POLITECNICO DI TORINO

**Corso di Laurea Magistrale in
Communications and Computer Network Engineering**

Tesi di Laurea Magistrale

Design and Evaluation of 5G Network Stress Testing Methods for Connected Vehicles

Realizzata in collaborazione di:
Ericsson GbmH
Institute EURECOM

Relatori

Dr. Prof. Carla Fabiana Chiasserini
Firma ....

Dr. Prof. Jérôme Härri
Firma ....

Supervisore Aziendale

Dr. Maciej Muehleisen
Firma ....

Candidato

Giuseppe Serra
Firma ....

A.A.2022

Abstract

The evolution of vehicular mobility led in recent years to the development of a new vision of the road environment, where connected vehicles and road infrastructure share information about their own state and other detected road entities with the final goal of predicting and coordinating the decisions of vehicles and other road users. Although with different goals, various proposed systems like the Digital Road Twin [59, 50] and Local Dynamic Maps [23] incorporate this vision of highly integrated vehicular connectivity, paving the way for future developments of the field.

The introduction of new functionalities and services in the EU market requires standardized certification procedures to ensure compliance with EU technical regulations [22]. These procedures must validate the functional aspects of the vehicle, including the connectivity aspects. Often dedicated trial networks are used to create a controlled testing environment, but the performance measurements obtained from such networks are only realistic if the offered data volume is also realistic. Unfortunately, due to the complexity of the system, the creation of these testbeds can often be not feasible or too expensive. Therefore, solutions to recreate realistic network conditions according to the testing scenarios need to be introduced.

This thesis will focus on the evaluation of several setups for background data generation in laboratory environment, proposing a scalable testbed architecture for Hardware-in-the-Loop (HiL) testing for vehicle connectivity. The goal is to achieve the highest possible realism with the lowest possible setup complexity in terms of hardware resources. Furthermore, a digital-twin based automatic emergency braking system has been developed and tested on the proposed testbed, showing how connectivity issues can impact the performances of such systems.

The thesis is structured as follows: a short introduction to the concepts of Digital Twin, target use-case of this study, and an overview on the state-of-the-art of testing methodologies for vehicular communication is provided in Chapters 2 and 3. Chapter 4 focuses on the technical description of the 5G NR Radio Access Network architecture. Chapter 5 describes the testbed architecture and the implemented functionalities, while Chapter 6 reports the results of the testbed validation. Chapter 7 describes the implementation and testing of the digital twin system providing functionalities of vehicle tracking and automatic emergency braking, focusing on the study of the system behavior with congested communication channel.

Résumés

L'évolution de la mobilité des véhicules a conduit ces dernières années au développement d'une nouvelle vision de l'environnement routier, où les véhicules connectés et l'infrastructure routière partagent des informations sur leur propre état et sur d'autres entités routières détectées, dans le but final de prédire et de coordonner les décisions des véhicules et des autres usagers de la route. Bien qu'ayant des objectifs différents, diverses architectures proposées, comme le Digital Road Twin [59, 50] et les Local Dynamic Maps [23], intègrent cette vision d'une connectivité véhiculaire hautement intégrée, ouvrant la voie aux développements futurs dans ce domaine.

L'introduction de nouvelles fonctionnalités et de nouveaux services sur le marché de l'UE nécessite des procédures de certification normalisées pour garantir la conformité aux réglementations techniques de l'UE [22]. Ces procédures doivent valider les aspects fonctionnels du véhicule, y compris les aspects liés à la connectivité. Souvent, des réseaux d'essai dédiés sont utilisés pour créer un environnement d'essai contrôlé, mais les mesures de performance obtenues à partir de ces réseaux ne sont réalistes que si le volume de données proposé l'est également. Malheureusement, en raison de la complexité du système, la création de ces bancs d'essai est souvent irréalisable ou trop onéreuse. Par conséquent, des solutions pour recréer des conditions de réseau réalistes en fonction des scénarios de test doivent être introduites.

Cette thèse se concentre sur l'évaluation de plusieurs configurations pour la génération de données de fond dans l'environnement de laboratoire, proposant une architecture de banc d'essai évolutive pour les tests Hardware-in-the-Loop pour les aspects de connectivité des véhicules. L'objectif est d'atteindre le plus grand réalisme possible avec la plus faible complexité de configuration possible. En outre, un système de freinage d'urgence automatique basé sur un jumelage numérique a été développé et testé sur le banc d'essai proposé, montrant comment les problèmes de connectivité peuvent avoir un impact sur les performances de tels systèmes.

La thèse est structurée comme suit: une brève introduction aux concepts de Digital Twin, le case-study de cette étude, et un aperçu de l'état de l'art des méthodologies de test pour la communication véhiculaire sont fournis dans les chapitres 2 et 3. Le chapitre 4 est axé à la description de l'architecture du réseau d'accès radio 5G NR. Le chapitre 5 décrit l'architecture du banc d'essai et les fonctionnalités mises en œuvre, et le chapitre 6 rapporte les résultats de la validation du banc d'essai. Le chapitre 7 décrit limplémentatione et le test du système Digital Twin offrant les fonctionnalités de suivi de véhicule et de freinage d'urgence automatique, en se concentrant sur l'étude du comportement du système avec un canal de communication encombré.

Contents

List of Figures

List of Tables

1 Introduction

Mandatory automated emergency calling (eCall) for vehicles sold in the EU creates, for the first time, a situation where a connectivity service becomes relevant for vehicle type approval [10, 32]. For this use case, certification was conducted by driving on open roads using off-the-shelf SIM cards, within available public mobile radio networks, and without any active involvement of the Mobile Network Operators (MNOs). This testing methodology allowed for a realistic assessment within state-of-the-art networks at that time, assuming that network quality will be the same or better by the time of the service introduction. This assumption may not always be realistic, as increased usage of networks due to new services can result in worse Quality of Service (QoS) if rollout and spectrum allocation do not keep up with the increased demand.

With the evolution of the Internet of Vehicles (IoV) and the consequential introduction of new systems such as the Digital Road Twin [59, 50] and Local Dynamic Maps [23], novel testing and certification methodologies need to be defined and standardized, also taking into consideration the network evolution and network user behavior, significantly influenced by MNOs policies.

1.1 Motivation

The introduction of new services and functionalities in the EU market require standardized certification procedures to ensure the compliance to EU regulations [13, 22]. The procedures must validate all the functional aspects, including connectivity. This assumes that certification centers have the necessary means to recreate, in a controlled environment, the network conditions required by regulation for the use-case test. The design of such testbeds has to take into consideration two fundamental aspects:

- the performance results obtained from such networks are only realistic if the offered data volume is also realistic, considering the traffic class and required QoS. In this settings, the design of background traffic becomes a central issue; for example, high-volume background traffic generated through large file down- or uploads would usually receive lower priority than more critical connected driving data, nullifying the testing objective. Therefore, testing requires background traffic generators with equal or higher QoS requirements than the one of the evaluated services.

- several hundred vehicles can be served by a cell, but creating several hundred background traffic generators can often be not feasible or too expensive. Therefore, solutions requiring fewer hardware resources need to be introduced.

1.2 Purpose and Objectives

The first objective of this work is to investigate how realistic network conditions can be recreated in a vehicular testing environment, analyzing how the realism of the generated traffic is conditioned by the employed hardware resources, i.e. radio-modems, computing units, etc. Therefore, the first part of this project has been spent designing and developing a scalable (in terms of hardware resources) and easily deployable testbed set up together with the required traffic generator software. Subsequently, a quantitative analysis of the generated traffic statistical characteristics in function of the employed hardware resources has been realized with measurements of the QoS (delay, throughput) experienced by vehicular applications traffic in a heavily loaded network. The second objective of this project is to analyze the effect of the network conditions on the performances of digital twin systems. For this purpose, vehicle position tracking and automatic emergency braking functionalities have been implemented using a digital twin architecture. The developed testbed was employed for the testing of such functionalities, allowing to measure the kinematic KPIs of the simulated vehicle in various network conditions.

Problem formulation

- How do the employed hardware resources influence the realism of the target network condition?
- What is the QoS experienced by data traffic flows in heavily congested network conditions?
- How does the traffic QoS affect the performances of digital twin systems?

1.3 Thesis Structure

The thesis will be structured as following:

- Chapters 2 and 3 provide an introduction to the concepts of Digital Twin, target use-case of this study, and an overview on the state-of-the-art of testing methodologies for vehicular communication. Chapter 4 will focus on the technical description of the 5G radio access architecture.
- Chapter 5 will describe the testbed architecture and the implemented functionalities, while Chapter 6 reports the results of the testbed validation.
- Chapter 7 will describe the implementation and testing of a digital twin system providing functionalities of vehicle tracking and automatic emergency braking, focusing on the study of the system behavior with congested communication channel.

2 Digital Twin

The continuous evolution of the Internet of Things (IoT) technologies opens new possibilities, where now the focus moves from user level (e.g., solutions adopted by individual consumers or companies) to system-level applications (e.g., solutions adopted by commercial or industrial sectors). Digital Twin is an emerging declination of the IoT concept that has attracted a lot of interest in the industry. All the major cloud providers rolled out significant digital twin capabilities in 2021. Microsoft revealed a digital twin ontology for construction and building management for its Azure Digital Twin platform [4]. Google launched a digital twin service for logistics and manufacturing [7]. AWS launched IoT TwinMaker [2] and FleetWise [1] to simplify digital twins of factories, industrial equipment, and vehicle fleets.

2.1 Definition and Categorization

Although the definition of digital twin varies from publication to publication ([36], [42], [52]), the basic ideas are very similar: a digital twin is a virtual representation of a physical object. Depending on the degree of integration between the physical object and its virtual counterpart, Kritzinger et al. [40] introduce more specific categorization:

- *Digital Model*, when the digital representation of an existing or planned physical object that does not use any form of automated data exchange with its physical counterpart, meaning that a change of state in one of the two representations does not have any direct consequence on the other.

- *Digital Shadow*, when there exists an automated one-way communication from the physical object to its virtual representation, meaning that change in the state in the physical object affects the virtual one, but not vice versa.

- *Digital Twin*, when exists an automated bi-directional communication between the physical and virtual object, thus allowing to track/modify the state of one of the objects by observing/acting on its counterpart.

2.2 Digital Twin in the Automotive Industry

The adoption of the digital twin concept in the automotive industry has been smoothed by the earlier adoption of similar and connected technologies such as Cyber-Physical Systems (CPS) [42] and IoT: the majority of the already established methodologies and/or algorithms answered to similar problems addressed by the digital twins and were developed for systems with a physical entity (i.e., vehicle) and its virtual replica (simulation model/environment). Magargle et al. [44] developed a digital twin for model-driven predictive maintenance and health monitoring of an automotive braking system. Liu et al. [43] introduces a possible improvement to the data collecting and elaboration capabilities of the digital twin by integration with data fusion models: for example, the *sensor fusion* allows combining the data from multiple sensors in the same area to obtain cleaner measurements, while the *sensor and data models fusion* allows improving the quality of the sensor data by integration with specific domain knowledge. An example of sensor and data models fusion is the *Simultaneous Localization and Mapping* (SLAM) algorithm where real-time data from a vehicle with different onboard sensors is combined with cloud-based high-definition map data to allow better estimation of the vehicle location and relative position to road elements and other vehicles [50].

Digital Twin Framework for Connected Vehicles

Figure 2.1 depict an example of digital twin architecture, using as reference the systems introduced in [59] and [58]. In both architectures the separation of the physical and cyber world by introducing a two layer architecture: the communication between the two domains is handled by the communication module using cellular communications, playing a crucial role in the real-time performances of the system.

The *physical world* of the digital twin framework contains all the physical entities and their interactions, including the vehicle and its components, drivers and passengers, roadway infrastructure, and the other users: its main tasks are sensing (or sampling) and actuation. The sensors detect the dynamic states of physical entities, changes in the operating process, or event occurrences, such as vehicle speed, driver's gaze, and traffic light status, and aggregate the measurements under different resolutions. The information is transmitted to the cyber world via the communication module for further processing. On the other hand, the processed results from the cyber world are received (again via the communication module) and serve as the actuation guidance for the entities or processes in the physical world.

The *cyber world* is instead responsible for all the computational effort. It provides an abstract representation of the physical world (i.e. digital replicas of physical entities and processes), as well as several useful functions. Firstly, sensed data from the physical world is cleaned (such as outlier detection and removal, missing data imputation) and fused (including time synchronization). Then, the pre-processed data may be stored in the database (e.g., for digital traceability) or be sent to the data mining knowledge discovery module for further exploration with advanced computational techniques (such as machine learning). The knowledge can be drawn to perform predictive analyses (with the combination of modeling/simulation tools) and find optimal strategies to support decision-making processes. The action decisions are transmitted back to the actuators in the physical world to improve the overall system performance.

Implementation challenges The Digital Twin frameworks, although composed of well understood functional blocks, introduces multiple implementation and integration chal-

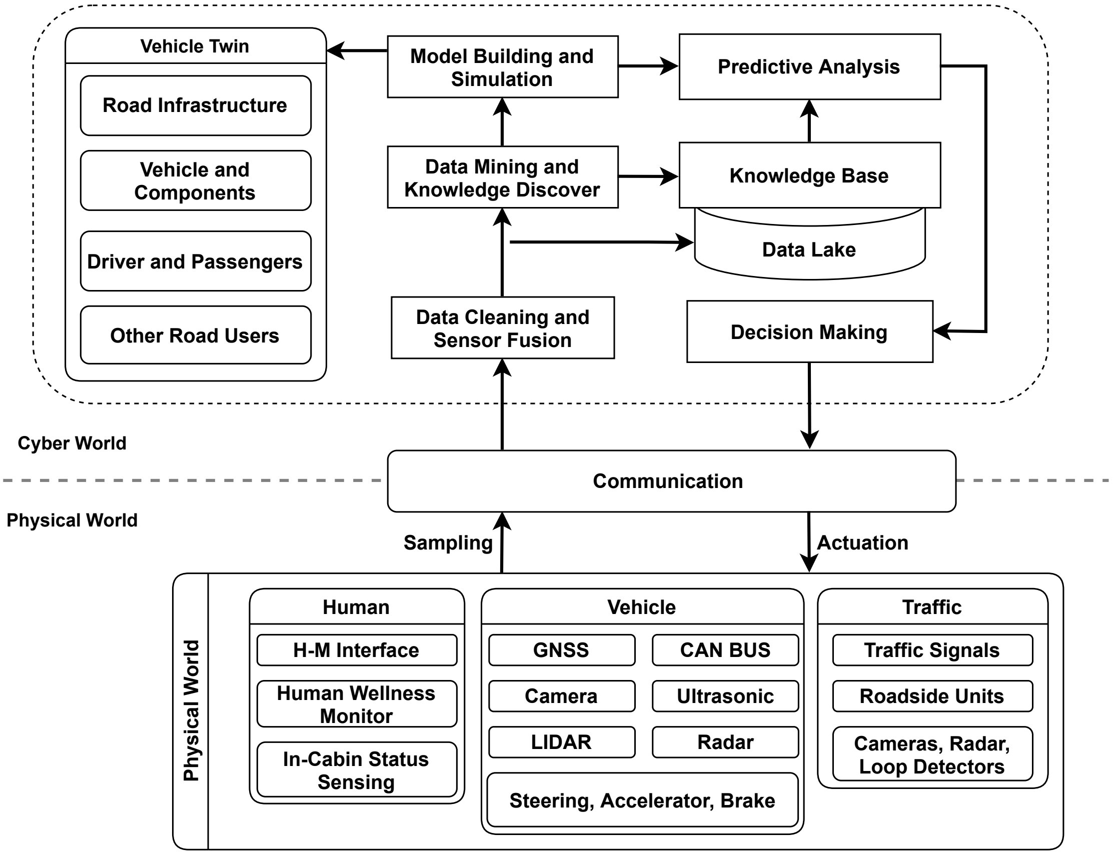


Figure 2.1: Example of a digital twin framework for connected vehicles. As shown in the *physical layer*, sensor data can be provided by various entities to be later correlated during the *cyber layer* processing.

lenges:

- **Data Integration and System-Level Harmonization**, that is the capability of linking heterogeneous devices in the real world and heterogeneous virtual objects on the virtual twin level in a seamless manner. Cross-collaboration between the different applications and fields was identified as the main challenge from Laaki et al. [41] while prototyping a Digital Twin for remote control over mobile networks.

- **Semantic Model design**, that is the logical structure beneath the virtual representation: a Digital Twin needs to model all the components of the physical object and their interaction to represent effectively the reality, always limited to the context of the use-case. Finding the balance between the granularity of the representation and computational cost to maintain it has been identified as one of the main challenges in the architecture design [26].

- **Communication Bottleneck**, representing probably the hardest constraint on the Digital Twin design; the real-time requirements of this technology translates to high communication system requirements, both in terms of latency and sheer traffic capacity. For example, the already mentioned SLAM use-case [50] produces for one hour drive-time of a single-vehicle up to 1 TB of data, requiring an average throughput of 2.2 Gbps. Szabó et. al. [53] provides instead an analysis of the impact of delay in a digital twin architecture for the control of real robot arm: the study highlights how the latency has a direct impact on the system performance, leading to a drop of the 40% in the accuracy of the arm positioning in case of high delay.

In this thesis we will assess the performances of 5G cellular technologies as the communication medium between physical and cyber domains, focusing on highly stressed network scenarios reproduced in a dedicated testbed.

3 Vehicular Communication Testing

The concept of Internet of Vehicles (IoV), expanding on the idea of Vehicular Adhoc NETworks (VANETs), describes the network architecture in which different entities (vehicles, sensors, road infrastructure, pedestrians, personal devices, etc.) exchange real-time information between themselves and with in the road infrastructure. As reported in the previous chapter, understanding how the information flows between real and cyber world is essential for an effective implementation of a digital twin: while a digital twin defines what data to collect and how to elaborate/use it, it's the communication framework (in this case the IoV) that defines how to deliver the information.

The set of communication technologies enabling the IoV fall under the name of *Vehicle-to-Everything* (V2X), where, depending on the communicating entities, it can be subdivided into different applications: *vehicle to vehicle* (V2V), *vehicle to infrastructure* (V2I), *vehicle to pedestrian* (V2P), and *vehicle to network/cloud* (V2N/V2C). Currently, there are two main V2X CT: Dedicated Short Range Communication (DSRC), based on IEEE 802.11 protocols [39], and Cellular V2X (C-V2X), based on 3GPP cellular communication [27, 55]. The IoV framework has to support a variety of applications, from intelligent transportation and intelligent and connected vehicles to automated driving [6]: such variety creates heterogeneous traffic with different requirements in terms of latency, reliability, throughput, user density, and safety of the V2X environment [28]. In this context, testing is an important tool to ensure the safety and security of a vehicle. V2X testing is conducted to identify specification flaws, design flaws, and implementation defects over the entire life cycle, and to determine the root cause of the problem: modeling, analyzing, testing, and evaluating the performances of V2X becomes essential to accelerate the commercialization of automatic driving solution.

3.1 Testing Methodologies

Although the availability of harmonized testing and validation methodology for self-driving vehicles is crucial for the future of automated mobility, only but few examples of procedure standardization have been introduced. Harmonized European Solution For Testing Automated Road Transport, or HEADSTART [16], is EU funded project (European Union's Horizon 2020 research [17]), whose goal is the definition of testing and validation procedure on specific functionalities of Connected and Automated Driving (CAD) function, including the Key Enabling Technologies (KET) such as communications, cyber-security, and positioning. The key idea of the project, described in [15], is the standardization of scenario-based testing procedures: depending on the CAD function, use-case, and testing methodology, relevant scenarios are extracted from a database, describing the environments in which the function under test needs to be tested. Functional requirements for the KETs are also defined for each scenario.

To better understand why different testing methodologies are required, it can be useful to introduce an example of product development workflow: ISO 26262 [21] introduces a framework for functional safety to assist the development of safety-related E/E systems, based on the V-model depicted in Figure 3.1. The V-Model splits the development process into two parts – the left arm of the V consists of requirement analysis, function design, and hardware/software development while the right arm concentrates on the verification and validation activities.

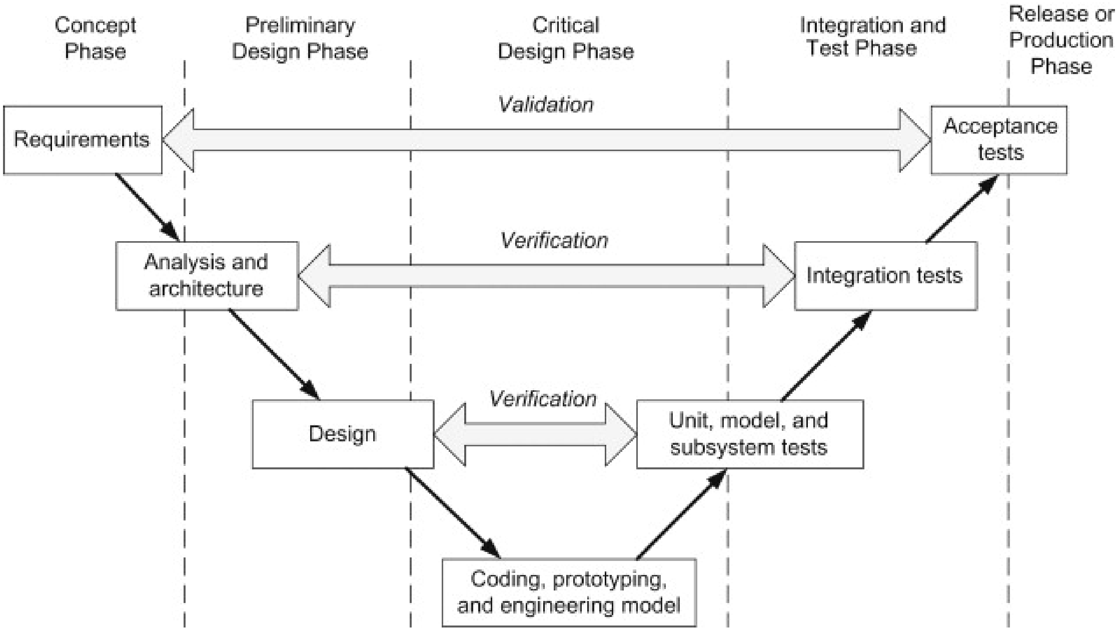

Concept Phase
Preliminary Design Phase
Critical Design Phase
Integration and Test Phase
Release or Production Phase
Requirements
Validation
Acceptance tests
Analysis and architecture
Verification
Integration tests
Design
Verification
Unit, model, and subsystem tests
Coding, prototyping, and engineering model

Figure 3.1: General structure of the V-Model [21]

The various testing methodologies can adapt the testing procedure to the various development stage, allowing to embed testing procedures in the design phase, reducing the length of the development cycle. In the automotive industries, most of the testing methodology belong to one of the following categories [15, 38]: *Virtual Testing*, *X-in-the-Loop Testing*, *Proving Ground Testing*, and *Real World Testing*. Given the scope of this thesis, we will discuss more in detail the X-in-the-loop methodologies. In this context, HEADSTART identifies for the communication KET requirements on conformance, interoperability, and stress-testing. In particular, the requirement for stress-testing, reported in [15], is "Generation of high message loads for the SuT, stress testing the V2X communication system (channel load, Rx buffers, message encoding/decoding, message filtering, application performance/robustness testing)".

X-in-the-Loop Testing

X-in-the-Loop is a family of model-based testing methodology where the X may represent a model, software, or hardware prototype, depending on the level of abstraction of the testing. In general, the X-in-the-Loop methodology consists of two elements: a system under tests (SuT), or device under test (DuT), and a simulated test environment (a.k.a plant), providing inputs (e.g. sensor data, controls input, etc.) and collecting output from the SuT/DuT.

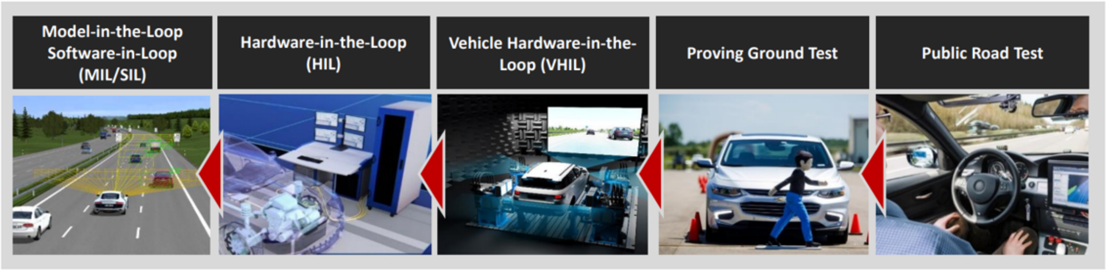

Model-in-the-Loop Software-in-Loop (MIL/SIL)
Hardware-in-the-Loop (HIL)
Vehicle Hardware-in-the-Loop (VHIL)
Proving Ground Test
Public Road Test

Figure 3.2: Instances of test methodologies belonging to the XiL family.

Model-in-the-Loop (MiL) Testing at the MiL level employs a functional model of the SUT that can be tested in an open-loop (without a plant model) or closed-loop (with a plant model and so without physical hardware). The test purpose is functional testing in early development phases in simulation environments. Development and testing at the MiL level are considerably fast as the engineers can quickly modify the control model and immediately test the system.

Software-in-the-Loop (SiL) At the Software-in-the-Loop (SiL) level, testing is still performed in a virtual and simulated environment (like MiL), but the focus is on the SuT code, which can run on the target platform.

Hardware-in-the-Loop (HiL) At the Hardware-in-the-Loop (HiL) level, the software component is fully installed on the hardware platform (e.g. ECUs). The plant is either a real piece of hardware or is some software that runs on a real-time computer with physical signal simulation to lead the controller into believing that it is being used on a real plant. The main objective of HiL is to verify the integration of hardware and software in a more realistic environment.

Vehicle-in-the-Loop (ViL) Vehicle-in-the-Loop (ViL) testing, also referred to as Proving Ground Testing, can be seen as the bridge between model-based testing and real-world test drives, where a working prototype of the vehicle is tested in semi-realistic conditions recreated in proving ground facilities. Szalay et al. [54] provides an overview of the solutions currently employed for the scenario creation: almost all the compared systems use a hybrid approach using mixed-reality technologies, where the Vehicle-under-Test (VuT) interacts with both real and virtual elements, recreated by the joint usage of traffic simulators (e.g. SUMO), simulated sensor reading, etc.

3.2 Examples of C-V2X Testbeds

The inclusion of C-V2X technologies in the testing procedures introduces new design elements in the design of vehicular testbeds. Given the high costs and complexity of testing vehicles in real network environments, lab testbeds are more commonly used in C-V2X testing:

- MOSAIC [11] proposes a modular solution for the creation of simulation environments for MiL/SiL testing of V2X systems. The framework provides a common runtime and interfaces for specialized simulation software (e.g. ns3 and Omnet++ for network simulations, SUMO and PHABMACS for vehicle modeling and traffic simulation, etc.). Although providing realistic results in normal conditions, the simulation-based methods may present flows in extreme scenarios, as highlighted by Wischhof et. al [60]: in their validation study of the Omnet++ LTE module, they discovered that, for overload conditions, the model produces completely different end-to-end delays compared with the measurement results.

- Wang et al.[57] propose function testing system based on HIL methods. The system uses communication devices instead of the network simulator. The testbed architecture, depicted in Figure 3.3, includes the V2X simulation platform, V2X communication module, GNSS simulator, channel interference unit, and CAN bus simulator. The V2X simulation platform performs application scenario simulation, test cases management, test result analysis, etc. It is used to generate dynamic simulation data of the vehicle, such as vehicle speed, position, distance between vehicles, etc., and manages the entire test activity. The V2X simulation platform generates a virtual traffic scenario based on the test cases. The scenario defines a host vehicle (SuT) and at least one remote vehicle (RV), used to assist the SuT to trigger applications. The V2X simulation platform is connected to the V2X communication module to control the transceiver behavior of the V2X communication module. The V2X communication module is used to simulate a RV in a virtual traffic scenario to generate application messages, and then sends it to a SuT through the channel interference unit. Meanwhile, the V2X communication module can also receive application messages sent by the SuT through the channel interference unit.

- 5GAA, in its V2X functional and performance testing report [8], provides another example of an HiL testing, designed for the assessment and comparison of DSRC and C-V2X radio technologies for their suitability to deliver broadcast V2V safety messages. Among the conducted tests, a testbed similar to the one discussed in this

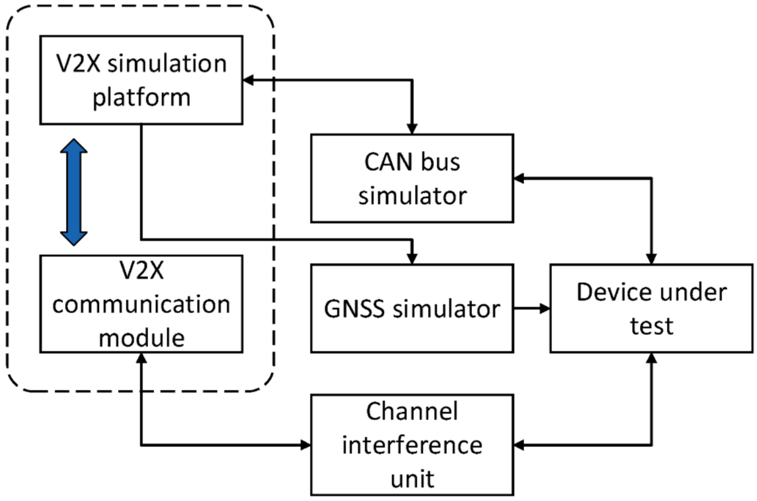


Figure 3.3: HiL Testbed Architecture proposed by Wang et al.[57]

thesis has been realized for the creation of congestion scenarios: the setup allows for the emulation of multiple congesting devices around the DUT, recreating traffic jam conditions. The trial consisted of two devices under test, stationary in the middle of the 1200-meter stretch of highway, transmitting periodically 200 Bytes packets every 100 ms. The system congestion is created by 48 reference lab devices, generating traffic in a ratio 12:1, thus able to emulate the traffic volume produced by up to 576 stationary vehicles. The end goal of the trial was the testing the performances of a congestion-control algorithm, which was able to operate in load of 192% on their system with reasonable packet drops using the PC5 V2X standard while respecting the defined delay budget of 100 ms.

Although useful for early validation, lab testbeds can provide only limited insights on the C-V2X real-world performances. In these cases, dedicated testing facilities allow for the realization of more realistic testbeds with support to C-V2X communication, usually provided by a partner mobile network operator (MNO). Various examples have already been created, in a common effort between the public and private sectors:

- IMEC Smart Highway testbed [45], located in Antwerp, Belgium. The Smart Highway test site is being built on the E313/E34 highway and shall be extended to the (urban) road network providing a mixed environment for testing. The testbed can be considered as a platform for V2X connectivity, Edge computing, and precise positioning.

- Other examples are provided by STAPLE in Deliverable 2.1 [31], with a comprehensive review of technological and non-technological aspects of the most relevant connected and automated driving test sites. Of the considered 39 test-site, only a subset provides support to C-V2X connectivity, for example, AstaZero (Sweden), AURORA (Finland), TRANSPOLIS (France).

4 Mobile Radio Access Network

To better understand the concept of Quality of Service (QoS) and the consequential effects on resource management of vehicular networks, an introduction to the system architecture of current generation of mobile cellular networks is needed. In particular, we will focus on the 5G ”Non-Stand Alone” (NSA) option 3 architecture [33], since it is the architecture used by cellular network on top of which the testbed has been implemented. Furthermore, this introduction will cover the Radio Access Physical Layer (PHY) and Radio Access Data Link Layer (DLL) of 5G New Radio (5G-NR), highlighting their role in the contest of vehicular applications and the improvements with respect to the previous 4G-LTE architecture.

4.1 5G NSA Network Architecture: Overview

The book by Dahlman et al. [30] provides an overview of the 5G NSA system and is used as a primary source of information during this section. In general, cellular network systems are composed by two subsystems, Radio Access Network (RAN) and Core Network (CORE). The RAN handles all radio-related aspect of the system, including radio resource management, scheduling, and retransmission protocols. The CORE, on the other hand, implements all the entities required for the setup of the end-to-end connection, authentication and service charging. The 5G NSA Option 3 architecture uses 5G RAN and its New Radio (NR) interface in conjunction with the existing LTE and Evolved Packet Core (EPC) infrastructure (respectively 4G Radio and 4G Core), thus making the NR technology available without network replacement. In this configuration, while only the 4G services are supported, it is possible to make use of the capacities offered by the 5G New Radio (higher bitrate, lower latency, etc). As depicted in Figure 4.1, Non-standalone option 3 radio access network is composed of eNBs (LTE) as the master node and gNBs (5G) as the secondary node: the control plane is entirely managed by the LTE infrastructure, while the user plane uses the supplementary gNB for their increased capacity. Depending on the management of the user plane, different solutions are standardized, denoted as Option 3, Option 3a, and Option 3x [33].

Although this thesis will focus on the RAN, for the sake of completeness a short description of the EPC structure will be now given. The essential entities composing the EPC are:

- *Mobility Management Entity (MME)*, which deals with the control plane: it handles the signalling related to mobility and security, the connection/release of radio bearers to the UE and UE status tracking (ACTIVE, IDLE).
- *Home Subscriber Service (HSS)*, database containing subscribers information.

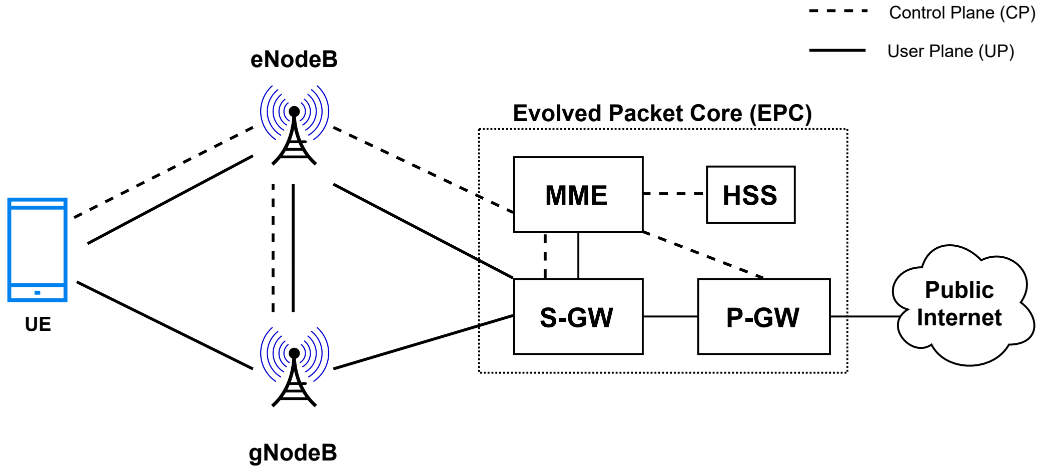


Figure 4.1: 5G NSA Option 3a - Architecture Overview

- *Serving Gateway (S-GW)*, which is in charge of the user data plane, connecting the RAN to the EPC.
- *Packet Data Network Gateway (P-GW)*, which is the point of interconnection between the EPC and the external IP networks.

4.1.1 5G-NR Physical Layer

Orthogonal Frequency Division Multiple Access (OFDMA) is used as multiplexing scheme in 5G NR. Introduced in the previous generation, where it was used only for downlink transmissions, OFDMA in 5G-NR can be used for both uplink and downlink.

OFDMA Grid OFDMA defines the radio frame structure using the underlying Orthogonal Frequency Division Multiplexing (OFDM) to divide the available bandwidth into narrower sub-carriers. Improving on the LTE implementation, 5G-NR supports flexible OFDM numerology allows to create sub-carrier spacings in the range from 15 kHz, as in LTE, to 240 kHz, suitable mostly for mm-wave domain. This flexibility make NR physical-layer specification band agnostic, meaning that it is able to easily adapt to different deployment scenarios, from large cells with sub-1 GHz carriers to mm-wave deployments. In time domain, instead, 5G-NR radio is organized in frames having duration of 10 ms, each divided into 10 equally sized 1 ms subframes. Each subframe is further divided into slots of 14 OFDM symbols. The length of each slot in milliseconds depends on the OFDM numerology, as shown in Figure 4.3.

Resource Units As in LTE, a Resource Element (RE) is the smallest physical resource in NR, consisting of one subcarrier during one OFDM symbol. A Physical Resource

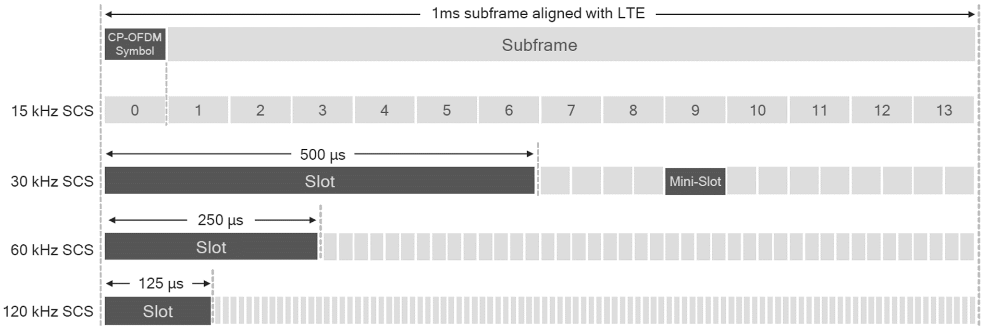


Figure 4.2: Slot duration for the different subcarrier spacings.

Block (PRB) is the smallest resource allocation element assigned by the base station scheduler and consists of 12 adjacent subcarriers. Assignment and grouping of REs to PRBs depends on the scheduler policy and constraints; pairs of Virtual Resource Blocks (VBRs) from the scheduler are mapped to pairs of PRBs. The mapping can be *localized*, where the two VBRs are mapped to adjacent PRBs transmitted consecutively in time, or *distributed*, where the VBRs are mapped to PRBs consecutive in time but on different frequencies, to take advantage of frequency diversity.

Duplex Schemes 5G NR structure supports the separation of uplink and downlink both in time and frequency while maintaining the same frame structure. The supported duplex schemes are:

- Frequency Division Duplex (FDD), where uplink and downlink use different carrier frequency, and thus can operate simultaneously.
- Half-Duplex FDD, where uplink and downlink are separated both in time and frequency.
- Time Division Duplex (TDD), where uplink and downlink transmission use the same carrier frequency and are separated only in time.

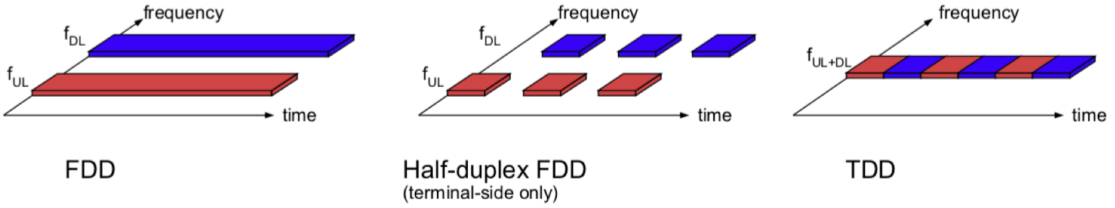


Figure 4.3: Slot duration for the different subcarrier spacings.

4.1.2 5G-NR Data Link Layer

The data link-layer (DLL) protocols enhances the services provided by the PHY layer by increasing reliability, security, and integrity. In addition, it is also responsible for the scheduling and multi-user medium access. As in LTE, the goal of 5G is to provide support for a wide range of different services and data rates; this required the layer to provide the required reliability levels and delays for Internet Protocol (IP) data flows. For example, while Voice-over-IP (VoIP) flows can tolerate delays on the order of 50 ms and packet losses of up to 1 percent on radio link, TCP file downloads are more sensitive to IP packet loss and perform better over links with low bandwidth-delay products. This wide range of requirements can be satisfied by designing the protocol to allow for scalable protool overhead. Overall, this led to the design of the 5G-NR DLL consisting of four sublayers: *Service Data Application Protocol* (SDAP), *Packet Data Convergence Protocol (PDCP), Radio Link Control protocol (RLC) and Medium Access Control protocol (MAC).* Figure 4.4 shows how an IP data flow is handled by the DLL layer down to the PHY layer, summarizing the services each protocol sublayer can provide.

Service Data Application Protocol (SDAP) SDAP is responasable for the mapping between QoS flows in 5G core network and data radio bearer, as well as marking the QoS Flow Identifier (QFI) in uplink and downlink packets. If the gNB is connected to the EPC, as in non-standalone Option 3 deployments, the SDAP is not used, acting as a transparent layer.

Packet Data Convergence Protocol (PDCP) At connection set up, critical and sensitive authentication data is exchanged between UE and gNodeB in order for the MME to grant access to the radio network. Packet Data Convergence Protocol (PDCP) has the responsibility to encrypt this data using advanced ciphering/deciphering keys and algorithms. Other functionalities implemented by the PDCP are:

- IP header Compression, performed using the standardized Robust Header Compression (ROCH). Fields in the IP header that are not changing between packets are stored and removed before transmission over the radio link channel to be reassembled in the receiving PDCP before forwarding. ROHC improves significantly the throughput efficiency especially when the size of actual data in packets is small in comparison to the IP header. ROHC can operate in several modes, all thoroughly described in [62].
- Duplicate removal and (optional) in-sequence delivery.

A PDCP entity is instantiated for each radio bearer, with configuration specified by the MME. In summary, transmitted data packets are fitted with a PDCP header, integrity protected, ciphered, compressed and transferred to the underlying RLC protocol. Received data packets are received from the RLC protocol, deciphered, decompressed and transferred to an application, as for UEs, or the S-GW, as for gNodeBs.

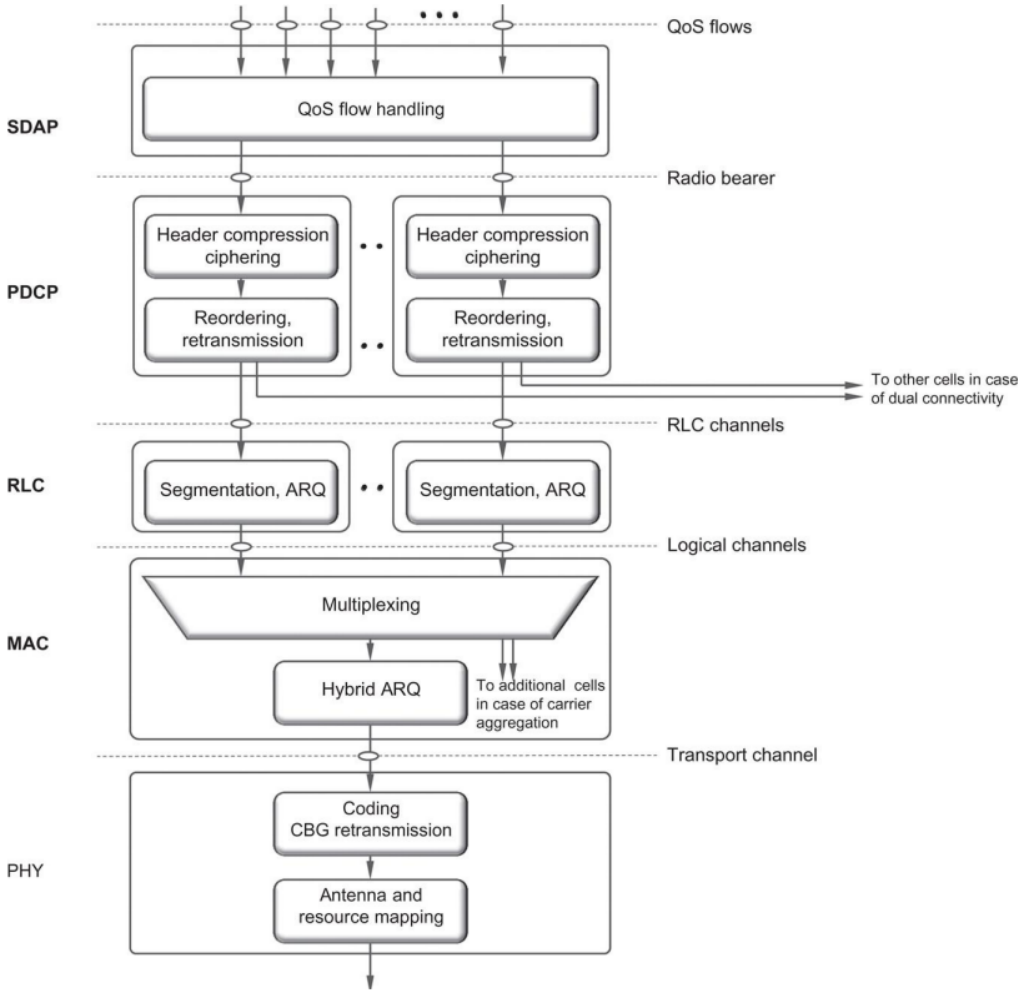

QoS flows
SDAP
QoS flow handling
Radio bearer
PDCP
Header compression ciphering
Header compression ciphering
Reordering, retransmission
Reordering, retransmission
To other cells in case of dual connectivity
RLC channels
RLC
Segmentation, ARQ
Segmentation, ARQ
Logical channels
MAC
Multiplexing
Hybrid ARQ
To additional cells in case of carrier aggregation
Transport channel
PHY
Coding
CBG retransmission
Antenna and resource mapping

Figure 4.4: 5G-NR Radio Link Protocol Architecture, as described in [29]. In case of NSA deployment, the *Service Data Application Protocol* (SDAP) sublayer becomes transparent, since not supported in the EPC.

Radio Link Control (RLC) As for PDCP, the MME specifies configurations for each and every Radio-Link Control (RLC) entity and radio bearer. The RLC protocol can operate in three modes:

- Acknowledged Mode (AM), implementing segmentation/concatenation, duplicate detection, and retransmission handling. All transmitted packets are acknowledged by the receiving RLC, and retransmitted if not acknowledged using the Automatic Repeat Request (ARQ) control.
- Unacknowledged Mode (UM), implementing the same functionalities of the AM mode, excluding ARQ retransmission.

- Transparent Mode (TM), in which packets simply pass through the protocol without any changes, meaning that no segmentation/concatenation is operated on the packet.

In summary, transmitted data packets are segmented, fitted with an RLC header and transferred to the underlying MAC protocol. Received data packets are received from the MAC protocol, acknowledged (RLC-AM), concatenated and transferred to PDCP.

Medium Access Control (MAC) MAC represents the multiplexing stage of the DLL layer; whereas for each radio-bearer a PDCP and RLC entities are instantiated, the MAC entity is common for all channels.The MAC protocol also implements, in addition to the ARQ recovery of lost packets used in the RLC protocol, a Hybrid ARQ (HARQ) with fast retransmission of transport blocks. These two protocols with ARQ packet loss recovery can provide a low probability of packet loss in an efficient manner.

4.1.3 Scheduling in 5G-NR

Scheduling represents one of the key component in a radio network, determining how radio channel resources are shared among multiple users. Multiple UEs are likely to be connected to each base station and the number of available resources is limited. The scheduling algorithm controls how data is transferred over each channel, based on channel quality and available resources. Unlike previous 3GPP mobile communication architectures, e.g. UMTS and GSM, which provided guaranteed bit rates by pre-allocating radio resources statically to dedicated channels, 5G-NR and its predecessor LTE provide QoS by dynamically scheduling users on shared channels, both in uplink and downlink.

It is important to notice that 3GPP standards do not include the scheduling behaviour: while in the EPC several default QoS solutions suitable for VoIP, signaling traffic, and Internet access have been standardized, in the RAN it is responsibility of the gNB implementation, and consequently of the scheduler implementation, to assign radio resources so that radio bearers obtain the requested QoS. Only a set of supporting mechanisms are standardized, on top of which vendor-specific strategies are implemented.

Downlink Scheduling

Upon data arrival at gNB RLC queues, the downlink scheduler will perform the resource assignment and communicate to the UE the allocation on the Physical Downlink Shared Channel (PDSCH) through the Physical Downlink Control Channel (PDCCH). In most cases the scheduling assignment is transmitted just before the data, but the timing information on the assigment can also schedule OFDM symbols later in the slot or in later slots.

Scheduling Decision Parameters Although the information needed by the scheduler depend on the specific algorithm, most schedulers need information about at least:

- Channel conditions at the device, including spatial-domain properties; typically the Channel State Information (CSI) reports from the device represent the main source of information for channel estimation, reporting the measured channel quality in time, frequency and spatial domain. Given the availability of detailed channel-quality information from the UE, the scheduler can perform channel-dependent scheduling, improving the overall system capacity.

- Buffer status of the different data flows: for downlink scheduling, buffer status and traffic priorities are easily obtainable since the scheduler and the transmission buffers are located in the same node. Prioritization of traffic flows is purely implementation specific.

- Properties of the different data flows, including amount of data pending retransmission, fairness indicators, etc.

The OFDMA structure allows for efficient abstraction of the available channel resources in the form of Virtual Resource Blocks (RBs). The resource allocation can be changed dynamically once per subframe, together with Modulation and Coding Scheme (MCS), and other physical level options (MIMO, beamforming, etc.).

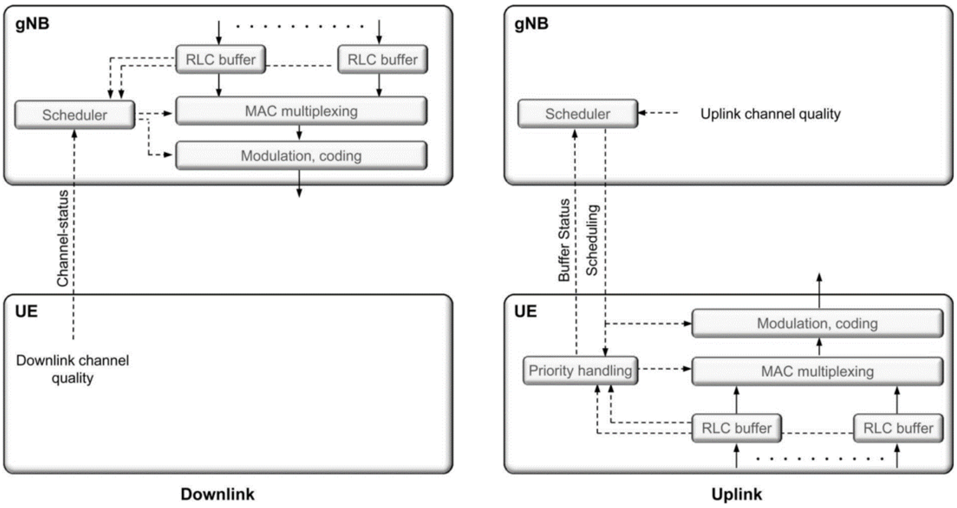


Figure 4.5: Downlink and uplink scheduling in 5G-NR

Uplink Scheduling

Upon data arrival at UE RLC queues, the UE requests an UL grant by sending a Scheduling Request (SR) to the gNB. Then, the gNB sends the UL grant to indicate the scheduling opportunity for the UE to transmit. To notice that the first scheduling assignment may not be sufficient for the complete UL data transmission, since the gNB does not know the accurate requirement (e.g., buffer occupation) at the UE yet. In this regard, since this is implementation-specific, we assume that the first scheduling opportunity consists of the minimum number of symbols that permit at least a 4 bytes transmission (equals to 1 OFDM symbol in good radio conditions). After receiving the UL grant, the UE performs the data transmission in the allocated resources, which may contain UL data and/or Buffer Status Report (BSR). If a BSR is received, the gNB knows the UE RLC buffer status and can proceed with another UL grant to account for the remaining data.

Transmission Grants Uplink data transmission can only take place if the devaice has a valid grant. If a device is scheduled for transmission, it will receive a scheduling grant indicating the set of resources to use in the uplink shared channel as well the associated transport format.

Scheduling Requests A scheduling request is a flag raised by the device to request uplink resources to the uplink scheduler: when data arrives in the transmit buffer on the device and the device has no valid grant, the device transmits a scheduling request at the next possible instant and the gNB can assign a grant to the device upon reception of the request. The request is transmitted on dedicated and periodically allocated resources on the Physical Uplink Control Channel (PUCCH) and handled by a dedicated scheduling-request mechanism, the identity of the device requesting to be scheduled is implicitly known from the resource upon which the transmission was sent. Although LTE used a similar approach, NR supports configuration of multiple scheduling requests from a single device, providing information not only on the presence of data to be transmitted, but also on the type of data, allowing the gNB to differentiate the handling of different flows: for example, the gNB may want to schedule for transmission of latency-critical information but not for non-latency-critical data. A device which has not been configured with dedicated scheduling request resources relies on a random-access mechanism, using a contention-based mechanism to request resources. Contention based designs are suitable for situation where there is a large number of devices in the cell having low traffic intensity.

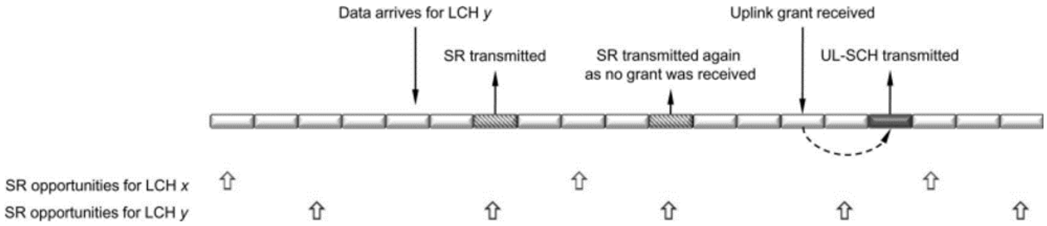

Data arrives for LCH y
Uplink grant received
SR transmitted
SR transmitted again as no grant was received
UL-SCH transmitted
SR opportunities for LCH x
SR opportunities for LCH y

Figure 4.6: Scheduling request and transmission grant in uplink transmission

Buffer Status Reports Devices that already posses a valid transmission grant do not need to request uplink resources; however, information regarding the buffer status and power availability are required by the uplink scheduler to assign future resources. Different formats of buffer reports are defined by standard, allowing for different information granularity. A buffer status report transmission is periodic, but can also be triggered by the arrival of new data with higher priority than the currently in transmission buffer.

4.2 Upper-Layer Protocols for Vehicular Communications

In this section we will introduce the transport layer protocols, application level protocols and application messages used in the testbed realization.

4.2.1 Transport Layer Protocols

Transport protocols reside on top of the IP protocol. Transport protocols run over the best-effort IP layer to provide a mechanism for applications to communicate with each other without directly interacting with the IP layer. In the IP protocol stack, the most widely used two transport protocols are User Datagram Protocol (UDP) and the Transport Control Protocol (TCP).

User Datagram Protocol (UDP) UDP provides a best-effort datagram delivery service. This mechanism is best-effort because the underlying IP network does its best to deliver the datagram, but does not guarantee that the datagrams are delivered at the destination. There are also no guarantees that the datagrams are delivered in the same order as they were sent. These shortcoming are balanced by the very low overhead for both header size (8 bytes, on top of 20 bytes of IP header) and protocol logic.

Transport Control Protocol (TCP) The Transmission Control Protocol (TCP) Complements IP's deficiencies by providing reliable, stream-oriented connections. TCP adds a great deal of functionality to the IP service it is layered over:

- *Reliable and in-order delivery* Sequence numbers are used to coordinate which data has been transmitted and received. Using acknowledgement mechanisms, TCP will arrange for retransmission if it determines that data has been lost.
- *Congestion control* TCP will dynamically adjust its operation to maximize throughput without overloading the network.
- *Flow control* TCP manages data buffers, and coordinates traffic so that the receiving buffers will never overflow. Fast senders will be stopped periodically to keep up with slower receivers.

The introduction of these features is supported by a more complex protocol logic and requires higher overhead in terms of header size (min. 20 bytes on top of 20 bytes of IP header).

4.2.2 Application Layer Protocols

Message Queue Telemetry Transport (MQTT) MQTT [20] is a client-server publish/subscribe messaging transport protocol. Designed to be light weight and easy to implement, it is used in a wide variety of situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT). The protocol runs over TCP/IP, providing a simple, message-oriented interface to the TCP sockets. Other features are:

- Use of the publish/subscribe message pattern which provides one-to-many message distribution and decoupling of applications.
- A messaging transport that is agnostic to the content of the payload and three qualities of service for message delivery: "At most once" (QoS-0), "At least once" (QoS-1), "Exactly once" (QoS-2).

4.2.3 Cooperative Intelligent Transport Systems (C-ITS) Messages

On top of the communication technologies belonging to the IoV ecosystem, major standard development organisations (e.g. ETSI, SAE, etc.) work on an international or regional level for the standardization of the architecture of Cooperative Intelligent Transport Systems (C-ITS). As descibed by ETSI, C-ITS embrace a wide variety of communications-related applications intended to increase travel safety, minimize environmental impact, improve traffic management and maximize the benefits of transportation to both commercial users and the general public [12]. Although different messages and application protocols are standardized depending on the use-case, in this section we will focus on the Cooperative Awareness Messages (CAM), standardized by ETSI: keeping in mind that also equivalent standards such as SAE's Basic Safety Message (BSM) exists, the information on vehicle status delivered by CAM messages will be used in Chapter 7 for the implementation of a vehicle digital twin.

Cooperative Awareness Messages (CAMs)

Cooperative Awareness Messages (CAMs) [24] contains status and attribute information of the originating ITS-S: for example, the status information of vehicle ITS-Ss includes time, position, motion state, activated systems, etc. and the attribute information includes data about the dimensions, vehicle type and role in the road traffic, etc. The survey on CAM statistics created by the CAR2CAR consortium [25] provides a general idea of the traffic characteristics in terms of packet generation frequency and message size, useful for the characterization of the testbed traffic sources.

Generation Frequency and Timing Requirements The CAM generation frequency defines the time interval between two consecutive CAM generations. Considering the requirements as specified by ETSI [24], the upper and lower limits of the transmission interval are set as follows:

- The CAM generation interval shall not be inferior to $T_{CAM_{Min}} = 100\,\text{ms}$. This corresponds to the CAM generation rate of 10 Hz.
- The CAM generation interval shall not be superior to $T_{CAM_{Max}} = 1000\,\text{ms}$. This corresponds to the CAM generation rate of 1 Hz.

Within these limits the CAM generation is triggered depending on the originating ITS-S dynamics and the channel congestion status.

Use Case	min Frequency [Hz]	max Latency [ms]
Emergency Vehicle Warning	10	100
Slow Vehicle Indication	2	100
Intersection Collision Warning	10	100
Collision Risk Warning	10	100
Speed Limits Notification	2	100

Table 4.1: CAM Messages minimum generation frequency and maximum expected latency depending on the use case.

Data Format and Message Size The CAM consists of a collection of data elements that are arranged in a hierarchical order:

- mandatory information i.e. a heading indicating the vehicle ID (referred to also as station ID).
- basic data like a timestamp and position, status data such as vehicle static and dynamic information (speed, heading, acceleration and curvature, etc.).
- optional attribute data container for low-dynamic information, like vehicle role or category.
- optional container relating to vehicle category details (public transport, rescue), signature and certificate.

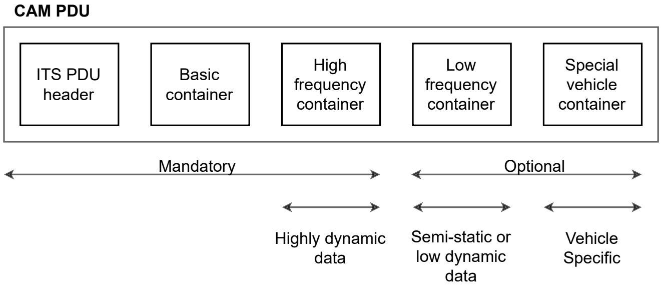


Figure 4.7: CAM Protocol Data Unit

The average CAM sizes is typically around 350 Bytes and the approximate CAM size distributions ranges between 200 Bytes and 750 Bytes. Typically, 30% of the messages are below 300 Bytes and more than 30% of the messages are above 450 Bytes.

5 Testbed Architecture

In this chapter we will focus our attention on the developed testbed architecture, providing a list of the architecture components and their logic, that will be used in Section 6 and 7 for the realization of the measurements campaigns in specific scenarios.

Many example of laboratory testbeds are present in the literature, used as preliminary step for the deployment of real world testbeds or to test the goodness of fit of theoretical models and simulations with real data. In [60], a testbed composed of and an Evolved Packet Core (EPC) and one gNodeB has been setup to measure UDP end-to-end delay and throughput between two UEs. Comparing the measurement data with the simulation results the open-source system-level simulator SimuLTE, the authors have discovered substantial disagreement between simulation and reality in case of overloaded network. In [62], the authors perform quality of experience (QoE) measurements of Voice-over-IP (VoIP) and video streaming applications using both an indoor and outdoor testbed; by using fading simulators, the indoor setup emulate the real world radio transmission conditions.

5.1 Traffic Generators Architecture

Traffic generators represent the basic element of our setup. Figure 5.1 depicts the main components of the entity and their interaction. At deployment and during run-time, the `Cluster Manager` will populate the `Flow Pool` according to the selected mobility model. The `Cluster Status Logger` will instead collect systems statistics (number of flows, flow lifetime, etc.) for diagnostic purposes. The `Flow` entity has the role of creating data traffic according to the specified configuration and it is composed by three elements (Figure 5.2):

- **Inter-Arrival Time (IAT) Generator**, configured with the requested IAT distribution, allows to sample from such distribution and use the IAT sample to trigger the Message Generator at specific times, acting as a timer.

- **Message Generator**, configured by setting the *packet length* and the *payload content*, generates the message when triggered by the IAT Generator and relies on the Traffic Socket Manager for its transmission.

- **Transmission Socket Manager**, configured by setting the requested protocol and eventual protocol level settings, e.g. QoS, it provides a protocol independent interface to transmit the generated packets.

Considering each `Flow` as an independent UE in the system, the `Cluster Manager` allows to integrate user mobility effects in traffic generation process, through the possibility of allocating and de-allocating traffic flows at run-time according to specified model. The list of all supported setting for each component is reported in Table 5.1.

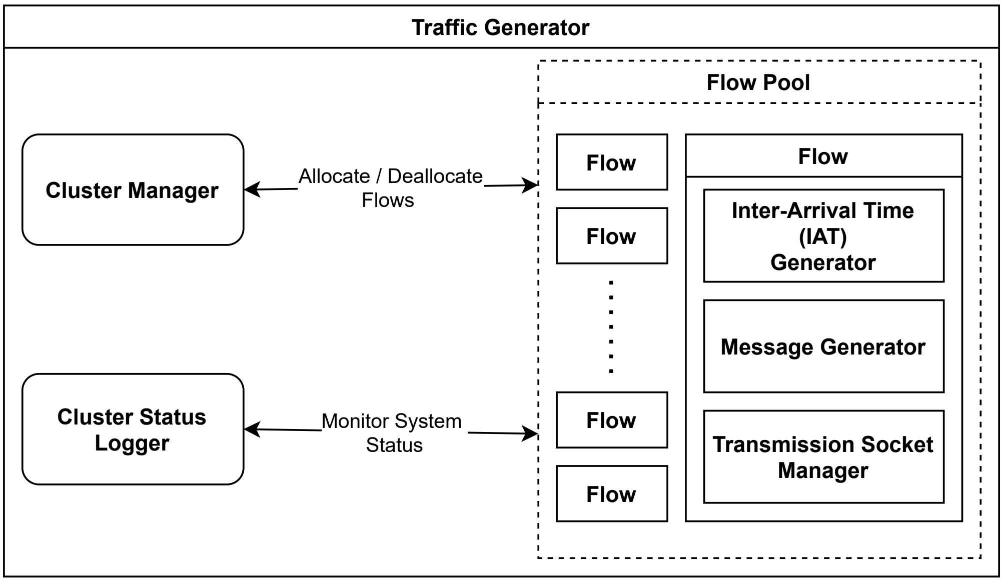


Figure 5.1: Traffic Generator Architecture.

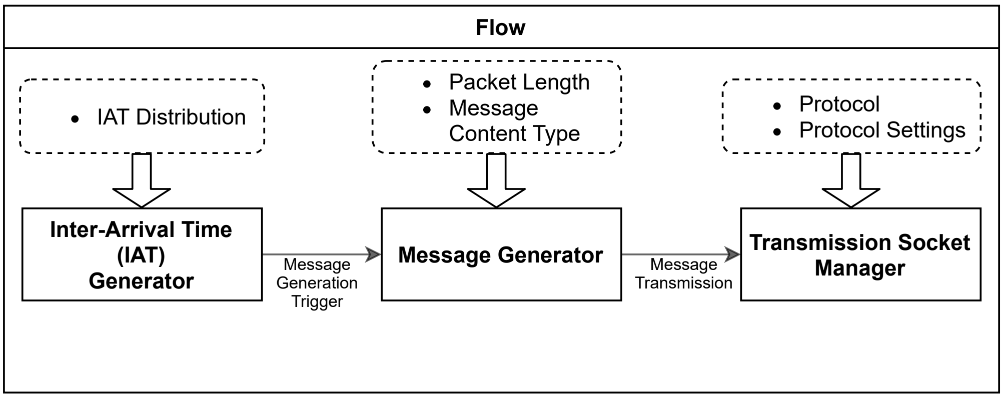


Figure 5.2: Flow Entity Architecture.

5.2 Testebed Architecture

Figure 5.3 shows the schematics diagram of the testbed. All measurements take place in a shielded lab room, therefore providing a highly reproducible wireless environment free of the interference of external sources.

Component	**Profile**	**Settings**	**Description**
Cluster Manager	Static Cluster	Initial State S	Static number of flows allocated, no de-allocation during runtime.
	Markovian Queue Cluster	Initial State s, Arrival Rate $\lambda(S)$, Service Rate $\mu(S)$	System dynamics follow a queue model with state dependent $\lambda(S)$ and $\mu(S)$.
	Finite State Space MC Cluster	State Space S, Transition Matrix P, Sojour Rates V	System dynamics follow the Markov Chain MC(S, P, V).
IAT Generator	Deterministic	Average IAT	Deterministic packet inter-arrival time.
	Exponential	Mean IAT	Exponentially distributed packet inter-arrival time.
Message Generator	Fixed Packet Size	Packet Size L	Fixed size packets containing timestamp and sequence number.
	GPS Messages	Packet Size L, GPS Trace File	Fixed size packets containing timestamp, sequence number and mock GPS position according to the provided file.
Socket Manager	UDP	Server Addr.	Packet transmission using UDP datagram to specified UDP Server.
	MQTT	Broker Addr, Topic, QoS	Packet transmission using the MQTT protocol, with specified QoS.

Table 5.1: Traffic Generator Supported Settings.

The used Non-Standalone 5G Network allows to perform measurements with both 4G and 5G radio access technologies. It is composed of a base station (gNodeB) and a locally hosted Evolved Packet Core (EPC): which includes the Mobility Management Entity (MME), Packet Data Network Gateway (P-GW), the Serving Gateway (S-GW), Home Subscriber Server (HSS). PC1 and PC2 have the role of emulating the User Equipment (UEs); for each connected radio-modem, the PCs will instantiate reserved a reserved namespace, allowing the deployment of multiple `Traffic Generators` sharing the same computational resources (see Subsection 5.2.1). The direct link between the core network and the traffic receiver, through the 1 Gbps switch completes our measurement loop, highlighted in blue in the figure. This setup allows to independently measure the uplink and downlink traffic statistics, assuming that the delay introduced by the switch is negligible compared to the wireless channel delay. Time synchronization between the PCs guarantees timestamps consistency between packets generated by different PCs: the NTP Server on PC1 acts as a time reference for the other units (NTP clients), allowing accurate measuring of the traffic delay-related features using the same traffic receiver for all the traffic generators.

A third control PC deploys both traffic generators and receivers while also collecting the measurement data from the two units. Table 5.2 lists the technical descriptions of the elements employed in the testbed.

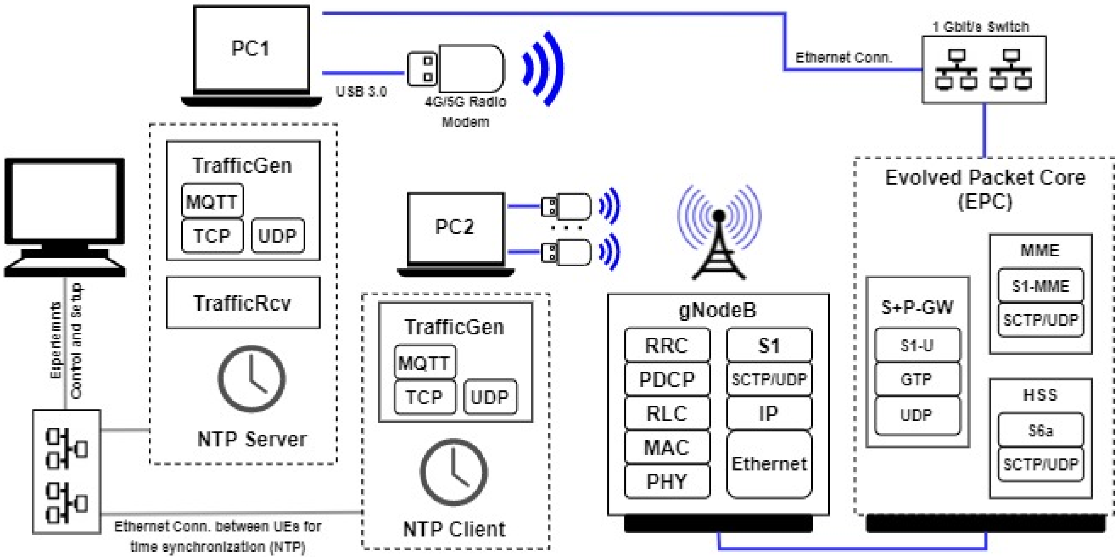


Figure 5.3: Structure of the testbed. In blue are indicated the links traversed by measured traffic.

Element	ID	Description
PC	PC1/PC2	Intel i5 6xxxx, 8GB RAM, 2xUSB3.0, OS: Debian 11.0
UE	RM0, RM1	Quectel RM500Q-GL - Product Specifications [5].

Table 5.2: Technical specifications of the elements of the testbed.

5.2.1 Namespaces and Flow Routing

To allow better usability and scalability of the testbed, an automated system has been developed to deploy independent traffic generators on the same PC, bounding the generated traffic to specific interfaces in order to control the load of each radio-modem. This abstraction is provided by the Linux kernel functionality known as network namespaces (`NS`) [18]: a network namespace is a logical copy of the network stack from the host system, thus having its own IP addresses, network interfaces, routing tables, etc. Figure 5.4 depicts the system functional structure:

- at creation, each network namespace is assigned to a radio-modem interface (`wwan`). The traffic generated in the `NS`, from now on referred to as `wwan-NS`, will re-routed internally to the associated radio interface. The reachability of the `NS` is ensured by setting up NAT port forwarding.
- when deploying a traffic generator, it is possible to select the hosting NS. Traffic generators deployed in different `NSs` will have independent network stacks. Additionally, if the association (`NS`,`wwan`) is unique, the traffic generators will also be decoupled at MAC level, resulting virtually in distinct devices.
- traffic receivers are hosted by the `server-NS`, present on PC1. The interface `eth0`, connected to the EPC, has been assigned to this `NS`.

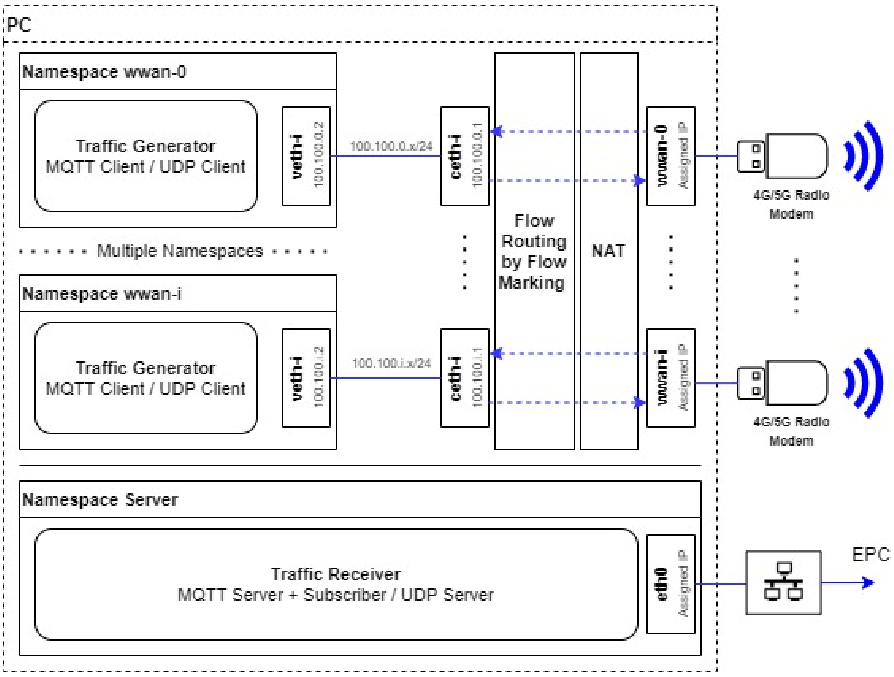


Figure 5.4: PC namespaces setup. The traffic produced in each namespace is forwarded through the assigned radio-modem interface, allowing for simple control over the introduced network load.

5.2.2 RAN Settings

Channel Capacity The channel capacity C is determined by both the observed radio conditions and the gNodeB configuration. The employed gNodeB operates in Frequency Range 1 (FR1, channel frequency $\leq 6\,\text{GHz}$) in Time Division Duplex (TDD). The TDD pattern is `DDDSU`: `D`and `U` represent, respectively, the allocated slots allocated for downlink and uplink transmissions, while `S` denotes special slots containing a Guard Period (GP). Given the small cell size, the GP consists of 2 OFDM symbols, assigning the remaining slot symbols as 10:2:2 (D:GP:U). Overall, the system resource ratio between downlink and uplink is approximately 4 : 1. The allocated channel bandwidth BW is 20 MHz, with sub-carrier spacing of 15 kHz (numerology $\mu = 0$). The complete gNodeB settings are reported in Table 5.3.

Parameter	Symbol	Value
UL/DL channel bandwidth	BW	20 MHz, TDD
Numerology	μ	30 kHz
TTI duration	d_{TTI}	0.5 ms
TDD Pattern		`DDDSU`
S Slot partition		`10:2:2`
MIMO Layers	v_{DL}, v_{UL}	4 (Downlink), 1 (Uplink)

Table 5.3: Summary of RAN Parameters

Given the gNB and UE parameters, ETSI TS 138-306 [34] provides an approximation to maximum channel capacity C_{max}, in Mbps, for a given band and a single carrier, reported in Equation 5.1:

$$C_{max} = 10^{-6} \cdot v_{Layers} \cdot Q_m \cdot R_{max} \cdot \frac{N_{PRB}^{BW,\mu} \cdot 12}{T_s^{\mu}} \cdot (1 - OH) \tag{5.1}$$

where:

- v_{Layers} is the number of MIMO supported layers [51]. In our case, the supported v_{DL} for the downlink and v_{UL} for the uplink are 4 and 1, respectively.
- Q_m is the maximum supported modulation order given by higher layer parameters. Assuming good channel conditions, we will consider the modulation order corresponding the higher end of values reported in ETSI TS 38 214, in Table 5.1.3.1-1 [35].
- T_s^{μ} is the average OFDM symbol duration in a subframe with numerology μ.
- $N_{PRB}^{BW,\mu}$ is the maximum resource block allocation in bandwidth BW for numerology μ, defined in [35].

- OH is the overhead coefficient. For FR 1, OH_{DL} for downlink channel is 0.14, while OH_{UL} for the uplink channel is 0.08.

With the reported settings and parameters, in our system the maximum downlink channel capacity C_{DL} we can observe is 262 Mbps, while for the uplink C_{UL} is 18 Mbps.

6 Testbed Validation

6.1 Traffic Generators Validation

The validation of the created traffic generators is an essential step in the testbed development, required to understand their operating region and how technical limitation can affect them. Although there is no consensus in the research community on an unique validation procedure, [47] presents an overview of the state-of-the-art metrics used for validation purposes. In the context of this work, we are interested in the distribution of the *Packet Inter-Arrival Times* (IAT), also referred to as *Inter Packet Times* (IPT), and the flow pool dynamics used to emulate user mobility.

6.1.1 Inter-Arrival Time Distribution

In our traffic generator, flows are implemented as threads; in this setting, the inter-arrival times can be realized by using functions that deactivate the thread for the required time, leaving to the operating system scheduler to reactivate the thread. Although easily adaptable to many systems, introduces a direct dependence on the OS scheduler performances. To quantify the impact of such dependency, the traffic generators have been tested with a set of different configurations, as reported in Table 6.1. Separate trials with deterministic IATs and exponential IAT distribution have been conducted with various values of requested $\mathbf{E}$[IAT] to analyse the effect of the scheduler dependency on the empirical IAT distribution. The IAT realizations have been measured using the logs of `tcpdump`, a packet capture utility.

Testbench Parameter	Setting
Channel Capacity	1 Gbps
Parallel Flows	1, 10, 30, 50, 100
IAT Distribution	Deterministic, Exponential
Requested $\mathbf{E}$[IAT]	$[10^{-3}, 10^{-1}]s$
Packet Size	1250 B
Protocol	UDP

Table 6.1: Summary of the evaluated traffic generator testbench configuration.

Figure 6.1 shows the performances of the traffic generator for a growing number of parallel flows: on the x-axis it is represented the mean value of the requested IAT for each flow, while the y-axis reports the actual average IAT measured. As expected, the non-realtime nature of the employed scheduler does not allow to have guarantees on the deadlines set by each flow. As the number of flows increases, the scheduler is not able to maintain the requested performances for the IAT generation.

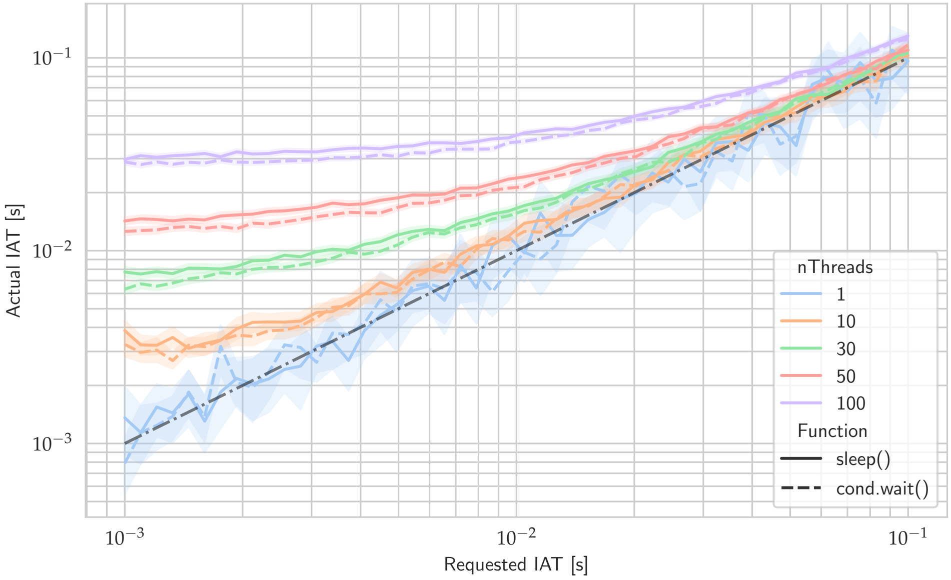


Figure 6.1: Realizations of the Packet Inter-Arrival Times (IAT), conditioned on the flows population and employed function.

These phenomena have implication on the maximum data volume that each generator can create, where for IAT ≈ 1 ms the scaling provided by the number of flows is hindered by the IAT generation accuracy. Figure 6.2 show both the traffic produced by a single flow and the aggregate traffic produced by all the flows in function of the flow mean IAT. As expected from previous results, the increase in the number of flows causes a decrease in the traffic volume produced by each single flow, thus creating an upper-bound on the maximum amount of traffic that a generator can produce.

Deterministic IATs

Figure 6.3 shows the empirical cumulative distribution function (eCDF) of the IAT realizations for the traffic generator configured to recreate a process with deterministic IAT. It is noticeable how the product between the number of parallel threads (nThreads) and the requirements on the mean IAT ($\mathrm{E}[IAT]$) influence the distribution, resulting in heavier tails in the case of large nThread $\cdot$ $\mathrm{E}[IAT]$ due to the scheduling bottleneck. The scheduler creates longer IAT due to not meeting the thread reactivation deadline, causing the distribution mean value to shift to larger values.

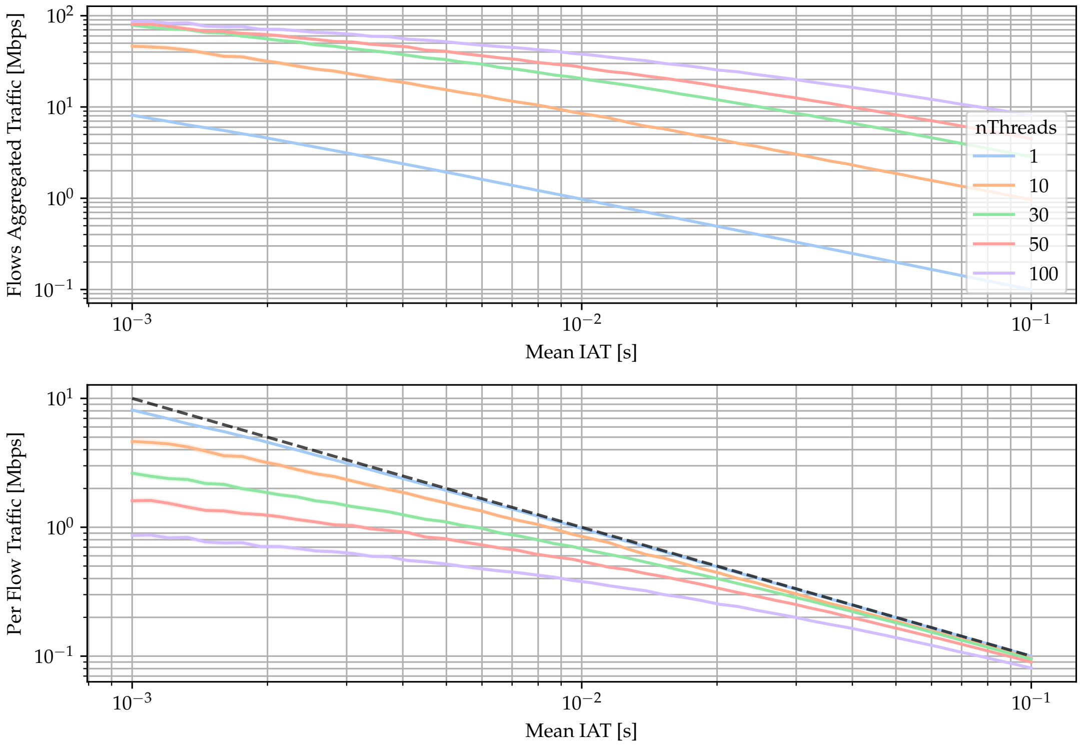


Figure 6.2: Generated per-flow traffic and aggregated traffic, considering fixed packet size of 1250 Bytes, conditioned on the flow population.

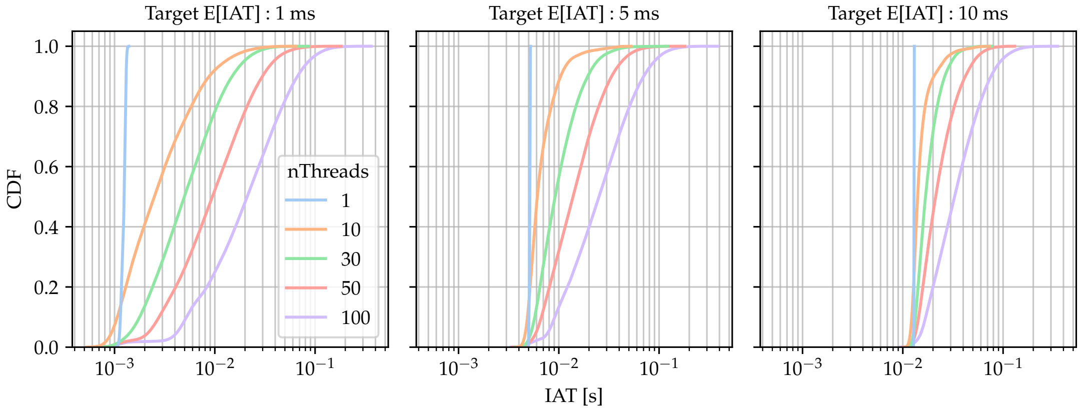


Figure 6.3: eCDF of the IAT realizations, under the assumption of deterministic inter-arrivals.

Exponentially Distributed IATs

Figure 6.4 shows the eCDF of the IAT realizations produced by the traffic generator configured to recreate a target arrival process with exponentially distributed IAT. The empirical distribution, although still approximating an exponential distribution, presents a shifted mean with respect to the theoretical target, for the same reasons explained previously.

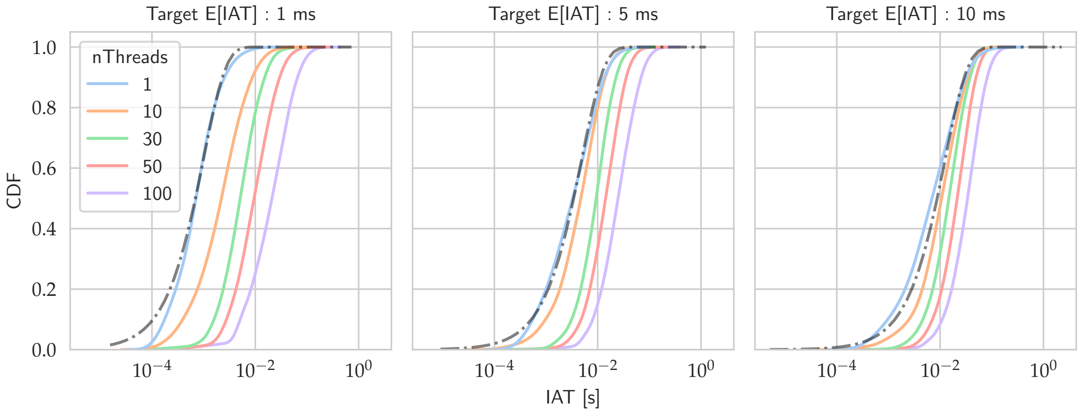


Figure 6.4: eCDF of the IAT realizations, under the assumption of exponentially distributed inter-arrivals (reference represented in black).

6.1.2 User Mobility

As already introduced in Section 5.1, the `Cluster Manager` can be programmed to modify the state of the `Flow Pool` according to the desired model. In this section we will describe the implementation of two models, `Three States Model` and $M/M/\infty$ Model, validate their behaviour, and introduce the context in which they can be used. Both models are based on a family of stochastic processes known as Markov Chain, widely used in different applications for their flexibility. Although it is assumed that this family of stochastic processes is known to the reader, an overview of the theoretical fundamentals can be found in [49].

Three Cell States Model

The `Three States Model` has been implemented using the `Finite State Space MC Cluster` profile of the `Cluster Manager`. The goal is to model the cell state that a tagged vehicle observes upon entering it, as depicted in Figure 6.8.

To validate this model, we will consider an uplink scenario where all the cells have the same channel capacity C of 10 Mbps. As introduced in Table 5.1 of Section 5.2, in order to describe the MC process, it is necessary to provide:

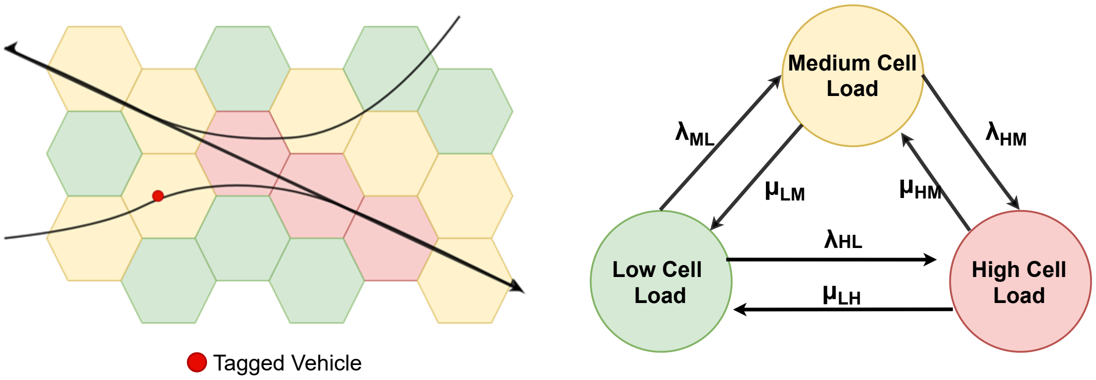


Figure 6.5: Visual representation of the Three State model. The `Tagged Vehicle`, moving through the road network, will experience different cell loads, determined by the MC dynamics.

- State Space S, in which the state variable represents the channel occupancy observed in each state. For this example the state space is defined as:

  $$S : \{60\%C, 80\%C, 100\%C\}$$

  where C is the considered channel capacity. Fixing the maximum number of parallel flows $nFlows_{max}$ producing a traffic volume equal to C, it is possible to map the state space to an equivalent representation as:

  $$S : \{60\%nFlows_{max}, 80\%nFlows_{max}, 100\%nFlows_{max}\}$$

- Transition Matrix P, which contains the probability of transition p_{ji} from state S_i to state S_j.

- Sojourn Rates V, where ν_i is the distribution parameter of the RV $T_{S_i} \sim Exp(\nu_i)$ describing the sojourn time in state S_i.

For this test, the transition matrix P and the sojourn rates V are defined as:

$$P = \begin{bmatrix} 0 & p_{lm} & p_{lh} \\ p_{ml} & 0 & p_{mh} \\ p_{hl} & p_{hm} & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{20} & \frac{19}{20} \\ \frac{1}{20} & 0 & \frac{19}{20} \\ \frac{2}{5} & \frac{3}{5} & 0 \end{bmatrix}, V = \begin{bmatrix} \nu_l \\ \nu_m \\ \nu_h \end{bmatrix} = \begin{bmatrix} \frac{1}{t_c} \\ \frac{1}{t_c} \\ \frac{1}{t_c} \end{bmatrix}$$

where t_c is the mean time required to traverse a cell, also referred to as expected sojourn time $\mathbf{E}[T_{S_i}]$. For this example we will consider cells of 1 km diameter and an average vehicle speed of $120\,\frac{\text{km}}{\text{h}}$, resulting in $t_c = \mathbf{E}[T_{S_i}] = 33.3\,\text{s}$. The summary of the test settings is provided in Table 6.2.

Given this settings, the stationary distribution of the process is $\Pi = [\pi_l, \pi_m, \pi_h] \simeq [\frac{2}{10}, \frac{3}{10}, \frac{5}{10}]$: the unbalance of the distribution toward the high cell load state has been designed to observe the behaviour of the traffic generator in a more stressful scenarios

Testbench Parameter	**Setting**
Channel Capacity	10 Mbps
$nFlows_{max}$	30 Flows
IAT Distribution	Exponential
Packet size and protocol	1250 B, UDP

Table 6.2: Summary of the evaluated mobility model testbench configuration.

and, more in general, to recreate network condition suitable for this study. To validate the behavior of the `Cluster Manager`, the results of five simulations of 600 s each have collected. Figure 6.6 shows the experimental stationary distribution of the process, reporting the probability $P(S = s_i)$ of observing the system in state S_i in stationary regime.

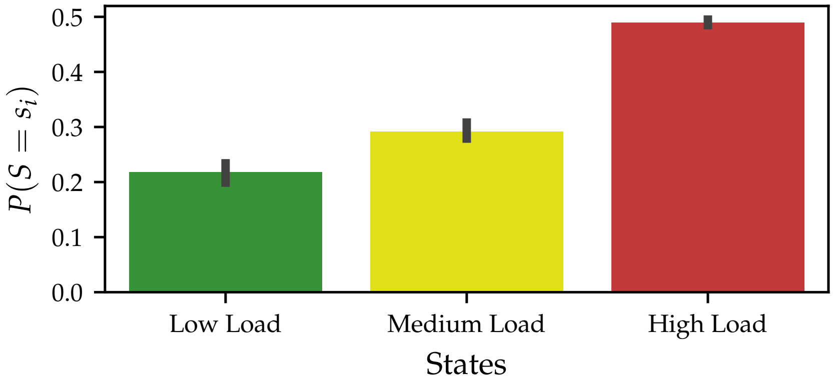


Figure 6.6: Experimental stationary distribution of the process

Figure 6.7 reports the results of one simulation, plotting both the state of the system, reported as the mean number of active flows, and the produced traffic volume versus time, showing how the state of the `Cluster Manager` has direct consequences on the background traffic volume produced.

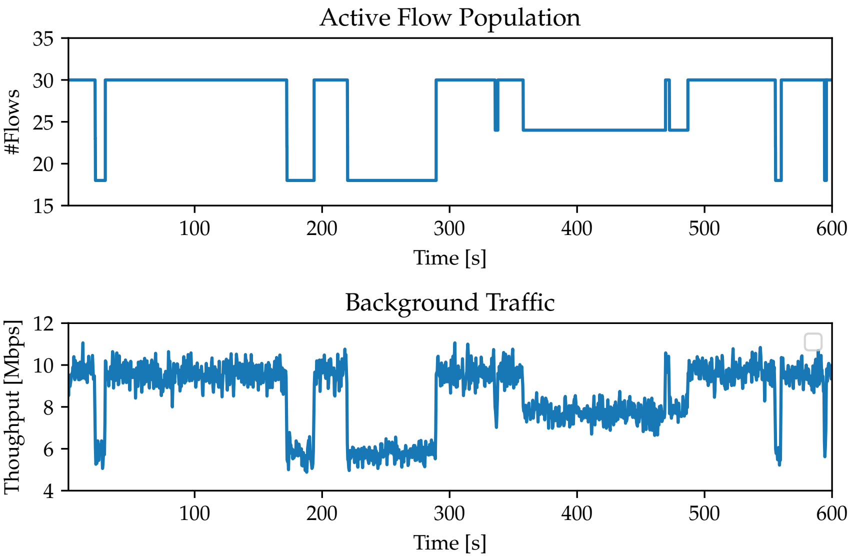


Figure 6.7: Three States Mobility model: on top, the `Cluster Manager` state; on the bottom, the volume the of produced background traffic.

Queue $M/M/\infty$ Model

The `Queue` $M/M/\infty$ `Model` has been implemented using the `Markovian Queue Cluster` profile of the `Cluster Manager`. The goal is to model the cell load variations that a tagged vehicle observes due to the arrival/departure of other vehicles, as depicted in Figure 6.8.

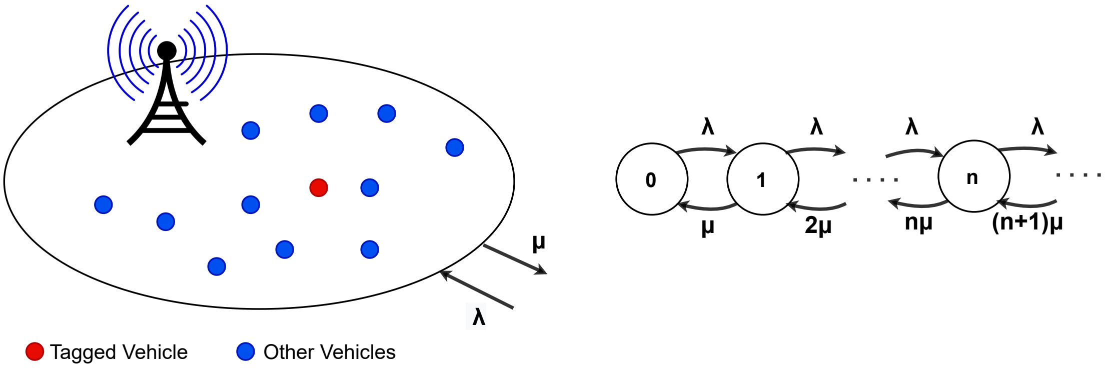


Figure 6.8: Visual representation of the Queue $M/M/\infty$ Model. The `Tagged Vehicle` inside the cell will experience variation in the cell state due to the departure/arrival of `Other Vehicles`, determined by the queue dynamics.

To validate this model, we will consider an uplink scenario with a single cell having channel capacity C of 10 Mbps. As introduced in Table 5.1 of Section 5.2, the underling model is a queue whose state variable is the number of active flows in the `Flow Pool`. To define the characteristics of the queue, it is necessary to declare:

- Arrival Rate $\lambda(s)$, passed as function handler to the `Cluster Manager`, which determines the arrival rate λ observed in state s_i. In our case, the function is:

$$\lambda(s_i) = \lambda \; \forall s_i \in S$$

- Departure Rate $\mu(s)$, also passed as function handler to the `Cluster Manager`, which determines the departure rate μ observed in state s_i. In our case, the function is:

$$\mu(s_i) = i \cdot \mu \; \forall s_i \in S$$

- Initial State s_0, which defines the number of `Other Vehicles` observed by the `Tagged Vehicle` at initialization.

The constructed model is well known in the literature [56] and, in stationary conditions, the probability distribution over the queue state space S follows a Poisson distribution $s \sim Poisson(\frac{\lambda}{\mu})$. Therefore the expected number of `Other Vehicles` observed in the long run is $\mathbf{E}[s] = \frac{\lambda}{\mu}$. The testing of this mobility model uses the same testbench configuration used for the `ThreeState` model, described in Table 6.2, while the characteristic parameters of the model λ and μ have been set to $10\,\mathrm{s}^{-1}$ and $\frac{1}{3}s^{-1}$. Figure 6.9 reports the results of one simulation, plotting both the state of the system,reported as the mean number of active flows, and the produced traffic volume versus time.

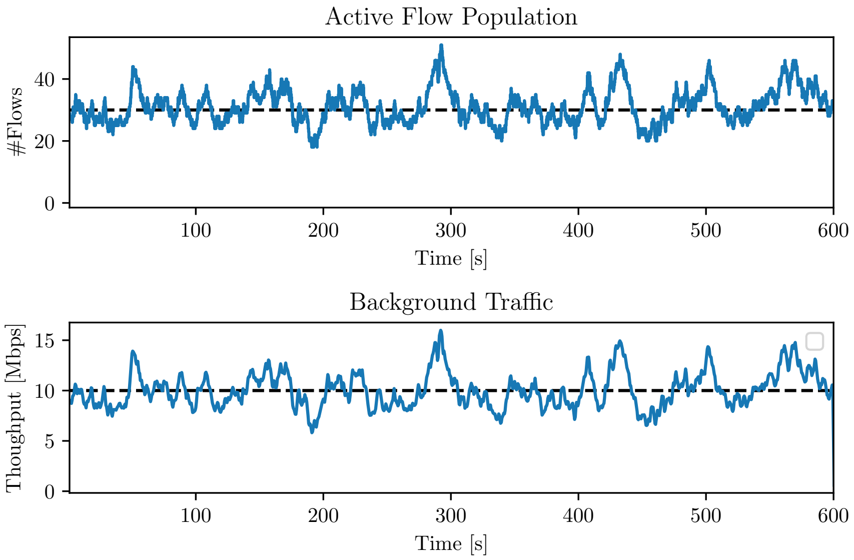


Figure 6.9: $M/M/\infty$ Mobility model: on the top, the `Cluster Manager` state; on the bottom, the volume the of produced background traffic.

6.2 Testbed Validation

The scenario design for the testbed validation focuses on a limited number of cases, analyzing in detail the network behavior under a wide range of working conditions. A typical uplink traffic situation with competing UEs is evaluated in scenarios SC1. In this scenario, the traffic is generated from the traffic generators placed in the namespaces `wwan-NSs` and to the EPC via gNodeB, where it will be routed to the `server-NS` namespace, as depicted in Figure 6.10. Under the same assumptions, downlink traffic in the competing UEs case is analyzed in scenarios SC2. In this case, the traffic generators are placed on the `server-NS` namespace and the traffic is transmitted in downlink to the receiver hosted on the `wwan-NSs` namespaces via the associated radio interface.

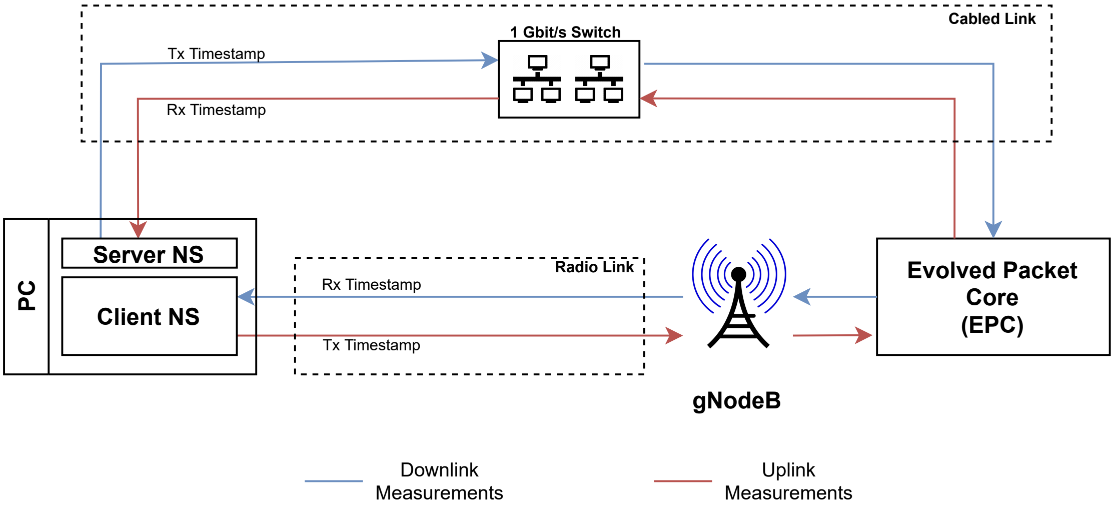


Figure 6.10: Diagram of the measurement loop.

All scenarios will be conducted testing using UDP and MQTT/TCP traffic, allowing the analysis of the side-effects of TCP (congestion control, flow control, re-transmissions, etc.). Each scenario will be run for 180 s, which allow the system to reach its stationary state, for a total of five times, to derive KPIs confidence intervals. Since the goal is to characterize the steady state conditions of the system, the first 30 s of each run have been discarded in the KPIs calculation to remove samples observed in transient state. Also, very low radio channel attenuation and no interference of other cells are assumed.

We also consider overload scenarios. They do not result in a total breakdown of the system as buffer sizes are limited and packets get eventually dropped. Furthermore, in case of MQTT with TCP, packet drops cause the congestion window to be reduced and sending rate to be reduced. But the traffic generation rate is not reduced, so a backlog of unsent messages builds up. Application layer solutions, not considered in this thesis, could detect such overload situations and adapt the sending rate of messages. In ETSI ITS this is known as “Distributed Congestion Control” [9] but typically not applied for long-range cellular transmissions.

Preliminary Measurements and Testbed Parameters

Deriving the setup parameters requires knowledge of the effective channel conditions, therefore a preliminary measurement campaign has been run using the iperf3 tool for both TCP and UDP traffic. The measured uplink channel capacity C_{SC1} is 16 Mbps while the downlink channel capacity C_{SC2} is 250 Mbps, respectively 20% and 5% lower respect to the theoretical values derived in Section 5.2.2.

Given this preliminary information, the scenarios will be realized employing $N_{RM} = 2$ radio-modems, attached to different PCs. For scenario SC1, we will test the system response varying the channel utilization U between 10% and 150% of C_{SC1} and, to further characterize the system, also the packet size L_{pkt} will vary between 300, 750 and 1250 Bytes. For SC2, instead, we will limit the measurement range between 10% and 130% of C_{SC2}, considering a packet size of 1250 Bytes: the saturation of the channel with packets of smaller size would require to increase the number of PCs and radio-modems employed, due to the trade-off documented in Section 6.1. Using the same empirical criteria, we decided to use as the number of flows $F_{SC1} = 30$ and $F_{SC2} = 120$ for scenario SC1 and SC2, respectively, where the flows are split equally between the N_{RM} radio-modems. For a fixed set of U,L_{pkt}, and F_{Si}, the throughput of each flow, and thus its $\mathbf{E}[IAT]$, can be calculated using equations 6.1.

$$T_{flow} = \frac{C_{Si}}{F_{Si}}, \qquad \mathbf{E}[\text{IAT}_{flow}] = \frac{L_{pkt}}{T_{flow}} \tag{6.1}$$

ID	Description	Testbed Parameters		
		Channel Utilization	Packet Size [Bytes]	IAT distribution
SC1	Uplink, 30 Flows	10% - 150%	300, 750, 1250	Exponential
SC2	Downlink, 120 Flows	10% - 130%	1250	Exponential

Table 6.3: Summary of the evaluated scenarios.

KPIs Overview

Channel Utilization The channel utilization is the ratio, expressed as a percentage, between the data volume produced by the testbed, or throughput, and the channel capacity. A sliding window of 500 ms with an overlap of 25% is used to smooth statistical fluctuations and obtain an average throughput.

End-to-End Delay The end-to-end delay is the time interval between packet transmission by the client and packet reception by the server. To characterize the KPI, both the delay complementary cumulative distribution function (CCDF) and the mean delay versus channel utilization will be reported.

Packet Error Rate (PER) The PER is the ratio, expressed as a percentage, of the number of missed packets at a receiver from a particular traffic flow and the total number of packets sent by that flow. PER is calculated using the sequence number contained in each message between each flow and the corresponding logging server. The average PER is calculated and plotted versus the channel utilization.

6.2.1 Scenario SC1: Competing UEs to EPC

Figure 6.11 shows an example of one of the measurement trials, plotting the effective channel occupation and the delay experienced by the packet of the tagged flows for packet size L_{pkt} of 1250 Bytes and target channel occupation of 60%, 90%, 100%, and 110%. It is noticeable that the generated data volume matches the target channel occupation. Furthermore, focusing on the delay samples over time, we can notice how the first tenths of seconds are characterized by a transient behavior, due to the device queues building-up.

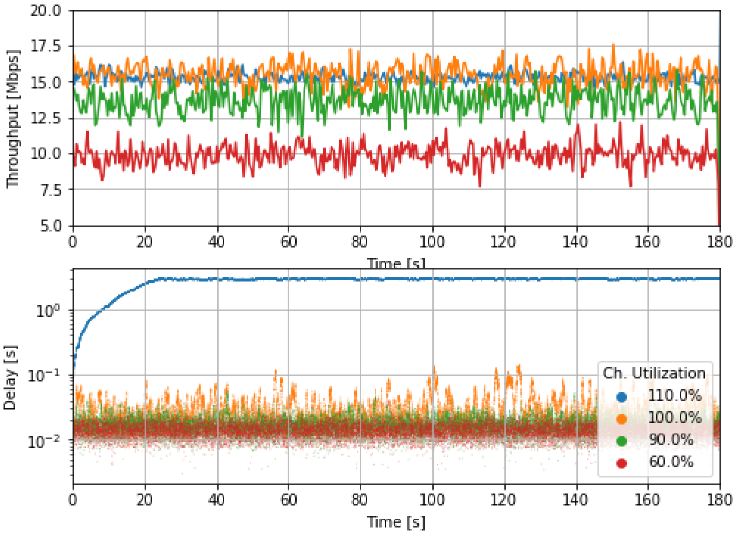


Figure 6.11: Uplink data volume produced and delay samples over time for runs with different target utilization (UDP, 1250 B).

End-to-End Delay

Mean Delay versus Channel Occupancy Figures 6.12 and 6.13 report the mean delay experienced by the tagged flows when using UDP and MQTT/TCP respectively.

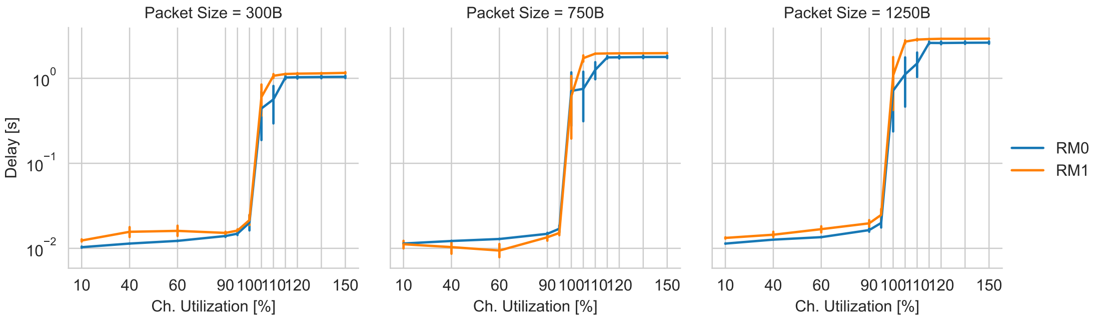


Figure 6.12: Mean delay with 95% confidence intervals versus channel utilization (UDP).

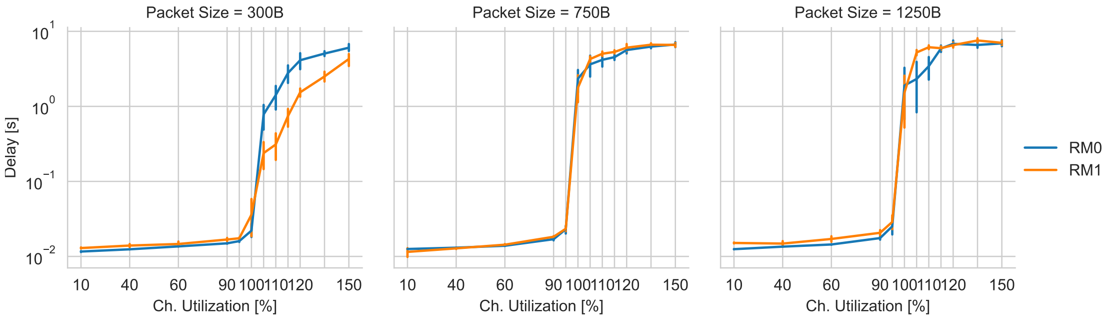


Figure 6.13: Mean delay with 95% confidence intervals versus channel utilization (MQTT).

It is noticeable how, in both protocols and for all the considered packet sizes, the system presents similar behavior: for channel utilization between 10% to 90%, the average delay oscillates between 10 ms to 20 ms, with slight increases in the upper side of the utilization range. For all packet sizes, a sudden degradation of the performances around the 95%-100% channel occupancy brings the average delay from tens of milliseconds up to the order of seconds. In the case of packet size 300 Bytes, this phenomenon appears to be slightly more lenient, suggesting that larger packets introduce complexity in resource allocation and thus lower the system efficiency. Lastly, for utilization levels above 100%, we can notice different behaviors between UDP and MQTT/TCP: once the UDP socket queue is filled, the protocol starts dropping packets, limiting the maximum delay that a packet can experience. On the other hand, MQTT/TCP, providing reliable transmission, continues to queue packets and thus resulting in higher mean delays. Once the TCP buffer is full (maximum size defined by the OS), MQTT will fail in the packet transmission (`MQTT_ERR_QUEUE_SIZE` [19]) due to QoS 0 setting in the MQTT client.

Delay Distribution Figures 6.14 and 6.15 report the experimental complementary cumulative distribution function (CCDF) of the delay experienced by the tagged flows.

The same observations reported in the mean delay analysis are remarked by the distribution. The distributions in the UDP and MQTT/IP cases show similar performances for channel utilization lower than 90%. The major difference arises for overload levels of channel utilization, where we can see how the two protocols show different behaviors in the management of the socket buffer under prolonged stress conditions.

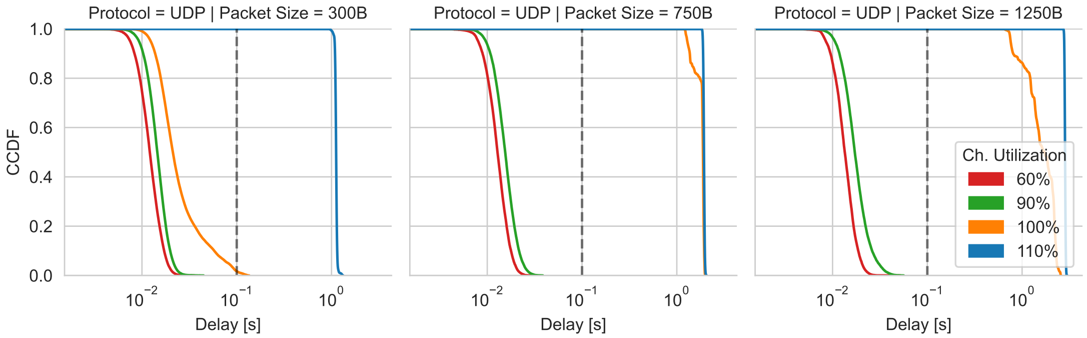


Figure 6.14: Delay Complementary CDF (UDP)

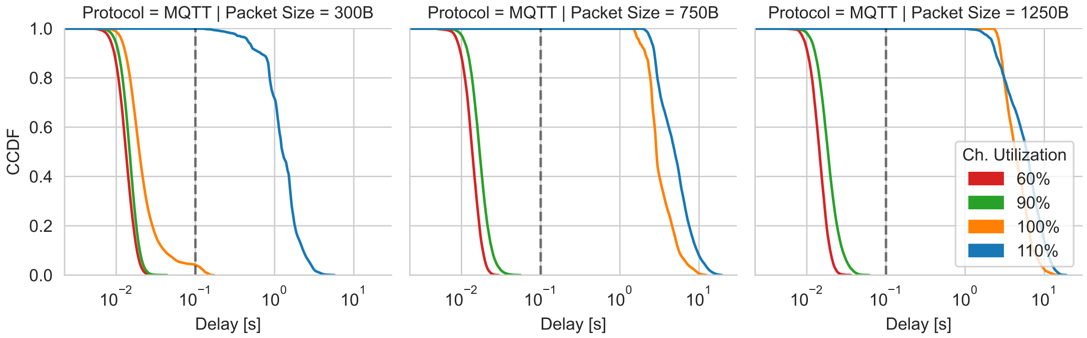
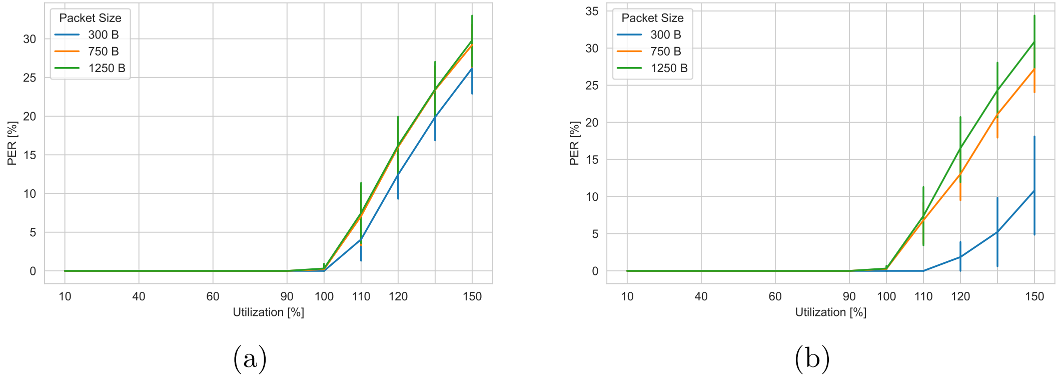


Figure 6.15: Delay Complementary CDF (MQTT)

Packet Error Rate

Figure 6.16 reports the packet error rate (PER) statistics for the UDP and MQTT protocols. Although MQTT should provide reliable transmission over TCP, the PER refers to the application level transmission error due to the status of the TCP buffer: The MQTT client will fail in the message publication (`MQTT_ERR_QUEUE_SIZE` []), and due to the MQTT QoS set to QoS 0, it will not reattempt to transmit the message.



(a)

(b)

Figure 6.16: Uplink packet error rate (PER) statistics versus channel utilization. (a) PER for UDP datagrams. (b) PER for MQTT messages.

6.2.2 Scenario SC2: EPC to Competing UEs

Figure 6.17 shows one of the downlink measurement traces for traffic, plotting the effective channel utilization and the delay experienced by the tagged flows for packet size L_{pkt} of 1250 Bytes and target channel occupation of 50%, 70%, 90%, and 100%. As in Scenario SC1, we can notice that the average generated data volume matches the target channel occupation.

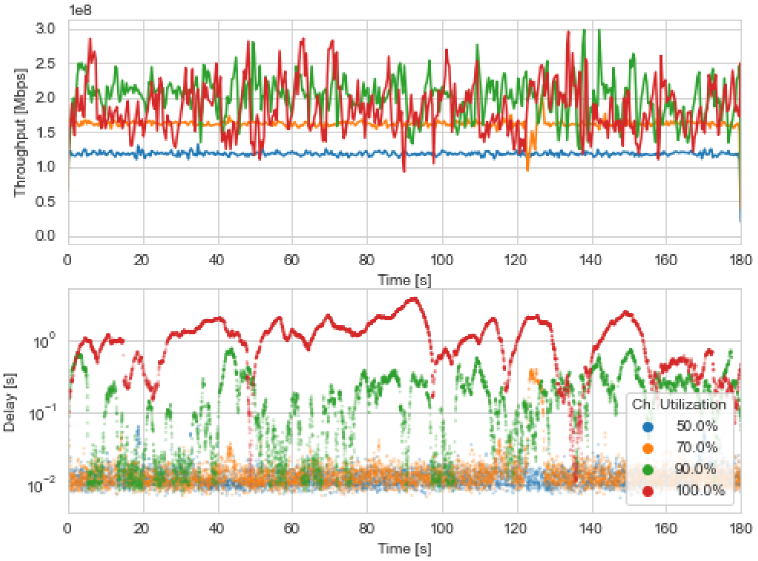
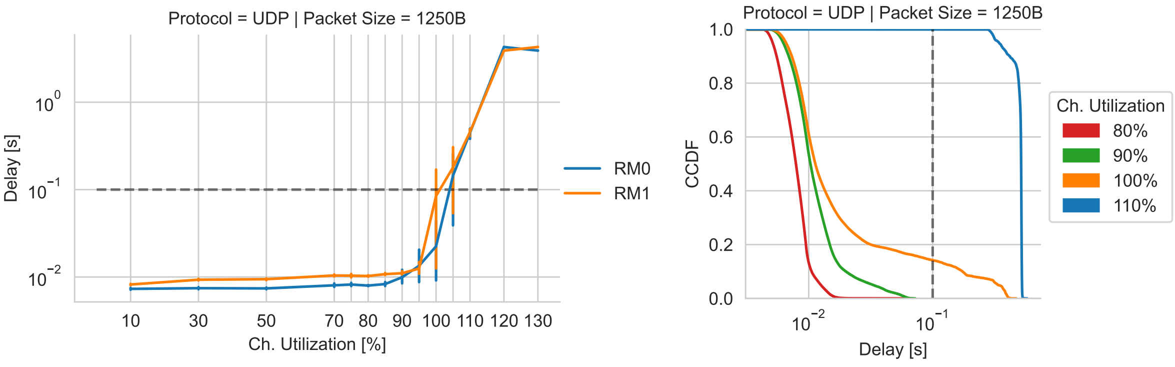


Figure 6.17: Downlink data volume produced and delay samples over time for runs with different target utilization (MQTT, 1250 B).

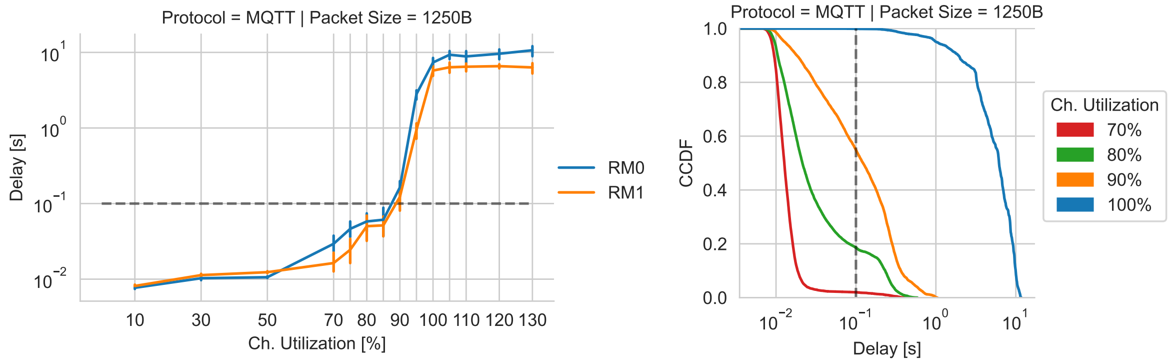
Delay Statistics

Figures 6.18 and 6.19 report the downlink delay statistics experienced by the tagged client for UDP and MQTT/TCP protocol, respectively. Comparing the performances of the two protocols for the same channel utilization, it is noticeable that the performances of MQTT in terms of average delay of the received packets are worse than the UDP counterpart in congested scenarios.



(a) (b)

Figure 6.18: UDP delay statistics. (a) reports the mean delay with relative 95% confidence intervals versus channel utilization. (b) shows the delay CCDF, conditioned on the channel utilization.



(a) (b)

Figure 6.19: MQTT delay statistics. (a) reports the mean delay with relative 95% confidence intervals versus channel utilization. (b) shows the delay CCDF, conditioned on the channel utilization.

The dashed lines in the figure represent the considered delay budget of 100 ms. In the UDP case, the delay constraint is consistently respected only for utilization < 100%: we can observe that for full channel utilization, 20% of the successfully received packets have experienced a delay larger than 100 ms. Instead, in the MQTT case, already at 70% utilization we can observe that 2% of messages does not respect the delay constraint, reaching 20% and 55% in case of 80% and 90% channel utilization, respectively.

Packet Error Rate

Figure 6.20 reports the downlink packet error rate (PER) for the UDP and MQTT protocols. The same observations regarding the behaviour of the MQTT client reported in Section 6.2.1 are valid also in this case.

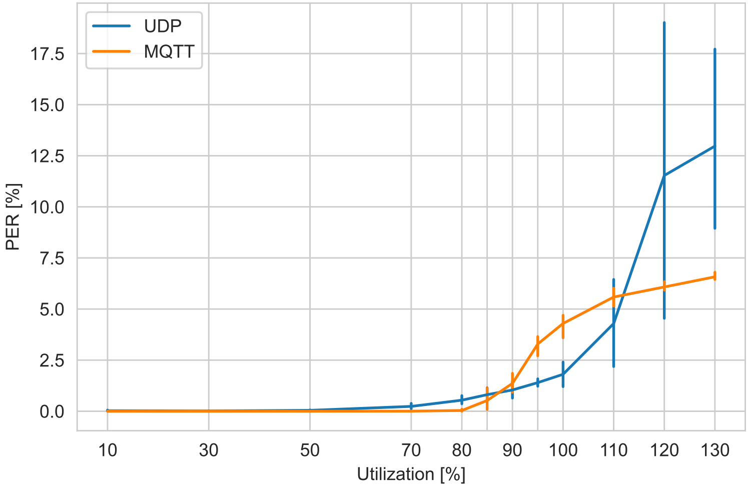


Figure 6.20: Downlink packet error rate (PER) statistics versus channel utilization.

7 Digital Twin Use-Case: Vehicle Position Tracking and AEB Systems

7.1 Scenario Design

System Architecture Figures 7.1 depicts the recreated system architecture. It comprises of a MEC server collocated within the base station [3]: the deployment of the digital twin application close to the gNodeB allows to consider the radio channel as the main source of the end-to-end delay experienced by the packets. The mock vehicles will transmit their status using CAM messages: the information regarding their position is pre-computed and stored in GPS traces, to be read according to the simulation time and configured speed of the vehicle in order to simulate the vehicles trajectory.

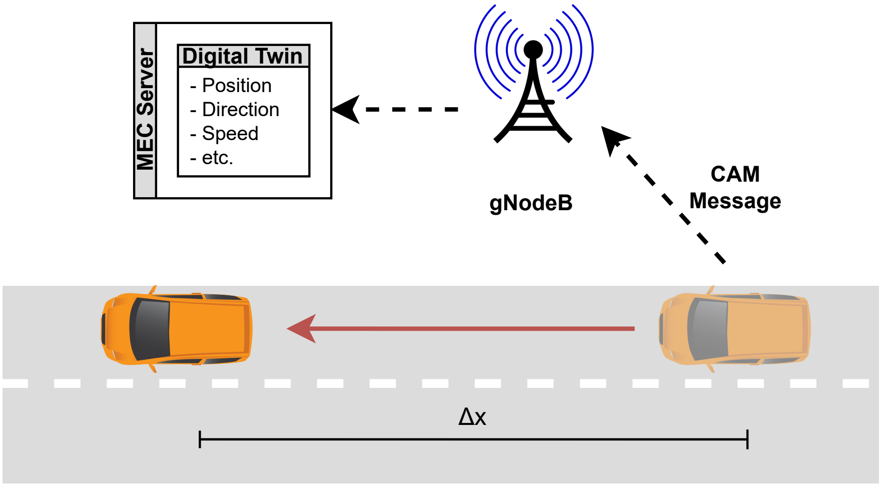


Figure 7.1: Communication scheme between vehicle and digital twin, highlighting the introduced error ΔX due to the CAM message reception delay.

Scenarios Assumptions The following simplifying assumptions are considered in the scenario design:

- The considered CAM messages generation frequency is 10 Hz with message size of 350B, motivated under the information reported in Section 4.2.3.
- The GPS positioning Error is assumed negligible. This assumption is backed up by the presence of centimeter-level accuracy GNSS positioning techniques, as for

example GNSS Real-Time Kinematics [46]. We will also assume that the sampling frequency of the GPS sensor matches the generation frequency of the CAM messages. Although some commercial units may not achieve measurements time of 100 ms, position data can be approximated using input from inertia sensors.

- The background traffic process is characterized by exponential inter-arrival times (IAT), whose mean value $\mathbf{E}[IAT]$ is set as described in section 6.2, while the packet size L_{pkt} is 350 Bytes.

Vehicle Position Tracking Scenario For various static utilization u, track the vehicle position X using CAM messages from the car to the digital twin and measure the digital twin tracking error ΔX. The trials will be run both in static network condition, fixing the channel utilization, and dynamic network conditions, using the mobility `Three State Model` previously introduced. We will consider a highway mobility scenario, thus considering an average vehicles speed $\overline{v}$ of 120 kmph. UDP and MQTT/TCP will be used as transport/application protocols.

Collision Avoidance Scenario For various static utilization u, measure the safety performances of a digital twin based automatic emergency braking (AEB) system to identify that the vehicle requires an emergency brake due to other entities on the road (traffic jam, stationary vehicles, etc.). The trials will be run in static network condition, measuring the performances for various channel utilization u. UDP will be used as transport/application protocols.

7.1.1 Positioning and Kinematic KPIs

Digital Twin Tracking Error Δx represents the position error, expressed in meters, between the position of the real vehicle P_R and the last known position by the digital twin P_{DT}.

$$\Delta x = |P_R - P_{DT}| = T_{CAM_A} \cdot \overline{v} \tag{7.1}$$

where $\overline{v}$ is the average velocity of the vehicle and T_{CAM_A} is the random variable time interval between two successful reception of CAM messages between real vehicle and digital twin. We can decompose T_{CAM_S} in order to highlight its dependence to the channel utilization U:

$$T_{CAM_A}|U = T_{CAM_T}|U + D|U$$
$$T_{CAM_T}|U = \frac{1}{f_{CAM}} \cdot N_S|U \qquad N_S|U \sim Geom(1 - PER(U))$$

where $D|U$ is the packet delay and $T_{CAM_T}|U$ is the time interval between two successful transmissions from the real vehicle to the digital twin, given channel utilization U. The probability of erroneous packet transmission can be empirically estimated considering the average packet error rate (PER) as a function of the channel utilization. Assuming the PER fixed for a given channel utilization U, the number of attempts between two successful transmission is geometrically distributed with parameter $p = PER(U)$.

Time to Collision Time to collision (TTC) is the most well known safety indicator, applied in different studies for evaluating rear-end collision risk. TTC value at an instant t is defined as the time for two vehicles to collide if they continue at their present speed and on the same path. So, the higher a TTC value, the safer a situation. As reported in [48], the TTC for a following vehicle f at instant t with respect to a leading vehicle l is:

$$TTC_f(t) = \frac{X_l(t) - X_f(t)}{v_f(t) - v_l(f)}$$

where X_i and v_i denote respectively the position and the velocity of the vehicle.

7.2 Results

7.2.1 Vehicle Position Tracking Scenario

Stationary Network Conditions In this trial, the vehicle will observe the same radio network conditions throughout the entire virtual drive, allowing to measure the position tracking KPI of the digital twin in stationary conditions. The testbed parameters are reported in Table 7.1. Figure 7.2 reports the mean position error ΔX versus the radio channel utilization U, also characterizing the error components relative to retransmission $T_{CAM_t}|U$ and delay $D|U$.

Category	Testbench Parameter	Setting
General	Channel C	Uplink, 16 Mbps
	Protocols	UDP, MQTT
Background Traffic Flows	$nFlows_{max}$	30 Flows
	IAT Distribution	Exponential
	Packet size	350 B
CAM Traffic Flow	Vehicle Speed $\overline{v}$	120 kph
	$nFlows$	1 Flow
	IAT Distribution	Periodic, 100 ms
	Packet size	350 B

Table 7.1: Vehicle Position Tracking - Summary of the testbench configuration with static network conditions.

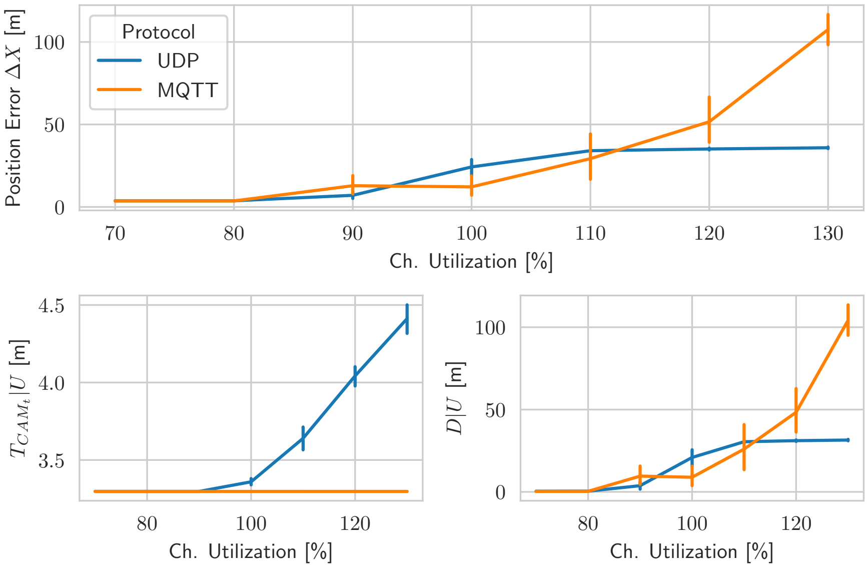


Figure 7.2: Mean position error ΔX versus channel utilization u. The error components due to packet generation time and re-transmission T_{CAM_t} and due to packet delay $D|U$ are also reported.

For channel occupancy lower than 90%, both UDP and MQTT protocols show similar behavior, with a tracking error mainly constitute by the T_{CAM_t} component: due to the CAM message frequency of 10 Hz and considering the vehicle velocity $\overline{v}$ of 120 kph, by the time the vehicle will generate a new message, it will have already moved 3 m from the last advertised position. For channel occupancy between 90% and 100%, the contribution from $T_{CAM_t}|U$ and $D|U$ becomes comparable, bringing the overall error to around 10 m. In case of channel overload instead, with channel utilization between 100% and 130%, the system shows a rapid degradation of performances due to the queue build-up. In the UDP case the unreliable transmission policy allows to drop packets, thus limiting the staleness of the transmitted position information and providing a bound on the position error, in this case of around 30 m. Instead, in the MQTT case, the underling TCP reliable transmission creates larger delays, translating in position errors of 100 m: only the QoS mechanism of MQTT (QoS 0) provides a bound to the experienced message delay.

Dynamic Network Conditions In this trial, the vehicle will observe the dynamic radio network conditions throughout virtual drive using the mobility model introduced in Section 6.1.2, allowing to measure the position tracking KPI of the digital twin in more realistic conditions. Table 7.2 reports the scenario settings.

Figure 7.3 shows the measured tracking error ΔX experienced by the vehicle: upon entering in a heavily loaded cell, denoted by state $S_H = 100\%$, the transmission delay causes the tracking error reaches its maximum of 30 m. For states $S_L = 70\%$ and $S_M = 80\%$ instead the tracking error shows the same behaviour observed in the static case, where the main source of error is due to the inter-generation time of the CAM messages.

Category	**Testbench Parameter**	**Setting**
General	Channel C	Uplink, 16 Mbps
	Protocols	UDP
Background Traffic Flows	$nFlows_{max}$	30 Flows
	IAT Distribution	Exponential
	Mobility Model	Three State (Sec. 6.1.2)
	Packet size	350 B
CAM Traffic Flow	Vehicle Speed $\overline{v}$	120 kph
	IAT Distribution	Periodic, 100 ms
	Packet size	350 B

Table 7.2: Vehicle Position Tracking - Summary of the testbench configuration with dynamic network conditions.

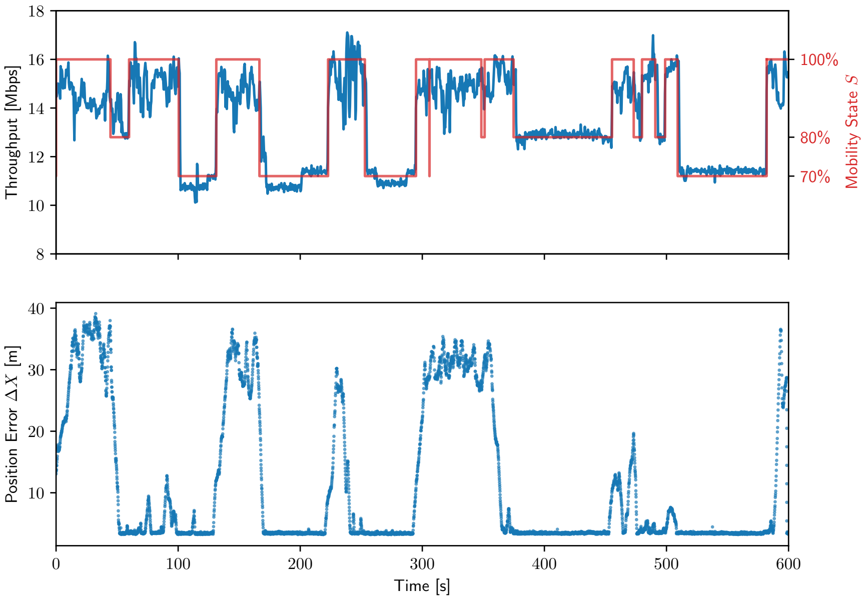


Figure 7.3: Vehicle Position Tracking - On the top, generated background traffic and process state over time. On the bottom, position error ΔX over time.

7.2.2 Automatic Emergency Braking Scenario

In this scenario we will investigate the performances of an Autonomous Emergency Braking (AEB) based on a digital twin architecture. Figure 7.4 depicts the recreated testing scenario and the overall AEB system architecture.

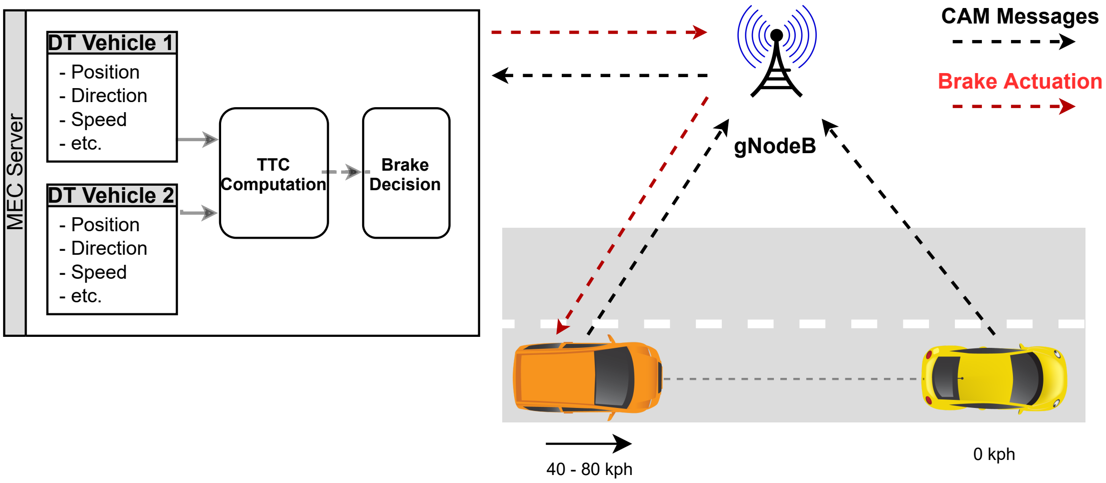


Figure 7.4: Communication scheme between vehicle and digital twin, highlighting the introduced error ΔX due to the CAM message reception delay.

The scenario respects the Euro-NCAP testing guidelines for the testing of AEB [14], considering the case of Car-to-Car Rear Stationary (CCRs) collision, where the vehicle-under-test (VuT) travels forwards towards another stationary vehicle (V_s) and the frontal structure of the vehicle strikes the rear structure of the other. Considering an inter-urban setting, the testing protocol requires to test the AEB performance varying the initial speed of the VuT in the range between 30 kph and 80 kph : for this test, the considered VuT initial velocity are 40 kph, 60 kph, and 80 kph.

The implemented AEB system uses as safety indicator the Time-to-Collision to determine the vehicle potential exposure to hazardous conditions, following the examples described in [61] and [37]. The system architecture works as follows: VuT and V_s will transmit their position and speed to the respective digital twin using CAM messages, updating the system view of the scenario. The implemented AEB function, following the protocol depicted in Figure 7.5, will access the position and velocity data of the vehicles and it will compute the instantaneous TTC: comparing to the configured TTC safety thresholds, the AEB function will determine the hazardous state and select the associated braking profile. The implemented braking profiles are derived from the NCAP testing protocol.

As described in Table 7.3, the trial will be conducted varying the uplink conditions, considering utilization levels u between 70% and 110%. We expect to observe a progressive performance deterioration of the AEB system due to the delay experienced by the VuT CAM messages: the introduced tracking error will delay the sensing of an hazardous situation, thus delaying the brake actuation. Depending on the magnitude of the tracking error, the VuT may not be able to perform a safe braking procedure and collide with the stationary vehicle.

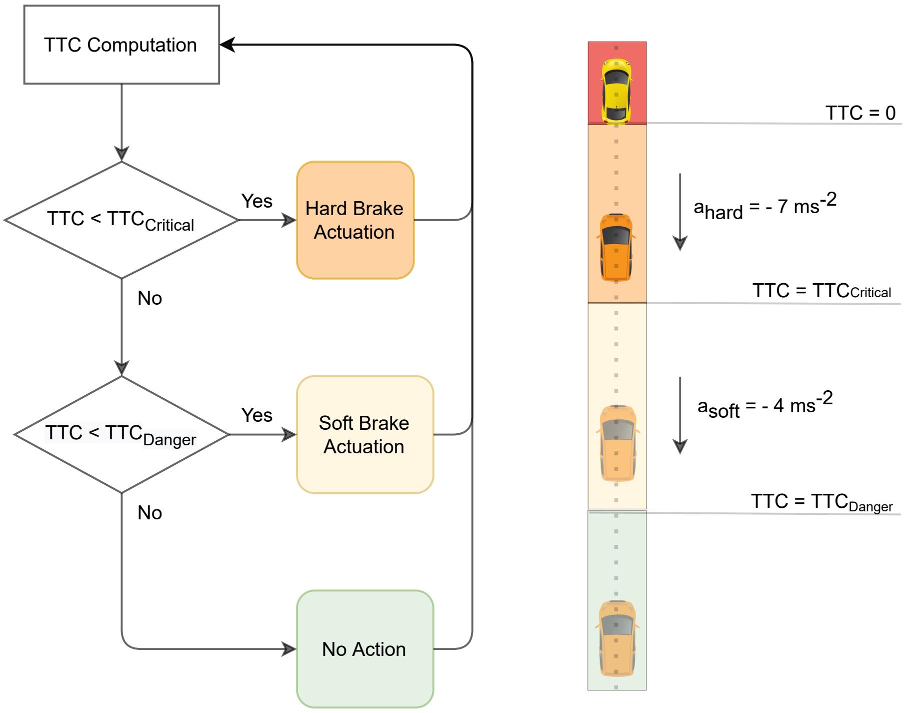


Figure 7.5: AEB protocol scheme and vehicle braking phases

Category	Testbench Parameter	Setting
General	Channel C	Uplink, 16 Mbps
	Protocols	UDP
	Utilization	[70% - 110 %]
Background Traffic Flows	$nFlows_{max}$	30 Flows
	IAT Distribution	Exponential
	Packet size	350 B
CAM Traffic Flow	Vehicle Speed $\overline{v}$	40 kph, 60 kph, 80 kph
	$nFlows$	1 Flow
	IAT Distribution	Periodic, 100 ms
	Packet size	350 B

Table 7.3: Collision Avoidance - Summary of the testbench configuration.

Braking Curves

Figures 7.6, 7.7, 7.8 report the trials results with VuT initial velocity of 40 kph, 60 kph, and 80 kph, respectively. In all three cases, we can notice how the network status heavily influence the ability of the AEB system to perform a safe stop procedure:

- For VuT speed equal to 40 kph, the collision was avoided in all network conditions. Although it is noticeable how the delay affected the digital twin sensing and the vehicle braking actuation, the soft braking profile was able to guarantee a safe stop of the vehicle in all the considered network conditions.

- For VuT speed equal to 60 kph, only in heavily congested scenario (110% utilization) the system was not able to prevent the collision, being able only to contain the damage by reducing the VuT speed to 10 kph.

- For VuT speed equal to 80 kph, the collision could not be avoided in cases of full channel (100%) utilization and congested scenario (110%): the AEB was only able to reduce the vehicle speed to 25 kph and 40 kph, respectively, causing in both cases heavy damages to the vehicle and potentially fatal injuries to the passengers.

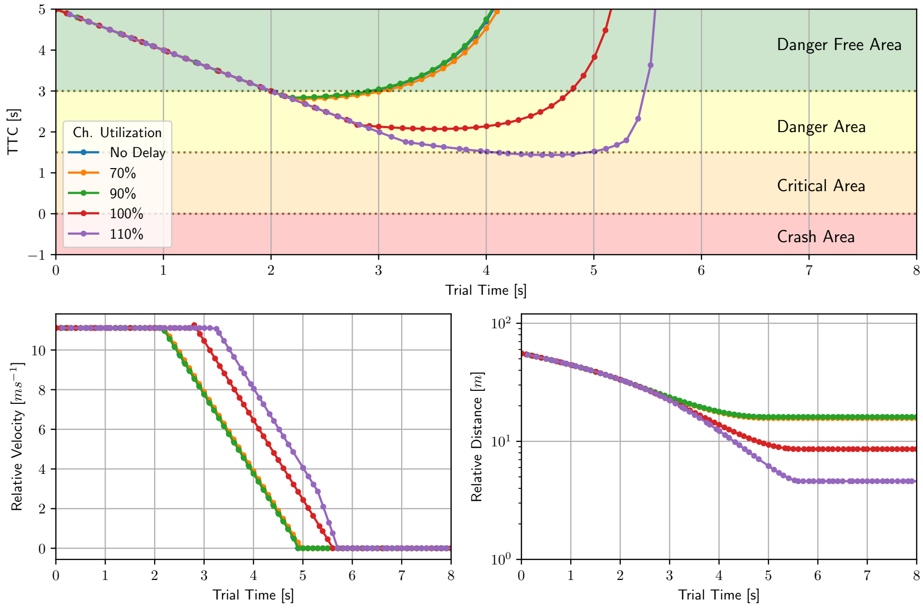


Figure 7.6: AEB CCRs Scenario Results - VuT Speed $\overline{v} = 40$ kph.

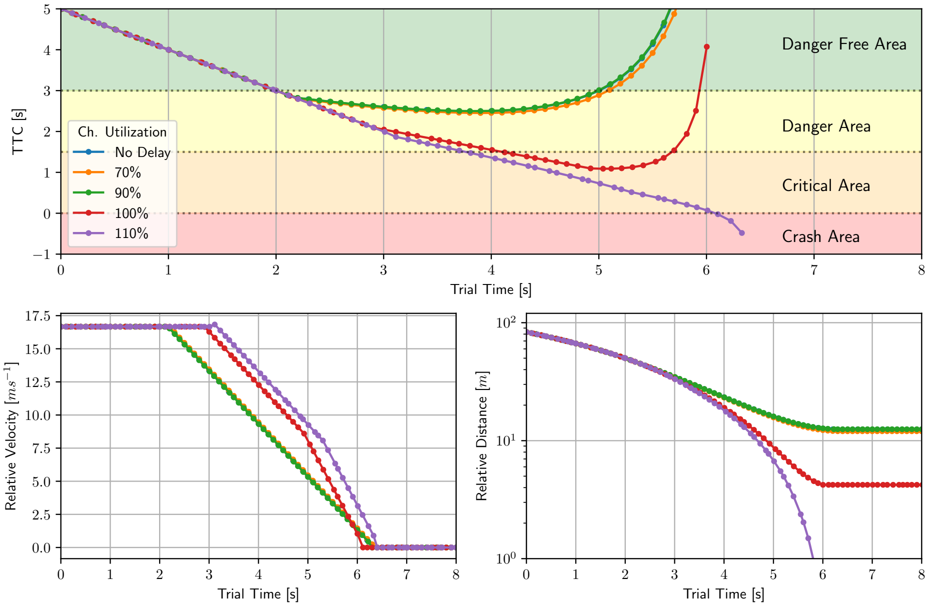


Figure 7.7: AEB CCRs Scenario Results - VuT Speed $\overline{v} = 60\,\text{kph}$.

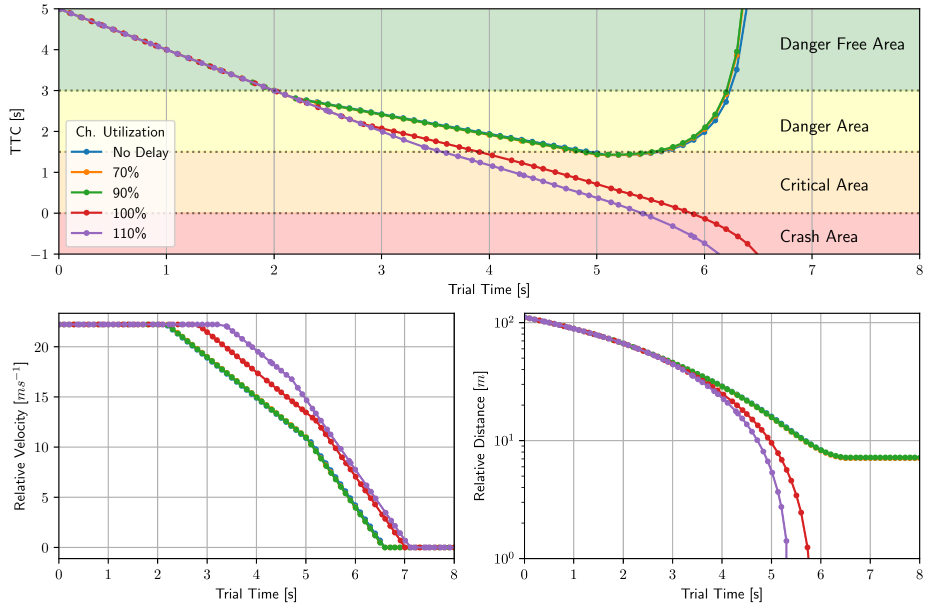


Figure 7.8: AEB CCRs Scenario Results - VuT Speed $\overline{v} = 80\,\text{kph}$.

8 Conclusions

This thesis concerned itself with the question of how to build scalable testbed for vehicular communication, analyzing how the employed hardware resources can constraint the realism of the recreated scenario. For such purpose, a testbed architecture based on the Hardware-in-the-Loop approach has been developed and validated.

The conducted experiments set out to identify the software and hardware constraints that limit the recreation of realistic background traffic patterns and which design parameters should be considered in the design of testbeds for connected vehicles. The results obtained in the validation of the traffic generators highlight how the generation of high traffic volumes requires strict timing constraints. The general purpose computing units employed in the testbed do not provide time guaranties, thus resulting in distortions in the inter-arrival distribution of the generated packets with respect to the target one and delivery of lower data volumes. Design curves have been created to quantify the introduced distortions and provide empirical rules for the testbed setup.

Furthermore, in the validation of the testbed setup, measurements on the QoS of vehicular traffic flows where conducted in both uplink and downlink, varying the background traffic characteristics (protocol, channel utilization, packet size).

As second objective of this thesis, we focused on the study of how the performances of Digital Road Twins is impacted by communication network conditions, analysing the effects on the kinematic KPIs of simulated vehicles. The results provided by the testing of the vehicle position tracking functionality show how the QoS (delay, packet loss) experienced by the vehicle traffic flow caused a tracking error of 50 m in saturated channel conditions. The tests on the Automatic Emergency Braking (AEB) functionality highlight further challenges in the large scale deployment of digital twin based safety systems: the delay introduced by the uplink channel due to channel utilization translate into delays in the recognition of road hazards and actuation of the required safety measures. Although extreme case were observed only in highly congested network scenarios, these test show the importance of mobile network operators for the provision and rollout of adequate network resources for IoV applications.

Bibliography

[1] Amazon FleetWise . https://aws.amazon.com/iot-fleetwise/.

[2] Amazon TwinMaker . https://aws.amazon.com/iot-twinmaker/.

[3] ETSI - MEC deployment in 5g Networks. https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf.

[4] Microsoft Azure Digital Twin. https://azure.microsoft.com/en-us/services/digital-twins/#features.

[5] Quectel RM500Q-GL Product Sheet. https://www.quectel.com/wp-content/uploads/2021/03/Quectel_RM500Q-GL_5G_Specification_V1.4.pdf.

[6] Report for 5GAA on C-V2X Socio-economic benefits. https://5gaa.org/wp-content/uploads/2017/12/Final-report-for-5GAA-on-cellular-V2X-socio-economic-benefits-051217_FINAL.pdf.

[7] Supply Chain Twin and Pulse . https://cloud.google.com/solutions/supply-chain-twin.

[8] 5GAA P190033: V2X Functional and Performance Test Report. https://5gaa.org/wp-content/uploads/2018/11/5GAA_P-190033_V2X-Functional-and-Performance-Test-Report_final-1.pdf.

[9] Decentralized Congestion Control Mechanisms for Intelligent Transport Systems. https://www.etsi.org/deliver/etsi_ts/102600_102699/102687/01.01.01_60/ts_102687v010101p.pdf.

[10] Decision no 585/2014/eu of the european parliament and of the council of 15 may 2014 on the deployment of the interoperable eu-wide ecall service. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014D0585from=EN.

[11] Eclipse MOSAIC Framework. https://www.eclipse.org/mosaic/.

[12] ETSI Cooperative Intelligent Transportation Systems. https://www.etsi.org/technologies/automotive-intelligent-transport#:~:text=Intelligent%20Transportation%20Systems%20(ITS)%20aim,smarter'%20use%20of%20transport%20networks.

[13] European Commission - On the road to automated mobility: An EU strategy for mobility of the future. "https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52018DC0283".

[14] European New Car Assessment Program (EURO-NCAP) - TEST PROTOCOL AEB systems. https://cdn.euroncap.com/media/26996/euro-ncap-aeb-c2c-test-protocol-v20.pdf.

[15] HEADSTART - D2.1 Common methodology for test, validation and certification. https://www.headstart-project.eu/results-to-date/deliverables/.

[16] HEADSTART Website. https://www.headstart-project.eu/.

[17] Horizon 2020. https://ec.europa.eu/programmes/horizon2020/en/what-horizon-2020.

[18] Linux Namespaces . https://man7.org/linux/man-pages/man7/namespaces.7.html.

[19] MQTT Paho Publishing. https://www.eclipse.org/paho/index.php?page=clients/python/docs/index.php#publishing.

[20] MQTT Webpage. https://mqtt.org/.

[21] Project PEGASUS - V Model and Process Analysis. https://www.pegasusprojekt.de/files/tmpl/PDF-Symposium/02_V-Model-and-Process-Analysis.pdf.

[22] Regulation (EU) 2018/858 of the European Parliament and of the Council of 30 May 2018 on the approval and market surveillance of motor vehicles and their trailers, and of systems, components and separate technical units intended for such vehicles. https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A32018R0858.

[23] Vehicular Communications Basic Set of Applications, Local Dynamic Map (LDM) - Rationale for and guidance on standardization. https://www.etsi.org/deliver/etsi_tr/102800_102899/102863/01.01.01_60/tr_102863v010101p.pdf.

[24] ETSI EN 302 637-2: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service . https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.01_30/en_30263702v010301v.pdf, 2015.

[25] Survey on ITS-G5 CAM statistics. https://www.car-2-car.org/fileadmin/documents/General_Documents/C2CCC_TR_2052_Survey_on_CAM_statistics.pdf, 2018.

[26] P. Aivaliotis, K. Georgoulias, Z. Arkouli, and S. Makris. Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance. *Procedia CIRP*, 81:417–422, 2019. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.

[27] Giuseppe Araniti, Claudia Campolo, Massimo Condoluci, Antonio Iera, and Antonella Molinaro. Lte for vehicular networking: a survey. *IEEE Communications Magazine*, 51(5):148–157, 2013.

[28] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, and Li Zhao. Vehicle-to-everything (v2x) services supported by lte-based systems and 5g. *IEEE Communications Standards Magazine*, 1(2):70–76, 2017.

[29] Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 8 - radio-interface architecture. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *4G LTE/LTE-Advanced for Mobile Broadband*, pages 109–126. Academic Press, Oxford, 2011.

[30] Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 5 - nr overview. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *5G NR: the Next Generation Wireless Access Technology*, pages 57–71. Academic Press, 2018.

[31] Isabela Erdelean, Abdelmename Hedhli, Martin Lamb, Niklas Strand, and Ewa Zofka. Catalogue of connected and automated driving test sites : Deliverable no2.1. Number 2.1 in Deliverable STAPLE, 2019.

[32] ETSI. Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); eCall data transfer; In-band modem solution; Conformance testing . `https://www.etsi.org/deliver/etsi_ts/126200_126299/126269/12.00.00_60/ts_126269v120000p.pdf`, 2014.

[33] ETSI. Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Release description; Release 15 (3GPP TR 21.915 version 15.0.0 Release 15)). `https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=58749`, 2019.

[34] ETSI. *5G; NR; User Equipment (UE) radio access capabilities (3GPP TS 38.306 version 16.6.0 Release 16)*, 10 2021. Rev. 16.6.0.

[35] ETSI. *5G; NR; Physical layer procedures for data (3GPP TS 38.214 version 16.2.0 Release 16)*, 01 2022. Rev. 16.2.0.

[36] Edward H. Glaessgen and Doane Stargel. The digital twin paradigm for future nasa and u.s. air force vehicles. 2012.

[37] Samantha Haus, Rini Sherony, and Hampton Gabler. Estimated benefit of automated emergency braking systems for vehicle–pedestrian crashes in the united states. *Traffic Injury Prevention*, 20:S171–S176, 06 2019.

[38] WuLing Huang, Kunfeng Wang, Lv Yisheng, and Fenghua Zhu. Autonomous vehicles testing methods review. pages 163–168, 11 2016.

[39] John B. Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[40] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022, 2018. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[41] Heikki Laaki, Yoan Miché, and Kari Tammi. Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery. *IEEE Access*, 7:20325–20336, 2019.

[42] Jay Lee, Behrad Bagheri, and Hung-An Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23, 2015.

[43] Zheng Liu, Norbert Meyendorf, and Nezih Mrad. The role of data fusion in predictive maintenance using digital twin.

[44] Ryan S. Magargle, Lee Johnson, Padmesh Mandloi, Peyman Davoudabadi, Omkar Kesarkar, Sivasubramani Krishnaswamy, John Batteh, and Anand Pitchaikani. A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system. In *Modelica*, 2017.

[45] Johann Marquez-Barja, Bart Lannoo, Bart Braem, Carlos Donato, Vasilis Maglogiannis, Siegfried Mercelis, Raf Berkvens, Peter Hellinckx, Maarten Weyn, Ingrid Moerman, and Steven Latré. Smart highway: Its-g5 and c-v2x based testbed for vehicular communications in real environments enhanced by edge/cloud technologies. 06 2019.

[46] Philipp Mayer, Michele Magno, Armin Berger, and Luca Benini. Rtk-lora: High-precision, long-range and energy-efficient localization for mobile iot devices. 03 2020.

[47] Sándor Molnár, Péter Megyesi, and Géza Szabó. How to validate traffic generators? pages 1340–1344, 06 2013.

[48] Seyed Naseralavi, Navid Nadimi, M. Saffarzadeh, and Amir Reza Mamdoohi. A general formulation for time-to-collision safety indicator. *Proceedings of the ICE - Transport*, 166:294–304, 10 2013.

[49] J.R. Norris and J.R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[50] Heiko G. Seif and Xiaolong Hu. Autonomous driving in the icity—hd maps as a key challenge of the automotive industry. *Engineering*, 2(2):159–162, 2016.

[51] Quentin Spencer, Chris Peel, A. Swindlehurst, and Martin Haardt. An introduction to the multi-use mimo downlink. *Communications Magazine, IEEE*, 42:60 – 67, 11 2004.

[52] Rainer Stark, Thomas Damerau, and Kai Lindow. *Industrie 4.0—Digital Redesign of Product Creation and Production in Berlin as an Industrial Location*, pages 171–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

[53] Géza Szabó, Sándor Rácz, Norbert Reider, Hubertus A. Munz, and József Pető. Digital twin: Network provisioning of mission critical communication in cyber physical production systems. In *2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 37–43, 2019.

[54] Zsolt Szalay. Next generation x-in-the-loop validation methodology for automated vehicle systems. *IEEE Access*, 9:35616–35632, 02 2021.

[55] Elisabeth Uhlemann. Initial steps toward a cellular vehicle-to-everything standard [connected vehicles]. *IEEE Vehicular Technology Magazine*, 12(1):14–19, 2017.

[56] University of Karlsruhe (TH),Department of Mathematics. Stochastic Methods in Industry . `https://www.math.kit.edu/stoch/lehre/queues2008w/media/queues%20and%20inv.pdf`.

[57] Jian Wang, Yameng Shao, Yuming Ge, and Rundong Yu. A survey of vehicle to everything (v2x) testing. *Sensors*, 19(2), 2019.

[58] Ziran Wang. Mobility digital twin with connected vehicles and cloud computing, Oct 2021.

[59] Ziran Wang, Xishun Liao, Xuanpeng Zhao, Kyungtae Han, Prashant Tiwari, Matthew Barth, and Guoyuan Wu. A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems. 05 2020.

[60] Lars Wischhof, Fabian Schaipp, Stefan Schuhbäck, and Gerta Köster. Simulation vs. testbed: Small scale experimental validation of an open-source lte-a model. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–7, 2020.

[61] Wei Yang, Xiang Zhang, Qian Lei, and Xin Cheng. Research on longitudinal active collision avoidance of autonomous emergency braking pedestrian system (aeb-p). *Sensors*, 19:4671, 10 2019.

[62] Liang Zhang, Takao Okamawari, and Teruya Fujii. Performance evaluation of end-to-end communication quality of lte. In *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2012.