as the message traffic density grows, so do the chances that messages become corrupted due to collisions, thereby making the resource grid goodput fall. Occupancy, unlike goodput, is monotonically increasing, although the amount of increase in occupancy decays exponentially with rising message traffic density.

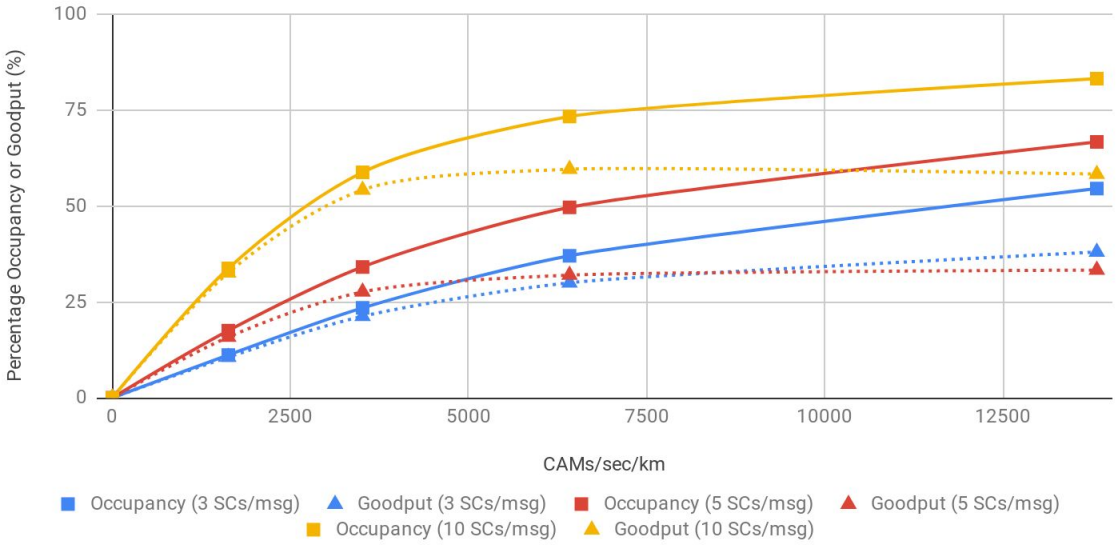

Figure 33. Occupancy and goodput when message size is varied (default configuration)

The points when the goodput values stagnate and go down shift depending on the message size. Looking again at Figure 33, the goodput curve of the 3 SCs/msg scenario seems to continue rising with the occupancy curve even beyond 12500 CAMS/sec/km, whereas the 5 SCs/msg scenario goodput is plateauing, and the 10 SCs/msg scenario goodput is already beginning to fall. This is just because the scenario with smaller message sizes tend to have less collisions than scenarios with larger message sizes, and are consequently less affected by the increase in collision power threshold. Thus, the goodput of the 10 SCs/msg scenario reaches its peak point at a lower message traffic density than that of the 3 SCs/msg scenario.

In addition, the results show that having the 10 SCs/msg scenario had the highest possible peak goodput value. However, it is not the case that merely increasing the message size necessarily leads to higher peak goodput values. As it could be seen in Figure 33, the goodput of the 5 SCs/ msg scenario has already peaked and is plateauing from around 6400 CAMs/sec/km, whereas the goodput of the 3 SCs/msg scenario still continues to rise, and even overtakes the goodput of the 5 SCs/msg scenario from about 9000 CAMs/sec/km. This may be attributed to the suboptimal resource allocations in the 5 SCs/msg scenario, as previously discussed in Section 4.1.2. Unfortunately though, the

simulation results were not able to show at which point the 3 SCs/msg scenario would arrive at its peak goodput value.

4.2.2. Impact of Collision Power Threshold

Figure 34 shows the effect on the on the resource grid occupancy and goodput of changing the message size (3, 5, or 10 SCs/msg) along with the collision power threshold (3 dB or 20 dB) of the default configuration.

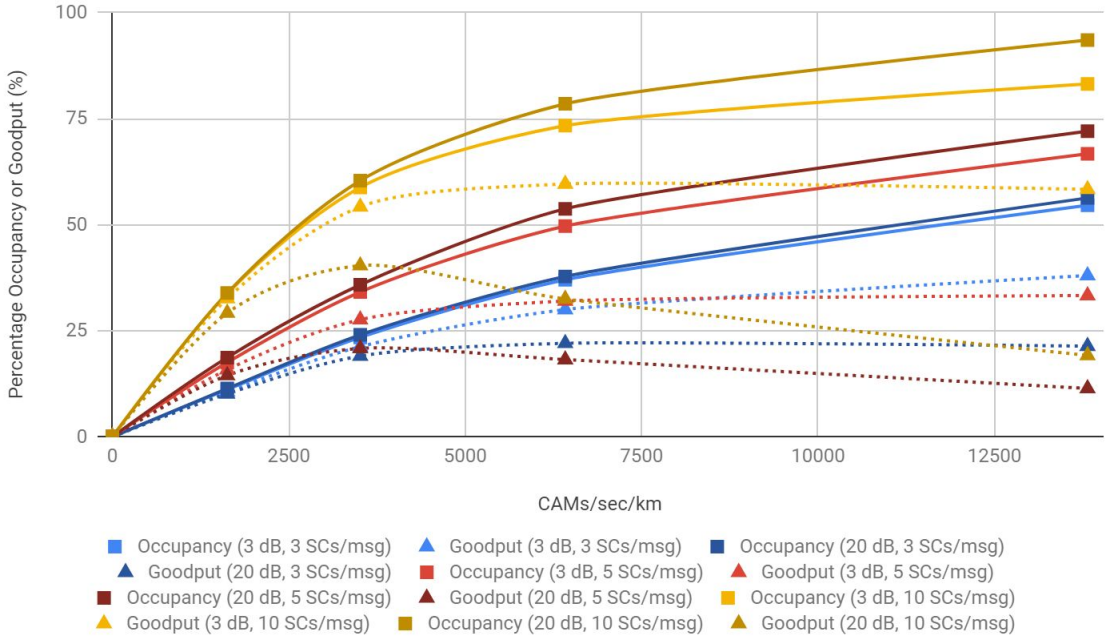

Figure 34. Occupancy and goodput when message size and collision threshold are varied (default configuration)

Regardless of the message size, increasing the collision threshold from 3 dB (light colors) to 20 dB (dark colors) would negatively affect the goodput curves because satisfying the 20 dB threshold in order to be considered as a good packet is already more difficult than satisfying the 3 dB threshold, and its impact would be more significant as message traffic densities increases. In fact, the rise, peak, and fall of the goodput curve is much more pronounced in the case with 10 SCs/msg with 20 dB collision threshold.

An unexpected result is that using a collision threshold of 20 dB actually led to a higher resource grid occupancy when the message traffic density is high. This effect is not as remarkable in the 3 SCs/msg scenarios, but it becomes more significant with increasing message size. Additional simulations or further studies may be necessary to more clearly establish the effect of collision threshold on the resource grid occupancy.

4.2.3. Sensing-based SPS versus Random Allocation

Figure 35, Figure 36, and Figure 37 show the impacts on the resource grid occupancy and goodput of changing the message sizes to 3, 5, or 10 SCs/msg when using

sensing-based SPS (*P*=0, 0.8, or 1.0) or random allocation variations of the default configuration.

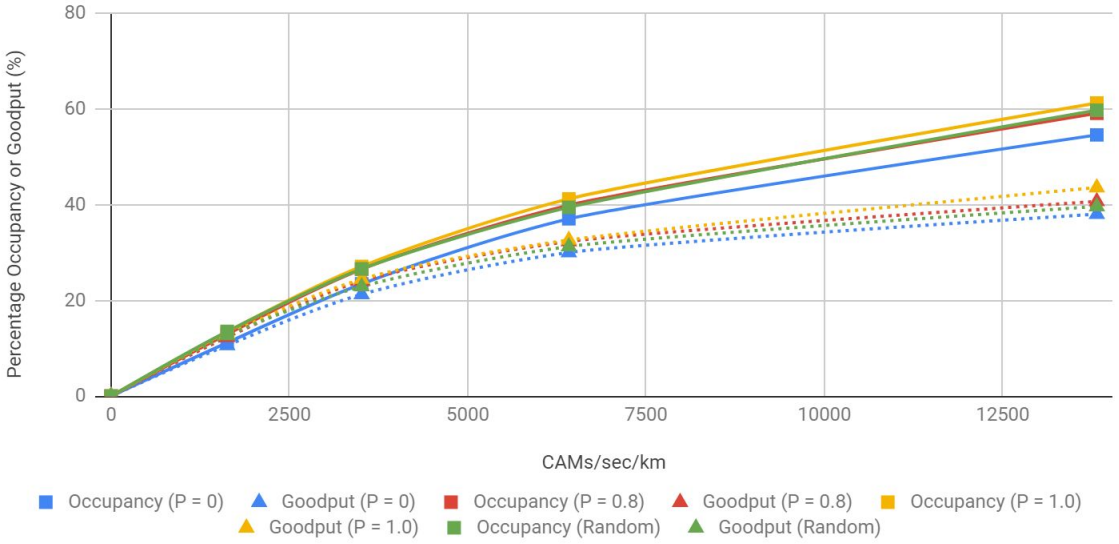

Figure 35. Occupancy and goodput of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (default configuration)

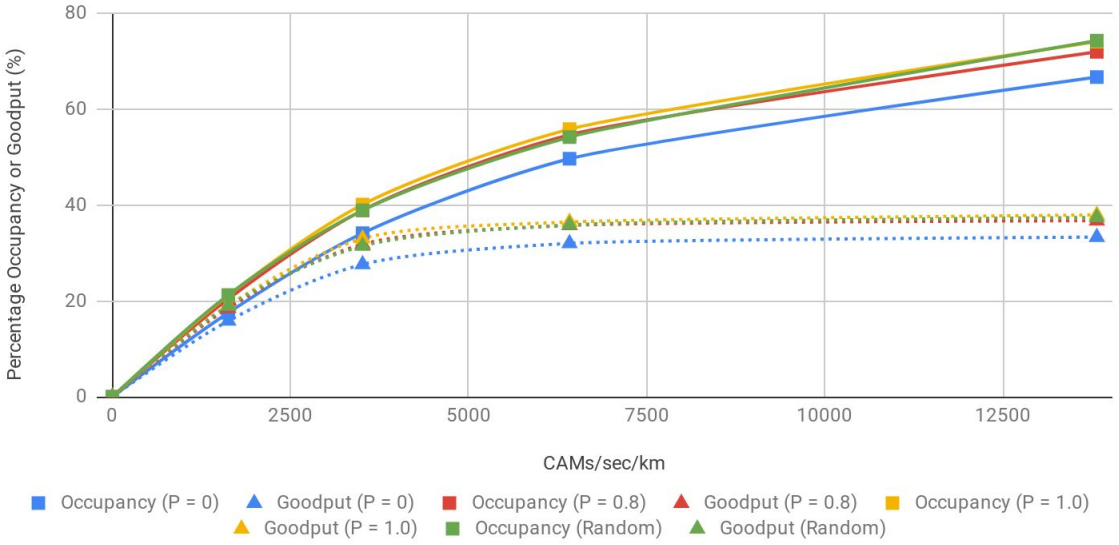

Figure 36. Occupancy and goodput of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (5 SCs/msg version of default configuration)

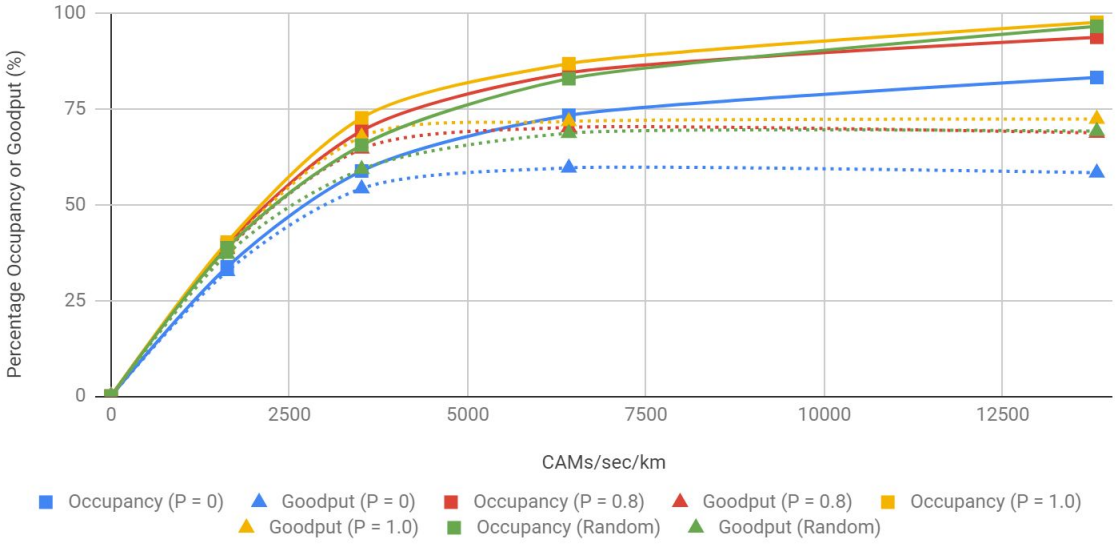

Figure 37. Occupancy and goodput of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (10 SCs/msg version of default configuration)

In terms of PRR, it was shown that a keep probability *P*=0 basically did not differ too much from *P*=0.8, but in terms of resource grid occupancy and goodput, the scenarios with *P*=0 were markedly worse off. Also, the scenarios with *P*=1 generally performed the best in terms of occupancy and goodput, even though their PRR results were generally the worst compared to the other two keep probabilities. These were rather surprising results, in a sense that it was initially expected that scenarios that had better PRR performances would also tend to have more efficient use of the resource grid, and would therefore have better occupancy and resource grid statistics. The results here show that it is not necessarily so.

The other surprising discovery was that, despite the fact that random allocation, which has an unrestricted scheduling policy that allows reservations to be very dynamic, has similar occupancy and goodput performance with *P*=0.8 and 1, which have more static reservations due to their high probability of keeping their previous reservations. This is especially prevalent in the 3 SCs/msg scenario.

Finally, Figure 38 shows once more the impact of sensing-based SPS with varying keep probability versus the random allocation, under a bidirectional highway scenario but using a threshold of 20 dB. Again, as in the previous subsection where the 3 dB case was discussed, increasing *P* under a 20 dB collision threshold still generally improves the occupancy and the goodput, while random allocation has a performance that is pretty close to that of *P*=0.8, but only in medium message traffic densities. Unlike the 3 dB case, the occupancy and goodput graphs start to converge at high traffic densities in the 20 dB case.

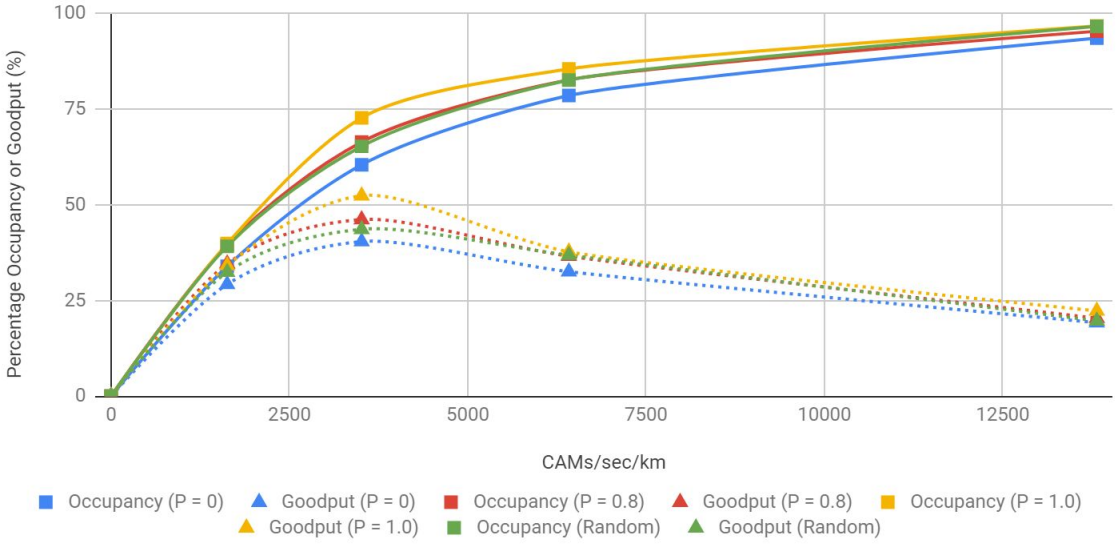

Figure 38. Occupancy and goodput of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (10 SCs/msg, 20 dB threshold version of default configuration)

It is recommended that the relationship of the resource grid occupancy and goodput statistics with the usage of sensing-based SPS with varying keep probabilities or random allocation be investigated future research studies.

4.3. Neighborhood Awareness Ratio (NAR)

The Neighborhood Awareness Ratio (NAR) may be used to evaluate the level of cooperative awareness in the simulated LTE-V2X Release 14 Mode 4-based system. NAR is calculated as follows:

$$NAR = \frac{perceived\ number\ of\ vehicles}{actual\ number\ of\ vehicles} \times 100$$

In this simulation, the NAR results are taken from the application layer data point of view, i.e. the perceived vehicles of a node are those that are recorded in its own application layer-level database (Section 3.2.3). The database stores information about other vehicles that a node sees within its range, and if no transmissions are received from a previously known vehicle for an arbitrarily-set duration of 20 message transmission intervals (or 2 seconds), its associated entry in the memory is deleted. The NAR results are then represented as functions of the distances between the actual positions of the nodes.

Following the flow of discussion in the previous sections, the effects of varying the vehicle traffic load (Section 4.3.1), message size (Section 4.3.2), and collision threshold (Section 4.3.3) on the NAR when using sensing-based SPS are henceforth discussed.

Comparisons between the NAR performances of sensing-based SPS and random allocation (Section 4.3.4) are also presented.

4.3.1. Impact of Traffic Load

Shown on Figure 39 is the effect on the NAR of varying the number of lanes (4, 8, or 16) of the default configuration.

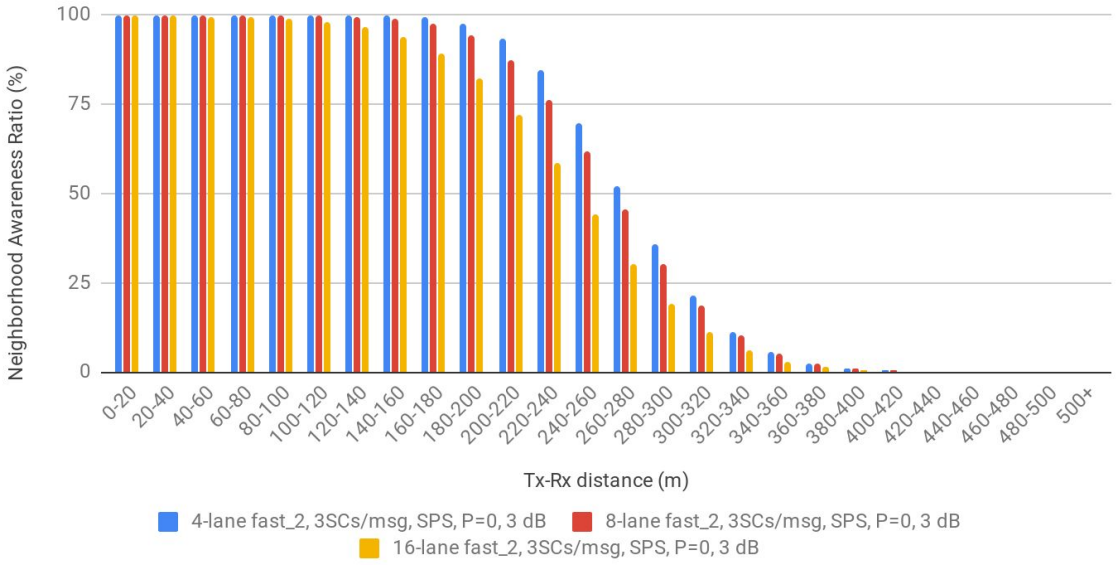

Figure 39. NAR when lane count is varied between 4, 8, and 16 lanes (default configuration)

Regardless of the vehicle traffic, the NAR generally stays relatively flat and very close to 100% up until the medium distances before it starts to fall dramatically. This is contrast with the PRR results, which always decreases significantly with increasing Tx-Rx distance without plateauing, regardless of the scenario. Observing the 4-lane scenario for instance, its NAR at a distance of 160-180 m is still above 99%, whereas its PRR is only about 56%. This means that even if some transmissions are lost, the application-layer level database allows the vehicle to still perceive the other vehicles. Beyond 400m however, the NAR is virtually zero for all cases, because the probability of getting successful receptions is already exceedingly low at those distances. This is the same as in the PRR results.

Looking at the differences between the NAR results of the three vehicle traffic scenarios, it could be seen that the NARs of the higher vehicle traffic scenarios tend to start falling at smaller distances compared to those of the smaller vehicle traffic scenarios. Same with the PRR, this could be explained by the fact that higher vehicle traffic leads to higher message traffic, therefore increasing the chances of collisions.

Now, Figure 40, Figure 41, and Figure 42 show the NAR when the vehicle speed is varied (fast and slow) in the 4, 8, and 16-lane variants of the default configuration, respectively. The NARs of the slower scenarios are lower than those of the faster scenarios, and again, this is due to the higher vehicle traffic in the slow scenarios. It could be noticed that

the differences between the NARs of the fast and slow scenarios (in the medium to long distances) of the 4-lane case are smaller compared to those in the 8 and 16-lane cases. This is because the differences in the densities of vehicle traffic between the fast and slow scenarios of the 4-lane case is also smaller (about 188 vehicles/km) compared to those of the 8 (379 vehicles/km) and 16-lane (740 vehicles/km) scenarios.

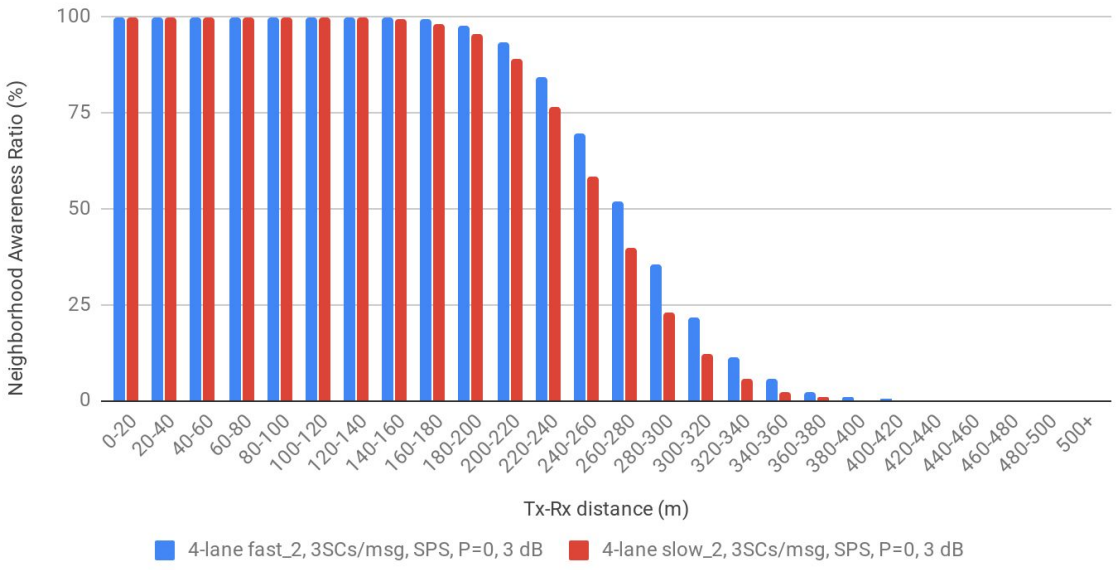

Figure 40. NAR when vehicle speed is varied between fast and slow (default configuration)

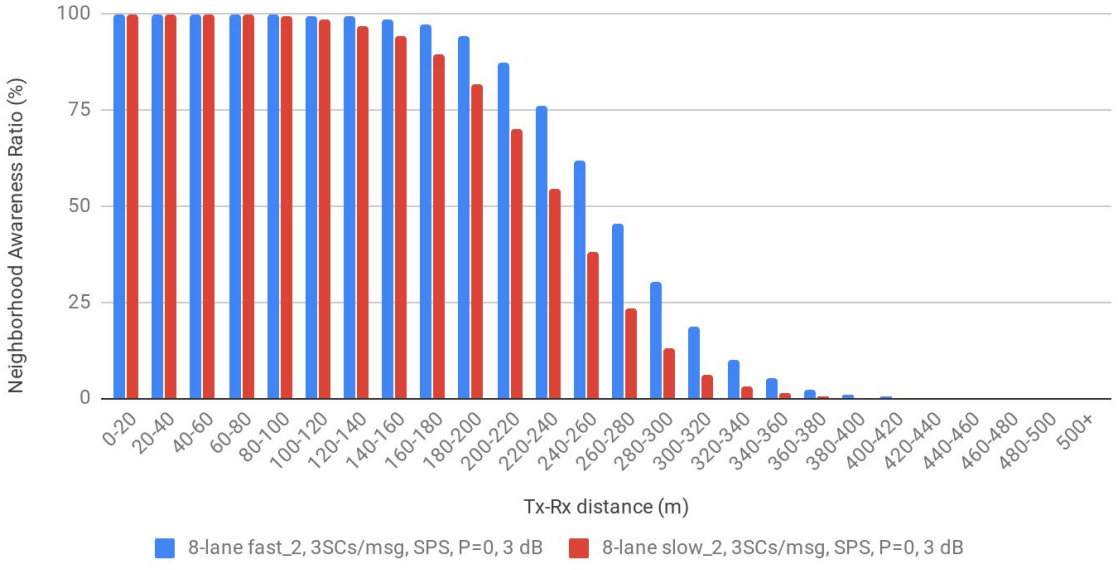

Figure 41. NAR when vehicle speed is varied between fast and slow (8-lane variant of the default configuration)

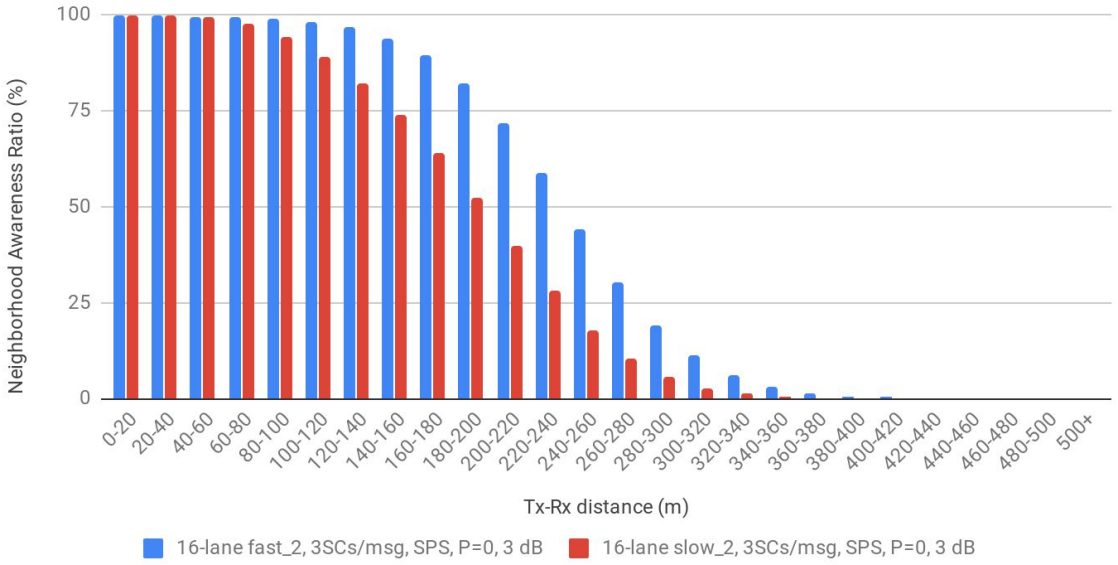

Figure 42. NAR when vehicle speed is varied between fast and slow (16-lane variant of the default configuration)

4.3.2. Impact of Message Size

Figure 43 shows the effect on the NAR of varying the message size (3, 5, or 10 SCs/msg) of the default configuration. Figure 44 shows the same thing, but on the slow variant of the default configuration.

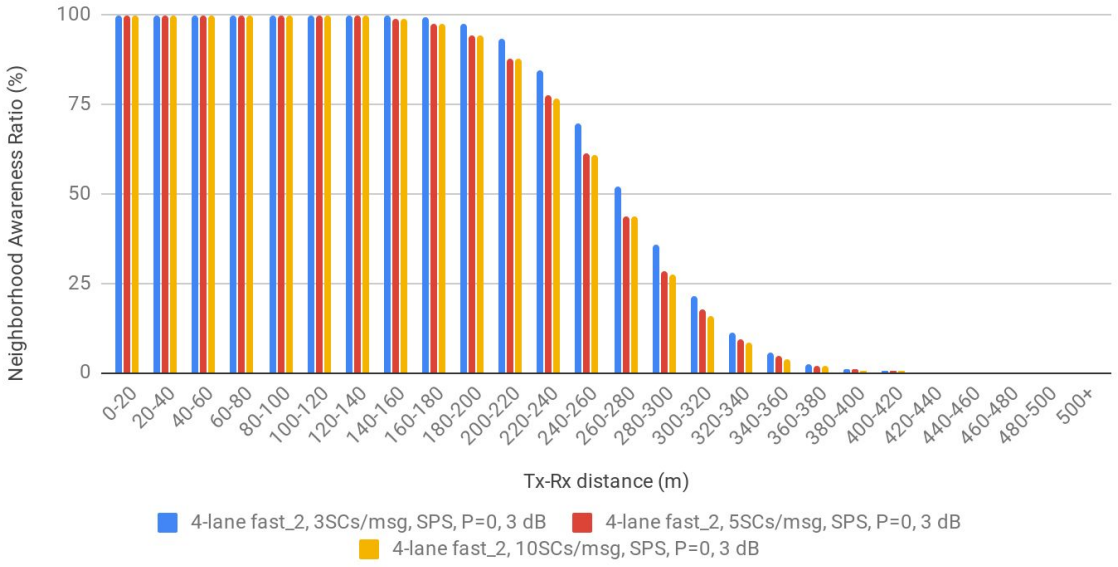

Figure 43. NAR when vehicle message size is varied between 3, 5, and 10 SCs/msg (default configuration)

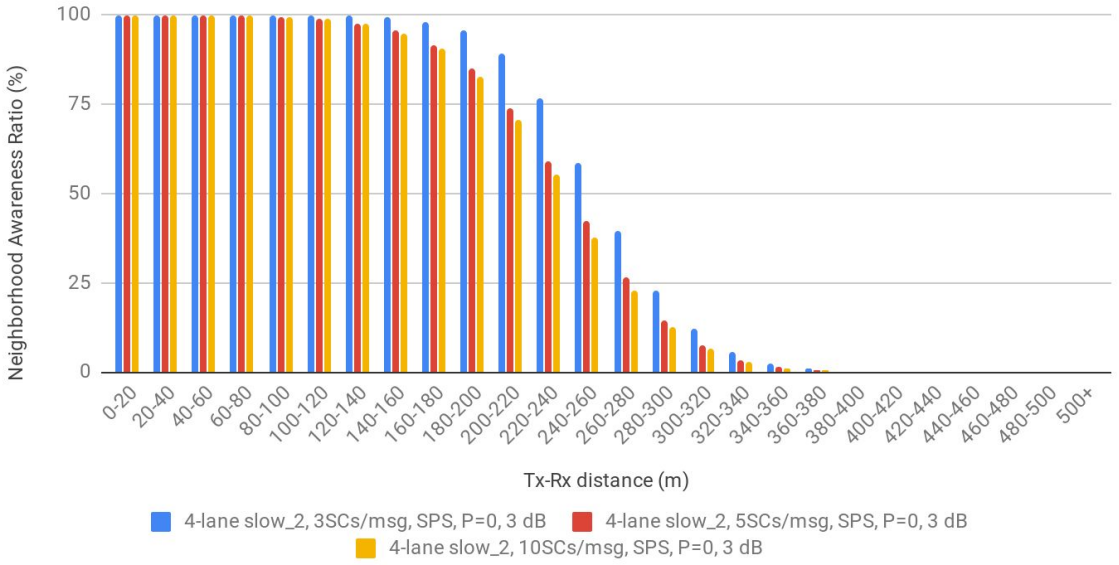

Figure 44. NAR when vehicle message size is varied between 3, 5, and 10 SCs/msg (slow variant of the default configuration)

In the fast vehicle scenario, the message size has no remarkable impact on the NAR in short to mid distances, but from about 160-180 m onwards, the difference between the 3 SCs/msg scenario and the other two scenarios become slightly more evident, although beyond 400m, all message size scenarios all have nearly zero NAR values. The gap between the NAR of the 3 SCs/msg and the other two scenarios becomes much more significant with the denser slow vehicle scenario. Looking on the other hand at Figure 45 and Figure 46, which respectively show the NAR in the 16-lane fast and slow variants of the default configuration, it could be seen that the gap is even much larger, and the gap arises at shorter distances.

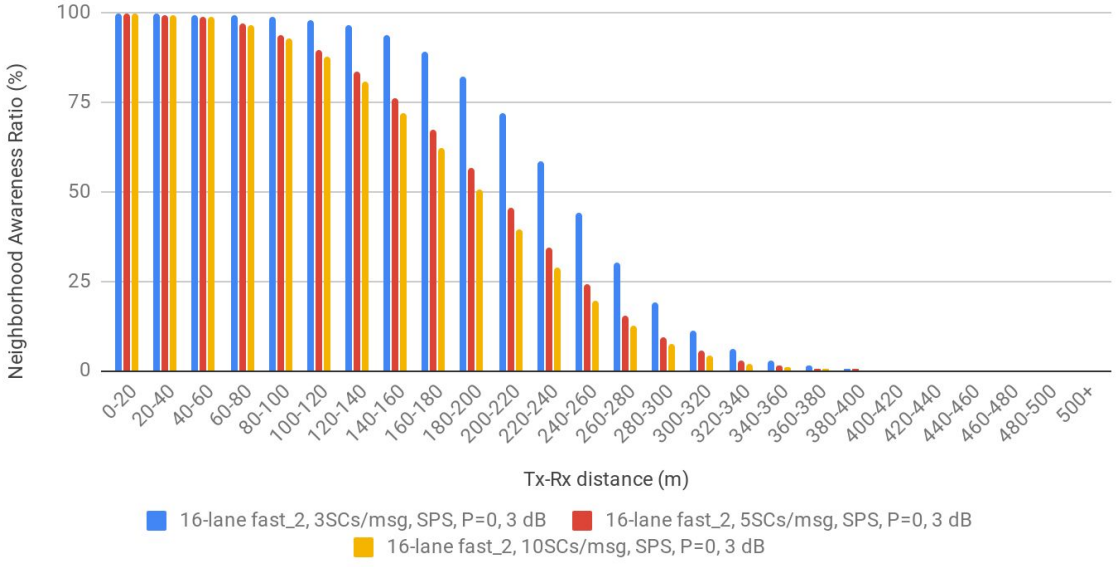

Figure 45. NAR when vehicle message size is varied between 3, 5, and 10 SCs/msg (fast bidirectional 16-lane highway, SPS with *P*=0, 3dB threshold)

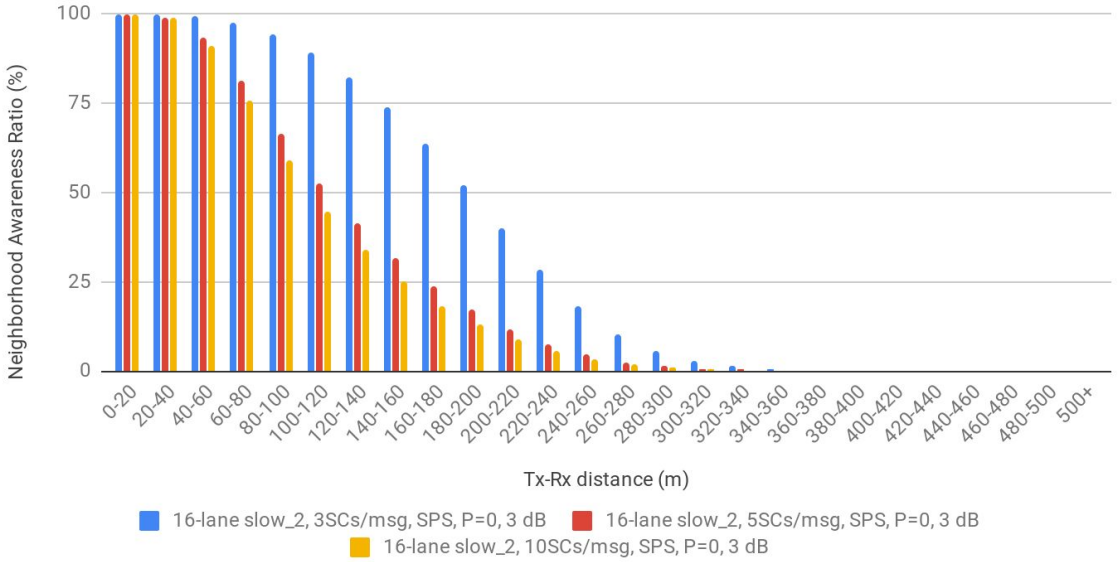

Figure 46. NAR when vehicle message size is varied between 3, 5, and 10 SCs/msg (slow bidirectional 16-lane highway, SPS with *P*=0, 3dB threshold)

Regardless of the traffic load however, the NARs of the 3 SCs/msg scenario is always better than those of the other two larger message scenarios because of the smaller collision probabilities of the 3 SCs/msg scenario. Also regardless of the traffic load, the NARs 5 SCs/msg and 10 SCs/msg scenarios are always not too far off from each other. Yet again, this may be explained by the suboptimal resource allocations in the 5 SCs/msg scenario, which make its collision probabilities to be nearly as bad as those of the 10 SCs/msg scenario (Section 4.1.2).

4.3.3. Impact of Collision Power Threshold

Figure 47 shows the effect on the NAR of varying the collision power threshold (3 or 20 dB) of the default configuration. Figure 48 shows the same thing, but on the 16-lane variant of the default configuration.

In both 4 and 16-lane cases, it seems that the impact of increasing the collision threshold to 20 dB was minimal. Still, the performance gap seems to be slightly larger in the 16-lane case than in the 4-lane case.

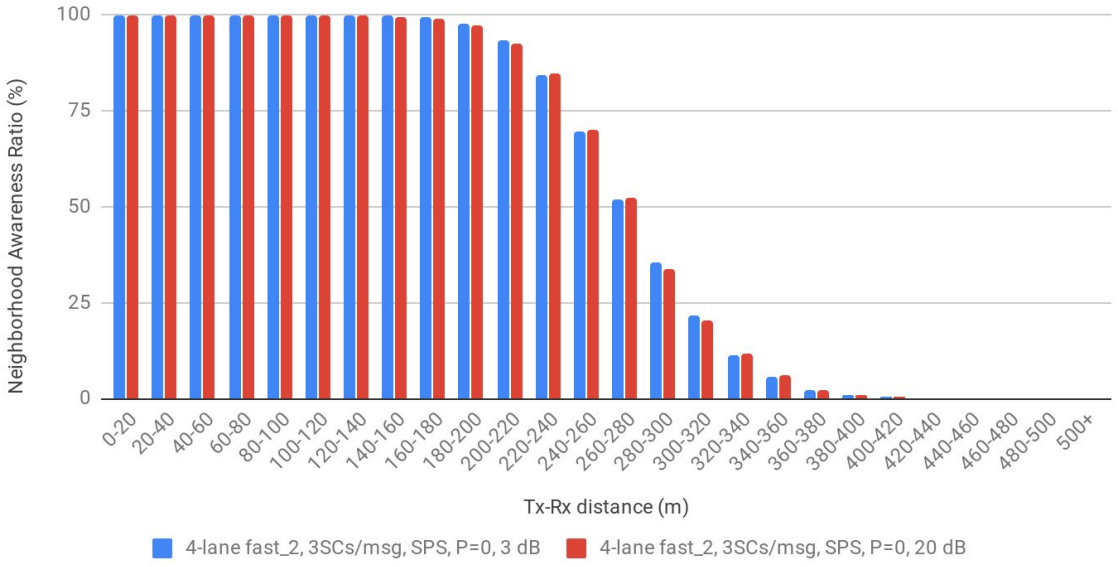

Figure 47. NAR when collision threshold is varied between 3dB and 20 dB (default configuration)

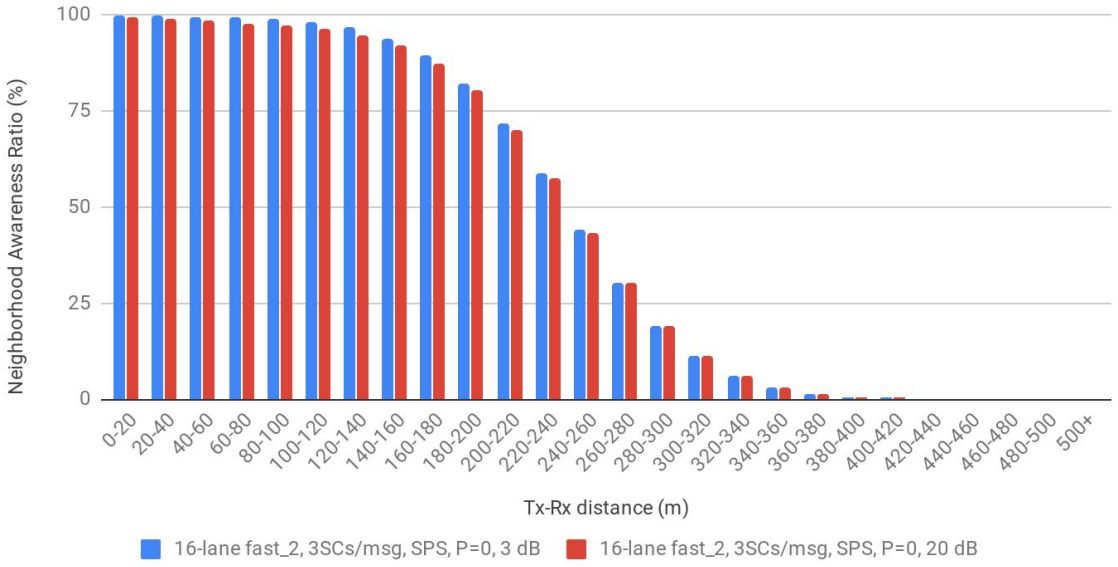

Figure 48. NAR when collision threshold is varied between 3dB and 20 dB (16-lane variant of the default configuration)

On the other hand, Figure 49 shows the effect on the NAR of varying the collision power threshold of the 16-lane, 10 SCs/msg variant of the default configuration, while Figure 50 shows the same, except using the slow vehicle scenarios. This time, the impact of the increase from a 3 dB to a 20 dB collision threshold is much more visible. As in the PRR results, the increase collision power threshold negatively impacts the NAR particularly in the short to mid distances (0 m to 180 m) due to the relative prevalence of collisions in this range.

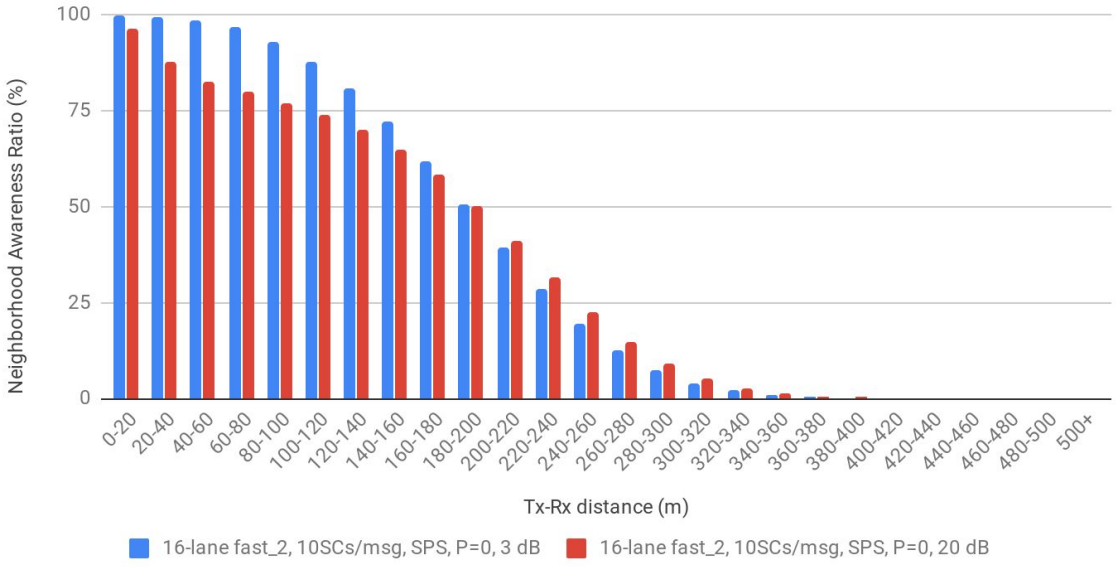

Figure 49. NAR when collision threshold is varied between 3dB and 20 dB (16-lane, 10 SCs/msg variant of the default configuration)

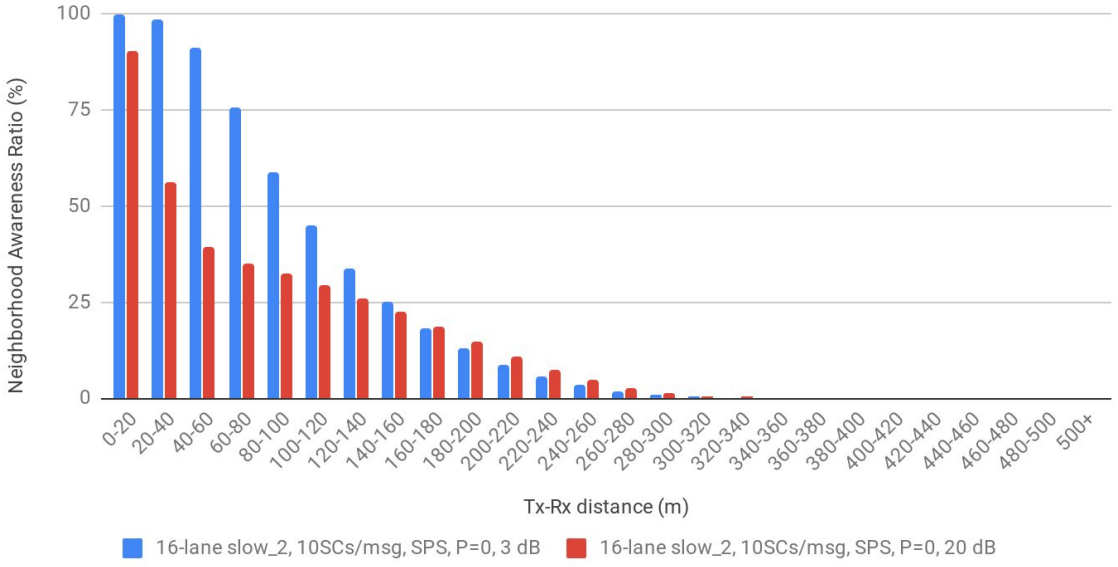

Figure 50. NAR when collision threshold is varied between 3dB and 20 dB (16-lane, slow, 10 SCs/msg variant of the default configuration)

4.3.4. Sensing-based SPS versus Random Allocation

Figure 51 shows the impacts on the NAR when using sensing-based SPS (*P*=0, 0.8, or 1.0) or random allocation variations of the default configuration. Figure 52 shows the same, but on the 16-lane variant.

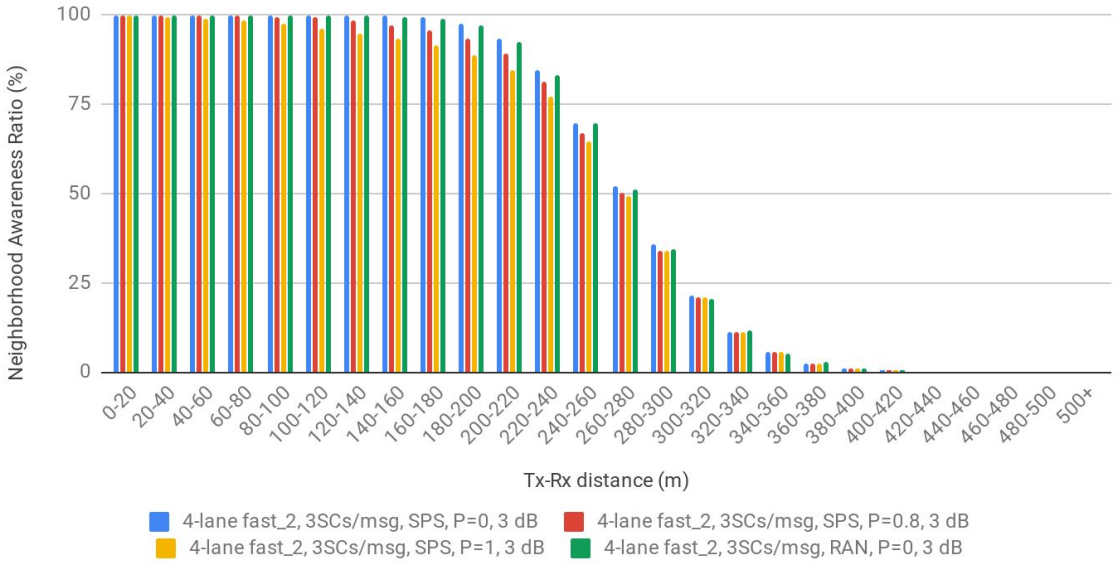

Figure 51. NAR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (default configuration)

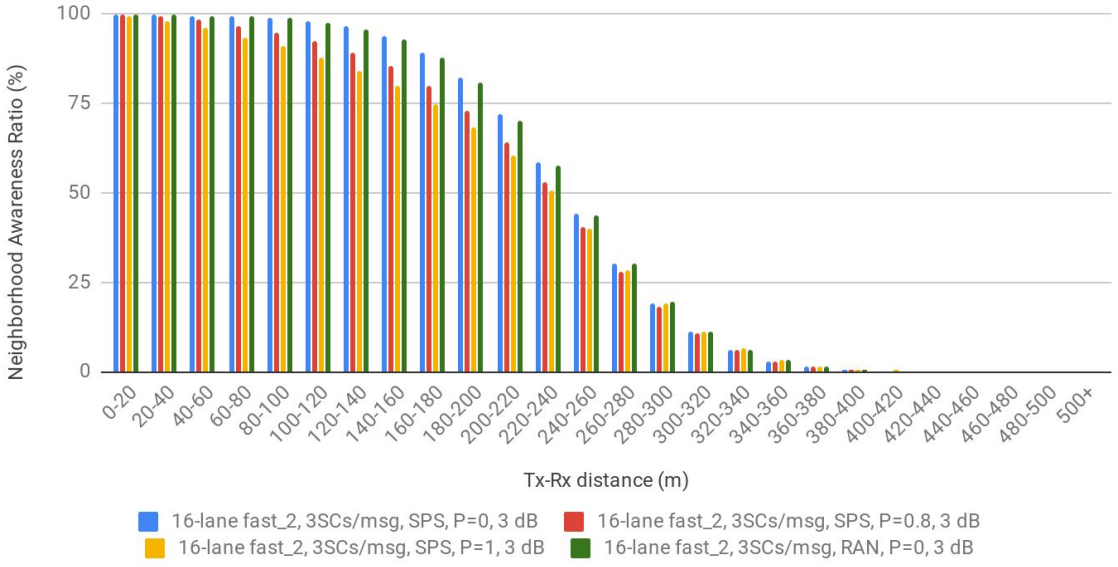

Figure 52. NAR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (16-lane variant of the default configuration)

Looking first at the differences between the keep probabilities when using sensing-based SPS, it could be observed that in both 4-lane and 16-lane scenarios, the NARs of scenarios with lower keep probabilities generally seem to perform better than those with higher keep probabilities, up until around 300 m. This is a reasonable result, because when the keep probabilities are high, there is a tendency for the same resources to have sustained collisions due to their inflexibility in changing resource allocations. On the contrary, when the keep probabilities are lower, there is a higher chance for a node to

escape sustained collision situations and reestablish its presence by being able to select another resource allocation before its entry is deleted from the database.

Also, it could be observed that in both 4-lane and 16-lane scenarios, random allocation performs only slightly worse than sensing-based SPS with *P*=0 until about 300m, and so it follows that random allocation is generally better than sensing-based SPS with *P*=0.8 and 1 at those distances. Beyond 300m however, random allocation becomes no different that any sensing-based SPS scenarios.

Finally, Figure 53 shows the 16-lane scenario, but this time with slow-moving vehicles transmitting with 10SCs/msg. As it is a much denser than the previous 16-lane, 3 SCs/msg, and fast scenario, NAR generally falls off more dramatically, and the surprising thing is that random allocation actually begins to perform better than any of the sensing-based SPS scenarios from about 120m. This means that at sufficiently high collision probability scenarios such as this one, it does not make any more sense to make use of the more complex sensing-based SPS algorithm over random allocation. However, for all distances, *P*=0 always performed significantly better than the other two keep probabilities.

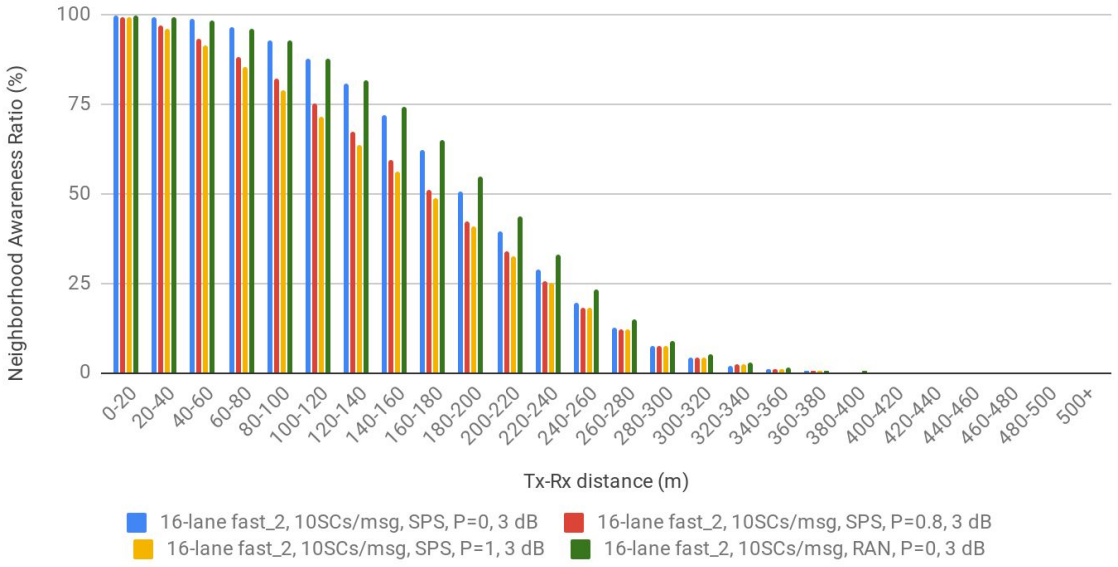

Figure 53. NAR of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (16-lane, 10 SCs/msg variant of the default configuration)

4.4. Position Error

The position error is defined as the difference between the last recorded position of a vehicle and its real position. In the simulator, the last recorded position is obtained by looking at the position information stored in the application layer-level database (Section 3.2.3) and the real position is obtained by inquiring the OMNeT++ simulator about the actual position of the car.

The position error may be a valuable metric of the performance of LTE-V2X in terms of safety and reliability because several V2X use cases require highly accurate positioning. If a vehicle V_1 fails to receive transmissions from a nearby vehicle V_2, it would cause V_1 to

think that V_2 is still in its previously known location, even though in reality, V_2 may have already moved. Knowledge of the position error statistics allows V2X application designers to gain a better understanding on how to make more appropriate decisions in order to avoid potentially fatal mistakes caused by discrepancies between perceived and actual vehicle positions. For example, if a position error of about 5 m can be reasonably expected in a given environment, then the V2X safety application of the vehicle running in this environment must be able to factor in this 5 m uncertainty when making decisions regarding its braking distance with the vehicle in front of it.

The position errors in this section are represented as box (or box and whisker) plots. The bottom whisker of the plot represents the minimum value of the dataset, the top whisker represents the maximum value, the bottom of the box represents the 1st quartile, the top of the box represents the 3rd quartile, and the red line in the middle of the box represents the median. The red dots beyond the whiskers represent statistical outliers, and the relative thickness of the box represents the amount of data used to create that particular box plot. The box plots are classified according to the real distances between the receiver and the transmitter at the time instant that the statistic is measured.

As in the previous sections, the effects of varying the vehicle traffic load (Section 4.4.1), message size (Section 4.4.2), and collision threshold (Section 4.4.3) on the position error when using sensing-based SPS are henceforth discussed. Comparisons between the position errors of sensing-based SPS and random allocation (Section 4.3.4) are also presented.

4.4.1. Impact of Traffic Load

Figure 54 shows the effect on the position error of changing the number of lanes (4, 8, or 16) of the default configuration.

Regardless of the number of lanes, the median position errors are more or less 2 m, until a Tx-Rx distance of 50 m. At distances beyond 50 m however, the median position error begins to increase more dramatically. Moreover, the position error also increases with increasing traffic load, which becomes more significant at larger distances. For instance, between 100 m to 200 m, the median position errors are about 3 m, 4 m, and more than 6 m for the 4, 8, and 16 lane cases, respectively. Note that at Tx-Rx distances between 0-5 m, the box plot is extremely thin, indicating that there are hardly any samples at this range mostly because, knowing that the vehicle itself is 5 m long and a lane is 3.2 m wide, it is quite unlikely that another neighboring vehicle is also within this small range. Also, at Tx-Rx distances beyond 340 m, there are also hardly any samples mainly because transmissions received at such long distances are already quite rare.

Looking at the outliers and the top whiskers, it could be noticed that the maximum position error is always capped at about 60 m, regardless of the scenario. This cutoff is caused by the 20 transmission interval (or 2 s) limit of the application layer-level database before an entry is deleted. Since the vehicles move fast with speeds near the 33.33 m/s maximum, if no transmissions are received for 2 s, leading to an maximum error of about 60 m.

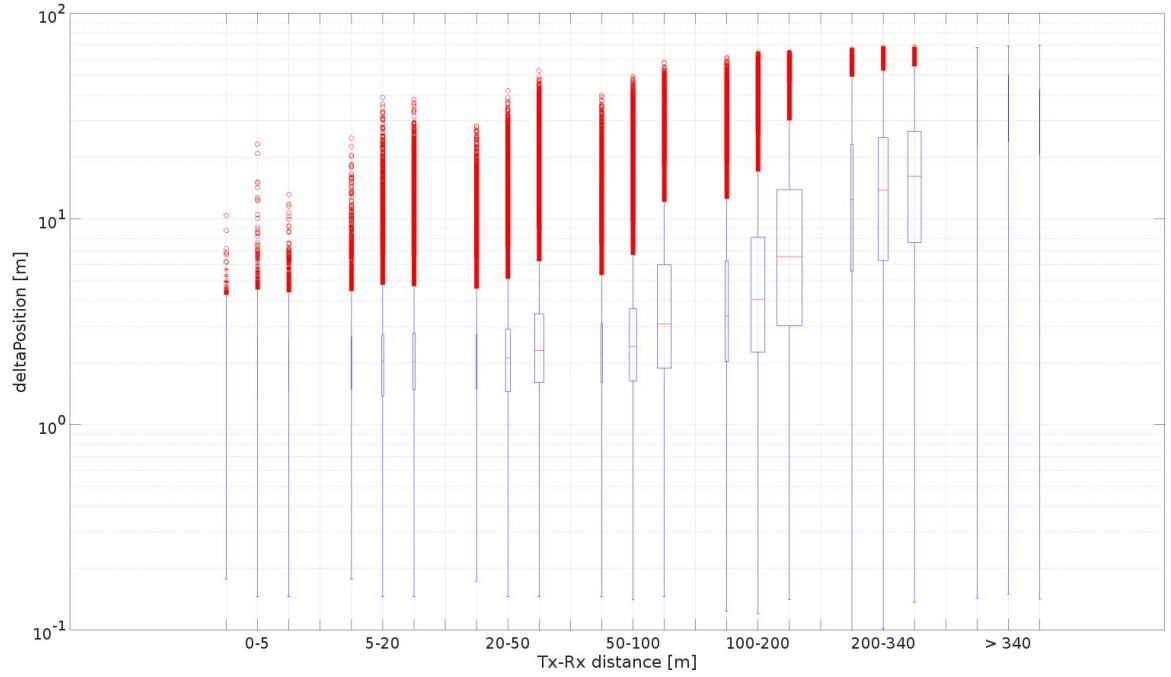

Figure 54. Position error when the number of lanes is varied between 4, 8, and 16 (default configuration)

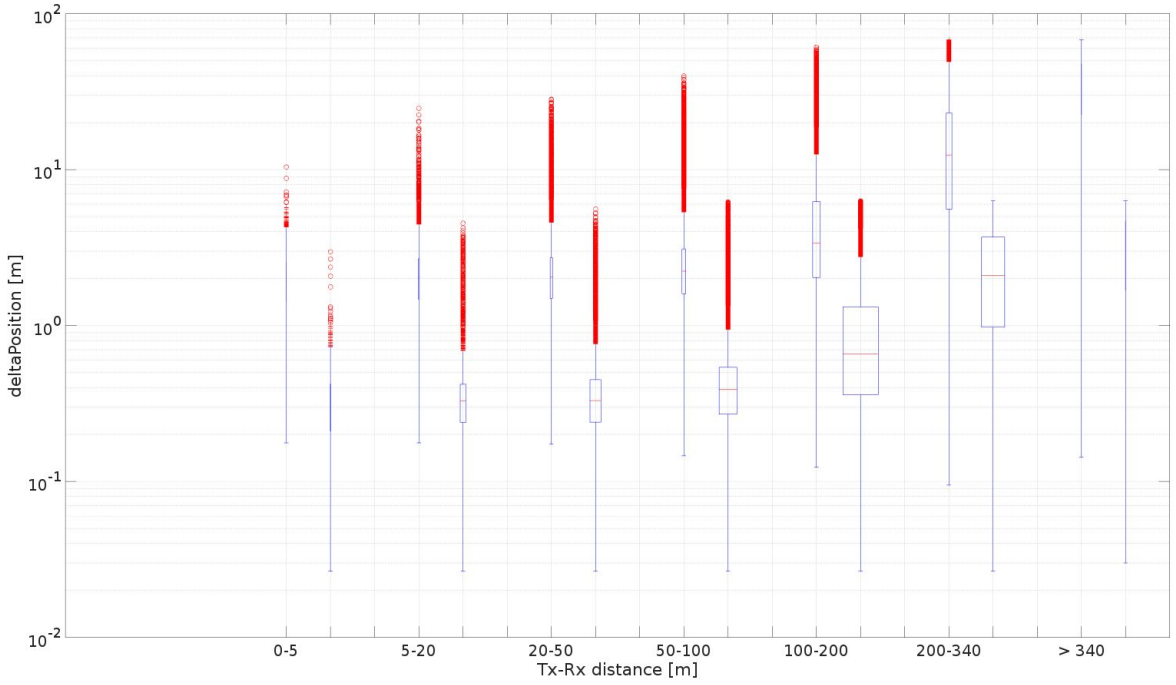

Figure 55. Position error when the speed is varied between fast and slow (default configuration)

Figure 55 shows the impact on the position error of changing the vehicle speed (fast or slow) of the default configuration. Regardless of the Tx-Rx distance, the difference in position error between the fast and slow vehicle scenarios is almost an very significant. For instance, looking at the 50-100 m distances, the median position error in the fast scenario is about 2 m, while the median error in the slow scenario is about 0.4 m. This difference may be explained primarily by the simple fact that the vehicles in the slow

scenario move at a speed that is about one order of magnitude lower than those in the fast scenario (3 m/s vs. 33.33 m/s).

It could be asked that if the speed is roughly one magnitude lower, why is the median position error in the slow scenario not exactly one magnitude lower, i.e. why is the median error for instance at 50 m to 100 m not more or less not 0.2 m instead of 0.4 m? This could be due to the fact that the slow scenario actually has more than double the vehicle density compared to the fast scenario (163 vs. 351 vehicles/km), thus the probability of collisions and interference in the slow scenario should also be higher, thereby causing a slight increase in the median position error.

The outliers and the top whiskers in the slow scenario show that the maximum position error is always capped at about 6 m, regardless of the scenario. Again, this cutoff is caused by the 2 s expiry limit of the application layer-level database. Since the vehicles move slowly with speeds near 3 m/s, then the maximum error is about 6 m if no transmissions are received for 2 s.

4.4.2. Impact of Message Size

Figure 56 shows the impact on the position error of changing the message size (3, 5, or 10 SCs/msg) between fast and slow variants of the default configuration. Figure 57 shows the same, but for the 16-lane variant of the default configuration.

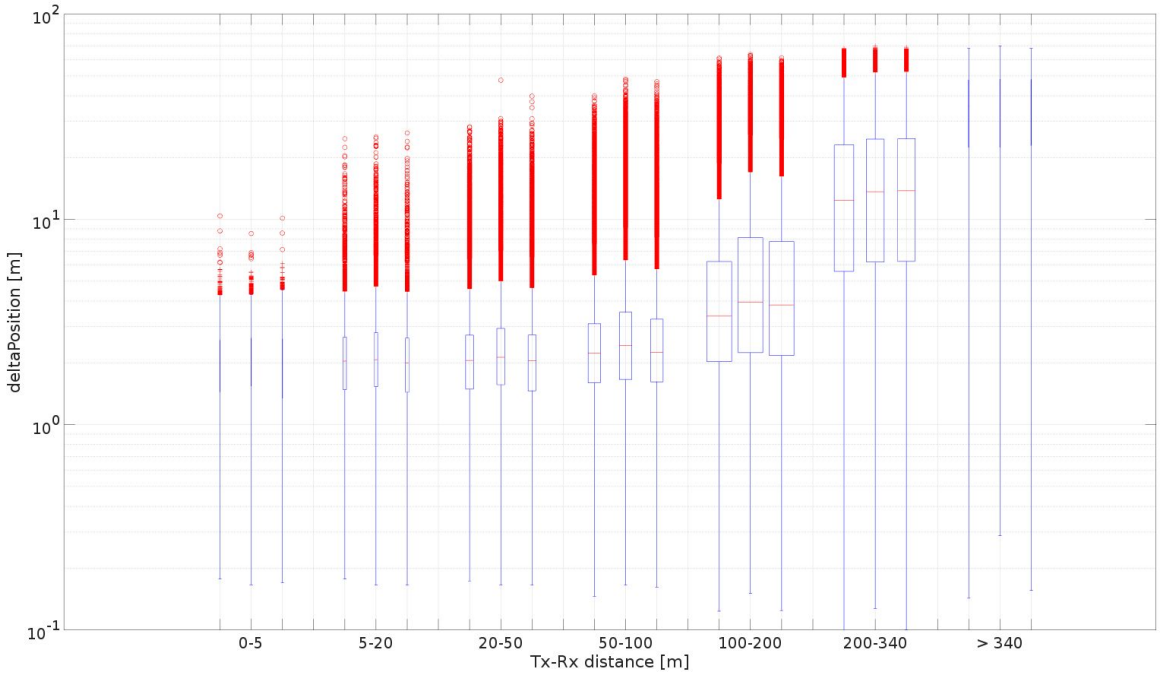

Figure 56. Position error when the message size is varied between 3, 5, and 10 SCs/msg (default configuration)

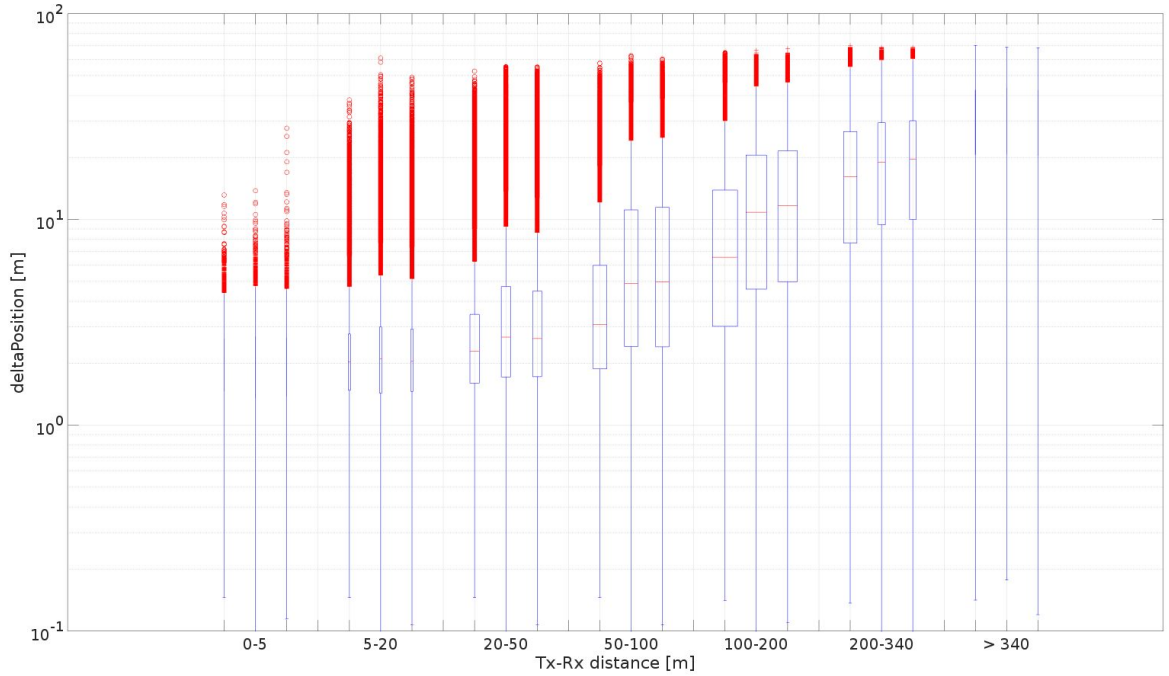

Figure 57. Position error when the message size is varied between 3, 5, and 10 SCs/msg (16-lane variant of default configuration)

In the 4-lane case, the impact of changing the message size on the position error seems to be minimal, however, in the 16-lane case, the impact is more obvious – the larger message size scenarios have slightly worse position errors. For example, looking at the 100 to 200 m distance, the 3 SCs/msg scenario has a median position error of about 5 m, while both the 5 SCs/msg and 10 SCs/msg scenarios both have median position errors slightly above 10 m. Yet again, the 5 SCs/msg and 10 SCs/msg scenarios have little difference in performance, which could be due to the suboptimal resource allocations in the 5 SCs/msg scenario.

4.4.3. Impact of Collision Power Threshold

Figure 58 shows the impact on the position error of changing the collision threshold (3 or 20 dB) of the default configuration. Figure 59 shows the same, but for the 16-lane variant of the default configuration.

Observing closely, it seems that there is hardly any noticeable difference between the two collision threshold scenarios. On the other hand, shows the impact of changing the collision threshold between 3 dB and 20 dB when the vehicles run fast on an 16 bidirectional highway, transmit a message size of 10 SCs/msg, and use sensing-based SPS with *P*=0. Now, the impact of the collision threshold is much more obvious. As it can be seen, the position error is higher in short to medium distances (0 to about 200 m) with higher collision thresholds. This effect is consistent with the PRR and NAR results, which also showed that higher collision thresholds generally have a negative impact in short to medium distances because of the relative dominance of collision-related errors at those distances.

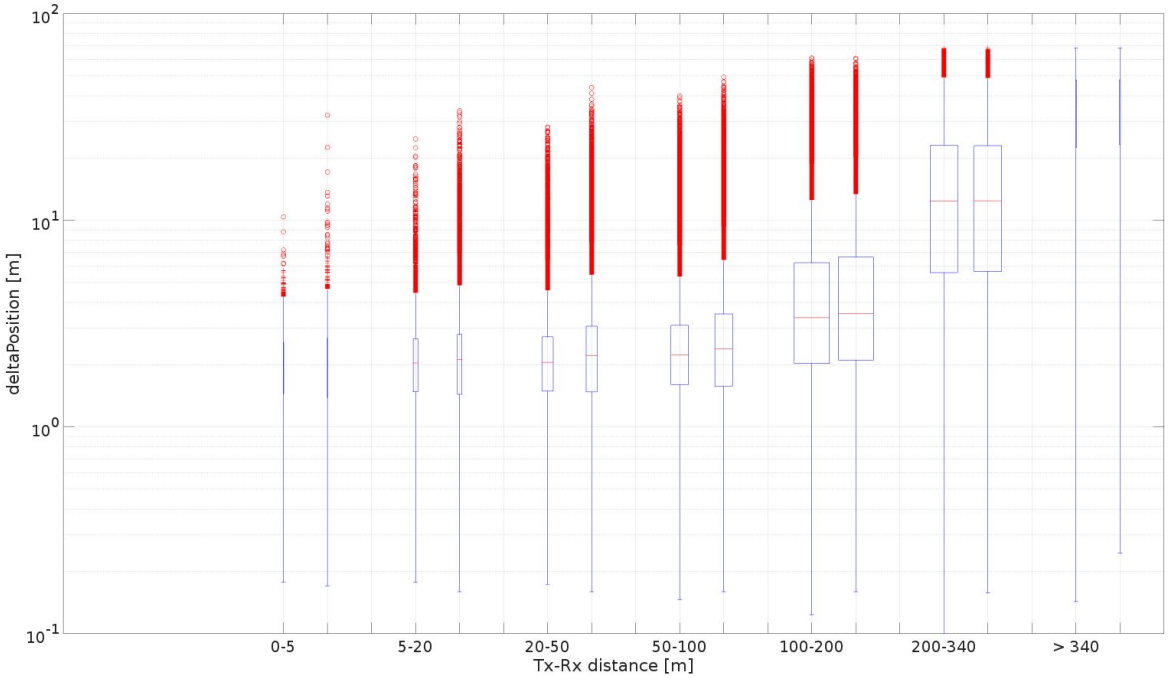

Figure 58. Position error when the collision threshold is varied between 3 dB and 20 dB (default configuration)

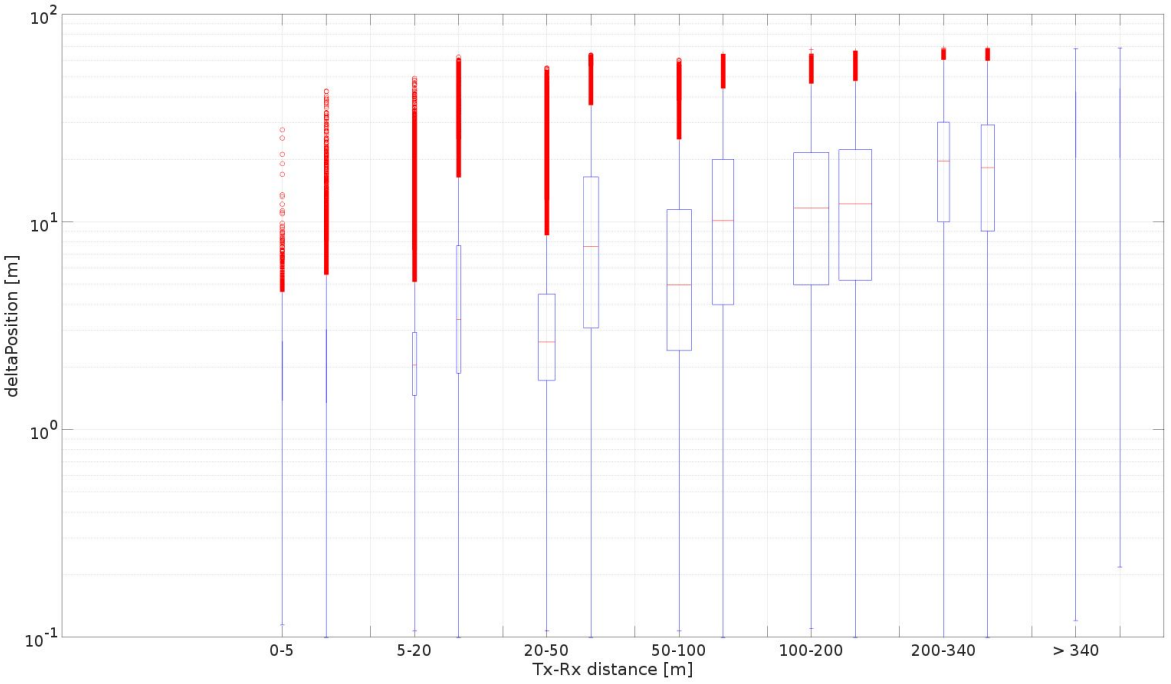

Figure 59. Position error when the collision threshold is varied between 3 dB and 20 dB (16-lane, 10 SCs/msg variant of default configuration)

4.4.4. Sensing-based SPS versus Random Allocation

Figure 60 shows the impacts on the position error when using sensing-based SPS (*P*=0, 0.8, or 1.0) or random allocation variations of the default configuration. Figure 61 shows the same, but on the 16-lane variant.

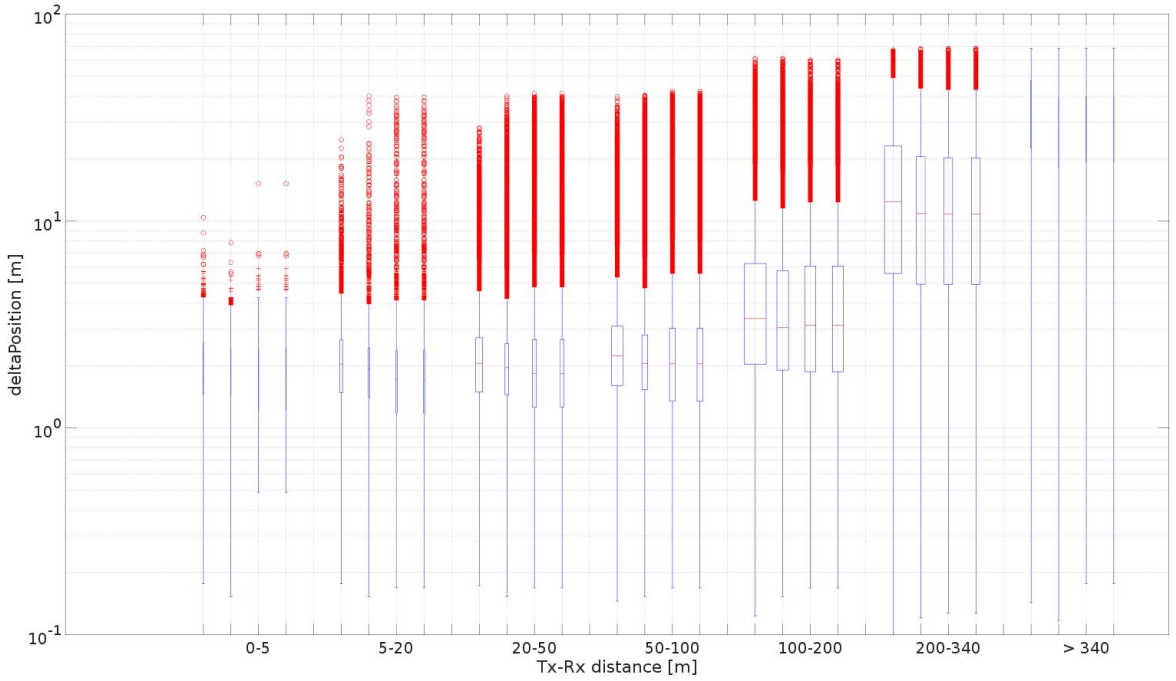

Figure 60. Position error of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (default configuration)

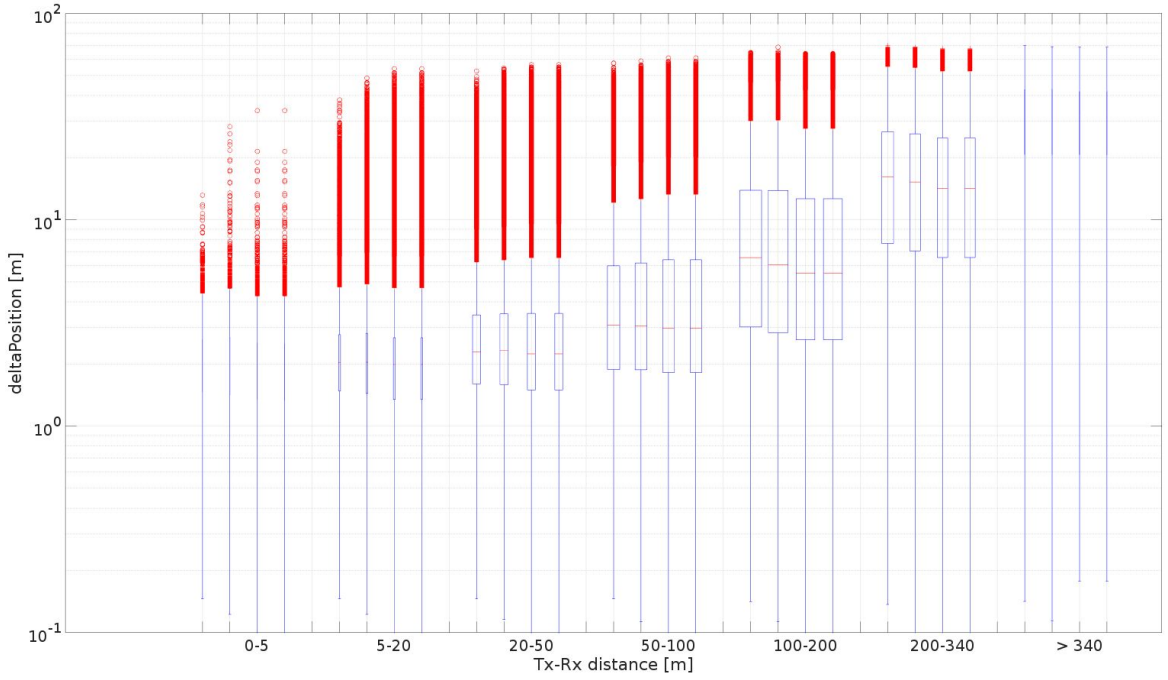

Figure 61. Position error of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (16 lane variation of the default configuration)

Looking first at the 4-lane case, it seems that the *P*=0.8 and 1.0 cases perform almost the same as in random allocation. However, the *P*=0 scenario performs slightly worse compared to the other three scenarios. On the other hand, looking at the 16-lane case, the trends of the different keep probabilities and random allocation begin to show; it

seems that sensing-based SPS with lower keep probabilities perform slightly worse than those with higher keep probabilities, and random allocation performs about the same as the *P*=1 scenario. The trend is more visible at distances beyond 100 m, however, the differences remain small.

Finally, Figure 62 shows the impacts on the position error when using sensing-based SPS (*P*=0, 0.8, or 1.0) or random allocation variations of the 16-lane, 10 SCs/msg variation of the default configuration. Compared to the previous 16-lane, 3 SCs/msg case, the overall position errors are higher due to the increased message size. For instance, the position errors of the *P*=0 and the random allocation scenarios in the 100 - 200 m distances of the previous 3 SCs/msg case are roughly 6.5 m and 5.5 m, respectively, while those of the 10 SCs/msg case are slightly above 10 m and 9 m, respectively.

Regardless, of the scenario, it could be thus said that *P*=0 is generally slightly the worst option for position error, and random allocation performs just as well as the *P*=1 scenarios, although the differences are very small.

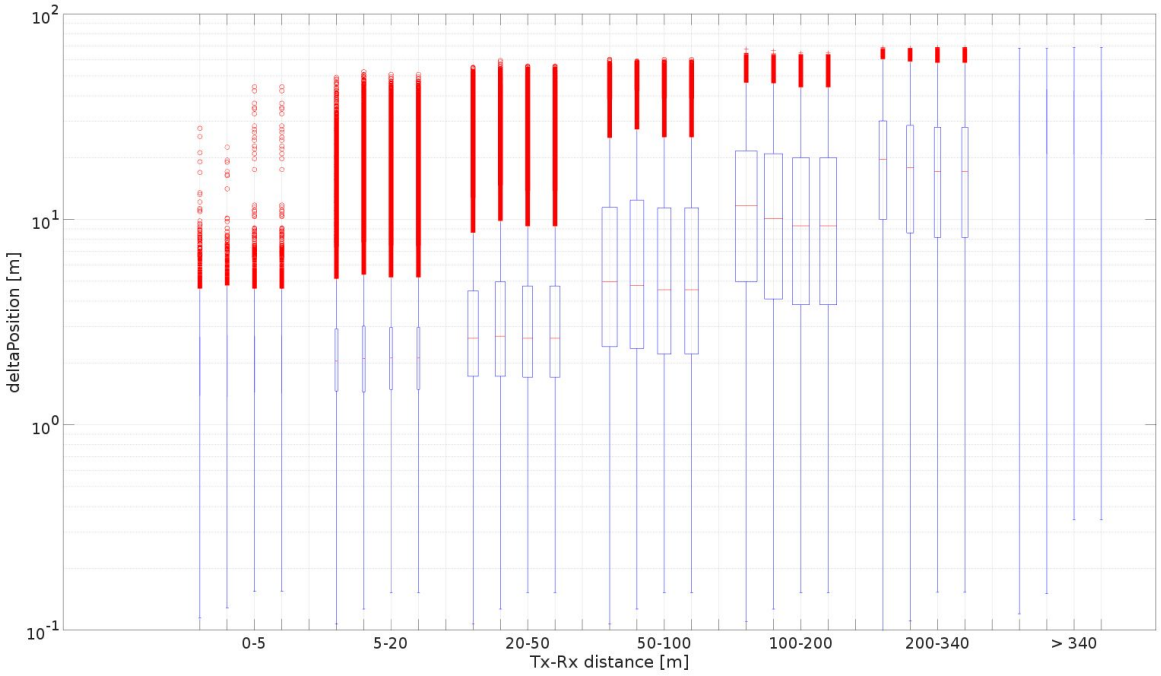

Figure 62. Position error of sensing-based SPS (*P*=0, 0.8, 1.0) vs Random allocation (16 lane, 10 SCs/msg variation of the default configuration)

4.5. Latency

The final statistic, which will only be briefly discussed, is the latency of LTE-V2X Mode 4. Latency is basically the amount of time it takes for a packet of data to travel from the transmitter to the receiver. It is obtained by taking the difference between the time that a transmitted packet was received by the MAC layer and the timestamp within the packet that indicates when the packet was sent down from the MAC layer of the transmitting node.

Figure 63 shows a probability distribution graph of having a certain amount of latency in seconds, averaged across all the scenarios run in the simulation. As it could be seen, the

probability graph approaches a uniform distribution between 1 and 100 ms. This is primarily caused by the resource selection method in LTE-V2X Mode 4, wherein a resource is selected from a pool of candidate resources within a selection window of 100 ms, regardless if it makes use of sensing-based SPS or random allocation. The results shown on Figure 63 therefore acts as a validation that the LTE-V2X Mode 4 simulator was able to satisfy the 100 ms latency requirement of LTE-V2X Release 14 (Section 2.1.2).

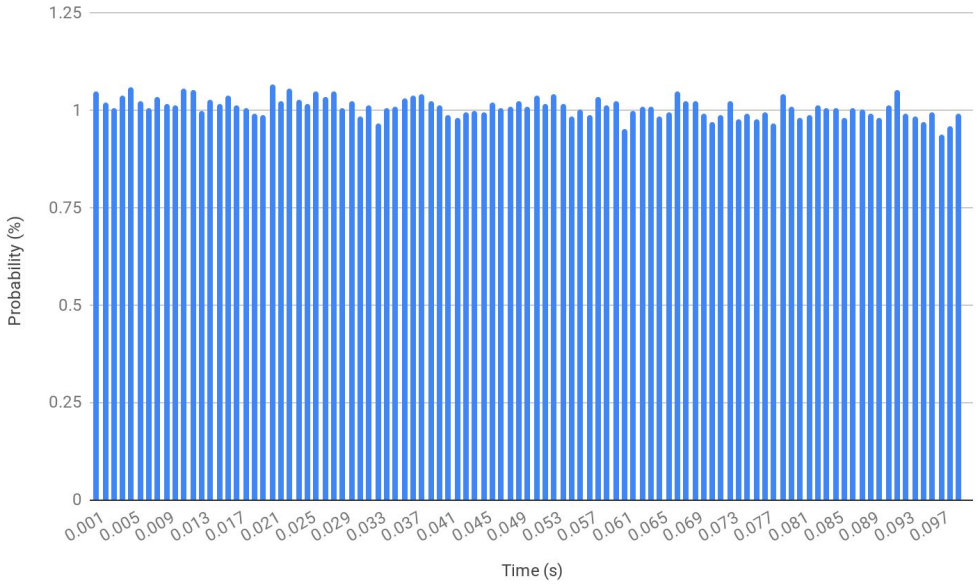

Figure 63. Probability distribution of latency using LTE-V2X Mode 4

5. Budget

There were no other additional financial costs aside from manpower. The hardware that was used was simply a desktop computer (8x Intel Core i7-4790 CPU 3.60GHz, 23.4 GB memory, 1 TB hard disk space) provided by UPC, and all of the software that were used were free and open source (Ubuntu 16.04.4 LTS, OMNeT++ 5.1.1, SUMO 0.30.0, etc.).

Looking instead from the perspective of man-hours, roughly 8 ½ months or 34 weeks were spent on the project from September 2018 to mid-May of 2019 (Section 1.2). Assuming an average of 40 hours done per week by a single researcher, the total labor invested on the project was 1360 man-hours.

6. Conclusions and Future Development

The main parameters that were varied in this simulation in order to evaluate the performance of LTE-V2X Release 14 Mode 4 were the traffic load, the message sizes, the keep probabilities (in case of using sensing-based SPS), and the collision power threshold. The performance was then evaluated from the perspective of the PRR, resource grid occupancy and goodput, NAR, position error, and latency.

The results of the simulations showed that scenarios with higher traffic load had increased message traffic, and therefore more collisions and interference. This then generally led to lower PRRs, lower NARs, and higher position errors. The vehicle speeds have had a large impact on the position errors.

Scenarios with larger message sizes (in terms of subchannels per message) generally had lower PRRs, more occupancy, higher peak goodput, lower NARs, and higher position errors than scenarios with smaller message sizes. It is important to note that, the exact relationships between message size and the performance metrics heavily depends on the resource grid design, for instance, the number of subchannels per subframe. This is due to the fact depending on the number of subchannels per subframe, having certain message sizes may actually create suboptimal resource allocations. It is recommended that an enhancement to the resource allocation of LTE-V2X Mode 4 be investigated in order to alleviate the issue regarding the suboptimal packing of messages in the resource grid.

Higher collision power threshold values lead to worse PRR, NAR, and position errors at short to medium distances, but at medium to long distances, there is no more performance difference between low and high threshold values. This is because higher collision threshold values are harder to satisfy, and because collision errors are problematic only at short to medium distances, whereas at medium to long distances, the signal is already too weak to cope for interference, regardless of collisions. Increasing the collision threshold leads to a compounding of the negative impact of collision probability. Scenarios with higher collision threshold values also cause the goodput values to fall earlier at a given message traffic density than those with lower collision threshold values. However, higher collision threshold values also unexpectedly caused a higher occupancy values, which can be investigated in further studies.

Varying the keep probabilities between the standard values of P=0 and 0.8 did not show to have a significant impact in PRR, although using the nonstandard value of 1 generally led to noticeably worse PRRs. Scenarios with lower keep probabilities had better NARs than those with higher keep probabilities. On the other hand, those with lower keep probabilities showed higher resource grid occupancy and goodput values, as well as slightly better position error performances.

In terms of PRR, sensing-based SPS generally performed better than random allocation. In terms of NAR, only sensing-based SPS configured with P=0 had a very slight edge over random allocation, while P=0.8 and 1 showed worse results than random allocation. In environments with sufficiently high message traffic, the PRR and NAR of sensing-based SPS performed no better, or even worse than random allocation. In terms of position error, sensing-based SPS with P=0 had a small disadvantage compared to random allocation, but the other configurations had only much slighter differences with

respect to random allocation. Sensing-based SPS generally with P=0.8 had a similar occupancy and goodput with random allocation, while sensing-based SPS with P=0 generally had worse occupancy and goodput than random allocation. The unexpected findings regarding the relationship of the resource grid occupancy and goodput statistics with the usage of sensing-based SPS with varying keep probabilities or random allocation merits more attention in future research work.

Lastly, the latency values of LTE-V2X Release 14 Mode 4 tended to form a uniform distribution between 1 and 100 ms, due to the 100 ms resource selection window regardless if sensing-based SPS or random allocation was used. This was within the 100 ms maximum latency requirement set for LTE-V2X Release 14.

There are several possible suggestions for those who wish to continue evaluating the performance of LTE-V2X Release 14 Mode 4. Much more simulations could be run for longer durations of time in order to establish more statistically sound relationships between the parameters and performance indicators. The performances using the unidirectional highway and Manhattan grid as the main simulation environments could be investigated more in-depth. The occupancy and goodput in other message traffic densities could also be investigated in order to increase the number of datapoints. The impact of varying the characteristics of the physical layer and radio propagation models might also be interesting to research. The performance in scenarios with much more complicated traffic mixes, wherein different cars travel at different speeds, transmit different message sizes at different time intervals, and use different configurations of the sensing-based SPS, etc. may also be evaluated.

For those who wish to improve the simulator itself, it could be recommended to incorporate the other features of LTE-V2X Release 14 that were intentionally not developed for this project, such as HARQ retransmissions, zone-based resource grid separation, or the adjacent SCI and TB resource grid scheme. It might also be worth integrating IEEE 802.11p functionality in order to make it possible to make a side-by-side comparison between the performance of IEEE 802.11p and LTE-V2X Release 14 using the same simulation environment. Mode 3 communications may also be included, possibly with the assistance of other simulation frameworks such as SimuLTE. It would also be interesting to extend the simulator to include the changes from the upcoming Release 15, especially since some of the enhancements in Release 15 will not be compatible with Release 14. A physical layer model that better emulates the performance of SC-FDMA could be developed and integrated into the simulator.

Finally, for those who wish to take advantage of the later versions of the INET framework (i.e. INET 4.0 and beyond), it might actually be better to simply start from scratch again and build an entirely new LTE-V2X simulator because the changes required to shift from earlier versions of INET to the later versions would require a nontrivial amount of work due to the fundamental changes in the packet API, network node architecture, etc. Nevertheless, making use of later versions of the simulation frameworks would future-proof the software in terms of support and more functionality.

Bibliography

[1] 3GPP. “Initial Cellular V2X standard completed”. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1798-v2x_r14. [Accessed: April 30, 2019].

[2] Qualcomm. “Let’s set the record straight on C-V2X” [Online]. Available: https://www.qualcomm.com/news/onq/2018/04/25/lets-set-record-straight-c-v2x. [Accessed: April 23, 2019].

[3] 3GPP TR 22.885 V14.0.0 (2015-12), “Study on LTE support for Vehicle to Everything (V2X) services (Release 14)”.

[4] 3GPP TS 22.185 V14.3.0 (2017-03), “Service requirements for V2X services; Stage 1”

[5] 3GPP TS 36.300 V14.3.0 (2017-06), “Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2”

[6] 3GPP TS 36.331 V14.3.0 (2017-06), “Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification”

[7] R. Molina-Masegosa and J. Gozalvez, "System Level Evaluation of LTE-V2V Mode 4 Communications and Its Distributed Scheduling," 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, Australia, 2017, pp. 1-5.

[8] S. Chen et al., "Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G," in IEEE Communications Standards Magazine, vol. 1, no. 2, pp. 70-76, 2017.

[9] European Telecommunications Standards Institute (ETSI),“Physical layer procedures (Release 14),” 3GPP TS 36.213,V14.2.0, Apr.2017.

[10] European Telecommunications Standards Institute (ETSI),“Medium Access Control (MAC) protocol specification (Release 14),” 3GPP TS36.321, V14.2.1, May 2017.

[11] 3GPP TR 22.886, “Study on enhancement of 3GPP support for 5G V2X services” (v15.0.0, Release 15)

[12] 3GPP MCC Support, “Draft Report of 3GPP TSG RAN WG1 #89 v0.2.0,” 3GPP TSG RAN WG1 Meeting #89, Hangzhou, China, May. 2017.

[13] Ericsson, “R1-1708942. Radio resource pool sharing between mode 3 and mode 4 UEs,” 3GPP TSG RAN WG1 Meeting #89, Hangzhou, China, May. 2017

[14] A.Turley, K. Moerman, A. Filippi, V. Martinez “C-ITS: Three observations on LTE-V2X and ETSI ITS-G5—A comparison” [Online]. Available: https://www.nxp.com/docs/en/white-paper/CITSCOMPWP.pdf. [Accessed: April 23, 2019]

[15] R. Molina-Masegosa, J. Gozalvez, and M. Sepulcre, “Configuration of the C-V2X Mode 4 Sidelink PC5 Interface for Vehicular Communications,” in 14th Conference on Mobile Ad-hoc and Sensor Networks (MSN 2018), 2018.

[16] A. Bazzi, G. Cecchini, B. M. Masini, and A. Zanella, “Study of the Impact of PHY and MAC Parameters in 3GPP C-V2V Mode 4,” in arXiv:1807.10699v1 [cs.NI], 2018

[17] G. Cecchini, A. Bazzi, B. M. Masini and A. Zanella, "LTEV2Vsim: An LTE-V2V simulator for the investigation of resource allocation for cooperative awareness," 2017 5th

IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Naples, 2017, pp. 80-85.

[18] A. Nabil, V. Marojevic, K. Kaur, and C. Dietrich. “Performance Analysis of Sensing-Based Semi-Persistent Scheduling in C-V2X Networks”. Wireless@Virginia Tech, Bradley Dept. Electrical and Computer Engineering

[19] Jeon, Y.; Kuk, S.; Kim, H. “Reducing Message Collisions in Sensing-Based Semi-Persistent Scheduling (SPS) by Using Reselection Lookaheads in Cellular V2X”. *Sensors* 2018, *18*, 4388.

[20] OMNeT++. [Online]. Available: https://omnetpp.org/. [Accessed: April 20, 2019].

[21] A. Varga, R. Hornig, “An overview of the OMNeT++ simulation environment”, in Proc. SIMUTools '08, Marseille, FR, March 2008.

[22] A. Varga, “OMNeT++ User Guide” Available: https://doc.omnetpp.org/omnetpp/UserGuide.pdf. [Accessed: April 20, 2019].

[23] “What is INET Framework?” [Online]. Available: https://inet.omnetpp.org/Introduction.html. [Accessed: April 20, 2019].

[24] SUMO [Online]. Available: https://sumo.dlr.de/. [Accessed: April 20, 2019].

[25] Veins [Online]. Available: https://veins.car2x.org/documentation/ [Accessed: April 20, 2019].

[26] Simulte. [Online]. Available: http://simulte.com/ [Accessed: April 30, 2019]

[27] LTE Toolbox [Online]. Available: https://www.mathworks.com/products/lte.html. [Accessed: April 30, 2019].

[28] Traci4matlab. [Online]. Available: https://es.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab. [Accessed: April 20, 2019].

[29] P. Yue, L. Wu and Z. Cui, "Integrating LTE-D2D and VLC techniques to support V2V communication," 2017 IEEE/CIC International Conference on Communications in China (ICCC), Qingdao, 2017, pp. 1-6.

[30] NS-3 User Documentation. [Online]. Available: https://www.nsnam.org/docs/models/html/lte-user.html.

[31] ArteryLTE. [Online]. Available: https://github.com/ibr-cm/artery-lte [Accessed: April 20, 2019].

[32] INET framework. [Online]. Available: https://inet.omnetpp.org/. [Accessed: April 30, 2019].

[33] L. Lopez, C. Mendoza, J. Casademont-Serra, and D. Camps-Mur. Understanding the Impact of the PC5 Resource Grid Design on the Capacity and Efficiency of LTE-V2X. Submitted to IEEE Vehicular Technology Magazine (under revision).

Glossary

3GPP - 3rd Generation Partnership Project

API - Application Programming Interface

BGP - Border Gateway Protocol

BPSK - Binary Phase Shift Keying

CAM - Cooperative Awareness Message

CPU - Central Processing Unit

CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance

CSV - Comma Separated Values

C-V2X - Cellular V2X

D2D - Device-to-Device

DCI - Downlink Control Information

DMRS - DeModulation Reference Signal

DSRC - Dedicated Short-Range Communications

ECU - Electronic Control Unit

eNB - EUTRAN Node B

EPC - Evolved Packet Core

ETSI - European Telecommunication Standards Institute

EUTRAN - Evolved Terrestrial Radio Access Network

HARQ - Hybrid Automatic Repeat reQuest

IDE - Integrated Development Environment

IE - Information Element

IEEE - Institute of Electrical and Electronics Engineers

IP - Internet Protocol

LENA - LTE-EPC Network simulAtor

LOS - Line-Of-Sight

LTE - Long Term Evolution

LTS - Long Term Support

MAC - Medium Access Control

MATLAB - MATrix LABoratory

MIB - Master Information Block

OFDM - Orthogonal Frequency Division Multiplexing

OFDMA - Orthogonal Frequency-Division Multiple Access

OMNeT++ - Objective Modular Network Testbed in C++

OSPF - Open Shortest Path First

PDCCH - Physical Downlink Control Channel

PDR - Packet Delivery Ratio

PDSCH - Physical Downlink Shared Channel

PPP - Point-to-Point Protocol

PRR - Packet Reception Ratio

PSBCH - Physical Sidelink Broadcast Channel

PSCCH - Physical Sidelink Control Channel

PSSCH - Physical Sidelink Shared Channel

QAM - Quadrature Amplitude Modulation

QPSK - Quadrature Phase-Shift Keying

RAT - Radio Access Technologies

RB - Resource Block

RC - Resource Counter

RLC - Radio Link Control

RSSI - Received Signal Strength Indicator

RSU - Roadside Unit

SBCCH - Sidelink Broadcast Control Channel

SC-FDMA - Single Carrier Frequency Division Multiple Access

SCI - Sidelink Control Information

SIB - System Information Block

SimuLTE - Simulator for LTE Networks

SL-BCH - Sidelink Broadcast Channel

SL-SCH - Sidelink Shared Channel

SPS - Semi-Persistent Scheduling

S-RSSI - Sidelink RSSI

SSMS - Sensor and State Map Sharing

STCH - Sidelink Traffic CHannel

SUMO - Simulation of Urban MObility

TB - Transport Block

TCP - Transmission Control Protocol

TraCI - Traffic Control Interface

TTI - Transmission Time Interval

UDP - User Datagram Protocol

UE - User Equipment

UPC - Universitat Politècnica de Catalunya

V2I - Vehicle-to-Infrastructure

V2N - Vehicle-to-Network

V2P - Vehicle-to-Pedestrian

V2V - Vehicle-to-Vehicle

V2X - Vehicle-to-Everything

Veins - Vehicles in network simulation

VLC - Visible Light Communication

VRU - Vulnerable Road User

WLAN - Wireless Local Area Network

XML - Extensible Markup Language

Appendices

A. OMNeT++ 5.1.1 Installation procedure

The following steps describe the installation of the OMNeT++ Discrete Event Simulator version 5.1.1 for an Ubuntu-based system.

Prior to installation of OMNeT++, a slew of prerequisite packages are required. To install these packages, run the following line in the Linux terminal:

```
sudo apt-get install build-essential gcc g++ bison flex perl tcl-dev tk-dev blt
libxml2-dev zlib1g-dev default-jre doxygen graphviz libwebkitgtk-1.0-0 openmpi-bin
libopenmpi-dev libpcap-dev autoconf automake libtool libproj-dev libgdal1-dev
libfox-1.6-dev libgdal-dev libxerces-c-dev qt4-dev-tools
```

Download `omnetpp-5.1.1-src-linux.tgz` from the OMNeT++ website [20]. Save in a directory of your choice, for example, in the user folder `v2x`.

```
v2x$ tar xvfz omnetpp-5.1.1-src-linux.tgz

v2x$ cd omnetpp-5.1.1

v2x:~/omnetpp-5.1.1$ . setenv
```

Run the following commands to configure and compile OMNeT++:

```
v2x:~/omnetpp-5.1.1$ ./configure

v2x:~/omnetpp-5.1.1$ make
```

Verify that the compilation completed without errors.

In order to be able to run OMNeT++, it is necessary to add OMNeT++ executable to the system path. To do this, add the following in `PATH` line in the file `/home/v2x/.profile` (note that this may be a hidden file, so it may be necessary to show hidden files in the folder):

```
/home/v2x/omnetpp-5.1.1/bin
```

After inserting the line, the PATH may look something like this:

```
PATH="$HOME/bin:$HOME/.local/bin:/home/v2x/omnetpp-5.1.1/bin:$HOME/.local/bin:/home
/v2x/sumo-0.30.0/bin:$PATH"
```

After setting the `PATH`, you may need to close and re-open the terminal for the changes to take effect). If the `PATH` configuration was successful, all that is needed to run the program is to type the following in the command line:

```
v2x$ omnetpp
```

If the `PATH` configuration was unsuccessful, it is still possible to run OMNeT++ simply straight from its installation directory:

```
v2x$ cd omnetpp-5.1.1/bin/

v2x:~/omnetpp-5.1.1/bin$ omnetpp
```

Upon opening OMNeT++, a pop-up window might appear asking you to install the INET framework. Do not install INET in this manner as it will install the newest version of INET (4.1 as of this time of writing), which will be incompatible with the LTE-V2X Mode 4 simulation.

B. INET and Veins Installation Procedure

Download INET framework version 3.6.0 from the INET website [32] and Veins version 4.6 from the Veins website [25]. Extract the packages into a directory of your choice.

Start the OMNeT++ IDE, then import the project by clicking on *File* > *Import* > *General* > *Existing* projects into the workspace. Select the root directory where INET and Veins are located. Make sure that the option *Search for nested projects* is ticked. It is optional to copy the projects into your workspace. Click Finish.

Build the projects by pressing CTRL+B, or by going to *Project* > *Build all*.

C. LTE-V2X Release 14 Mode 4 Simulation Module Installation Procedure

Obtain a copy of the LTE-V2X Release 14 Mode 4 Simulation Module from the i2CAT Foundation.

To install, open the OMNeT++ IDE, then create a new project using *File* > *New* > *OMNeT++ Project*. The project name is up to you. Also, make sure that the project is an empty project.

Copy the LTE-V2X Release 14 Mode 4 files into the new project. This can be done simply by dragging the files and then dropping them into the project in the OMNeT++ IDE.

When the files have been added, right-click on the project folder, then go to *Project References*. Tick the boxes for the *inet*, *veins*, and *veins_inet* projects, then click OK.

Build the project by pressing CTRL+B, or by going to *Project* > *Build all*.

Finally, check out the `readme.txt` file to see if there are any additional steps that need to be done.

D. SUMO 0.30.0 Installation Procedure

Download `sumo-src-0.30.0.tar.gz` from the SUMO website [24]. Save the file in your a directory of your choice, for example, in the user folder `v2x`.

```
v2x$ tar xvfz sumo-src-0.30.0.tar.gz

v2x$ cd sumo-0.30.0
```

Run the following commands to configure and compile SUMO:

```
v2x/sumo-0.30.0$ ./configure

v2x/sumo-0.30.0$ make
```

Create a script file called `sumoexec.cmd` using any text editor and placed in a directory of your choice. Open the empty script file and add the following lines. Note that the second line should refer to the `sumo-launchd.py` file which should be located in the directory where Veins was installed:

```
clear

/home/v2x/simulte_veins/veins-veins-4.6/sumo-launchd.py -vv -c
/home/v2x/sumo-0.30.0/bin/sumo

sleep 1

echo ""
```

Alternatively, a script for running the GUI version of SUMO can be made. To distinguish this from the previous file, name this `sumoexec-gui.cmd`

```
clear

/home/v2x/simulte_veins/veins-veins-4.6/sumo-launchd.py -vv -c
/home/v2x/sumo-0.30.0/bin/sumo-gui

sleep 1

echo ""
```

Now, whenever you wish to run the LTE-V2X Mode 4 simulator in OMNeT++, you need to run the above script in order to execute SUMO in parallel with the OMNeT++ simulation. Do this by entering this in the command line:

```
sh ./sumoexec.cmd
```

Or alternatively, you can run:

```
sh ./sumoexec-gui.cmd
```

E. The Configuration File `omnetpp.ini`

The `omnetpp.ini` configuration file is a convenient way of changing the parameters of the OMNeT++ simulation. The file should be located on the LTE-V2X Release 14 Mode 4 project folder.

This appendix section aims to give a quick guide on some of the more relevant parameters that are available on the `omnetpp.ini` file of the LTE-V2X Release 14 Mode 4 simulation module.

E.1 General Settings

- **`sim-time-limit`**: specifies the length of the simulation.
- **`warmup-period`**: specifies the initial period of time for which no statistics are recorded.

E.2 Veins Manager Settings

- **`*.manager.updateInterval`**: specifies the time interval between updates of SUMO.
- **`*.manager.firstStepAt`**: specifies when to start synchronizing with TraCI.
- **`*.manager.launchConfig`:** specifies the file name of the XML file that contains the launch configuration data for SUMO.

E.3 Application Layer settings

- **`*.node[*].numUdpApps`:** specifies the number of UDP apps that are simultaneously running on the node.
- **`*.node[*].udpApp[*].typename`**: specifies the type of UDP application.
- **`*.node[*].udpApp[*].messageLength`:** specifies the length of the message in bytes. Note that the current implementation of the LTE-V2X Mode 4 simulator is independent from the UDP message size, i.e. the UDP message length will not affect how many subchannels or RBs that are taken in the resource grid.
- **`*.node[*].udpApp[*].startTime`:** specifies the starting time when the UDP application becomes active and starts sending messages.
- **`*.node[*].udpApp[*].sendInterval`:** specifies the time interval of sending UDP messages.
- **`*.node[*].udpApp[*].destAddresses`:** specifies the destination IP addresses of the UDP messages.
- **`*.node[*].udpApp[*].joinLocalMulticastGroups`:** specifies whether to join a multicast group (true) or not (false).

E.4 Radio Medium Settings

- **`*.radioMedium.obstacleLossType`:** specifies how signal power is lost as it passes through physical objects. INET provides `"IdealObstacleLoss"` that determines total power loss when there are obstructions, or alternatively `"DielectricObstacleLoss"` that computes power loss based on the material and shape of the obstructing object.
- **`*.radioMedium.propagationType`:** specifies the manner in which the signal moves through space over time. INET provides a relatively simple `"ConstantTimePropagation"` model or alternatively, a more realistic `"ConstantSpeedPropagation"` model.
- **`*.radioMedium.pathLossType`:** specifies the path loss model of the propagation. Some of the readily-available types provided by INET include: `"FreeSpacePathLoss"`,`"BreakpointPathLoss"`,`"LogNormalShadowing"`,`"TwoRayGroundReflection"`,`"TwoRayInterference"`,`"RicianFading"`,`"RayleighFading"`, and `"NakagamiFading"`.
- **`*.radioMedium.backgroundNoiseType`:** specifies the background noise type. INET provides `IsotropicScalarBackgroundNoise`, which is a background noise type that is independent of space-time coordinates.
- **`*.radioMedium.backgroundNoise.power`:** specifies the background noise power of the noise type selected in `backgroundNoiseType`.
- **`*.radioMedium.mediumLimitCache.carrierFrequency`**: specifies the carrier frequency of the signals.
- **`*.radioMedium.pathLoss.alpha`:** specifies the alpha parameter or the path loss gradient. Typical values are 2.0 for free space, 2.7 to 3.5 for urban areas, 3 to 5 for suburban areas, and 1.6 to 1.8 for indoor (LOS) areas.

E.5 Node Radio Settings

- **`*.node[*].numRadios`**: specifies the number of radio components (transmitter/ receiver) of the node.
- **`*.node[*].wlan[*].bitrate`:** specifies the bit rate. Note that in the current implementation of the LTE-V2X Mode 4 simulation, this parameter does not model the LTE bit rates, so it just needs to be set to a high value to ensure that the messages are delivered within the duration of a subframe.
- **`*.node[*].wlan[*].radio.carrierFrequency`:** specifies the carrier frequency in Hz.
- **`*.node[*].wlan[*].radio.typename`:** specifies the type of radio. INET provides the generic types `"UnitDiskRadio"`, which is a simple and fast model, and `"ApskScalarRadio"`, which models PSK and QAM modulations.
- **`*.node[*].wlan[*].radio.bandwidth`:** specifies the channel bandwidth in Hz. Note that in the current implementation of the LTE-V2X Mode 4 simulation, this parameter is independent from the resource grid, i.e. setting the radio bandwidth to 10 MHz for instance will not set the number of RBs of the resource grid to 50. The size of the resource grid has to be set manually in the MAC settings.

- **`*.node[*].wlan[*].radio.transmitter.power`:** specifies the transmitter power.
- **`*.node[*].wlan[*].radio.receiver.sensitivity`:** specifies the receiver sensitivity, i.e. if the signal power is below the sensitivity, reception is not possible, that is, the receiver cannot go from the channel busy state to receiving.
- **`*.node[*].wlan[*].radio.receiver.energyDetection`:** specifies the energy detection threshold. If the reception power is below this threshold, it is considered that no signal is detected and the channel is empty.
- **`*.node[*].wlan[*].radio.receiver.snirThreshold`:** specifies the Signal to Noise plus Interference threshold. The reception is considered to be unsuccessful if the SNIR is below this threshold.

E.6 Node MAC Settings

- **`*.node[*].wlan[*].mac.spsCandidateRatioMin`**: specifies the minimum ratio of candidate resources over total resources before proceeding to the selection process. For example, if `spsCandidateRatioMin` is 0.2 and the ratio of candidate over total resources is 18%, then the SPS power threshold is increased by 3dB and the two-step exclusion process is repeated.
- **`*.node[*].wlan[*].mac.spsSelectionWindow`:** specifies the size of the sensing-based SPS resource selection in terms of subframes. Note that the maximum value according to 3GPP specifications is 100.
- **`*.node[*].wlan[*].mac.spsMaxCounter`**: specifies the maximum RC value.
- **`*.node[*].wlan[*].mac.spsMinCounter`**: specifies the minimum RC value.
- **`*.node[*].wlan[*].mac.spsMemoryLimit`**: specifies the number of consecutive message intervals that are skipped before removing the transmitting node from the memory of the receiving node. For example, given that the memory limit is set to 20 and the message interval is 0.1s, suppose 2 seconds have passed but Node 0 has not received any messages from Node 1, then Node 0 considers that Node 1 no longer exists or is already too distant, and Node 0 subsequently removes Node 1 from its memory. Note that this parameter is not part of the 3GPP standard.
- **`*.node[*].wlan[*].mac.spsPowerThreshold`**: specifies the base power threshold in the sensing-based SPS in linear (not in decibels).
- **`*.node[*].wlan[*].mac.spsGridSizeCols`**: specifies the size of the perceived resource grid of each node in terms of subframes.
- **`*.node[*].wlan[*].mac.spsGridSizeRows`**: specifies the number of subchannels per subframe of the perceived resource grid. Note that allowed values are 1, 3, 5, 8, 10, 15, and 20 according to the `numSubchannel` parameter of the LTE-V2X standard.
- **`*.node[*].wlan[*].mac.spsRequiredResources`**: specifies the number of subchannels occupied by each packet.
- **`*.node[*].wlan[*].mac.spsUseSensing`**: specifies whether to use sensing-based SPS (true) or random allocation (false).
- **`*.node[*].wlan[*].mac.spsProbability`**: specifies the keep probability.
- **`*.node[*].wlan[*].mac.spsCollisionThreshold`**: specifies the difference in power a message should have over another colliding message so

that the more powerful message is not considered as corrupt. For example, if the collision threshold is set to 10 (which is equivalent to 10 dB), and there are two colliding messages with a S-RSSI of -90dBm and -80dBm, then the latter message will be considered as good and the former as corrupt. If for example, the threshold is 10, and the colliding messages have an S-RSSI of -90dBm and -93dBm, then both messages are considered as corrupt. Note that this parameter is not specified in the 3GPP standards.

- **`*.node[*].wlan[*].mac.statRegionMinX`**: specifies in meters the starting point of the statistical region in the x-axis.
- **`*.node[*].wlan[*].mac.statRegionMinY`**: specifies in meters the starting point of the statistical region in the y-axis.
- **`*.node[*].wlan[*].mac.statRegionMaxX`**: specifies in meters the end point of the statistical region in the x-axis.
- **`*.node[*].wlan[*].mac.statRegionMaxY`**: specifies in meters the end point of the statistical region in the y-axis.
- **`*.node[*].wlan[*].mac.enableStatRegionX`** : specifies whether to create a statistical region in the x-axis or not.
- **`*.node[*].wlan[*].mac.enableStatRegionY`** : specifies whether to create a statistical region in the y-axis or not.

F. SUMO Settings

This appendix section aims to give a quick guide on the files relevant for the configuration of the SUMO-related parameters.

F.1 Network file (`*.net.xml`)

The network file contains the description of the physical topology of the scenario, including the roads, intersections, traffic logics and even roundabouts. Using the SUMO naming convention, the roads or streets are referred to as *edges*, and the intersections as *junctions* or *nodes*. That is, two edges are connected by junctions. Figure 64 shows the contents of the network file.

```
<?xml version="1.0" encoding="UTF-8"?>

<net version="0.27" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/net_file.xsd">

    <location netOffset="0.00,0.00" convBoundary="0.00,0.00,1000.00,0.00" origBoundary="2.852656,41.882297,2.898048,41.899504"
projParameter="+proj=utm +zone=31 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"/>

    <edge id="E0" from="J0" to="J1" priority="1">
        <lane id="E0_0" index="0" allow="passenger" speed="33.33" length="1000.00" shape="0.00,-11.55 1000.00,-11.55"/>
        <lane id="E0_1" index="1" allow="passenger" speed="33.33" length="1000.00" shape="0.00,-8.25 1000.00,-8.25"/>
        <lane id="E0_2" index="2" allow="passenger" speed="33.33" length="1000.00" shape="0.00,-4.95 1000.00,-4.95"/>
        <lane id="E0_3" index="3" allow="passenger" speed="33.33" length="1000.00" shape="0.00,-1.65 1000.00,-1.65"/>
    </edge>

    <junction id="J0" type="unregulated" x="0.00" y="0.00" incLanes="" intLanes="" shape="0.00,-0.05 0.00,-13.15"/>
    <junction id="J1" type="unregulated" x="1000.00" y="0.00" incLanes="E0_0 E0_1 E0_2 E0_3" intLanes="" shape="1000.00,-13.15 1000.00,-0.05"/>

</net>
```

Figure 64. SUMO network file

The `location` field specifies details about the network projection in case the original network was not using Cartesian coordinates, and therefore, needed to be transformed. The `edge` field describes the created lanes including the allowed type of vehicles (or pedestrian), speed limit, lane length and geometry. The `junction` field defines the lanes that the junctions connect.

The network file may be created using a tool called NETEDIT, a graphical editor for creating and modifying networks.

F.2 Routes file (`*.rou.xml`)

The routes file specifies the vehicle types and routes for the vehicles in the simulation. The `vehicles type` field includes the physical properties of the vehicle, including shape and color, as well the maximum speed and the minimum gap from the vehicle in front. Different routes are identified by their `route id`, and each of them defines the relevant edges and direction of movement of vehicles (e.g., going to the right). Moreover, a `flow` contains the following information, which control how the vehicles are inserted in the scenario and how they behave during the simulation.

- `type` - vehicle type previously defined
- `begin` - departure time of first vehicle
- `period` - insertion period of vehicles
- `departLane` - lane on which the vehicle will be inserted
- `departSpeed` - speed with which the vehicle will enter the scenario

- `departPos` - position at which the vehicle will enter the scenario
- `route` - route of vehicle previously defined

```
<routes>

        <!--.................-->
        <!-- VEHICLE TYPES -->
        <!--.................-->
        <vType  id="type1"              color="green"          guiShape="passenger/sedan"          maxSpeed="33.33"              minGap="2"/>
        <vType  id="type2"              color="yellow"         guiShape="passenger/sedan"          maxSpeed="33.33"              minGap="2"/>
        <vType  id="type3"              color="blue"           guiShape="passenger/sedan"          maxSpeed="33.33"              minGap="2"/>
        <vType  id="type4"              color="grey"           guiShape="passenger/sedan"          maxSpeed="33.33"              minGap="2"/>
        <vType  id="type5"              color="red"            guiShape="passenger/sedan"          maxSpeed="33.33"              minGap="2"/>
        <vType  id="type6"              color="white"          guiShape="passenger/sedan"          maxSpeed="33.33"              minGap="2"/>

        <!--.................-->
        <!-- ROUTES for CAR -->
        <!--.................-->
        <!-- example horizontal route: "3hr" == edge 3 horitzontal to right -->
        <route id="route0hr" edges="E0"/>

        <!--.............................-->
        <!-- CAR VEHICLES nodes[16..x]...-->
        <!--.............................-->
        <!-- for perpetual flow, "end=" > simulation time in .cfg file-->
        <flow   id="flow0hr"                                   type="type2"
                        begin="0"                              end= "7200"                          period="0.001"
                        departLane="random"                    departSpeed="random"
                        departPos="random"
                        route="route0hr"/>

</routes>
```

Figure 65. SUMO routes file

F.3 Configuration file (`*.sumo.cfg`)

The configuration file specifies the associated network and routes files for a given scenario. Moreover, it is possible to configure the `step-length`, which is the granularity of the simulation and has a minimum value of 1ms. It also corresponds to the time interval with which vehicle positions are updated.

```
<?xml version="1.0" encoding="iso-8859-1"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">

    <input>
        <net-file               value="single_road_4lanes.net.xml"/>
        <route-files            value="single_road_4lanes.rou.xml"/>
    </input>

    <time>
        <begin value="0"/>
        <end value="710"/>
        <step-length value="0.001"/>
    </time>

    <report>
        <no-step-log value="true"/>
    </report>

    <gui_only>
        <start value="false"/>
    </gui_only>

</configuration>
```

Figure 66. SUMO configuration file

The parameter values are given in meters (for distance), seconds (for time) and meters per second (for speed). As detailed in [24], other values may be configured, apart from those used in this project. Moreover, additional attributes may be defined in the SUMO files. This allows customizing the scenarios according to the requirements and individual goals of the simulations.

G. Understanding the Impact of the PC5 Resource Grid Design on the Capacity and Efficiency of LTE-V2X (Under Revision)

(Refer to attached document)

Understanding the impact of the PC5 Resource Grid design on the capacity and efficiency of LTE-V2X

Journal:	*IEEE Vehicular Technology Magazine*
Manuscript ID	Draft
Manuscript Type:	Original Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Lopez, Leandro; Universitat Politecnica de Catalunya Mendoza, Charmae; Universitat Politecnica de Catalunya Casademont, Jordi; Universitat Politecnica de Catalunya, Network Engineering Camps-Mur, Daniel; Fundació i2cat
Keywords:	LTE-V2X, PC5 Resource Grid, Connected vehicle

Understanding the impact of the PC5 Resource Grid design on the capacity and efficiency of LTE-V2X

Leandro Miguel Lopez, Charmae Franchesca Mendoza, Jordi Casademont Serra, Daniel Camps-Mur

Universitat Politècnica de Catalunya (UPC)

i2CAT Foundation

3GPP Release 14 presents the first standard for supporting V2X in LTE. Several enhancements are introduced, including a new arrangement of the physical resource grid, where subchannels are the minimum resource unit instead of Resource Blocks. The resource grid is defined by several design parameters, some of them with constraints imposed by 3GPP specifications, that affect the maximum message transmission rate and efficiency of the system. Moreover, the optimum choice of these parameters is closely linked to message length, which is another variable parameter. This paper provides an analysis of the relationship between these design parameters (Resource Block per Subchannel, Transport Block Size Index and Coding rate), message size and system's maximum capacity and efficiency. In doing so, we do not consider channel reuse or radio transmission characteristics because the focus of this paper is trying to find the resource grid design parameters that optimize system capacity, which is a very important aspect to consider by V2X operators.

I. Introduction

Up to now, IEEE 802.11p has been the de facto wireless technology standard for Vehicle-to-Everything (V2X) communications. It is a relatively mature technology and has already been validated by over a decade of field trials. In spite of that, IEEE 802.11p, which uses Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), suffers from a high level of collisions under heavy traffic conditions, mainly due to hidden terminal situations. Moreover, there is also a lack of clear plans for future enhancements, and a lack of business models to support the deployment of additional infrastructure such as Road Side Units (RSUs) [1].

Long-Term Evolution (LTE) based V2X from the Third Generation Partnership Project (3GPP) is a relatively new alternative to IEEE 802.11p-based V2X communications. The first version of LTE-V2X was published in June 2017 under Release 14, which came with numerous enhancements to the existing Device-to-Device (D2D) communications in order to accommodate vehicular communications. The proposed enhancements include a new arrangement of the resource grid of the physical layer, and two types of D2D channel access mechanisms: i) a mechanism coordinated by the evolved NodeB (eNB), named Mode 3, and ii) a distributed mechanism, where User Equipments (UEs) access the channel on their own, named Mode 4. Moreover, LTE-V2X employs different radio interfaces: i) interface between the vehicle and eNB, named LTE-Uu, and ii) interface between vehicles, named PC5.

In order to operate a PC5 resource grid, the operator has to select the value of several design parameters (Resource Block per Subchannel, Transport Block Size Index and Coding rate). The selection of these parameters affects the maximum message transmission rate and efficiency of the system. Our main contribution in this paper is an analysis of the theoretical maximum message capacity and efficiency of the PC5 resource grid considering the influence of these design parameters, that, in fact, have several constraints imposed by the 3GPP specifications.

Assuming that the resource grid is fully dedicated to PC5 communication (i.e., any communication with the eNB takes place using different LTE carriers), our analysis applies to both Modes 3 and 4, and uncovers a high sensitivity between the application message size and the aforementioned design parameters.

There are several works that analyse the capacity of PC5 channel from the different aspects. [2, 3] present a general overview of the LTE-V2X and provide simulation results for the behaviour of the sensing-based semipersistent scheduling (SPS) Medium Access Control (MAC) algorithm for channel reuse and give results of the Packet Delivery Ratio (PDR). [4] analyses the impact of PHY and MAC design parameters as sensing period,

power threshold, time window or time before evaluation, on the system performance. [5] is a 3GPP's technical report that presents the suggested evaluation methodology to study LTE-V2X and provide many simulation results of Packet Reception Ratio under multiples scenarios with channel reuse. [6] shows an innovative approach proposing a new metric called spatial capacity total amount of bits that can be successfully delivered in 1 km and 1 s.

Our work eschews these concepts in order to focus on how the upper bounds of the system capacity are affected by the resource grid design parameters. To the best of our knowledge, this is the first paper that studies how the capacity and efficiency of the PC5 resource grid is affected by: i) the number of Resource Blocks per Subchannel, ii) the Transport Block Size Index and Coding rate, iii) the type of V2X message and the associated traffic model (derived from a real ETSI-G5 stack), and iv) the transmission time interval and message size. Reference [7] provides an interesting work where the authors relate Modulation and Codification Scheme (MCS), that can be mapped to the Transport Block Size Index, with two parameters, Firstly, with the awareness and reuse ranges, and then they derive the maximum number of neighbour cars that the system can sustain. Our work differs from theirs in two approaches. As reference [7] considers channel reuse, their system's maximum capacity computations are not only affected by resource grid design parameters, but also by the interference produced by this channel reuse. Moreover, they focus on MCS but do not deal with subchannel design (number of Resource Blocks per Subchannel), which affects the system performance when there is the transmission of massages of different sizes, as it happens in most real use cases, or the effect of the 3GPP specification restrictions when the message size varies.

This paper is organized as follows. Section II provides an overview of the LTE-V2X resource grid structure. Section III introduces traffic models and message types, commonly used by V2X applications. Section IV and V derive respectively the maximum capacity and efficiency achieved by the PC5 resource grid. Finally, Section VI summarizes and concludes the paper.

II. The LTE-V2X resource grid structure

Release 12 D2D was not designed with the strict latency and vehicle density requirements of V2X in mind. In order to improve the support for V2X, Release 14 introduced a new arrangement on the resource grid, as illustrated on Figure 1.

In the time domain, the resource grid is divided into subframes of 1 ms each. In the frequency domain, the grid is divided, in every subframe, into physical Resource Blocks (RBs) that span 180 kHz each with 12 subcarriers of 15 kHz. Every subcarrier transmits a total of 14 symbols per subframe, in which 9 symbols are devoted to data transmission, 4 symbols are used for Demodulation Reference Signals (DMRS), and the remaining symbol is used for Tx-Rx turnaround.

Apart from these fixed parameters, there are some others that depend on design implementation and its selection will define the maximum system capacity and efficiency, as described below.

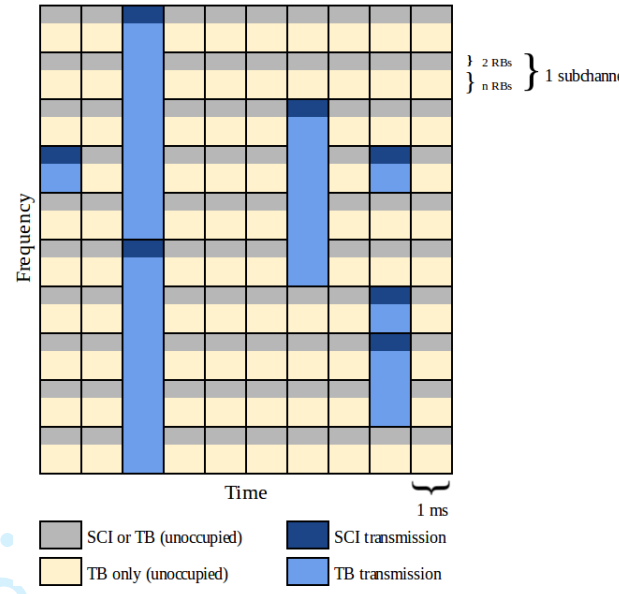

Figure 1: *Resource grid with 10 subchannels per subframe.*

The number of RBs per subframe depends on the available channel bandwidth. LTE-V2X supports channel bandwidths of 10 MHz and 20 MHz, which would have 50 RBs per subframe and 100 RBs per subframe respectively [8]. LTE-V2X supports QPSK and 16QAM modulation schemes. For QPSK, 2 bits per symbol are transmitted, while 4 bits per symbol are utilized in 16QAM. Since 9 symbols per RB are used for data transmission, this translates to 216 total bits per RB for QPSK and 432 bits for 16-QAM. These total number of bits include the payload plus redundancy to cope with transmission errors. Depending on the required transmission robustness, different coding rates (ratio between payload bits plus Cyclic Redundancy Check (CRC) and total transmitted bits) can be set.

RBs in the same subframe are grouped into subchannels. A key characteristic of LTE-V2X is that resource allocation is in subchannel granularity (rather than in RB granularity). Every subchannel is composed of control information and user data. The control portion occupies 2 RBs, corresponding to the Sidelink Control Information (SCI) transmitted in PSCCH (Physical Sidelink Control Channel). The SCI includes information such as the MCS (Modulation and Coding Scheme), and the resource reservation information needed in sensing-based semi-persistent scheduling [9]. The group of RBs

carrying user data in a subchannel is named Transport Block (TB). A transmission of a TB requires the transmission of the corresponding SCI, as the latter is necessary for the correct reception and demodulation of the TB. A TB is transmitted in the PSSCH (Physical Sidelink Shared Channel) and occupies a variable number of RBs depending on the message size. The PSCCH and PSSCH are either adjacent or non-adjacent to each other on the same subframe. Depending on this configuration, the allowed subchannel size or number of RBs per subchannel differs as specified in [10]. In the case of adjacent PSCCH-PSSCH, 5, 6, 10, 15, 20, 25, 50, 75 or 100 RBs may be configured per subchannel, while 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 30, 48, 72 or 96 RBs per subchannel for the non-adjacent case. In this paper, only the adjacent case is considered. If the amount of data is large, it may occupy the succeeding subchannels (including the 2 RBs for the next SCI). Since the PSCCH and the PSSCH are always in the same subframe, it is possible to decode a PSSCH immediately after a PSCCH is recovered. Hence, the proposed vertical arrangement of data in the resource grid reduces decoding latency, which is one of the key objectives in V2X. Chapters IV and V study the influence of choosing the value of these parameters.

The eNB informs UEs of the V2X resource grid structure (including the number of subchannels in a subframe) through the System Information Block (SIB) 21 message transmitted using the Physical Downlink Shared Channel (PDSCH). In-coverage vehicles operating in either Mode 3 or 4 receive these messages from the LTE network, which may occupy a percentage of the resource grid. However, in this paper, we assume that a separate carrier is used for V2X communications, for example the 10 MHz unlicensed ITS-G5A carrier at 5.9 GHz, while communications with the eNB occur through a traditional licensed LTE carrier. Therefore, the overhead introduced by SIB signalling is not accounted for in our capacity and efficiency calculations.

III. Analysis of ETSI-G5 traffic characteristics

On top of the wireless access technologies, there are two main protocol stacks for V2X communications, namely ETSI-G5, used in Europe, and WAVE (Wireless Access in Vehicular Environments), used in the United States [11]. While providing similar functionality, the two stacks differ in the actual messages used. In this paper, we focus on ETSI-G5, although we posit that our conclusions apply also to WAVE.

ETSI-G5 defines two main messages for safety related V2X services, namely Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM). CAMs can be characterized as short messages broadcasted periodically by each vehicle. CAMs from a vehicle inform surrounding vehicles about its status (such as its speed, position, direction, etc.). CAMs are generated at an interval of at least 100 ms and at most 1000 ms [12]. DENMs are event-triggered messages to inform or warn other vehicles about situations such as hazardous road conditions and abnormal traffic situations [13], thus DENMs are transmitted aperiodically unlike CAMs. When using the ETSI-G5 protocol stack, these messages are encapsulated in the GeoNetworking protocol (GN) and Basic Transport Protocol (BTP), before being sent to the LTE subchannel.

Protocol header sizes are variable. Thus, the GeoNetworking header size, although it depends on the target address region, is typically between 40-60 bytes. The BTP header has a length of 4 bytes. The typical CAM size falls within the range of 30-300 bytes. The size of a DENM is also variable and is commonly between 60-800 bytes.

In order to get a distribution of the real message size, traces have been captured using a V2X testbed that we developed. Although this testbed uses transmission over IEEE 802.11p with Cohda Wireless MK5 devices [14], upper layer protocol message sizes are independent of the wireless technology used to transmit them, so we can use it to empirically determine realistic V2X payload sizes.

Figure 2 shows the distribution of the sizes of CAMs and DENMs (including GN and BTP protocol headers) acquired in the testbed with use cases of Emergency Vehicle Warning (EVW) and Cooperative Forward Collision Warning (CFCW). In both cases, vehicles always send 10 CAMs per second. Additionally, in the EVW testbed, DENMs are continuously generated by an ambulance whenever it is moving or stopped. In the CFCW case, DENMs are temporally generated by any vehicle whenever it detects that it is critically approaching another vehicle, in a trajectory of collision. This situation is detected using CAM messages.

We observe that, above 80% of the CAMs transmitted are either 175 bytes or 160 bytes while about 15% are 291 bytes in size. Meanwhile, about 80% of DENMs have a 317-byte packet size.

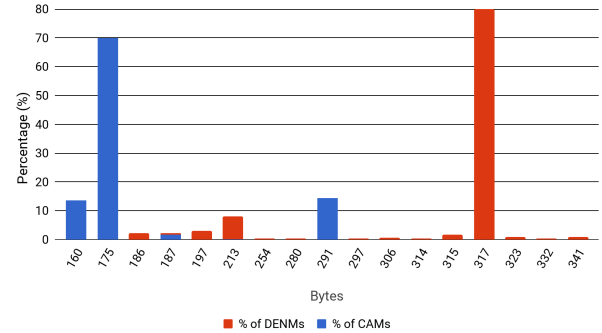

Figure 2: *Histogram of CAM and DENM message sizes taken from a COHDA Wireless device.*

On the other hand, 3GPP published in Annex A of [15], a methodology to evaluate LTE V2X, in which it is

suggested to use "short" CAMs with a size of 190 bytes transmitted 4 times every 5 CAMs, "long" CAMs with a size of 300 bytes transmitted once every 5 CAMs, and DENMs with a size of 800 bytes, all of them including GN and BTP headers. The CAM packet sizes and transmission frequencies from the empirical data are close to those suggested by 3GPP. This is not the case for the DENM.

Nevertheless, for the evaluation of the capacity limits in the following chapters, this paper adopts short CAM message sizes of 190 bytes and long CAM messages sizes of 300 bytes, which are similar to empirical results and 3GPP proposal, and DENMs of 300 bytes, which are closer to our empirical results.

IV. Capacity limits of the adjacent PSCCH-PSSCH PC5 resource grid

The number of CAMs and DENMs that can be transmitted in a given time is constrained by the design of the LTE-V2X sidelink resource grid, hence, the number of vehicles that can transmit V2X messages is also bounded. This analysis focuses on the influence that different design parameters of the resource grid have in the final system capacity. For this reason, we do not consider channel reuse that has already been studied in different papers. Our goal is to maximize the number of V2X messages that can be transmitted in a resource grid without channel reuse.

Prior to performing the analyses, we present the design parameters of a resource grid. Firstly, to establish the resource grid structure, the system designer has to choose the transmission modulation QPSK (modulation order 2) or 16QAM (modulation order 4), and the MCS Index (I_{MCS}). These two parameters are merged into a single one named Transport Block Size Index (I_{TBS}) as indicated in Table 8.6.1-1 of [5]. I_{TBS} defines the robustness of the radio modulation and the coding rate, which in turn, affect the coverage area of radio transmissions and the number of bits that can be transmitted in a fixed size TB. The lower the I_{TBS}, the more robust the communication is, and therefore, the farther messages are correctly received, but the fewer payload bits per fixed size TB are transmitted. Reference [7] provides values of typical rages that can be reached with different MCS values.

Secondly, the designer has to choose the number of RBs per TB, which affects to the efficiency of the resource grid, as will be seen in section V.

The relationship between I_{TBS}, the number of physical RBs per TB (N_{PRB}) and the Transport Block payload Size (TBS) in bits is defined in Table 7.1.7.2.1-1 of [5].

In order to design the resource grid, the most common message size has to be considered first. As per our measurements, it is the 175-byte CAM, and per the suggested evaluation methodology of [15], it is the 190-byte CAM. As both values are very similar, our design will be based on the 190-byte (1520 bits) message.

As a design factor, different possibilities of a subframe division into subchannels, which are able to transmit 1520 bits in the payload, are summarized in Table 1. In the cases where the number of RBs per subframe (50 RBs in a 10 MHz grid) is not divisible by the number of subchannels per subframe, the last subchannel will be bigger than the previous ones in the subframe. If a message does not fully occupy all the RBs of a subchannel, the unused RBs are left empty.

Subchannels per subframe	RBs per subchannel	Subchannels per short CAM	Short CAMs per subframe	RBs per TB	TB size (bits)	TBS Index (I_{TBS})	MCS Index (I_{MCS})	Modulation	Coding rate
10	5	3	3	13	1608	7	7	QPSK	0.58
8	6	2	4	10	1544	9	9	QPSK	0.73
5	10	1	5	8	1608	11	12	16QAM	0.47

Table 1: *Subchannel design parameters for a 10 MHz grid considering the short CAM (190 bytes).*

The coding rate is calculated as:

$$coding\ rate = \frac{A + 24(N_{CB} + 1)}{N_{Ch}}$$

where $N_{CB} = 0\ if\ C = 1, or\ C\ if\ C > 1$

Payload size A must be a valid TB size from Table 7.1.7.2.1-1 of [12], which specifies the corresponding TB size given the I_{TBS} and N_{PRB} [8]. N_{CB} is determined by the number of code blocks C calculated according to section 5.1.2 of [9], which specifies that, as all our messages are smaller than 6144 bits, N_{CB} should be 0. The number of channel bits N_{ch} is calculated considering the 216 bits per RB (for QPSK) or 432 bits per RB (for 16QAM) and then multiplying it by the number of RBs per TB (N_{PRB}). Taking the first row of Table 1 as an example, where the payload size A is 1608 bits, N_{CB} is zero, and N_{ch} is 216 bits multiplied by 13 RBs/TB, the coding rate is thus calculated as 0.58 by using the above formula.

According to the 3GPP evaluation methodology [5], the suggested transmission robustness configuration is QPSK with target coding rate of 0.5 for baseline, and QPSK with coding rate of 0.7 or 16QAM with coding rate of 0.5 as optional, with the first one being the most robust and the last, the least robust.

As per our evaluation, we select the intermediate case in Table 1 where the resource grid is designed to have 6 RBs per subchannel and a robustness defined by I_{TBS} equal to 9. With this configuration, it is possible to define 8 subchannels per subframe with each message occupying 2 subchannels each, or 10 RBs per TB, which leads to a coding rate of 0.73. This configuration also provides a TBS closest to the nominal packet size. Note that the last subchannel of the subframe has 8 RBs, therefore, the TB size of the joined last 2 subchannels is 12 RBs. This extra space can be used to send the same number of bits with a more robust I_{TBS} or simply leave it unused.

Now, to allocate "long" CAMs and DENMs, which have a size of 300 bytes, it is possible to either increase

the payload size reducing the transmission robustness, or, maintain the robustness level while increasing the number of RBs per TB by joining more than two subchannels into a bigger one. As we consider that the coverage area of a transmitted message has to be maintained, the approach is to join two or more subchannels into a bigger one. As we have previously defined a basic resource grid structure of 6 RBs per subchannel, we have to continue using this structure. Nevertheless, Table 2 shows other two possible configurations, matching those presented in Table 1.

Subchannels per subframe	RBs per subchannel	Subchannels per DENM	DENMs per subframe	RBs per TB	TB size (bits)	TBS Index (I_{TBS})	MCS Index (I_{MCS})	Modulation	Coding rate
10	5	5	2	23	2408	6	6	QPSK	0.49
8	6	4	2	22	2664	7	7	QPSK	0.57
5	10	2	2	18	2536	8	8	QPSK	0.66

Table 2: *Subchannel design parameters for a 10 MHz grid considering the long CAM or the DENM (300 bytes).*

We thus consider that 300-byte long CAMs and DENMs occupy 4 basic subchannels in the intermediate case that we use per our evaluation. As indicated in Table 2, a 300-byte (2400-bit) long CAM or a DENM fits a 2664-bit TB payload size, defined by a TB formed with 22 RBs coded at QPSK with $I_{TBS} = 7$, which specifies a coding rate of 0.57. This is slightly more robust than the profile QPSK with coding rate of 0.73 used in short CAMs, which is a good configuration, as information transmitted in DENMs is usually more critical than the one transmitted in CAMs.

In summary, we now have defined the 10 MHz resource grid to be used for the analyses. The resource grid has a total of 50 RBs per subframe, where each subframe is divided into 6 RBs per subchannel. A 190-byte short CAM modulated in QPSK with a code rate of 0.73 occupies two subchannels and uses $I_{TBS} = 9$, while 300-byte long CAMs and DENMs modulated in QPSK with a code rate of 0.57 occupy four subchannels and use $I_{TBS} = 7$.

As we can see, LTE-V2X regulations enable a high degree of flexibility on defining the resource grid structure which will depend on the choice of each operator. This is a very interesting characteristic but we have also to understand the impact that this choice has with the traffic model generated by vehicles. To evaluate this aspect, we consider: i) a traffic mix with varying portion of short and large messages, ii) varying the packet generation interval, and iii) varying the message sizes.

The target of this evaluation is to compute the maximum number of vehicles that the resource grid can hold under ideal resource scheduling conditions depending on design parameters. Thus, the paper assumes that channel reuse, zoning, collisions, hidden terminal situations, noise, interference and redundant transmissions are ignored and that as much as possible, the resource grid is fully occupied by messages.

A) Capacity with varying traffic mix

In the following two scenarios, the capacity when vehicles generate a mix of CAMs and DENMs is analyzed. We consider that 80% of the CAMs transmitted are short CAMs (190 Bytes) while the other 20% are long CAMs (300 Bytes), as used in 3GPP technical reports [13].

We start analyzing two scenarios: i) a case where vehicles generate either CAM or DENM messages every 100 ms, and ii) a case where vehicles always generate CAMs every 100 ms, and may additionally generate a DENM.

In the scenario shown in Fig. 3, a vehicle generates a message *m* every 100 ms, where the probability of *m* being a DENM is signalled by the x-axis *(x)*, whereas *m* is a short CAM with probability equal to *(1-x)*0.8*, or a long CAM with probability equal to *(1-x)*0.2*.

In the next scenario shown in Fig. 4, a vehicle generates either only a message *m*, which is a CAM, or a message *m* together with another message *m'*, which is a DENM, every 100 ms. The probability of generating a DENM plus CAM signalled by the x-axis *(x)*. In addition, *m* is a short CAM with a probability of 0.8 and a long CAM with a probability of 0.2.

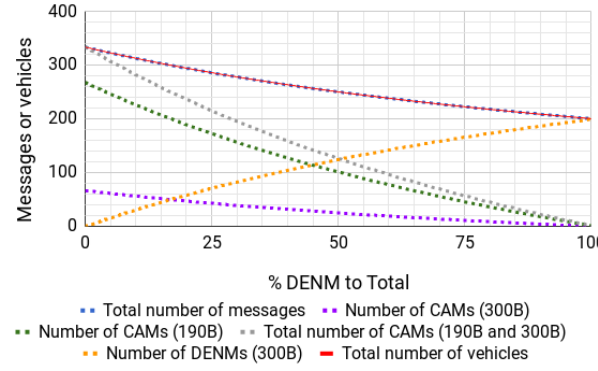

Figure 3: *Capacity (number of messages or number of vehicles) with CAM only (190B and 300B) or DENM only mix. Transmission period 100 ms.*

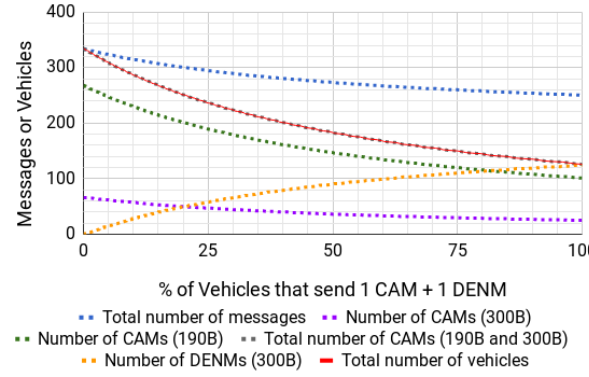

Figure 4: *Capacity (number of messages or number of vehicles) with CAM only (190B and 300B) or CAM+DENM. Transmission period 100 ms.*

To compute the maximum capacity, we consider the transmission period *t*, that is messages generated from a vehicle every *t* ms can be allocated in any subframe within a *t* ms interval.

We can observe from Fig. 3 and 4, that, in the best case, when 100% of vehicles send only a mix of short and long CAMs (without any DENM), the LTE-V2X frame could sustain 334 messages every 100 ms from 334 vehicles.

On the other hand, when 100% of vehicles send only DENMs, the LTE-V2X resource grid could sustain up to 200 DENMs every 100 ms, coming from 200 vehicles.

Moreover, when 100% of vehicles send a DENM together with a CAM, the LTE-V2X frame could sustain only up to 125 CAMs (100 short and 25 long) and 125 DENMs every 100 ms, coming from 125 vehicles.

The observed differences are a consequence of the considered message sizes and, in the second case, the increment of messages per second.

B) Capacity with varying the transmission time interval

The previous scenarios assumed that all message transmission time intervals were 100 ms. We study now the impact of increasing the transmission interval. Fig. 5 shows the capacity of the same 10 MHz resource grid with 6 RBs per subchannel and a robustness defined by I_{TBS} = 9 for short CAMs (190 bytes) and I_{TBS} = 7 for long CAMs and DENMs (300 bytes), when there is a mix of vehicles that transmit a message *m* every 100 ms and vehicles that transmit *m* every 1000 ms. The probability of a vehicle sending *m* every 100 ms is indicated by the x-axis *(x)* in Fig. 5, while the probability of a vehicle sending *m* every 1000 ms equals *(1-x)*. In this scenario, *m* is assumed to be always a short CAM.

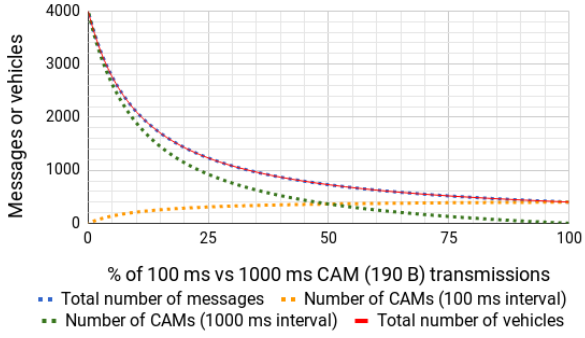

Figure 5: *Capacity with a mix of 100 and 1000 ms transmission periods.*

When all vehicles transmit only 1 short CAM every 100 ms, the LTE-V2X resource grid can sustain 400 vehicles (as opposed to the 334 vehicles maximum capacity previously discussed, when vehicles transmit a 80-20 percent mix of short and long CAMs). However, when vehicles that transmit every 1000 ms are introduced into the mix, the capacity of the LTE resource grid increases exponentially, up to a maximum of 4000 messages from unique sources when all of the vehicles transmit every 1000 ms.

As it can be observed, an increase in transmission time interval exponentially increases the capacity of the resource grid in ideal conditions. Also, varying the transmission time interval could have one of the biggest impacts to capacity, but it is nonlinear. This relationship between transmission time interval and capacity is exploited by IEEE 802.11p, where congestion control is performed by increasing the transmission time interval when there are many vehicles. LTE-V2X would benefit from a similar congestion control mechanism as well.

C) Capacity with varying the message size

In the previous scenarios, it was assumed that messages were of fixed-length. In the following scenario, the effect of changing the message size itself on the capacity is analyzed. The impact of combinations of different parameters such as the modulation and coding scheme (dictated by the I_{TBS} value) and channel bandwidth (10 MHz vs 20 MHz), using 6 RBs per subchannel is considered. The target coding rate for all transmissions is 0.5, thus we compare I_{TBS} = 7 (QPSK) vs. I_{TBS} = 12 (16QAM). It is assumed that the messages are transmitted at a period of 100 ms. Ideal resource scheduling conditions are again assumed. Fig. 6 illustrates this scenario.

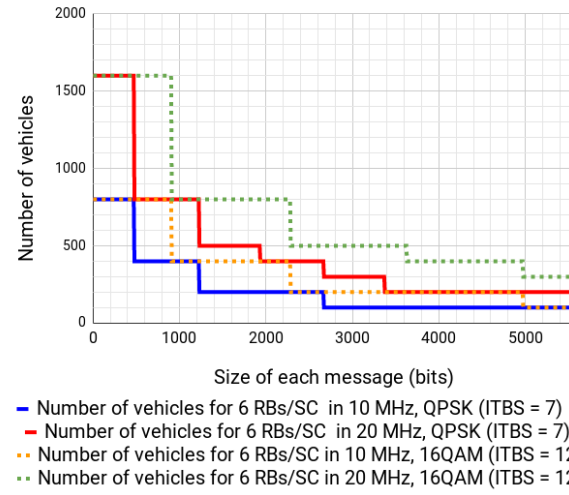

Figure 6: *Capacity when message size is varied.*

As depicted in Fig. 6, the number of messages sustained by the LTE frame decreases in a staircase-like manner as the message size increases. The reason behind this can be visualized better in Fig. 7, which depicts the 10 MHz resource grid with a division of 6 RBs per subchannel. In subframe 1, 8 messages, occupying one subchannel each, have TBs that are partially occupied by the useful data of the message, 2 RBs occupied by the SCI, and the rest is unused space. This corresponds to a

resource grid message capacity of 800 messages given a transmission time interval of 100 ms. As the message size is increased, the unused space is gradually filled up, until all the unused space is occupied as in subframe 3. However, in subframe 4, when the message size is increased further, one subchannel is no longer sufficient to accommodate the message. Thus, an additional subchannel has to be allocated per message, since the smallest unit of resource allocation in LTE-V2X is in subchannels rather than in RBs. An additional subchannel per message means that the subframe can now be filled by only 4 messages of the same size instead of 8, resulting in a capacity drop from 800 to 400 vehicles. Resource grid capacity in LTE-V2X is thus extremely sensitive to the message size.

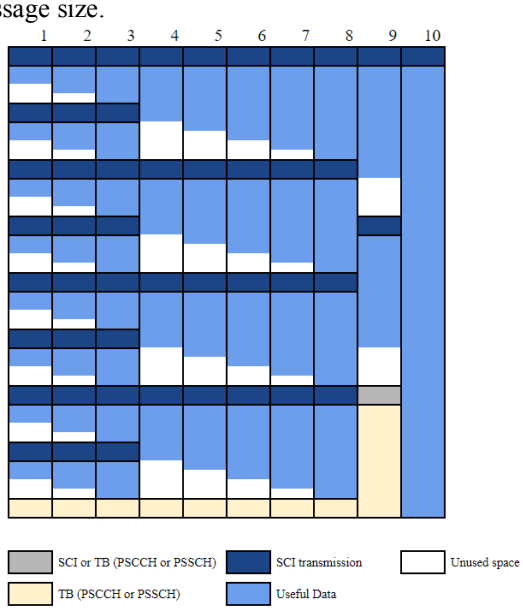

Figure 7: *Effect of growing message size on subchannel occupancy in a 10 MHz grid with 6RBs per subchannel.*

As the message size is increased further, the newly available unused space from the extra subchannel is gradually filled up until all the space for data is fully occupied in subframe 8. The process continues until only one message fully occupies an entire subframe for all subframes in the resource grid as in subframe 10. In this case, the resource grid message capacity is 100 messages. Returning to Fig. 6, it can be observed that increasing the bandwidth from 10 MHz to 20 MHz increases the message capacity by a factor of 2, while increasing the robustness of the modulation changing I_{TBS} from 7 to 12, means that each subchannel could carry approximately twice as much information, thus the steps are about twice as wide. However, they are not exactly double due to the fact that coding rate steps are nonlinear.

V. Efficiency limits in the LTE-V2X resource grid

The efficiency of the resource grid is affected and decreased by three factors: the coding rate described in section IV, the two RBs that are automatically dedicated for control in each message transmission and, depending on the size of the subchannels and the size of the message, the unused space on the TB. In this section, the efficiency in the LTE-V2X resource grid is investigated.

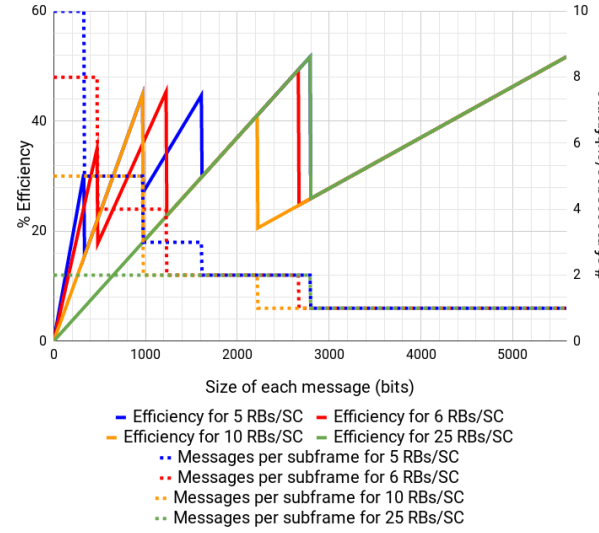

Figure 8: *Message size vs. efficiency for I_{TBS} = 7 (QPSK) at 10 MHz channelization.*

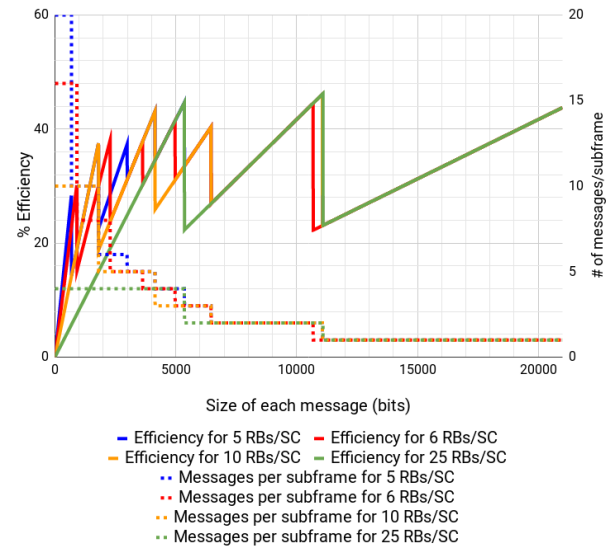

Figure 9: *Message size vs. efficiency for I_{TBS} = 12 (16QAM) at 20 MHz channelization.*

Efficiency is defined here simply as the ratio of the size taken up by sum of all upper-layer message bits over the total size occupied (in the case of QPSK and 10 MHz bandwidth, 216 bits per RB by 50 RBs). In the following scenario, again a channel bandwidth of 10 MHz is used, thus, there would be 50 RBs per subframe. It is assumed that the Transport Block Size Index is I_{TBS} = 7 (QPSK), looking for a target coding rate for all transmissions close to 0.5. All messages are assumed to be of the same size

and are transmitted at a period of 100 ms. Also, ideal resource scheduling conditions are again assumed, and zoning and redundant transmissions are ignored.

The efficiency as message size is increased using resource grids with 5, 6, 10 and 25 RBs per subchannel is illustrated in Fig. 8. The resulting sawtooth-like shape is due to the jumps in the number of messages per subframe. This is the same reason for the staircase-like shape of the capacity graph in the scenario in Section IV-C.

In Fig. 8, peaks are due to the local maxima of efficiency, i.e. the message perfectly fills all the allocated space for data. However, increasing the message size even by a single bit means that the allocated space is no longer sufficient, thus requiring an additional subchannel. The additional subchannel contributes to the sudden drop in efficiency. These sudden efficiency drops show that the efficiency is highly sensitive to the message size, similar to capacity. The amount of unused space from the additional subchannels could be visualized in subframe 4 in Fig. 7.

It can be observed that the local peak efficiencies generally increase with larger message sizes. This is because larger message sizes mean that there are fewer messages on the resource grid, consequently the overhead taken up by the SCIs diminishes. For instance, in Fig. 7, there are only 4 messages in subframe 8 compared to subframe 3 which has 8 messages, but both subframes are fully occupied. However, subframe 8 is more efficient in that there are fewer RBs occupied by SCIs and more useful data in subframe 8 compared to subframe 3. This means that there is a trade-off between efficiency and number of messages that could be accommodated: having larger messages generally increases efficiency because more space is spent on transmitting useful data rather than control information, however, this means decreasing the number of messages that the resource grid can accommodate. With a target coding rate of 0.5, the maximum efficiency that could be achieved is around 50%. However, even at the largest possible message size, the efficiency would be under the percentage imposed by the coding rate because of the presence of the SCIs. Upon a closer look at Fig. 8, it could be seen that it is not always the case that the local peaks in efficiency are monotonically increasing with message size. This could be explained in subframe 9 of Fig. 7. Subframe 9 has 2 messages each occupying 3 subchannels, but it also has 2 remaining subchannels that cannot be occupied by a message of the same size. This contributes to a significant inefficiency.

The efficiency on a 20 MHz resource grid using I_{TBS} = 12 (16QAM), also targeting a coding rate of 0.5, is shown on Fig. 9. The general trend is similar to that of the previous scenario, except that each subchannel has twice the capacity due to the increase in modulation, and that twice as many messages can be accommodated due to the increase from 10 to 20 MHz, thus, the capacity is increased fourfold.

VI. Summary and conclusions

In this paper, we have investigated the effect of resource grid design parameters in the capacity and efficiency of LTE-V2X.

Our main contribution is to expose the sensitivity of LTE-V2X performance parameters to the three basic resource grid design parameters (Resource Block per Subchannel, Transport Block Size Index and Coding rate), and also to the traffic pattern generated by vehicles.

In order to present these results, we do not consider channel reuse and radio propagation conditions that have already been studied by other authors. Nevertheless, as channel reuse improves LTE-V2X global capacity in a geographical region, the values presented in this paper have to be considered as the upper bound of correctly received messages possibly received by a specific vehicle.

Our main findings are that capacity and efficiency in LTE-V2X are very sensitive to the relation between message size and subchannel configuration. For instance, the number of messages sustained by the LTE-V2X resource grid decreases in a staircase-like manner as the message size increases, therefore, a single increase of one byte in the message size, could produce a reduction close to 50% on the number of messages that can fit in this resource grid.

An implication of this high sensitivity is that mobile network operators would have to deeply study the traffic pattern of vehicles in their area of service in order to define LTE-V2X operating parameters.

Acknowledgements

This work has been supported in part by the ERDF and the Spanish Government through project TEC2016-79988-P, AEI/FEDER, UE; and by the Spanish Ministerio de Industria, Energía y Turismo and CELLNEX through project TSI-100102-2015-13 V2XArch.

References

[1] A. Bazzi, B. M. Masini, A. Zanella and I. Thibault, "On the Performance of IEEE 802.11p and LTE-V2V for the Cooperative Awareness of Connected Vehicles", IEEE Transactions on Vehicular Technology, vol. 66, no. 11, pp. 10419-10432, Nov. 2017

[2] R. Molina-Masegosa and J. Gozalvez, "System Level Evaluation of LTE-V2V Mode 4 Communications and Its Distributed Scheduling", 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, Australia, 2017, pp. 1-5.

[3] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5G V2X vehicular communications", IEEE Vehicular Technology Magazine, vol. 12, no. 4, pp. 30-39, December 2017.

[4] A. Bazzi, G. Cecchini, B.M. Masini, A. Zanella, "Study of the impact of PHY and MAC parameters in 3GPP C-V2V Mode 4", arXiv:1807.10699v1. 2018.

[5] 3GPP, "TR 36.885 Technical Specification Group Radio Access Network; Study on LTE-based V2X Services (V14.0.0, Release 14)", 3GPP, Tech. Rep., June 2016.

[6] Y. Park, S. Weon, I. Hwang, H. Lee, J. Kim and D. Hong, "Spatial capacity of LTE-based V2V communication", 2018 International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, HI, 2018, pp. 1-4.

[7] A. Bazzi, B. M. Masini and A. Zanella, "How many vehicles in the LTE-V2V awareness range with half or full duplex radios?", 2017 15th International Conference on ITS Telecommunications (ITST), Warsaw, 2017, pp. 1-6.

[8] 3GPP, "TS 36.101 Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (V14.6.0 Release 14)", 3GPP, Tech. Rep., January 2018.

[9] 3GPP, "TS 36.212 Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (V14.5.1, Release 14)", 3GPP, Tech. Rep., January 2018.

[10] 3GPP, "TS 36.331 Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol Specification (V14.5.1, Release 14)", 3GPP, Tech. Rep., January 2018.

[11] IEEE Standard for Wireless Access in Vehicular Environments (WAVE) -- Networking Services", IEEE Std 1609.3-2016 (Revision of IEEE Std 1609.3-2010), April 29 2016.

[12] European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service", ETSI TS 102 637-2, European Telecommunications Standards Institute, March 2011.

[13] European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service", ETSI TS 102 637-3, European Telecommunications Standards Institute, September 2010.

[14] Cohdawireless.com, 'Hardware', 2018. [Online]. Available: http://www.cohdawireless.com/solutions/hardware/. [Accessed: 19-Mar-2018].

[15] 3GPP, "TR 36.213 Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (V14.5.0, Release 14)", 3GPP, Tech. Rep., January 2018.

[16] 3GPP, "TR 36.785 Technical Specification Group Radio Access Network; Vehicle to Vehicle (V2V) services based on LTE sidelink; User Equipment (UE) radio transmission and reception (V14.0.0, Release 14)", 3GPP, Tech. Rep., October 2016.

LIST OF FIGURES WITH LARGER FORMAT

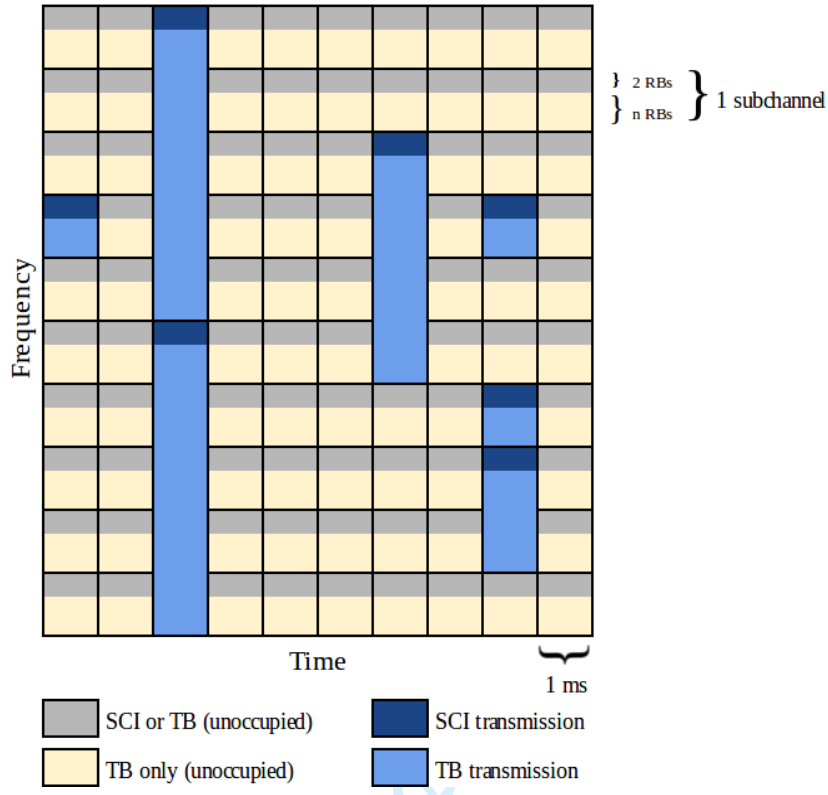

Figure 1: *Resource grid with 10 subchannels per subframe*

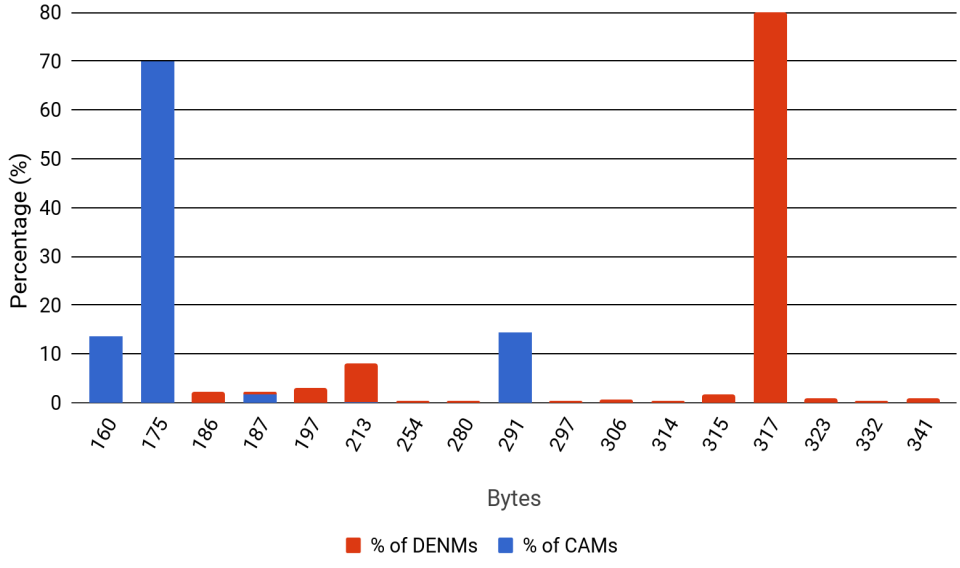

Figure 2: *Histogram of CAM and DENM message sizes taken from a COHDA Wireless device.*

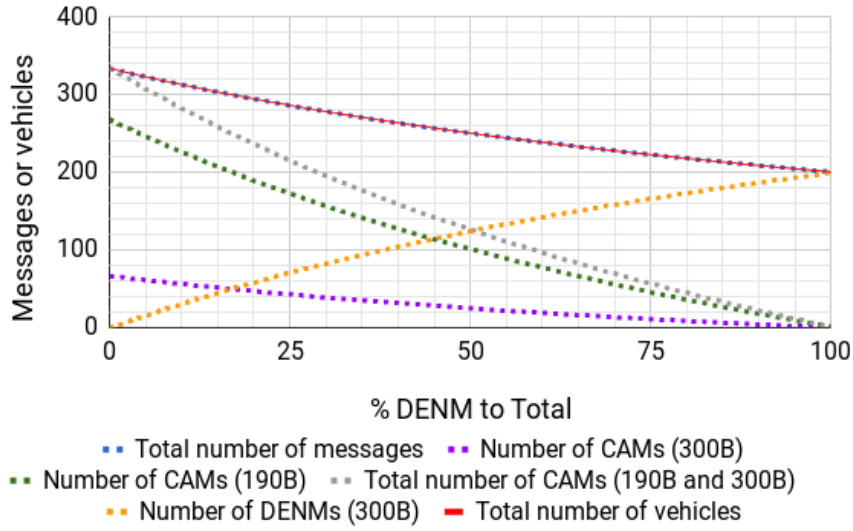

Figure 3: *Capacity (number of messages or number of vehicles) with CAM only (190B and 300B) or DENM only mix. Transmission period 100 ms.*

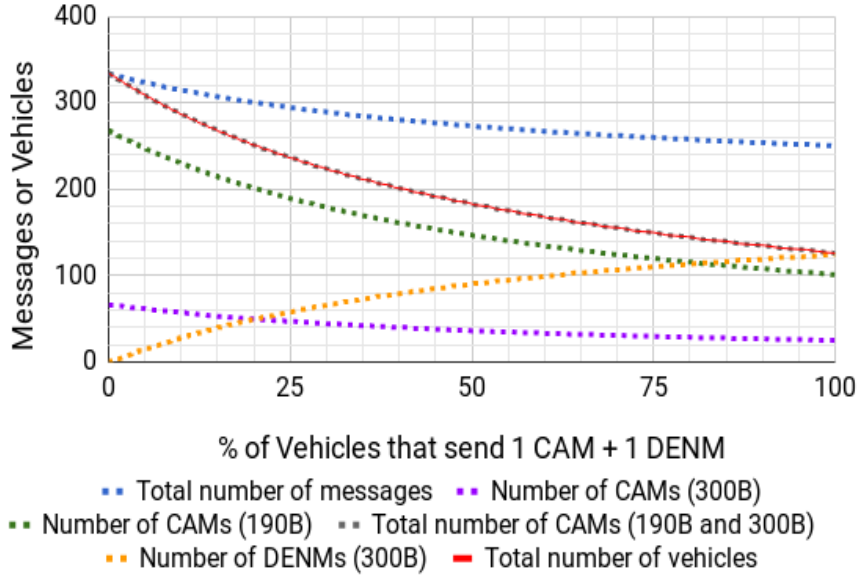

Figure 4: *Capacity (number of messages or number of vehicles) with CAM only (190B and 300B) or CAM+DENM. Transmission period 100 ms.*

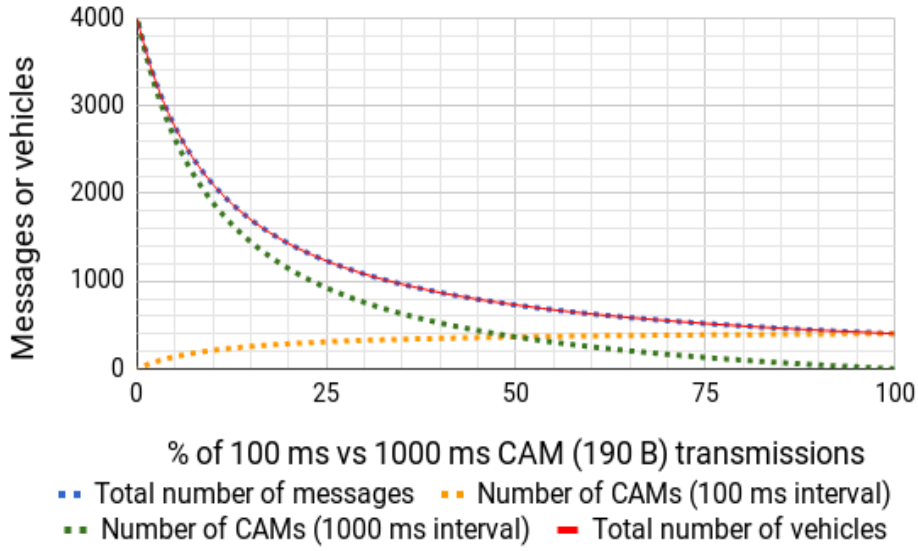

Figure 5: *Capacity with a mix of 100 and 1000 ms transmission periods.*

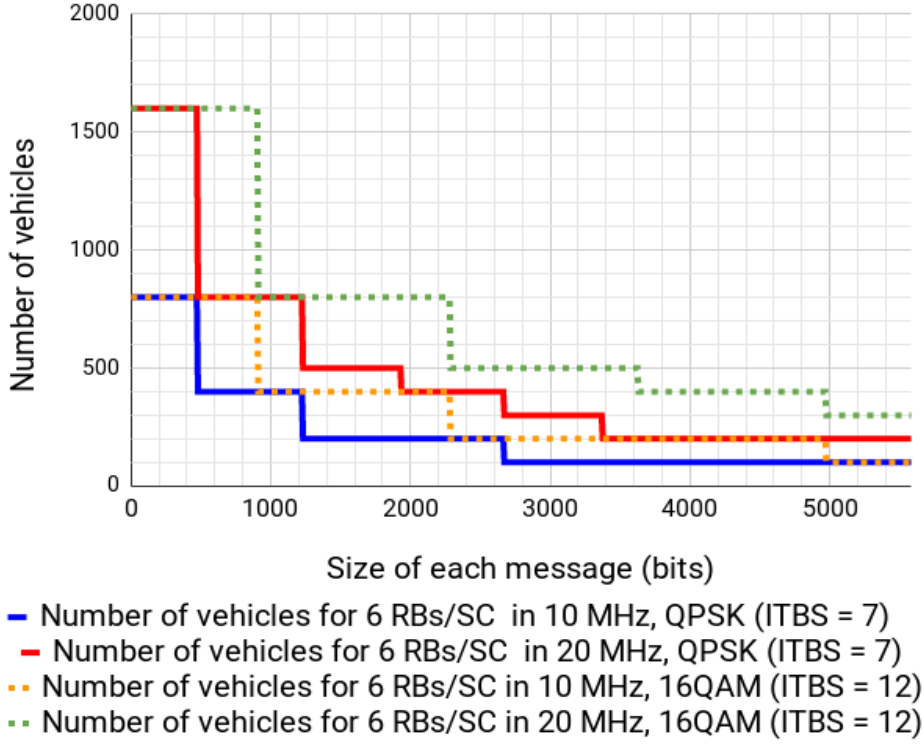

Figure 6: *Capacity when message size is varied.*

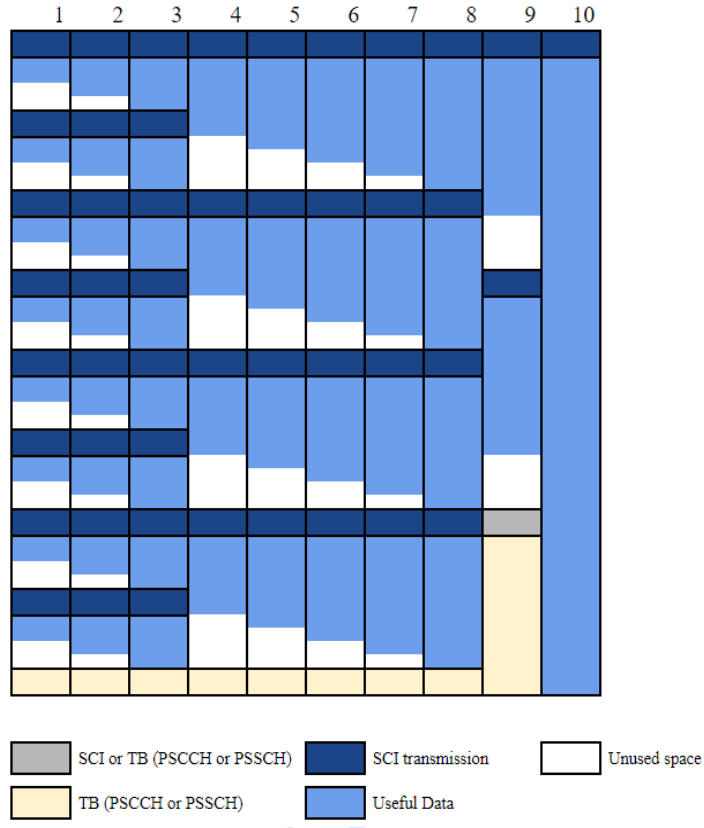

Figure 7: *Effect of growing message size on subchannel occupancy in a 10 MHz grid with 6RBs per subchannel.*

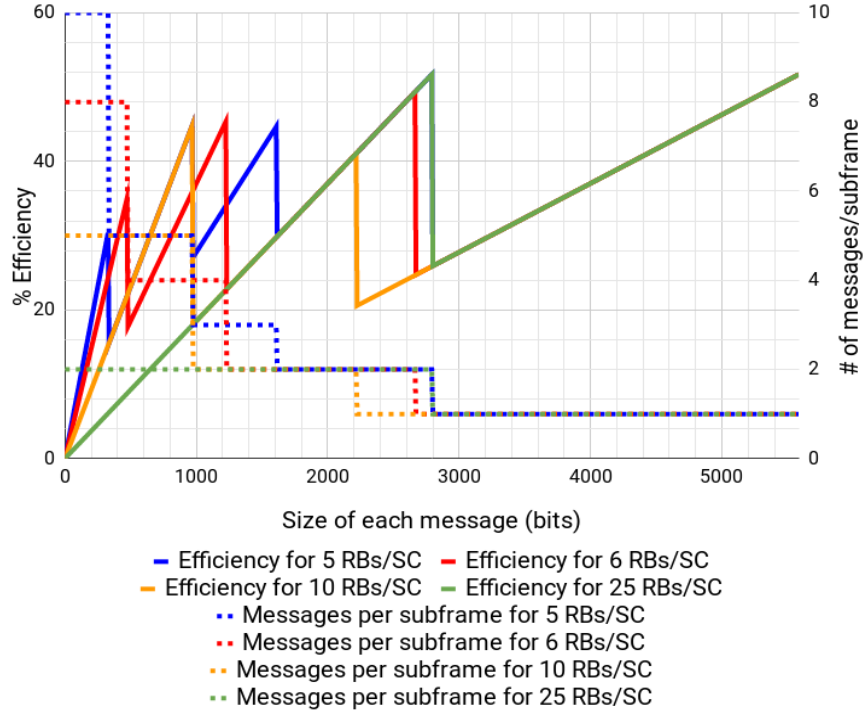

Figure 8: *Message size vs. efficiency for I_{TBS} = 7 (QPSK) at 10 MHz channelization.*

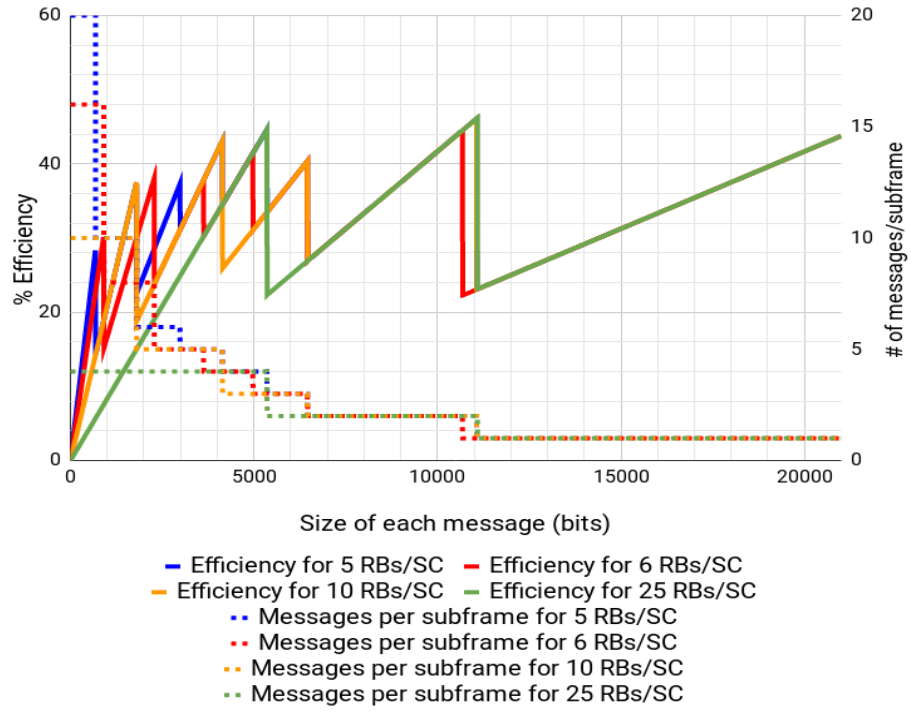

Figure 9: *Message size vs. efficiency for I_{TBS} = 12 (16QAM) at 20 MHz channelization.*