D5.2 – Simulation Based Evaluation of SAFERtec Assurance Framework

Security Assurance Framework for Networked Vehicular Technology

Abstract

SAFERtec proposes a flexible and efficient assurance framework for security and trustworthiness of Connected Vehicles and Vehicle-to-I (V2I) communications aiming at improving the cyber-physical security ecosystem of "connected vehicles" in Europe. The project will deliver innovative techniques, development methods and testing models for efficient assurance of security, safety and data privacy of ICT related to Connected Vehicles and V2I systems, with increased connectivity of automotive ICT systems, consumer electronics technologies and telematics, services and integration with 3rd party components and applications. The cornerstone of SAFERtec is to make assurance of security, safety and privacy aspects for Connected Vehicles, measurable, visible and controllable by stakeholders and thus enhancing confidence and trust in Connected Vehicles.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

DX.X & Title:	D5.2 Simulation Based Evaluation of SAFERtec Assurance Framework
Work package:	WP5 Assurance Framework Evaluation
Task:	T5.2 Simulation Based Evaluation and Extended Modules
Due Date:	M30
Dissemination Level:	PU
Deliverable Type:	R

Authoring and review process information	
EDITOR	**DATE**
Sammy HADDAD / OPPIDA	06/05/2020
CONTRIBUTORS	**DATE**
Sammy Haddad / OPP	24/04/2020
Andras Varadi / CMS	17/03/2020
Alessandro Marchetto / CRF	18/03/2020
Guillemette MASSOT / CCS	14/03/2020
Konstantinos Maliatsos / UPRC	15/04/2020
REVIEWED BY	**DATE**
Mattia Zeni / TomTom	30/04/2020
Claudio Griglione / Swarco	30/04/2020
Panagiotis Pantazopoulos / ICCS	02/05/2020
LEGAL & ETHICAL ISSUES COMMITTEE REVIEW REQUIRED?	
NO	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Document/Revision history

Version	Date	Partner	Description
V0.1	02/03/2020	OPP	Complete draft of the document structure.
V0.2	14/03/2020	CCS	Contribution to sections: Evaluation methodology and evaluation results.
V0.3	17/03/2020	CMS	Description of V2X OBU updates
V0.4	18/03/2020	CRF	Section Functional simulation
V0.5	15/04/2020	UPRC	Inputs to Section 3 and Appendix A 2
V0.6	29/04/2020	OPP	First complete version for review
V0.7	30/04/2020	TomTom	Internal review comments
V0.8	30/4/2020	Swarco	Internal review comments
V0.9	02/05/2020	ICCS	Internal review comments and edits
V1.0	06/05/2020	OPP	Final version

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table of Figures

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Acronyms and abbreviations

Abbreviation	Description
ACK	Acknowledgment
ADV	Development (CC evaluation task)
AGD	Guides (CC evaluation task)
ALC	Life-cycle (CC evaluation task)
ARP	Address Resolution Protocol
ASE	Security target Evaluation (CC evaluation task)
ASN.1	Abstract Syntax Notation One
ATE	Tests (CC evaluation task)
C2C-CC	Car2Car Communication Consortium
CA	Certificate Authority
CAM	Cooperative Awareness Message
CC	Common Criteria
CRL	Certificate Revocation List
CSI	Channel State Information
CTL	Certificate Trust List
CU	Communication Unit
CVE	Common Vulnerabilities and Exposures
CVS	Connected Vehicle System
D	Deliverable
DENM	Decentralized Environment Notification Message
DoS	Denial of Service
ETSI	European Telecommunications Standards Institute
GUI	Graphical User Interface
HMI	Human Machine Interfaced
HSM	Hardware Security Module
ITS	Intelligent Transportation Systems

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

ITS-S	ITS Station
IVN	In Vehicle Network
LDM	Local Dynamic Map
LLC	Logical Link Control
LVI	Local Vehicle Information
MAC	Medium Access Control
MAP	Intersection Map
MPDU	MAC Protocol Data Unit
NAV	Network allocation vector
NOK	Not OK
OBD2	On-board Diagnostics II
OBU	On Board Unit
OSI	Open Systems Interconnection
PKI	Public Key Infrastructure
PP	Protection Profile
QoS	Quality of Service
RSU	Road Side Unit
SAF	SAFERtec Assurance Framework
SFR	Security Functional Requirement
SPaT	Signal Phase and Timing
T	Task
TOE	Target of Evaluation
TSF	TOE Security Function
TSFI	TOE Security Function Interface
V2X	Vehicle to everything
V-ITS-S	Vehicle – ITS Station
WP	Work Package

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

2G GSM	Second Generation Global System for Mobile communications

Table 1: List of Abbreviations

Executive Summary

The aim of T5.2 task is to evaluate the SAF framework.

We have already identified in the D5.1 'Comparative Analysis' deliverable different parameters to drive the comparison of existing evaluation frameworks. Those parameters are used for theoretical comparison with other approaches we have identified in the state of the art. In this deliverable we further try to evaluate the SAF efficiency relying on the complete WP4 implementation referred to as the 'Connected Vehicle System' (CVS). The idea is to get a more concrete feedback of its efficiency and the benefits it provides in terms of security on real implementations; on the other hand, a SAFERtec-developed ITS/V2X network system-level simulator allows us to execute detailed testing on selected security controls of the CVS and gain further insights.

This challenge is not trivial and requires from us to identify how to provide further security evidences different from the ones already produced for D3.3 and D5.1. In the D3.3 'Assurance Framework Testing and Refinement', we have already tested and gained feedback on the security assurance that SAF provides through identified and fixed flaws, security objectives reviews, etc. It has also been theoretically evaluated in the D5.1 'Comparative Analysis' deliverable through comparison of its main characteristic's with other cyber-security evaluation approaches. Here we mainly aim at providing extra and different feedback to assess SAF benefits in a close to real environment with online attacks mimicking real threats. With the system-level simulator we aim to evaluate the efficiency of employed controls and thus, complement the real environment consideration.

In fact, if we have already evaluated different security properties of the ITS components developed in the context of the WP4 (AppOBU - docker application, HSM and V2X OBU - Communication Protocol Control), these evaluations have been made in a specific context, under specific conditions. The context is the one defined by the STs which determine and limit the threats to be taken into account and suggest the evaluations being made under laboratory conditions. This means offline tests, i.e. not a full operational system (only the mandatory environment component) and grey box conditions (accessed to privileged accounts, partial knowledge of TOEs internal architecture, etc.). Which does not reflect what real attackers faces.

In order to assess the security provided in more real context, we propose to simulate attacks in a more realistic way. The implementation of this context is twofold:

- simulation of complete use cases
- and black box vulnerability testing

This task will provide additional test cases and reveal its sensitivity to capture the assurance levels with respect to the simulated conditions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1 Introduction

This deliverable presents simulation of attacks mimicking real operational threats in order to evaluate their impact on the system and the resilience provided by the SAF framework; those are still exploited by having the SAFERtec-developed simulator (presented in the Appendices) to derive module-level testing of interest (see D3.3)

For the sake of coherence, the description of a complete system-level simulator instance is also provided as an Annex. The simulator helps the testing, validation and configuration of CVS-employed security controls and has been used to this end with the dedicated test-cases and results being presented in D3.3.

In this deliverable, we first present in section 2 the detailed methodology used to run the tests and we justify how the results provided by this approach will allow us to evaluate concretely the SAF efficiency. To do so, we have chosen to compare the results of two penetration tests campaigns: one before and one after running the SAF; in a close to real environment. On a parallel testing activity and for the sake of coherence, we briefly describe the ITS/V2X network system-level SAFERtec simulator and its intended usage for testing individual CVS security controls. The results are relevant-to and placed in D3.3.

Then, we present the general concepts of the simulation framework used for the tests. It is composed of the first version of the 'Connected Vehicle System' (VCS) use cases produced in the WP4 (section 2.1) without the security module extension that will be tested in D5.3, as well as the attacks used for the simulation (section 2.2).

Section 3, presents more precise technical details of the platform and the tests: exact configuration, tools, etc. Together with the summary of the evaluation results. In fact, full evaluation details are provided in the vulnerability tests report provided as a confidential annex to the partners involved in the developments.

Finally, section 4 presents the analysis of the evaluation results. It tries to identify:

- where vulnerabilities tests failed (no vulnerability found), how SAF impacted those observations
- when a vulnerability is found identify if and why SAF did not manage to provide the proper assurance. These results will also be used to discuss the evaluation of the SAF parameters presented in D5.1.

1.1 Purpose of the Document

The idea here is to test on simulated ITS use cases the impact of cyber-attacks, after the system and its components have been evaluated and validated by the SAF approach (task T3.3). Those tests will be run on a complete system running the use cases defined in WP2 and developed in WP4. It will provide the attacker's point of view by executing black box testing, i.e. the attacker (the tester) will only have access to the system external interfaces and will not use any further information than public information.

Compared to D3.3 the vulnerability tests produced for this task will not limit the attacker capabilities. The T3.3 evaluation limited the level of an attacker's actions to the ones associated to AVA_VAN.2 and defined by the CC as 'Basic attack potential' (i.e., the introduced SAF1 assurance level is used to evaluate system components includes AVA_VAN.2).

The goal is to evaluate how many vulnerabilities can be exploited in the system by a real attacker after having applied the SAF and then compare them to the risk analysis performed in the deliverable D2.3 and the threats selected in the SAFERtec STs. These results will be used, to confirm if SAF and its different components manage to provide suitable assurance. We will assess both the evaluation tasks relevance and the data/tools provided to enhance the developer's inputs (mainly the protection profile).

1.2 Intended readership

Besides the project reviewers, this deliverable is addressed to any interested reader (i.e., PU dissemination level).

1.3 Inputs from other projects

This deliverable does not use any inputs from other projects.

1.4 Relationship with other SAFERtec deliverables

This deliverable will present tests made on the 'Connected Vehicle System' (CVS) specified and developed in the WP4 and presented in the deliverables D4.2 'Modules and Applications of Connected Vehicle'. These tests results will be compared to tests done in the D3.3 'Results of SAFERtec Assurance Framework Testing**'.** They will be used to discuss the validity of the security targets used for those tests.

The deliverable D5.3 'Extended Modules of the Connected Vehicle System' will further compared those results with the one obtained on extend modules. Finally, the system-level simulator from which we obtain some of the D3.3 results, is described in the Annex of the present deliverable.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

2 The SAF Evaluation methodology

The goal of this deliverable is to evaluate the SAF. SAF is an assurance framework that aims to provide confidence in the security properties of the ITS systems. SAF should both:

- (i) help to identify the proper security counter-measures to be implemented thanks to the development of a new and innovative risk analysis method adapted to ITS systems (presented in D2.2)
- and (ii) be able to provide efficiently high level of confidence (assurance) that those security counter-measures are in fact correctly implemented.

SAF has been executed partially in the WP3 i.e., a set of diverse yet not-exhaustive evaluation activities have been executed (justifications for the encountered evaluation limitation are presented in D3.3). Thanks to this execution we managed to identify and correct different security problems. Some corrections are still processed. Here we will assess the impact of this evaluation at a larger scale and in conditions close to the real environment using simulation.

To do so, we have chosen to compare the results of two penetration tests campaigns: one before and one after running the SAF; in a close to real environment. We will show the differences and identify how security issues have been fixed thanks to the evaluations done in the D3.3.

The penetration tests have been executed with the following objectives in mind:

- Take control of the principal assets of the test bench;
- Alter the expected behaviour of the existing use cases;
- Access to sensitive personal information.

The main goals in using penetration testing on a simulated system are to:

- Provide a form of security audit
- Assess the risks of intrusion
- Run actual tests instead of a review process
- Adopt the point of view of a real attacker (the “black-box” approach)
- Carry-out a relevant evaluation of impact and exploitability in the real system

Thus, this approach compared to the SAF evaluation should provide different results (a different point of view in the security assessment of the system) with the specific constraints to use less time and resources.

In order to provide the most relevant results, the attack simulation aspects in this task are twofold:

- Use a functional simulation of a real system as the test environment
    - Trying to simulate real ITS use cases and real vehicle behaviour including impact of attacks on those behaviour
- Simulate real attacks
    - Trying to play attacks in conditions clause to reality for the attacker

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

- Use the SAFERtec system-level simulator (see A 2: The SAFERtec System-level Simulator) to simulate attacks and verify the configuration and the effectiveness of security controls

We present those two simulation concepts in more details in the following two subsections (2.1 and 2.2, respectively).

2.1 Functional simulation

As input for the functional simulation a bench prototype of connected vehicle eco-systems has been provided. This bench has been developed in WP4 and is presented in full details in the WP4 deliverables. We present here a summary of the ITS functions and components deployed in the system that allow us to simulate a real system used to realize our attacks. Figure 1 shows the logical architecture of the bench eco-system prototype and a picture of the actual implementation, in the figure's right corner. The bench implements some execution scenarios related to the following use-cases:

- The Optimal Driving Speed Advice via V2X.
- Provision of Real-Time Traffic-hazard information via V2X.
- Priority request in intersection crossing via V2X
- The Optimal Driving Speed Advice via Cellular links
- Provision of Real-Time Traffic-hazard information via V2X and Cellular links
- Management of sensitive user information

Additionally, some documentation about the bench hardware and software composition, and about the executed use-case scenarios has been provided (D4.2 and D4.3) aiming at describing the bench composition, set-up and behaviours in different situations and contexts. The bench represents a full CVS prototype testbed that enables the realization of a broad set of timely Vehicle-to-Infrastructure (V2I) use-cases and serve as the basis for automotive cybersecurity testing.

Using the bench as a prototype testbed is a good solution for testing and validating communication-based vehicular technology considering that full computer-based simulation is complicated and expensive. Indeed, it requires detailed system models while field-testing requires physical resources authorization hard to provide and focusing only on a few components to be tested, knowing it may raise safety concerns.

The bench we have used for our attack simulation is composed of the following main components: vehicle, road-side unit (R-ITS-S), cloud services; and their interconnections.

- The vehicle is a mobile ITS station equipped with communication technologies: (i) ETSI ITS-G5, short-range (up to 1km) V2I wireless connectivity based on the IEEE 802.11p; (ii) 3G/4G/LTE cellular, the long-range mobile connectivity for the vehicle-to-cloud (V2C) link; and (iii) in-vehicle Ethernet/CAN/Wi-Fi, mainly for board-to-device link.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The Roadside ITS Station (R-ITS-S) is the fixed infrastructure, installed close to the road and connected to the vehicle via ETSI ITS-G5[1] and to the cloud via wired connectivity. The R-ITS-S acts as a gateway between the mobile ITS station (vehicle) and the services provided by the Central ITS station (C-ITS-S, cloud service). Hence, it implements the ETSI ITS-G5 stack and it uses a private IP network over a wired connection to transfer and receive notifications and data to/from the cloud.
- The C-ITS-S (cloud) service is connected to: (i) other services, such as traffic management centre (TMC), traffic light controllers (TLC), and traffic information providers; and (ii) R-ITS-S, thus reaching vehicles. Other used cloud services provide enhanced functionality such as real-time traffic information, road-events notification, traffic management and user authentication. The bench uses both mock-up cloud testing version such services and also real-world version, e.g., of the traffic traffic-info service that provides information about real-time traffic events for a given geographic area, e.g., traffic jam information or presence of road-works.

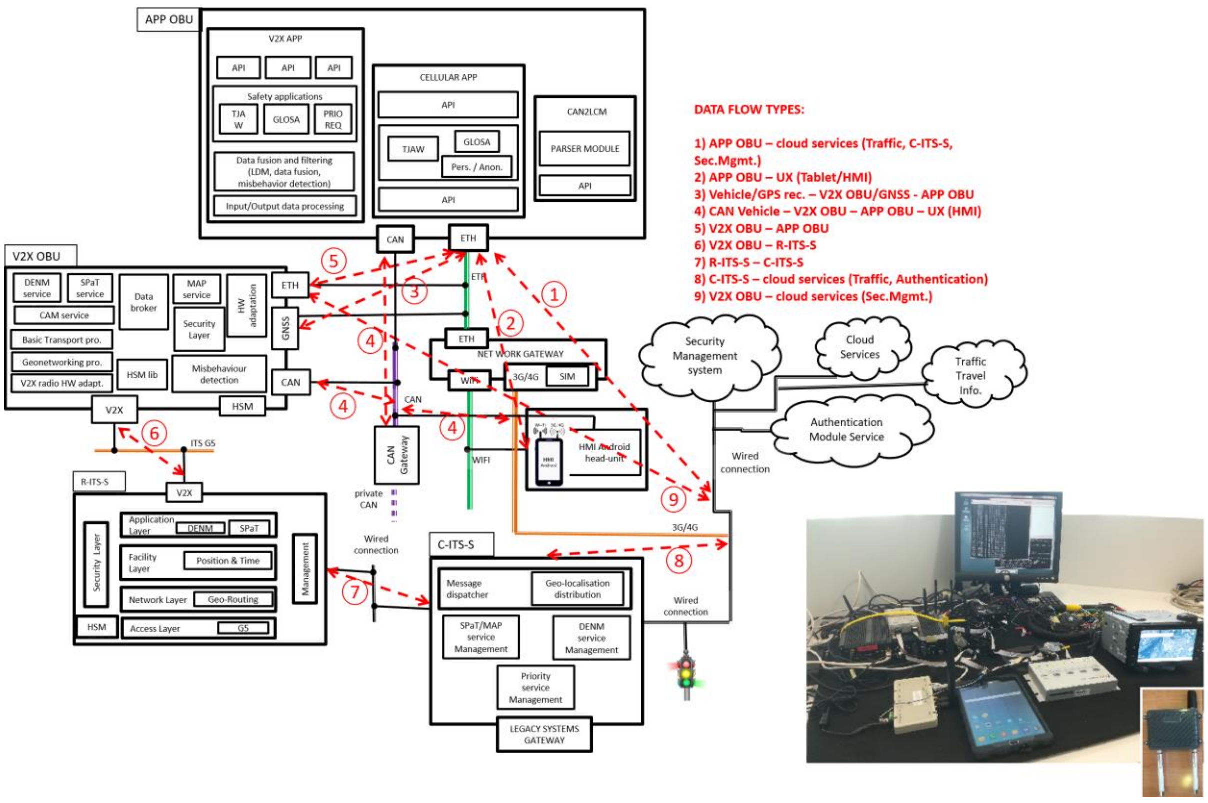


Figure 1: Architecture of the bench and picture of the prototype (right-bottom)

[1] standard ETSI EN 103 097 121/131

In the SAFERtec project, the bench prototype has been used in a widely extensive functional validation, in which the main bench components, functionality and the use-case reproduction has been widely tested by the project partners. This testing activity and the achieved output have been documented in the deliverable in D4.3 "Integration of Connected Vehicle System"). In the SAFERtec bench prototype, the main components under functional test have been the following one:

- The in-vehicle CAN network based crossed by vehicle signals, such as speed, brake, and vehicle body signals. The in-vehicle network is segmented in two main areas by the CAN Gateway, thus acting as network proxy and firewall.
- The application on-board unit (APP OBU) executes the applications that realize the project use-cases. It is an embedded PC based on a Linux OS implementing security mechanisms as recommended by the Center for Internet Security [1]. The Linux OS hosts also a Docker platform that executes third-party software modules.
- The V2X on-board unit (V2X OBU) manages V2X messages according to the ETSI ITS-G5 standard. It, hence, de-/encapsulates V2X messages (CAM [2], DENM [3], SPaT [4], MAP [5])[2] and exposes their payload via APIs.
- The Network Gateway connects the vehicle components through a private Ethernet network and provides the mobile Internet access
- The vehicle user-interface consists of a tablet connected to the in-vehicle Wi-Fi provided by the Network Gateway and an after-marked vehicle user-interface
- The R-ITS-S is connected to the vehicle via ETSI ITS-G5 and to the cloud via wired connectivity. It hence is equipped with a V2X module (V2X module) that de-/encapsulates V2X messages (CAM [2], DENM [3], SPaT [4], MAP [5]) according to the ETSI ITS-G5 standard and exposes their payload via APIs or collect the content to be sent by the V2X through these APIs. The R-ITS-S is also connected through private IP network to the C-ITS-S (a cloud service) to: (i) share the collected information about the vehicles, and (ii) receive notifications and data to share with the vehicles.

The SAF execution has focused one the most critical parts and tested the V2X OBU, the APP OBU and the HSM. The bench prototype has been implemented as a bench to work within laboratory conditions in order to allow extensive testing of the use-cases of interest and assess the complete security of the obtained simulated system. To this aim:

- The CAN data for the vehicle is provided by replaying previously recorded traces collected by running the use-cases, i.e., driving the car in the road according to pre-defined scenarios.
- The GNSS signal for both the vehicle and the RITS-S are provided by replaying previously recorded traces collected by driving the car on the road.
- The C-ITS-S station use both data previously collected by tracing actual execution and also simulated data about traffic events (traffic jam, traffic accidents etc.) and traffic light phases.

[2] See the SAFERtec D4.2 and D4.3 deliverables for detailed description of the usage of those messages

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The bench uses both mock-up testing version of cloud services as well as real-world version of the service, such as the traffic information service.

Thanks to this data (real and simulated ones) the bench can reproduce continuously the selected use-cases (cf D2.1), switch them on-the-fly, and notably, use real data. The following use-cases have been reproduced for testing purpose.

- The Optimal Driving Speed Advice via V2X. A vehicle approaching an intersection receives phase-and-timing messages from C-ITS-S via R-ITS-S by using short-range communication technology ETSI ITS-G5 (MAP[5] and SPAT[4] messages). By using this information, the vehicle can calculate the appropriate speed to be suggested to the driver for adequately reaching and crossing an intersection.
- Provision of Real-Time Traffic-hazard information via V2X. Real-time traffic information (e.g., road events and traffic-flow) is continuously retrieved by vehicles from the R-ITS-S by using short-range communication technology ETSI ITS-G5 (DENM messages [3]).
- Priority request in intersection crossing via V2X. An emergency vehicle that is approaching an intersection requests, through the R-ITS-S by issuing ETSI ITS G5 (ad-hoc CAM messages [2]), the priority from the infrastructure to cross the intersection in a quick and safe way. All vehicles are informed by the R-ITS-S about the presence of the emergency vehicle.
- The Optimal Driving Speed Advice via Cellular. The vehicle uses the cellular connectivity to receive phase-and-timing messages (MAP and SPAT messages) in order to calculate the appropriate speed for approaching and crossing an intersection. This use-case is similar to “Optimal Driving Speed Advice” use-case but in this case, the cellular network is used for the information exchange.
- Provision of Real-Time Traffic-hazard information via V2X and Cellular. The vehicle, running on the road, receives real-time traffic information (e.g., road-works, traffic-jam ahead) from a R-ITS-S, via short range communication ETSI ITS G5 (DENM messages), and additionally it receives traffic information from an online (real-world) cloud services, via cellular connectivity (DENM messages).
- Management of sensitive user information. The actual real-time traffic information is provided only to allowed (i.e. pre-authorized and registered) drivers via cellular connectivity.

Those are the use cases to be tested under simulated attacks using the methodology presented in the next section.

2.2 Attack simulation

The attack simulation will follow the classical vulnerability methodology aiming at identifying and trying to exploit vulnerabilities. A vulnerability test methodology is usually composed of two steps:

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

- Identification of potential vulnerabilities by a critical view of the information gathered throughout the evaluation and by carrying out testing (functional and security oriented):
    - On the external interfaces and the identifiable technologies
    - On similar products (products of the same functional type and scope)
- Verification and classification of the possibility of exploitation of these potential:
    - Not exploitable vulnerability: identified possible failure of a security function but the evaluator did not manage to exploit it during the test campaign;
    - Exploitable vulnerability: vulnerability tested and exploited allowing access to protected data or allowing the tampering of the TOE integrity (changing TOE executions results or expected behaviour).

2.2.1 Identification of potential vulnerabilities

The objective of this first step is to gather information in order to identify potential vulnerabilities:

- **Public vulnerabilities**: this search focuses on the TOE as a whole and on every component integrated into the TOE (using the information available in the configuration list and the design documentation). This could be done automatically via the execution of scanners.
- **Vulnerabilities of external interfaces**: search of vulnerabilities of the interfaces used to access the TOE (open communication interfaces, configuration files, GUI, console, communication protocols etc.) that could be used by an attacker to penetrate the TOE.
- **Vulnerabilities of security mechanisms**:
    - **Functional testing**: tests of the proper execution of security functions as specified in available product documentation.
    - **Independent testing**: modified functional tests (mostly unexpected inputs) and tests outside the interfaces providing the product security functions (looking for debug ports, unauthenticated interfaces, etc.).
    - **Inherent or cryptography-related vulnerabilities**: the objective is to identify the inherent vulnerabilities of the used mechanisms (for example: towards a brute force attack or a weakness of the cryptographic algorithms used).
    - **Coding vulnerabilities**: if (part of) the source code of the TOE is publicly available or by reverse-engineering, the evaluator searches for errors and design weaknesses in the source code.

2.2.2 Verification and classification of the possible vulnerabilities

The objective is to determine whether the vulnerabilities identified in the previous steps can be exploited in the testing conditions. In fact, not all potential flaws can be exploited in any configuration or execution environment of the TOE. So, it is really important to see if the vulnerability can in fact be exploited and demonstrate how protected data can be accessed or the TOE integrity can be tampered changing its expected behaviour.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

2.2.3 Selected attack vectors

One important aspect of the tests we run is the position and techniques used to run the attack. In fact, we will try to run those attacks as if we were in one of the following attacker profiles:

- Remote attacker
  - Radio media: An attacker able to emit or receive ITS G5 radio signals
  - Rogue ITS-S (vehicle or roadside unit): An attacker using a rogue equipment sends and receives ITS messages to the TOE
  - Internet: Remote attacker sending or intercepting TOE messages through the ITS central system communication network
- Local (in vehicle) Attacker
  - Rogue user: A user having a physical access to the car, provides, intercept or modify information to the TOE via the internal vehicle HMI or networks
  - Rogue administrator: An attacker using the administration interface

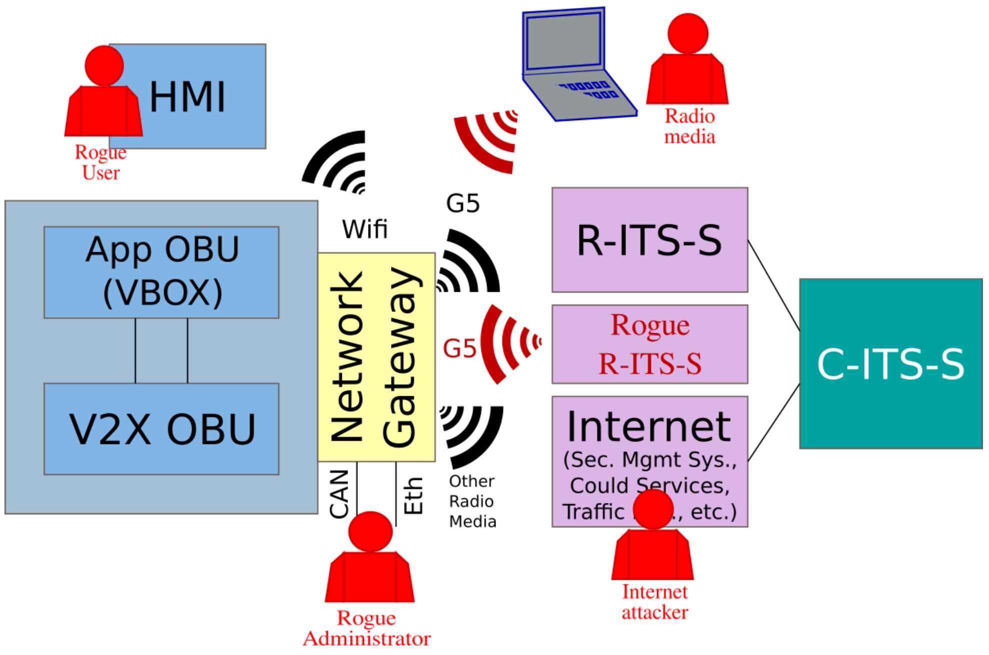


Figure 2: Attackers simulation

2.2.4 The SAFERtec System-level simulator

On a parallel testing activity, simulated attacks have been directed to instances of the SAFERtec simulator (see Appendix A 2: The SAFERtec System-level Simulator) developed to evaluate the complete stack of the ITS-V2X communication system. The software is a simulator for the full ITS radio network stack (from Physical to the Facility Layer) that can be used for *system-level* testing of security

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

controls and thus, we provide its description in this deliverable. However, its usage to validate the functionality of a CVS security control, to optimally configure it and determine the involved operational parameters has been made possible for certain CVS modules by investigating their impact on the overall system; the relevant results appear in D3.3, since they correspond to specific security functional requirements of the SAFERtec protection profile.

3 Evaluation

In order to demonstrate the resilience of the system regarding real attacks we have followed the methodology presented in section 2 and used the environment described in sub-section 2.2. We have placed the testers in the condition of a high-level attacker, corresponding to the following parameters.

3.1 Tests context

3.1.1 Evaluation parameters

- Time: 224 hours
- Conditions:
  - Black box
  - Accessible interfaces: External TOE's interfaces
  - Test mainly the presence of known vulnerability

3.1.2 Tools

- Scanner: Nmap [6], sslscan [7], Nessus [8]
- Bruteforce tools: Hydra [9]
- Proxyfyer of network traffic: BurpSuitePro [10]
- Reverse engineering tool: Javac and IDA Pro [11]
- Local Privilege Escalation Checker: LinEnum [12]
- Network Protocol Analyser Wireshark [13]

The following sections present the results of penetration tests performed on the different TOE and a summary of the vulnerabilities found.

Indeed, *due to confidentiality concerns, details on the tests and especially on the results cannot be presented in this deliverable and have been put in confidential appendices (provided only to consortium partners)*.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.1.3 Bench updates

The first round of testing done for this deliverable as well as the first execution of SAF have identified a set of vulnerabilities or weak points in the CVS. These results have been identified by the respective partners and changes have been developed in order to avoid or mitigate the vulnerabilities.

We present here how the different components have been updated between the two vulnerability tests rounds presented in section 3.

3.1.3.1 V2X OBU

The V2X Stack in the V2X OBU had an optional interface for OEM applications to connect to if needed. This optional interface was unnecessary and thus posed a threat or at least an additional open interface (and thus potential attack vector), thus the update focused to find and eliminate any interface that was not essential to the defined functions.

This led to the following changes. Within the CFF (Commsignia) docker image, the "*native.legacyUpl.enable*" was switch off in the *saf.json* configuration file (i.e. set to false). This is to disable unnecessary upload of files.

Setting of the following values on the Craton2 Commsignia its.cfg were also modified for the same reasons:

- enable-udp-inject-api=N instead of Y
- removing udp-inject-api-port=7946

3.1.3.2 Network gateway

The in-vehicle network gateway was subject to some issues; thus, the following modifications have been made to apply the recommendations.

- The firmware of the network gateway underlying the software providing the connectivity functionality has been updated to the last version, available at the time of the tests. The first pre-installed version of such a firmware was subject to some known security threats fixed by the gateway developers in the new firmware version (i.e., the last available at the time of the tests). Therefore, the new firmware version released by the gateway developers has been installed in the network gateway of the bench aiming at reducing the open issue identified and recognized by the component developers.
- Some services provided by the network gateway (i.e., SSH, Web UI based on simple http – i.e., that does not use protected communication, Command Line Interface, and Modbus) have been disabled since they are not used to reproduce the project use-cases. Instead, other services used to reproduce them or executed for administrative purposes have been preserved. For instance, the Web user interface over HTTPS has been left open for administration operations on the network gateway.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The password of the administrator user to access the Web user interface for administration operations of the network gateway has been changed. This is to follow a more restrictive indication on the password creation, thus making it difficult to be discovered by, e.g., brute force attacks.
- The port scan possibility has been disabled for preventing scanning and monitoring activities on the gateway opened ports by external entities.

3.1.3.3 VBOX (APP OBU)

The in-vehicle application on board unit (APP OBU) was subject to some issues, thus the following modifications have been made to apply the recommendations. The APP OBU is an on-board unit acting as an embedded PC based on a Linux OS that hosts the software modules and applications that implement the project use-cases.

- The world readable access to some sensible files of the Linux OS has been removed, thus preventing other entities to read such file while not explicitly allowed.
- Linux OS kernel has been updated to the last version available at the time of tests, thus patching all known issues and security threats.

3.2 Tests results

In the following summary table, we present four possible vulnerability tests classification:

- OK
    - A vulnerability identified in the first vulnerability tests round has been corrected
- NOK
    - A vulnerability found in the first round of tests and not corrected in the second
- NOK - Under SAF correction
    - Vulnerability found in the first round of tests correctly identified by the SAF application and reported to the involved partners but still remains in the module. The way and time to fix it involved their internal decisions which remained outside of the SAFERtec scope.
- NOK - Residual
    - The vulnerability is a residual vulnerability in the assurance evaluation context.
        - It is in the TOE environment and is covered by assumptions that are not here satisfied and lead to the TOE possible corruption.
        - The vulnerability does not violate any SFRs or assumptions (outside the TSF).
        - Or it corresponds to a vulnerability for which it has not been possible to demonstrate exploitability in the context of the chosen assurance level which limits the attacker capabilities.

In the complete report, every issue or vulnerability identified within the given scope includes details of the finding, the risk it implies and also a recommendation to resolve the issue. Moreover, for each

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

issue or vulnerability identified within the given scope, in order to evaluate the risk score the information presented in annex A1 has been evaluated.

3.2.1 Analysis of the V2X On Board Unit

Executive summary

The following table resumes the results of the two rounds of penetrations tests performed on the component.

ID	Description	Score Audit n°1	Score Audit n°2	State
	Classical tests			
1.1	Out-of-date kernels missing latest security fixes	Low Risk	Low Risk	NOK Under SAF correction
1.2	Root accounts used for direct logins	Low Risk	Low Risk	NOK Under SAF correction
1.3	Presence of guessable system passwords	High Risk	High Risk	NOK Under SAF correction
1.4	Network resources are not properly isolated	High Risk	High Risk	NOK Residual
	Deep dive			
1.5	Denial of Service on the ITS Access Layer	-	-	NOK Residual
1.6	Memory bugs leading to Out-of-Band read	-	-	NOK Residual
1.7	Denials of service on the ITS Application Layer (MAP protocol)	-	-	NOK Residua
1.8	Possibility to send unsecured V2X messages even if the security verification is set.	-	-	NOK Residua

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1.9	Possible to receive fake GPS position without checking the authentication of the sender	-	-	NOK Residua
1.10	Potential memory leak or denial of service	-	-	NOK Residua
1.11	Arbitrary file overwrite	-	-	NOK Residua
1.12	Out-of-band vulnerability which can potentially lead to a memory leak	-	-	NOK Residua

Table 2 Results of two rounds of SAFERtec penetration tests on V2X OBU

Kernel settings: Version

As no updates aiming at covering this vulnerability have been applied on the V2X On Board Unit, this vulnerability is still present on the V2X OBU.

Account management: Root access

This vulnerability is due to the fact that this TOE aims at finally being integrated in the WP4 Connected Vehicle and so in order to ease this integration and the use of the bench in WP5, some root accounts in the TOE have not been disabled. As no updates aiming at covering this vulnerability have been applied on the V2X On Board Unit, this vulnerability is still present on the V2X OBU.

Password policy: Guessable passwords

This vulnerability is due to the fact that this TOE aims at finally being integrated in the WP4 Connected Vehicle and so in order to ease this integration and the use of the bench in WP5, some weak passwords have been used to ease the access to the bench. As no updates aiming at covering this vulnerability have been applied on the V2X On Board Unit, this vulnerability is still present on the V2X OBU.

Network exposure: Network segregation

As no updates aiming at covering this vulnerability have been applied on the V2X On Board Unit, this vulnerability is still present on the V2X OBU.

3.2.1.1 *Deep-dive analysis on the V2X on-board unit*

After realizing the first round of penetration test, partners decided to focus their investigation on the V2X On Board Unit. So, they realized a deep-dive analysis on it in order to identify complex and unknown vulnerabilities. This a one of the major differences between penetration testing and deep-dive analysis: the latter focuses on one solution by studying how it is designed and implemented in

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

order to find unknown vulnerabilities, where penetration tests mainly exploit already known vulnerabilities.

Methodology

- Evaluation parameters
    - Time: 232 hours
    - Conditions:
        - Black box
        - Accessible interfaces: External TOE's interfaces
        - Test focus and unknown vulnerabilities
- Tools:
    - Reverse engineering tool: IDA Pro
    - Wireshark
    - Gdb
    - AFL-fuzz
    - Unicorn

A precise description of the instrumentation developed by auditors in order to find vulnerabilities on the V2X On Board Unit can be found in the appendix belowA3.

Overview of the V2X On Board Unit

This overview reflects only the auditors' understandings of the V2X On Board Unit after a first study of its architecture, so it can be inaccurate or inexact. The aim of this first study was to identify interesting critical software which can be more deeply investigated during the deep-dive analysis.

The V2X OBU is based on "Autotalks" second generation, with a Linus Operating System and a CPU architecture based on ARM.

The main application *"/usr/bin/its"* is developed in C language (labelled *VEHICLE_COMMUNICATION_APP* in the following).

This application starts many threads. Each of them manages only one specific task. Once a job is finished, produced data are forwarded to others. Due to confidentiality purpose other details on the threads cannot be shared in this document, but can be found on the appendix.

Moreover, the V2X OBU is connected to three networks:

1. On Board Device LAN, to communicate with others on board devices such as:
    a. APPS V2X device,
    b. *VEHICLE_COMMUNICATION_APP* which listens on TCP port 7942.
    Note: This API is used to get and set information on V2X OBU, like GPS position, traffic jam and so on.
2. 802.11p radio communication, to exchange information with other vehicles or road side units

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3. CAN network, to get and set data with vehicle ECUs (engine status, parking brake status)

Focus has been made on media that could be reached by extern attackers and more specifically on the API in On Board Device LAN, and on exchanges through the 802.11p radio communication.

The following sections present the results of the deep dive analysis performed on the V2X OBU TOE and a summary of the vulnerabilities found. Indeed, due to confidentiality concerns, details on the tests and results cannot be presented in this deliverable and have been put in confidential appendices.

Detailed findings

a) ITS access layer implementation (V2X radio)

Denial of Service on the ITS Access Layer has been found, which can represent a risk for the TOE and highlight an incomplete implementation of the SFR regarding error management. At the time of the evaluation, auditors have judged that the risk was very high, as it can lead to a crash of the VEHICLE_COMMUNICATION_APP.

- Description

The access layer bundles the data link layer and the physical layer, and is situated at the bottom of the ITS protocol stack.

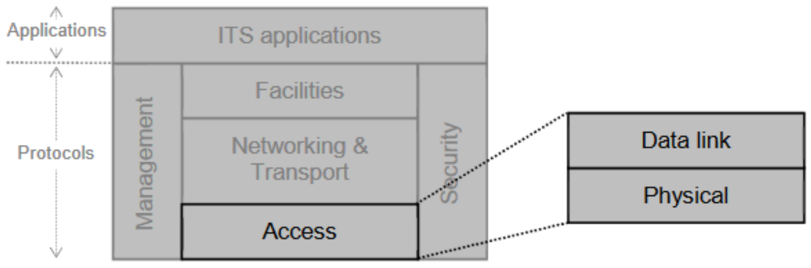


Figure 3 ITS protocol stack – Access layer

As the physical layer relies on electronic and frequency aspect, only the implementation of data link layer has been tested.

The data link layer consists of two sublayers:

- medium access control (MAC);
- logical link control (LLC);

The former provides means for distinguishing between different network layer protocols and the latter is responsible for scheduling transmissions to minimize interferences between ITS stations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- Finding

Auditors have identified one critical vulnerability on this network layer leading to a crash of VEHICLE_COMMUNICATION_APP. This issue is due to an improper management of error code. In a particular case, a check is achieved between the real size of the received V2X frame and the size provided into a field of the received V2X frame. When these two values are not equal, the software detects the issue but unfortunately returns a success code instead of an error code. This leads further to a memory access violation.

- Faced risk

A malicious user can cause a Denial of Service of the VEHICLE_COMMUNICATION_APP, resulting in making the vehicle blind.

- Recommendation

Fix the VEHICLE_COMMUNICATION_APP source code in order to return the appropriate error code.

- Updates realised on the TOE to cover the vulnerability

In the limited time between the two rounds of penetration tests, no updates aiming at covering this vulnerability have been made on the V2X OBU. However involved partners have worked or are still working to update their component and cover it.

b) ITS networking and transport layer implementation

Memory bugs leading to Out-of-Band read have been found, which can represent a risk for the TOE and highlights an incomplete implementation of the SFR. At the time of the evaluation, auditors have judged that the risk was medium.

- Description

The ITS networking & transport layer is composed of protocols for data delivery among ITS stations and from ITS stations to other network nodes, such as network nodes in the core network (e.g. the Internet). ITS network protocols particularly include data routing from source to destination through intermediate nodes and the efficient dissemination of data in geographical areas.

The ITS networking and transport layer gathers several networking and transport protocols.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

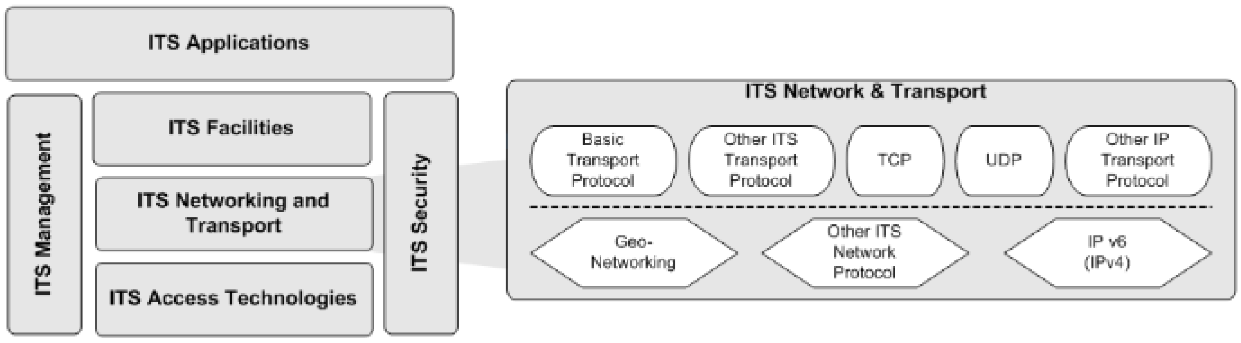


Figure 4 ITS protocol stack – Networking and Transport layer

During the duration of the project, auditors have only had time to analyse the following protocols from the ITS networking and transport layer:

- GeoNetworking protocol,
- Transport protocols over GeoNetworking: (BTP-A and BTP-B).

However, it should be noticed that in order to check the security assurance level and also to decrease the security risk, vulnerability research should be also performed on the uncovered protocols to test them.

- Finding

We have identified three memory bugs leading to an Out-of-Band read, i.e. an access memory which occurs out of the received V2X data frame.
These issues come from a missing check before comparing data into the frame and stored data into the VEHICLE_COMMUNICATION_APP.

- Faced risk

These issues can be used for a memory leak.

- Recommendation

A security check on the accessed data must be done before any comparisons between the V2X frame and the VEHICLE_COMMUNICATION_APP.

c) ITS applications layer implementation

Denials of service on the ITS Application Layer (MAP protocol) have been found, which can represent a risk for the TOE and highlight an incomplete implementation of the SFR specially regarding error

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

management for MAP messages. At the time of the evaluation, auditors have judged that the risk was very high as it can lead to a crash of the VEHICLE_COMMUNICSATION_APP.

- Description

The ITS applications on top of the protocol layers realize the use-cases for road safety, traffic efficiency, infotainment and business.

The service layer consists of the implementation of the respective standards:

  - CAM;
  - DENM;
  - SPaT;
  - MAP;

- Finding

Auditors have identified a critical vulnerability on the implementation of MAP protocol which results in a crash of VEHICLE_COMMUNICATION_APP. This issue is due to an unintended behaviour during an ASN1 processing. The latter leads to an infinite loop of memory allocations and causes a memory exhaustion.
Auditors suppose that the created *malicious* frame can also lead to a complete Denial of Service of the application if they are able to generate frame that drive to multiple allocations until complete memory resource exhaustion on real device. But during the duration of the project, auditors did not have time to test it and so confirm this hypothesis.

- Faced risk

A malicious user can cause a Denial of Service of the VEHICLE_COMMUNICATION_APP, resulting in making the vehicle blind.

- Recommendation

Due to the complexity of ASN1 process, auditors have not investigated the original source of this issue. But from the detailed analysis provided in confidential appendix, a developer should be able to identify the bug and to fix it into the VEHICLE_COMMUNICATION_APP code source.

d) Security verification of V2X messages

Auditors have found that it was possible to send unsecured V2X messages even if the security verification was set, when they have tested these SFRs. That implies and highlights that the SFRs have not been fully implemented specially regarding error management for CAM messages. At the time of the evaluation, auditors have judged that the risk was very high as it can lead to a crash of the VEHICLE_COMMUNICATION_APP.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- Description

For incoming messages, the stack verifies the frame and provides it to the respective service for processing (e.g. to CAM service if the V2X payload has been identified to be a CAM). The stack uses security verifications to check the sender of the message and the received certificate chains are verified to have a trusted authority at the end. It also checked that all messages contain only allowed information and all the needed ones – e.g. an emergency vehicle CAM message must have a Service Specific Permission within its certificate. Only verified messages are sent and used by upper layers.

- Finding

Auditors have noticed that in specific cases unsecured V2X messages are processed up to the ITS Applications Layer even if the security verification is set.

- Faced risk

A malicious user is able to send non-secured messages towards the ITS applications layer without being filtered or rejected. Consequently, it is possible to impersonate any ITS stations and so to deliver fake information.

- Recommendation

Depending on whether this issue is a misconfiguration issue or a design issue, the recommendations are the following:

  - Configure more precisely the ITS station configuration file,
  - Modify the VEHICLE_COMMUNICATION_APP source code in order to filter or reject unsecured packet as soon as secure mode is set.

e) GPS position and time attack of V2X messages

Auditors have found that it was possible to receive fake GPS position without checking the authentication of the sender, when they tested this SFR. That implies and highlights that the SFR has not been fully implemented on the TOE. At the time of the evaluation, auditors have judged that the risk was very high as fake GPS information can affect the vehicle behaviour.

- Description

Each vehicle exchanges information about its position, its trajectory and to some extent drivers' intentions (e.g., turn indicators status, acceleration, engaged brake) through the ETSI ITS-G5 protocol. This information is collected by other surrounding vehicles and road-side units.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- Finding

Auditors have noticed that any GeoNetworking frames transport GPS information of the sender vehicle. These GPS coordinates seem to be directly interpreted by other vehicles without any authentication even if the secured protocol is used.

- Faced risk

An attacker can use a compromised ITS station or any 802.11p interfaced in order to send fake GPS position, in order to affect the vehicle behaviour. Moreover, the Road Side Unit is probably also impacted as it collects and aggregates the information from the vehicles and forwards it to the Central ITS station.

- Recommendations

When the secured GeoNetworking protocol is used, the recommendation is to authenticate the ITS station before taking into account the received GPS coordinates.

f) V2X OBU API

- Finding
    - Denial of Service

A potential memory leak or denial of service has been identified from a static analysis. But, at the time of writing this report, auditors did not have enough time to confirm this finding with a physical test on the TOE.

- Faced risk

These two Denials of Service can potentially make the vehicle blind by crashing the VEHICLE_COMMUNICATION_APP.

- Recommendation

A security check on the accessed data must be done before any comparisons between the V2X frame and the VEHICLE_COMMUNICATION_APP.

g) V2X OBU API

- Finding

Arbitrary file overwrite has been found.

An attacker can exploit this vulnerability and send lots of frame. This will lead to multiple file creations with different and random names, conducting to inode exhaustion and so unavailability of the device.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Also, the created file name may be controlled by the attacker, for example, by spraying the stack memory with previously sent frames. We do not have time to verify this scenario. If he can do such action, he can rewrite important system files, make the device forever unusable.

- Faced risk

This can result in corrupting the on-board Autotalks file system, which will require a factory service.

- Recommendation

A security check on the accessed data must be done before any comparisons between the V2X frame and the VEHICLE_COMMUNICATION_APP.

h) V2X OBU API

Out-of-band vulnerability which can potentially lead to a memory leak has been found by testing this SFR implying that the SFR has not been fully implemented. At the time of the evaluation, auditors have judged that the risk was medium.

- Faced risk

This vulnerability can potentially lead to a memory leak.

- Recommendations

A security check on the accessed data must be done before any comparisons between the V2X frame and the VEHICLE_COMMUNICATION_APP.

3.2.2 Analysis of the VBOX (APP OBU)

Executive summary

The following table resumes the results of the two rounds of penetrations tests performed on the component.

ID	Description	Score Audit n°1	Score Audit n°2	State
2.1	Out-of-date kernels missing latest security fixes	Low Risk	Low Risk	NOK Under SAF correction
2.2	Presence of services running with administrative privileges	Medium Risk	Medium Risk	NOK Under SAF correction
2.3	Presence of sensitive world-readable files	High Risk	Covered Risk	OK
2.4	Presence of guessable system passwords	High Risk	High Risk	NOK

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

				Under SAF correction
2.5	Network resources are not properly isolated	High Risk	High Risk	NOK Residual

Table 3 Results of two rounds of SAFERtec penetration tests on VBOX

Kernel settings: Version

Linux OS kernel has been updated from version 6 to 11. However Common Vulnerabilities and Exposures exist on the version 11[3]. So even with this upgrade, the VBOX kernel can still be affected by numerous vulnerabilities.

Account management: Services privileges

Some services are running with administrative privileges.

As no updates aiming at covering this vulnerability have been applied on the VBOX, this vulnerability is still present on the VBOX.

Access rights and permissions: World-readable files

The world readable access to some sensible files of the Linux OS has been removed, thus preventing other entities to read such file while not explicitly allowed.

This vulnerability does no more exist in the VBOX.

Password policy: Guessable passwords

This vulnerability is due to the fact that this TOE aims at finally being integrated in the WP4 Connected Vehicle and so in order to ease this integration and the use of the bench in WP5, some weak passwords have been used to ease the access to the bench.

As no updates aiming at covering this vulnerability have been applied on the VBOX, this vulnerability is still present on the VBOX.

Network exposure: Network segregation

[3] https://www.cvedetails.com/vulnerability-list/vendor_id-33/product_id-47/version_id-208084/Linux-Linux-Kernel-4.9.11.html

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Network resources are not properly isolated: some information or some services are accessible to anyone.

As no updates aiming at covering this vulnerability have been applied on the VBOX, this vulnerability is still present on the VBOX.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.2.3 Analysis of the Network Router

Executive summary

The following table resumes the results of the two rounds of penetrations tests performed on the component.

ID	Description	Score Audit n°1	Score Audit n°2	State
3.1	Out-of-date kernels missing latest security fixes	Low Risk	Low Risk	OK
3.2	Use of unmaintained or out-of-date software	Medium Risk	Medium Risk	NOK
3.3	Network services unneeded for the proper operation of the systems are open	Medium Risk	Covered Risk	OK
3.4	Clear text submission of password during authentication	N/A	Medium Risk	NOK
3.5	Root accounts used for direct logins	Low Risk	Low Risk	NOK
3.6	Presence of guessable system passwords	High Risk	High Risk	OK
3.7	Network resources are not properly isolated	High Risk	High Risk	NOK

Table 4 Results of two rounds of SAFERtec penetration tests on network router

Kernel settings

The firmware of the Network Router has been updated to the last version available at the time of tests. So, this vulnerability is covered at the time of the evaluation.

Software security

Out-of-date or unmaintained software pose a greater risk of having a vulnerability exploited.

As no updates aiming at covering this vulnerability have been applied on the Network Router, this vulnerability is still present on the Network Router.

Open services

The auditors have noticed that no more network services unneeded for the proper operation of the system were open. So, this vulnerability is now covered.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Authentication

A new vulnerability has been identified by auditors: during authentication they have noted that relevant passwords were submitted in clear text.

- Finding

In order to access the traffic data, the bench needs to authenticate to a Swarco online service. This authentication is made by the VBOX through HTTP and credentials are thus sent in cleartext.

- Faced risk

Without encrypted data exchanges, it would be possible for an attacker which has properly caught the "bench" within the range of a fake 2G GSM base station to retrieve the credentials that are used to authenticate to the Swarco service.

- Recommendation

Use HTTPS for authentication.

Root access

Having multiple persons using the root account increases the risks of administrative credentials theft. Moreover, in case of a compromise, it will make it more difficult to investigate. As no updates aiming at covering this vulnerability have been applied on the Network Router, this vulnerability is still present on the Network Router.

Guessable passwords

As the password of the administrator user has been hardened, this vulnerability has been covered.

Network segregation

As the port scan possibility has been disabled on the Network Router for preventing scanning and monitoring activities on the gateway opened ports by external entities, this vulnerability has been covered.

3.2.4 Analysis of the Road Side Unit

Executive summary

The following table resumes the results of the two rounds of penetrations tests performed on the component.

ID	Description	Score Audit n°1	Score Audit n°2	State
4.1	Out-of-date kernels missing latest security fixes	Low Risk	Low Risk	NOK

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4.2	Root accounts used for direct logins	Low Risk	Low Risk	NOK
4.3	Presence of services running with administrative privileges	Medium Risk	Medium Risk	NOK
4.4	Presence of sensitive world-readable files	High Risk	High Risk	NOK
4.5	Presence of guessable system passwords	High Risk	High Risk	NOK
4.6	Network resources are not properly isolated	High Risk	High Risk	NOK

Table 5 Results of two rounds of SAFERtec penetration tests on the RSU

Version

Out-of-date kernels missing the latest security fixes pose risk of being used by a malicious person to carry out privilege escalation attacks. In some cases, a public exploit may even be available.

As no updates aiming at covering this vulnerability have been applied on the RSU, this vulnerability is still present on the RSU.

Root access

This vulnerability is due to the fact that this TOE aims at finally being integrated in the WP4 Connected Vehicle and so in order to ease this integration and the use of the bench in WP5, some root accounts in the TOE have not been disabled.

As no updates aiming at covering this vulnerability have been applied on the RSU, this vulnerability is still present on the RSU.

Service privileges

Services running with administrative privileges have been found, which can represent a risk for the product.

As no updates aiming at covering this vulnerability have been applied on the RSU, this vulnerability is still present on the RSU.

World-readable files

Sensitive world-readable files have been found, which can represent a risk for the product.

As no updates aiming at covering this vulnerability have been applied on the RSU, this vulnerability is still present on the RSU.

Guessable passwords

This vulnerability is due to the fact that this TOE aims at finally being integrated in the WP4 Connected Vehicle and so in order to ease this integration and the use of the bench in WP5, some weak passwords have been used to ease the access to the bench.
As no updates aiming at covering this vulnerability have been applied on the RSU, this vulnerability is still present on the RSU.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Moreover, a new weak password has been found during the second audit. Indeed, the Knopflerfish OSGi console exposed by the RSU is accessible with the username admin and the password admin. This console can be used to install custom bundles and gain arbitrary code execution on the RSU.

Network segregation

Network segregation involves developing and enforcing rules controlling which network hosts are permitted to communicate with which other network hosts. It is an effective way to mitigate the second stage of a malicious intrusion, propagation and lateral movement. If implemented correctly, it can make it significantly more difficult for a malicious person to locate and gain access to sensitive information.

As no updates aiming at covering this vulnerability have been applied on the RSU, this vulnerability is still present on the RSU.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4 SAF suitability analysis

During the vulnerability tests run in the second campaign we have obtained the following results:

- A total of 30 vulnerabilities have been found during the two vulnerability tests campaigns
    - 12 for the V2X OBU
        - 3 under SAF correction (identified by SAF execution in T3.3 but still under correction process)
            - Including 1 to be covered by administrator guidance and TOE set-up
        - 9 residual vulnerabilities (1 that should be covered by an assumption, 7 residual due to lack of demonstration of exploitability for chosen assurance level, 1 outside the TSF)
    - 5 for the VBOX
        - 1 corrected by SAF between the two evaluation rounds
        - 3 under SAF correction (identified by SAF execution in T3.3 but still under correction process)
        - 1 outside the TOE boundaries
    - 7 for Network Router
        - 3 have been corrected in the context of WP4 following the first vulnerability tests results
        - 4 vulnerabilities remaining for which no exploits have been developed to demonstrates impacts on the TOEs SFRs
    - 6 for the RSU
        - 6 vulnerabilities remaining for which no exploits have been developed to demonstrates impacts on the TOEs SFRs

For the component that have been evaluated in T3.3 we can note that all the potential vulnerabilities found are either under correction process and would have been corrected by the SAF evaluation if ran completely thanks to product updates. One identified vulnerability (guessable password) will actually not be corrected by a product update but has actually been identified by SAF as the necessity to update the product guidance to clearly notify the administrator to change the default passwords during the product set-up. Two vulnerabilities have been found in the environment of the product. The analysis of these vulnerabilities that are actually covered by STs assumptions, has revealed a real potential for threats that even if acceptable for Day 1 use cases, they cannot be tolerated for more sensitive use cases like semi-autonomous or autonomous driving. In fact, for those use cases this threat is too important to be considered as covered by nontechnical counter-measures. The identified vulnerabilities allow to attack the TOE from inside the vehicle network where the interfaces are not protected by SFRs. We recommend for those vulnerabilities to add new SFRs that enforce stronger identification/authentications mechanisms on all TOEs interfaces and also enforce data validity checks in order to refuse tampered data.

For the non-evaluated components of the CVS, many vulnerabilities have been found (13). Since they are not developed by the project developers SAF cannot enforce the correction of all of them.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

However, an important information to be noted, is that even in a system where only partial evaluation have been run (partial STs for a limited number of components and partial execution of the evaluation framework), the vulnerabilities found did not allow us to exploit them to bypass the evaluated SFRs. So, at the very least, the SAF evaluation made the TOE not sensitive to basic attacks even in the case of vulnerable devices in their environment.

As a more general note, thanks to SAF, (i) no vulnerabilities have been found in the TOE that are not to be corrected or deemed acceptable by the chosen STs and (ii) the TOEs are resilient even in the case of vulnerable environment. The vulnerabilities found that could be exploited were outside the scope of the chosen STs and were willingly considered acceptable. Those are demanding attacks that can only be orchestrated by advanced attackers. The D5.3 will study the capabilities of SAF to handle those vulnerabilities and the associated development of counter-measures to mitigate them.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

5 Conclusions

The first analysis done is WP5 has been a theoretical study of the SAF characteristics and its comparison to other existing approaches. The second analysis that task 5.2 aims at providing, is a more practical demonstration of the real benefits of SAF on (close to) real use cases. In this deliverable we have demonstrated the capacity of the elements validated by the SAF framework to counter real threats on the test bench developed in WP4.

The idea is to get a more concrete feedback of its efficiency and the benefits it provides in terms of security on real implementations. For this deliverable we identified how to provide further evidences than the one already produced for D3.3 and D5.1. In fact, D3.3 provides tests and feedback on the security assurance that SAF achieves through the identification and fixing of flaws, security objectives reviews, etc. In the D5.1 we have also already theoretically evaluated the framework through its comparison with other cyber-security evaluation approaches. In this task, we have provided an additional feedback as to how we assess the SAF benefits, this time in a close to real environment and with online attacks mimicking real ones. Two vulnerability tests campaign were performed before and after the SAF application and the relevant results are compared.

In order to assess the security provided in a real context, we propose to simulate attacks in a more realistic way. The implementation of this context is twofold:

- the simulation of complete use cases (cf. section 2.1)
- and the black box vulnerability testing (cf. section 2.2)

So, we provided additional test cases and revealed SAF's sensitivity to capture the assurance levels with respect to the simulated conditions. Here, tests undertaken were not bounded by any STs restrictions or oriented by any risk analysis. The attacks tested were fully left to the evaluator choices using any possible means available in the WP4 implementation to evaluate the final resilience of the system, not its individual components under specific assumptions.

The deliverable first presents the complete methodology: the general description of the vulnerability test methodology used (system black-box testing), the simulation environment (reminder of the WP4 use case implementation) and the evaluation technical details (tools, time spent, etc.). The main idea was to use black box testing, i.e. the attacker (the tester) had only access to the system external interfaces and did not use any further information than public information in order to evaluate the final security properties of the system after we ran SAF. We chose to run two vulnerability tests campaigns in order to compare results before and after running SAF. Also the second vulnerability tests phase introduced a new set of deeper analysis to further increase potential vulnerabilities feedback (deep-dive analysis).

In this document we provide the final test results: a description of the potential vulnerabilities found and the analysis of the results regarding the SAF execution. The approach allowed us to make new tests and find a broad set of vulnerabilities. The analysis of these new results helped us classify all the vulnerabilities found in the following sets:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- Vulnerability under a pending correction already identified in T3.3 (due to the SAF execution limitations in the context of the project).
    - This is most of the vulnerabilities founds for the TOEs. These vulnerabilities actually demonstrate SAF value.
- Vulnerabilities outside the TOE's scope identified and evaluated in T3.3 with no impact on TOE's assets and system security objectives.
- Residual vulnerabilities exploitable outside the scope of the identified STs defined in D3.3 (attacks actually covered by the assumptions of physical protection or secure usage of specific devices under user's responsibility).

The first set of vulnerability comes from the fact that project resources and objectives limited the SAF execution, since a *full execution* would lie outside the project's capabilities and appears to be not necessary (to prove the SAF efficiency). Our carefully designed testing still validates the efficiency of the framework having already identified specific needs for correction.

The second set demonstrates that the global approach of the framework (defining security objectives at system level and deriving the associated local security requirements) in fact enforces the proper system's security. And finally, the last sets demonstrate that the framework manages to specifically enforce the chosen security, demonstrating that the SAF allows stakeholders to tailor security validation to their actual needs. It also demonstrates how important a good security target is in the assurance security approach.

This presents the introduced framework with the capacity to evaluate efficiently the chosen security, focusing resources only where the stakeholders choose-to and not losing them on unnecessary assurance efforts. Of course, the last set depends on the stakeholder's capacities and risk analysis definition to correctly identify real threats and security challenges. But with the appropriate risk methodology as the one defined by SAF, our results suggest that they will. Finally, the need for a reference PP to gather this knowledge and define commonly accepted requirements, is highlighted.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

6 References

[1] Center for Internet Security, Inc.: non-profit entity to safeguard private and public organizations against cyber threats. [Online] https://www.cisecurity.org/

[2] ETSI EN 302 637-2 V1.4.1 (2019-04) Specification of Cooperative Awareness Basic Service

[3] ETSI EN 302 637-3 V1.2.1 (2014-09) Specifications of Decentralized Environmental Notification Basic Service

[4] Amsterdam Group, "Signal Phase and Time (SPAT) and Map Data (MAP)", white paper, 2015-09-01 https://amsterdamgroup.mett.nl/downloads/handlerdownloadfiles.ashx?idnv=500795

[5] see the above reference

[6] Nmap: open-source network scanner. [Online] https://nmap.org/

[7] SSLScan is a fast SSL port scanner. [Online] https://github.com/rbsec/sslscan

[8] Nessus open-source network vulnerability scanner. [Online] https://www.tenable.com/products/nessus

[9] Hydra brute force password cracking tool. [Online] https://www.hackingarticles.in/comprehensive-guide-on-hydra-a-brute-forcing-tool/

[10] Burp Suite platform for security testing of web applications. [Online] https://portswigger.net/burp

[11] IDA Pro multi-platform, multi-processor dis-assembler [Online] https://www.hex-rays.com/products/ida/

[12] [Online] https://github.com/rebootuser/LinEnum

[13] Wireshark network protocol analyser [Online] https://www.wireshark.org/

[14] OpenC2X, Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5 [Online] https://www.ccs-labs.org/software/openc2x/

[15] ETSI ES 202 663 Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band

[16] K. Shi and E. Serpedin, "Coarse frame and carrier synchronization of OFDM systems: a new metric and comparison," IEEE Transactions on in Wireless Communications, vol. 3, no. 4, pp. 1271-1284, Jul July 2004.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

7 Appendices

A 1: Norms and conventions used for penetration tests results

Every vulnerability found and presented in the complete (confidential) reports includes details of the finding and the risk it covers. Similarly, every issue or vulnerability identified within the given scope includes details of the finding, the risk it implies and also a recommendation to resolve the issue. Moreover, for each issue or vulnerability identified within the given scope, we also assess the following information:

- The likely source of the vulnerability:

Design	These vulnerabilities come from mistakes in the specification process.
Implementation	These vulnerabilities come from mistakes in the development process.
Configuration	These vulnerabilities come from errors inside the settings of components.
Operation	These vulnerabilities come from errors done by working operators.

- The impact the vulnerability may have on the audited asset or group of assets:

Low	These vulnerabilities pose little threat and can be useful only together with other vulnerabilities or as a mean to speed up the information gathering process.
Medium	These vulnerabilities pose a direct but limited threat to security or may be used to gather sensible and otherwise hard-to-get information.
High	These vulnerabilities pose a significant and direct threat to security but have some limitations on the extent to which they can be exploited.
Very High	These vulnerabilities pose a critical and immediate threat to security and therefore should be dealt with in the shortest possible timeframe.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- How easy it would be for a malicious person to exploit the vulnerability (exploitability):

Easy	These vulnerabilities require standard software to exploit them and little to no knowledge on behalf of a malicious person.
Moderate	These vulnerabilities require standard software or tools publicly available to exploit them and some knowledge on behalf of a malicious person.
Difficult	These vulnerabilities require standard software or tools publicly available to exploit them and specific knowledge on behalf of a malicious person.
Very difficult	These vulnerabilities require specific access or custom-made tool to exploit them and also significant knowledge on behalf of a malicious person.

- The risk implied by the vulnerability, based on its impact and exploitability ratings:

Exploitability Impact	**Very difficult**	**Difficult**	**Moderate**	**Easy**
Low	Low	Low	Medium	High
Medium	Low	Medium	Medium	High
High	Medium	High	High	Very high
Very high	Medium	High	Very high	Very high

It should be noted that impact and exploitability ratings are deduced by the auditors from more precise metrics from the open industry standard CVSS (Common Vulnerability Scoring System). We also provide them for each issue or vulnerability identified within the given scope.

- The impact on the confidentiality of the targeted asset (C):

None (N)	There is no impact on the confidentiality of the target.
Partial (P)	There is considerable disclosure of information, but the scope of the loss is constrained such that not all of the data is available.
Complete (C)	There is total information disclosure, providing access to any data hosted on the target.

- The impact on the integrity of the targeted asset (I):

None (N)	There is no impact on the integrity of the target.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Partial (P)	The modification of some data is possible but the scope of the modification is limited.
Complete (C)	There is total loss of integrity and the attacker can modify any data on the target.

- The impact on the availability of the targeted asset (A):

None (N)	There is no impact on the availability of the target.
Partial (P)	There is reduced performance or loss of some functionality.
Complete (C)	There is total loss of availability of the target.

- The access vector that assesses the potential sources of exploitation (AV):

Local (L)	The attacker must have physical access to the target and possibly also own a local user account.
Adjacent network (A)	The attacker must have access to the broadcast or collision domain of the target (e.g. ARP cache or wireless network attacks).
Network (N)	The vulnerable interface is working at layer 3 or above of the OSI network stack (these types of vulnerabilities are often described as remotely exploitable like a buffer overflow in a network service).

- The access complexity that assess the conditions an exploitation requires (AC):

High (H)	Specialized conditions exist, such as a race condition with a narrow window, or a requirement for social engineering methods that would be readily noticed by knowledgeable people.
Medium (M)	There are some additional requirements for access, such as a limit on the origin of the attacks or a requirement for the target to be running with an uncommon, non-default configuration.
Low (L)	There are no special conditions for access, such as when the target is available to a large number of users or when the required configuration is ubiquitous.

- The number of authentications required to exploit the vulnerability (Au):

Multiple (M)	The attacker must authenticate two or more times in order to exploit the vulnerability, even if the same credentials are used each time.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Single (S)	The attacker must authenticate once in order to exploit the vulnerability.
None (N)	There is no requirement for the attacker to authenticate.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

A 2: The SAFERtec System-level Simulator

In the course of the project, a simulator was built enabled to evaluate the complete stack of the V2X communication system. The simulator consists of two subsystems:

- The radio system level simulator, implemented in MATLAB/ OCTAVE.
- The ETSI ITS stack implementation based on the Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5 [14] (OpenC2X).

In the following paragraphs, the two modules of the simulators are briefly analyzed.

1.1 ITS-G5 Simulator

For the SAFERtec project, the radio technology that is considered for adhoc V2X networking is the ITS-G5 [15] based on the IEEE 802.11p standard. This platform may be used to test, evaluate, and examine security risks and controls for the physical (PHY) and medium access control (MAC) layers of V2V/V2I links as well as the system tolerance in interference and errors that propagate through the higher layers of the ETSI ITS stack.

1.1.1 PHY Layer

The block-diagram in Figure 5represents the system structure. The system can be divided into 3 main parts, transmitter, receiver, and channel. An object-oriented approach was selected and an object for each module has to be instantiated in order to define a transceiver simulation node. Besides the object specific functions, a common-use set of functions was created, like fast Fourier transform (FFT) and Inverse FFT (IFFT) implementation. Transmitter’s model includes adaptive modulation and coding mechanism, supporting 4 types of modulations and three coding rates. Also, scrambling and interleaving functions are available. Data symbols are modulated using orthogonal frequency division multiplexing (OFDM), which is easy to achieve via proper symbol mapping and use of FFT. Furthermore, short training preamble, long training preamble and signalling data are created, as specified by the IEEE802.11p standard. Receiver’s model, on the other hand, consists of the transmitter’s “mirror” functions, such as de-mapper, de-modulator, etc. Essential receiver’s functions are considered to be channel estimation, where CSI is given by long training preamble, and detector, where signal sensing is achieved from reaped correlations of short training preamble. The detection algorithm that has been used is based primarily on the Shi-Serpedin algorithm [16].

The software diagrams shown in Figure 6and Figure 7offer a more vivid image of the transmitter and receiver functions. Note that the [1 x n] notation indicates a vector, while [m x n] denotes an m×n matrix.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

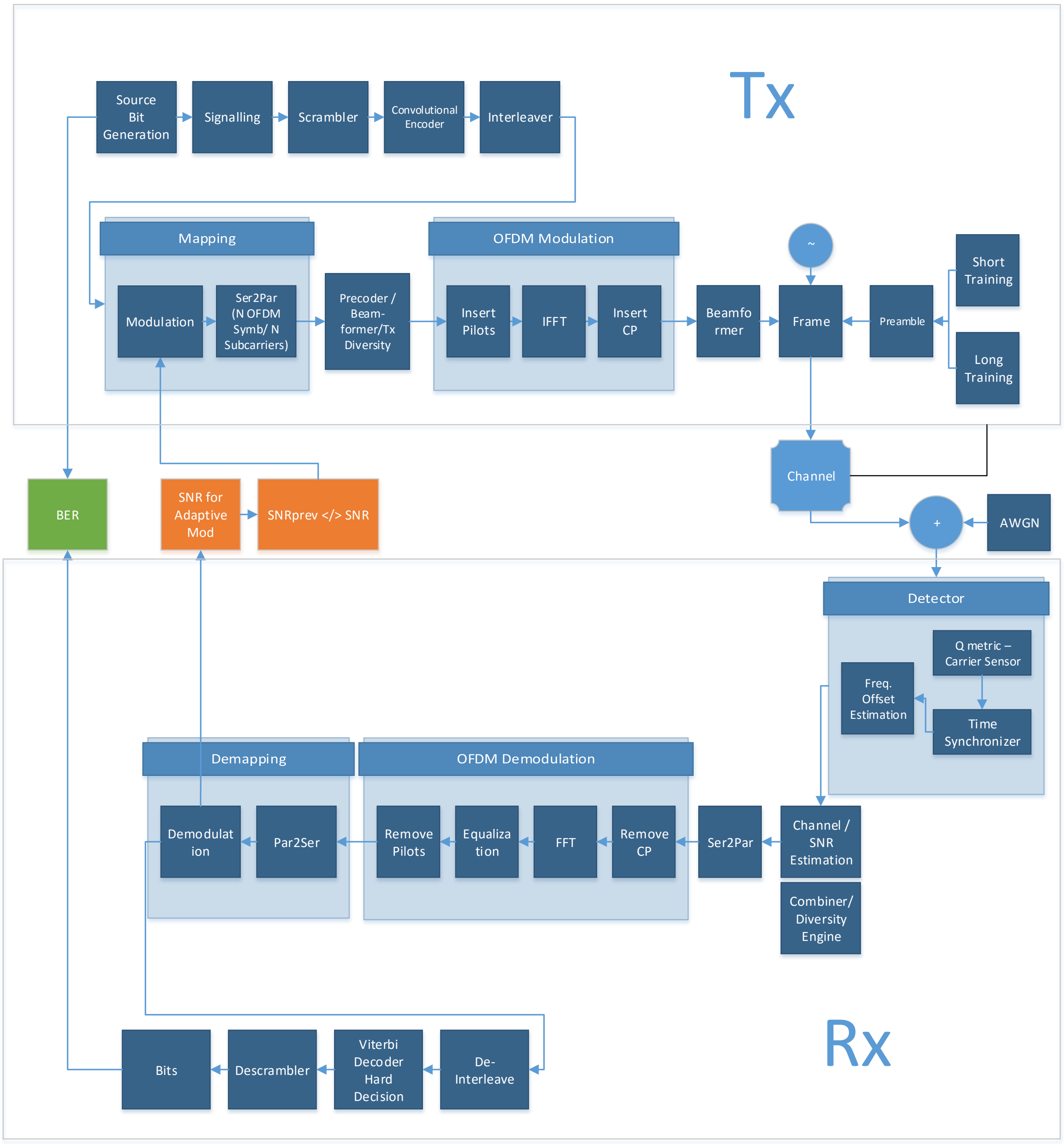


Figure 5 The 802.11p simulator model.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

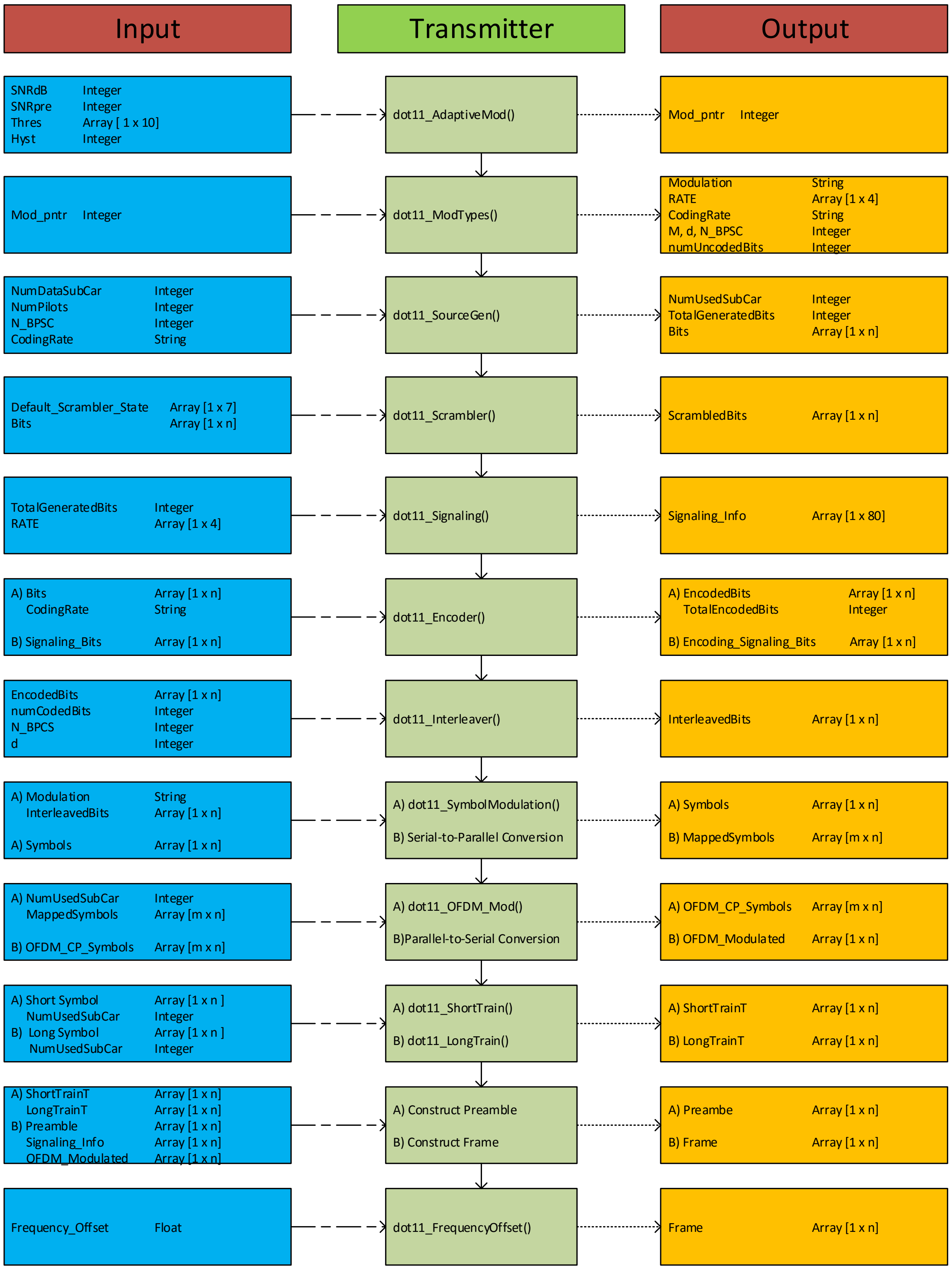


Figure 6 The transmitter software diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

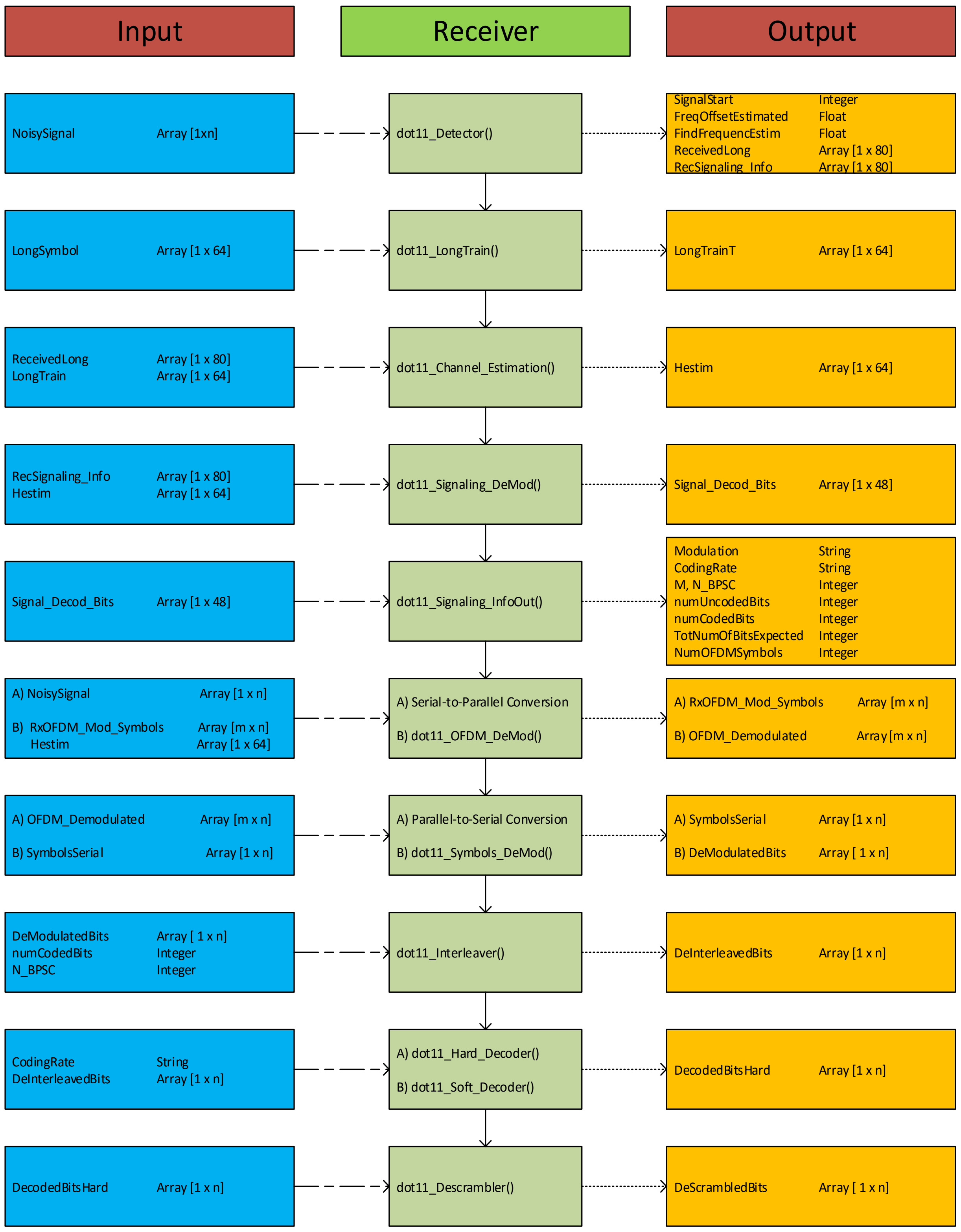

Input
Receiver
Output
NoisySignal Array [1xn]
dot11_Detector()
SignalStart Integer
FreqOffsetEstimated Float
FindFrequencEstim Float
ReceivedLong Array [1 x 80]
RecSignaling_Info Array [1 x 80]
LongSymbol Array [1 x 64]
dot11_LongTrain()
LongTrainT Array [1 x 64]
ReceivedLong Array [1 x 80]
LongTrain Array [1 x 64]
dot11_Channel_Estimation()
Hestim Array [1 x 64]
RecSignaling_Info Array [1 x 80]
Hestim Array [1 x 64]
dot11_Signaling_DeMod()
Signal_Decod_Bits Array [1 x 48]
Signal_Decod_Bits Array [1 x 48]
dot11_Signaling_InfoOut()
Modulation String
CodingRate String
M, N_BPSC Integer
numUncodedBits Integer
numCodedBits Integer
TotNumOfBitsExpected Integer
NumOFDMSymbols Integer
A) NoisySignal Array [1 x n]
B) RxOFDM_Mod_Symbols Array [m x n]
Hestim Array [1 x 64]
A) Serial-to-Parallel Conversion
B) dot11_OFDM_DeMod()
A) RxOFDM_Mod_Symbols Array [m x n]
B) OFDM_Demodulated Array [m x n]
A) OFDM_Demodulated Array [m x n]
B) SymbolsSerial Array [1 x n]
A) Parallel-to-Serial Conversion
B) dot11_Symbols_DeMod()
A) SymbolsSerial Array [1 x n]
B) DeModulatedBits Array [1 x n]
DeModulatedBits Array [1 x n]
numCodedBits Integer
N_BPSC Integer
dot11_Interleaver()
DeInterleavedBits Array [1 x n]
CodingRate String
DeInterleavedBits Array [1 x n]
A) dot11_Hard_Decoder()
B) dot11_Soft_Decoder()
DecodedBitsHard Array [1 x n]
DecodedBitsHard Array [1 x n]
dot11_Descrambler()
DeScrambledBits Array [1 x n]

Figure 7 The receiver software diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1.1.2 MAC Layer

In this subsection, a description of the developed MAC simulator is presented. The basic operation of the MAC sublayer in IEEE 802.11p is presented in Figure 8 and Figure 9. The IEEE 802.11p is a random-access protocol utilizing a carrier sense multiple access with collision avoidance (CSMA/CA) technique. Distributed radio access is implemented using the enhanced distributed channel access (EDCA) function.

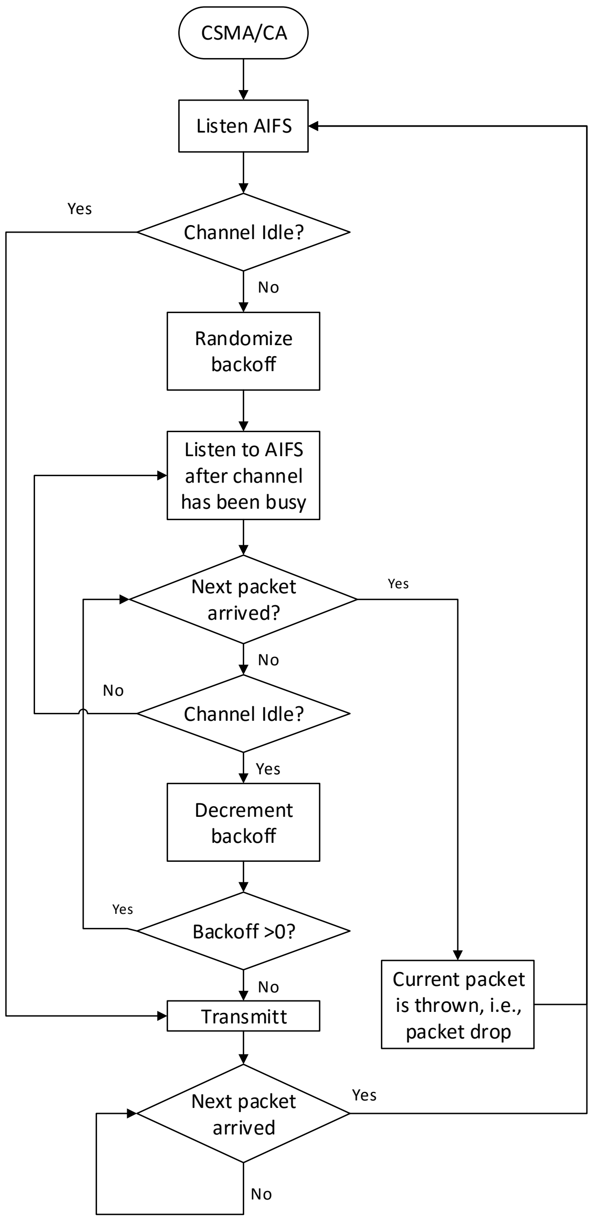


Figure 8 CSMA/CA implementation in IEEE 802.11p

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

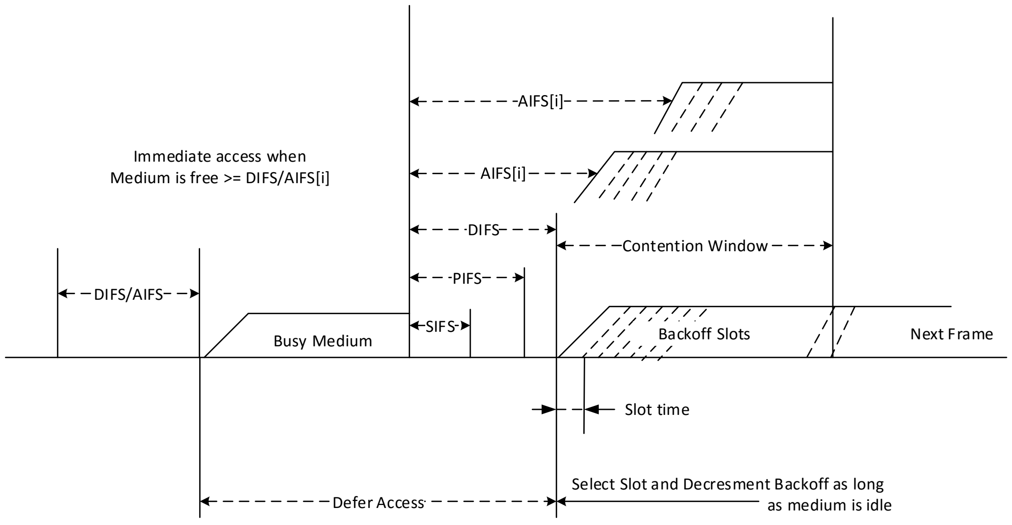

AIFS[i]
AIFS[i]
Immediate access when Medium is free >= DIFS/AIFS[i]
DIFS
Contention Window
PIFS
DIFS/AIFS
SIFS
Busy Medium
Backoff Slots
Next Frame
Slot time
Select Slot and Decresment Backoff as long as medium is idle
Defer Access

Figure 9 EDCA implementation in IEEE 802.11p

The simulator is implemented in MATLAB with an object-oriented approach. Five main classes are defined:

1. The ITSG5_MAC class that initializes global properties for the MAC layer of all network nodes
2. The ITSG5_Simulator class that implements the functionality of an ITSG5 network with multiple network nodes. The ITSG5_simulator_loop method implements the main simulator actions. The ITSG5_Simulator contains and manages the simulator clock, i.e., the simulated time line for the network operation
3. The ITSG5_Tranceiver class implements the PHY and MAC procedures per network node. Each network node in the simulator uses an instance of the ITSG5_Transceiver class. The ITSG5_Transceiver inherits properties from the ITSG5_MAC class
4. The ITSG5_Transmitter is a class-property for the ITSG5_Transceiver. ITSG5_Transmitter implements all the PHY functions and operations as described in Subsection 1.1.1 for transmitter operation. ITSG5_Transceiver controls MAC operation and assigns transmitting operation to its ITSG5_Transmitter property
5. The ITSG5_Receiver is a class-property for the ITSG5_Transceiver. ITSG5_Receiver implements all the PHY functions and operations as described in Subsection 1.1.1 for receiving operation. ITSG5_Transceiver controls MAC operation and assigns receiving operation to its ITSG5_Receiver property

Extra classes may be defined in order to embed in the simulator specific types of messages, like Content Awareness Messages (CAMs) – with the introduction of a proper ITSG5_CAM class.
During the simulator initialization stage, one ITSG5_Simulator instance is produced that performs the main network/simulator tasks. Moreover, based on the selected user generation procedure (implemented as a method in the simulator class), new network nodes are generated either in the initialization stage or continuously during the simulator loop. New network nodes are generated with new ITSG5_Transceiver instances. Each ITSG5_Transceiver instance retains as properties one

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

ITSG5_Transmitter instance and one ITSG5_Receiver instance. At all times, each ITSG5_Transmitter uses either the receiver or transmitter operation.

At this stage, and based on the description of the protocol, the following five transmission types are supported:

- Broadcast – i.e., a transceiver gains access to the medium and broadcasts a QoS data frame. No ACK is expected
- Multicast – i.e., a transceiver gains access to the medium and sends a QoS data frame to a group of users. No ACK is expected
- Unicast without ACK – i.e., the transceiver sends directly a QoS data frame to a specified destination but it does not require an ACK
- Unicast with ACK – i.e. the transceiver sends directly a QoS data frame to a specified destination and an ACK is expected as a response
- RTS-CTS Unicast with ACK – i.e. the transceiver sends an RTS (ready to send) frame towards a destination. A CTS (clear to send) response is expected. When the CTS is received, then a QoS data frame is send with an expected ACK as a response. RTS-CTS type of transmission is expected for frames with MPDU size greater than 1Kbyte

The simulator supports the following types of Frames:

- Management frames:
    - Action frames
    - Time advertisement frames
- Control Frames
    - RTS
    - CTS
    - ACK
- Data Frames
    - QoS data (since EDCA is used)
    - Null (without practical use for the simulator)

The following status are defined per network node:

0. Idle – Sensing:
1. Waiting to Tx (transmitter) – Sensing:
2. Transmitting (data or ACK)
3. Waiting to transmit ACK
4. Receiving
5. Waiting to receive ACK

In order to implement the slotted operation of CSMA/CA, the simulator implements a time line in nano-seconds. The time line is updated with the use of a “while” loop (until the end of the simulation). The time line is increased using the following rationale:

- Simulator global time increases in slot duration steps, where slot duration is the MAC slot time duration in nanoseconds. The exception in this procedure is the existence of an event at a time

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

instance less than the current slot duration. The existence of an event is specified by a number of counters retained by each network node that participates in the simulator

- Each network node (user) retains the following counters:
    - Timers that count short interframe spacing (SIFS), arbitration interframe space (AIFS), or extended interframe space (EIFS) duration. (AIFS, SIFS, and EIFS counters – AIFS counter is a 4-vector, since four QoS queues are defined by the standard)
    - Timers that implement the contention procedure for each node and each priority group of data (contention window (CW) timers)
    - Timers that count the duration of the currently transmitted packet from other sources (information acquired with demodulation of the NAV field)
    - Timer that counts the remaining time for transmission for a packet originating by the transceiver (Tx Timer)

All counters are initialized (based on an event) and continuously reduced until reaching zeros. Zeroing of a timer constitutes an event. The simulator time controller is depicted in Figure 10. The general flow of the simulator is described in Figure 11.

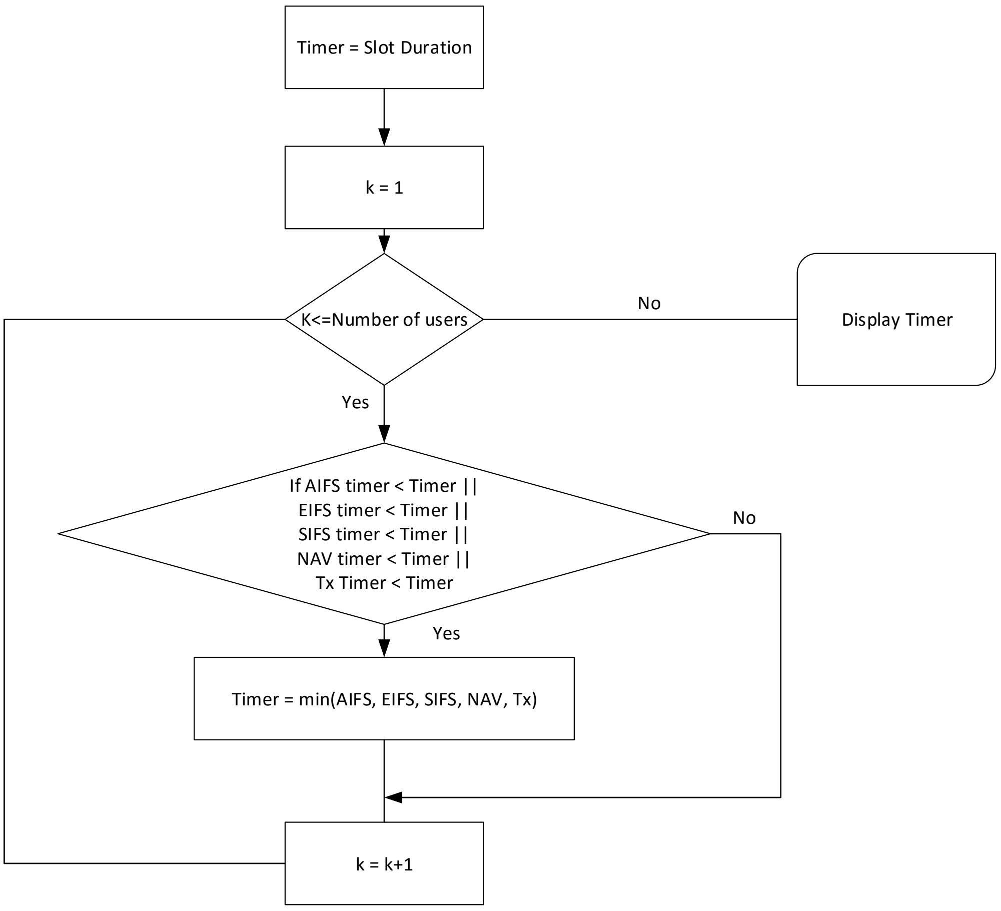


Figure 10 Simulator time controller.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1.1.2.1 Initialization of Users:

Each generated user initially has:

- No data to send
- No information about adjacent network nodes

Therefore, no a-priori knowledge is available at each transmitter.

1.1.2.2 Data Generation Procedure:

Initially, each user has no data. Based on a predefined method, new data are produced stochastically with a certain rate during each time progression step. Data are produced with a different rate for each QoS data queue of each transceiver. Moreover, the size of the currently produced data frame is stochastically determined. Therefore, the current data frame size is determined randomly between 200 bytes up to 4Kbytes. In Figure 12, the data generation procedure is depicted.

1.1.2.3 Simulator Actions per Status:

The analysis in the following paragraphs is implemented per transceiver Status.

1.1.2.3.1 Status 0:

When a node is in status 0, then:

- There are no available data into the QoS data queues to compose a full frame
- The node operates as a receiver, sensing the medium
- The node is operating as a receiver performing carrier sense.
- New data are created during each time step. When the data in one or multiple queues are enough to compose a full frame, then the node moves to Status 1.

The receiving operation produces a decision regarding the medium status. If medium status is busy, then the receiver demodulates the headers in order to:

- Update NAV counters and determine the end of the transmission
- Decide if the node is the destination for the specific frame. In this case, the node is moving to State 4 and demodulates.
- All the above are shown in Figure 13

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

-

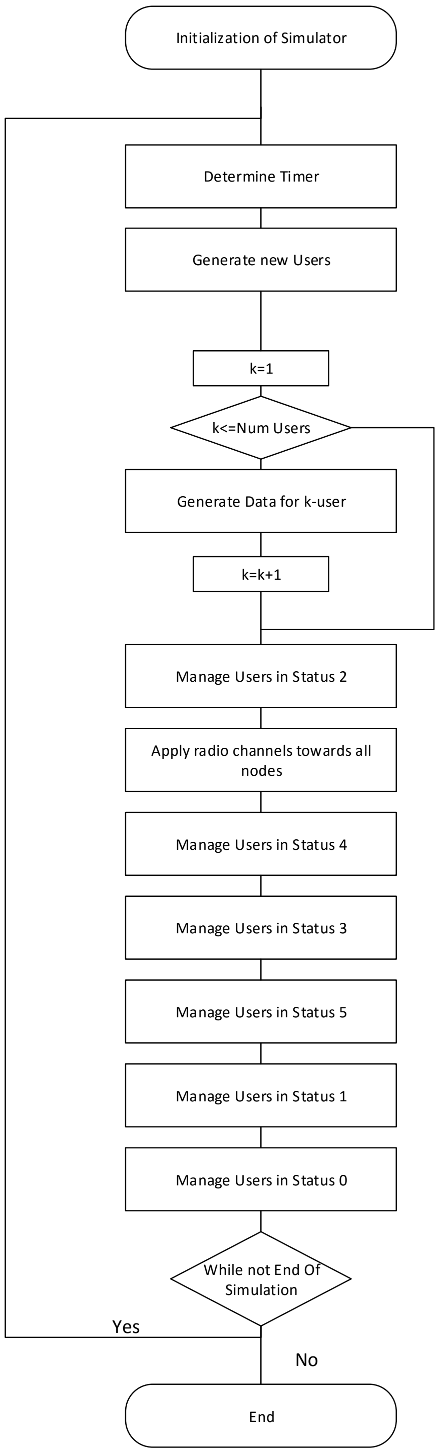


Figure 11: Basic flow of the Simulator loop.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

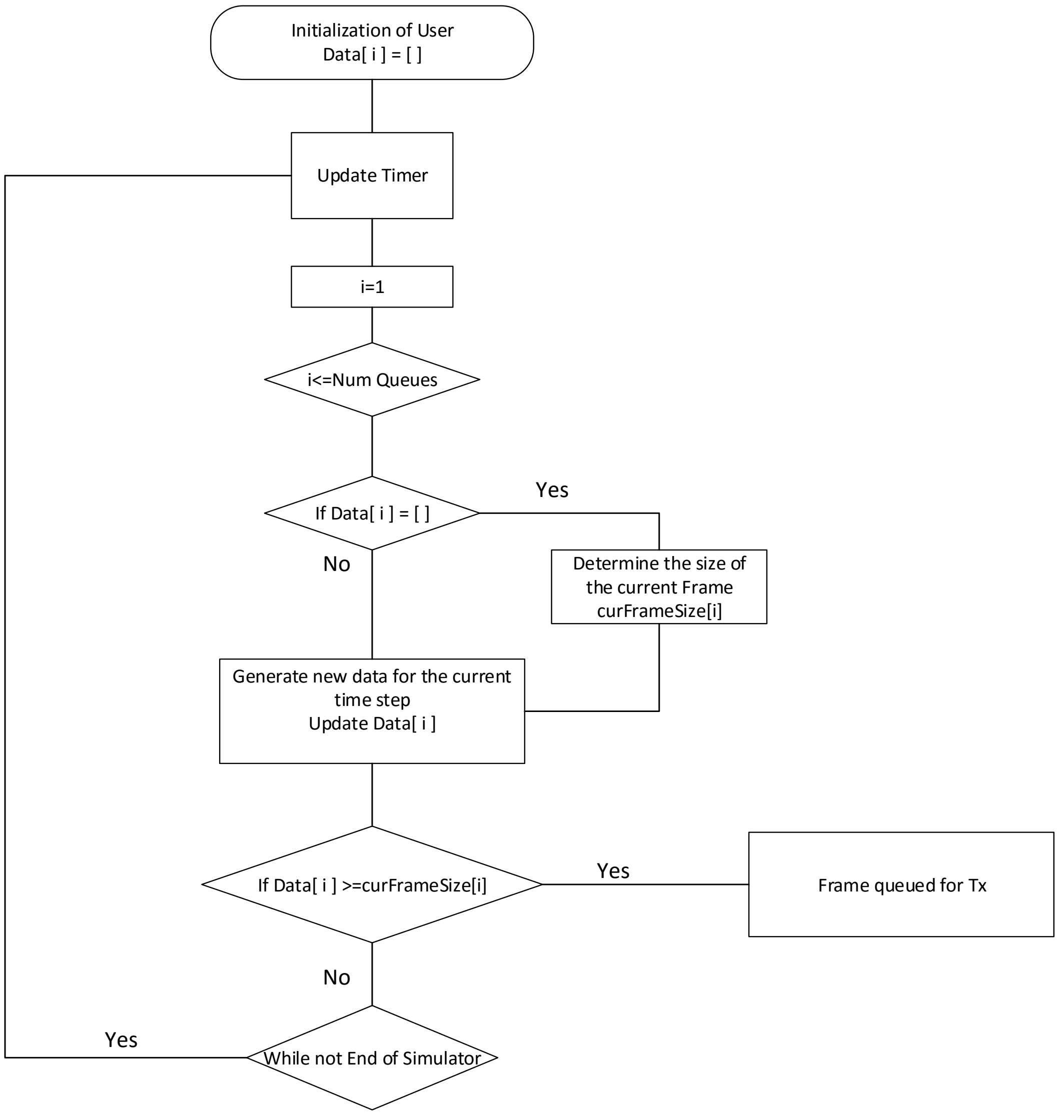

Initialization of User
Data[i] = [ ]
Update Timer
i=1
i<=Num Queues
If Data[i] = [ ]
Yes
No
Determine the size of the current Frame curFrameSize[i]
Generate new data for the current time step
Update Data[i]
If Data[i] >=curFrameSize[i]
Yes
Frame queued for Tx
No
While not End of Simulator
Yes

Figure 12 Data generation procedure

1.1.2.3.2 Status 1:

When a node is in status 1, then:

- There are available data into the QoS data queues to compose a full frame
- The node has initialized and it continuously updates
  - AIFS counters
  - CW counters (if a collision has been already sensed in previous instances)
- The node operates as a receiver, sensing the medium
- If during the sensing procedure, a signal is sensed
- The node reinitializes all AIFS counters

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The node pauses all CW counters
- It remains in State 1, and it tries to extract Destination and NAV information.

If the identified destination is the ID of the node, then the node moves to State 4 and demodulates the signal. If no signal is sensed, and AIFS and CW counters are zeroed, then the node will transmit data and it moves in State 2. If more than on AIFS/CW counter are zeroed simultaneously, then internal collision is detected. The queue with the highest priority is qualified, while Backoff procedure and AIFS counters are reinitialized for the rest of the queues.

All the above are shown in Figure 14.

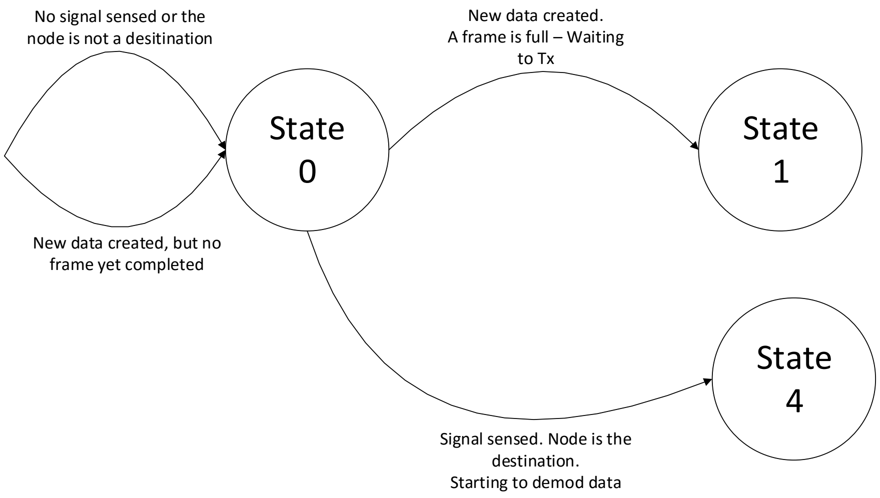

No signal sensed or the node is not a desitination
New data created. A frame is full – Waiting to Tx
State 0
State 1
New data created, but no frame yet completed
State 4
Signal sensed. Node is the destination. Starting to demod data

Figure 13 State transition for current State = 0

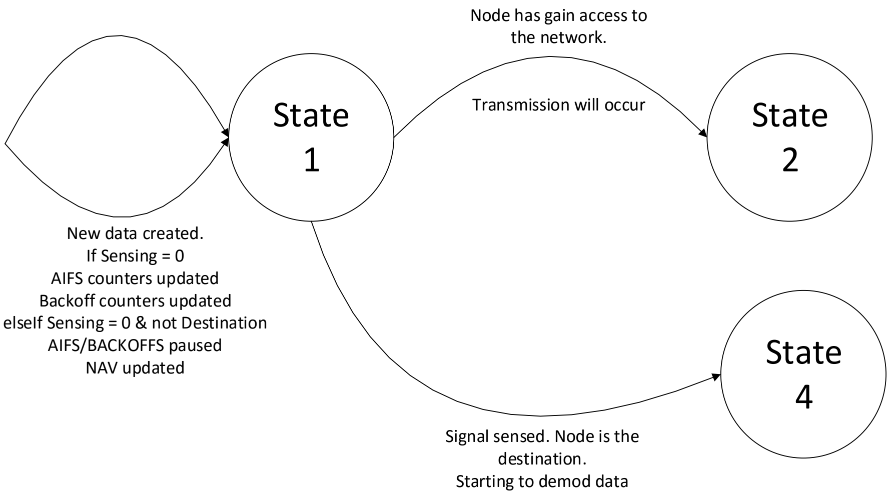

Node has gain access to the network.
Transmission will occur
State 1
State 2
New data created. If Sensing = 0 AIFS counters updated Backoff counters updated elseIf Sensing = 0 & not Destination AIFS/BACKOFFS paused NAV updated
State 4
Signal sensed. Node is the destination. Starting to demod data

Figure 14 State transition for current State = 1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1.1.2.3.3 Status 2:

When a node is in status 2, then:

- The node is in transmitter node
- The frame with the highest order from the queue that won contention is transmitted
- If during the current time period, transmission is not completed (indicated by the Tx timer), then the node remains at State 2 until completion
- If Tx timer is zeroed (i.e., transmission is completed), the basic transmission scheme is used (i.e. no ACK) and the node has more data to send then it moves to State 1. If no other data are available, then the node moves to State 0
- If Tx timer is zeroed (i.e., transmission is completed) and ACK or CTS is needed, then the node moves to State 3.

All the above are shown in Figure 15.

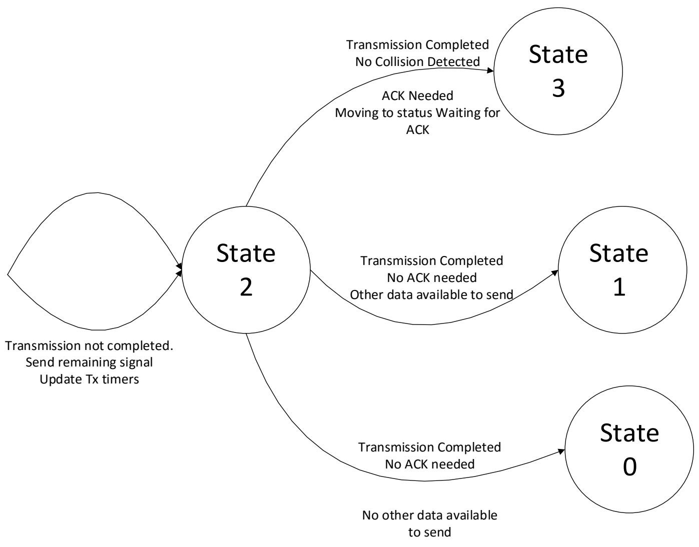


Figure 15 State transition for current State = 2

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

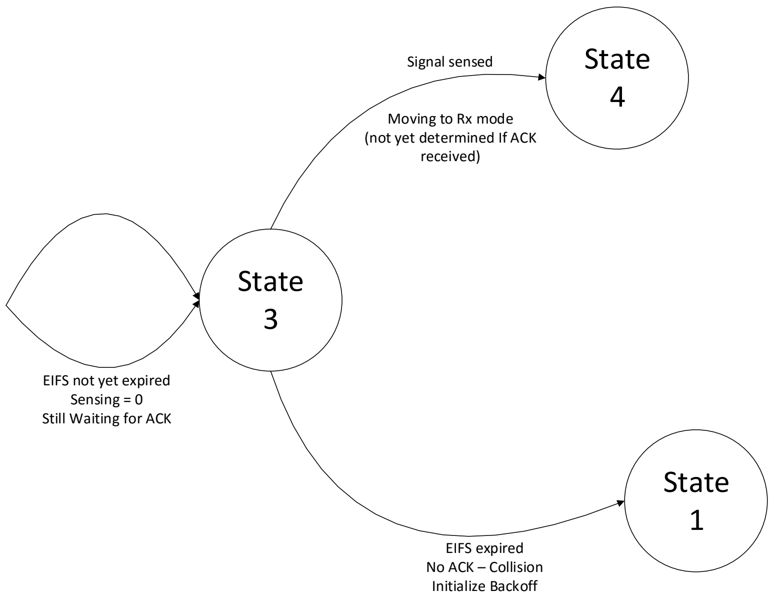


Figure 16 State transition for current State = 3

1.1.2.3.4 Status 3:

When a node is in status 3, then:

- The node is waiting to receive an ACK for a frame send during its previous state
- The node will wait for duration EIFS for ACK

During the EIFS waiting period, the medium should be determined as busy. If EIFS expires with no reception of an ACK, then the node determines that a collision occurred since no response from the destination was received. If the medium is sensed as busy, then the node moves to Receiver node. After demodulation of the received signal, the node will determine if the desired ACK was received (successful transmission) or a different signal was received (collision detected). All the above are shown in Figure 16.

1.1.2.3.5 Status 4:

When a node is in status 4, then:

- The node receives and demodulates the signal
- It is assumed that the node has identified itself as a destination of the signal

If the NAV timer for the received frame has not yet expired, then reception continues and the node remains at State 4.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

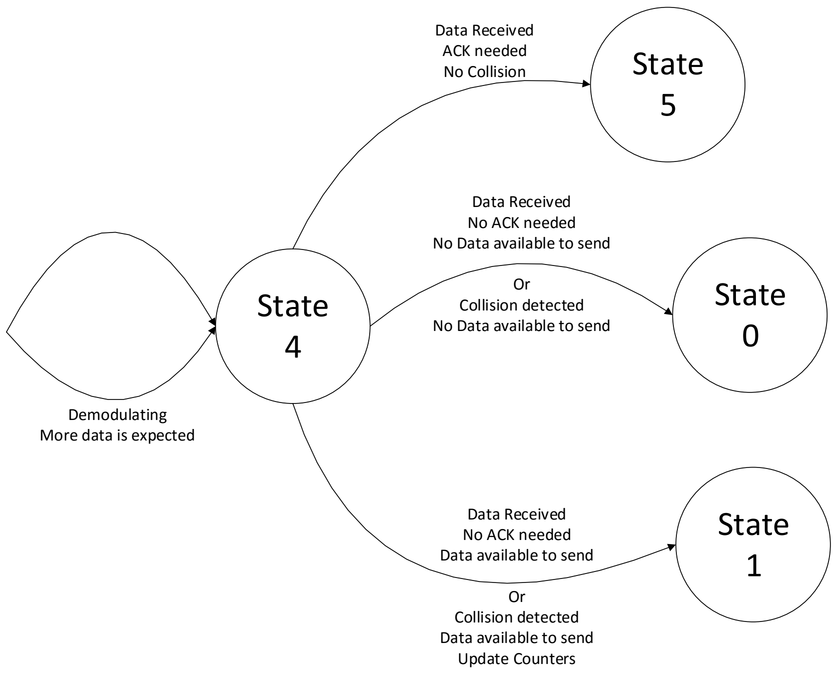


Figure 17 State transition for current State = 4

If data reception is completed then:

- If no ACK is needed, then it moves in State 0 or State 1 depending on the availability of data.
- If no ACK is needed, however CRC does not check and collision is detected, CW timers are properly updated.
- If ACK is needed and collision is detected, then the node moves in State 0 or State 1 depending on the availability of data with proper adjustment of CW timers.
- If ACK is needed and no collision is detected, then the node moves in State 5 (waiting to transmit an ACK).

All the above are shown in Figure 17.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

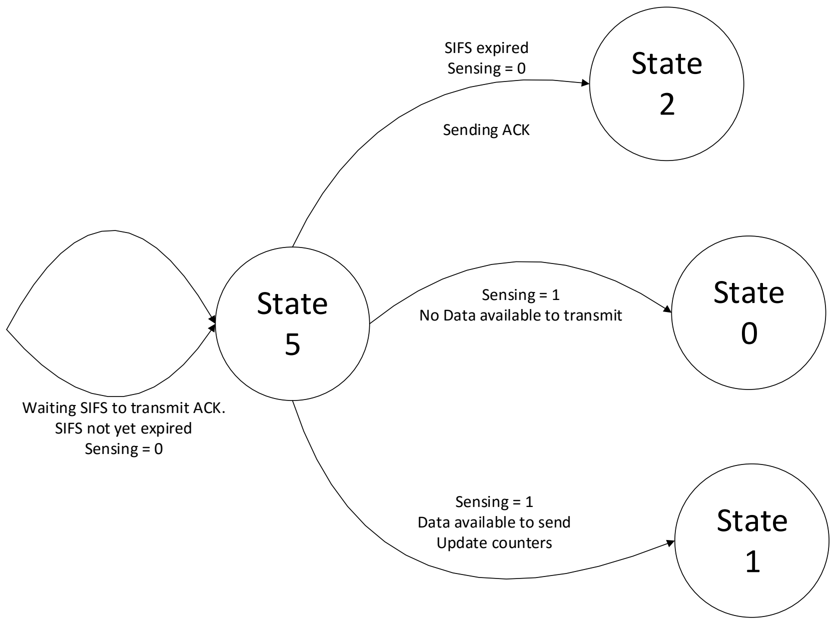


Figure 18 State transition for current State = 5

1.1.2.3.6 Status 5:

When a node is in status 5, then it waits SIFS duration and then transmits an ACK for a frame received during its previous state that needs acknowledgement. If SIFS expires and the medium is considered free, then the node moves to Transmitting Mode State 2 and it sends the ACK. If during SIFS, the medium status changes to busy, then collision is detected and the node moves either to State 0 (no data available) or State 1 (data available – with necessary CW timer adjustment). All the above are shown in Figure 18.

1.1.3 **Summary**

This subsection presented the System Level MATLAB/OCTAVE simulator for both PHY and MAC layers of the ITSG5/IEEE 802.11p standard. In particular, all the functionalities of the PHY and MAC layers have been developed, based on the latest releases of this standard. Moreover, this simulation platform was utilized to evaluate security risks and define plausibility check that are relevant for the SAFERtec project. Our experimentation results are reported in D3.3.

1.2 OpenC2X

A key component of the implemented SAFERtec simulator is the OpenC2X framework [14]. OpenC2X is an open source experimental and prototyping platform supporting ETSI ITS-G5. OpenC2X is developed in C++ and has been the core application, implementing in parallel several necessary functionalities for the simulator.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

In the following paragraph, a brief analysis of the OpenC2X functionalities, and how each functionality is implemented is presented:

Data Creation

The Data Creation functionality relies on the GPS and OBD2 services of the OpenC2X project. The OBD2 is a service that collects data from the on-board diagnostics of the assumed vehicle (through the OBD2 interface of the vehicle CAN bus). By default, the OBD2 service provides speed information solely. Since no OBD2 interconnection is generally available, the service exists only for demonstration purposes. In addition, there is the possibility to use simulated data for test and validation of operation.

On the other hand, the GPS service is configured to operate as a client of the gpsd deamon. Generally, there are two options: a) simulated GPS data for test and validation of operation, b) data from the gpsd. However, in the CV2XinFIRE VxF case, since there is no physical host, it is impossible to install a GPS device directly to the virtual machine. Therefore, the following changes were made:

- Instead of accepting data from the linux gps daemon, the program was modified to wait localization data from a Zero-MQ sender that operates on the physical host of the GPS device. In an alternative configuration, the service listens to a standard UDP port for localization data that are send by a GPS device in a pre-decided format (the format used in the original OpenC2X implementation was used.
- Since no OBD2 connection is available, the OBD2 service was fed with the speed and course angle of the GPS – identically with the previously described process.

The framework also provides the opportunity to include various types of data by defining a DENM application. In this context, a user/experimenter can define a custom app that contains various other types of information (actions, events, alarms, situations etc.). This information can be provided by calling a specific framework function (`triggerDenm)`.

ETSI ITS Stack Generator/Sender

The OpenC2X framework includes two types of ITS facilities:

- CAservice: uses GPS and OBD2 data to generate CAM ITS messages according to the ETSI standard [2].
- DENservice: uses data from DENM application to generate DENM ITS messages according to the ETSI standard [3].

The necessary parameterization of the services is made through XML configuration files.

ETSI ITS Stack Receiver

In its conventional operation, the OpenC2X framework sends data, whenever available according to the aforementioned process, and simultaneously waits for incoming messages. In the context of the project the configuration of the experiment is the following:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1. Data are generated by the aforementioned procedures and are forwarded to the desired interface according based on the MAC address set in the configuration file of the CAM or DENM process.
2. However, if instead of a MAC address, 00:00:00:00:00 is selected, then the CAM/DENM message is forwarded to the MATLAB simulator described in the previous subsection.
3. A different service, dedicated to listen for the incoming data is executed.
4. Similarly, the data from either the MAC layer of an existing network interface, or from the MATLAB simulation engine is forwarded in the CAM/DENM consumer service.

The original OpenC2X framework does not support multiple instantiations of the software in the same machine. However, in the context of the SAFERtec simulation engine, multiple network nodes are considered, thus there is a need for multiple OpenC2X instances running in parallel – one for each simulation transceiver object. Therefore, the following modifications were made by SAFERtec:

- All ports and parameters that may cause conflict were identified and customized in order to avoid crashes.
- Different IDs were assigned to the Sender process and to the Receiver process, despite the fact that both functionalities are executed into the same machine. The use of different IDs allows us to compare the content and metadata of the transmitted packets vs. the respective received packets

Local Dynamic Map (LDM)

An SQL data base is generated that stores all exchanged data from the CAM and DENM messages. The database implements the LDM facility of the ETSI ITS stack. The LDM data can be accessed by any external application with SQL queries. The database is created using SQLite framework and named as `ldm-$expNo.` The experiment number is defined in the service configuration file. The database file is in the db folder of the ldm path.

Data Routing and Mode Switching

The original OpenC2X framework supports ITS-G5 transmission through a WiFi interface. In this context, the framework assumes that the ITS-G5 modem is installed as an interface with a given MAC address. A service implementing Distributed Congestion Control (DCC) of the ITS stack assumes the responsibility to forward the ITS messages to the selected interface. In order to integrate the OpenC2X framework in the Simulation Engine, the Data Routing and Mode Switching functionality was implemented/modified as follows:

- The user can select:
    - to send the data through a specific interface of the machine.
    - to send the data through an interface, hosted by a different physical or virtual machine with a given IP address. Thus, the DCC service forwards the data towards a specific IP with a specific interface. In order to successfully implement this functionality, a packet listener should be executed in the remote host.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- to “send” the data through the SAFERtec simulation engine that is executed in the host PC (that also runs the OpenC2X framework) or to a different host with the provision of an IP address (exactly as described in the previous bullet). Practically, a UDP server that forwards the message is implemented. On the other end, the simulator should configure a UDP receiver that monitors a specific port.
- to forward the ITS messages to both physical interfaces and simulators for the performance of parallel tests.

Measurement, benchmarking and KPI extraction

The specific functionality creates listeners with the following roles:

1. To accept the message id, message type and the timestamp of the transmitted CAM and DENM messages of the OpenC2X framework.
2. To accept the message id, message type and the timestamp of the received CAM and DENM messages of the ETSI ITS Receiver of the OpenC2X.
3. To accept:
   - The number of bit errors during the last identified packet reception.
   - The SNR and the EVM for the last received packet as estimated by the receiver.
   - The timestamp that indicates the end of subframe reception (at the MAC sublayer of the protocol).
   - The number of bits carried by the demodulated and decoded subframe (payload).
   - The id of the receiving modem

   The information is sent by the receiver instance that has just received an incoming subframe.

The specific functionality is not used strictly for data and measurement collection, but it includes post-processing tasks on the received information. More specifically:

- The functionality cross-checks (using message IDs) if a CAM/DENM message has arrived successfully at the destination. If the message was received successfully, the latency in the application layer is calculated using the timestamps.
- The functionality calculates the bit error rate. There is a cumulative value, where the overall number of errors to the number of received bits is calculated for the whole duration of the experiment, as well as the “instantaneous” value where the X (default is X=10) latest broadcast channels are used to estimate a Bit Error Rate value.
- The functionality calculates the instantaneous throughput by calculating the number of received bits per second of receiver operation.
- The functionality indirectly calculates the packet errors (and the packet error rate).

The extracted KPIs are stored into an InfluxDB database called statsdemo which is accessible at port 8086. InfluxDB is a time series database specialized for monitoring metrics and events, thus it is suitable for KPI storage and manipulation. Data are also available in the MATLAB simulator variables.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

Result presentation Layer

OpenC2X hosts an application web service with the objective to provide a user-friendly, easy and efficient way to monitor the experiment results in real time. More specifically, OpenC2X creates and manages a web service, that outputs the ego and incoming messages of the ITS stack and presents in an on-line map the location of the vehicles participating in the ITS constellation. Thus, the simulation results can be represented on real maps, if configured properly. In order to do that, the OpenC2X framework allows the user to import a specific route (a set of coordinates) for each OpenC2X instance, thus, using real coordinates of real-world routes recorded with the use of GPS device (or with the use of a gps-fake service). This allows us to implement a more realistic simulation environment using real-world traffic information.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

A3: Emulation and Fuzzing

In order to speed up the vulnerability discovery process, auditors have decided to emulate the VEHICLE_*COMMUNICATION_APP*. Even if debugger tools were installed on the device, they were not so easy to use. Moreover they were limited in the amount of breakpoint/watch point auditors can use, so it was difficult to patch and control program execution flow.

Emulation of VEHICLE_COMMUNICATION_APP

- qemu-user

As previously mentioned, the CPU architecture is ARM, which is now largely supported by many emulators available on the market. Auditors have chosen to emulate the *VEHICLE_COMMUNICATION_APP* with qemu.

Their first attempt was to emulate the entire program with qemu-user. To achieve this goal, they built a framework that allows them to patch the *VEHICLE_COMMUNICATION_APP* (Ex: hardware access). A configuration file indicates what and where to patch.

In their experimentations, they successfully started *VEHICLE_COMMUNICATION_APP* in an emulated context, but they faced many problems:

- Their patching framework was very basic. They were able to modify some parameters (like: return code of a function, variable's data, etc..), but not to do some advanced patches, like memory leak detection. Unfortunately, the estimated time to upgrade it was not compatible with the duration of the project.
- To have an emulation execution closer to the real device execution, they needed to understand precisely how the hardware works. This task was very difficult and time consuming as they did it with no information/documentation on V2X OBU. For example, to emulate just the GPS hardware access, they had to debug the device, understand the driver, and report it in the emulation. This type of work had been repeated for every interface.
- They must place their emulate application in specific context (Ex: GPS sync, RSU peering, Network IP peering, etc…). Also, this work is very hard, as they need to replay and emulate all peering operation of external devices.

For all this reasons they finally gave up to emulate the *VEHICLE_COMMUNICATION_APP* with classic qemu-user way.

- Unicorn-qemu

For the instrumentation aspect, they have chosen to use *unicorn-qemu*. *Unicorn-qemu* is based on *qemu-user* but allows making specific instrumentations for each instruction executed and each memory byte access.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

However, they faced some troubles:

- Emulation execution with *unicorn-qemu* was very slow (nearly 50 times more slowly than emulation with qemu-user and around 200 times more slowly than the real execution)
- They had to emulate hardware access
- *Unicorn-qemu* allowed them to emulate execution instruction by instruction, so they had to implement by themselves OS mechanisms like scheduling (*VEHICLE_COMMUNICATION_APP* launches many threads) and system calls.

In order to fuzz *VEHICLE_COMMUNICATION_APP* 802.11p stack, they made a snapshot of the process on 802.11p low level frame reception. Then, they used Python to develop hardware stub, instrumentation (Tracing tool, memory check and so on) and OS mechanisms. Finally they used *AIRBUS CyberRange* to run the fuzzing campaigns.

After few days of fuzzing, they didn't have significant results as the emulation was too slow. So, they decided to make some optimizations to improve the emulation speed.

They took the decision to stop trying emulating the full *VEHICLE_COMMUNICATION_APP* process on each emulation execution. As mentioned in paragraph V2X OBU Architecture overview, *VEHICLE_COMMUNICATION_APP* starts many threads, each one in charge of only one specific task.

So they decided to run multiple AFL instances on each thread, considered as interesting for vulnerability research. As a result, AFL executed fewer instructions during an execution and the emulation became more efficient.

Blind fuzzing

Auditors wanted to do some fuzzing tests to discover new vulnerabilities that could be difficult to find by static analysis. There are many fuzzing ways: the first one is called "blind fuzzing" and consists of sending random data to the physical equipment and waiting for abnormal behaviour (for example, a crash, a slow response, a configuration change and so on).

In order to make this "blind fuzzing" tests efficient, auditors needed:

- To develop an instrumentation to detect abnormal behaviour;
- To develop a binary instrumentation to detect bugs like memory corruption;
- To save all the frames sent in order to replay them (for example, to investigate a potential crash);
- A controlled environment (Faraday cage), in order to avoid 802.11p frames alterations;
- Some reboot proceedings in order to put the device in the desired state (peering with RSU, peering with GPS…);

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Moreover, to be efficient, “blind fuzzing” needs important reverse engineering task. To maximise chance to find vulnerability, auditors must indeed understand the format of data that the application is waiting for like size field, function code field, CRC... Otherwise, there are no chances to find potential vulnerabilities.

Fuzzing with code coverage

A second approach which has been used to fuzz is call “Fuzzing with code coverage”. The idea is to drive the fuzzing engine in order to execute all the code. In practice, it notices the fuzzing engine when it generates a sample that discovers a new execution path.

Originally, this approach was only available to fuzz source code (code coverage instrumentation is added at the compilation time). But in SAFERtec, auditors decided to emulate the VEHICLE_COMMUNICATION_APP, in order to instrument the emulator to notice the fuzzing engine when it executes a new path. So code coverage fuzzing can be used in the project.

With this approach auditors have been able to do “smart fuzzing“ and to expect being more efficient to find vulnerabilities. The advantages are:

- Restoration of desired state is fast and easiest;
- Fuzzing speed depends of processing capabilities;
- Binary instrumentation can be made at emulation level, so it’s easier to implement;
- No more need to be in controlled environment;

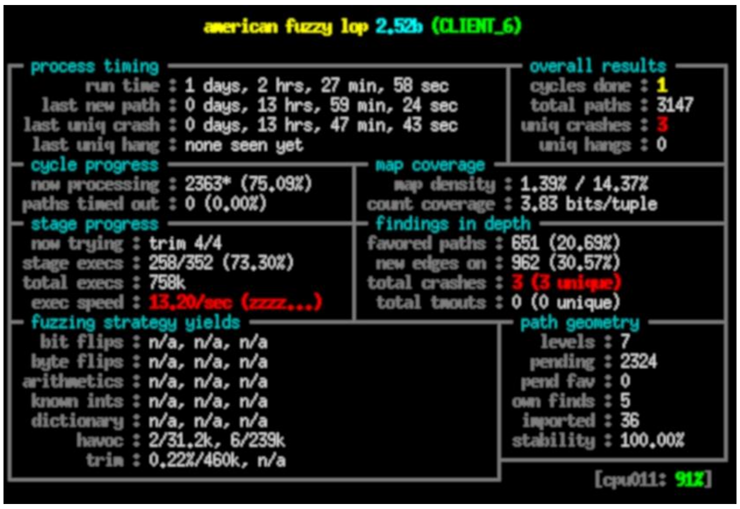
But this approach needs:

- To be able to emulate the code;
- Develop some stubs (Hardware access, send/receive functions and so on);

These steps can be very time consuming. Also, auditors take care of implementing stubs in order to have an emulation very close to the real execution. Indeed, when potential vulnerabilities are found through emulation and fuzzing, auditors must check if they are false positive by replaying it on the real system. This step can be a little time consuming because auditors have to check if the execution path in emulation and real system are the same.

Results

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319



```
                  american fuzzy lop 2.52b (CLIENT_6)

- process timing --------------------------------------- overall results --
       run time : 1 days, 2 hrs, 27 min, 58 sec        |  cycles done : 1
  last new path : 0 days, 13 hrs, 59 min, 24 sec       |  total paths : 3147
last uniq crash : 0 days, 13 hrs, 47 min, 43 sec       | uniq crashes : 3
 last uniq hang : none seen yet                        |   uniq hangs : 0
- cycle progress ----------------- map coverage ----------------------------
 now processing : 2363* (75.09%)  |    map density : 1.39% / 14.37%
paths timed out : 0 (0.00%)       | count coverage : 3.83 bits/tuple
- stage progress ----------------- findings in depth -----------------------
 now trying : trim 4/4            | favored paths : 651 (20.69%)
stage execs : 258/352 (73.30%)    |  new edges on : 962 (30.57%)
total execs : 758k                | total crashes : 3 (3 unique)
 exec speed : 13.20/sec (zzzz...) |  total tmouts : 0 (0 unique)
- fuzzing strategy yields --------------------------- path geometry -------
  bit flips : n/a, n/a, n/a                          |    levels : 7
 byte flips : n/a, n/a, n/a                          |   pending : 2324
arithmetics : n/a, n/a, n/a                          |  pend fav : 0
 known ints : n/a, n/a, n/a                          | own finds : 5
 dictionary : n/a, n/a, n/a                          |  imported : 36
      havoc : 2/31.2k, 6/239k                        | stability : 100.00%
       trim : 0.22%/460k, n/a                        ---------------------
                                                             [cpu011: 91%]
```

Figure 19: Fuzzing with AFL and Unicorn emulation

Once the fuzzing development became stable, they ran the fuzzing campaign during 10 days. They found several anomalies, which need to be confirmed as execution can be different on real system (this is mainly due to hardware stub implementations). So for each potential vulnerability identified during the fuzzing campaign, auditors had to replay it on the V2X OBU. If the attack did not affect the V2X OBU as expected, auditors had to understand why to improve the next emulation execution.

Discovered vulnerabilities and bugs have been presented in the deliverable; however the fuzzer has also found new issues, which have not been confirmed at the time of writing the report on the real component.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319