D3.3

Report on design of T2G Test Environment

Project number:	730830
Project acronym:	Safe4RAIL
Project title:	Safe4RAIL: SAFE architecture for Robust distributed Application Integration in roLling stock
Start date of the project:	1st of October, 2016
Duration:	24 months
Programme:	H2020-S2RJU-OC-2016-01-2

Deliverable type:	Report
Deliverable reference number:	ICT-730830 / D3.3 / 1.1
Work package	WP 3
Due date:	July 2017 – M10
Actual submission date:	31st of July, 2017

Responsible organisation:	UNI
Editor:	Tomas Tichy
Dissemination level:	Public
Revision:	1.1

Abstract:	Description of system design and architecture of the Train-to-ground test environment. Includes analysis of the use-cases and example test scenarios.
Keywords:	Safe4RAIL; Train2Ground; T2G; IEC 61375-2-6

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730830.

Editor

Tomas Tichy (UNI)

Contributors (ordered according to beneficiary numbers)

Cruces, Cristina (IKL)

Rodriguez, Pedro (IKL)

Pecl, Richard (UNI)

Tichy, Tomas (UNI)

Berbineau, Marion (IFS)

Kassab Mohamed (IFS)

Bouaziz, Maha (IFS)

Soler, José (DTU)

Yan, Ying (DTU)

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author’s view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Executive Summary

The IEC standard 61375-2-6 ELECTRONIC RAILWAY EQUIPMENT – TRAIN COMMUNICATION NETWORK (TCN): Part 2-6: On-board to Ground Communication defines tools and methods of data exchange between train-side and way-side parties. Such data can be used for remote management, monitoring and servicing (e.g. tracking of the train position, monitoring health of the train devices, controlling passenger information systems, updating software in the train devices, streaming video from on-board cameras, passing messages to the on-board crew). The on-board to ground communication is designed with an accent on a train operation safety and security.

The goal for the Work Package 3 Task 3.4 in Safe4RAIL project is to design a test environment which could be used to validate that the implementation of the on-board to ground (also called train-to-ground, abbreviated as T2G) communication in various components conforms to the standard mentioned above. For this reason the task focuses on designing tools for testing T2G interfaces of an on-board (mobile) communication gateway (MCG) and ground communication gateway (GCG). The MCG and the GCG are the components of the T2G communication system that directly use the communication protocol defined in the standard. Such test tools include MCG and GCG simulators, controllable hard-wired or wireless data link, ground application simulator, and also support automatic tests.

The task work is done in close cooperation with EU CONNECTA project as one of its planned output is the refinement of the T2G communication use cases and requirements, and the sample implementation of a mobile communication gateway. The T2G test environment will then be used for validation of the provided gateway.

The T2G communication test environment will enable easier integration of newly developed or modified T2G communication devices with guaranteed compatibility between components. Another outcome will be lowering the costs of the development and faster placement in the market.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

This chapter describes the purpose of this document as well as the scope of the problem solved in the Work Package 3, Task 3.4.

1.1 Purpose

The purpose of the document is to describe the design, use cases, configurability, and properties of the Train-To-Ground communication Test Environment (TE).

Train-To-Ground communication, i.e. interconnection and data exchange between train mobile on-board devices and ground applications, is defined in IEC standard 61375-2-6 [1].

The standard breaks down the structure of the system from the first level to the sub-system level. The Figure 1 below shows the high level architecture of the train-to-ground system at first level. It is divided into two main parts, which are the Mobile Communication Gateway (MCG) and GCG (Ground Communication Gateway), highlighted in grey. Both parts are connected via a wireless link. MCG and GCG are connected to other EDs on their side and provide communication services for these external systems.

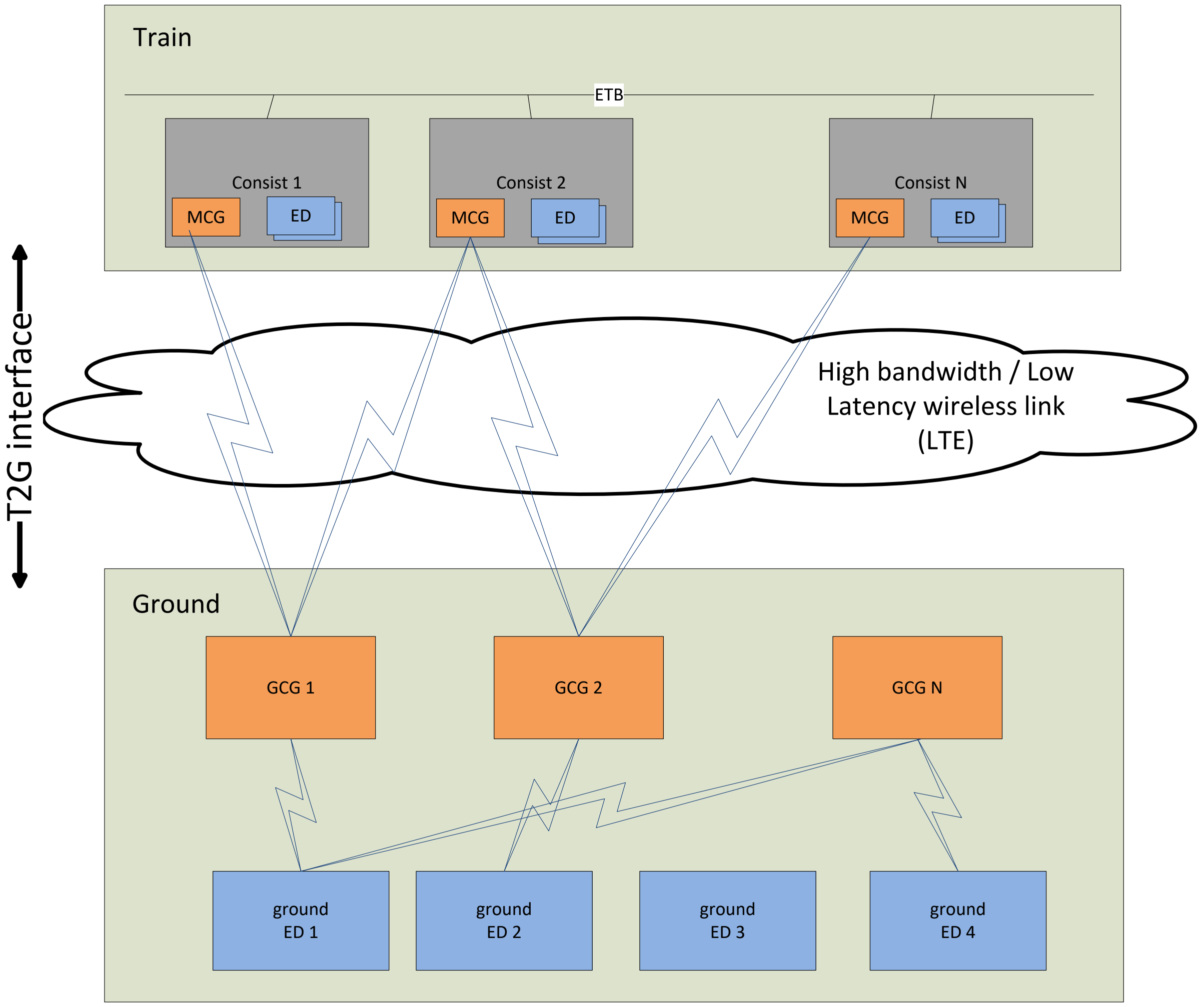


Figure 1: Train-to-ground system

The Test Environment will provide means for validation of mobile (MCG) and ground (GCG) communication devices implementation with respect to the defined T2G protocol.

The GCG component will be implemented for the proof of concept in the Safe4RAIL project, while MCG will be delivered by the CONNECTA EU project.

1.2 Scope

The scope of the project:

1. Design test environment
2. Define test tools for controlling the test
3. Define interfaces to various components (GCG, MCG, ground app and TCMS simulators) allowing interaction for the users and the test automation tools
4. Perform proof of concept (run the test with real MCG and GCG devices)

In deliverable D3.3 the output of the first 3 design tasks is documented. The design is extracted from the CONNECTA requirements [3] and from extra requirements for the T2G test environment defined in S4R deliverable D3.6 [4].

The designed test environment includes:

- Controlled wireless communication link between MCG & GCG (herein after called Access Network Simulator, ANS)
    - for testing MCG/GCG during component, integration and system tests
- GCG simulator (GCGSim)
    - for testing the MCG during component and integration tests
- MCG simulator (MCGSim)
    - for testing the GCG during component and integration tests
- Ground Application (end device) simulator (GAppSim)
    - for testing the MCG/GCG during system tests

Figure 2 depicts the scope of the project (green box). The TE allows varying configuration depending on the required test setup. For some test, the real MCG or GCG is not required (because of testing the other components or subsystems), then the MCG or GCG simulator can be used instead (boxes with a dashed line).

The TE shall test only the T2G interface between the MCG and the GCG. But for real MCG and GCG devices to work properly it may be necessary to connect some real devices (such as a vehicle control unit on the train side, or an application module on the way side).

The actors for the TE system are real human users or some test automation tools. For them the T2G TE will provide command & control interface.

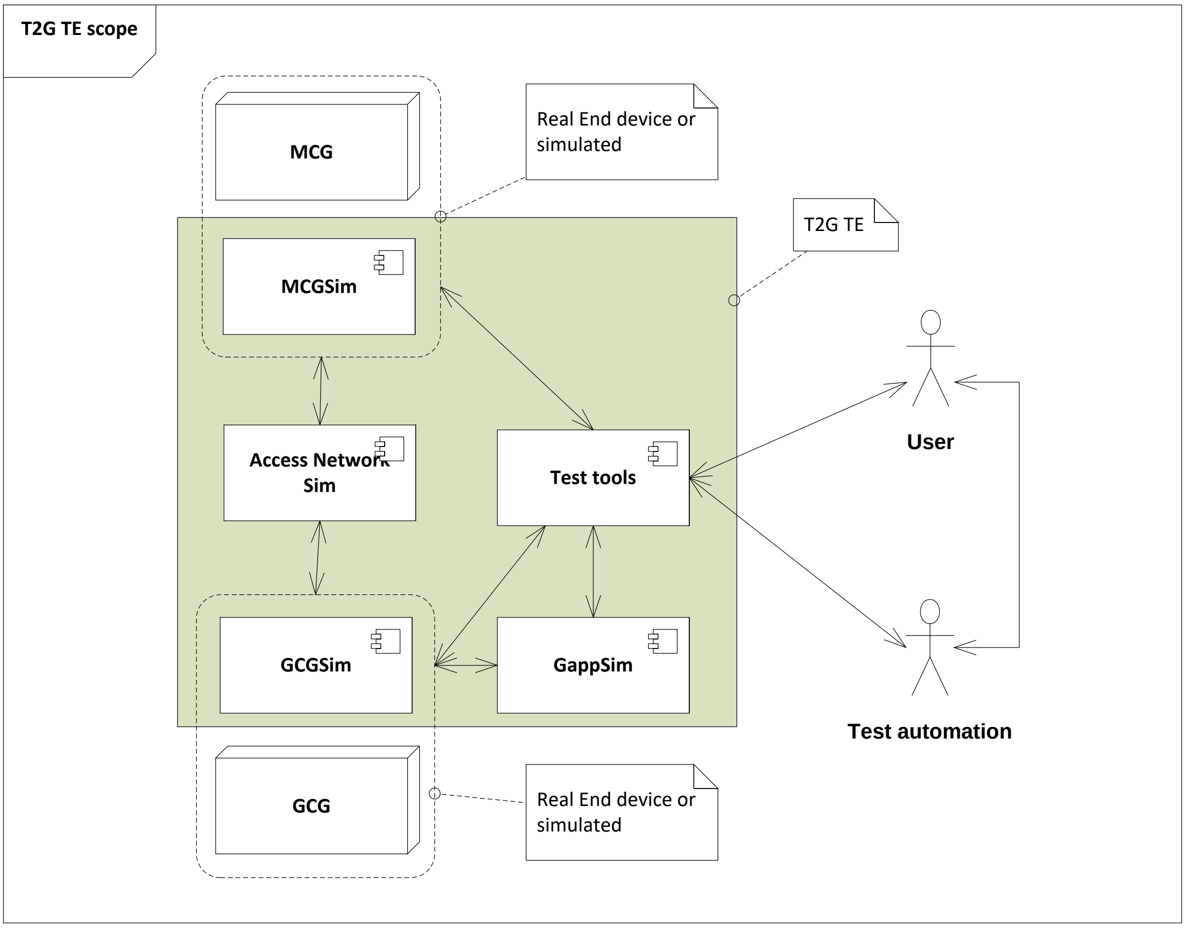


Figure 2: T2G Test Environment Scope

The three test classes (component, integration, system) and their scope are defined in [2], and summarized in 2.2.

Chapter 2 Analysis

The EU partnership project CONNECTA is elaborating the requirements for the T2G communication, and specifying the test cases for the T2G components validation. Together with the IEC 61375-2-6 standard [1], these are the base inputs to be analysed for definition of requirements on the T2G test environment. In this chapter the outcome of the analysis is presented.

2.1 Analysis of the CONNECTA T2G system requirements

In the chapter, the CONNECTA project's deliverable D2.1 – Requirements and Specification for the T2G System [3] is analysed.

These requirements are input for the T2G test environment, as they define the MCG and the GCG behaviour regarding T2G communication, thus several components of the TE should meet all or some of the requirements. Such components include for example a GCG test implementation, a GCG simulator or a ground end device simulator.

The requirements [3] are based on the IEC 61375-2-6 standard [1]. They define more thoroughly some aspects of the communication which is not fully covered in the standard [1].

2.1.1 Requirements to be covered

Each requirement is classed as *mandatory* or *optional* as specified by the standard [1]. Since the T2G TE should be able to validate all of the T2G communication possibilities, both mandatory and optional functionalities shall be implemented in the TE. On the other hand, while the scope of the Safe4RAIL project is to deliver just a proof of concept of the test environment, not all of the requirements will be covered by the implementation.

The necessary requirements to be covered by the T2G TE implementation should emerge from the CONNECTA's T2G Test Cases specification, which is not available to date.

The conclusions regarding T2G TE design defined herein are based on all requirements defined in [3]. They should be reviewed before the final implementation of the TE in Safe4RAIL project as soon as test cases will be available.

2.1.2 Communication link

The communication link to be used between mobile (MCG) and ground (GCG) systems is neither defined precisely in the standard [1] nor in the requirements [3]. In both documents the term "wireless" is used. An IP link is required in the standard [1] while Wi-Fi technology and 3G are given in the requirements [3] as examples of the wireless link with IP capabilities. For performance mobile connections, LTE is considered as possible and useful today.

Based on that and the discussions with CONNECTA partners, it emerged the *T2G TE should be ready to connect the MCG over Wi-Fi as a minimum and over LTE if possible*. Moreover CONNECTA doesn't plan to deliver the MCG with another wired or wireless interface towards the ground systems.

Safe4RAIL partners discussed various scenarios for the MCG-GCG connection for the project implementation, which highly depend on the possibilities and capabilities of the partners. For example some partners have accessibility to wireless access points, attenuators, while the others use and are familiar with software network link simulation equipment. The different scenarios are depicted in the following sections.

2.1.2.1 Setup A – simulation of complete wireless network

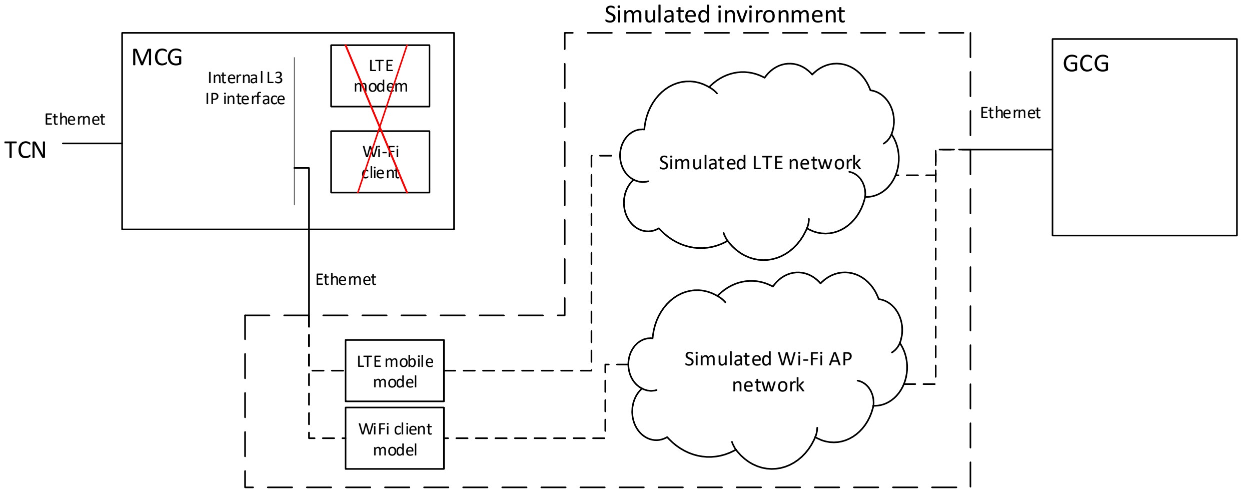


Figure 3: T2G communication link - Setup A

Features:

+ most accurate simulation of the wireless network

+ allows simulation of handovers, multiple mobile clients, network capacity…

+ suitable also for MCG software testing – doesn't require physical hardware

- requires additional internal interface in the MCG for testing

- does not test physical wireless interface of the MCG

2.1.2.2 Setup B – lab WiFi

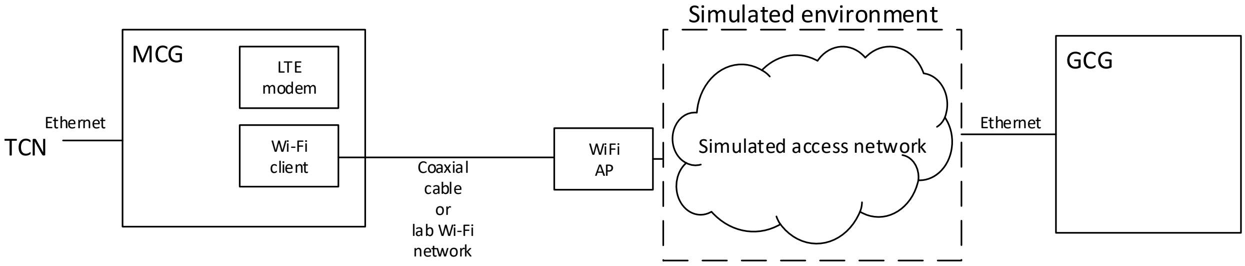


Figure 4: T2G communication link - Setup B

Features:

+ cheap and easy to setup

+ no requirements for testing the interface in the real MCG

- limited to WiFi only

- cannot test real handovers (effects of the handover can be simulated)

2.1.2.3 Setup C – simple LTE network

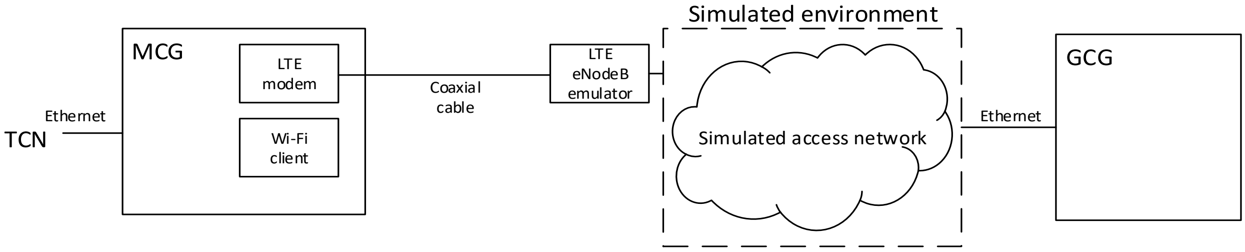


Figure 5: T2G communication link - Setup C

Features:

+ no requirements for testing interface in the real MCG

- limited to LTE only

- cannot test handovers (effects of the handover can be simulated)

2.1.2.4 Setup D – physical channel emulator

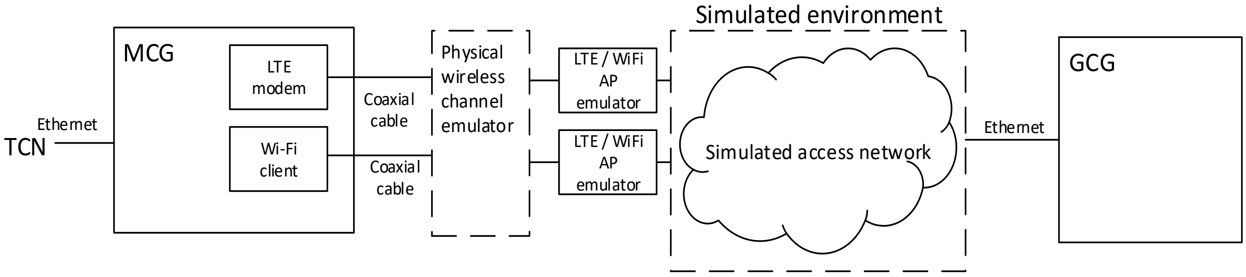


Figure 6: T2G communication link - Setup D

Features:

+ no requirements for testing the interface in the real MCG

+ can test real physical environment, interferences etc.

+ can test handovers

- most complicated test setup

- requires expensive lab equipment – LTE emulators, channel emulator, RF equipment

2.1.2.5 Setup E – LTE network of public operator

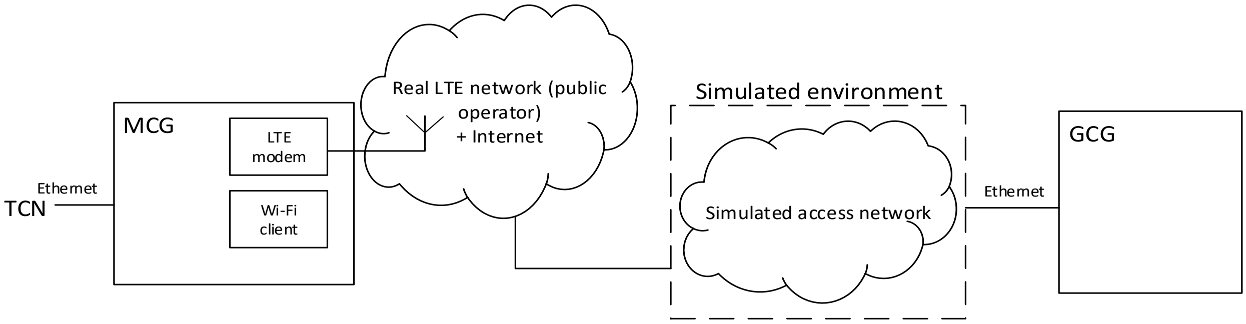


Figure 7: T2G communication link - Setup E

Features:

+ no requirements for testing the interface in the real MCG

+ easier to setup compared to using real wireless base station equipment

- uncontrolled environment – limited test reproducibility, suitable only for demonstration purposes

- access network simulator used only for test monitoring purposes

From the above, the following setups have been chosen for the implementation:

Setup A:

- For its completely controllable com. link behavior producing exactly the same results on the same repeated test (test reproducibility)
- Although the real MCG implemented in the CONNECTA project will not provide an additional Ethernet interface for the “wireless” connection, this setup can be useful as a proof of concept for validation of future MCG implementations. In some implementations it could be important or valuable to validate MCG functionalities under various aspects of T2G link behavior.

Setup B (possibly combined with Setup E):

- Provides a reasonable level of link control (especially in the case when a coaxial cable is used between MCG and Wi-Fi access point (AP))
- While MCG’s LTE is not involved here, it can be combined with setup E, thus for example allowing the Network Selector on the MCG to be validated. The Network Selector selects the T2G communication interface from several of them (when available) depending on the actual communication requirements and link performances.

While other setups could also be used for the T2G communication validations, it is not possible to implement them in the scope of the Safe4RAIL project now (mainly because of lacking equipment).

2.1.3 Routing & Application Level Gateway modes

The requirements [3] specify two modes a MCG shall be able to operate in with respect to exchange communication data between ground and on-board devices):

- Routing mode
    - MCG shall act as an IP router (network level in OSI model)
    - Routing firewall shall be used for controlling traffic
    - Both the standard [1] and the requirements [3] propose the usage of a Virtual Private Network (VPN) for securing the connection
    - Direct communication between ground and mobile end devices is possible
    - Not allowed in normal train operation for safety reasons
- Application Level Gateway mode (ALG)
    - MCG shall act as an application gateway (application level in OSI model)
    - MCG does not allow a direct connection to end devices
    - MCG converts (and checks) messages from the T2G interface format (HTTP messages) to an on-board interface (TRDP e.g.) and vice versa

To validate both modes, the T2G TE shall implement both of them. Specifically a GCG shall provide the possibility to use the routing mode & also the ALG mode by a ground end device and/or test equipment.

Neither the requirements [3] nor the standard [1] define how the data exchange mode should be switched. In the requirements [3] and the standard [1], there are listed some of the train operational conditions under which the routing mode is allowed (CCTV, maintenance time etc.), but it is not defined if routing is allowed automatically, on request and what kind of request (remote message, local user interaction).

Neither the requirements [3] nor the standard [1] define the type or the implementation of a VPN proposed to be used to secure the routing mode.

2.1.4 IP address plan

In the requirements [3] and the standard [1], there is no IP address assignment plan defined for MCG, GCG and end devices.

Because the T2G protocol requires TCP/IP to be used for communication between the MCG and the GCG/GED, and request is initiated from both sides depending on service used, an MCG must be accessible directly from a GCG/GED and vice versa.

This can be achieved if either:

- there is no NAT router (changing IP addresses in the packets) between MCG and GCG/GED, or
- some kind of VPN is used – such VPN can be T2G communication specific (VPN connection is established from the MCG to the GCG first, and then all communication is performed inside this VPN), or provided by a mobile network operator with dedicated Access Point Name (APN).

If the IP addressing is not in the scope of the standard [1], we should assume that for T2G validation, all devices (MCGs, GCGs, both on-ground and on-board devices) have unique IP addresses and each device is reachable from any other using this device address.

2.1.5 DNS

It is defined in the requirements [3] that MCGs are reachable from the ground by a unique symbolic name – an extended TCN-URI. For the actual IP address resolution, a DNS system shall be used.

A GCG shall implement a dynamic DNS, registering each connected MCG with its consist ID and train journey (forming extended TCN-URIs) and associated MCG's IP address. Furthermore, a GCG shall register on-board DNS server for forwarding resolution of names in train domain.

The requirement (informative) ID 1262_MCG_MCGImpl_217 states that: "Onboard TCN-DNS name resolution for ground services will be provided on the ground side. The MCG does not need to have a name resolution service." But other requirements (1262_MCG_MCGImpl_330 and 1262_MCG_GCG_12) and also the standard [1] definition contradict that.

It is assumed for the T2G validation that an on-board DNS server will resolve URIs in the train domain.

2.1.6 File transfer service

While the requirements [3] (1262_MCG_MCGImpl_280) only list possible protocols to be used for file transfer service (HTTPS, FTP, SFTP), the standard [1] defines HTTP(S) as mandatory and the others as optional.

For the T2G TE, HTTP(S) protocol is considered as mandatory for the file transfer service.

2.1.7 Telemetry service (MQTT)

The requirements [3] specify an additional (optional) service – Telemetry service, which is not defined by the standard [1]. The protocol defined to be used by the service is Message Queue Telemetry Transport (MQTT) (for its better performance in comparison to HTTP messages).

In case of telemetry services validation, *a GCG in the T2G TE shall act as a MQTT broker* allowing topics subscription from both, an MCG and ground end devices. An MCG will subscribe to GCG's broker in order to publish telemetry data, the ground end device will subscribe to GCG's broker in order to receive such data.

The requirements [3] define functions regarding telemetry control - creating groups of variables for telemetry transmission, listing the groups, transmit cancelation etc. But the requirements [3] do not define the means for the control (e.g. message format and protocol). In the Conclusions chapter in the requirements [3] there is only a remark about sending a proposal IEC WG43 to add a chapter regarding telemetry to the standard [1].

2.2 Analysis of the CONNECTA T2G system test plan

This chapter presents outputs from the analysis of the Master test plan [2] defined by CONNECTA.

The Master test plan (MTP herein after) is used as the overall test planning and management in CONNECTA WP2 T2.3. The test plans involves designing the T2G interface and its validation with respect mainly to the development of the MCG, but also to the whole TCMS-MCG-[T2G]-GCG-ED chain.

2.2.1 Identification of the applicable test classes

The MTP defines a terminology for test levels as common base for all of the tests to be performed during development, testing and validation of the products. The three levels are listed and described with their appropriate application in Table 1.

Term	Description
Component Test	Component testing refers to testing a component of the T2G communication system. The MCG and GCG are components of the T2G communication system, so the term **component test** therefore refers to testing a device, comprising both HW+SW, of the T2G communication system.
Integration Test	Integration testing refers to integrating the MCG (or GCG) into a T2G communication system. The integrated component shall be tested for interacting with the environment according to expectations. The term **integration test** therefore refers to integrating the MCG (or GCG) into a T2G communication environment. The MCG (GCG) is integrated into a test environment that emulates a GCG (MCG).
System Test	System testing refers to testing the whole T2G communication chain. The T2G communication chain contains at least • an MCG on the train • a GCG on the ground • a data **source** on the train and a data **sink** on the ground **or**: a data **sink** on the train and a data **source** on the ground The term **system test** therefore refers to testing the combination of all these components, working together to provide T2G communication capabilities for one TCMS function.

	The MCG component interacts with the test environment emulating the GCG. Together they form a part of the T2G system which is tested. In a second step, the MCG interacts with the GCG, forming a part of the T2G system which is tested. The scope of WP2 T2.3 is to test at least one TCMS function using the T2G communication chain. The T2G communication chain will contain: • as a first step: the MCG, the test environment developed in S4R acting as the GCG and one pair of a data-source/sink for the on-board/ground side. • as a second step: the MCG, a GCG and one pair of a data-source/sink for the on-board/ground side. If the integration tests of the MCG and GCG did not take place during integration testing, both devices have to be integrated and tested in this phase first, according to the tests defined in the integration test phase.

Table 1: Test level terminilogy and application [2]

2.2.2 Test plan

According to the MTP and the test classification, the component testing and validation progress should be as following:

1. Development tests of the developed components performed by contributors on their own:
    a. MCG (CTA)
    b. TE / Access Network Simulator – emulated (S4R)
    c. TE / Access Network Simulator – real (S4R)
    d. TE / GCG simulator (S4R)
    e. TE / MCG simulator (S4R)
    f. TE / Ground end device simulator (S4R)
    g. GCG (S4R)
2. Integration tests incorporating components
    a. MCG (CTA) + Access Network Simulator (S4R) + GCG simulator (S4R)
    b. MCG (CTA) + Access Network Simulator (S4R) + GCG (S4R)
3. System wide tests (test with real MCG & GCG components)
    a. TCMS (CTA) + MCG (CTA) + Access Network Simulator (S4R) + GCG (S4R) + Ground end device simulator (S4R)

Figure 8 shows a timeline for each level of the tests and components involved [2].

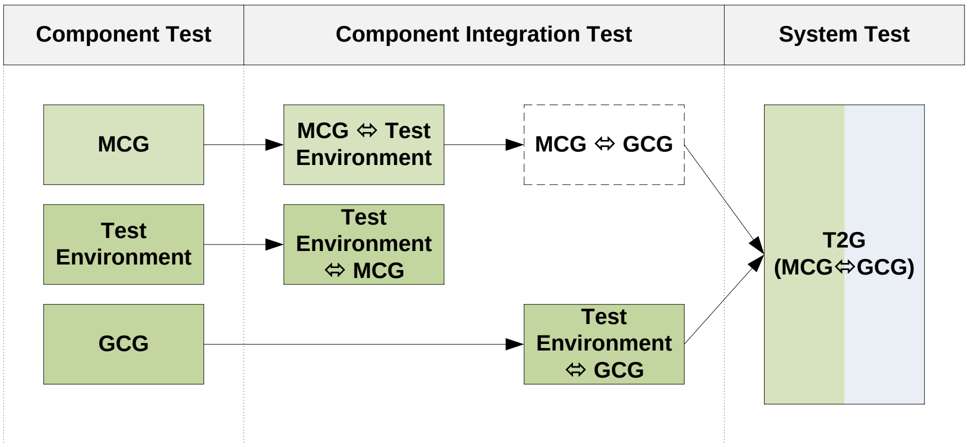


Figure 8: Timeline for test levels and components involved [2]

2.3 Requirements for the T2G TE

The requirements for T2G test environment are defined in [4], specifically relevant ones have the prefix ‘RWP3_11’ in their ID.

The TE shall provide means to test & validate the Mobile Communication Gateway placed on the train together with the Ground Communication gateway placed in the operator’s facility with respect to the communication interface defined in the IEC 61375-2-6 standard [1].

The TE shall be able to validate both components of the respective railway & ground equipment and the system in its entirety.

Those requirements are based on the TCN standard as well as on use cases and requirements specified in [3].

Chapter 3 T2G TE High level design

This chapter describes the T2G Test Environment system, its components to be developed, and interconnections between individual components including command, control and diagnostic interfaces.

3.1 T2G TE General Architecture

The general TE architecture is depicted in Figure 9 (deployment for the most comprehensive system test in Setup B).

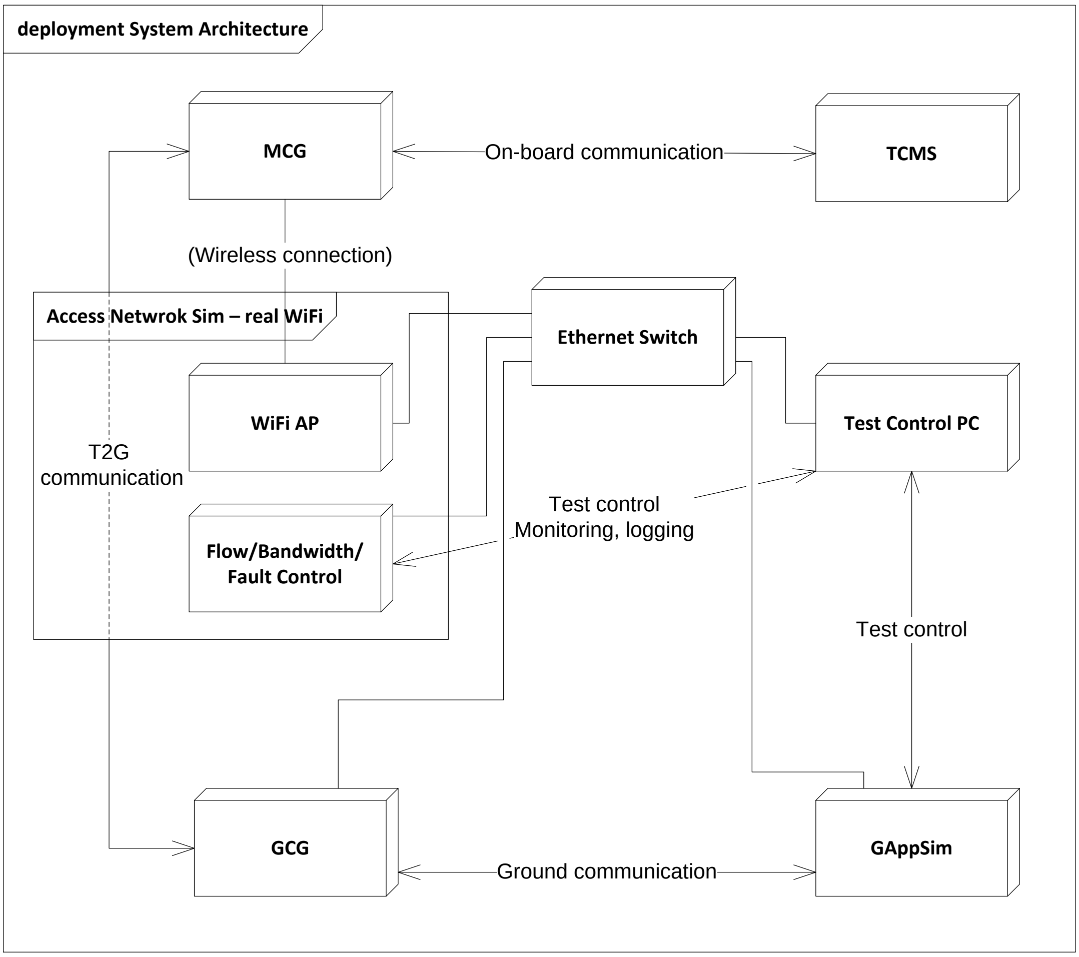


Figure 9: T2G TE general architecture

3.1.1 T2G TE configurations for test classes

The Test Plan [2] defines three levels: component test, integration test and system test (see Table 1). For each test level, different parts of the TE will be setup.

3.1.1.1 Component tests:

- A component of the system is tested (usually by vendor himself)
- Configuration:
    - Specific component
    - Vendor's test tools
- E.g. real GCG's HW & SW is tested with test tools which are able to generate and consume T2G communication data (HTTP messages) and ground application data on the ground-side interface.
- Figure 10 depicts a test configuration for a component test of a real GCG. A similar configuration is used for an MCG component test.

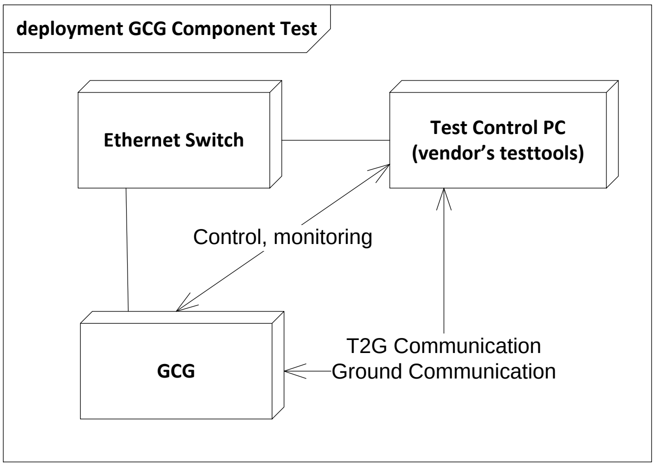


Figure 10: GCG component test configuration

3.1.1.2 Integration tests:

- Several components of the system are tested together
- In case of T2G interface testing, integration of a component into the Test Environment will be tested
- Configuration:
    - A component (GCG or MCG)
    - Optional an Access Network Simulator in case of testing MCG
    - Test tools
- Figure 11 depicts configuration for integration test of MCG into TE

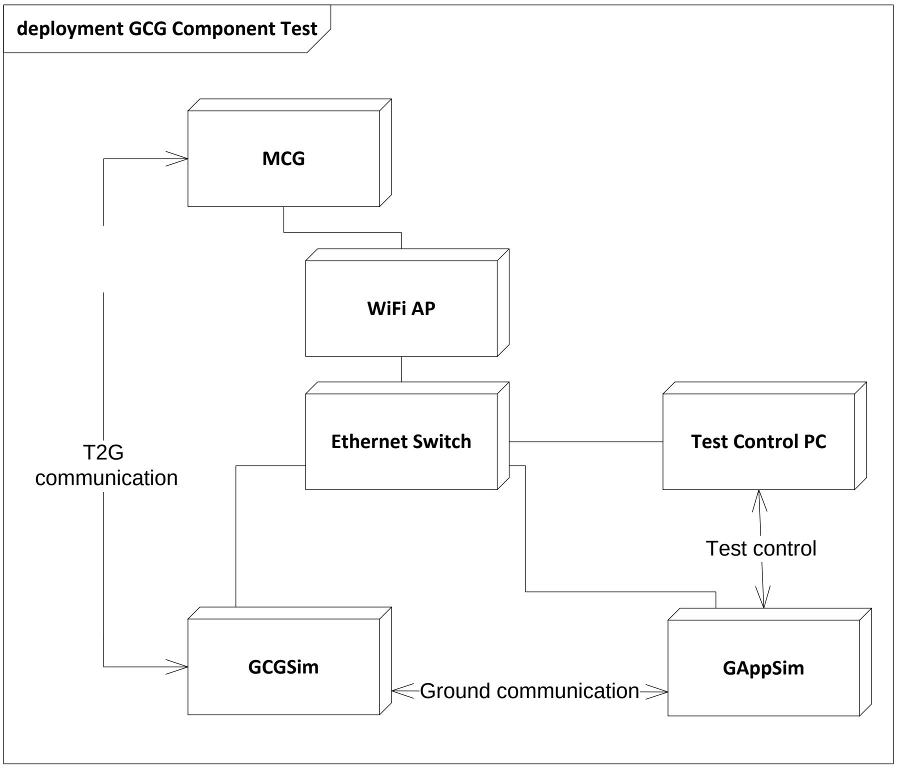


Figure 11: MCG & GCG integration test configuration

3.1.1.3 System tests:

- Complete T2G system is tested
- Configuration (see Figure 12):
  - MCG + on-board data endpoint (TCMS),
  - GCG + on-ground data endpoint (GAppSim),
  - Access Network Simulator
  - Test tools

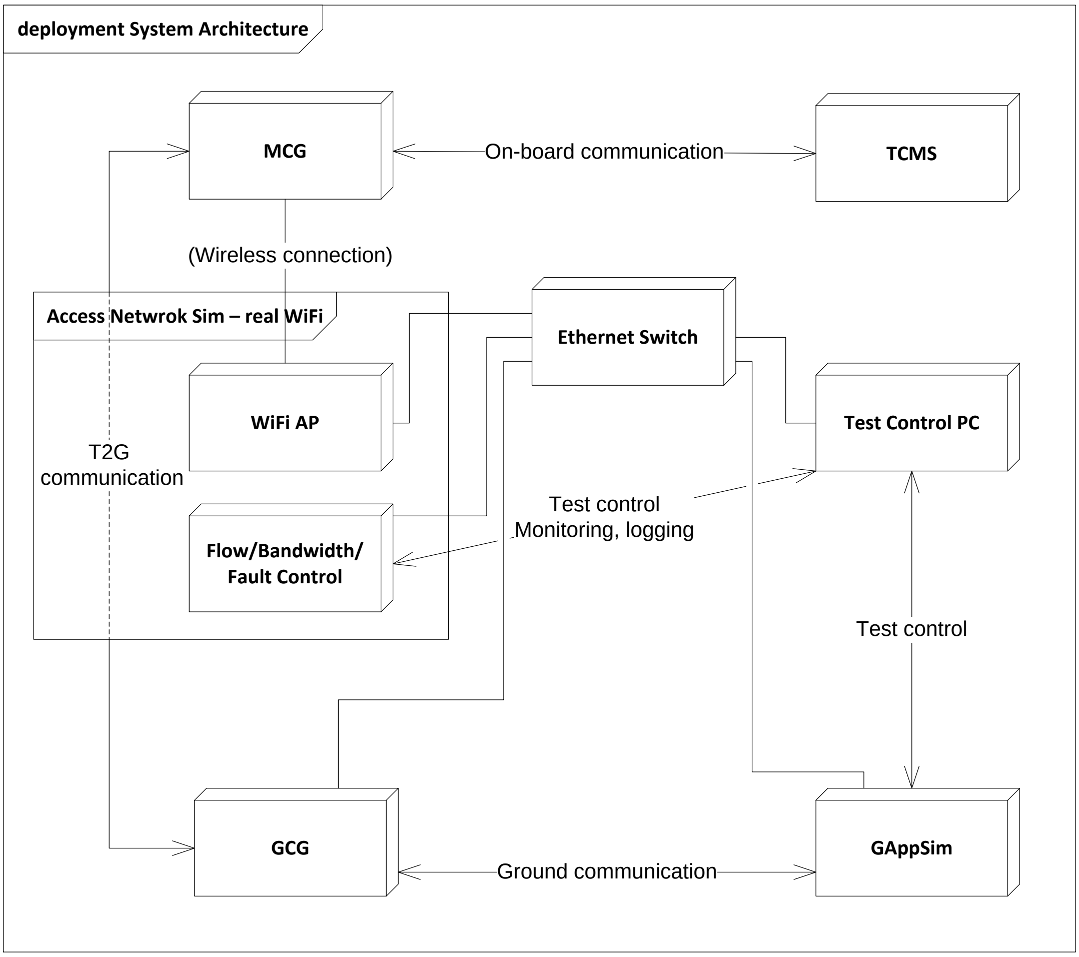


Figure 12: System test configuration

3.2 T2G TE Components

In this chapter, the individual components used in the TE system are described.

3.2.1 Access network simulator – emulated wireless connections

The (LTE + WIFI) network simulator based on Riverbed Modeler is in charge of providing a network simulator that allows the interaction between the MCGs, the GCGs and different configuration of the T2G network. This simulator is able to offer statistics about performances perceived by data exchanges through the T2G network regarding the latency, jitter, throughput, message errors and link. This allows testing the behaviour of the MCGs and GCGs under different network architectures and loads.

The LTE network simulator based on Riverbed Modeler is responsible for evaluating the network performances for the high speed train communication. The simulator provides the use case scenarios to configure the base station attributes, the MCG mobility attributes and the various applications offered by GCGs.

The hardware/software co-simulation simulator based on the OpenAirInterface (OAI) and Riverbed System in the Loop (SITL) interfaces aims to combine the simulated network for the LTE core network with the real LTE traffic.

3.2.1.1 Scope Model

The proposed (LTE+WIFI) network simulator will be based on Riverbed Modeler modules.

The Mobile Communication Gateway (MCG), the Ground Communication Gateway (GCG) and an external user to configure and control the system are shown in Figure 1 as the actors of the system, they are described below:

- Mobile Communication Gateway (MCG): train-side system which provides board-to-ground services to the on-board end devices.
- Ground Communication Gateway (GCG): For each train fleet, a Ground Communication Gateway (GCG) is defined and initiated to manage fleet train to ground communication. From ground side, this GCG is used as an IP gateway to communicate with any consists (including a Mobile Communication Gateway, MCG) of the fleet. GCG is connected to the system using the regular Ethernet interface
- User: an external user that configures, reconfigures, starts, stops the system. It uses the graphical user interfaces of Riverbed Modeler.

In the High speed scenario cases for rural environment scenario, we consider a standard LTE network for the T2G communication in Safe4RAIL. The scope model is shown in Figure 13, which represents the interactions between the users (or actors) of the system, and the system. Users simulate an LTE network with the User Equipment UE, eNodeB and core network (EPC) in the Riverbed modeller environment. Users interact with the simulator by configuring various LTE network attributes and collecting statistics in the Riverbed simulator. The goal of the high speed scenarios will analyse the impact of the LTE eNodeB configuration, the LTE UE mobility, and the offered traffic load on the application transmission performance.

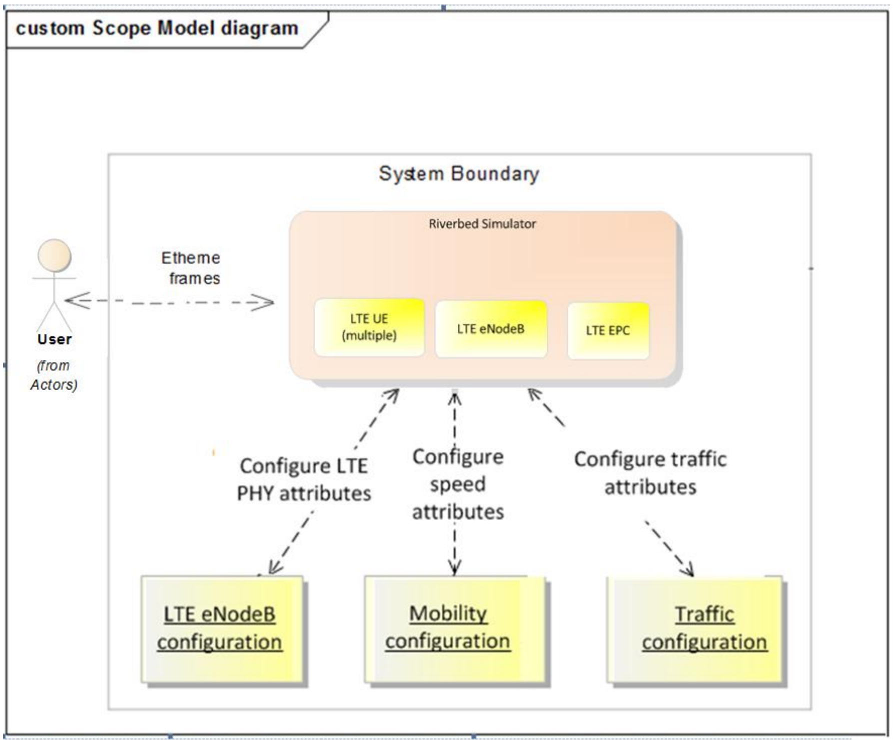


Figure 13: Scope model of the LTE network simulator

In the SITL integration with OpenAirInterface scenario, the co-simulation system interacts with both the OAI subsystem and the Riverbed Modeler via the SITL interface (as shown in

Figure 14). The OAI subsystem is dedicated to the LTE eNodeB implementation. The Riverbed simulator contains the real-sim and sim-real interfaces via the SITL gateways and the LTE core network functionalities via the LTE EPC module.

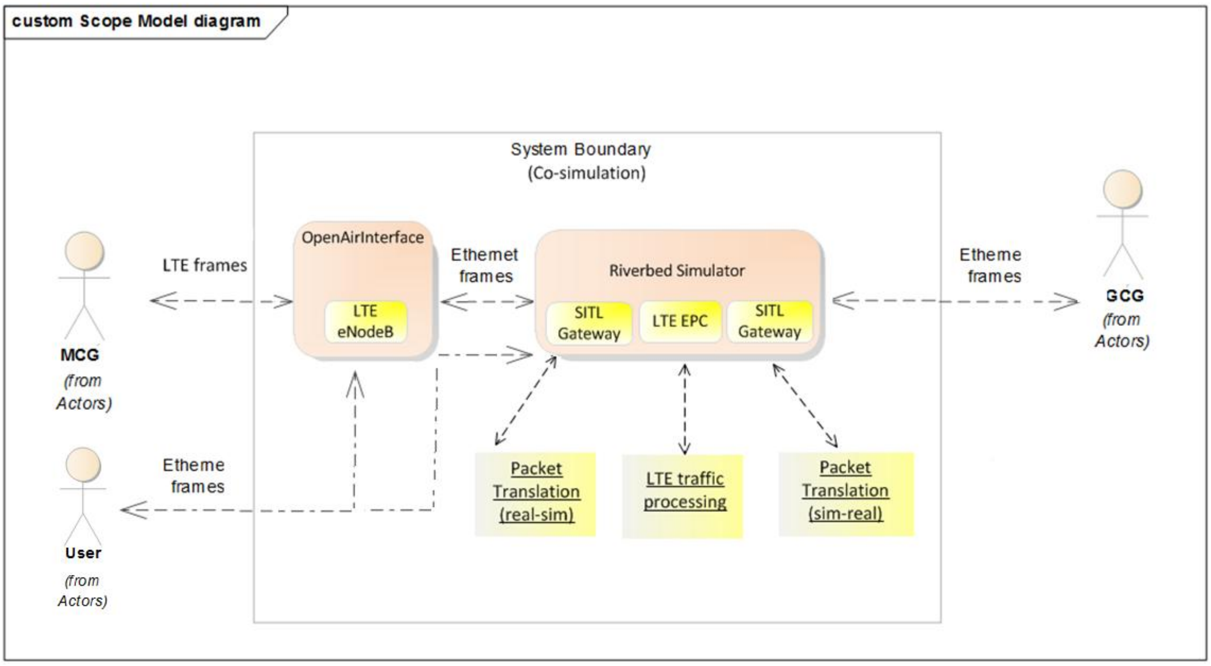


Figure 14: Scope model of the hardware/software co-simulation network simulator

3.2.1.2 Use Case Model

- The system is configured using the graphical user interface of Riverbed Modeler
- The statistics to be recorded are defined
- The duration of the simulations are fixed
- The simulations are launched
- The simulator generates files with the expected data

The Use Case model describes the relation between the system input and output. The use cases for LTE model and the hardware/software co-simulation model are described as following:

3.2.1.2.1 Use Case 1: Configuration to Riverbed LTE model

Goal: configure the Riverbed LTE system to be ready to start its operation. The necessary information is related to:

- LTE UE attributes
- LTE eNodeB PHY and MAC attributes
- LTE EPC attributes
- Application and service definition

3.2.1.2.1.1 Scenario 1: LTE eNodeB configuration

Precondition: the LTE system works successfully.

Steps:

1. UE communicates with the application server via LTE eNodeB and LTE EPC.
2. Configure the eNodeB transmission power.
3. Configure the number of deployed eNodeB.
4. Collect network performance results and statistics.
5. END.

3.2.1.2.1.2 Scenario 2: Train speed.

Precondition: the LTE system works successfully.

Steps:

1. UE communicates with the application server via LTE eNodeB and LTE EPC.
2. Configure the speed of UE trajectory.
3. END.

3.2.1.2.1.3 Scenario 3: Traffic load.

Precondition: the LTE system works successfully.

Steps:

1. UE communicates with the application server via LTE eNodeB and LTE EPC.
2. Configure the email application.
3. Configure the file transfer application.
4. Configure the voice communication application.
5. Configure the video streaming application.
6. END.

3.2.1.2.2 Use Case 2: Configure the OAI-Riverbed co-simulation system

Goal: configure the OAI LTE eNodeB system, the real-sim-real SITL interface and the Riverbed LTE EPC simulation to be ready to start its operation. The necessary information is related to:

- Install OAI LTE eNodeB
- SITL gateway attributes
- LTE EPC attributes

Steps:

1. Install and configure OAI eNodeB.
2. Configure networking settings at eNodeB
3. Set up SITL gateway and link
4. Configure SITL interface
5. Configure LTE features in EPC model
6. Test the communication and collect results via SITL statistic and EPC statistic.
7. END.

3.2.1.3 Architecture Models

The architecture models represent the different options for test environments defined based on of the access network simulator to be implemented. Three architecture models of the (LTE+WIFI) Riverbed based access network simulator are proposed. This section presents the internal interactions between riverbed modules (Simulation runtime model Wireless model, LTE specialized model, IPV6 model, System-In-the-Loop model) as well as the OpenAirInterface system and the internal interactions between the users and the network simulator.

3.2.1.3.1 Test environment A: Pure simulation

With the pure simulation case, there is no interaction with real equipment. MCGs and GCGs are simulated. This architecture enables the evaluation of the effect of network configuration, traffic load and mobility on data exchange performances.

The proposed TE using a pure simulation does not need much equipment allowing the possibility to test the two existing technologies of wireless networks (LTE and Wi-Fi). In addition, the idea of using the simulated wireless networks in the environment of simulation may provide more accurate simulation results. This part focuses on the design of the TE corresponding to the railway central station, characterized by the low speed for the urban environment. This TE will allow us testing the communication behavior of MCG during mobility, which moves across the two technologies.

Thus, an optimized handover management mechanism (fast vertical handover) is required in order to provide continuous connectivity of the MCG [4]. Moreover, the handover triggering between the two access technologies is based on usual parameters such as radio conditions (SNR and BER) and user preferences. In addition, it is possible to propose scalability performance study for MCG and GCG, based on the extension of the simulated mobile clients number.

3.2.1.3.2 Test environment B: SITL integration with OpenAirInterface

With the architecture model in Figure 15, The Riverbed based access network simulator is able to connect one GCG using the Ethernet interfaces and multiple MCGs through an LTE wireless link. This architecture enables the evaluation of the effect of network configuration on data exchange performances of a Real MCG but not the traffic load and the mobility effects. The co-simulation network framework is composed of three devices:

- USRP B210: The USRP B210 provides a fully integrated, single-board, Universal Software Radio Peripheral (USRP™) platform with continuous frequency coverage from 70 MHz – 6 GHz. This device is used to transmit and receive LTE signal from the LTE UE.
- OAI LTE eNodeB: provides a base station which allows the connection of the MCGs and transform the LTE frame into an Ethernet frame and vice versa.
- Central PC: provides the LTE EPC functionalities for data communication between the MCGs and GCGs. Riverbed SITL interface is used to translate packets between the real and simulated environment.

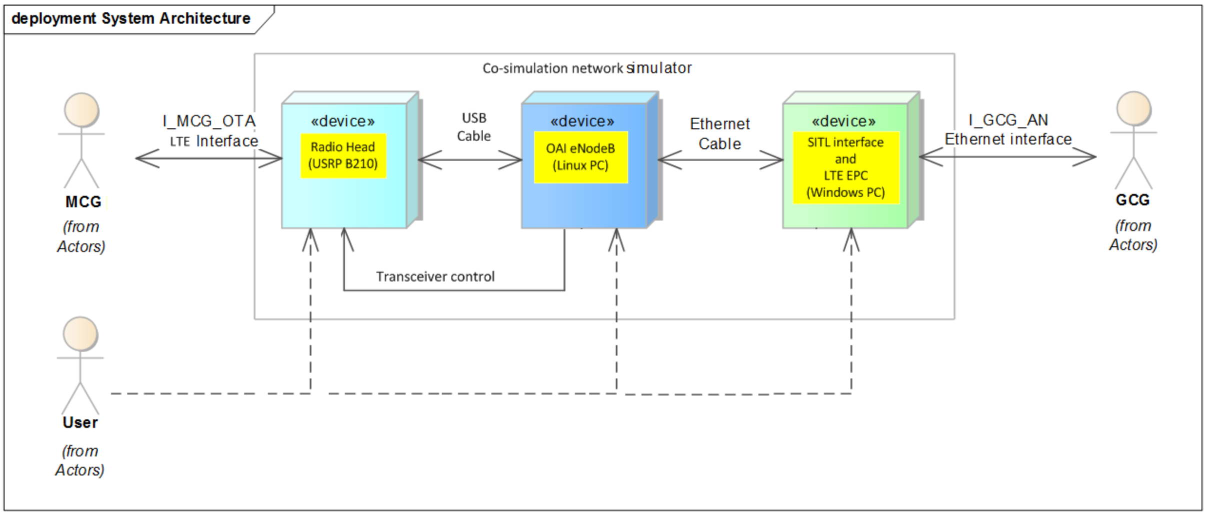


Figure 15: Architecture model of the network simulator with OpenAirInterface Real MCGs and one Real GCG

This Test Environment (shown in Figure 16) presents an in-lab mobile network demonstration, which enables the investigation of LTE based communication network. In this setup, the hardware measurements and software simulation environment are connected.

This scenario aims to setup real wireless communication between the Mobile Communication Gateway (MCG) equipment and the LTE eNodeB under realistic conditions. For example, the radio channel environment and the transmission power are changed due to the signal interference. The results reflect measurements from real networks. The LTE backhaul network that interconnects base stations to the Internet is implemented under the simulated environment. The railway services are further delivered to the Ground Communication Gateway (GCG) in order to view the realistic performances.

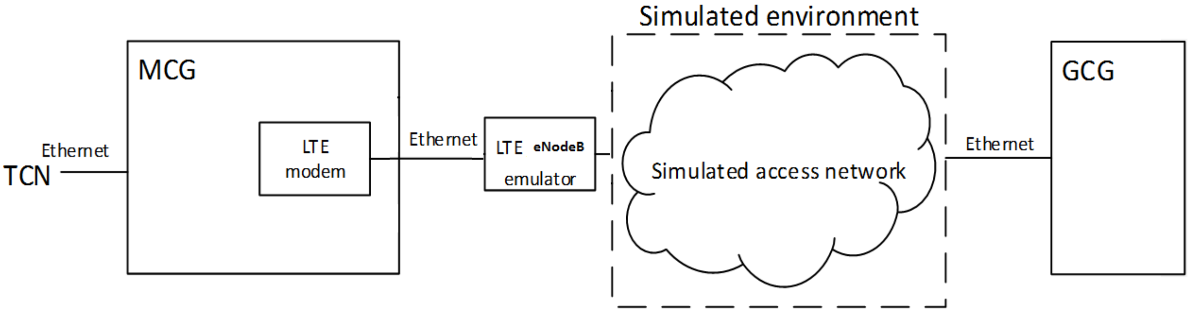


Figure 16: Physical channel emulator

In this scenario, we evaluate railway services running on a real mobile device (to mimic the MCG equipment). Communication is established between the user equipment to the eNodeB with consideration of effects of the radio access. On the other hand, the LTE backhaul, which consists of a large amount of equipment in reality, is simulated to simplify the setup.

3.2.1.3.2.1 Elements description from the scenario:

The overall setup as realization of the proposed emulation is depicted in Figure 17. It is a real-simulation-real implementation. An application (e.g. data transfer or video streaming) running on a mobile device is connected via a real mobile link to an emulated eNodeB. The LTE backhaul, EPC, is simulated by Riverbed Modeler. The connection to the eNodeB and the application servers is using Ethernet in real environment.

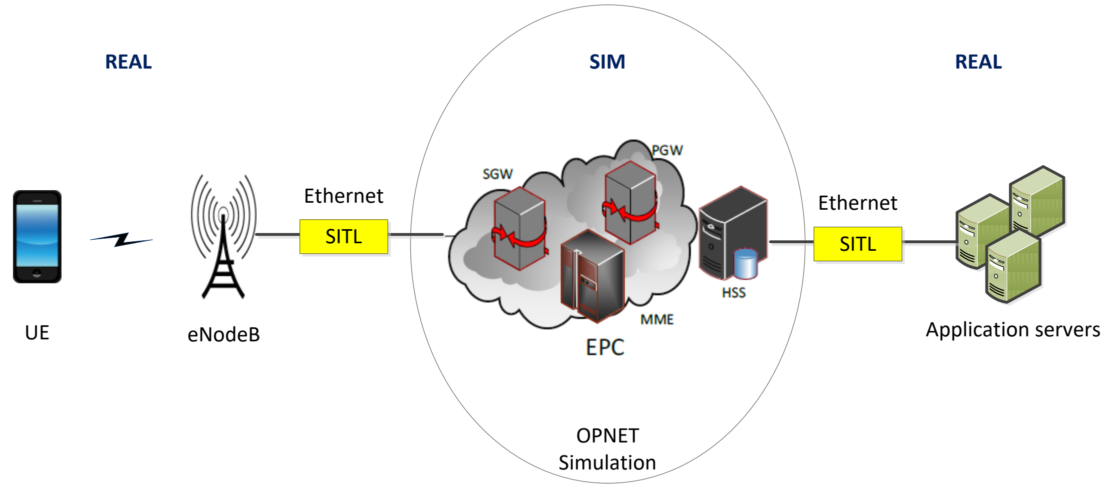


Figure 17: The real-sim-real experimental setup

3.2.2 Access network simulator – real Wi-Fi

The Wi-Fi access network simulator is in charge of providing a network simulator that allows the interaction between the MCGs and the GCGs. This simulator can monitor and manage the latency, jitter, throughput, message errors and link failure of the communication of the network. This allows testing the behaviour of the MCGs and GCGs under different situations of the communication.

3.2.2.1 Scope Model

The model for the Wi-Fi Access network simulator, which represents the boundary of the system to be designed, is shown in Figure 18. Within this model, the users (or actors) of the system, the libraries (or entities) contained within the system and the inter-relationship between them are defined.

The Mobile Communication Gateway (MCG), the Ground Communication Gateway (GCG) and an external user to configure and control the system are shown in Figure 18 as the actors of the system. They are described below:

- Mobile Communication Gateway (MCG): train-side system which provides the on-board end devices with board-to-ground services.
- Ground Communication Gateway (GCG): For each train fleet, a Ground Communication Gateway (GCG) is defined and initiated to manage fleet train to ground communication. From ground side, this GCG is used as an IP gateway to communicate with any consists (including a Mobile Communication Gateway, MCG) of the fleet.
- User: an external user which configures, reconfigures, starts, stops and injects faults in the system. It sends Ethernet frames to command the system.

In Figure 18 the part of the system which we are going to develop is represented within a rounded square, meanwhile the rest of the existing system components (libraries or entities) are represented with square boxes. The entities used in the system are:

- Network traffic capturing entity: library used to capture Ethernet frames which are received by the system.
- Wi-Fi Access Point (Wi-Fi AP) configuration entity: library to configure the Wi-Fi AP. The wireless network will be configured by the system using this library.

- Attenuator configuration entity: library to configure an attenuator which attenuates the wireless signal of the Wi-Fi AP. This allows emulating a lost connection in the wireless network.

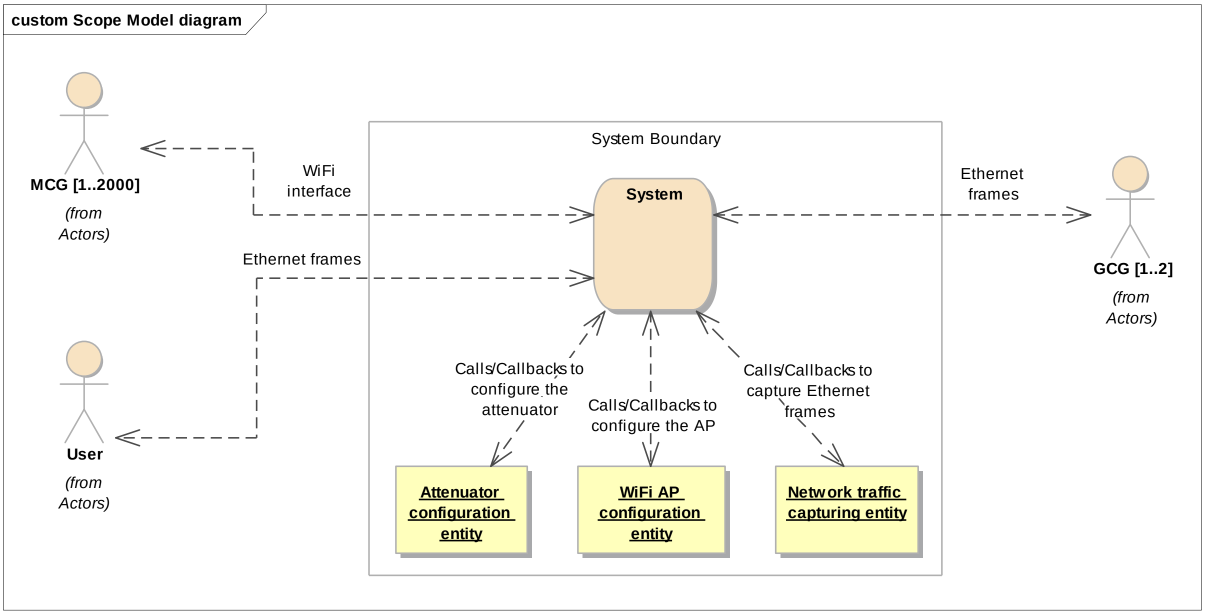

custom Scope Model diagram
System Boundary
System
MCG [1..2000]
(from Actors)
User
(from Actors)
GCG [1..2]
(from Actors)
WiFi interface
Ethernet frames
Ethernet frames
Calls/Callbacks to configure the attenuator
Calls/Callbacks to configure the AP
Calls/Callbacks to capture Ethernet frames
Attenuator configuration entity
WiFi AP configuration entity
Network traffic capturing entity

Figure 18: Scope model of the network simulator

3.2.2.2 Use Case Model

The Use Case model is a catalogue of repeatable interactions or steps that a user (or “actor” in the Unified Modelling Language (UML)) experiences when using the system. A Use Case includes one or more “scenarios”, and each scenario describes the interactions between the actor and the system.

3.2.2.2.1 Use Case 1: Configuration

Goal: configure the system to be ready to start its operation. The necessary information is related to:

- Access Point (AP) configuration.
- Attenuator configuration.
- Identification of the Wi-Fi interface (Identification and IP).
- Identification of the Ground Communication Manager (GCG) interface (Id. and IP).
- Priority or non-priority based operation.
- Maximum throughput of the network.
- Delay and jitter of the network.
- Packet Error Rate (PER).

3.2.2.2.1.1 Scenario 1: The configuration file is correct.

Precondition: the system has been switched on and it is waiting for the configuration file. The file sent by the user is correct.

Steps:

1. The user sends the configuration file to the system.

2. The system analyses the file and configures itself.
3. The system configures the attenuator to prevent the connection of the MCGs (attenuation high).
4. The system confirms to the user that the configuration has been done correctly.
5. END.

3.2.2.2.1.2 Scenario 2: The configuration file is incorrect.

Precondition: the system has been switched on and it is waiting for the configuration file. The file sent by the user is incorrect.

Steps:

1. The user sends the configuration file to the system.
2. The system analyses the file and detects there is an error.
3. The error is reported to the user.
4. END.

3.2.2.2.2 Use Case 2: Start

Goal: start the operation of the network simulator.

Precondition: the system has been correctly configured.

Steps:

1. The user sends the 'start' command to the system.
2. The system configures the attenuator to allow the connection of the MCGs (attenuation low).
3. The monitoring process starts.
4. Ethernet frames capture starts.
5. The system manages the received frames taking the configured Quality of Service (QoS) parameters into account before sending these frames to the destination.
6. The system confirms that it is running correctly to the user.
7. END.

3.2.2.2.3 Use Case 3: Stop

Goal: stop the operation of the network simulator.

Precondition: the system is running correctly.

Steps:

1. The user sends the 'stop' command to the system.
2. The system configures the attenuator to lose the connection of the MCGs attenuating the Wi-Fi signal.
3. Ethernet frames capture stops.
4. The system stop the current monitoring session.
5. The system confirms it has been stopped correctly to the user.
6. END.

3.2.2.2.4 Use Case 4: Fault injection

3.2.2.2.4.1 Scenario 1: Wireless lost connection

Goal: inject a wireless lost connection in the system.

Precondition: the system is running correctly.

Steps:

1. The user sends the ‘wireless_lost’ command to the system and the duration of the connection loss.
2. The system ends up the current monitoring session and starts a new one when the link is recovered.
3. The system configures the attenuator to lose the connection.
4. The system reports that the connection is lost to the user.
5. Wait until the connection loss is finished.
6. The system starts a new monitoring session.
7. The system configures the attenuator to allow the connection of the MCGs again.
8. The system reports that the Wi-Fi connection to the AP is available again to the user.
9. END.

3.2.2.2.4.2 Scenario 2: Packet loss start

Goal: the system starts to discards packets.

Precondition: the system is running correctly.

Steps:

1. The user sends the ‘packet_loss_start’ command to the system.
2. The system finishes the current monitoring session.
3. The system is configured to discard packets following the predefined PER.
4. The system starts a new monitoring session.
5. The system reports to the user that the packet loss has been started correctly.
6. END.

3.2.2.2.4.3 Scenario 3: Packet loss stop

Goal: the system stops discarding packets.

Precondition: the system is running correctly and the packet loss has been previously started.

Steps:

1. The user sends the ‘packet_loss_stop’ command to the system.
2. The system ends up the current monitoring session.
3. The system is configured to stop discarding packets.
4. The system starts a new monitoring session.
5. The system reports to the user that the packet loss has been stopped correctly.
6. END.

3.2.2.2.5 Use Case 5: QoS reconfiguration

Goal: reconfigure some QoS parameter (throughput, delay, jitter or PER) of the system.

Precondition: the system has been started.

Steps:

1. The user sends to the system the parameter to be reconfigured together with the new value.
2. The system ends up the current monitoring session.
3. The system reconfigures this parameter.
4. The system starts a new monitoring session
5. The system reports to the user that the parameter has been reconfigured correctly.
6. END.

3.2.2.2.6 Use Case 6: Monitoring reset

Goal: stop the current monitoring period and start a new monitoring period.

Precondition: the system has been started.

Steps:

1. The user sends ‘monitoring_reset’ command to the system.
2. The system stops the current monitoring session and starts a new one.
3. The system reports to the user that the monitoring has been reset correctly.
4. END.

3.2.2.3 Architecture Model

The architecture model represents the overall framework of the access network simulator to be implemented. As depicted in Figure 19, the access network simulator is composed of three devices:

- Signal attenuator: is able to reduce the power of the Wi-Fi signal based on the commands provided by the PC. Reducing the signal can lead to a loss of connection with the MCG.
- Wi-Fi access point: provides an access point which allows the connection of the MCGs and to transform the Wi-Fi frame into an Ethernet frame and vice versa.
- Central PC: provides the control over the latency, jitter, throughput, message errors and link failure of the communication between the MCGs and GCGs

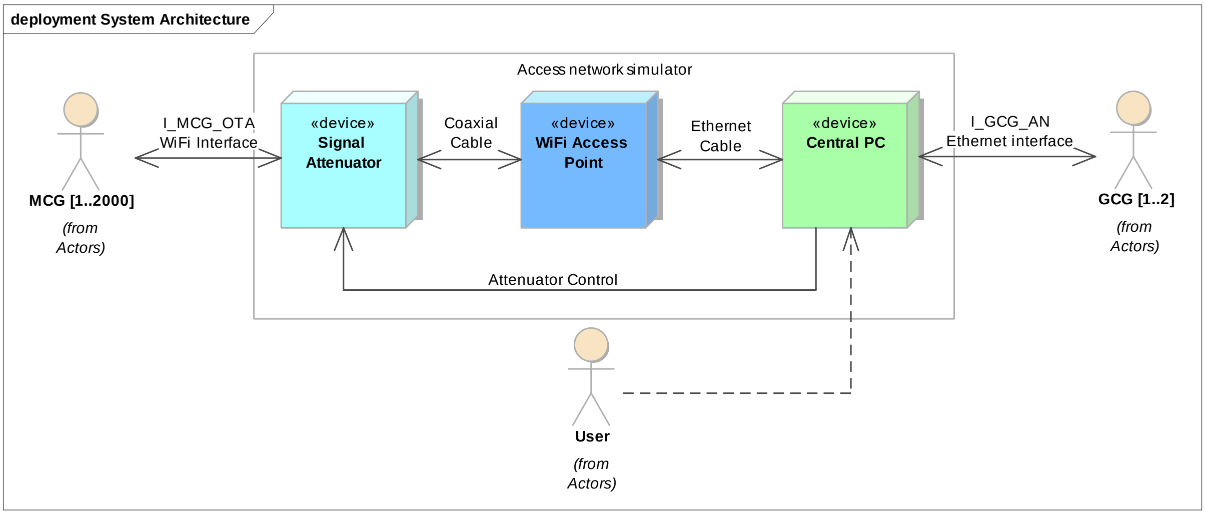


Figure 19: Architecture model of the access network simulator

The Wi-Fi Access Point can provide connections for up to 250 MCGs at the same time. However, in case the simulation needs more than 250 MCGs, two different configurations have been proposed.

The first proposed configuration introduces multiple Signal Attenuators and Wi-Fi Access Points which are connected to a central PC using switches.

The second proposed configuration is based on the use of multiple Signal Attenuators, Wi-Fi Access Points, and central PCs, which are connected to the GCGs via a switch. This means that each central PC deals with the packages of a singular Wi-Fi Access Point.

Furthermore, there are two possibilities to support two GCG. The first possibility is that the central PC is connected to a switch where the two GCG are connected. The handover from one GCG to another is done by changing the destination IP of the Ethernet frames which the central PC sends. In the second option, two network simulators are available, one for each GCG. In this option, the handover from one GCG to another could be done in a more realistic way: the signal attenuations in each network simulator can be configured to emulate that the train is approaching one AP (decreasing the attenuation) and leaving the other (increasing the attenuation).

3.2.3 Test tools

Test tools provide controlling & monitoring for the other components, so that either a user or a test automation tool can execute the tests.

Test tools are set of command line tools which interact with:

- Access network simulator
- MCG simulator
- GCG simulator
- Ground Application simulator

3.2.3.1 Use case 1: Write Access network simulation configuration

- Test tool will write configuration to the ANS and returns the result of the operation

3.2.3.2 Use case 2: Start Access network simulation

- Test tool sends a ‘start’ command to the ANS and returns the result of the operation

3.2.3.3 Use case 3: Stop Access network simulation

- Test tool sends a ‘stop’ command to the ANS and returns the result of the operation

3.2.3.4 Use case 4: Break wireless connection in simulated T2G network

- Test tool sends a ‘wireless_lost’ command to the ANS and returns the result of the operation

3.2.3.5 Use case 5: Start discarding packets in simulated T2G network

- Test tool sends a ‘packet_lost_start’ command to the ANS and returns the result of the operation

3.2.3.6 Use case 6: Stop discarding packets in simulated T2G network

- Test tool sends a ‘packet_lost_stop’ command to the ANS and returns the result of the operation

3.2.3.7 Use case 7: Write QoS configuration for simulated T2G network

- Test tool sends new QoS parameters to the ANS and returns the result of the operation

3.2.3.8 Use case 8: Reset monitoring in ANS

- Test tool sends a ‘monitoring_reset’ command to the ANS and returns the result of the operation

3.2.3.9 Use case 9: Read status data from ANS

- Test tool sends a ‘get_status’ command to the ANS and returns the result of the operation including data monitored

3.2.3.10 Use case 10: Start MCGSim

- Test tool starts an MCGSim instance with appropriate parameters and returns the result of the operation

- Several instances of MCGSim can be started

3.2.3.11 Use case 11: Terminate MCGSim

- Test tool stops a specific MCGSim instance and returns the result of the operation

3.2.3.12 Use case 12: Trigger data transfer request from MCGSim to ground

- Test tool sends a ‘data_transfer’ command with appropriate parameters to a specific MCGSim instance and returns the result of the operation

3.2.3.13 Use case 13: Trigger MCG announcement on GCG

- Test tool sends an ‘initiate_announcement’ command with appropriate parameters to a specific MCGSim instance and returns the result of the operation

3.2.3.14 Use case 14: Allow/Disallow accepting data transfer from ground to MCGSim

- Test tool sends an ‘accept_data_transfer’ command with appropriate parameters to a specific MCGSim instance and returns the result of the operation

3.2.3.15 Use case 15: Read status data from MCGSim

- Test tool sends a ‘get_status’ command to the MCGSim and returns the result of the operation including data monitored

3.2.3.16 Use case 16: Start GCGSim

- Test tool starts a GCGSim instance with appropriate parameters and returns the result of the operation
- Several instances of GCGSim can be started

3.2.3.17 Use case 17: Terminate GCGSim

- Test tool stops a specific GCGSim instance and returns the result of the operation

3.2.3.18 Use case 18: Trigger data transfer request from GCGSim to MCG

- Test tool sends a ‘data_transfer’ command with appropriate parameters to a specific GCGSim instance and returns the result of the operation

3.2.3.19 Use case 19: Allow/Disallow accepting MCG announcement on GCGSim

- Test tool sends an ‘accept_announcement’ command with appropriate parameters to a specific GCGSim instance and returns the result of the operation

3.2.3.20 Use case 20: Allow/Disallow accepting data transfer from MCG to GCGSim

- Test tool sends an ‘accept_data_transfer’ command with appropriate parameters to a specific GCGSim instance and returns the result of the operation

3.2.3.21 Use case 21: Read status data from GCGSim

- Test tool sends a ‘get_status’ command to the GCGSim and returns the result of the operation including data monitored

3.2.3.22 Use case 22: Start GAppSim

- Test tool starts a GAppSim instance with appropriate parameters and returns the result of the operation
- Several instances of GAppSim can be started

3.2.3.23 Use case 23: Terminate GAppSim

- Test tool stops a specific GAppSim instance and returns the result of the operation

3.2.3.24 Use case 24: Trigger data transfer request from GAppSim to MCG

- Test tool sends a ‘data_transfer’ command with appropriate parameters to a specific GAppSim instance and returns the result of the operation

3.2.3.25 Use case 25: Trigger data subscription on MCG

- Test tool sends a ‘data_subscription’ command with appropriate parameters to a specific GAppSim instance and returns the result of the operation

3.2.3.26 Use case 26: Allow/Disallow accepting data transfer from MCG to GAppSim

- Test tool sends an ‘accept_data_transfer’ command with appropriate parameters to a specific GAppSim instance and returns the result of the operation

3.2.3.27 Use case 27: Read status data from GAppSim

- Test tool sends a ‘get_status’ command to the GAppSim and returns the result of the operation including data monitored

3.2.4 *MCG simulator (MCGSim)*

MCG simulator is used for tests where real MCG is involved.

MCG simulator:

- Simulates behaviour of a MCG with respect to the T2G communication
- Provides a control & monitor interface for test tools

3.2.4.1 Use case 1: Start MCGSim

- MCGSim instance is started by an executable with appropriate parameters
- Several instances of MCGSim can be started

3.2.4.2 Use case 2: Terminate MCGSim

- Command ‘terminate’ will trigger process termination

3.2.4.3 Use case 3: Perform data transfer from MCGSim to ground

- Command ‘data_transfer’ will initiate data transfer to the ground application
- Data type, content and destination is specified in the parameters

3.2.4.4 Use case 4: Perform MCG announcement on GCG

- Command ‘initiate_announcement’ will initiate MCG announcement on GCG
- The announcement request content and GCG destination is specified in the parameters

3.2.4.5 Use case 5: Allow/Disallow accepting data transfer from ground to MCGSim

- Command ‘accept_data_transfer’ will allow or disallow accepting data transfers from ground application to MCGSim
- The response content to a data transfer request is specified in the parameters

3.2.4.6 Use case 6: Read status data from MCGSim

- Command ‘get_status’ will return data monitored on MCGSim

3.2.5 *GCG simulator (GCGSim)*

GCG simulator is used for tests where a real GCG is involved.

GCG simulator:

- Simulates behaviour of a GCG with respect to the T2G communication
- Provides the control & monitor interface for test tools

3.2.5.1 Use case 1: Start GCGSim

- GCGSim instance is started by an executable with appropriate parameters
- Several instances of GCGSim can be started

3.2.5.2 Use case 2: Terminate GCGSim

- Command ‘terminate’ will trigger a process termination

3.2.5.3 Use case 3: Perform data transfer from GCGSim to MCG

- Command ‘data_transfer’ will initiate a data transfer to the MCG

- Data type, content and destination is specified in the parameters

3.2.5.4 Use case 4: Allow/Disallow accepting MCG announcement on GCGSim

- Command ‘accept_announcement’ will allow or disallow MCG announcement on GCGSim
- The response content to an announcement request is specified in the parameters

3.2.5.5 Use case 5: Allow/Disallow accepting data transfer from MCG to GCGSim

- Command ‘accept_data_transfer’ will allow or disallow accepting data transfers from MCG to GCGSim
- The response content to a data transfer request is specified in the parameters

3.2.5.6 Use case 6: Read status data from GCGSim

- Command ‘get_status’ will return data monitored on GCGSim

3.2.6 Ground Application Simulator (GAppSim)

GCG Application simulator acts as an on-ground data endpoint (end device).

GCG Application simulator:

- Uses the GCG’s application interface to interact with the GCG, so that it is able to utilize T2G interface between MCG and GCG
- Provides the control & monitor interface for test tools

3.2.6.1 Use case 1: Start GAppSim

- GAppSim instance is started by an executable with appropriate parameters
- Several instances of GAppSim can be started

3.2.6.2 Use case 2: Terminate GAppSim

- Command ‘terminate’ will trigger process termination

3.2.6.3 Use case 3: Perform data transfer from GAppSim to MCG through GCG

- Command ‘data_transfer’ will initiate a data transfer to the MCG
- Data type, content and destination is specified in the parameters

3.2.6.4 Use case 4: Perform data subscription on MCG through GCG

- Command ‘data_subscription’ will initiate a subscription for defined data on the MCG
- The subscription request content and MCG destination is specified in the parameters

3.2.6.5 Use case 5: Allow/Disallow accepting data transfer from MCG to GAppSim through GCG

- Command ‘accept_data_transfer’ will allow or disallow accepting data transfers from MCG to GAppSim through GCG
- The response content to a data transfer request is specified in the parameters

3.2.6.6 Use case 6: Read status data from GAppSim

- Command ‘get_status’ will return data monitored on GC GAppSim

3.3 Interface description

3.3.1 Test tools

The test tools is a set of command line utilities. Therefore the general interface for using them is:

- Test tool accepts parameters as command line switches
- More complex parameters are read from plain text files
- Operation result is indicated by program exit code
- Additional result content (e.g. result description, details, command output) is printed on the standard output

3.3.2 MCGSim

- MCGSim provides control & monitor interface as HTTP RESTful API
- Input/output data format is JSON

3.3.2.1 Start command

Invocation:

- ‘mcgsim’ on the command line

Parameters:

- –c <configuration_file>

Output:

- Execution log printed on a configured output device

3.3.2.2 ‘terminate’ command

Invocation:

- RESTful URL: ‘/terminate’

Parameters:

- -none-

Output:

- Command execution result code
- MCGSim program is terminated

3.3.2.3 ‘data_transfer’ command

Invocation:

- RESTful URL: ‘/data_transfer’

Parameters:

- T2G request message content
- data content

Output:

- Command execution result code

3.3.2.4 ‘initiate_announcement’ command

Invocation:

- RESTful URL: ‘/initiate_announcement’

Parameters:

- Target GCG to be MCG announced to

Output:

- Command execution result code

3.3.2.5 ‘accept_data_transfer’ command

Invocation:

- RESTful URL: ‘/accept_data_transfer’

Parameters:

- Allow/disallow accepting data transfer from GCG
- T2G response message content to data transfer request

Output:

- Command execution result code

3.3.2.6 ‘get_status’ command

Invocation:

- RESTful URL: ‘/get_status’

Parameters:

- -none-

Output:

- Command execution result code
- MCGSim status data

3.3.3 GCGSim

- GCGSim provides the control & monitor interface as HTTP RESTful API
- Input/output data format is JSON

3.3.3.1 Start command

Invocation:

- ‘gcgsim’ on the command line

Parameters:

- –c <configuration_file>

Output:

- Execution log printed on a configured output device

3.3.3.2 ‘terminate’ command

Invocation:

- RESTful URL: ‘/terminate’

Parameters:

- -none-

Output:

- Command execution result code
- GCGSim program is terminated

3.3.3.3 'data_transfer' command

Invocation:

- RESTful URL: '/data_transfer'

Parameters:

- T2G request message content
- Data content

Output:

- Command execution result code

3.3.3.4 'accept_announcement' command

Invocation:

- RESTful URL: '/accept_announcement'

Parameters:

- Accept/Reject announcements

Output:

- Command execution result code

3.3.3.5 'accept_data_transfer' command

Invocation:

- RESTful URL: '/accept_data_transfer'

Parameters:

- Allow/disallow accepting data transfer from MCG
- T2G response message content to data transfer request

Output:

- Command execution result code

3.3.3.6 'get_status' command

Invocation:

- RESTful URL: '/get_status'

Parameters:

- -none-

Output:

- Command execution result code
- GCGSim status data

3.3.4 GAppSim

- GAppSim provides the control & monitor interface as HTTP RESTful API

- Input/output data format is JSON

3.3.4.1 Start command

Invocation:

- ‘gappsim’ on the command line

Parameters:

- –c <configuration_file>

Output:

- Execution log printed on a configured output device

3.3.4.2 ‘terminate’ command

Invocation:

- RESTful URL: ‘/terminate’

Parameters:

- -none-

Output:

- Command execution result code
- GCGSim program is terminated

3.3.4.3 ‘data_transfer’ command

Invocation:

- RESTful URL: ‘/data_transfer’

Parameters:

- T2G request message content
- Data content

Output:

- Command execution result code

3.3.4.4 ‘initiate_subscription’ command

Invocation:

- RESTful URL: ‘/initiate_subscription’

Parameters:

- Target data to be subscribed and destination MCG

Output:

- Command execution result code

3.3.4.5 ‘accept_data_transfer’ command

Invocation:

- RESTful URL: ‘/accept_data_transfer’

Parameters:

- Allow/disallow accepting data transfer from MCG
- T2G response message content to data transfer request

Output:

- Command execution result code

3.3.4.6 ‘get_status’ command

Invocation:

- RESTful URL: ‘/get_status’

Parameters:

- -none-

Output:

- Command execution result code
- GCGSim status data

Chapter 4 Hints for low-level design and implementation

In this chapter, hints and suggestions are listed for:

- the implementation with detailed interfaces, process' and interactions descriptions;
- an existing hardware and software components to be used in the test environment;
- an optional hardware and software equipment possibly extending the capabilities of the TE

4.1 Hardware & Software platform for Access network simulator – emulated wireless connection

4.1.1 Operating system

The used network simulator Riverbed Modeler (formally OPNET Modeler) v.18.6.0 (Release Oct 31, 2016) is able to run in both Windows and Linux operating systems (we are using Windows operating system).

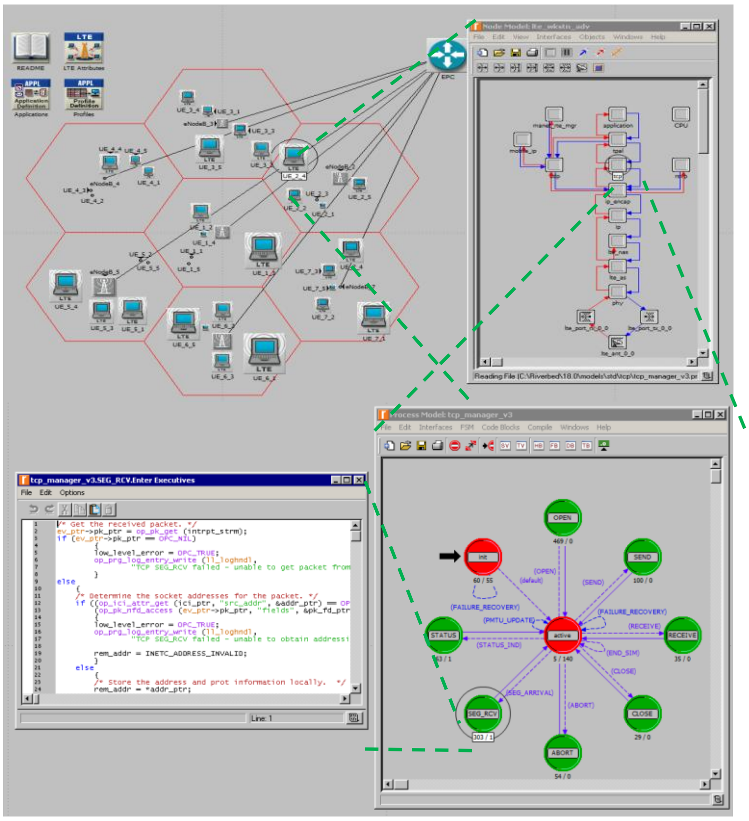
The semi-physical simulation platform is based on the Riverbed SITL (system in the loop) in Windows operating system and the Open Air Interface (OAI) LTE eNodeB [10] implementation running in Linux operating system. The software implementation of the LTE system in the OAI platform is written in standard C. The programming in the communication network simulator, Riverbed modeller, is using C language.

4.1.2 Frameworks and libraries

We choose Riverbed (formally OPNET) Modeler as the test environment for the TCMS interface to the ground (T2G) wireless systems.

Riverbed Modeler is a discrete event simulation tool used for the network modelling and performance evaluation. It supports various network types and technologies allowing to design and study communication networks, devices, protocols, and applications. This simulator is a graphical-based editor that makes it easy to use for network components structure (shown in Figure 20).

- Network level, which allows presenting the network topology. This topology consists of objects referring to the nodes and links.
- Node level, which identifies the internal structures, behaviors and capabilities of a specific device in the network. Thus, it consists of some modules of decision-making processes representing protocols, resource managers, operating systems, among others.
- Process level, which develops modules, using some states configured by an input and output process to control the functionality of the node models.
- Proto C level, which refers to the algorithm of each process.



README
LTE Attributes
APPL Application Definition
APPL Profile Definition
EPC
Node Model: lte_wkstn_adv
Process Model: tcp_manager_v3
tcp_manager_v3.SEG_RCV.Enter Executives
File Edit Options
Line: 1
OPEN
SEND
RECEIVE
CLOSE
ABORT
SEG_RCV
STATUS
init
active

```
/* Get the received packet. */
ev_ptr->pk_ptr = op_pk_get (intrpt_strm);
if (ev_ptr->pk_ptr == OPC_NIL)
        {
        low_level_error = OPC_TRUE;
        op_prg_log_entry_write (ll_loghndl,
                "TCP SEG_RCV failed - unable to get packet from
        }
else
    {
    /* Determine the socket addresses for the packet. */
    if ((op_ici_attr_get (ici_ptr, "src_addr", &addr_ptr) == OP
        (op_pk_nfd_access (ev_ptr->pk_ptr, "fields", &pk_fd_ptr
        {
        low_level_error = OPC_TRUE;
        op_prg_log_entry_write (ll_loghndl,
                "TCP SEG_RCV failed - unable to obtain addressi

        rem_addr = INETC_ADDRESS_INVALID;
        }
    else
        {
        /* Store the address and prot information locally.  */
        rem_addr = *addr_ptr;
```

Figure 20. The hierarchical modelling architecture

The current version of Riverbed Modeler supports the 3GPP standard, in which some characteristics of the LTE technology have been designed and implemented. The basic LTE architecture in Riverbed consists of three main nodes namely, UE, eNodeB and Evolve Packet Core (EPC). EPC is represented by a single node performing all the EPC-related functionalities, therefore it manages the Mobility Management Entity (MME), PDN, and Serving Gateway (SGW) functionalities. In addition, the LTE model adapts the extendable modularized approach providing high flexibility of supporting different scenarios and system configurations. The concept of the LTE in OPNET consists in forwarding a defined traffic between the UE and Server through the eNodeB and LTE core network.

The co-simulation software modules capture the real LTE traffic from an LTE base station in hardware, process the traffic in the simulation environment and forward the traffic to the application servers in the real environment. As shown in Figure 21, there are three main components in the developed framework:

- *LTE eNodeB:* the protocol stack of 3GPP standard in E-UTRAN in the OAI LTE base station;
- *Traffic capture and interpretation:* the packet translation function between the realistic and simulated environment in the Riverbed SITL gateway;
- *LTE EPC:* the basic features in the LTE core network (EPC) in Riverbed Modeler based on 3GPP Release 8 and Release 9.

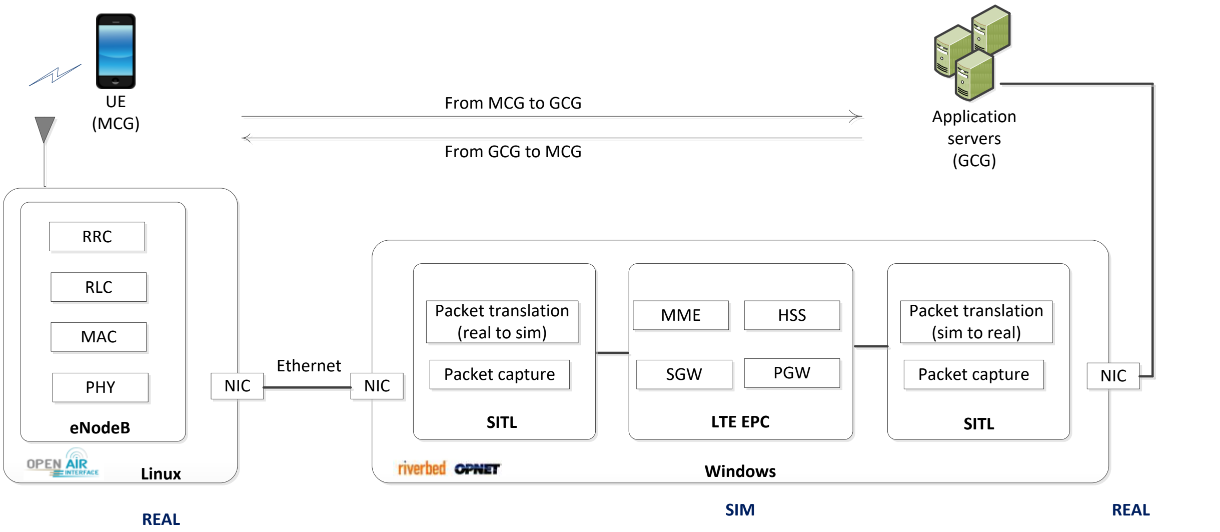


Figure 21. Co-simulation software modelling architecture

4.1.2.1 Riverbed Modules

The different Riverbed modules that we are using for this simulation framework are:

- *Simulation runtime* module
- *Wireless module: to create a radio network with the radio link and the mobile node.*
- *LTE specialized model; to create an LTE network with LTE UE node, eNodeB node and EPC nodes as well as LTE network attribute module.*
- *IPV6 module: to support the Internet Protocol version 6 in the simulation.*
- *System-In-the-Loop module: to convert data packets from the real to the simulated environment or from the simulated to the real environment.*

4.1.2.2 Real-sim-real communication via Riverbed SITL module:

The Riverbed System-in-the-loop (SITL) module provides an interface between a simulated network running in - Riverbed Modeler and a realistic test platform. With SITL, a simulation environment can connect and interact with the external physical hardware in real-time. Packets are exchanged and converted between the simulated packet and the real packet via an Ethernet link. There are three basic network configurations for Riverbed SITL simulations: real-sim, real-sim-real, and sim-real-sim. The latter two configurations require multiple gateways to connect to the different external physical ports.

In the proposed Riverbed model, an additional SITL gateway is used as the interface module, which is developed secondarily based on WinPcap. Therefore, Riverbed SITL interface module can capture data of the Ethernet card on the workstation hosting the Riverbed Modeler software. A SITL component routes the packets intended for the SITL simulation directly from the network adapter to the simulation (shown in Figure 22). At the beginning of the simulation, SITL initializes the packet translation function and registers all translation pairs. Upon receiving a packet, SITL attempts to determine the packet format and then calls the translation function to convert the packet. The packet translation process includes a set of built-in translation functions. The SITL gateway attributes module supports following functions):

- *Initialization function*: initializes the translation functions.
- *Registration functions*: register/de-register a pair of translation and test functions.

- *Test functions*: determine the packet format based on a field in the packet. These functions are paired with translation functions.
- *Translation functions from Real to Simulated*: used as a translation entry point and convert packets from real to simulated ones.
- *Translation functions from Simulated to Real*: used as a translation entry point and convert packets from simulated to real ones.
- *Payload translation functions*: convert the inner packet of the packet that it is currently converting.

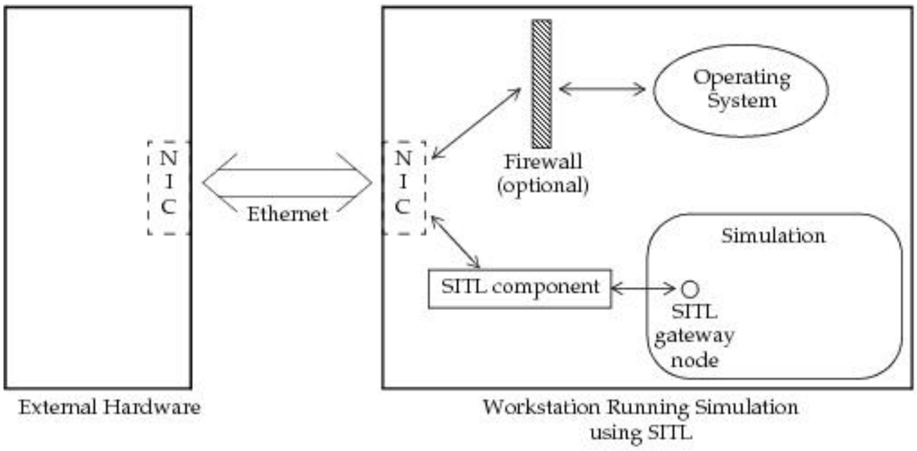


Figure 22. Packet pathway in an OPNET SITL simulation [9]

In order to run the simulation correctly and efficiently, the Riverbed Modeler and the SITL gateway node needs to be pre-configured. The real-time execution ratio property should be set, so that the simulation time in Riverbed is ensured to be consistent with the real time. The SITL gateway node is configured with the right network card. The rule for packet filtering is set appropriately in order to reduce the amount of processed data.

4.1.2.3 LTE access network communication via OpenAirInterface (OAI):

OpenAirInterface (OAI) [10] is an open-source hardware and software initiative created by the Mobile Communications Department at EURECOM. OAI is a Software Defined Radio (SDR) based solution and provides a complete software implementation of all elements of the 4G LTE system architecture. It includes the entire protocol stack from the physical to the networking layer, including standard compliant (Rel 8 and a subset of Rel-10) implementations of Physical (PHY), Medium-access Control (MAC), Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP) and Radio Resource Control (RRC), for both eNodeB and UE and in both Frequency Division Duplex (FDD) and Time Division Duplex (TDD) [10]. Regarding the core network, the standard compliant (Rel 9 and Rel 10) implementations of a subset of 3GPP LTE EPC components are supported by OAI. Figure 23 shows a schematic of the implemented LTE software in the OAI.

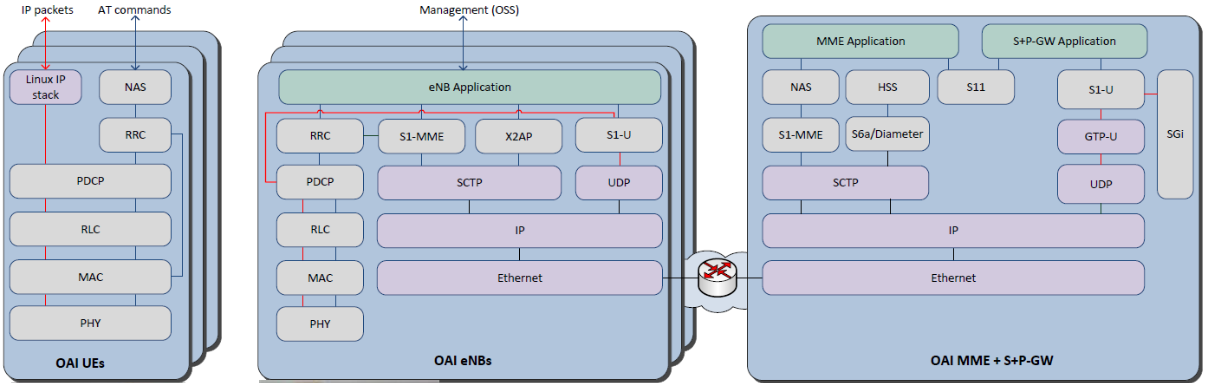


Figure 23. OpenAirInterface LTE software stack [10]

The methodology in OAI platform is to use the real stack to perform more realistic and reliable simulations. Regarding the configurations for an in-lab radio network experiment, the OAI installs the software platform on the host computer and runs the full protocol stack in the emulation mode.

The proposed eNodeB experimental testbed consists of two elements: RF hardware as the radio part and a computer as the baseband unit. The transceiver functionality is realized using a software defined radio card connected to a host computer for processing. The OAI software is written in C language and will be preferably run on a low latency Linux kernel. This project plans to use the USRP B210 SDR card from Ettus Research as the radio part. The hardware platform for the USRP-B210 based OAI eNodeB installation is a computer equipped with an Intel Xeon Processor with four cores and using Linux on Ubuntu 16.04.

The OAI version 0.6 currently implements Rel 8/10 stack for the E-UTRAN protocol stack [10] (illustrated in Figure 24).

- Implements the MAC, RLC, PDCP and RRC layers;
- protocol service for all Rel8 channels and Rel10 eMBMS (MCH, MCCH, MTCH);
- Channel-aware proportional fair scheduling;
- Fully reconfigurable protocol stack;
- Integrity check and encryption using the AES and Sonw3G algorithms;
- Supports RRC measurement with measurement gap;
- Standard S1AP and GTP-U interfaces to the Core Network;
- IPv4 and IPv6 support.

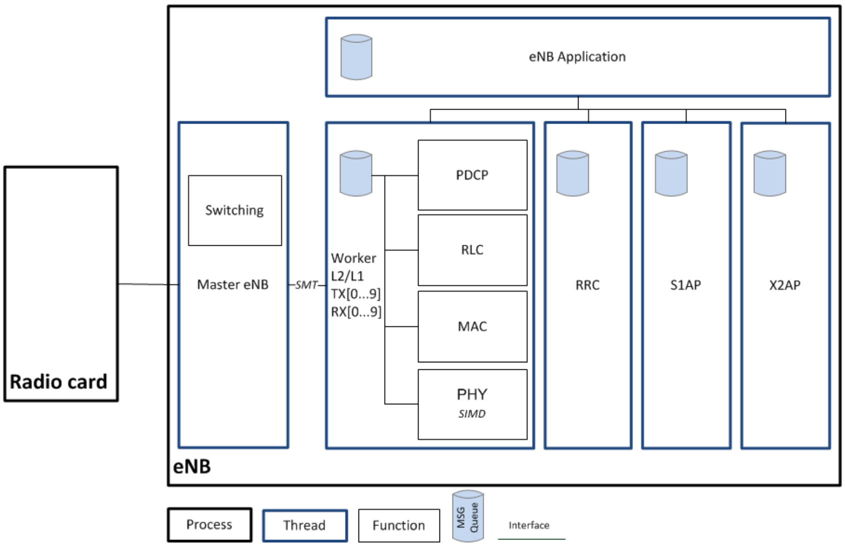


Figure 24. Software Architecture – LTE eNodeB [10]

An LTE-enabled mobile phone equipped with a test-SIM will be used as UE. Alternatively, a laptop can be used and connected via USB to the mobile phone as a modem.

4.1.2.4 Third party libraries

The software components of the co-simulation simulator are bases on a series of third party libraries:

- OAI LTE eNodeB: The access network software is freely distributed by the OAI under the terms stipulated by a new open source licence, the OAI Public License. The license caters to the intellectual property agreements used in 3GPP and allows contributions from 3GPP members holding patents on key procedures used in the standard [10].
- Real-sim-real interface:
  - Riverbed modeller (Release 14.5 or later)
  - 2 SITL licenses (independent SITL license for real-sim and sim-real interfaces)
  - WinPcap/LibPcap: each network interface card used for a SITL interface must be compatible with the Pcap library that SITL uses to capture and send packets. The WinPcap library is based on Windows, while the LibPcap is based on Linux.
- Riverbed LTE:
  - Requires a license for the Riverbed LTE module

4.2 Hardware & Software platform for Access network simulator – real Wi-Fi

The Wi-Fi access network simulator detailed in Section 3.2.2 is delved in this section. Firstly, the hardware used to build the network simulated is explained. Then, the operative system which the system supports and the required libraries are detailed. Furthermore, the different subsystems in the simulator are delved.

4.2.1 Hardware for Access network simulator

As explained in the high level design, the access networks simulator is made up of several elements:

- A signal attenuator which attenuates the Wi-Fi signal.
- A Wi-Fi access point to convert the wireless signal to Ethernet frames.
- A central PC to add delay, jitter, errors, etc. to the received frames.
    - The central PC must include at least 2 Ethernet boards.
- Switch/es to allow the system to handle more than 250 MCGs or two GCGs.

These components are shown in Figure 25, and all of them are explained below.

4.2.1.1 Signal Attenuator

A signal attenuator is a device which attenuates an electromagnetic signal without distorting it. The attenuation can be configurable to emulate a connectivity loss in the wireless link. The interface to configure the attenuation has not been defined because a specific Signal Attenuator has not been chosen yet. However, there are available devices which provide several options such as parallel communication [15], serial communication [16] or Ethernet [17].

4.2.1.2 A Wi-Fi Access Point

As a WiFi access point, a commercial one will be used. Commercial WiFi Access Points typically have a WiFi interface and several Ethernet interfaces. Furthermore, a web interface is usually available for configuration.

4.2.1.3 Central PC

The central PC runs Windows or Linux and has two Ethernet interfaces. The central PC is connected to one of the Ethernet interfaces in the Access Point, and it sniffers the Ethernet frames. It processes these frames to emulate the wireless network and, finally, sends them to the GCG.

4.2.1.4 Switches

To consider more than 250 MCGs or two GCGs, some configurations have been proposed in the high level design. These configurations demand some switches to be added as explained above. If more than 250 MCGs should be connected to the system, more than one access point has to be used. All the access points will be connected to a switch, which then is connected to the central PC. On the other hand, if two GCGs are considered, a switch is required between the central PC and the GCGs in order to send the frames to both of them.

4.2.1.5 Interfaces

Several interfaces have been defined to connect all the elements in the network emulator:

- MCG - Signal attenuator: the MCG is connected to the signal attenuator using a Wi-Fi interface; both devices are connected via a SubMiniature version a (SMA) cable.
- Signal attenuator – Wi-Fi Access Point: the signal is transmitted to the Wi-Fi Access Point through an SMA coaxial cable.
- Wi-Fi access point - central PC: an Ethernet cable connects the access point to the central PC and is used to transmit the Ethernet frames.

- Central PC - GCG: the processed frames are sent them to the GCG through an Ethernet cable.

As for the configuration of the system, some other interfaces have been defined:

- User – Central PC: the user can send commands to the central PC to configure the whole system. A socket is listening in the central PC to carry out this task.
- Central PC – Wi-Fi access point: as the central PC is the only element in the system which receives configuration commands from the user, it has to send these commands to the other elements in the network emulator. To send the commands to the Wi-Fi access point, a web interface is used.
- Central PC – Signal attenuator: the interface to configure the attenuator has not been defined because the attenuator has not been chosen yet. Several options are available in commercial RF attenuators: parallel communication [15], serial communication [16] or Ethernet [17].

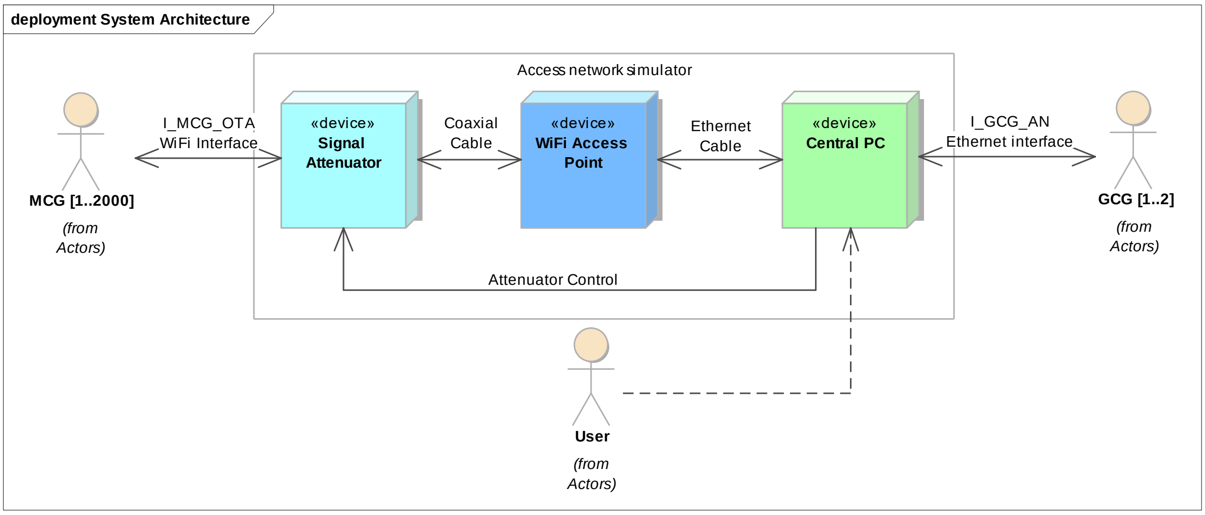


Figure 25. Architecture model of the access network simulator.

4.2.2 *Software platform for Access network simulator*

Below, the software platform for the access network simulator is explained. The operating systems which the system supports are detailed, and the different subsystems in the simulator are delved.

4.2.2.1 Operating system

The access network simulator has been designed to be able to run in both Windows and Linux operating systems. To allow this platform independence the different modules of the simulator have been design using Java programming language.

4.2.2.2 Frameworks and libraries

The Software modules that compose the management and monitoring application of the central PC of the access network simulation are:

- *Capture and Store Packet Subsystem*: is in charge of collecting the packets from the Ethernet card periodically.

- *Throughput Queuing Subsystem*: provides an output packet queue which arranges the Ethernet packets based on the arrival time, their priorities and the throughput of the system.
- *Delay, Jitter and Packet Loss Subsystem*: calculates the output time of each packet based on the provided communication delay and jitters. This module also provides the fault injection based on packet loss.
- *Monitoring Subsystem*: this module stores the information regarding the jitter, delay, throughput, and packet loss of the communication into a monitoring file.
- *Connection Subsystem*: is in charge of managing the signal attenuator in order to activate or deactivate the Wi-Fi connection.
- *User Interface*: is in charge of receiving user commands to configure and control the different subsystems, and to send user information about the result of the command execution.

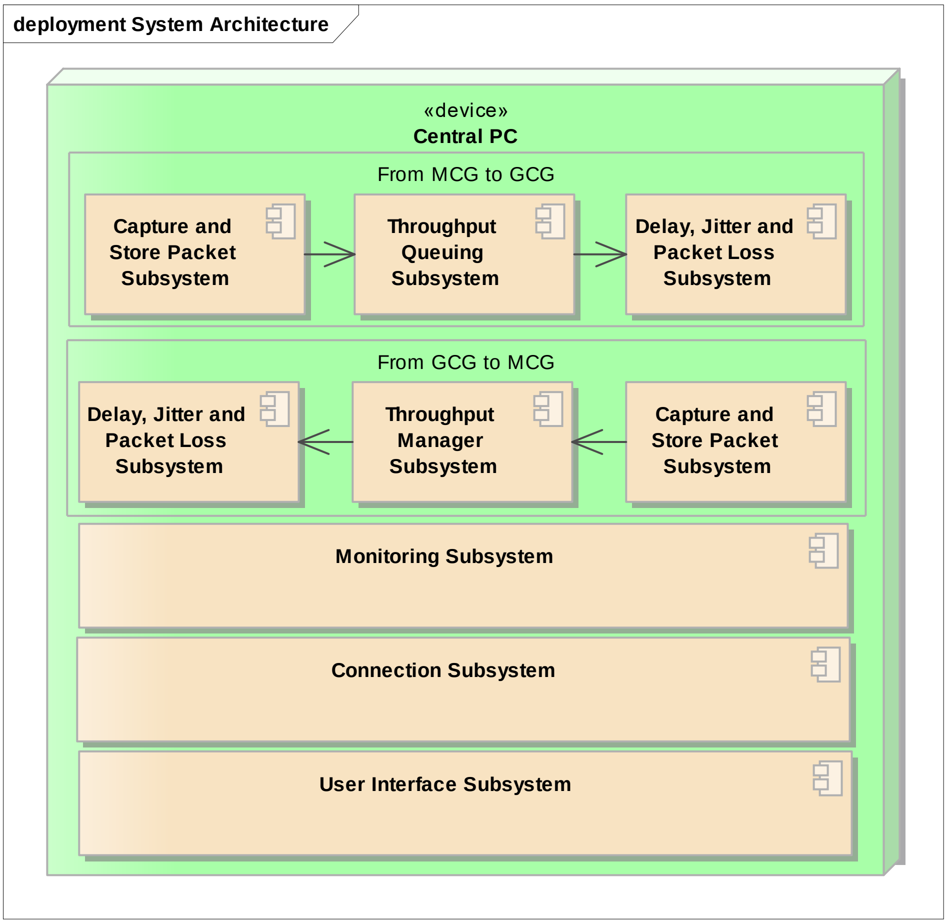

deployment System Architecture
«device»
Central PC
From MCG to GCG
Capture and Store Packet Subsystem
Throughput Queuing Subsystem
Delay, Jitter and Packet Loss Subsystem
From GCG to MCG
Delay, Jitter and Packet Loss Subsystem
Throughput Manager Subsystem
Capture and Store Packet Subsystem
Monitoring Subsystem
Connection Subsystem
User Interface Subsystem

Figure 26. Architecture model of the Central PC.

It is important to note that the communication from the MCG to the GCG and the communication from the GCG to the MCG need their own captured and stored packet, Throughput queuing and Delay, jitter and packet loss subsystems.

4.2.2.2.1 Capture and Store Packet Subsystem

The different states in the model are:

- *Not configured*: The system is waiting for an external user to configure it.
- *Configured*: The subsystem has been configured.

- *Started*: The subsystem has been started and is waiting for the recepction of a packet from the Ethernet card. The packet reception is checked periodicaly.
- *Ethernet packet extracted and stored*: The Ethernet packet has been extracted from the Ethernet card and sent to the throughput queuing subsystem to be stored.
- *Stopped*: The subsystem has been stopped.

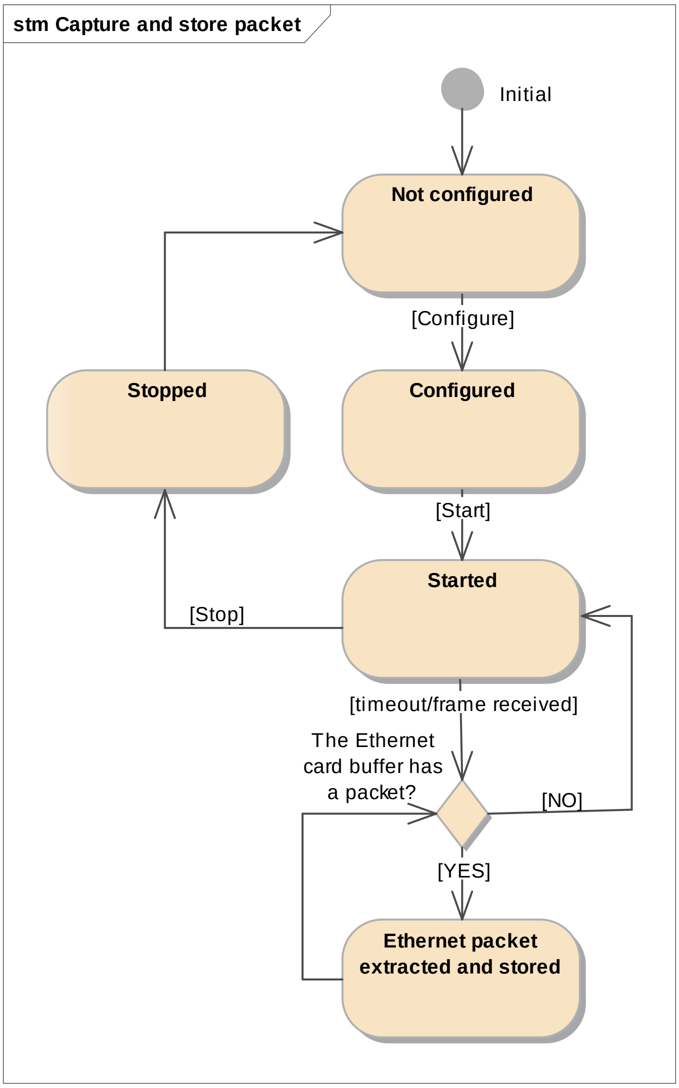

stm Capture and store packet
Initial
Not configured
[Configure]
Configured
Stopped
[Start]
Started
[Stop]
[timeout/frame received]
The Ethernet card buffer has a packet?
[NO]
[YES]
Ethernet packet extracted and stored

Figure 27. Dynamic model of the capture and store packet subsystem.

4.2.2.2.2 Throughput Queuing Subsystem

The different states in the model are:

- *Not configured*: The system is waiting for an external user to configure it.
- *Configured*: The subsystem has been configured.
- *Started*: The subsystem has been started and is waiting for an Ethernet packet.
- *Packet added*: The packet has been moved to the throughput queue.
- *Packet eliminated*: The packet has been eliminated since the throughput queue is full and the packet has a lower priority than the packets in the queue.

- *Packet with lowest priority eliminated from the queue*: The packet with the lowest priority has been eliminated from the throughput queue to make space for another packet with higher priority.
- *Stopped*: The subsystem has been stopped.

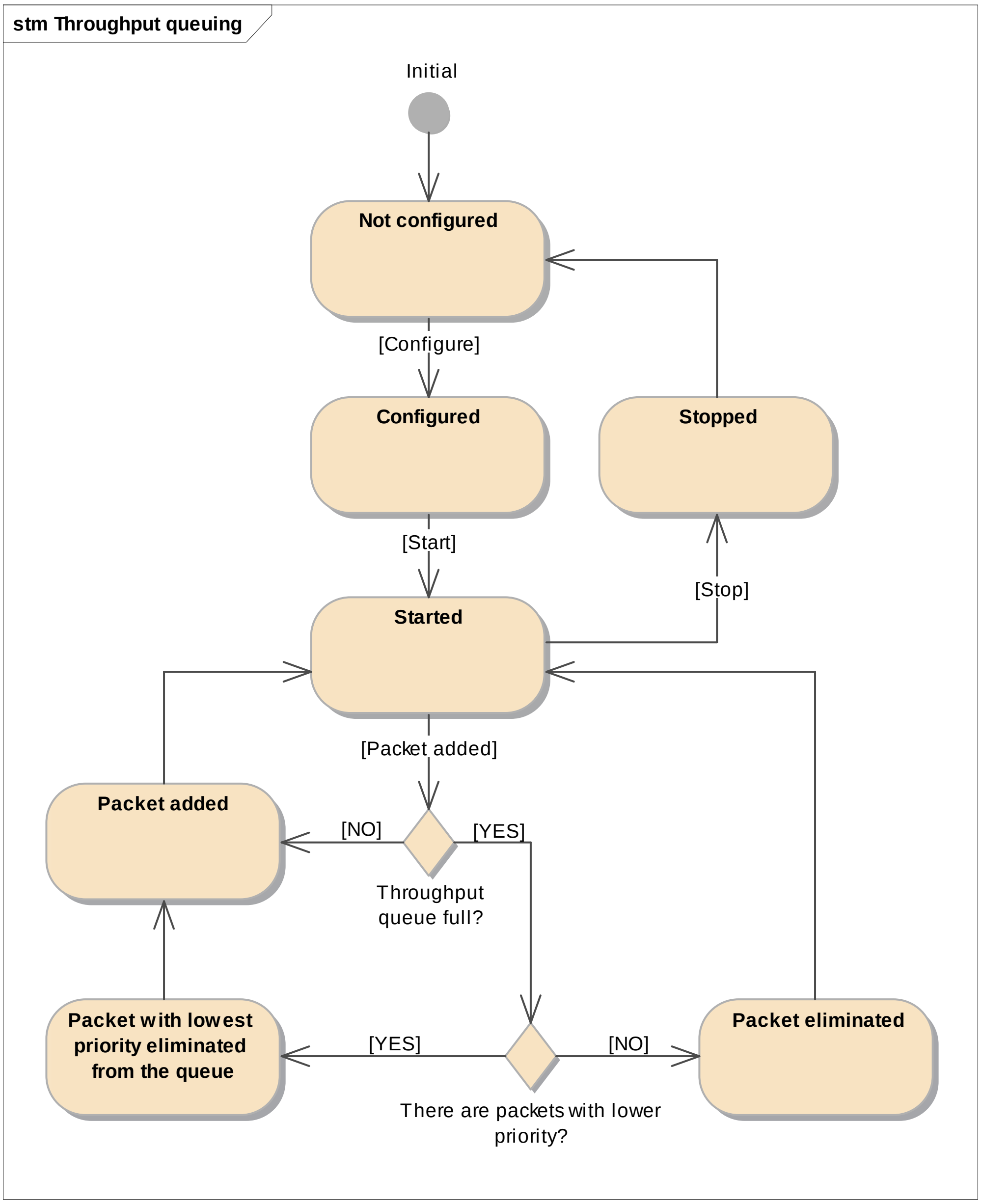


Figure 28. Dynamic model of the Throughput Queuing Subsystem.

4.2.2.2.3 Delay, Jitter and Packet Loss Subsystem

The different states in the model are:

- *Not configured*: The system is waiting for an external user to configure it.
- *Configured*: The subsystem has been configured.
- *Started*: The subsystem has been started.
- *Delay and jitter time reached*: The exit time for the packet has been reached.
- *Packet sent*: The packet has been sent.

- *Packet loss*: The subsystem has changed to packet loss mode.
- *Packet eliminated*: The packet has been eliminated instead of been sent out.
- *Stopped*: The subsystem has been stopped.

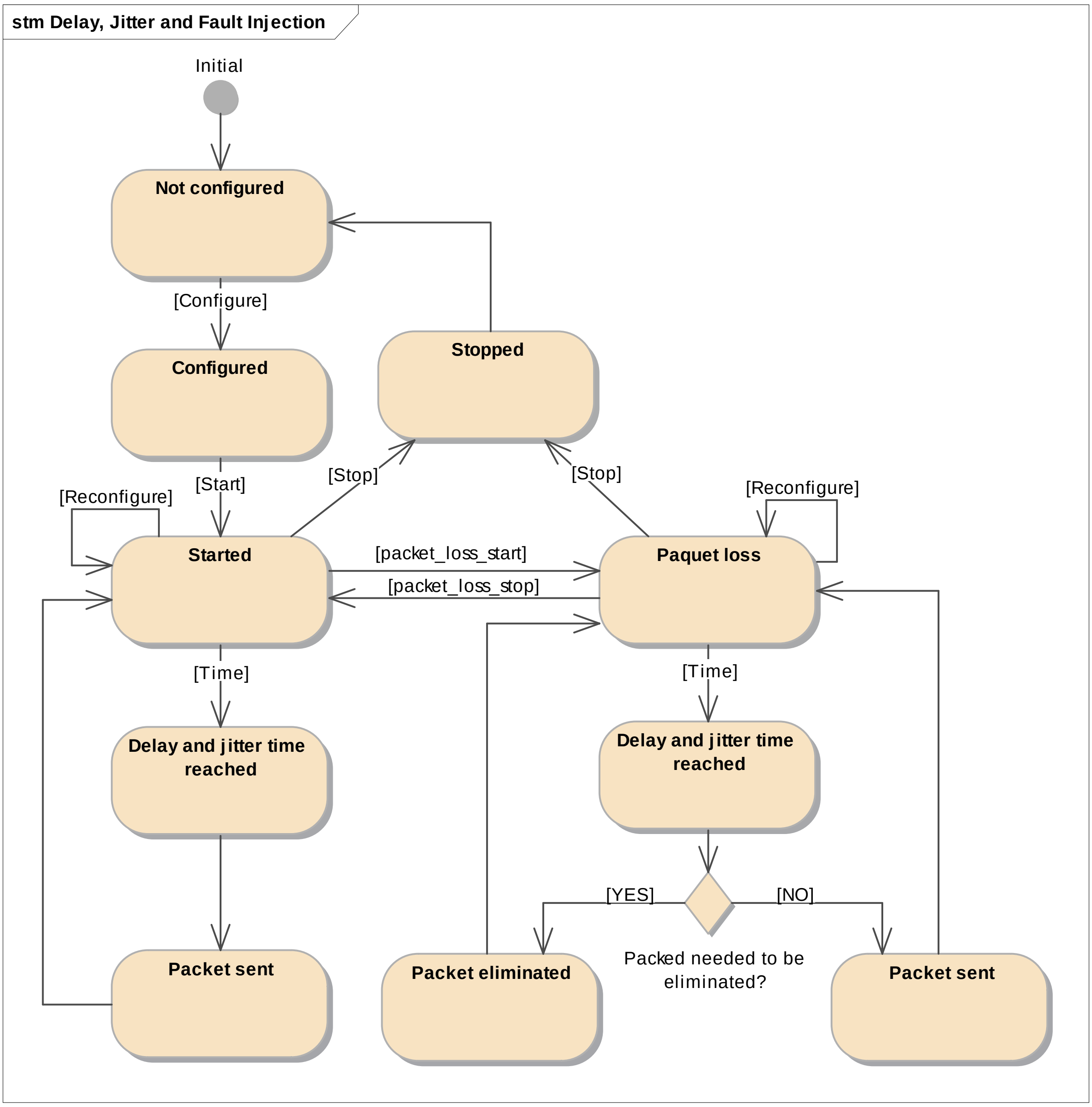


Figure 29. Dynamic model of the Delay, Jitter and Fault Injection Subsystem.

4.2.2.2.4 Monitoring Subsystem

The different states in the model are:

- *Not configured*: The system is waiting for a ‘start’ command.
- *Started*: The subsystem has been started.
- *Ethernet packet arrival time stored*: The information regarding the packet arrival time is stored into the file.
- *Monitoring data calculated*: The monitoring data referring to the delay, jitter and throughput is calculated.
- *Data referring to the packet is written into the monitoring file*: The information regarding the sent or eliminated packet has been stored.

- *Monitoring file created*: The monitoring file has been created.
- *Overall communication data written into the monitoring file*: The data referring to the delay, jitter and throughput is stored into the monitoring file

Monitoring file closed: the monitoring file has been close.

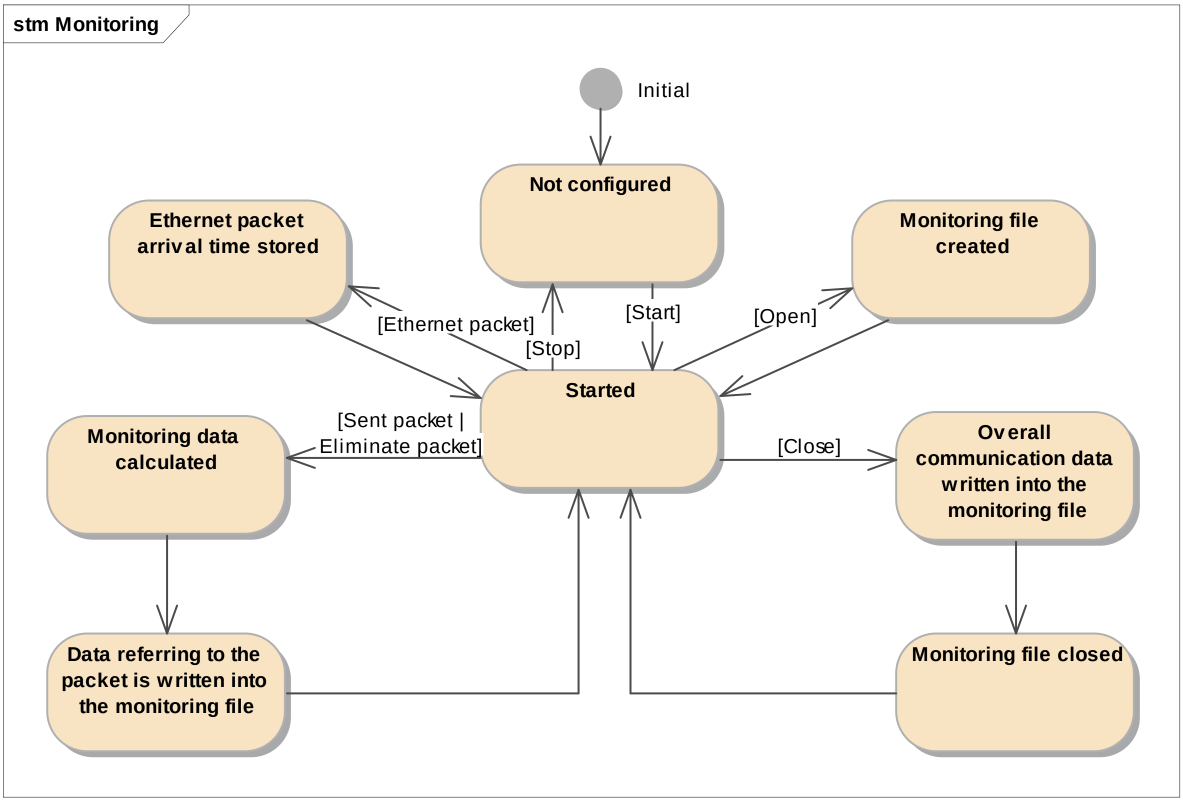

stm Monitoring
Initial
Not configured
Ethernet packet arrival time stored
Monitoring file created
[Ethernet packet]
[Start]
[Open]
[Stop]
Started
Monitoring data calculated
[Sent packet | Eliminate packet]
[Close]
Overall communication data written into the monitoring file
Data referring to the packet is written into the monitoring file
Monitoring file closed

Figure 30. Dynamic model of the Monitoring Subsystem.

4.2.2.2.5 Connection Subsystem

The different states in the model are:

- *Not configured*: The system is waiting for a 'start' command.
- *WiFi connection inhibited*: The WiFi connection has been inhibited by means of the signal attenuator.
- *Configured*: The sub-system has been configured.
- *WiFi connection activated*: The WiFi connection has been activated by means of the signal attenuator.
- *Started*: The subsystem has been started.
- *Stopped*: The subsystem has been stopped.

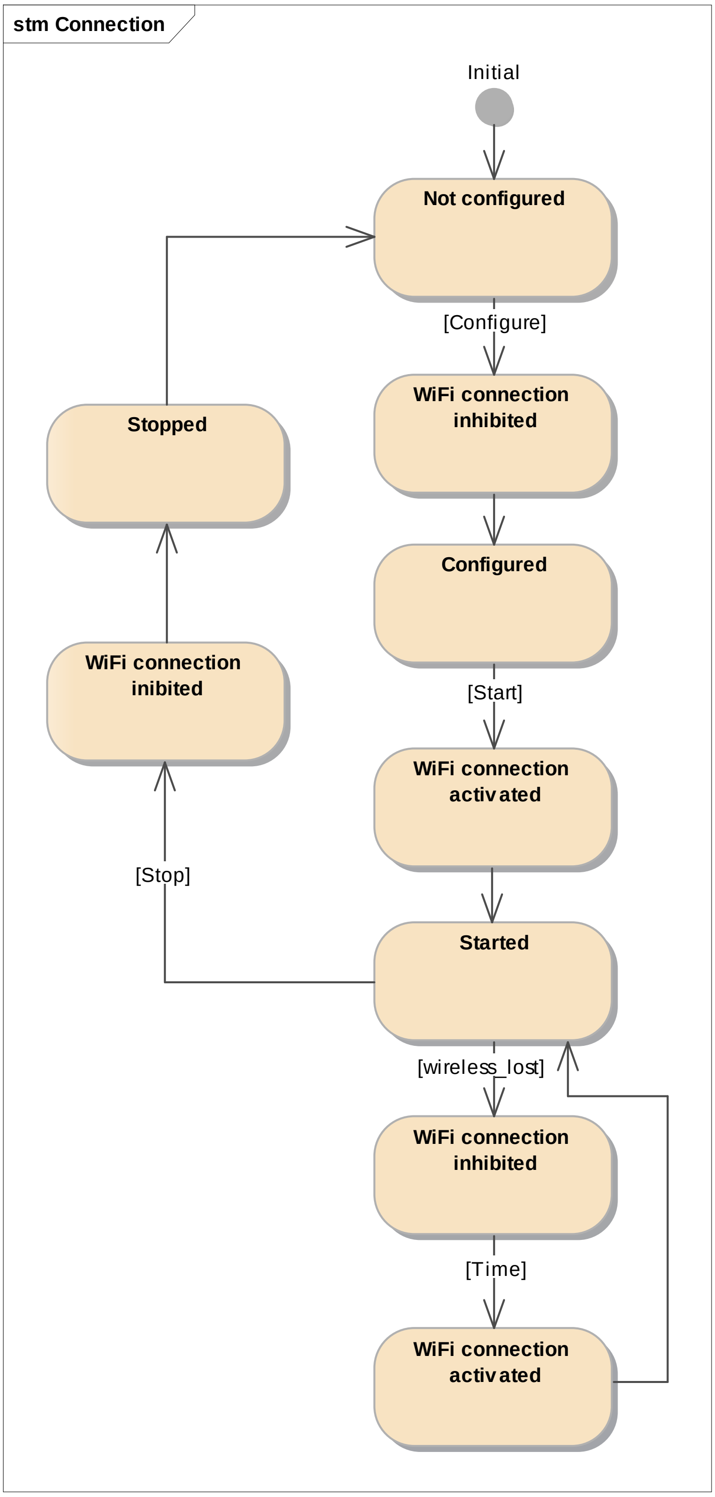


Figure 31. Dynamic model of the Fail Connection Subsystem.

4.2.2.2.6 User Interface Subsystem

The different states in the model are:

- *Not configured*: The system is waiting for an external user providing configuration files.
- *Error reported:* An error has been detected in the configuration. In this state errors can only occur because the configuration file is not correct.

- *Configure cmd received*: The ‘Configure’ command and the configuration file have been received.
- *Configured*: The configuration information has been send to the corresponding subsystems and devices. The signal attenuator has been activated preventing a connection from the MCGs.
- *Start cmd received*: The ‘Start’ command has been received.
- *Started*: The signal attenuation has been deactivated and the different subsystems have started their execution.
- *Reconfigure cmd received*: The ‘Reconfigure’ command has been received.
- *Reconfigured*: The new configuration parameters have been sent to the corresponding subsystems.
- *Wireless lost cmd received*: The ‘wireless_lost’ command has been received from the user and it has been sent to the connection subsystem.
- *Monitoring file closed*: The current monitoring file has been closed by sending a ‘Close’ command to the monitoring subsystem.
- *Monitoring file created*: A new monitoring file has been opened by sending an ‘Open’ command to the monitoring subsystem.
- *Packet loss cmd received*: The ‘packet_loss_start’ and ‘packet_loss_stop’ command has been received from the user and it has been sent to the delay, jitter and packet loss subsystem.
- *Stopped*: The different subsystems have stopped their execution.

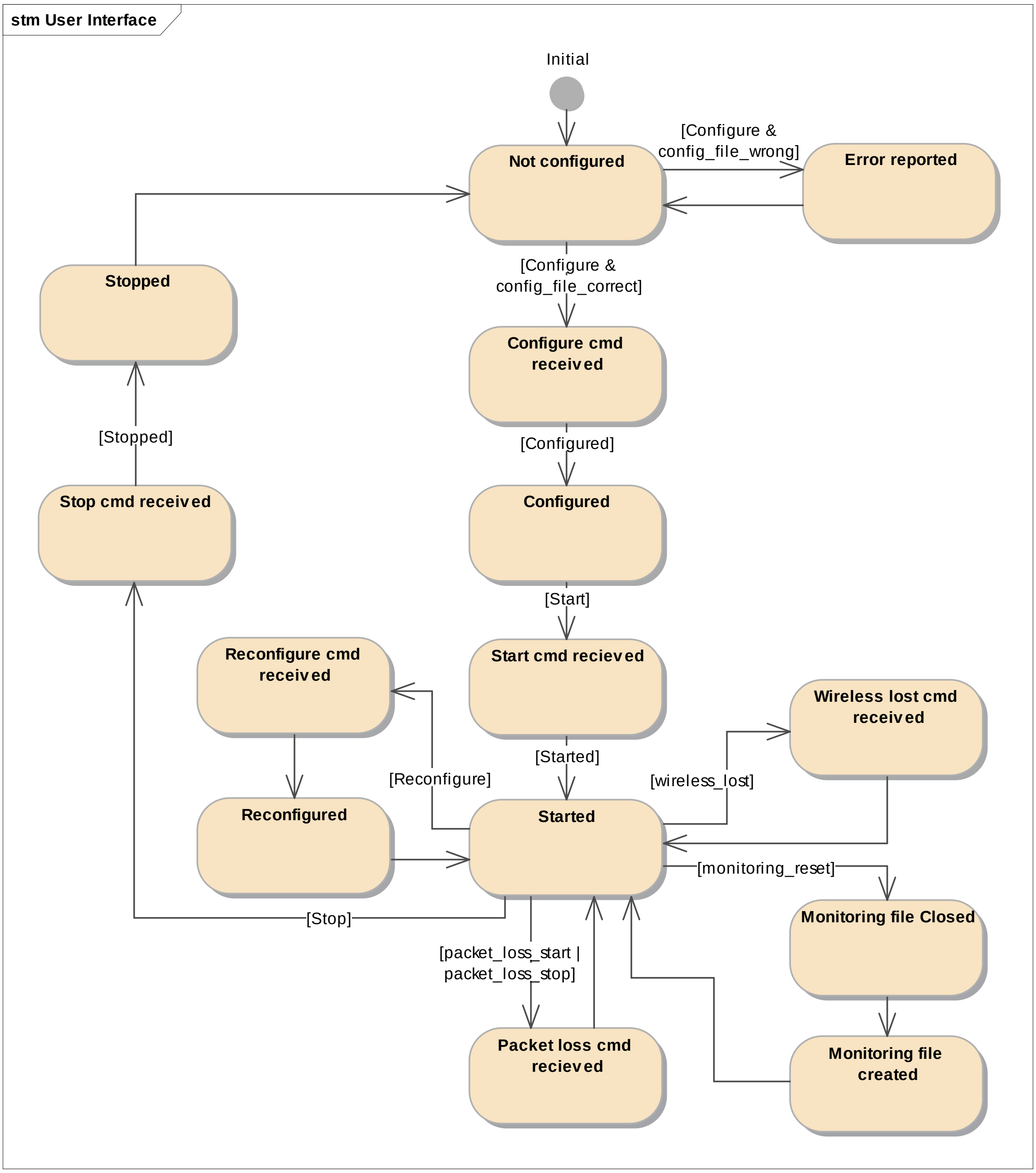


Figure 32. Dynamic model of the Configure Subsystem.

4.2.2.3 Third party libraries

The software components of the access network simulator are based on a series of third party libraries:

- WinPcap [12]/LibPcap [13]: These libraries allow the capture of Ethernet frames which are received by central PC. The WinPcap library is based on Windows, while the LibPcap is based on Linux.
- JNetPcap [14]: This library provides Java wrapper which allows the use of both the WinPcap and LibPcap libraries as part of a Java program.

4.3 Hardware & Software Platform for MCGSim, GCGSim, GAppSim

All the TE software components are glued with the HTTP RESTful API. They can be executed on the same test computer or distributed over several computers connected via Ethernet link.

The distribution over different computers may be necessary for performance test, where many multiple instances of MCGSim, GCGSim and/or GAppSim are to be executed at the same time. A machine's computation power may not be good enough to handle such high number of working processes.

It is recommended to allow executing multiple instances of MCGSim, GCGSim and GAppSim components on one computer in one operating system instance. It is expected to execute hundreds of instances of each during the performance tests. The other solution is to use operating system virtualization, but this would possibly add significant overhead.

For the test computer, a usual PC with Linux (preferably Debian distribution) or Microsoft Windows operating system is considered. Therefore MCGSim, GCGSim, GAppSim and the test tools will be implemented using cross-platform framework and tools.

Open-source tools and libraries will be used preferably.

Chapter 5 T2G TE Validation and acceptance criteria (meta-tests)

As the T2G Test Environment is a new product to be developed, its functionality and behaviour needs to be validated so that it is confirmed that the TE implementation fulfils the requirements defined and is in accordance with the design. This chapter defines such validation tests.

5.1 Access Network Simulator / emulated wireless test case

5.1.1 Test Equipment

- Network Simulator Central PC: PC running the network simulator Riverbed with two Ethernet interfaces
- OpenAirInterface system
- 1 MCG with 2 Ethernet interfaces
- 1 MCG with LTE interface
- 1 GCG with 1 Ethernet interface

5.1.2 Test Scenarios

Figure 33 shows an LTE based mobile network. In Safe4RAIL, the user equipment will be the MCG. The eNodeB is a base station providing radio coverage, managing radio resources and scheduling packets. The Evolved Packet Core (EPC) is the backbone of LTE network, which consists of Packet Data Gateway (P-GW), Serving Gateway (S-GW), Home Subscriber Register (HSS) and Mobility Management Entity (MME).

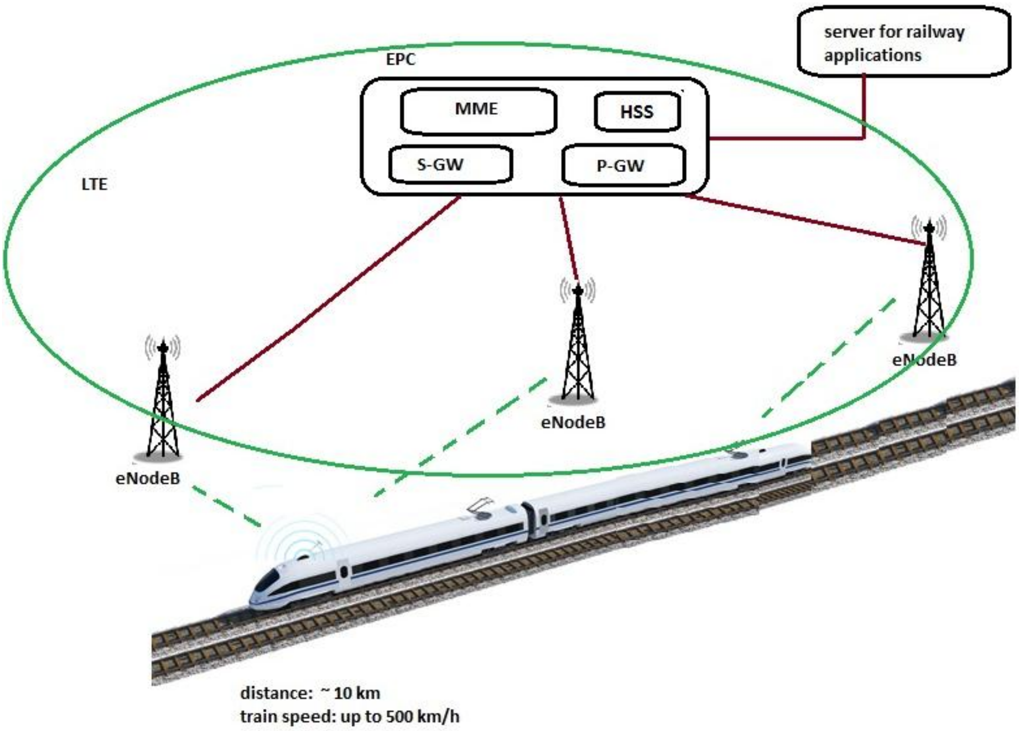


Figure 33. High speed train communication based on LTE technology

For the test of the TCMS interface to the ground wireless systems, we refer two cases:

5.1.2.1 Test Scenario 1.1: Low speed scenario cases for urban environment

In urban environments trains generally move with low and various speeds. Some parameters have an impact on the performance provided for the train connectivity:

a. Ability to use the two different communication technologies LTE and Wi-Fi, according to usual parameters such as radio conditions (e.g. Signal-to-Noise Ratio (SNR) and Bit Error Rate (BER)) and user preferences. Hence, trains have to perform the two kinds of handover: 1) A vertical handover is triggered to change the radio access technologies (RAT), when it is necessary to improve performance, 2) a horizontal handover is required to only change the attachment point (eNodeB for the LTE or AP for the Wi-Fi), keeping connected to the same RAT.

b. Use of Micro and Macro cells, according to the environment characteristics. In fact, the topology deployment can use various coverage areas provided by each eNodeB for the LTE technology. It aims to provide connectivity in the entire railway domain with the minimum number of eNodeB.

c. Use of various speeds. In the urban environment, trains are limited to use a high speed. However, they are required to modify their speeds following the travelling zones and stations. These different speeds may impact on the railway application communication, for this reason it is necessary to evaluate performance with different speeds.

d. Change of the traffic load and types. Using the LTE for the railway application allows us to introduce a more critical communicating data based on the IP level. Thus, for the purpose to fulfill the required performance and succeed the railway application, it is interesting to study the change of the traffic load and types.

Some performances are required for the railway application. It is crucial to evaluate these performances, when trains move through their defined trajectory and change their communicating technologies in the urban environment:

a. Transfer delay, refers to the necessary delay of data transmission between the train and its corresponding server.

b. Data integrity, refers to the succeeded delivering data between the two communicating entities.

c. Handover delay, refers to the required delay to perform handovers and attachment point change, for the vertical and horizontal handover.

d. Priority consideration, because of the different traffic used by the railway application. It is necessary to assign some priorities for the critical real time data.

5.1.2.2 Test Scenario 1.2: High speed scenario cases for rural environment

Capabilities and performance of the mobile communication systems affect railway operations. By improving the communication reliability and reducing the information delay, the wireless transmission system becomes higher efficient and more robust [7]:

- The communication network must support trains among the other users. Trains can travel up to 320 km/h. For a high speed train line, the rolling stocks normally run over a long distance and pass uninterrupted without frequent intermediate stops. The highest speed limit on this type of railway line varies from country to country, for example 180 km/h (e.g. the Snoghoej-Odense line in Denmark) and 320 km/h (e.g. the Lille-Paris line in France).
- Various railway services and applications delivered by the standard mobile communication network have different importance and different impact on the operation. There is a need to differentiate between these services and to provide them with various processing (e.g. routing and scheduling strategies) in the network in order to guarantee the performance requirements such as delay and throughput. The case of the high-speed communication aims to evaluate the performances of the applied railway applications with respect to the LTE network deployment, the speed of mobility, and the traffic load [7].

a. LTE network deployment

If we consider a standard LTE network for the T2G communication in Safe4RAIL, the handover could be a critical event, especially for high speed. The LTE network is not deployed specifically for the railway. Therefore, a key parameter is the network planning and deployment, for example the number of base stations used for providing the necessary radio coverage. It may be decided to deploy relatively few base stations (eNodeBs), which would transmit at a high power. This setup would cover a railway line with just a few large radio cells. Alternatively, the same railway line may be covered with more base stations, which transmit at a lower power. Thus, the coverage would be provided by many—relatively small—radio cells.

The chosen deployment strategy has an impact on the capacity, relative traffic load per cell, interference and handover frequency. Therefore, the deployment may have an impact on the performance of railway applications.

b. Train speed

According to the requirements established by the 3GPP, LTE is supposed to support user speed up to 500 km/h. Despite that, the LTE transmission in a high-speed scenario is worse compared to static or slow-speed scenarios. There are multiple factors (e.g. multipath, handover) that degrade LTE transmission performance in a high-speed environment.

For the purpose of analyzing train speed impact on railway application transmission, this high-speed test case will examine the transfer delay and data loss with varying the train speed.

c. *Traffic load*

The problem of insufficient capacity occurs in GSM-R in the areas with high-density of railway traffic, such as big train stations and junctions, due to the availability of channels. GSM-R is a circuit-switched network, where data is transmitted over virtual circuits. The drawback is that the network resources can not be assigned based on the actual demand. Therefore, LTE is expected to significantly increase the network capacity and to solve the communication bottleneck, since it is a packet-based technology.

The goal of the high-speed scenarios will analyze the impact of the traffic load on the application transmission performance in terms of delay and data integrity, also in the event of the horizontal handovers.

5.2 Access Network Simulator / real Wi-Fi test case

5.2.1 Test Equipment

- Network Simulator Access Point: Wi-Fi access point.
- Network Simulator Signal Attenuator: Wi-Fi signal attenuator
- Network Access Simulator Central PC: PC running the access network emulator subsystems.
- PC: PC containing a Wi-Fi card that allows the connection to the access point.
- Signal sniffer: Device that allows monitoring and logging the signal strength of a WiFi signal.
- Ethernet packet generator: Device (PC) which generates and transmits a predefined packet stream.
- Packet sniffer: Device which allows monitoring and logging the different packets in a network.
- Wi-Fi host: Network element that allows the connection of different wired devices to a Wi-Fi access point.
- Ethernet switches: Network element that allows the connection of multiple wired devices. This element also allows to duplicate all Ethernet traffic and send it to an specific device.
- Ethernet HUB: Network element that allows the connection of multiple wired devices.

5.2.2 Test Scenarios

5.2.2.1 Scenario 2.1

This scenario uses the following elements:

- Network access simulator access point.
- Network access simulator signal attenuator.
- Network access simulator central PC.
- PCs.

These elements are connected as presented in Figure 34.

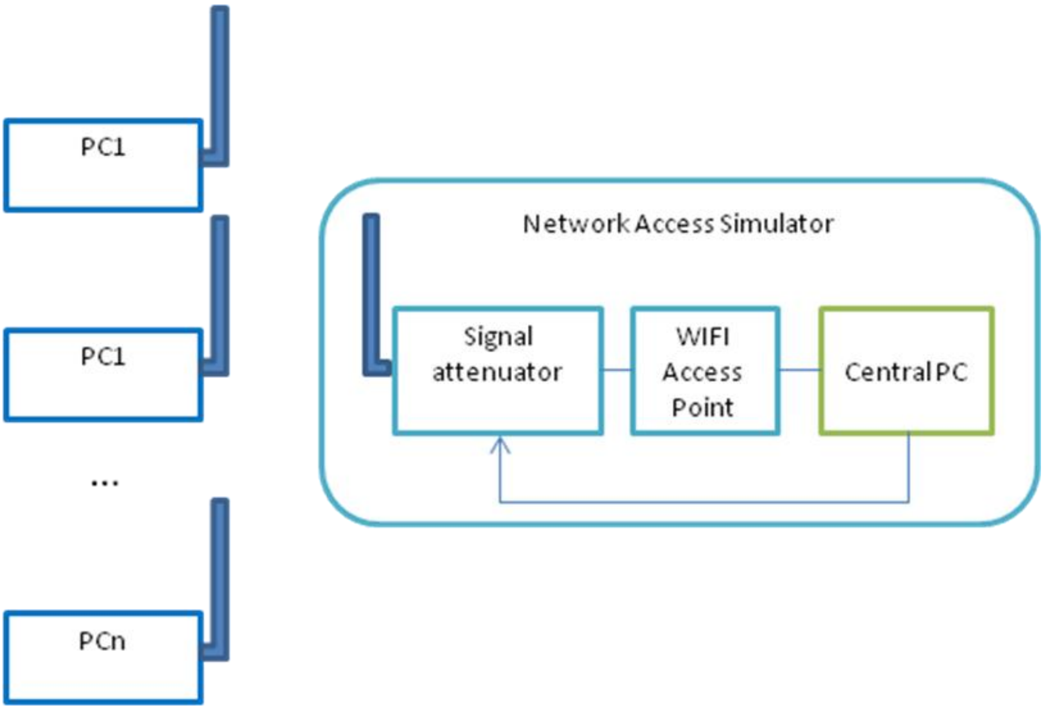


Figure 34. Scenario 2.1

5.2.2.2 Scenario 2.2

This scenario uses the following elements:

- Network access simulator access point.
- Network access simulator signal attenuator.
- Network access simulator central PC.
- Signal Sniffer.

These elements are connected as presented in Figure 35.

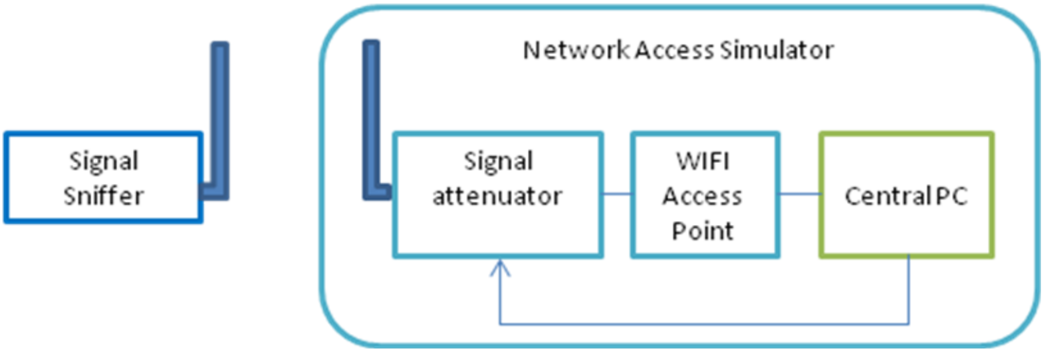


Figure 35. Scenario 2.2

5.2.2.3 Scenario 2.3

This scenario uses the following elements:

- Network access simulator access point.
- Network access simulator signal attenuator.
- Network access simulator central PC.
- Ethernet packet generator.

- Packet Sniffer.

These elements are connected as presented in Figure 36.

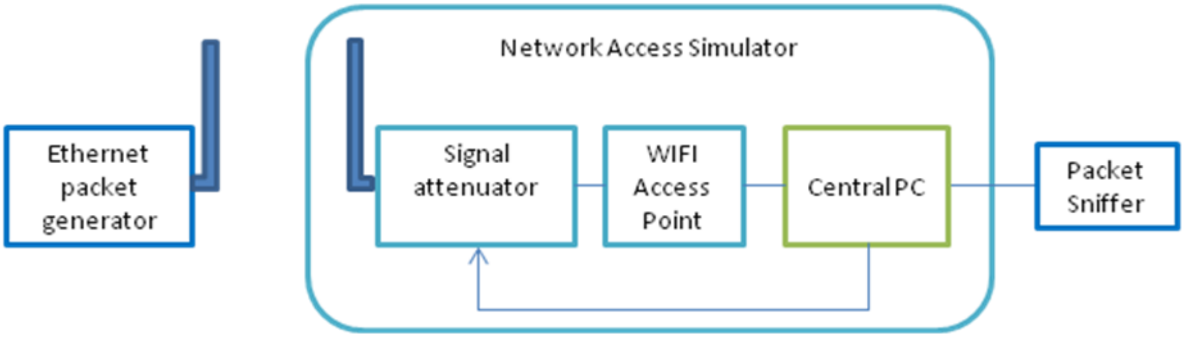


Figure 36. Scenario 2.3

5.2.2.4 Scenario 2.4

This scenario uses the following elements:

- Network access simulator access point.
- Network access simulator signal attenuator.
- Network access simulator central PC.
- Two Ethernet packet generators.
- Two packet Sniffers.
- WiFi host.
- Ethernet switches.
- Ethernet HUB.

These elements are connected as presented in Figure 37.

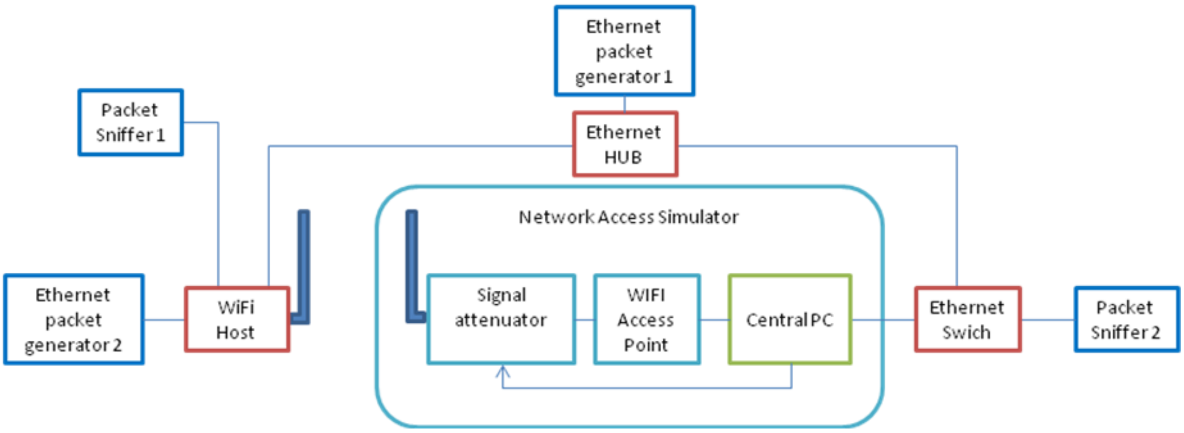


Figure 37. Scenario 2.4

5.2.3 Test Case

5.2.3.1 Test Case 2.1: Access point test

ID	T2.1
Description	This test checks the connections of multiple PCs to a WiFi Access Point
Requirements	RWP3_11_7
Scenario	Scenario 2.1
Precondition	• The Access Point is activated and configured

	• The Signal Attenuator is activated and the signal is not been modified • The Central PC is activated and configured	
Step	Step Description	Step Result
1	Activate the PCs	All PCs are started and have their WiFi card enable
2	Connect each PC to the Access Point.	All PCs are connected to the Access Point

5.2.3.2 Test Case 2.2: Wi-Fi signal interrupt test

ID	T2.2	
Description	This test interrupts the Wi-Fi signal during a set time with the attenuator and checks that the signal is lost during a set time	
Requirements	RWP3_11_6	
Scenario	Scenario 2.2	
Precondition	• The Access Point is activated and configured • The Signal Attenuator is activated and the signal is not been modified • The Central PC is activated and configured	
Step	Step Description	Step Result
1	Activate the Signal sniffer	The Signal sniffer is started and the WiFi signal of the Network Access Simulator is visible
2	Start logging the signal in the signal sniffer	The signal logging process has been started
3	Sent a 'wireless_loss' command to the central PC with a predefined time	The WiFi signal has been lost for the set amount of time
4	Stop the logging signal in the signal sniffer	A signal file has been created
5	Check in the signal file the duration of the signal loss	The duration of the signal loss logged in the file and the one configured are the same

5.2.3.3 Test Case 2.3: Throughput test

ID	T2.3	
Description	This test verifies that packets with lower priorities are lost when the throughput of the system is reduced	
Requirements	RWP3_11_6	
Scenario	Scenario 2.3	
Precondition	• The Ethernet packet generator is loaded with 100 packets with three different priorities • The Access Point is activated and configured • The Signal Attenuator is activated and the signal has not modified • The Central PC is activated and configured with a throughput that allows all packets to be transmitted	
Step	Step Description	Step Result
1	Activate the packet sniffer	The packet sniffer is started
2	Start logging the packets	The packet logging process has been started
3	Start sending packets from the Ethernet packet generator	All packets have been received by the packet sniffer
4	After all packets have been sent, stop the logging of the packets.	The test packet file has been created

5	Reduce the throughput of the Network Access Simulator	The Network Access Simulator informs the change of throughput
6	Start logging the packets	The packet logging process has been started
7	Start sending packets from the Ethernet packet generator	All packets have been received by the packet sniffer
8	After all packets have been sent, stop the logging of the packets	The test packet file has been created
9	Compare the log files before and after the reduction of the throughput, in order to determine which packets have been lost.	The packets that have been lost are the packets with a lower priority.

5.2.3.4 Test Case 2.4: Latency and jitter test

ID	T2.4	
Description	This test verifies that the Access Network Simulator can produce a specific delay and jitter	
Requirements	RWP3_11_6	
Scenario	Scenario 2.4	
Precondition	• Ethernet packet generator 1 is loaded with a predefined synchronization packet • Ethernet packet generator 2 is loaded with 100 packets • The Access Point is activated and configured • The Signal Attenuator is activated and the signal is not modified • The Central PC is activated and configured with a set of delay and jitter for the communication • The Wi-Fi host is connected to the access point on the Network Access Simulator • The Wi-Fi host and the Ethernet switches have been configured in order to allow the logging of the packets using the packet sniffers	
Step	Step Description	Step Result
1	Activate both packet sniffers	The packet sniffer is started
2	Start logging the packets on both packet sniffers	The packet logging processes have been started
3	Send the synchronization packet from the Ethernet packet generator 1	The synchronization packets have been received by both packet sniffers
4	Start sending packets from the Ethernet packet generator	All packets have been received by the packet sniffers
5	After all packets have been sent, stop the logging of the packets	The test packet files have been created
6	Compare the arrival time of the Ethernet packets collected by packet sniffer 1 and packet sniffer 2 in order to obtain the delay and jitter of the communication. The delay of the communication is the average value of the arrival time difference of each packet. The jitter is the range of	The delay and jitter calculated from the files are the same as the ones provided by configuration

	variation of the packet delays	

5.3 Test tools, MCGSim, GCGSim test case

5.3.1 *Test Equipment*

- A PCs with MCGSim, GCGSim and test tools installed
- ANS (for T3.3)

5.3.2 *Test Cases*

5.3.2.1 Test Case 3.1: MCGSim, GCGSim announcement

ID	T3.1	
Description	The test checks the announcement service provided by MCGSim, GCGSim and test tools	
Requirements	RWP3_11_15, RWP3_11_16, RWP3_11_17	
Scenario		
Precondition	• MCGSim & GCGSim is configured to interact directly with each other on the same PC	
Step	Step Description	Step Result
1	With the test tool, start one instance of MCGSim	MCGSim process is running
2	With the test tool, start one instance of GCGSim	GCGSim process is running
3	Using the test tool, invoke the ‘accept_announcement’ command on the GCGSim	Test tool returns code 0
4	Using the test tool, invoke the ‘initiate_ announcement’ command on MCGSim with the destination of the GCGSim	Test tool returns code 0

5	Using the test tool, invoke the ‘get_status’ command on GCGSim	Returned GCGSim status:
		announcement=accepted
		registered_mcgs[].size=1
		registered_mcgs[]=[MCGSim_ID]

5.3.2.2 Test Case 3.2: MCGSim, GCGSim data_transfer

ID	T3.2	
Description	The test checks the data_transfer service provided by MCGSim, GCGSim and test tools	
Requirements	RWP3_11_15, RWP3_11_16, RWP3_11_17	
Scenario		
Precondition	• MCGSim & GCGSim are configured to interact directly with each other on the same PC	
Step	Step Description	Step Result
1	Repeat T3.1	No error detected
2	Using the test tool, invoke the ‘accept_data_transfer’ command on GCGSim	Test tool returns code 0
3	Using the test tool, the invoke	Test tool returns code 0

	‘data_ transfer’ command on MCGSim with the destination of the GCGSim, data size 1MB	
4	Using the test tool, invoke the ‘get_status’ command on MCGSim and wait until MCGSim reports status tx_data_transfers_ok=1	Test tool returns code 0

5	Using test tool, invoke the ‘get_status’ command on GCGSim	Returned GCGSim status:
		rx_data_transfers_ok=1
		rx_data_transfers_err=0

6	Using the test tool, invoke the ‘accept_data_transfer’ command on MCGSim	Test tool returns code 0
7	Using the test tool, invoke the ‘data_ transfer’ command on GCGSim with the destination of the MCGSim, data size 1MB	Test tool returns code 0
8	Using the test tool, invoke the ‘get_status’ command on GCGSim and wait until the GCGSim reports status tx_data_transfers_ok=1	Test tool returns code 0

9	Using test tool, invoke ‘get_status’ command on MCGSim	Returned MCGSim status:
		rx_data_transfers_ok=1
		rx_data_transfers_err=0

5.3.2.3 Test Case 3.3: Multiple MCGSim instances & performance

ID	T3.3
Description	The test checks if the TE is able to serve many multiple MCG instances
Requirements	RWP3_11_7
Scenario	
Precondition	• MCGSim & GCGSim are configured to interact with each other through the ANS • Computer with MCGSim connected by wire to the ANS (Wi-Fi AP is not used)

Step	Step Description	Step Result
1	With the test tool, start 2000 instances of MCGSim	2000 MCGSim processes are running
2	With the test tool, start one instance of GCGSim	GCGSim process is running
3	Using the test tool, invoke the ‘accept_announcement’ command on GCGSim	Test tool returns code 0
4	Using the test tool, invoke the ‘initiate_ announcement’ command on all MCGSim with the destination of the GCGSim	Test tool returns code 0

5	Using the test tool, invoke the ‘get_status’ command on GCGSim	Returned GCGSim status:
		announcement=accepted
		registered_mcgs[].size=2000
		registered_mcgs[]=[MCGSim_ID1 to MCGSim_ID2000]

6	Wait for a delay > configured MCG heartbeat ping to GCG	

7	Repeat step 5 once (then continue with the step 8)	Same as step 5 (no lost MCG)
8	Using the test tool, invoke the ‘accept_data_transfer’ command on GCGSim	Test tool returns code 0
9	Using the test tool, invoke the ‘data_ transfer’ command on all MCGSim with the destination of the GCGSim, data size 1MB	Test tool returns code 0
10	Using the test tool, invoke the ‘get_status’ command on all MCGSim and wait until all MCGSim report status tx_data_transfers_ok=1	Test tool returns code 0

11	Repeat step 5 once (then continue with the step 12)	Same as step 5 (no lost MCG)
		Returned GCGSim status:
		rx_data_transfers_ok=2000
		rx_data_transfers_err=0

12	Using the test tool, invoke the ‘accept_data_transfer’ command on MCGSim	Test tool returns code 0
13	Using the test tool, invoke the ‘data_ transfer’ command on GCGSim with the destination of all the MCGSim, data size 1MB	Test tool returns code 0
14	Using the test tool, invoke the ‘get_status’ command on GCGSim and wait until the GCGSim reports status tx_data_transfers_ok=2000	Test tool returns code 0
15	Using the test tool, invoke the ‘get_status’ command on all MCGSim	Returned MCGSim status: rx_data_transfers_ok=1

5.4 GAppSim test case

5.4.1 Test Equipment

- A PC with MCGSim, GAppSim and test tools installed
- GCG implementing GCG-GAppSim application interface
- Ethernet switch

5.4.2 Test Cases

5.4.2.1 Test Case 4.1: GAppSim data subscription

ID	T4.1
Description	The test checks the data subscription service provided by GAppSim and test tools

Requirements	RWP3_11_30, RWP3_11_31, RWP3_11_32, RWP3_11_33, RWP3_11_34	
Scenario		
Precondition	• GAppSim and MCGSim are configured to interact with the GCG and vice versa	
Step	Step Description	Step Result
1	With the test tool, start one instance of MCGSim	MCGSim process is running
2	With the test tool, start one instance of GAppSim	GAppSim process is running
3	Using the test tool, invoke the 'initiate_ announcement' command on MCGSim with the destination of the GCG	Test tool returns code 0
4	Using the test tool, invoke the 'data_subscription' command on GAppSim for topology data	Test tool returns code 0

5	Using the test tool, invoke the 'get_status' command on MCGSim	Returned MCGSim status:
		subscribed.apps[].size=1
		subscribed.apps[]=[GAppSim_ID]

6	Using the test tool, invoke the 'push_data' command on MCGSim with topology data	Test tool returns code 0

7	Using the test tool, invoke the 'get_status' command on GAppSim	Returned GAppSim status:
		rx_data_pushed_ok=1
		rx_data_pushed_err=0
		act_topology={<from_step_6>}

5.4.2.2 Test Case 4.2: GAppSim data transfer

ID	T4.2	
Description	The test checks the data transfer service provided by GAppSim and test tools	
Requirements	RWP3_11_35, RWP3_11_37	
Scenario		
Precondition	• GAppSim and MCGSim are configured to interact with the GCG and vice versa	
Step	Step Description	Step Result
1	Repeat T4.1, steps 1-3	No error detected
2	Using the test tool, invoke the 'accept_data_transfer' command on GAppSim	Test tool returns code 0
3	Using the test tool, invoke the 'data_ transfer' command on MCGSim with the destination of the GAppSim, data size 1MB	Test tool returns code 0
4	Using the test tool, invoke the 'get_status' command on MCGSim and wait until the MCGSim reports status tx_data_transfers_ok=1	Test tool returns code 0
5	Using the test tool, invoke the 'get_status' command on	Returned GAppSim status: rx_data_transfers_ok=1

	GAppSim	rx_data_transfers_err=0
6	Using the test tool, invoke the ‘accept_data_transfer’ command on MCGSim	Test tool returns code 0
7	Using the test tool, invoke the ‘data_ transfer’ command on GAppSim with the destination of the MCGSim, data size 1MB	Test tool returns code 0
8	Using the test tool, invoke the ‘get_status’ command on GAppSim and wait until the GAppSim reports status tx_data_transfers_ok=1	Test tool returns code 0

9	Using the test tool, invoke the ‘get_status’ command on MCGSim	Returned MCGSim status:
		rx_data_transfers_ok=1
		rx_data_transfers_err=0

5.4.2.3 Test Case 4.3: Multiple GAppSim instances & performance

ID	T4.3	
Description	The test checks if the TE is able to serve many multiple instances of MCGs	
Requirements	RWP3_11_37, RWP3_11_39	
Scenario		
Precondition	• Computer with MCGSim is connected by wire to the ANS (Wi-Fi AP is not used) • GAppSim and MCGSim is configured to interact with GCG and vice versa	
Step	Step Description	Step Result
1	With the test tool, start 2000 instances of MCGSim	2000 MCGSim processes are running
2	With the test tool, start one instance of GAppSim	GCGSim process is running
3	Using the test tool, invoke the ‘initiate_ announcement’ command on all MCGSim with the destination of the GCG	Test tool returns code 0

5	Using the test tool, invoke the ‘get_status’ command on GCG	Returned GCG status:
		registered_mcgs[].size=2000
		registered_mcgs[]=[MCGSim_ID1 to MCGSim_ID2000]

6	Wait for a delay > configured MCG heartbeat ping to GCG	
7	Repeat step 5 once (then continue with the step 8)	Same as step 5 (no lost MCG)
8	Using the test tool, invoke the ‘accept_data_transfer’ command on GAppSim	Test tool returns code 0
9	Using the test tool, invoke the ‘data_ transfer’ command on all MCGSim with the destination of the GAppSim, data size 1MB	Test tool returns code 0
10	Using the test tool, invoke the ‘get_status’ command on all MCGSim and wait until all MCGSim report status	Test tool returns code 0

	tx_data_transfers_ok=1	
11	Repeat step 5 once (then continue with the step 12)	Same as step 5 (no lost MCG)

12	Using the test tool, invoke the ‘get_status’ command on GAppSim	Returned GAppSim status:
		rx_data_transfers_ok=2000
		rx_data_transfers_err=0

13	Using the test tool, invoke the ‘accept_data_transfer’ command on MCGSim	Test tool returns code 0
14	Using the test tool, invoke the ‘data_ transfer’ command on GAppSim with the destination of all the MCGSim, data size 1MB	Test tool returns code 0
15	Using the test tool, invoke the ‘get_status’ command on GAppSim and wait until the GAppSim reports status tx_data_transfers_ok=2000	Test tool returns code 0
16	Using the test tool, invoke the ‘get_status’ command on all MCGSim	Returned MCGSim status: rx_data_transfers_ok=1
17	Repeat step 5 once	Same as step 5 (no lost MCG)

Chapter 6 Requirements traceability matrix

Not all the requirements for the T2G test environment are clear and fully specified now. It is expected to be largely revised after complete T2G interface specification and T2G Test Cases specification is available from CONNECTA. Because of that, the traceability matrix which maps the T2G TE requirements to the T2G TE test cases will be provided in further documents.

Chapter 7 Conclusion and Future Work

During the requirements collection and design process, some challenges emerged due to expected and required input from the CONNECTA project.

One of the challenges is to design the test environment for the train-to-ground communication without a complete train-to-ground interface specification, and without test cases for train-to-ground interface validation. The interface specification defined in the related IEC standard [1] lacks many implementation related details. The validation test cases have not been provided by the CONNECTA project in time of designing the test environment because of the project timeline mismatch – some tasks' deliverables are not synchronized with Safe4RAIL project. It was agreed that the T2G test environment will be designed with the current knowledge, and later, during the implementation phase, the design will be revised as all of the required inputs should be available by that time. That is also the reason why the requirements traceability matrix has not been provided in this document.

During the project assignment it was decided, that Safe4RAIL Work Package 3 partner IFSTTAR will implement the TE component for in-software simulation of a mobile network link between mobile communication gateway and ground communication gateway. That will enable the MCG T2G validation tests to be executed under defined and known conditions, and so every aspect of the test will be precisely repeatable. Using a real MCG's wireless link (LTE has been considered) with dedicated laboratory cell equipment has not been feasible due to the lack of required hardware among Safe4RAIL project partners. Because of that limitation, the implemented mobile network simulation will require an MCG with an Ethernet interface used as a connection to the network simulation instead of a real wireless modem. Unfortunately the CONNECTA partners cannot provide such an MCG for the planned integration and system test. After discussions and consulting with the CONNECTA partners, the decision has been made to provide two implementations of the Access Network Simulator – one complete wireless simulation and the other using real wireless connection. The wireless simulation will serve as a proof-of-concept and a reference to be used in the future by interested parties. The real wireless connection is limited to Wi-Fi if the defined environment has to be used for T2G link. A real LTE interface can be used using commercial mobile network operator, but such tests will suffer from undefined actual quality of service of the link and very limited control.

Chapter 8 List of Abbreviations

Term	Description
ALG	Application Level Gateway (one of the 2 modes an MCG should exchange communication data between ground and on-board devices)
ANS	Access Network Simulator for T2G TE
AP	WiFi Access Point
CCTV	Closed Circuit Television
Consist	Single vehicle or a group of vehicles which are not separated during normal operation
Consist Network	Communication network interconnecting communication devices in one Consist
ED (End Device)	Unit connected to one Consist Network
GED	Ground End Device (a device on the ground using GCG's and/or MCG's services)
GAppSim	Ground Application (ground end device) simulator for the T2G TE
GCG	Ground communication gateway (ground-side component of the T2G system)
GCGSim	GCG simulator for the T2G TE
IP (Internet Protocol)	Principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its routing function enables internetworking, and essentially establishes the Internet.
LAN (Local Area Network)	Computer network that interconnects computers within a limited area such as a residence, school, laboratory, university campus or office building[1] and has its network equipment and interconnects locally managed
LTE	4G mobile communication standard (Long-Term Evolution)
MCG	Mobile communication gateway (train-side component of the T2G system)
MCGSim	MCG simulator for T2G TE
MQTT	Message Queue Telemetry Transport (ISO/IEC PRF 20922 standard publish-subscribe-based "lightweight" messaging protocol)

MVB	Multifunction Vehicle Bus
OPNET	Previous name for Riverbed Modeler. Tools for (software) modelling communication networks (trademark by OPNET Technologies, Inc.)
QoS	Quality of Service (with regard to network communication)
SITL (Software In The Loop)	A software system executed under simulated input conditions for the purpose of evaluating how well the software system functions under such input conditions.
SSL (Secure Socket Layer)	Cryptographic protocol that provides communication security over a computer network.
SSTP (Secure Socket Tunnelling Protocol)	Form of VPN tunnel that provides a mechanism to transport PPP (data link layer 2 protocol used to establish a direct connection between two nodes) traffic through an SSL/TLS channel.
SuT	System under Test
TE	Test Environment (comprises Test System and System under Test)
Test System	Test system provides test tools, simulated components, and connects System under Test
TT (Test Tool)	Tool to control the execution of tests and the comparison of actual outcomes with predicted outcomes.
TCMS	Train Control and Monitoring System
TCU	Train Control Unit
TRDP (Train Real-Time Data Protocol)	Train real-time data protocol (TRDP) used for the exchange of TCN process data and TCN message data over ETB.
UI (User Interface)	Space where interactions between humans and machines occur.
UML (Unified Modeling Language)	General-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system
VCU	Vehicle Control Unit
VPN (Virtual Private Network)	It extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network.
WiFi	Wireless local area networking technology

Chapter 9 Bibliography

[1] IEC. 2017. 9/2185/CDV : IEC 61375-2-6: ELECTRONIC RAILWAY EQUIPMENT – TRAIN COMMUNICATION NETWORK (TCN) : Part 2-6: On-board to Ground Communication. IEC, 19.8.2016.

[2] EU. CONNECTA. 2017. Technical Contribution to WP2 – WP2 T2.3 Master Test Plan. CTA- CONNECTA, 25.4.2017. T2.3-T-BTD-027-02.

[3] EU. CONNECTA. 2017. CONTRIBUTING TO SHIFT2RAIL'S NEXT GENERATION OF HIGH CAPABLE AND SAFE TCMS AND BRAKES : D2.1 – Requirements and Specification for the T2G System. CONNECTA, 9.6.2017.

[4] EU. Safe4RAIL. 2017. D3.6 Draft : Report on final requirements. Safe4RAIL, 23.6.2017.

[5] SNIADY, Aleksander et SOLER, Jose. Impact of the traffic load on performance of an alternative LTE railway communication network. In : *ITS Telecommunications (ITST), 2013 13th International Conference on*. IEEE, 2013. p. 396-401.

[6] SNIADY, Aleksander et SOLER, Jose. LTE for railways: Impact on performance of ETCS railway signaling. *IEEE Vehicular Technology Magazine*, 2014, vol. 9, no 2, p. 69-77.

[7] Aleksander Sniady, Jose Soler, and Lars Dittmann, “Communication Technologies Support to Railway Infrastructure and Operations”. Ph.D. thesis, DTU Fotonik, 2015, DOI: 10.11581/DTU:00000010

[8] KASSAB, Mohamed, BONNIN, Jean-Marie, BELGHITH, Abdelfettah. Technology integration framework for fast and low cost handovers: case study: WiFi-WiMAX network. *Journal of computer systems, networks, and communications*, 2010, vol. 2010, p. 9.

[9] Riverbed modeler, www.riverbed.com

[10] OpenAirInterface, www.openairinterface.org

[11] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. “OpenAirInterface: A Flexible Platform for 5G Research. SIGCOMM Comput. Commun” Rev. 44, 5, 33-38, October 2014.

[12] https://www.winpcap.org/

[13] http://www.tcpdump.org/

[14] http://jnetpcap.com/

[15] https://www.pasternack.com/60db-programmable-sma-female-sma-female-0-watts-attenuator-pe70a5000-p.aspx

[16] https://www.minicircuits.com/WebStore/dashboard.html?model=RUDAT-6000-110

[17] http://www.jfwindustries.com/product/50ba-011-95-sma-mini-benchtop-programmable-attenuator-assembly/