ENSEMBLE

EUROPEAN COMMISSION

HORIZON 2020
H2020-ART-2016-2017/H2020-ART-2017-Two-Stages
GA No. 769115

ENSEMBLE

ENabling **S**af**E** **M**ulti-**B**rand p**L**atooning for **E**urope

Deliverable No.	D2.8	
Deliverable Title	Platooning protocol definition and communication strategy	
Dissemination level	Public	
Written By	Boris Atanassow, Katrin Sjberg, Marcus Larsson, Volvo AB Edoardo Mascalchi, CLEPA Joost van Doorn, NXP Arne Ehlers, Tobias Werle, ZF Roman Alieiev, MAN	03-03-2020

	Antoine Schmeitz, Dehlia Willemsen, TNO	
Checked by	Edoardo Mascalchi, CLEPA	28-02-2022
Approved by	Marika Hoedemaeker, TNO	03-03-2022
Status	APPROVED BY EC	13-10-2022

Please refer to this document as:

Atanassow B., Sjöberg K. et all (2022). *Platooning protocol definition and communication strategy.* D2.8 of H2020 project ENSEMBLE, (www.platooningensemble.eu)

Disclaimer:

ENSEMBLE is co-funded by the European Commission, DG Research and Innovation, in the HORIZON 2020 Programme. The contents of this publication is the sole responsibility of the project partners involved in the present activity and do not necessarily represent the view of the European Commission and its services nor of any of the other consortium partners.

Revision history

Version	Date	Author	Summary of changes	Status
0.1	03/03/2020	Boris Atanassow (Volvo AB)	Initial input starting from previous version of the deliverable	Prepared
0.2	12/03/2020	Boris Atanassow (Volvo AB)	Update of platooning protocol sequences	Prepared
0.3	04/02/2021	Boris Atanassow (Volvo AB)	Restructuring and formatting done.	Prepared
0.4	05/02/2021	Boris Atanassow (Volvo AB)	Added solved issues from WP3 issue list.	Prepared
0.5	12/02/2021	Boris Atanassow (Volvo)	Updated sequences, comments removed.	Prepared
0.6	19/05/2021	Boris Atanassow (Volvo)	Comments from issue list included.	Prepared
0.7	09/11/2021	VOLVO/CLEPA	Feedback by partners implemented	Draft
1.0	22/12/2021	VOLVO/CLEPA	Ready for WP Leader review	Final
1.1	31/01/2022	CLEPA	Ready for Coordinator review	Final
1.2	28/02/2022	CLEPA	Feedbacks from Coordinator implemented	Final
2.0	03/03/2022	TNO	Approved	Submitted

TABLE OF CONTENTS

FIGURES

TABLES

EXECUTIVE SUMMARY

Context

Platooning technology has made significant advances in the last decade, but to achieve the next step towards deployment of truck platooning, an integral multi-brand approach is required. Aiming for Europe-wide deployment of platooning, 'multi-brand' solutions are paramount. It is the ambition of ENSEMBLE to realize pre-standards for interoperability between trucks, platoons and logistics solution providers, to speed up actual market pick-up of (sub)system development and implementation and to enable harmonization of legal frameworks in the member states.

Project scope

The main goal of the ENSEMBLE project is to pave the way for the adoption of multi-brand truck platooning in Europe to improve traffic safety, fuel economy, and throughput. This has been demonstrated by driving up to seven differently branded trucks in one (or more) platoon(s) under real world traffic conditions. During the years, the project was organised as follows:

- Year 1: setting the specifications and developing a reference design;
- Year 2 and 3: implementing this reference design on the OEM own trucks, as well as performing impact assessments with several criteria;
- Year 4: focus on testing the multi-brand platoons on test tracks and public road.

The technical results were evaluated against the initial requirements, after which these were updated. Also, the impact on fuel consumption, drivers and other road users was established. In the end, all activities within the project aim to accelerate the deployment of multi-brand truck platooning in Europe.

Platooning levels

Two levels of platooning have been defined:

- **Platooning Support Function**: the driver is responsible for the driving task. Hence (s)he is also responsible to choose a safe following distance and monitor the system e.g. whether the right platooning partner is being followed (though supported by the system as much as possible). To give the driver sufficient time to react, minimum time gaps around 1.5 s have to be respected. The Platooning support function is a longitudinal control function, but lateral driver assistance systems, such as e.g. lane keeping, might be optionally available as well.

- **Platooning Autonomous Function**: The lead truck has a driver responsible for the driving task, but the following trucks are fully automated, i.e. the system performs the complete driving task within the specified (limited) operational design domain. Taking the driver(s) out-of-the-loop offers the possibility to reduce time gaps to a minimum of 0.3 s.

In contrast to the Platooning Support Function, implementation of the Platooning Autonomous Function is not part of the ENSEMBLE project and the specification of the Platooning Autonomous Function and its use cases is solely done on theoretical considerations to sketch a future vision of platooning. The latter is also due to the low technology readiness level of certain required autonomous driving subfunctions at the time of writing.

For the interest of the reader, the main documents that describe the two platooning levels defined in ENSEMBLE are:

- Levels definitions and Use Cases – D2.3 [27]
- Requirements and Specifications - D2.5 [28]

Additional details on the Communication protocol and the strategic and services layers can be also found in:

- V2X Protocol - D2.8 (this deliverable)
- Security - D2.9 [26]
- Intelligent infrastructure - Strategic and Services Layers – D2.6 [29] and D2.7 [30]

Furthermore, the deliverable related to the safety analysis performed on the two levels are:

- Safety of the intended functionality (SOTIF) - D2.13 [31]
- Functional Safety - D2.14 [32]
- Item Definition - D2.15 [33]

Abstract of this Deliverable

This deliverable provides a specification of the V2X communication protocol to enable platooning using the wireless communication standard ITS-G5. The protocol specifies how vehicles inform each other about their ability to form a platoon, and which data has to be exchanged when executing the actual procedure of joining, driving in the platoon and leaving a platoon. It includes messages and how to exchange them to allow for the aforementioned situations.

In ENSEMBLE, a new facilities layer protocol supporting the platooning application is developed. This makes use of already standardized lower layer protocols in ETSI TC ITS. The platooning protocol uses already available message types and signals, and where necessary new ones are introduced. The protocol logic for joining, platooning, and leaving has been derived from the use cases in deliverable D2.3 [27] and the requirements and specifications in deliverable D2.5 [28] of ENSEMBLE. The available security framework for cooperative intelligent transport system (C-ITS) in Europe is used for signing and verifying unencrypted messages to establish a trust domain. In addition, the security is extended with encryption of platoon application data. Deliverable 2.9 will specifically treat security in platooning. Deliverable D2.5 contains the lessons learned and future considerations on the communication protocol from the final testing of the seven-brand truck platooning in Spain taking place in September 2021. Deliverable D6.15 [34] will also provide guidance for the upcoming standardization on platooning.

1. INTRODUCTION

1.1. Background

Cooperative Intelligent Transport Systems (C-ITS) refers to applications using wireless communication between vehicles, vehicle-to-vehicle communication (V2V), and between vehicles and smart road infrastructure, vehicle-to-smart road infrastructure communication (V2I), for increasing road traffic safety and efficiency. V2V and V2I communications are collectively known as V2X communication. Present document specifies a facilities layer protocol for supporting truck platooning using the wireless technology ITS-G5 (a.k.a. IEEE 802.11p [9]/WLANp) at the 5.9 GHz band.

Direct communication between vehicles and between vehicles and smart infrastructure has the potential to save lives and reduce the environmental impact. Frequency bands for V2X were allocated in 2008 in Europe and already in 1999 in the US at a carrier frequency of 5.9 GHz. In Europe, standardization has been carried out in the EC acknowledged standards development organization (SDO) ETSI[1] and its Technical Committee on Intelligent Transport Systems (TC ITS). Pre-standardization and deployment issues are treated in CAR 2 CAR Communication Consortium[2] (C2C-CC), a non-profit organization collecting OEMs, suppliers, universities and research institutes. More information about ETSI's protocols and deployment plans are found in [1,2], respectively. Standards are necessary to create an interoperable system between different brands.

SAE[3] and IEEE[4] have created an interoperable V2X system in the US. SAE has focused on message sets for V2X and IEEE has developed all lower layer protocols. Crash Avoidance Metric Partnership (CAMP) has collected OEMs and CAMP has run several public funded research projects and conducted pre-standardization tasks. The wireless technology (IEEE 802.11p) is used both in Europe and in the US. An overview of the protocol stack in the US is found in [3].

Focus on standardization has been to increase the awareness horizon for the driver by alerting the driver about impending dangerous situations and then the driver needs to take appropriate action (no automated control of the vehicle based on received V2X data). Several so-called day-one applications (or services) have been defined such as stationary vehicle warning, slow vehicle warning, emergency electronic brake light etc., by C2C-CC and further elaborated in the Commission work "C-ITS deployment platform" [16]. These day-one services are using two distinct facilities layer protocols developed by ETSI TC ITS called Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Message (DENM), where the former are always present

[1] European Telecomunications Standards Institute, see http://www.etsi.org/
[2] CAR 2 CAR Communication Consortium, https://www.car-2-car.org/
[3] Society of Automotive Engineers, see http://www.sae.org/
[4] Institute of Electrical and Electronics Engineers, see https://www.ieee.org/

triggered by vehicle dynamics containing information about the vehicle such as type, speed, position and heading. DENMs are only triggered on behalf of a dangerous situation and contain information about the dangerous event itself. The V2X communication is closing the gap between line-of-sight (LOS) sensors such as camera, lidar and radar, and the long-range cellular technology, by providing the possibility to see beyond physical barriers within milliseconds.

Platooning and cooperative adaptive cruise control (C-ACC) are enabled through the transmission of V2X data and they are regarded as safety applications as well as efficiency applications. C-ACC can mitigate shockwaves through traffic and thereby, avoid rear-end collisions but at the same time increase the number of vehicles on the roads without increasing congestion. C-ACC is a distributed application using longitudinal information contained in V2X packets whereas platooning is a closed-loop application between well-known participants which can use both longitudinal as well lateral V2X data for operation.

Platooning can make today's spontaneous platooning safer (trucks are already today driving too close without help from technology, violating regulation and safety) and support the driver in the monotonous task of driving in a highway environment by alerting the driver about impending hazardous events. The first truck in a platoon sees further ahead using conventional line-of-sight (LOS) technologies (radar and camera), and when the first truck detects any anomalies it will inform the other trucks in the platoon facilitating orchestrated braking for example. Regardless of distances between the trucks, a truck using only conventional sensors cannot see beyond physical barriers, by adding the V2X component the driving of trucks will be made safer since the first truck can inform other trucks behind it about dangerous situations. And of course, from a fuel economy perspective less jerky driving and reduced air drag due to decreased distances between the trucks will reduce the environmental impact due to fuel consumption reduction.

1.2. Purpose

The purpose of this deliverable is to define a facilities layer protocol supporting platooning. The protocol consists of logic, new message and data types along with added security (encryption of messages) to enable a pan-European multi-brand platooning system.

1.3. Scope

This deliverable describes the protocol logic, message sets, data formats and security, for enabling platooning on public roads using IEEE 802.11p/ITS-G5 communication on a carrier frequency of 5.9 GHz. (This deliverable does not address cellular communication for accessing a back-office system.)

1.4. Outline

Chapter 2 provides information about already standardized protocols that will be used by the ENSEMBLE project. Further, in Chapter 3, detailed parameter settings of the standardized protocols are provided. The ENSEMBLE platooning protocol is detailed in Chapter 4 and a summary is outlined in Chapter 5. References are provided in Chapter 6. At the end of the deliverable two appendices outlining ASN.1 structure of messages and abbreviations are provided.

2. V2X STANDARDIZED PROTOCOL STACK

2.1. Overview of already standardized protocols

Protocols have been developed in ETSI TC ITS[5] to achieve interoperability between different brands supporting day-one applications. Protocols are organized into layers to break down the complexity of communication, i.e., protocol stacks. Two distinct protocols in the facilities layer support traffic safety, the ubiquitous position messages (called cooperative awareness message, CAM) and the event-triggered hazard warnings, which are only present as long as the event is valid (called decentralized environmental notification message, DENM). Triggering conditions for DENMs have been developed by C2C-CC. In Figure 1, the C-ITS protocol stack is depicted. The platooning protocol developed in ENSEMBLE is situated at the facilities layer, see Figure 1, and it shares the same lower layer protocols as for transmitting CAMs and DENMs. The platooning capability of a vehicle will in the ENSEMBLE project be announced in existing CAMs by extending the CAM with a platooning container.

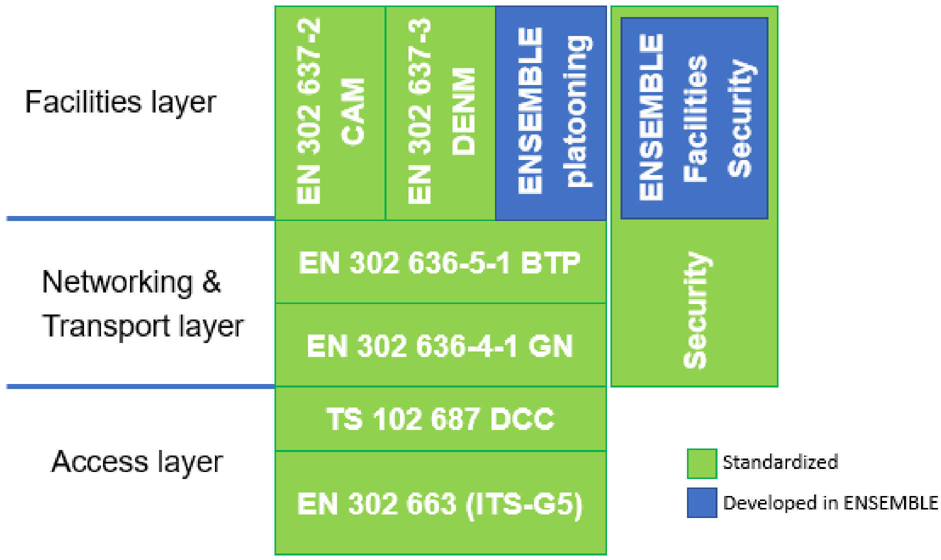

Figure 1: Standardized C-ITS protocol stack and the ENSEMBLE protocol

All dissemination of CAMs and DENMs is broadcast communication, implying that no session or similar is established between communicating parties. Broadcast communication has been selected because no sessions can be created due to highly dynamic networks and vehicles can move in and out of each other communication ranges quickly. However, nothing prevents from having session-

[5] European Telecommunications Standards Institute (www.etsi.org), consists of several technical committees (TC) treating communication in different areas, TC on Intelligent Transport Systems (TC ITS), was established in December 2007, to address the interoperability between vehicles and vehicles and smart infrastructure. EC issued mandate M/453 to accelerate the standardization process in 2008.

based applications such as platooning. Focus on standardization of the C-ITS protocol stack has been on broadcasting CAMs and DENMs, and it is up to every OEM to digest these and display appropriate information to the driver (i.e., the driver is responsible for taking appropriate action based on displayed warnings). In other words, C-ITS day-one applications aim at extending the information horizon of the driver and it is on the receiving side where competition between brands will take place. All approved ETSI standards are available for free download on the Internet[6].

The facilities layer protocols CAM and DENM are mainly executed on the vehicle but also other protocols only residing on smart infrastructure are also possible to support such as MAP, which transmits the outline of an intersection, and SPAT (Signal, Phase and Timing), which informs vehicles approaching red lights about the next green phase. Day-one services, based on the facilities layer protocols CAM, DENM, MAP and SPAT, and platooning will co-exist in C-ITS. This implies that vehicles capable of platooning will also implement and receive information about upcoming dangerous situations, DENM reception, (stationary vehicle warning, slow vehicle warning, road works warning etc.) and next green phase at traffic signals (green light optimal speed advisory). Thus, platooning enabled trucks will also do day-one services. The ENSEMBLE project has implemented CAM and the platooning protocol.

Security is treated in Deliverable 2.9. In short, the platooning protocol is part of the public key infrastructure (PKI) developed for C-ITS where messages are signed and verified using a temporarily authorization ticket. In addition, platooning data will also be encrypted.

The ENSEMBLE platooning protocol is specified in Chapter 4 in this deliverable and it will support the platooning application, which is fusing information from several sources to control the vehicle in a safe way in the platoon.

In the following subchapters, already standardized protocols used in the ENSEMBLE project are briefly outlined. Chapter 3 provides a parameter setting of the standardized protocol stack.

2.2. EN 302 663 access layer technology ITS-G5

EN 302 663 V1.3.1 [4] defines the access layer technology (access layer consists of the physical and datalink layer in the OSI model). The access layer technology outlined in EN 302 663 is IEEE 802.11p[7], which defines the physical layer and the sublayer medium access control (MAC) together with the logical link control (LLC) protocol, IEEE 802.2. LLC provides means of differentiating between network layer protocols (e.g., GeoNetworking, internet protocol, IP).

[6] ETSI Search & Download of standards, https://www.etsi.org/standards-search

[7] IEEE 802.11p is superseded and it is no longer a standalone amendment to IEEE 802.11. It was enrolled in the legacy standard IEEE Std 802.11-2012. But for simplicity the notion of IEEE 802.11p will be used throughout the document. An IEEE 802.11p equipped vehicle operates outside the context of a basic service set (BSS) and this is enabled through `dot11OCBActivated` set to TRUE.

For the interested reader, Annex C of EN 302 663 V1.3.1 [4] contains a description of IEEE 802.11p and how it differs from traditional WiFi networking. Further, it describes the MAC procedure for broadcast and the 4 different priorities (queues) provided for Quality of Service (QoS).

EN 302 663 V1.2.1 [4] refers normatively to TS 102 687 V1.1.1 [6], addressing decentralized congestion control (DCC), and TS 102 792 V1.1.1 [7] addressing co-existence with CEN DSRC.

2.3. TS 102 687 DCC

TS 102 687 V1.2.1 [6] treats decentralized congestion control (DCC) with the aim of controlling the network load in situations when there are many ITS stations within radio range wanting access to the shared frequency channel. DCC is not a necessity to achieve interoperability between different brands but merely to specify a common behaviour once the channel busy ratio (CBR) will increase when many ITS stations in the same geographical area want to access the channel. The purpose of the DCC is to have graceful and predictable performance degradation of ITS applications at large when the CBR increases. This is achieved by each ITS station executing suitable DCC algorithms. TS 102 687 (V1.2.1) provides two different algorithms – adaptive and reactive – where the former was introduced in version V1.2.1 of TS 103 687.

The adaptive approach was introduced to combat drawbacks with the reactive approach, where ITS stations experience fluctuating number of transmission opportunities from time to time when the DCC algorithm is executed. The oscillating behaviour has been confirmed by several research articles. The adaptive approach is a closed loop where the CBR value is fed back to the controller. This in comparison to the reactive being an open-loop control.

2.4. EN 302 636-4-1 GN

The GeoNetworking (GN) protocol resides in the networking and transport layer. It supports four different communication scenarios: point-to-point, point-to-multipoint, GeoAnycast, and GeoBroadcast, and to facilitate these, geographical addressing and forwarding are the key concepts. The addresses used for forwarding packets among ITS stations are based on geographical positions of the ITS stations and the forwarding itself is relying upon that each station has a perception of its part of the network, in other words, the nearest neighbours of the ITS station and their positions. The ego ITS station does not maintain a traditional routing table instead it keeps a list of neighbours it can currently hear (receive packets from) and based on the geographical address of an incoming packet, the ITS station forwards the packet if suitable. With GeoNetworking packets can be addressed to certain geographical regions of interest without knowing if there are ITS stations in the destination area or not.

Defined day-one applications (services) are predominantly one-hop broadcast communication and some of the geographical addressing schemes require high penetration rates of co-located ITS stations. The GeoNetworking framework is spread over several standards and in Table 1 these are outlined. However, the main standard to reach interoperability between vehicles is ETSI EN 302 636-

4-1 V1.4.1 [8], describing the different fields of the GeoNetworking (GN) header and applicable settings. This standard is “media-independent functionality” implying that it disregards what radio technology is used. There is also a “media-dependent functionality” standard outlined in ETSI TS 102 636-4-2 [10] that can be used for, e.g., disseminating CBR values between ITS stations. This TS is tailored towards ITS-G5.

Table 1: Overview of GeoNetworking standard series

Standard	Name	Description
EN 302 636-1 (V1.2.1)	Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; **Part 1: Requirements**	Describes the functional requirements in GeoNetworking.
EN 302 636-2 (V1.2.1)	Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; **Part 2: Scenarios**	Describes different communication scenarios such as traditional point-to-point and point-to-multipoint scenarios as well as GeoBroadcast and GeoAnycast supported by GeoNetworking.
EN 302 636-3 (V1.2.1)	Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; **Part 3: Network Architecture**	Describes the different components within the GeoNetworking architecture.
EN 302 636-4-1 (V1.3.1)	Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; **Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality**	The GeoNetworking protocol used when transmitting data. Defines packet types for the different communication modes.
TS 102 636-4-2 (V1.1.1)	Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking**;** **Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 2: Media-dependent functionalities for ITS-G5**	Specifies DCC mechanisms at the network layer when the access technology ITS-G5 is used.

2.5. EN 302 636-5-1 BTP

BTP (Basic Transport Protocol) outlined in EN 302 636-5-1 [17], is a connectionless transport protocol specifically developed within ETSI to support low overhead communications and it adds a 4-byte header to the incoming protocol datagram unit (PDU) from the layer above. BTP multiplexes between different services found at the layer above by using port numbers in the same way as the transmission control protocol (TCP) and user datagram protocol (UDP) on the Internet do.

2.6. EN 302 637-2 CAM

Cooperative Awareness Messages (CAM) are always present, triggered based on vehicle dynamics with 1-10 Hz (1-10 times per second). For example, if the vehicle is driven above 140 km/h then it will transmit 10 CAMs per second (10 Hz). If the vehicle is stuck in a traffic jam, it will transmit with 1 Hz. The CAM contains the position, heading, speed of the vehicle amongst other things. By receiving CAMs, the vehicle can detect approaching vehicles not yet seen by the driver or line-of-sight sensors such as radar and camera. The CAM protocol is detailed in EN 302 637-2 [18].

2.7. EN 302 637-3 DENM

Decentralized Environmental Notification Messages (DENM) are triggered on behalf of an ITS application when a dangerous situation is detected by other in-vehicle sensors. They contain information about the event itself but as well speed, heading, position and they will only be transmitted when the event is still valid. DENMs can be transmitted with 1-20 Hz. The DENM protocol is detailed in EN 302 637-2 [19].

2.8. TS 103 097 Security

ETSI Technical Specification (TS) 103 097 V1.3.1 "Intelligent Transport Systems (ITS); Security; Security header and certificate formats" [14] specifies the onboard security and the format of the added signature and so forth to reach interoperability between different vehicles. As mentioned earlier, security is subject to its own deliverable (D2.9 [26]) since platooning will extend the already available PKI framework by including symmetric encryption of platooning data.

2.9. EN 302 571

2.9.1. Introduction

Harmonized EN 302 571 V2.1.1 [5] is the only standard that needs to be fulfilled to put wireless equipment operating at the 5.855-5.925 GHz band on the European market. It outlines radio related requirements on the transceiver itself such as output power, spectrum mask, spurious emission limits, etc., but it also puts up requirements on duty cycle and co-existence with CEN DSRC, see Clause 3.9.4.

2.9.2. Frequency channels

Commission implementing decision (EU) 2020/1486 [20] harmonizes the 5.875-5.935 GHz for road traffic safety applications and for usage by urban rail ITS in Europe. The frequency band was extended in 2020 with 20 MHz for road ITS V2X communication. In Figure 2, the frequency band together with corresponding documentation for the designation of the frequency band is depicted. The IEEE channel numbers found in Figure 2 are stemming from a standardized way of referring to the frequency channels in the 5-6 GHz frequency band (see Clause 17.3.8.4.2 of IEEE 802.11-2020 [9]).

The commission implementing decision 2020/1486 states that 60 MHz shall be allocated in all EU member states for traffic safety related services, where road ITS has access to 50 MHz and urban rail has exclusive access to 10 MHz and share 10 MHz with road ITS, see Figure 2 below. None of the documents regulating the 5.9 GHz band mandates a specific wireless technology to use. All wireless technologies fulfilling EN 302 571 can enter the frequency band.

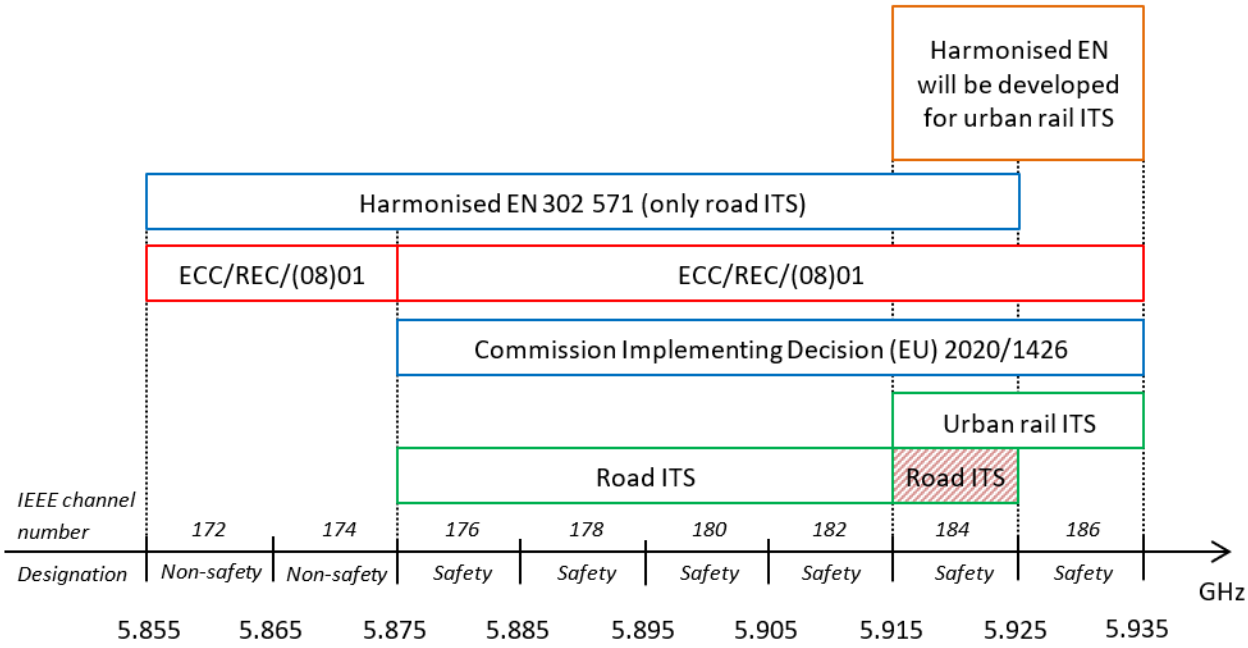

Figure 2: Overview of the 5.9 GHz band in Europe

2.9.3. Duty cycle requirements and congestion control

EN 302 571 outlines the maximum allowance for each ITS station to use the channel. The duty cycle is set to 3 % over a one second period, implying that 30 ms can be used by a single ITS station during one second. However, the 30 ms cannot be used consecutively, the maximum packet duration (T_{on}) is set to 4 ms and the time between two packets must be 25 ms (a.k.a. T_{off}), implying that if max T_{on} is used the channel can be accessed a maximum of 7 Hz. Thus, small T_{on} will result in more channel access possibilities per second and large T_{on} in fewer number of channel access. This is valid until a channel busy ratio (CBR) of 62% is reached. Then an equation kicks in increasing the duty cycle of 3%. EN 302 571 does not outline a specific DCC algorithm to be used but merely provides the limit that possible DCC algorithms need to adhere to.

NOTE: When an ITS station perceives a CBR of 60%, then there are approx. 1200 packets/s in the air given 400 byte packets transmitted with 6 Mbps. At around 2000 packets/s, a CBR of 100% is reached.

2.9.4. Co-existence with CEN DSRC[8]

EN 302 571 also puts up requirements on co-existence with CEN DSRC, which is used for collecting electronic toll (ETC) in free flow passages in Europe. It is based on a simple radio frequency identification (RFID) technology, where the onboard unit (OBU) in the vehicle is based on backscatter technology and the roadside unit (RSU) is emitting much energy to wake the OBU up. The OBU is only responding to incoming requests from the RSU and the network topology is centralized and the OBU cannot be used for anything else than ETC.

ITS-G5 equipment operating at 5.9 GHz will disturb the sensitive ETC transactions when there are many ITS-G5 equipped vehicles in the vicinity of an ETC plaza. And since ETC is a big business collecting a lot of money for road operators, the ETC business was concerned with the introduction of ITS-G5. When an ITS-G5 equipped vehicle approaches a toll plaza, it needs to adapt its number of transmissions and output power depending on the number of ITS-G5 equipped vehicles in the close vicinity of the plaza. However, this requirement on co-existence with CEN DSRC is only restricting ITS-G5 equipped vehicles in close vicinity of toll plazas. Everywhere else, transmissions and output power are restricted by EN 302 571 (duty cycle requirements) and DCC.

The technical requirements for co-existence with CEN DSRC are outlined in TS 102 792 V1.2.1 [10] (which is normatively referred to in EN 302 571). This TS outlines in essence, two different technical approaches to detect an ETC plaza (and when approaching the plaza countermeasures need to be performed to not disturb CEN DSRC):

- Carry a database onboard the ITS-G5 equipped vehicle containing the placement of all toll plazas in Europe.
- Include a CEN DSRC detector in the ITS-G5 hardware to detect when approaching a toll plaza. If a toll plaza is detected, the detecting vehicle needs to include information about the toll plaza in upcoming cooperative awareness messages (CAM) to alert other vehicles using a CEN DSRC detector (i.e., vehicles using the database do not need to react on CAMs transmitted from vehicles alerting about toll plaza).

There is an option for new installations of toll plazas to broadcast information using ITS-G5 for alerting vehicles approaching a toll plaza and all vehicles need to adhere to this information (regardless if the vehicle is using option 1 or option 2 above). Once an identification of a toll plaza is

[8] CEN DSRC should not be mixed up with DSRC technology in the US, where the latter equals IEEE 802.11p (ITS-G5). CEN DSRC will also be used for the upcoming European legislations on smart tachograpgh (Regulation (EU) No. 165/2014) and weight and dimension (Directive (EU) 2015/718) for road checks to substitute manual inspection in certain situations.

made, then countermeasures in terms of fewer transmission and reduced output power will be done depending on the number of C-ITS equipped vehicle in the vicinity. See TS 102 792 for full technical details [10].

3. PARAMETER SETTINGS OF STANDARDIZED PROTOCOLS AND RADIO

3.1. Introduction

This chapter provides parametrization of standardized protocols and radio settings for ENSEMBLE. DENM triggering of day-one applications (services) will not be required in the ENSEMBLE project but individual vehicles might have the capability of receiving and transmitting DENMs.

REQ_V2V_001: No specific triggering of DENMs will be implemented in ENSEMBLE.

3.2. Decentralized congestion control

The DCC algorithm adapts to the current CBR value and shapes the data traffic that the ego vehicle wants to transmit. The CBR is the best possible feedback currently available for understanding the status of the communication channel. DCC is not necessary to reach interoperability between vehicles. The adaptive approach in TS 102 687 V1.2.1, is applicable to platooning since it has graceful degradation of performance and will not result in major differences in transmission opportunities in between CBR assessments (important to avoid oscillations in transmission opportunities).

In the ENSEMBLE project, it is very unlikely that the CBR will increase to levels where the DCC needs to be in place to shape the data traffic. Therefore, it is decided that ENSEMBLE will not implement and evaluate any DCC algorithm since this will not be activated due to very low CBR values. But DCC needs to be investigated for platooning.

REQ_V2V_002: No DCC will be implemented in ENSEMBLE.

3.3. EN 302 571

EN 302 571 puts up requirements on radio related parameters such as output power, and in addition also on duty cycle and co-existence with CEN DSRC.

The default output power selected for CAM/DENM disseminations in Europe based on the C2C-CC profile is 23 dBm e.i.r.p. (equivalent isotropically radiated power), implying the power leaving the antenna accounting for antenna gains, cable losses, and radio chipset. ENSEMBLE will use the same output power.

REQ_V2V_003: The output power is set to 23 dBm e.i.r.p.

The duty cycle requirements are outlined in Clause 4.2.10 of EN 302 571 [5], where the overall duty cycle is set to 3 % over a one second period.

REQ_V2V_004: The ENSEMBLE platooning communication system needs to follow the duty cycle requirements outlined in Clause 4.2.10 of EN 302 571 V2.1.1 [5].

Co-existence with CEN DSRC implies that the ITS station needs to be aware of upcoming electronic toll collection (ETC) zones in Europe, which are using a carrier frequency of 5.8 GHz. When in a tolling zone, the ITS station needs to adhere to the rules set out in TS 102 792 [7] and this standard is normatively referred to from EN 302 571 [5].

REQ_V2V_005: The ENSEMBLE platooning communication system needs to implement the database solution found in TS 102 792 V1.2.1.

3.4. Multichannel support

Platooning will use a separate frequency channel since sharing channel with the verbose CAM and DENM broadcasting will force the platooning application to reduce its number of transmissions in certain situations. Many ITS-G5 equipped vehicles within radio range imply many CAM transmissions and probable DENM disseminations on channel 180 (a.k.a. control channel, CCH) and channel resources can quickly be consumed and DCC will kick in reducing the number of packet transmissions. To avoid disrupting the platooning application due to high CBR values as much as possible, the platooning control messages (PCM) shall be transmitted on a separate channel (the most suitable one is channel 176, a.k.a., service channel 1, SCH1). This implies that platooning enabled vehicles are required to have dual-radio implementations, where one radio is tuned into channel 180 to be part of both CAM and DENM transmission/receptions and find other platooning enabled vehicles on the road. The second radio is tuned into suitable channel for platooning communication as decided during the platoon establishment.

However, in ENSEMBLE, all communication will take place on channel 180 (CCH) and thus no physical channel switching will be performed. However, the protocol developed herein will contain the details and logic for changing frequency channel for the PCMs and the default channel will be set to 180.

REQ_V2V_006: All ITS-G5 communication in ENSEMBLE will take place on channel 180 (CCH).

3.5. Packet encapsulation of platooning messages and CAMs

3.5.1. Introduction

Two types of messages are used in platooning – management and control – and these are detailed in Chapter 4. The platooning messages will be encapsulated in lower layers headers and trailers. This paragraph provides information about the encapsulation at the different layers. The encapsulation will look the same regardless if it is a management or control frame that is going to be transmitted and the CAMs will also use the same encapsulation.

In Figure 3, a high-level picture of the packet encapsulation of the platooning PDU is found. For CAM it looks exactly the same but then it is the CAM PDU.

	PHY header	MAC header	LLC header	GN headers	BTP header	Platooning PDU	Security certificates	MAC trailer
Layers	Access			Network	Transport	Facilities	Network	Access

Figure 3: Packet encapsulation of the platooning PDU

3.5.2. BTP

BTP provides a connectionless transport of protocol data units (PDU). The platooning PDU will use the BTP-B header to pass its PDUs to the network layer and the GN protocol. The CAM PDU is going to use the BTP-B header. BTP is outlined in ETSI EN 302 636-5-1 V2.2.1 [17].

The BTP-A header carries both the source and the destination port (Figure 4). The destination port identifies the protocol entity at the ITS facilities layer in the destination of a BTP-PDU. The source port indicates the port that the ITS facilities layer protocol entity in the source has used to send the PDU. The source port represents the port to which a reply to the BTP PDU should be addressed in the absence of other information. The BTP-B header does not contain source port information since replies are not expected. Instead of source port information, there is a destination port field, which is set to 0 for CAM. There is a set of well-known BTP ports defined in ETSI TS 103 248 V1.3.1 [23]. BTP port 2001 shall be used for extended CAMs. Platooning needs to have two BTP ports: management and control.

NOTE: A revision of ETSI TS 103 248 [23] needs to be performed to receive port numbers for platooning. This is done by invoking a revision of TS 103 248 through initiating a new work item and then allocation of port numbers are performed. This standard is often opened to accommodate new port numbers.

REQ_V2V_007: CAMs shall use the BTP-B header.

REQ_V2V_008: CAMs shall set the field destination port to 2001.

REQ_V2V_009: CAMs shall set the field destination port info to 0.

REQ_V2V_010: Platooning PDUs shall use the BTP-B header.

REQ_V2V_011: Platooning management PDU shall set the field destination port 3005 and destination port info to 0.

REQ_V2V_012: Platooning control PDU shall set the field destination port 3006 and destination port info to 0.

3.5.3. GeoNetworking

The ENSEMBLE platooning protocol messages will be encapsulated, together with the BTP header, into a GN PDU and it will use the secured packet header as outlined in ETSI TS 103 097 V1.3.1 [14]. This imposes that the GN Secure Packet format as defined in Clause 9 of ETSI EN 302 636-4-1 [8] shall be used. During development or for testing purposes security might be disabled. In that case, the GN Header as defined in Clause 9.3 of ETSI EN 302 636-4-1 [8] can be used. Figure 4 shows the secured GN header.

The ENSEMBLE platooning protocol will use the single-hop broadcast GN messages to distribute CAMs, platoon control as well as platoon management messages such as join/leave request/response. Although GN unicast messages could have been selected for management messages, the GN unicast message header imposes unnecessary overhead and complexity. All the management messages defined in the ENSEMBLE platooning protocol already include destination and source identifiers that allow the delivery of a message to its final destination.

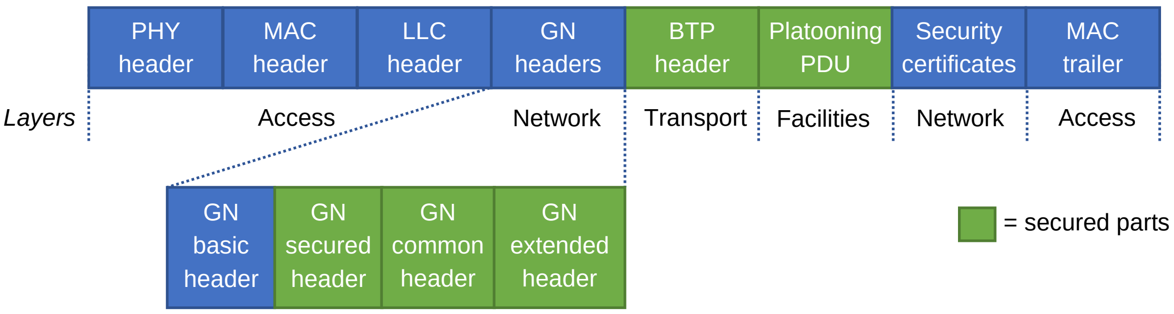

Figure 4: Secured GN encapsulation of the platooning PDU and BTP header

The single-hop broadcast GN header structure is found in Clause 9.8.4 of ETSI EN 302 636-4-1 [8]. In Table 2, the parameter setting of the different fields throughout the three headers belonging to GN constituting the single-hop broadcast message is detailed.

Table 2: Parameter setting for single-hop broadcast

Header	Field	Bits	Description	Extended CAMs	Platoon control message	Platoon management message
BASIC	Version	4	Identifies the version of the GN protocol	1	1	1
	NH	4	Next Header. Type of header immediately following the GeoNetworking	2 (Secured Header)	2 (Secured Header)	2 (Secured Header)
	Reserved	8	Set to 0	0	0	0
	LT	8	Life time	1000 ms (as per normal CAMs)	50 ms	1000 ms
	RHL	8	Remaining Hop Limit. Max number of hops. Drop if RHL is zero	1	1	1
COMMON	NH	4	Next Header. Type of header immediately following the GeoNetworking common header	2 (BTP-B)	2 (BTP-B)	2 (BTP-B)
	Reserved	4	Set 0	0	0	0
	HT	4	Header Type. Identifies the type of the GeoNetworking Header	0 (TSB)	0 (TSB)	0 (TSB)
	HST	4	Header Sub-Type. Identifies the type of the GeoNetworking Sub-Header	5 (SHB)	5 (SHB)	5 (SHB)
	TC	8	Traffic class that represents facilities layer requirements on packet transport.	SCF=0, CO=0, TCID=AC_BE	SCF=0, CO=0, TCID=AC_VO	SCF=0, CO=0, TCID=AC_BK
	Flags	8	Bit 0: Indicates whether the ITS-S is mobile or stationary (GN protocol constant itsGnIsMobile), Bit 1 to bit 7 set to 0.	1 (mobile)	1 (mobile)	1 (mobile)

	PL	16	PayLoad. Length of the GeoNetworking payload in bytes	Depends on packet size	Depends on packet size	Depends on packet size
	MHL	8	Maximum Hop Limit	1	1	1
	Reserved	8	Set to 0	0	0	0
EXTENDED	SO PV	192	Source Position Vector and it is of the type Long Position Vector containing the reference position of the source. 24 bytes	Current EGO position	Current EGO position	Current EGO position
	Media dependent data	32	4 bytes used by the media-dependent functionality if supported (see TS 102 636-4-2). Not used in ENSEMBLE. Set to 0.	0	0	0

REQ_V2V_013: The single-hop broadcast packet header as outlined in Clause 9.8.4 of ETSI EN 302 636-4-1 V1.4.1 shall be used.

REQ_V2V_014: The parameter setting of the single-hop broadcast packet with its different headers shall be as outlined in Table 2.

3.5.4. LLC and SNAP headers

The Logical Link Control (LLC) header outlined in IEEE 802.2 together with the extension Subnetwork Access Protocol (SNAP) header outlined in IEEE 802-2001 provides the possibility to distinguish between different network layer protocols through the unique Ethernet Types. GeoNetworking has the EtherType 8947 (hex). See Figure 5 for an outline of LLC and SNAP.

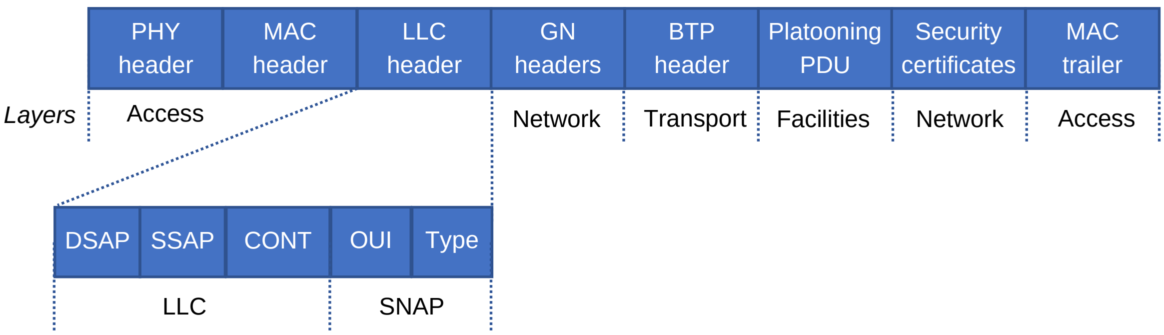

Figure 5: LLC encapsulation of the platooning PDU

Table 3: Parameter setting of LLC and SNAP fields

	Field	Bits	Description	Value (hexadecimal)
LLC	DSAP	8	Destination Service Access Point. Set to default value of AA (hex) to inform that the SNAP header is included.	0xAA
	SSAP	8	Source Service Access Point. Set to default value of AA (hex) to inform that the SNAP header is included.	0xAA
	CONTROL	8	LLC offers three different services to the protocols above: (*i*) unacknowledged connectionless, (*ii*) connection mode, (*iii*) acknowledged connectionless. The unacknowledged connectionless service is used in LLC.	0x03
SNAP	OUI	24	IEEE organizationally unique identifier (OUI) is set to 0 if Ethernet Type (EtherType) is used for distinguishing between diffent network protocols.	0x000000
	Type	16	The EtherType is a unique identifier for the network protocol on top of LLC. GeoNetworking's EtherType is 8947 (hex).	0x8947

REQ_V2V_015: The LLC and SNAP headers as outlined in Table 3 shall be used.

3.5.5. IEEE 802.11p MAC and PHY

The medium access control sublayer and the physical layers of IEEE 802.11p are contained in the radio chipset. The MAC layer exposes four different priority queues to higher layers to prioritize between internal applications. These queues are called: AC_BK (access category background), AC_BE (AC best effort), AC_VI (AC video), and AC_VO (AC voice). Informative Annex B in ETSI EN 302 663 [4] contains a thorough explanation about the MAC procedure and the queues for the interested reader.

The PHY layer of IEEE 802.11p offers 8 different transfer rates where 3 are mandatory to support. The different rates are achieved by altering the modulation scheme together with the coding. The transfer rates are: **3**, 4.5, **6**, 9, **12**, 18, 24, and 27 Mbps (the bold ones are mandatory). The default rate for transmitting packets has been set to 6 Mbps. All CAM and DENM transmissions will use 6 Mbps. It is possible to change the transfer rate on a packet-per-packet basis.

The MAC address is changed every time the station ID in CAM is changed. The procedure of changing MAC address needs to be detailed here.

REQ_V2V_016: The default rate for all messages transmitted in ENSEMBLE is 6 Mbps.

REQ_V2V_017: The platoon control message shall use access category AC_VO.

REQ_V2V_018: The platoon management message shall use access category AC_BK.

REQ_V2V_019: The CAM shall use access category AC_BE.

REQ_V2V_020: The MAC address shall change every time the station ID in the CAM is changed.

3.6. CAM

CAMs are used for announcing that a vehicle has platooning capability in ENSEMBLE. The trucks will transmit CAMs according to the generation rules outlined in ETSI EN 302 637-2 [18]. CAMs are always transmitted on channel 180 (CCH). One new container is introduced in the CAM in support of platooning, which holds one flag called `isJoinable`. When this flag is set to true, the vehicle is interested in receiving a JOIN REQUEST from a vehicle from behind. In Figure 6, the placement of the new Platooning container in the CAM structure is depicted. The platooning container does not break backward compatibility and vehicles not supporting the platooning container will disregard it.

```
CamParameters ::= SEQUENCE {
    basicContainer BasicContainer,
    highFrequencyContainer HighFrequencyContainer,
    lowFrequencyContainer LowFrequencyContainer OPTIONAL,
    specialVehicleContainer SpecialVehicleContainer OPTIONAL,
    ...,
    platooningContainer PlatooningContainer OPTIONAL
}
```

Figure 6: The placement of the platooning container in the CAM

REQ_V2V_021: The platooning Container shall always be included in the CAM.

The details of the platooning container are found in Figure 7. It contains one flag called `isJoinable`, which will be used for signalling if a vehicle accepts other vehicles to platoon with or not. The platooning container contains "three dots" implying that it can be expandable to contain more parameters that might be identified as necessary.

```
PlatooningContainer ::= SEQUENCE {
        isJoinable BOOLEAN,
        ...
}
```

Figure 7: The PlatooningContainer and the `isJoinable` flag

REQ_V2V_022: The PlatooningContainer shall include one BOOLEAN flag.

3.7. Application ID

Each ITS application is globally identified by an Intelligent Transport Systems Application Object Identifier (ITS-AID) and these are outlined in ETSI TS 102 965 [11]. ISO 17419 [35] regulates allocation of new ITS-AID globally. When starting the ENSEMBLE project no ITS-AID number has been defined for platooning. To request the assignment of a new ITS-AID during the project the template available at [12] should be used. Until a new Application ID is assigned for platooning, a testing/private ITS-AID shall be used.

REQ_V2V_023: The CAM shall use ITS-AID 0x24.

ITS-AID is also referred as Provider Service Identifier (PSID) at IEEE 1609.2 [13]. PSID are used to indicate the permissions that a certificate holder has to sign application data with a given certificate. This point will be considered when issuing/requesting new certificates from the PKI (Public Key Infrastructure).

REQ_V2V_024: The platooning application shall use ITS-AID 0x3FF.

3.8. Security

The platooning protocol will use the already existing PKI developed for day-one services in Europe. Each truck shall keep a list of valid certificates provided by the PKI and use them for signing and verifying messages following ETSI TS 103 097 [14]. In addition, encryption of platoon data is enabled by the exchange of two keys that can be generated by each truck in the platoon, the participant key, and the group key. The participant key secures communication between two trucks in the platoon whereas the group key secures the overall platoon communication between all trucks. The ENSEMBLE security approach is outlined in more detail in ENSEMBLE deliverable D2.9 [26].

4. PLATOONING PROTOCOL

4.1. Introduction

This chapter gives an overview of the platooning protocol's logic together with the different message types (control and management) and new data elements. To enable the functionality of platooning a set of preconditions must likely be fulfilled, however, it is out of scope for this deliverable to define those. In Table 4, examples of preconditions are outlined, and the preconditions are divided into dynamic and static, where the former can change on a daily basis down to on a minute basis whereas the static ones are changed on weekly up to monthly basis. In this deliverable the assumption is that all preconditions are fulfilled and the vehicle together with driver are eligible to be part of a platoon.

Table 4: Examples of preconditions for platooning

Precondition		Description
Static	Inspection	The truck and trailer have been through its annual inspection.
	Subscription	There might be a subscription to a specific platooning service regulating access to the platooning system.
	Country	There might be country-specific regulation for platoons, such as how many vehicles that are allowed in a platoon in a specific country. It might take some time to harmonize legislation around platoon operation on real roads.
	Road infrastructure	There might be bridges and similar physical infrastructure influencing for example distance and/or speed of a platoon.
	Geofencing	There might be special environmental zones to consider.
Dynamic	Location	The vehicle needs to be on a road network permitting platoons.
	Weather	Local weather conditions may prohibit platooning.
	Road conditions	Road conditions could differ from weather conditions. Bad road conditions will prohibit platooning. For example, a road can be icy even though the sun is shining in the winter.
	Day-one services	There might be accidents, traffic jams and road works, during the platoon travel that will inactivate the platooning capability (this depends of course on how advanced use cases the platoon can handle, probably more advanced as time goes by).
	In-vehicle warnings	Before and during the course of the platoon no warnings in the dashboard shall appear or be present.
	Heavy/Long vehicle	The truck could exceed possible physical dimension due to carried load to be able to platoon.
	Other vehicle specific features	There might be other vehicle related features that need to be set. For example, should the adaptive cruise control (ACC) be a precondition?

4.2. Message types

Platooning is based on a set of messages. A vehicle will signal its willingness to platoon through cooperative awareness messages (CAM) in a new container carrying a flag called `isJoinable`, stating this feature.

> NOTE: In a real-world deployment, CAMs will not be used for signalling the platooning capability due to backward compatibility issues in the long run. Instead, service announcement messages will be used. The ENSEMBLE project selected CAMs to avoid the extra complexity of adding a message not crucial for the platooning functionality. Using CAMs or service announcements do not change the proposed logic or other protocol messages detailed in present deliverable.

There are two types of platooning messages: platooning management message (PMM) and platooning control message (PCM). The former consists of three subtypes: JoinRequest, JoinResponse, and PlatoonUpdate (former KeyUpdate). The JoinRequest message is transmitted by a vehicle interested in joining a platoon that it has identified through CAM receptions containing the flag `isJoinable` set to true from a vehicle in front. The JoinResponse message is transmitted in a response to the JoinRequest message and PlatoonUpdate is used, e.g., to regularly updating the encryption key for PCMs. When a successful join has been carried out, PCMs are transmitted by all vehicles in the platoon with 20 Hz (20 times per second). No explicit acknowledgements are transmitted in response to a successful PCM reception. Instead, implicit ACKs are used, and all platoon members expect a new PCM from other members every 50 ms. When a vehicle intends to leave the platoon, it signals its intention in the PCMs allowing for continuous information exchange during the leave process. ASN.1 descriptions of the new messages together with the addition to CAMs are detailed in Appendix B. In Figure 8, a high-level picture of two vehicles joining a platoon, transmit control messages and leave the platoon is provided in which Vehicle V0 is in front of Vehicle V1.

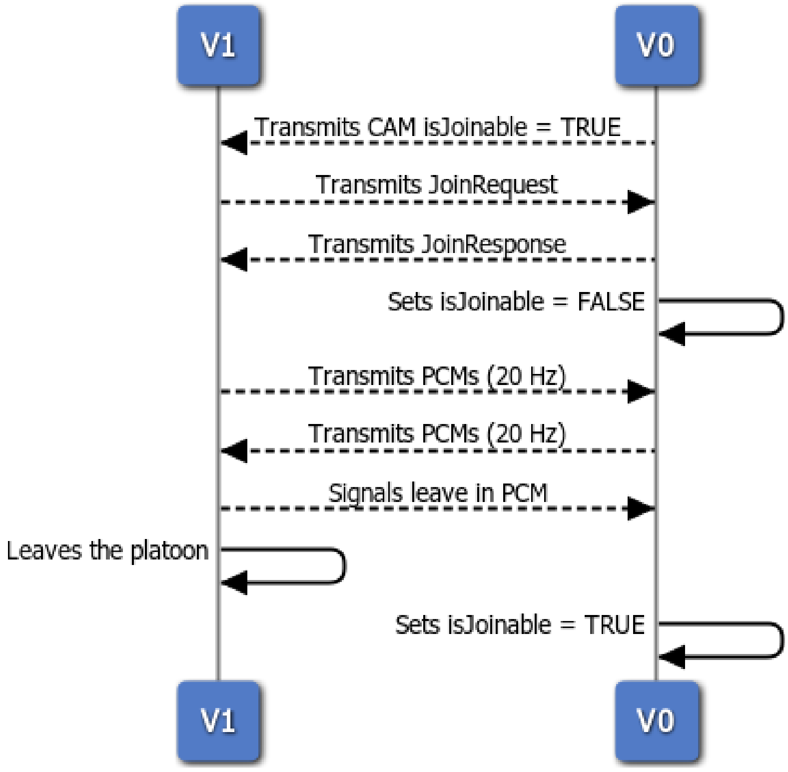

Figure 8: High level platooning from join to leave

4.3. Security

Securing the communication is an essential part of the work within the ENSEMBLE project and the security approach is described in deliverable D2.9 [26].

4.4. Protocol building blocks

From the ENSEMBLE use case descriptions found in deliverable D2.3 [27] and the requirements and specifications in deliverable D2.5 [28], the following building blocks were identified and further developed into sequence diagrams. The sequences are structured into: a short description, the diagram itself, preconditions, main sequence, post-condition, and parameter definitions. The parameters are constants based on tests and results collected under the road testing activities in the ENSEMBLE project. The source code to the sequence diagrams is found in Chapter 9.2 under Appendix B.

4.4.1. Joining

A vehicle finding a platoon in front of it to join can ask for joining under the condition that the platooning support function is activated. The "joinable" platoon can be a "platoon" consisting of only one single vehicle or two and more vehicles. Figure 9 illustrates the joining procedure in a sequence diagram. Vehicle V1 is travelling behind Vehicle V0, and V0 has set its isJoinable flag in the CAM to TRUE. Vehicle V1 will initiate the joining procedure by transmitting a JoinRequest. Depending on the answer of the JoinRequest, different operations will take place. Further, Vehicle V1 can be a leader of a platoon, i.e., it can have follower(s). This is also captured in Figure 9.

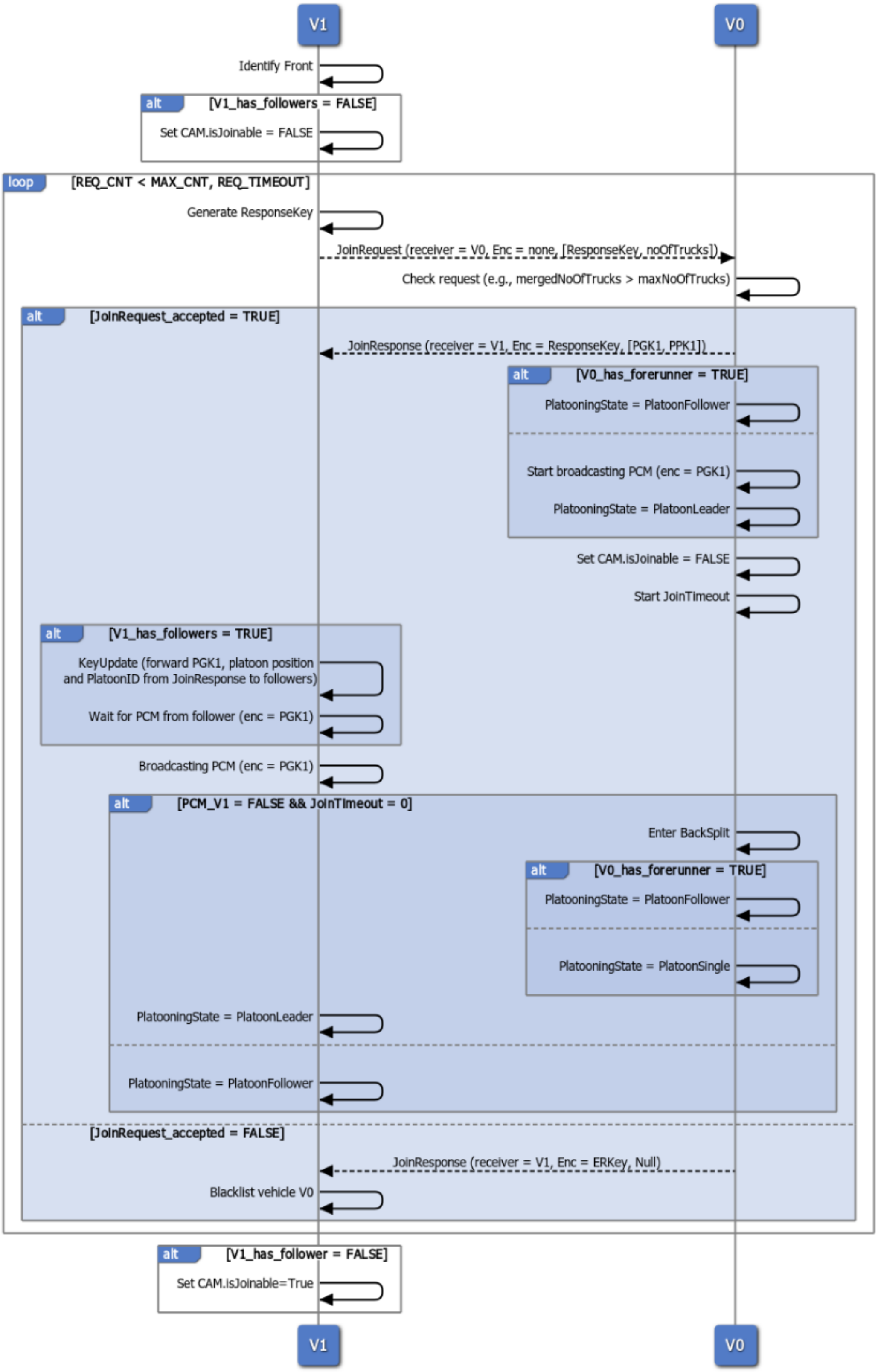

Figure 9: Joining sequence

Pre-condition:

- Ego vehicle is driving with the platooning function enabled.
- The vehicle in front may either be a standalone vehicle, or the last follower in a platoon.

Main sequence:

1. Ego vehicle is receiving CAMs from another vehicle in front, with `isJoinable` set to true.
2. The ego vehicle is approaching another vehicle, and by comparing the information in the CAM (vehicle length and GNSS position) with input from its local vehicle sensors, it concludes that the vehicle immediately in front is the same vehicle from which it receives CAMs.
3. If the vehicle in front is not black-listed, the ego vehicle sets `isJoinable` flag to false (only when the ego vehicle is a standalone vehicle, otherwise it is already false) and a JoinRequest message is sent. The ego vehicle now enters the joining state.
4. If no answer is received within REQ_TIMEOUT, the JoinRequest message is repeated (step 3).
5. The vehicle in front receives the JoinRequest message, evaluates it and broadcasts a positive JoinResponse message, containing among other things the platoon position of the joining vehicle and the security keys (PGK, PPK) for securing the platoon communication.
6. The ego vehicle receives the JoinResponse message, enters the platooning state and starts to broadcast PCMs.
7. The ego vehicle sets its `isJoinable` flag to true, if it was a single vehicle before joining, otherwise, it stays false if ego vehicle was in a platoon already (i.e., the leader of a platoon).

Exceptions:

3a. The vehicle in-front is black-listed. No JoinRequest message is sent. Ego vehicle continues to drive with a safe distance to the vehicle in front. The use case ends.

4a. The JoinRequest message has been repeated MAX_CNT number of times. Ego vehicle blacklists the vehicle in front based on its stationID and no further attempts to connect will be performed until the vehicle in front has changed its stationID. Ego vehicle continues to drive with standalone distance to the vehicle in front. The use case ends.

5a. The vehicle in front broadcasts a negative JoinResponse message. Ego-vehicle blacklists the vehicle in front based on its stationID and no further attempts to connect will be performed, until the vehicle in front has changed its stationID. Ego vehicle continues to drive with a safe distance to the vehicle in front. The use case ends.

5b. The vehicle in front broadcasts a negative JoinResponse message, but the ego-vehicle fails to receive the JoinResponse message. Go to main flow (4).

5c. The vehicle in front broadcasts a positive JoinResponse message, but the ego vehicle fails to receive the JoinResponse message. The vehicle in front (V0) will wait for JOIN_TIMEOUT milliseconds to receive the first PCM from the ego vehicle (V1) thereafter the neighbour watchdog use case is entered, which will result in a back split. The Joining use case is then restarted.

Alternative flow:

5d. The platoon in front is consisting of a single truck, that, after broadcasting the positive JoinResponse message, becomes the leader of the whole platoon and starts to broadcast PCMs.

6a. If the ego vehicle is a leading truck of a joining platoon it must generate a PlatoonUpdate message and send the PGK and platoon ID backwards, as well as the updated position for the following vehicle(s) (received that information via the JoinResponse message).

Post-condition:

- Ego-vehicle has entered platooning. Counter on repeating the JoinRequest message is reset to 0, time-outs for the request and join are reset to 0.
- The neighbour watchdog is activated.
- The longitudinal control algorithm on ego-vehicle may, depending on situation, decrease the gap to the vehicle in front.

Constants:

REQ_TIMEOUT	500 ms
MAX_CNT	5
JOIN_TIMEOUT	3000 ms

Requirements:

REQ_V2V_025: The joining vehicle shall send a JoinRequest message to indicate its intention to join the platoon. The JoinRequest message shall include the receiver's StationID, number of joining trucks, the encryption response key (ERKey, to be used by the receiving vehicle to encrypt the JoinResponse message), its vehicle configuration and its platooning level.

REQ_V2V_026: The receiving vehicle shall evaluate the received JoinRequest message and answer with a JoinResponse message including either a negative (joinResponseStatus = NULL) or a positive (joinResponseStatus = JoinResponseInfo) response status. A positive JoinResponse message shall include the Platooning Group Key (PGK), the Platooning Participant Key (PPK), frequency channel selected, PlatoonID, the maximum number of vehicles in the platoon, and the position of the joining vehicle in the platoon.

REQ_V2V_027: A vehicle becoming the leader of the newly formed platoon shall generate PGK, and PPK and PlatoonID

REQ_V2V_028: A vehicle becoming the leader of a newly formed platoon shall start sending out PCMs immediately after the positive JoinResponse was sent out.

REQ_V2V_029: A joining vehicle without followers shall start sending out PCMs immediately after a positive JoinResponse was received on its JoinRequest.

4.4.2. Front split

The front split is the action a vehicle executes when leaving the vehicle in front. This may, e.g., occur when a non-platooning vehicle cuts into the platoon in front of the ego vehicle. The vehicle signals its intention to split up the platoon in front of it. Result could be that the split initiating vehicle becomes standalone (when it is the last one in the current platoon) or a leader of a new platoon (when it has following vehicles).

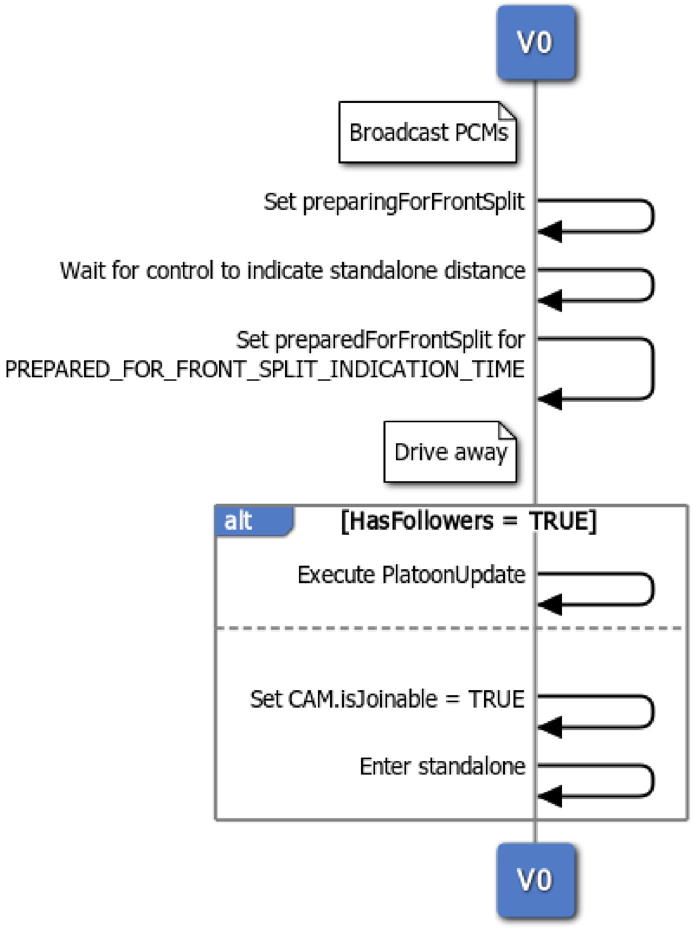

Figure 10: FrontSplit sequence

Pre-condition:

- Ego vehicle is driving in a platoon.
- Ego vehicle is not the leader of the platoon.
- Ego vehicle has not attached the SplitStatus container in its PCMs.

Main sequence:

1. The vehicle sets the FrontSplit enum to preparingForFrontSplit. for PREPARING_FOR_FRONT_SPLIT_INDICATION_TIME (see step 3).
   OPTIONAL: The vehicle increases it distance to a safe distance.
2. The vehicle sets the FrontSplit enum to preparedForFrontSplit in the PCM for PREPARED_FOR_FRONT_SPLIT_INDICATION_TIME.
3. If there are followers: The vehicle becomes leader and enters the platoon update use case.
4. If no followers are present, the vehicle becomes standalone sets the isJoinable flag to true and stops sending PCMs.

Alternative flows:

5a. There are no followers: The vehicle is standalone and stops sending PCMs. This use case ends.

Post condition:

- Ego vehicle has entered standalone (when it was a trailing vehicle in the old platoon) or is leader of a new platoon (when it had followers in the old platoon).

Constants:

PREPARING_FOR_FRONT_SPLIT_INDICATION_TIME	0-10 seconds (depending on how long the truck needs to increase distance)
PREPARED_FOR_FRONT_SPLIT_INDICATION_TIME	150 ms

Requirements:

REQ_V2V_030: A vehicle that is splitting the platoon in front, shall indicate that to its direct preceding vehicle by setting preparingForFrontsplit in its PCMs. If the vehicle is becoming a leader after a split, it shall initiate a platoon update sequence immediately (see 5.4.5 Platoon update).

4.4.3. Back split

The back split is the action a truck executes when leaving the vehicle behind. Reason could be that the split initiating vehicle wants to leave (if it happens to be the leader) or that it wants to split up the current platoon for other reasons (different routes to take, etc.).

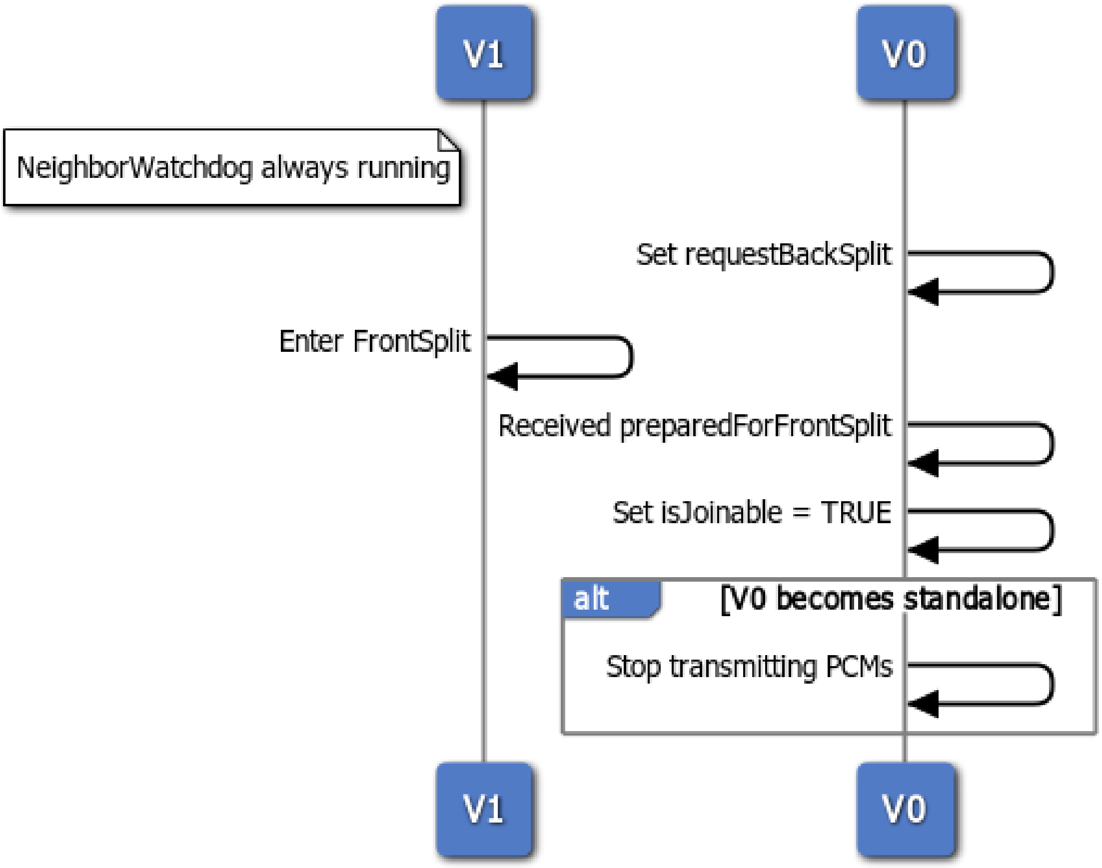

Figure 11: BackSplit sequence

Pre-condition:

- Ego vehicle is driving in a platoon.
- Ego vehicle is not the trailing vehicle of the platoon.

Main sequence:

1. Ego vehicle is requesting a back split, by setting the corresponding flag (requestBackSplit) in its PCMs.
2. The vehicle immediately behind notices the back split request from the vehicle in front and responds with setting the preparing-for-front-split flag in its PCMs, while starting to increase its distance to the vehicle in front.
3. When the vehicle immediately behind the ego vehicle has increased its distance to a safe distance, it sets the prepared-for-front-split flag in its PCMs.
4. The ego vehicle receives PCMs from the vehicle immediately behind, which indicates that it is prepared for the split.
5. Ego vehicle sets the `isJoinable` flag in its CAMs to true.
6. If the ego vehicle has no preceding vehicles, it stops sending PCMs.

Alternative flow:

6a. The back split was issued because the platooning function of the ego vehicle was disabled. Therefore, `isJoinable` is not set to true.

Exceptions:

2a. If the ego vehicle is about to become standalone and does not receive the preparing-for-front-split it should stop sending PCMs after BACKSPLIT_TIMEOUT.

2b. If the ego vehicle is about to become standalone and does not receive the prepared-for-front-split it should stop sending PCMs after BACKSPLIT_TIMEOUT.

Post-condition:

- Ego vehicle is a trailing or standalone vehicle.

Constants:

BACKSPLIT_TIMEOUT	10 s

Requirements

REQ_V2V_031: A vehicle that is splitting the platoon in the back
shall indicate that to its direct following vehicle by setting the requestBackSplit flag in its PCMs.

REQ_V2V_032: A vehicle that is receiving requestBackSplit flag in the PCMs from its preceding vehicle shall immediately set preparingForFrontSplit in its PCMs.

REQ_V2V_033: A following vehicle that received via PCM the request for back split and has reached the standalone distance to its preceding vehicle shall set preparedForFrontSplit.

REQ_V2V_034: A vehicle that is receiving requestBackSplit flag in the PCMs from its preceding vehicle shall forward the request to its control system (to increase distance to standalone).

4.4.4. PlatoonUpdate (former KeyUpdate)

The PlatoonUpdate message is triggered when the platoon group key (PGK) expires or when the platoon changes leader. It is the responsibility of the leader to trigger this message periodically. The PlatoonUpdate contains PlatoonID, PGK, platoon participant key (PPK), and the position in the platoon of the vehicle transmitting the PlatoonUpdate. The PlatoonID will not be changed if the leader is unchanged, hence, this is only updated when the platoon leader changes. Figure 12 illustrates the PlatoonUpdate sequence with a platoon consisting of four vehicles. The PlatoonUpdate sequence is initiated by the leader, called V0 in the figure. The PlatoonUpdate is a unicast between two trucks, i.e., the vehicle transmitting the PlatoonUpdate and its immediate follower, see Figure 12. More details on the security are found in Deliverable D2.9 [26].

NOTE: In the early phase of the ENSEMBLE project a KeyUpdate message was defined to fulfil security and privacy requirements. During the runtime of the project the message and its mechanism of distribution became more relevant even to update other platoon data elements that need to be shared. Therefore, the name was changed to PlatoonUpdate.

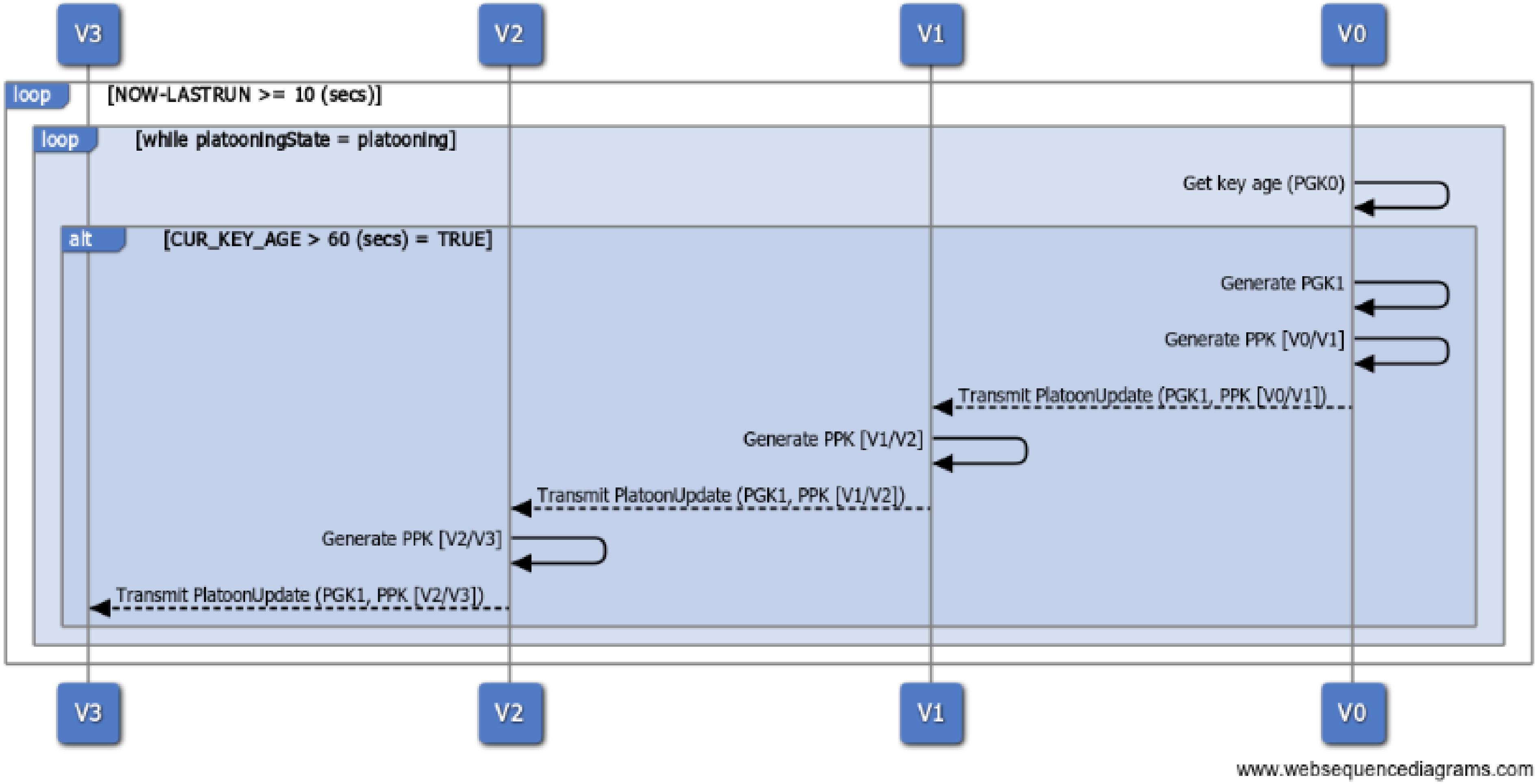

Figure 12: PlatoonUpdate sequence

Pre-condition:

- Ego vehicle is driving in a platoon.
- Ego vehicle is the leader of the platoon.

Main sequence:

1. The ego vehicle checks every 10 seconds (timer triggered, CHECK_KEY_INTERVAL) if the current-key-age (i.e., age of the PGK that is used for encrypting the PCMs) is greater than MAX_KEY_AGE.
2. If the key age is older than MAX_KEY_AGE, then the ego vehicle generates a new PGK and a new PPK.
3. The ego vehicle sends a PlatoonUpdate message to its direct follower including the newly generated keys (PGK and PPK).

Post-condition:

- Ego vehicle is the leading vehicle in the platoon.

Constants:

CHECK_KEY_INTERVAL	10 seconds
MAX_KEY_AGE	60 seconds

Requirements

REQ_V2V_035: A vehicle leading the platoon shall check the current age of the PGK used for encryption every 10 seconds.

REQ_V2V_036: A vehicle leading the platoon shall generate a new PGK and a new PPK if the age of the currently used PGK is older than 60 seconds. It shall send the PGK included in a PlatoonUpdate message to its following vehicle.

REQ_V2V_037: A follower (but not a trailing vehicle) in a platoon shall on reception of a received PlatoonUpdate message (from its preceding vehicle) generate a new PPK and send a PlatoonUpdate message to its direct follower (which can be the trailing vehicle), including the received PGK.

REQ_V2V_038: The trailing vehicle in the platoon shall on reception of a PlatoonUpdate message immediately start using the PGK for its PCM data encryption.

REQ_V2V_039: A vehicle being the leader or follower in a platoon shall on recognition of PCMs from the direct following vehicle, which uses the new PGK for encryption, immediately start using the new PGK itself for its PCM data encryption.

4.4.5. Neighbour watchdog

The watchdog shall supervise the reception of PCMs from the truck in front and behind. If it has not received messages for a certain period, then the truck should be brought into standalone mode by initiating front or back split or both. Figure 13 shows a situation where the middle truck (V1) in a three-truck platoon might for some reason do not receive the PCMs transmitted by the vehicle in front (V0) or/and behind (V2), then V1 can initiate a front split and/or back split. All trucks (V0, V1, V2) transmit PCMs with 20 Hz.

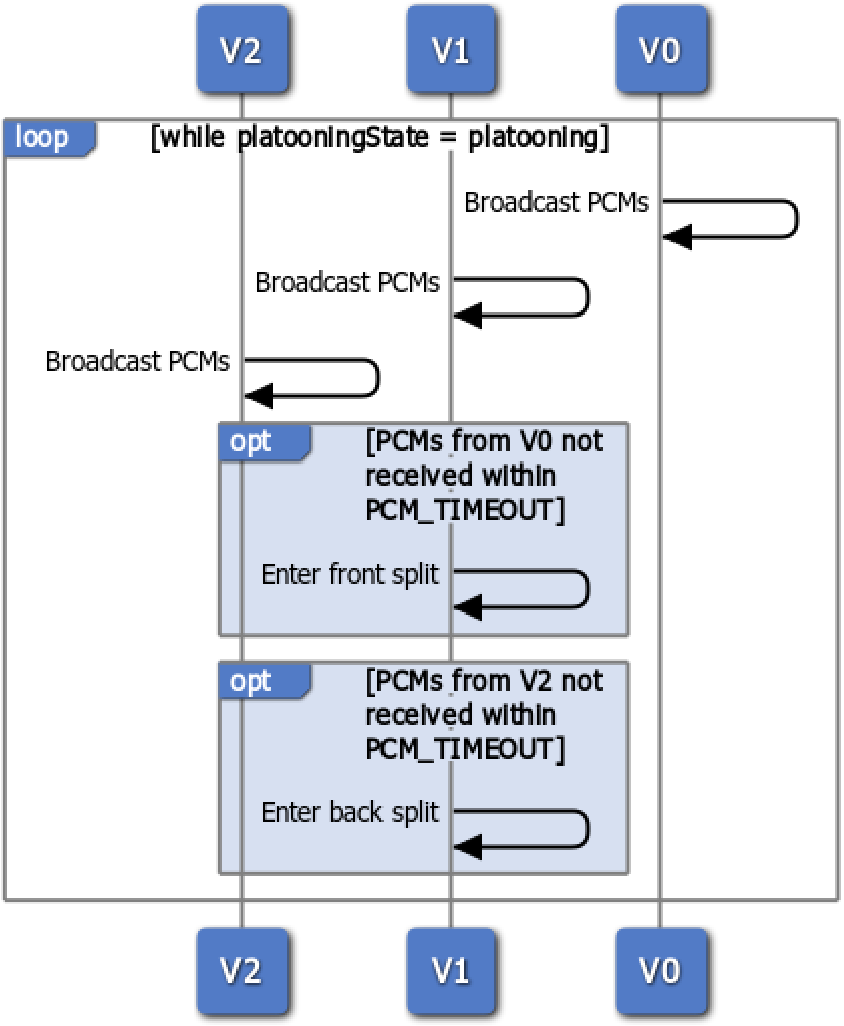

Figure 13: Neighbour watchdog sequence

Pre-condition:

- Ego vehicle is driving in a platoon and has thus completed the join process.

Main sequence:

1. Ego vehicle monitors the received PCM from its closest neighbours immediately in front and behind.
2. If the time elapsed since the last received PCM from the vehicle in front exceeds PCM_TIMEOUT, then the use case front split is performed.
3. If the time elapsed since the last received PCM from the vehicle behind exceeds PCM_TIMEOUT, then the use case back split is performed.
4. Go to (1).

Alternative flow:

2a. Ego vehicle is the platoon leader. Go to (3).
3a. Ego vehicle is the trailing vehicle of the platoon. Go to (4).

Post-condition:

- Any vehicle that is not broadcasting PCM has been excluded from the platoon.

Constants:

PCM_TIMEOUT	150 ms

Requirements:

REQ_V2V_040: A vehicle being the leading vehicle in a platoon shall check if PCMs from the direct following vehicle are received with a frequency of 20 Hz.

REQ_V2V_041: A vehicle being the following vehicle in a platoon shall check if PCMs from the direct preceding vehicle and direct following vehicle are received with a frequency of 20 Hz.

REQ_V2V_042: A vehicle being the trailing vehicle in a platoon shall check if PCMs from the direct preceding vehicle are received with a frequency of 20 Hz.

REQ_V2V_043: A vehicle recognizing that no PCMs from the direct follower have been received during PCM_TIMEOUT shall initiate a back split.

REQ_V2V_044: A vehicle recognizing that no PCMs from the direct preceding vehicle have been received during PCM_TIMEOUT shall initiate a front split.

4.5. Communication with smart road infrastructure

Smart road infrastructure equipped with short-range communication can communicate directly with platoons passing by using the Infrastructure to Vehicle Information Message (IVIM) described in TS 103 301 V1.2.1 [24]. An IVIM supports mandatory and advisory road signage such as contextual speeds and road works warnings. It should be noted that road works warnings can also use DENMs.

An IVIM can be transmitted based on specific timings (e.g., only valid between 8:00 and 20:00) and/or based on the context (e.g., applicable in case of fog). IVIM uses the digital representation of traffic signs defined in ISO TS 14823 [25], developed to create a common basis for transmitting encoded information for existing road traffic signs and pictograms.

The IVIM can be used to, e.g., signal maximum speed allowed and the minimum distance between vehicles. The distance between vehicles in the platoon might need to be increased for certain road infrastructure such as bridges which might have been designed for a certain load and cannot cope with several heavily loaded trucks at short distances. This to prevent damages to the road infrastructure but also for safety reason. The same manoeuvre might also be necessary due to safety when driving in a tunnel such as the Mont Blanc tunnel.

In the ENSEMBLE project, CAMs were used to transmit information about speed limits and minimum distance between vehicles but for real-world deployment the IVIM shall be used.

5. SUMMARY AND CONCLUSION

The present deliverable specifies a facilities layer protocol, supporting the platooning application by introducing management and control messages. It makes use of already standardized protocols such as ITS-G5, GeoNetworking and BTP. Further, the security framework developed for C-ITS day-one applications based on PKI is used to create a trust domain with the addition of encrypting platooning data. The developed protocol resides together with other protocols such as CAM in the facilities layer. Vehicles find each other by signalling in CAMs that they are available for platooning (NOTE: CAMs will not be used for this in a real-world deployment, but instead service announcements). Once a vehicle capable of platooning detects another vehicle in front of it, the follower vehicle will initiate a join procedure to establish a platoon or join an already existing platoon. When a vehicle wants to leave the platoon, it indicates that for a certain time in the control frames (PCMs).

The deliverable also includes ASN.1 descriptions of the different message types and included signals, which are found in Appendix B. The security for platooning is detailed in deliverable D2.9 [26].

Deliverable D2.5 contains the lessons learned and future considerations on the communication protocol as concluded from the final testing of the PSF with seven-brands in Spain taking place in September 2021. Deliverable D6.15 [34] provides guidance for the upcoming standardization on platooning.

6. REFERENCES

[1] A. Festag, “Cooperative Intelligent Transport Systems Standards in Europe,” in IEEE Communications Magazine, Vol. 52, No. 12, December 2014, pp. 166-172.

[2] K. Sjöberg, P. Andres, T. Buburuzan, and A. Brakemeier, “Intelligent Transport Systems in Europe – Current Deployment Status and Outlook,” in IEEE Vehicular Technology Magazine, Vol. 52, No. 12, June 2017, pp. 89-97.

[3] J. Kenney, “Dedicated Short-Range Communications (DSRC) Standards in the United States,” in Proceedings of the IEEE, Vol. 99, No. 7, July 2011, pp. 1162-1182.

[4] ETSI EN 302 663 V1.3.1 (2020-01), “Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band,” January 2020.

[5] ETSI EN 302 571 V2.1.1 (2017-02), “Intelligent Transport Systems (ITS); Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonised Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU,” February 2017.

[6] ETSI TS 102 687 V1.2.1 (2018-04), “Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part,” July 2011.

[7] ETSI TS 102 792 V1.2.1 (2015-06), “Intelligent Transport Systems (ITS); Mitigation techniques to avoid interference between European CEN Dedicated Short Range Communication (CEN DSRC) equipment and Intelligent Transport Systems (ITS) operating in the 5 GHz frequency range,” October 2012.

[8] ETSI EN 302 636-4-1 V1.4.1 (2020-01), “Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geogrpahical addressing for point-to-point and point-to-multipoing communications; Sub-part 1: Media-Independent Functionality,” January 2020.

[9] IEEE Std. 802.11-2020, “IEEE Standard for Information technology – Telecommunications and information exchange between systems Local and metropolitan area networks – Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”

[10] ETSI TS 102 636-4-2 V1.4.1 (2021-02), “Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geogrpahical addressing for point-to-point and

point-to-multipoint communications; Sub-part 1: Media-dependent functionalities for ITS-G5," February 2021.

[11] ETSI TS 102 965 V1.5.1 (2020-01), "Intelligent Transports Systems (ITS); Application Object Identifier (ITS-AID); Registration," January 2020.

[12] Respitory for Application IDs, http://standards.iso.org/iso/ts/17419/TS17419%20Assigned%20Numbers/

[13] IEEE Std 1609.2™-2016, "IEEE Standard for Wireless Access in Vehicular Environments – Security Services for Applications and Management Messages – Amendment 1," January 2016.

[14] ETSI TS 103 097 V1.3.1 (2017-10), "Intelligent Transport Systems (ITS), Security; Security header and certificate formats," October 2017.

[15] ETSI TS 102 894-2 V1.3.1 (2018-08), "Intelligent Transport Systems (ITS); User and applications requirements; Part 2: Applications and facilities layer common data dictionary," August 2018.

[16] C-ITS Deployment Platform, https://ec.europa.eu/transport/themes/its/c-its_en

[17] ETSI EN 302 636-5-1 V2.2.1 (2019-05), "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol," May 2019.

[18] ETSI EN 302 637-2 V1.4.1 (2019-04), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," April 2019.

[19] ETSI EN 302 637-3 V1.3.1 (2019-04), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decentralized Environmental Notification Basic Service," April 2019.

[20] COMMISSION IMPLEMENTING DECISION (EU) 2020/1426 of 7 October 2020 on the harmonised use of radio spectrum in the 5 875-5 935 MHz frequency band for safety-related applications of intelligent transport systems (ITS).

[21] ECC decision (08)01, "The harmonised use of the 5875-5925 MHz frequency band for Intelligent Transport Systems (ITS)," approved 14 March 2008, latest amendment on 6 March 2020.

[22] ECC recommendation (08)01, "Use of the band 5855-5875 MHz for Intelligent Transport Systems (ITS), approved 21 February, latest amendment on 6 March 2020.

[23] ETSI TS 103 248 V1.3.1 (2019-04), “Intelligent Transport Systems (ITS); GeoNetworking; Port numbers for Basic Transport Protocol (BTP),” April 2019.

[24] ETSI TS 103 301 V1.2.1 (2018-08), “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services,” August 2018.

[25] ISO TS 14823:2017, “Intelligent transport systems – Graphic Data Dictionary,” 2017.

[26] ENSEMBLE project Deliverable D2.9, “Security framework of platooning,” January 2022. https://platooningensemble.eu/library.

[27] ENSEMBLE project Deliverable D2.3, “V2 Platooning use cases, scenario definition and Platooning Levels,” January 2022. https://platooningensemble.eu/library.

[28] ENSEMBLE project Deliverable D2.5, “Final Version Functional specification for white-label truck” January 2022. https://platooningensemble.eu/library.

[29] ENSEMBLE project Deliverable D2.6, “Functional specification for intelligent infrastructure - Strategic and Services Layers,” January 2022. https://platooningensemble.eu/library.

[30] ENSEMBLE project Deliverable D2.7, “V2 Functional specification for intelligent Infrastructure (Strategic/Services layers),” January 2022. https://platooningensemble.eu/library.

[31] ENSEMBLE project Deliverable D2.13, “SOTIF Safety Concept” January 2022. https://platooningensemble.eu/library.

[32] ENSEMBLE project Deliverable D2.14, Final version Hazard Analysis and Risk Assessment and Functional Safety Concept,” January 2022. https://platooningensemble.eu/library.

[33] ENSEMBLE project Deliverable D2.15, “Final version of Iterative development process and Item Definition” January 2022. https://platooningensemble.eu/library.

[34] ENSEMBLE project Deliverable D6.15, “ENSEMBLE standardisation process” March 2022.

[35] ISO 17419 “Intelligent transport systems — Cooperative systems — Globally unique identification“, May 2018.

7. APPENDIX A - GLOSSARY

7.1. Definitions

Term	Definition
Convoy	A truck platoon may be defined as trucks that travel together in convoy formation at a fixed gap distance typically less than 1 second apart up to 0.3 seconds. The vehicles closely follow each other using wireless vehicle-to-vehicle (V2V) communication and advanced driver assistance systems.
Cut-in	A lane change manoeuvre performed by vehicles from the adjacent lane to the ego vehicle's lane, at a distance close enough (i.e., shorter than desired inter vehicle distance) relative to the ego vehicle.
Cut-out	A lane change manoeuvre performed by vehicles from the ego lane to the adjacent lane.
Cut-through	A lane change manoeuvre performed by vehicles from the adjacent lane (e.g., left lane) to ego vehicle's lane, followed by a lane change manoeuvre to the other adjacent lane (e.g., right lane).
Ego Vehicle	The vehicle from which the perspective is considered.
Emergency brake	Brake action with an acceleration of < -4 m/s2
Event	An event marks the time instant at which a transition of a state occurs, such that before and after an event, the system is in a different mode.
Following truck	Each truck that is following behind a member of the platoon, being every truck except the leading and the trailing truck, when the system is in platoon mode.
Leading truck	The first truck of a truck platoon
Legal Safe Gap	Minimum allowed elapsed time/distance to be maintained by a standalone truck while driving according to Member States regulation (it could be 2 seconds, 50 meters or not present)
Manoeuvre ("activity")	A particular (dynamic) behaviour which a system can perform (from a driver or other road user perspective) and that is different from standing still, is being considered a manoeuvre.
ODD (operational design domain)	The ODD should describe the specific conditions under which a given automation function is intended to function. The ODD is the definition of where (such as what roadway types and speeds) and when (under what conditions, such as day/night, weather limits, etc.) an automation function is designed to operate.

Term	Definition
Operational layer	The operational layer involves the vehicle actuator control (e.g., accelerating/braking, steering), the execution of the manoeuvres, and the control of the individual vehicles in the platoon to automatically perform the platooning task. Here, the main control task is to control the inter-vehicle distance or velocity and, depending on the Platooning Level, the lateral position relative to the lane or to the preceding vehicle. Key performance requirements for this layer are vehicle following behaviour (longitudinal and lateral) and string stability of the platoon, where the latter is a necessary requirement to achieve a stable traffic flow and to achieve scalability with respect to platoon length, and the short-range wireless inter-vehicle communication is the key enabling technology.
Platoon	A group of two or more automated cooperative vehicles in line, maintaining a close distance, typically such a distance to reduce fuel consumption by air drag, to increase traffic safety by use of additional ADAS-technology, and to improve traffic throughput because vehicles are driving closer together and take up less space on the road.
Platoon Automation Levels	In analogy with the SAE automation levels subsequent platoon automation levels will incorporate an increasing set of automation functionalities, up to and including full vehicle automation in a multi-brand platoon in real traffic for the highest Platooning Automation Level. The definition of “platooning levels of automation” will comprise of elements such as the minimum time gap between the vehicles, whether there is lateral automation available or not, driving speed range, operational areas (e.g., highway).
Platoon candidate	A truck who intends to engage the platoon either from the front or the back of the platoon.
Platoon cohesion	Platoon cohesion refers to how well the members of the platoon remain within steady state conditions in various scenario conditions (e.g., slopes, speed changes).
Platoon disengaging	The ego vehicle decides to disengage from the platoon itself or is requested by another member of the platoon to do so. When conditions are met the ego vehicle starts to increase the gap between the trucks to a safe non-platooning gap. The disengaging is completed when the gap is large enough (e.g. time gap of 1.5 seconds, which is depends on the operational safety based on vehicle dynamics and human reaction times is given). This is the same as leaving the platoon.
Platoon dissolve	All trucks are disengaging the platoon at the same time. A.k.a. decoupling, a.k.a. disassemble.
Platoon engaging	Using wireless communication (V2V), the Platoon Candidate sends an engaging request. When conditions are met the system starts to decrease the time gap between the trucks to the platooning time gap. This is the same as joining the platoon.

Term	Definition
Platoon formation	Platoon formation is the process before platoon engaging in which it is determined if and in what format (e.g., composition) trucks can/should become part of a new / existing platoon. Platoon formation can be done on the fly, scheduled or a mixture of both. Platoon candidates may receive instructions during platoon formation (e.g., to adapt their velocity, to park at a certain location) to allow the start of the engaging procedure of the platoon.
Platoon split	The platoon is split in 2 new platoons who themselves continue as standalone entities.
Requirements	Description of system properties. Details of how the requirements shall be implemented at system level
Scenario	A scenario is a quantitative description of the ego vehicle, its activities and/or goals, its static environment, and its dynamic environment. From the perspective of the ego vehicle, a scenario contains all relevant events. A scenario is a combination of a manoeuvre (“activity”), ODD and events.
Service layer	The service layer represents the platform on which logistical operations and new initiatives can operate.
Steady state	In systems theory, a system or a process is in a steady state if the variables (called state variables) which define the behaviour of the system or the process are not changed in the time domain. In the context of platooning, this means that the relative velocity and gap between trucks is unchanged within tolerances from the system parameters.
Strategic layer	The strategic layer is responsible for the high-level decision-making regarding the scheduling of platoons based on vehicle compatibility and Platooning Level, optimized with respect to fuel consumption, travel times, destination, and impact on highway traffic flow and infrastructure, employing cooperative ITS cloud-based solutions. In addition, the routing of vehicles to allow for platoon forming is included in this layer. The strategic layer is implemented in a centralised fashion in so-called traffic control centres. Long-range wireless communication by existing cellular technology is used between a traffic control centre and vehicles/platoons and their drivers.
Tactical layer	The tactical layer coordinates the actual platoon forming (both from the tail of the platoon and through merging in the platoon) and platoon dissolution. In addition, this layer ensures platoon cohesion on hilly roads, and sets the desired platoon velocity, inter-vehicle distances (e.g., to prevent damaging bridges) and lateral offsets to mitigate road wear. This is implemented through the execution of an interaction protocol using the short-range wireless inter-vehicle communication (i.e., V2X). In fact, the interaction protocol is implemented by message sequences, initiating the

Term	Definition
	manoeuvres that are necessary to form a platoon, to merge into it, or to dissolve it, also considering scheduling requirements due to vehicle compatibility.
Target Time Gap	Elapsed time to cover the inter-vehicle distance by a truck indicated in seconds, agreed by all the Platoon members; it represents the minimum distance in seconds allowed inside the Platoon.
Time gap	Elapsed time to cover the inter vehicle distance by a truck indicated in seconds.
Trailing truck	The last truck of a truck platoon
Truck Platoon	Description of system properties. Details of how the requirements shall be implemented at system level
Use case	Use cases describe how a system shall respond under various conditions to interactions from the user of the system or surroundings, e.g., other traffic participants or road conditions. The user is called actor on the system and is often but not always a human being. In addition, the use case describes the response of the system towards other traffic participants or environmental conditions. The use cases are described as a sequence of actions, and the system shall behave according to the specified use cases. The use case often represents a desired behaviour or outcome. In the ENSEMLE context, a use case is an extension of scenario which add more information regarding specific internal system interactions, specific interactions with the actors (e.g., driver, I2V) and will add different flows (normal & alternative, e.g., successful and failed in relation to activation of the system / system elements).

7.2. Acronyms and abbreviations

Acronym / Abbreviation	Meaning
ACC	Adaptive Cruise Control
ADAS	Advanced driver assistance system
AEB	Autonomous Emergency Braking (System, AEBS)
ASIL	Automotive Safety Integrity Level
ASN.1	Abstract Syntax Notation One
BTP	Basic Transport Protocol
C-ACC	Cooperative Adaptive Cruise Control

Acronym / Abbreviation	Meaning
C-ITS	Cooperative ITS
CA	Cooperative Awareness
CAD	Connected Automated Driving
CAM	Cooperative Awareness Message
CCH	Control Channel
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Message
DITL	Driver-In-the-Loop
DOOTL	Driver-Out-Of-the Loop
DSRC	Dedicated Short-Range Communications
ETSI	European Telecommunications Standards Institute
EU	European Union
FCW	Forward Collision Warning
FLC	Forward Looking Camera
FSC	Functional Safety Concept
GN	GeoNetworking
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
HARA	Hazard Analysis and Risk Assessment
HIL	Hardware-in-the-Loop
HMI	Human Machine Interface
HW	Hardware
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITL	In-The_Loop
ITS	Intelligent Transport System

Acronym / Abbreviation	Meaning
IVI	Infrastructure to Vehicle Information message
LDWS	Lane Departure Warning System
LKA	Lane Keeping Assist
LCA	Lane Centring Assist
LRR	Long Range Radar
LSG	Legal Safe Gap
MAP	MapData message
MIO	Most Important Object
MRR	Mid Range Radar
OS	Operating system
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OOTL	Out-Of The-Loop
PAEB	Platooning Autonomous Emergency Braking
PCM	Platooning Control Message
PDU	Packet Data Unit
PMC	Platooning Mode Control
PMM	Platooning Management Message
QM	Quality Management
RSU	Road Side Unit
SA	Situation Awareness
SAE	SAE International, formerly the Society of Automotive Engineers
SCH	Service Channel
SDO	Standard Developing Organizations
SIL	Software-in-the-Loop
SPAT	Signal Phase and Timing message
SRR	Short Range Radar
SW	Software

Acronym / Abbreviation	Meaning
TC	Technical Committee
TOR	Take-Over Request
TOT	Take-Over Time
TTG	Target Time Gap
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to any (where x equals either vehicle or infrastructure)
VDA	Verband der Automobilindustrie (German Association of the Automotive Industry)
WIFI	Wireless Fidelity
WLAN	Wireless Local Area Network
WP	Work Package

8. APPENDIX B – PROTOCOL SPECIFICS

8.1. New data types

Only new data types related to the platooning protocol are being described in the present appendix. For all other data types that are not listed here are found in ETSI TS 102 894-2 [15].

8.1.1. DE_DistanceToVehicleAhead

Descriptive Name	DistanceToVehicleAhead
ASN.1 representation	DistanceToVehicleAhead ::= INTEGER {oneCm(1), unavailable(16383)} (0..16383)
Definition	
Unit	cm
Category	Platoon information

8.1.2. DE_GrossCombinationVehicleWeight

Descriptive Name	GrossCombinationVehicleWeight
ASN.1 representation	GrossCombinationVehicleWeight ::= INTEGER {tenKg(1), outOfRange(32766), unavailable(32767)} (0..32767)
Definition	
Unit	10 kg
Category	Vehicle information

8.1.3. DE_HdAccelerationConfidence

Descriptive Name	HdAccelerationConfidence
ASN.1 representation	HdAccelerationConfidence ::= INTEGER { pointZeroOneMeterPerSecSquared(1), outOfRange(1022), unavailable(1023)} (0 .. 1023)
Definition	
Unit	0.01m/s2
Category	Vehicle information

8.1.4. DE_LaneMarkingDistance

Descriptive Name	LaneMarkingDistance
ASN.1 representation	LaneMarkingDistance ::= INTEGER {oneCm(1), unavailable(511)} (0..511)
Definition	
Unit	cm

Category	Vehicle information

8.1.5. DE_LongitudinalHdAccelerationValue

Descriptive Name	LongitudinalHdAccelerationValue
ASN.1 representation	LongitudinalHdAccelerationValue ::= INTEGER { pointZeroOneMeterPerSecSquaredForward(1), pointZeroOneMeterPerSecSquaredBackward(-1), unavailable(1610)} (-1600 .. 1610)
Definition	
Unit	0.01m/s2
Category	Vehicle information

8.1.6. DE_PlatoonID

Descriptive Name	PlatoonID
ASN.1 representation	PlatoonID ::= OCTET STRING (SIZE(16))
Definition	The PlatoonID identifies a platoon within a geographical area. When a new platoon is formed the vehicle in front must generate the PlatoonID and provide it to the joining truck in the `JoinResponse` message. The PlatoonID has to be re-generated when the leading vehicle in a platoon changes (e.g., the leading vehicle is leaving, a split of the platoon, etc.). ENSEMBLE specific solution BBBMMddHHmmssSSS. • BBB = Brand (DAI, DAF, IVE, MAN, REN, SCA,VOL) • MM = month in current year as number • dd = day in current month as number • HH = hour of the day in 24h scheme • mm = minute of current hour • ss = second of current minute • SSS = milliseconds
Unit	N/A
Category	Platoon information

8.1.7. DE_PlatoonPosition

Descriptive Name	PlatoonPosition
ASN.1 representation	PlatoonPosition ::= INTEGER {leader(1), firstFollower(2), unavailable(32)} (1..32)
Definition	The ego trucks position in the platoon.
Unit	N/A
Category	Platoon information

8.1.8. DE_PowerToMassRatio

Descriptive Name	PowerToMassRatio
ASN.1 representation	PowerToMassRatio ::= INTEGER { oneWperKg(1), outOfRange(255), unavailable(256)} (1 .. 256)
Definition	
Unit	1W/kg
Category	Vehicle information

8.1.9. DE_RoadInclination

Descriptive Name	RoadInclination
ASN.1 representation	RoadInclination ::= INTEGER {pointOnePercentUp(1), pointOnePercentDown(-1), unavailable(128)} (-127 .. 128)
Definition	
Unit	1%
Category	Vehicle information

8.1.10. DE_VehicleID

Descriptive Name	VehicleID
ASN.1 representation	VehicleID ::= IA5String (SIZE (11..20))
Definition	For ENSEMBLE the syntax of BBBTRUCKXXXX was selected. • BBB, 3 letters of the brands name (DAI, DAF, IVE, MAN, REN, SCA,VOL) • XXXX stand for the last 4 digits of VIN Example: **SCATRUCK0815**
Unit	N/A
Category	Vehicle information

8.1.11. DE_ReasonForSpeedOrGapAdjustment

Descriptive Name	ReasonForSpeedOrGapAdjustment
ASN.1 representation	ReasonForSpeedOrGapAdjustment ::= INTEGER { unknown(0), safety(1), efficiency(2), trafficAhead(3), intruder(4), emergency(5), leave(6), cohesion(7) }
Definition	Provides the reason for speed or gap adjustment.
Unit	N/A
Category	Vehicle information

8.2. Sequence diagram sources

All sequence diagrams were produced in https://www.websequencediagrams.com/. In the present chapter the source code is provided.

8.2.1. Join

```
participant V1
participant V0

V1->V1: Identify Front

alt V1_has_followers = FALSE
    V1->V1: Set CAM.isJoinable = FALSE
end

loop REQ_CNT < MAX_CNT, REQ_TIMEOUT
    V1->V1: Generate ResponseKey
    V1-->V0: JoinRequest (receiver = V0, Enc = none, [ResponseKey, noOfTrucks])

V0->V0: Check request (e.g., mergedNoOfTrucks > maxNoOfTrucks)

alt JoinRequest_accepted = TRUE
    V0-->V1: JoinResponse (receiver = V1, Enc = ResponseKey, [PGK1, PPK1])
    alt V0_has_forerunner = TRUE
        V0->V0: PlatooningState = PlatoonFollower
    else
        V0->V0: Start broadcasting PCM (enc = PGK1)
        V0->V0: PlatooningState = PlatoonLeader
    end
    V0->V0: Set CAM.isJoinable = FALSE
    V0->V0: Start JoinTimeout
    alt V1_has_followers = TRUE
        V1->V1: KeyUpdate (forward PGK1, platoon position\n and PlatoonID from
JoinResponse to followers)
        V1->V1: Wait for PCM from follower (enc = PGK1)
    end
    V1->V1: Broadcasting PCM (enc = PGK1)
```

```
    alt PCM_V1 = FALSE && JoinTimeout = 0
        V0->V0: Enter BackSplit
        alt V0_has_forerunner = TRUE
            V0->V0: PlatooningState = PlatoonFollower
        else
            V0->V0: PlatooningState = PlatoonSingle
        end
        V1->V1: PlatooningState = PlatoonLeader
    else
        V1->V1: PlatooningState = PlatoonFollower
    end

else JoinRequest_accepted = FALSE
    V0-->V1: JoinResponse (receiver = V1, Enc = ERKey, Null)
    V1->V1: Blacklist vehicle V0
end
end

alt V1_has_follower = FALSE
    V1->V1: Set CAM.isJoinable=True
end
```

8.2.2. Front split

```
note left of V0: Broadcast PCMs
V0->V0: Set preparingForFrontSplit
V0->V0: Wait for control to indicate standalone distance
V0->V0: Set preparedForFrontSplit for\nPREPARED_FOR_FRONT_SPLIT_INDICATION_TIME
note left of V0: Drive away

alt HasFollowers = TRUE
    V0->V0: Execute PlatoonUpdate
else
    V0->V0: Set CAM.isJoinable = TRUE
    V0->V0: Enter standalone
end
```

8.2.3. Back split

```
participant V1
participant V0

note left of V1: NeighborWatchdog always running
V0->V0: Set requestBackSplit
V1->V1: Enter FrontSplit
V0->V0: Received preparedForFrontSplit
V0->V0: Set isJoinable = TRUE

alt V0 becomes standalone
    V0->V0: Stop transmitting PCMs
```

```
end
```

8.2.4. Platoon update

```
participant V3
participant V2
participant V1
participant V0

loop NOW-LASTRUN >= 10 (secs)
loop while platooningState = platooning
    V0->V0: Get key age (PGK0)
    alt CUR_KEY_AGE > 60 (secs) = TRUE
        V0->V0: Generate PGK1
        V0->V0: Generate PPK [V0/V1]
        V0-->V1: Transmit PlatoonUpdate (PGK1, PPK [V0/V1])
        V1->V1: Generate PPK [V1/V2]
        V1-->V2: Transmit PlatoonUpdate (PGK1, PPK [V1/V2])
        V2->V2: Generate PPK [V2/V3]
        V2-->V3: Transmit PlatoonUpdate (PGK1, PPK [V2/V3])
    end
end
```

8.2.5. Neighbour watchdog

```
participant V2
participant V1
participant V0

loop while platooningState = platooning
    V0->V0: Broadcast PCMs
    V1->V1: Broadcast PCMs
    V2->V2: Broadcast PCMs
opt PCMs from V0 not received within PCM_TIMEOUT

    V1->V1: Enter front split
end
opt PCMs from V2 not received within PCM_TIMEOUT

    V1->V1: Enter back split

end
end
```

8.2.6. Merge

```
note left of StandAlone: StandAlone, will be joined from another platoon
note left of OldLeader: Already in a platoon with Back
OldLeader->StandAlone: JoinRequest(to=Trailing, enc=none, ERKey, no=2)
StandAlone-->OldLeader: JoinResponse (to=OldLeader, enc=ERKey, PGK1,
PlatoonPosition=2)
```

```
OldLeader->+OldLeader: Block decreasing distance
OldLeader->Back: KeyUpdate (enc=PPK[OldLeader/Back], PGK1, PlatoonPosition=3)
Back->Back: Use PGK1
OldLeader->-OldLeader: Use PGK1
```

8.2.7. Leave middle

```
note left of Back: All have NeighborWatchdig running
note left of Ego: Going to leave
note left of Front: Continues platooning

Ego->Ego: PCM (Enc=PGK1, [SplitStatus=(preparingForFrontSplit, requestBackSplit)])
Back->Back: PCM (Enc=PGK1, [SplitStatus=(preparingForFrontSplit)])
Ego->Ego: Increase distance
Back->Back: Increase distance
Ego->+Ego: PCM (Enc=PGK1, [SplitStatus=(preparedForFrontSplit, requestBackSplit)])
note left of Ego:  PREPARED_FOR_FRONT_SPLIT_INDICATION_TIME
Front->Front: Received preparedForFrontSplit
alt NewTrailing=True
    Front->Front: Set isJoinable=True
else
    Front->Front: Set PlatooningState=StandAlone
end
Ego->-Ego: PCM (Enc=PGK1, [SplitStatus=(preparedForFrontSplit, requestBackSplit)])

Back->+Back: PCM (Enc=PGK1, [SplitStatus=(preparedForFrontSplit)])
note left of Back:  PREPARED_FOR_FRONT_SPLIT_INDICATION_TIME
Ego->Ego: Received preparedForFrontSplit
Ego->Ego: Set PlatooningState=StandAlone
Back->-Back: PCM (Enc=PGK1, [SplitStatus=(unpreparedForFrontSplit)])
note left of Ego: Drive out

alt NewLeader=True
    Back->Back: Enter KeyUpdateSimplified
else
    Back->Back: Set PlatooningState=StandAlone
end
```

8.3. ASN.1 Schemes

8.3.1. CAM

```
CAM-PDU-Descriptions {
itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) en (302637) cam
(2) version (1)
}

DEFINITIONS AUTOMATIC TAGS ::=
```

```
BEGIN

IMPORTS
ItsPduHeader, CauseCode, ReferencePosition, AccelerationControl, Curvature,
CurvatureCalculationMode, Heading, LanePosition, EmergencyPriority, EmbarkationStatus,
Speed, DriveDirection, LongitudinalAcceleration, LateralAcceleration,
VerticalAcceleration, StationType, ExteriorLights, DangerousGoodsBasic,
SpecialTransportType, LightBarSirenInUse, VehicleRole, VehicleLength, VehicleWidth,
PathHistory, RoadworksSubCauseCode, ClosedLanes, TrafficRule, SpeedLimit,
SteeringWheelAngle, PerformanceClass, YawRate, ProtectedCommunicationZone,
PtActivation, Latitude, Longitude, ProtectedCommunicationZonesRSU, CenDsrcTollingZone
FROM ITS-Container {
itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) ts (102894) cdd
(2) version (1)
};

--  The root data frame for cooperative awareness messages

CAM ::= SEQUENCE {
       header       ItsPduHeader,
       cam          CoopAwareness
}

CoopAwareness ::= SEQUENCE {
       generationDeltaTime        GenerationDeltaTime,
       camParameters              CamParameters
}

CamParameters ::= SEQUENCE {
       basicContainer             BasicContainer,
       highFrequencyContainer     HighFrequencyContainer,
       lowFrequencyContainer      LowFrequencyContainer OPTIONAL,
       specialVehicleContainer    SpecialVehicleContainer OPTIONAL,
       ...,
       platooningContainer        PlatooningContainer OPTIONAL
}

PlatooningContainer ::= SEQUENCE {
    isJoinable BOOLEAN,
    ...,
    messageContainer I2VMessageContainer OPTIONAL -- I2V ENSEMBLE support
}

-- I2V ENSEMBLE support, BEGIN
I2VMessageContainer ::= SEQUENCE {
       position            ReferencePosition,
       maxRecSpeed         SpeedLimit OPTIONAL,
       minRecInterdistance MinInterdistance OPTIONAL
}
-- I2V ENSEMBLE support, END

HighFrequencyContainer ::= CHOICE {
```

```
       basicVehicleContainerHighFrequency      BasicVehicleContainerHighFrequency,
       rsuContainerHighFrequency        RSUContainerHighFrequency,
       ...
}

 LowFrequencyContainer ::= CHOICE {
    basicVehicleContainerLowFrequency BasicVehicleContainerLowFrequency,
    ...
}

SpecialVehicleContainer ::= CHOICE {
       publicTransportContainer   PublicTransportContainer,
       specialTransportContainer  SpecialTransportContainer,
       dangerousGoodsContainer    DangerousGoodsContainer,
       roadWorksContainerBasic    RoadWorksContainerBasic,
       rescueContainer            RescueContainer,
       emergencyContainer         EmergencyContainer,
       safetyCarContainer         SafetyCarContainer,
       ...
}

BasicContainer ::= SEQUENCE {
       stationType         StationType,
       referencePosition   ReferencePosition,
       ...
}

BasicVehicleContainerHighFrequency ::= SEQUENCE {
       heading                    Heading,
       speed                      Speed,
       driveDirection             DriveDirection,
       vehicleLength              VehicleLength,
       vehicleWidth               VehicleWidth,
       longitudinalAcceleration   LongitudinalAcceleration,
       curvature                  Curvature,
       curvatureCalculationMode   CurvatureCalculationMode,
       yawRate                    YawRate,
       accelerationControl        AccelerationControl OPTIONAL,
       lanePosition               LanePosition OPTIONAL,
       steeringWheelAngle         SteeringWheelAngle OPTIONAL,
       lateralAcceleration        LateralAcceleration OPTIONAL,
       verticalAcceleration       VerticalAcceleration OPTIONAL,
       performanceClass           PerformanceClass OPTIONAL,
       cenDsrcTollingZone         CenDsrcTollingZone OPTIONAL
}

BasicVehicleContainerLowFrequency  ::= SEQUENCE {
       vehicleRole         VehicleRole,
       exteriorLights      ExteriorLights,
       pathHistory         PathHistory
}
```

```
PublicTransportContainer ::= SEQUENCE {
       embarkationStatus   EmbarkationStatus,
       ptActivation        PtActivation OPTIONAL
}

SpecialTransportContainer ::= SEQUENCE {
       specialTransportType      SpecialTransportType,
       lightBarSirenInUse        LightBarSirenInUse
}

 DangerousGoodsContainer ::= SEQUENCE {
       dangerousGoodsBasic       DangerousGoodsBasic
 }

 RoadWorksContainerBasic ::= SEQUENCE {
       roadworksSubCauseCode     RoadworksSubCauseCode OPTIONAL,
       lightBarSirenInUse        LightBarSirenInUse,
       closedLanes               ClosedLanes OPTIONAL
 }

RescueContainer ::= SEQUENCE {
       lightBarSirenInUse  LightBarSirenInUse
}

EmergencyContainer ::= SEQUENCE {
       lightBarSirenInUse  LightBarSirenInUse,
       incidentIndication  CauseCode OPTIONAL,
       emergencyPriority   EmergencyPriority OPTIONAL
}

SafetyCarContainer ::= SEQUENCE {
       lightBarSirenInUse  LightBarSirenInUse,
       incidentIndication  CauseCode OPTIONAL,
       trafficRule         TrafficRule OPTIONAL,
       speedLimit          SpeedLimit OPTIONAL
}

RSUContainerHighFrequency ::= SEQUENCE {
    protectedCommunicationZonesRSU ProtectedCommunicationZonesRSU OPTIONAL,
    ...
}

GenerationDeltaTime ::= INTEGER { oneMilliSec(1) } (0..65535)

END
```

8.3.2. ITSContainer

```
ITS-Container {
```

```
itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) ts (102894) cdd
(2) version (1)
}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

ItsPduHeader ::= SEQUENCE {
    protocolVersion INTEGER{currentVersion(1)} (0..255),
    messageID INTEGER{ denm(1), cam(2), poi(3), spatem(4), mapem(5), ivim(6), ev-
rsr(7), tistpgtransaction(8), srem(9), ssem(10), evcsn(11), pmm(13), pcm(14), pim
(15)} (0..255),  -- Mantis #7209, #7005
    stationID StationID
}

StationID ::= INTEGER(0..4294967295)

ReferencePosition ::= SEQUENCE {
    latitude                     Latitude,
    longitude                    Longitude,
    positionConfidenceEllipse    PosConfidenceEllipse ,
    altitude                     Altitude
}

DeltaReferencePosition ::= SEQUENCE {
    deltaLatitude   DeltaLatitude,
    deltaLongitude  DeltaLongitude,
    deltaAltitude   DeltaAltitude
}

Longitude ::= INTEGER {oneMicrodegreeEast (10), oneMicrodegreeWest (-10),
unavailable(1800000001)} (-1800000000..1800000001)

Latitude ::= INTEGER {oneMicrodegreeNorth (10), oneMicrodegreeSouth (-10),
unavailable(900000001)} (-900000000..900000001)

Altitude ::= SEQUENCE {
    altitudeValue          AltitudeValue,
    altitudeConfidence     AltitudeConfidence
}

AltitudeValue ::= INTEGER {referenceEllipsoidSurface(0), oneCentimeter(1),
unavailable(800001)} (-100000..800001)

AltitudeConfidence ::= ENUMERATED {
    alt-000-01 (0),
    alt-000-02 (1),
    alt-000-05 (2),
    alt-000-10 (3),
    alt-000-20 (4),
    alt-000-50 (5),
```

```
    alt-001-00 (6),
    alt-002-00 (7),
    alt-005-00 (8),
    alt-010-00 (9),
    alt-020-00 (10),
    alt-050-00 (11),
    alt-100-00 (12),
    alt-200-00 (13),
    outOfRange (14),
    unavailable (15)
}

DeltaLongitude ::= INTEGER {oneMicrodegreeEast (10), oneMicrodegreeWest (-10),
unavailable(131072)} (-131071..131072)

DeltaLatitude ::= INTEGER {oneMicrodegreeNorth (10), oneMicrodegreeSouth (-10) ,
unavailable(131072)} (-131071..131072)

DeltaAltitude ::= INTEGER {oneCentimeterUp (1), oneCentimeterDown (-1),
unavailable(12800)} (-12700..12800)

PosConfidenceEllipse ::= SEQUENCE {
    semiMajorConfidence    SemiAxisLength,
    semiMinorConfidence    SemiAxisLength,
    semiMajorOrientation   HeadingValue
}

PathPoint ::= SEQUENCE {
    pathPosition    DeltaReferencePosition,
    pathDeltaTime   PathDeltaTime OPTIONAL
}

PathDeltaTime ::= INTEGER {tenMilliSecondsInPast(1)} (1..65535, ...)

PtActivation ::= SEQUENCE {
    ptActivationType PtActivationType,
    ptActivationData PtActivationData
}

PtActivationType ::= INTEGER {undefinedCodingType(0), r09-16CodingType(1), vdv-
50149CodingType(2)} (0..255)

PtActivationData ::= OCTET STRING (SIZE(1..20))

AccelerationControl ::= BIT STRING {
    brakePedalEngaged (0),
    gasPedalEngaged (1),
    emergencyBrakeEngaged (2),
    collisionWarningEngaged (3),
    accEngaged (4),
    cruiseControlEngaged (5),
    speedLimiterEngaged (6)
```

```
} (SIZE(7))


SemiAxisLength ::= INTEGER{oneCentimeter(1), outOfRange(4094), unavailable(4095)}
(0..4095)

CauseCode ::= SEQUENCE {
    causeCode       CauseCodeType,
    subCauseCode    SubCauseCodeType
}

CauseCodeType ::= INTEGER {
    reserved (0),
    trafficCondition (1),
    accident (2),
    roadworks (3),
    adverseWeatherCondition-Adhesion (6),
    hazardousLocation-SurfaceCondition (9),
    hazardousLocation-ObstacleOnTheRoad (10),
    hazardousLocation-AnimalOnTheRoad (11),
    humanPresenceOnTheRoad (12),
    wrongWayDriving (14),
    rescueAndRecoveryWorkInProgress (15),
    adverseWeatherCondition-ExtremeWeatherCondition (17),
    adverseWeatherCondition-Visibility (18),
    adverseWeatherCondition-Precipitation (19),
    slowVehicle (26),
    dangerousEndOfQueue (27),
    vehicleBreakdown (91),
    postCrash (92),
    humanProblem (93),
    stationaryVehicle (94),
    emergencyVehicleApproaching (95),
    hazardousLocation-DangerousCurve (96),
    collisionRisk (97),
    signalViolation (98),
    dangerousSituation (99)
} (0..255)

SubCauseCodeType ::= INTEGER (0..255)

TrafficConditionSubCauseCode ::= INTEGER {unavailable(0), increasedVolumeOfTraffic(1),
trafficJamSlowlyIncreasing(2), trafficJamIncreasing(3),
trafficJamStronglyIncreasing(4), trafficStationary(5),
trafficJamSlightlyDecreasing(6), trafficJamDecreasing(7),
trafficJamStronglyDecreasing(8)} (0..255)

AccidentSubCauseCode ::= INTEGER {unavailable(0), multiVehicleAccident(1),
heavyAccident(2), accidentInvolvingLorry(3), accidentInvolvingBus(4),
accidentInvolvingHazardousMaterials(5), accidentOnOppositeLane(6),
unsecuredAccident(7), assistanceRequested(8)} (0..255)
```

```
RoadworksSubCauseCode ::= INTEGER {unavailable(0), majorRoadworks(1),
roadMarkingWork(2), slowMovingRoadMaintenence(3), shortTermStationaryRoadworks(4),
streetCleaning(5), winterService(6)} (0..255)

HumanPresenceOnTheRoadSubCauseCode ::= INTEGER {unavailable(0), childrenOnRoadway(1),
cyclistOnRoadway(2), motorcyclistOnRoadway(3)} (0..255)

WrongWayDrivingSubCauseCode ::= INTEGER {unavailable(0), wrongLane(1),
wrongDirection(2)} (0..255)

AdverseWeatherCondition-ExtremeWeatherConditionSubCauseCode ::= INTEGER
{unavailable(0), strongWinds(1), damagingHail(2), hurricane(3), thunderstorm(4),
tornado(5), blizzard(6)} (0..255)

AdverseWeatherCondition-AdhesionSubCauseCode ::= INTEGER {unavailable(0),
heavyFrostOnRoad(1), fuelOnRoad(2), mudOnRoad(3), snowOnRoad(4), iceOnRoad(5),
blackIceOnRoad(6), oilOnRoad(7), looseChippings(8), instantBlackIce(9),
roadsSalted(10)} (0..255)

AdverseWeatherCondition-VisibilitySubCauseCode ::= INTEGER {unavailable(0), fog(1),
smoke(2), heavySnowfall(3), heavyRain(4), heavyHail(5), lowSunGlare(6), sandstorms(7),
swarmsOfInsects(8)} (0..255)

AdverseWeatherCondition-PrecipitationSubCauseCode ::= INTEGER {unavailable(0),
heavyRain(1), heavySnowfall(2), softHail(3)} (0..255)

SlowVehicleSubCauseCode ::= INTEGER {unavailable(0), maintenanceVehicle(1),
vehiclesSlowingToLookAtAccident(2), abnormalLoad(3), abnormalWideLoad(4), convoy(5),
snowplough(6), deicing(7), saltingVehicles(8)} (0..255)

StationaryVehicleSubCauseCode ::= INTEGER {unavailable(0), humanProblem(1),
vehicleBreakdown(2), postCrash(3), publicTransportStop(4), carryingDangerousGoods(5)}
(0..255)

HumanProblemSubCauseCode ::= INTEGER {unavailable(0), glycemiaProblem(1),
heartProblem(2)} (0..255)

EmergencyVehicleApproachingSubCauseCode ::= INTEGER {unavailable(0),
emergencyVehicleApproaching(1), prioritizedVehicleApproaching(2)} (0..255)

HazardousLocation-DangerousCurveSubCauseCode ::= INTEGER {unavailable(0),
dangerousLeftTurnCurve(1), dangerousRightTurnCurve(2),
multipleCurvesStartingWithUnknownTurningDirection(3),
multipleCurvesStartingWithLeftTurn(4), multipleCurvesStartingWithRightTurn(5)}
(0..255)

HazardousLocation-SurfaceConditionSubCauseCode ::= INTEGER {unavailable(0),
rockfalls(1), earthquakeDamage(2), sewerCollapse(3), subsidence(4), snowDrifts(5),
stormDamage(6), burstPipe(7), volcanoEruption(8), fallingIce(9)} (0..255)
```

```
HazardousLocation-ObstacleOnTheRoadSubCauseCode ::= INTEGER {unavailable(0),
shedLoad(1), partsOfVehicles(2), partsOfTyres(3), bigObjects(4), fallenTrees(5),
hubCaps(6), waitingVehicles(7)} (0..255)

HazardousLocation-AnimalOnTheRoadSubCauseCode ::= INTEGER {unavailable(0),
wildAnimals(1), herdOfAnimals(2), smallAnimals(3), largeAnimals(4)} (0..255)

CollisionRiskSubCauseCode ::= INTEGER {unavailable(0), longitudinalCollisionRisk(1),
crossingCollisionRisk(2), lateralCollisionRisk(3), vulnerableRoadUser(4)} (0..255)

SignalViolationSubCauseCode ::= INTEGER {unavailable(0), stopSignViolation(1),
trafficLightViolation(2), turningRegulationViolation(3)} (0..255)

RescueAndRecoveryWorkInProgressSubCauseCode ::= INTEGER {unavailable(0),
emergencyVehicles(1), rescueHelicopterLanding(2), policeActivityOngoing(3),
medicalEmergencyOngoing(4), childAbductionInProgress(5)} (0..255)

DangerousEndOfQueueSubCauseCode ::= INTEGER {unavailable(0), suddenEndOfQueue(1),
queueOverHill(2), queueAroundBend(3), queueInTunnel(4)} (0..255)

DangerousSituationSubCauseCode ::= INTEGER {unavailable(0),
emergencyElectronicBrakeEngaged(1), preCrashSystemEngaged(2), espEngaged(3),
absEngaged(4), aebEngaged(5), brakeWarningEngaged(6), collisionRiskWarningEngaged(7)}
(0..255)

VehicleBreakdownSubCauseCode ::= INTEGER {unavailable(0), lackOfFuel (1),
lackOfBatteryPower (2), engineProblem(3), transmissionProblem(4),
engineCoolingProblem(5), brakingSystemProblem(6), steeringProblem(7), tyrePuncture(8)}
(0..255)

PostCrashSubCauseCode ::= INTEGER {unavailable(0), accidentWithoutECallTriggered (1),
accidentWithECallManuallyTriggered (2), accidentWithECallAutomaticallyTriggered (3),
accidentWithECallTriggeredWithoutAccessToCellularNetwork(4)} (0..255)

Curvature ::= SEQUENCE {
 curvatureValue            CurvatureValue,
 curvatureConfidence       CurvatureConfidence
}

CurvatureValue ::= INTEGER {straight(0), reciprocalOf1MeterRadiusToRight(-30000),
reciprocalOf1MeterRadiusToLeft(30000), unavailable(30001)} (-30000..30001)

CurvatureConfidence ::= ENUMERATED {
    onePerMeter-0-00002 (0),
    onePerMeter-0-0001 (1),
    onePerMeter-0-0005 (2),
    onePerMeter-0-002 (3),
    onePerMeter-0-01 (4),
    onePerMeter-0-1 (5),
    outOfRange (6),
    unavailable (7)
}
```

```
CurvatureCalculationMode ::= ENUMERATED {yawRateUsed(0), yawRateNotUsed(1),
unavailable(2), ...}

Heading ::= SEQUENCE {
    headingValue           HeadingValue,
    headingConfidence      HeadingConfidence
}

HeadingValue ::= INTEGER {wgs84North(0), wgs84East(900), wgs84South(1800),
wgs84West(2700), unavailable(3601)} (0..3601)

HeadingConfidence ::= INTEGER {equalOrWithinZeroPointOneDegree (1),
equalOrWithinOneDegree (10), outOfRange(126), unavailable(127)} (1..127)

LanePosition::= INTEGER {offTheRoad(-1), hardShoulder(0),
outermostDrivingLane(1), secondLaneFromOutside(2)} (-1..14)

ClosedLanes ::= SEQUENCE {
    hardShoulderStatus     HardShoulderStatus OPTIONAL,
    drivingLaneStatus      DrivingLaneStatus,
    ...
}

HardShoulderStatus ::= ENUMERATED {availableForStopping(0), closed(1),
availableForDriving(2)}

DrivingLaneStatus ::= BIT STRING (SIZE (1..14))


PerformanceClass ::= INTEGER {unavailable(0), performanceClassA(1),
performanceClassB(2)} (0..7)

SpeedValue ::= INTEGER {standstill(0), oneCentimeterPerSec(1), unavailable(16383)}
(0..16383)

SpeedConfidence ::= INTEGER {equalOrWithinOneCentimeterPerSec(1),
equalOrWithinOneMeterPerSec(100), outOfRange(126), unavailable(127)} (1..127)

VehicleMass ::= INTEGER {hundredKg(1), unavailable(1024)} (1..1024)

Speed ::= SEQUENCE {
    speedValue      SpeedValue,
    speedConfidence SpeedConfidence
}

DriveDirection ::= ENUMERATED {forward (0), backward (1), unavailable (2)}

EmbarkationStatus ::= BOOLEAN

LongitudinalAcceleration ::= SEQUENCE {
    longitudinalAccelerationValue LongitudinalAccelerationValue,
```

```
    longitudinalAccelerationConfidence AccelerationConfidence
}

LongitudinalAccelerationValue ::= INTEGER {pointOneMeterPerSecSquaredForward(1),
pointOneMeterPerSecSquaredBackward(-1), unavailable(161)} (-160 .. 161)

AccelerationConfidence ::= INTEGER {pointOneMeterPerSecSquared(1), outOfRange(101),
unavailable(102)} (0 .. 102)

LateralAcceleration ::= SEQUENCE {
    lateralAccelerationValue LateralAccelerationValue,
    lateralAccelerationConfidence AccelerationConfidence
}

LateralAccelerationValue ::= INTEGER {pointOneMeterPerSecSquaredToRight(-1),
pointOneMeterPerSecSquaredToLeft(1), unavailable(161)} (-160 .. 161)

VerticalAcceleration ::= SEQUENCE {
    verticalAccelerationValue VerticalAccelerationValue,
    verticalAccelerationConfidence AccelerationConfidence
}

VerticalAccelerationValue ::= INTEGER {pointOneMeterPerSecSquaredUp(1),
pointOneMeterPerSecSquaredDown(-1), unavailable(161)} (-160 .. 161)

StationType ::= INTEGER {unknown(0), pedestrian(1), cyclist(2), moped(3),
motorcycle(4), passengerCar(5), bus(6),
lightTruck(7), heavyTruck(8), trailer(9), specialVehicles(10), tram(11),
roadSideUnit(15)} (0..255)

ExteriorLights ::= BIT STRING {
    lowBeamHeadlightsOn      (0),
    highBeamHeadlightsOn     (1),
    leftTurnSignalOn         (2),
    rightTurnSignalOn        (3),
    daytimeRunningLightsOn   (4),
    reverseLightOn           (5),
    fogLightOn               (6),
    parkingLightsOn          (7)
} (SIZE(8))

DangerousGoodsBasic::= ENUMERATED {
    explosives1(0),
    explosives2(1),
    explosives3(2),
    explosives4(3),
    explosives5(4),
    explosives6(5),
    flammableGases(6),
    nonFlammableGases(7),
    toxicGases(8),
    flammableLiquids(9),
```

```
    flammableSolids(10),
    substancesLiableToSpontaneousCombustion(11),
    substancesEmittingFlammableGasesUponContactWithWater(12),
    oxidizingSubstances(13),
    organicPeroxides(14),
    toxicSubstances(15),
    infectiousSubstances(16),
    radioactiveMaterial(17),
    corrosiveSubstances(18),
    miscellaneousDangerousSubstances(19)
}

DangerousGoodsExtended ::= SEQUENCE {
    dangerousGoodsType      DangerousGoodsBasic,
    unNumber                INTEGER (0..9999),
    elevatedTemperature     BOOLEAN,
    tunnelsRestricted       BOOLEAN,
    limitedQuantity         BOOLEAN,
    emergencyActionCode     IA5String (SIZE (1..24)) OPTIONAL,
    phoneNumber             IA5String (SIZE (1..24)) OPTIONAL,
    companyName             UTF8String (SIZE (1..24)) OPTIONAL
}

SpecialTransportType ::= BIT STRING {heavyLoad(0), excessWidth(1), excessLength(2),
excessHeight(3)} (SIZE(4))

LightBarSirenInUse ::= BIT STRING {
    lightBarActivated (0),
    sirenActivated (1)
} (SIZE(2))

HeightLonCarr ::= INTEGER {oneCentimeter(1), unavailable(100)} (1..100)

PosLonCarr ::= INTEGER {oneCentimeter(1), unavailable(127)} (1..127)

PosPillar ::= INTEGER {tenCentimeters(1), unavailable(30)} (1..30)

PosCentMass ::= INTEGER {tenCentimeters(1), unavailable(63)} (1..63)

RequestResponseIndication ::= ENUMERATED {request(0), response(1)}

SpeedLimit ::= INTEGER {oneKmPerHour(1)} (1..255)

StationarySince ::= ENUMERATED {lessThan1Minute(0), lessThan2Minutes(1),
lessThan15Minutes(2), equalOrGreater15Minutes(3)}

Temperature ::= INTEGER {equalOrSmallerThanMinus60Deg (-60), oneDegreeCelsius(1),
equalOrGreaterThan67Deg(67)} (-60..67)

TrafficRule ::= ENUMERATED {noPassing(0), noPassingForTrucks(1), passToRight(2),
passToLeft(3), ...
}
```

```
WheelBaseVehicle ::= INTEGER {tenCentimeters(1), unavailable(127)} (1..127)

TurningRadius ::= INTEGER {point4Meters(1), unavailable(255)} (1..255)

PosFrontAx ::= INTEGER {tenCentimeters(1), unavailable(20)} (1..20)

PositionOfOccupants ::= BIT STRING {
    row1LeftOccupied (0),
    row1RightOccupied (1),
    row1MidOccupied (2),
    row1NotDetectable (3),
    row1NotPresent (4),
    row2LeftOccupied (5),
    row2RightOccupied (6),
    row2MidOccupied (7),
    row2NotDetectable (8),
    row2NotPresent (9),
    row3LeftOccupied (10),
    row3RightOccupied (11),
    row3MidOccupied (12),
    row3NotDetectable (13),
    row3NotPresent (14),
    row4LeftOccupied (15),
    row4RightOccupied (16),
    row4MidOccupied (17),
    row4NotDetectable (18),
    row4NotPresent (19)} (SIZE(20))

PositioningSolutionType ::= ENUMERATED {noPositioningSolution(0), sGNSS(1), dGNSS(2),
sGNSSplusDR(3), dGNSSplusDR(4), dR(5), ...}

VehicleIdentification ::= SEQUENCE {
    wMInumber WMInumber OPTIONAL,
    vDS VDS OPTIONAL,
    ...
}

WMInumber ::= IA5String (SIZE(1..3))

VDS ::= IA5String (SIZE(6))

EnergyStorageType ::= BIT STRING {hydrogenStorage(0), electricEnergyStorage(1),
liquidPropaneGas(2), compressedNaturalGas(3), diesel(4), gasoline(5), ammonia(6)}
(SIZE(7))

VehicleLength ::= SEQUENCE {
    vehicleLengthValue                  VehicleLengthValue,
    vehicleLengthConfidenceIndication   VehicleLengthConfidenceIndication
}
```

```
VehicleLengthValue ::= INTEGER {tenCentimeters(1), outOfRange(1022),
unavailable(1023)}  (1..1023)

VehicleLengthConfidenceIndication ::= ENUMERATED {noTrailerPresent(0),
trailerPresentWithKnownLength(1), trailerPresentWithUnknownLength(2),
trailerPresenceIsUnknown(3), unavailable(4)}

VehicleWidth ::= INTEGER {tenCentimeters(1), outOfRange(61), unavailable(62)} (1..62)

PathHistory::= SEQUENCE (SIZE(0..40)) OF PathPoint

EmergencyPriority ::= BIT STRING {requestForRightOfWay(0),
requestForFreeCrossingAtATrafficLight(1)} (SIZE(2))

InformationQuality ::= INTEGER {unavailable(0), lowest(1), highest(7)} (0..7)

RoadType ::= ENUMERATED {
    urban-NoStructuralSeparationToOppositeLanes(0),
    urban-WithStructuralSeparationToOppositeLanes(1),
    nonUrban-NoStructuralSeparationToOppositeLanes(2),
    nonUrban-WithStructuralSeparationToOppositeLanes(3)}

SteeringWheelAngle ::= SEQUENCE {
    steeringWheelAngleValue      SteeringWheelAngleValue,
    steeringWheelAngleConfidence SteeringWheelAngleConfidence
}

SteeringWheelAngleValue ::= INTEGER {straight(0), onePointFiveDegreesToRight(-1),
onePointFiveDegreesToLeft(1), unavailable(512)} (-511..512)

SteeringWheelAngleConfidence ::= INTEGER {equalOrWithinOnePointFiveDegree (1),
outOfRange(126), unavailable(127)} (1..127)

TimestampIts ::= INTEGER {utcStartOf2004(0), oneMillisecAfterUTCStartOf2004(1)}
(0..4398046511103)

VehicleRole ::= ENUMERATED {default(0), publicTransport(1), specialTransport(2),
dangerousGoods(3), roadWork(4), rescue(5), emergency(6), safetyCar(7),
agriculture(8),commercial(9),military(10),roadOperator(11),taxi(12), reserved1(13),
reserved2(14), reserved3(15)}

YawRate::= SEQUENCE {
    yawRateValue           YawRateValue,
    yawRateConfidence      YawRateConfidence
}

YawRateValue ::= INTEGER {straight(0), degSec-000-01ToRight(-1), degSec-000-
01ToLeft(1), unavailable(32767)} (-32766..32767)

YawRateConfidence ::= ENUMERATED {
    degSec-000-01 (0),
    degSec-000-05 (1),
```

```
    degSec-000-10 (2),
    degSec-001-00 (3),
    degSec-005-00 (4),
    degSec-010-00 (5),
    degSec-100-00 (6),
    outOfRange (7),
    unavailable (8)
}

ProtectedZoneType::=  ENUMERATED { cenDsrcTolling (0), ... }

RelevanceDistance ::= ENUMERATED {lessThan50m(0), lessThan100m(1), lessThan200m(2),
lessThan500m(3), lessThan1000m(4), lessThan5km(5), lessThan10km(6), over10km(7)}

RelevanceTrafficDirection ::= ENUMERATED {allTrafficDirections(0), upstreamTraffic(1),
downstreamTraffic(2), oppositeTraffic(3)}

TransmissionInterval ::= INTEGER {oneMilliSecond(1), tenSeconds(10000)} (1..10000)

ValidityDuration ::=    INTEGER {timeOfDetection(0), oneSecondAfterDetection(1)}
(0..86400)

ActionID ::= SEQUENCE {
       originatingStationID       StationID,
       sequenceNumber             SequenceNumber
}

ItineraryPath ::= SEQUENCE SIZE(1..40) OF ReferencePosition

ProtectedCommunicationZone ::= SEQUENCE {
       protectedZoneType          ProtectedZoneType,
       expiryTime                 TimestampIts OPTIONAL,
       protectedZoneLatitude      Latitude,
       protectedZoneLongitude     Longitude,
       protectedZoneRadius        ProtectedZoneRadius OPTIONAL,
       protectedZoneID            ProtectedZoneID OPTIONAL
}

Traces ::= SEQUENCE SIZE(1..7) OF PathHistory

NumberOfOccupants ::= INTEGER {oneOccupant (1), unavailable(127)} (0 .. 127)

SequenceNumber ::=  INTEGER (0..65535)

PositionOfPillars ::= SEQUENCE (SIZE(1..3, ...)) OF PosPillar

RestrictedTypes ::= SEQUENCE (SIZE(1..3, ...)) OF StationType

EventHistory::= SEQUENCE (SIZE(1..23)) OF EventPoint

EventPoint ::= SEQUENCE {
       EventPosition       DeltaReferencePosition,
```

```
        EventDeltaTime      PathDeltaTime OPTIONAL,
        InformationQuality  InformationQuality
}

ProtectedCommunicationZonesRSU ::= SEQUENCE (SIZE(1..16)) OF
ProtectedCommunicationZone

CenDsrcTollingZone ::= SEQUENCE {
        ProtectedZoneLatitude       Latitude,
        protectedZoneLongitude      Longitude,
        cenDsrcTollingZoneID        CenDsrcTollingZoneID OPTIONAL
}

ProtectedZoneRadius ::= INTEGER {oneMeter(1)} (1..255,...)

ProtectedZoneID ::= INTEGER (0.. 134217727)

CenDsrcTollingZoneID ::= ProtectedZoneID

END
```

8.3.3. Platoon-Container

```
Platoon-Container {
itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) en (302637) pc
(16) version (1)
}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS
        VehicleLength FROM ITS-Container {
                itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) ts
(102894) cdd (2) version (1)
        };

VehicleConfiguration ::= SEQUENCE {
    vehicleLength                       VehicleLength,
    powerToMassRatio                    PowerToMassRatio,
    brakeCapacity                       LongitudinalHdAccelerationValue,
    ...
}

-- Similar as LongitudinalAcceleration from the ITS-Container but with 0.01m/s2
resolution
LongitudinalHdAcceleration ::= SEQUENCE {
    longitudinalAccelerationValue       LongitudinalHdAccelerationValue,
    longitudinalAccelerationConfidence  HdAccelerationConfidence
```

```
}

BrakeCapacityConfidence ::= ENUMERATED {
    unavailable (0)
    --TODO: Add more confidence levels
}

--hash of key?! how to generate it? who will be using it for which purpose?
PlatoonID ::= OCTET STRING (SIZE(16))

GrossCombinationVehicleWeight ::= INTEGER {tenKg(1), outOfRange(32766),
unavailable(32767)} (0..32767)

LongitudinalHdAccelerationValue ::= INTEGER {
pointZeroOneMeterPerSecSquaredForward(1), pointZeroOneMeterPerSecSquaredBackward(-1),
unavailable(1610)} (-1600 .. 1610)

HdAccelerationConfidence ::= INTEGER { pointZeroOneMeterPerSecSquared(1),
outOfRange(1022), unavailable(1023)} (0 .. 1023)

PowerToMassRatio ::= INTEGER { oneWperKg(1), outOfRange(255), unavailable(256)} (1 ..
256)

VehicleID ::= IA5String (SIZE (11..20))

PlatoonPosition ::= INTEGER {leader(1), firstFollower(2), unavailable(32)} (1..32)

END
```

8.3.4. PlatoonControl

```
PCM-PDU-Descriptions {
itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) en (302637) pcm
(14) version (1)
}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS
    ItsPduHeader, StationID, StationType, SequenceNumber, ReferencePosition,
VehicleLength, VehicleLengthValue, LateralAcceleration, YawRate, Curvature, Speed,
SpeedValue, CauseCode, Heading FROM ITS-Container {
    itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) ts (102894)
cdd (2) version (1)
}

    GrossCombinationVehicleWeight, PowerToMassRatio, LongitudinalHdAcceleration,
LongitudinalHdAccelerationValue, VehicleID, PlatoonID, PlatoonPosition
```

```
      FROM Platoon-Container

    GenerationDeltaTime FROM CAM-PDU-Descriptions
    VehicleConfiguration FROM Platoon-Container {
        itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) en
(302637) pc (16) version (1)
};

PCM ::= SEQUENCE {
    header                      ItsPduHeader,
    platoonControlContainer     PlatoonControlContainer
}

PlatoonControlContainer ::= SEQUENCE {
       stationType                      StationType,
       referencePosition                ReferencePosition,
       heading                          Heading,
       generationDeltaTime              GenerationDeltaTime,
       sequenceNumber                   SequenceNumber,
       vehicleID                        VehicleID,
       vehicleInFrontID                 VehicleID OPTIONAL,
       longitudinalControlContainer     LongitudinalControlContainer,
       lateralControlContainer          LateralControlContainer OPTIONAL,
       statusSharingContainer           PlatoonStatusSharingContainer,
       vehicleConfiguration             VehicleConfiguration OPTIONAL,
       tacticalPlanning                 TacticalPlanningContainer OPTIONAL,
       splitStatus                      SplitStatus OPTIONAL,
       ...
}

PlatoonStatusSharingContainer ::= SEQUENCE {
       -- numberOfTrucks is aggregated/forwarded rear-to-front
       numberOfTrucks              INTEGER(2..31),
       -- platoonSpeed is aggregated/forwarded front-to-rear
       platoonSpeed                Speed OPTIONAL,
       -- platoonPosition is the ego vehicle’s position in the platoon
       platoonPosition             PlatoonPosition,
       platoonID                   PlatoonID,
       reasonForSpeedOrGapAdjustment    ReasonForSpeedOrGapAdjustment OPTIONAL,
       ...
}

TacticalPlanningContainer ::= SEQUENCE {
    -- The tactical planning container is forwarded/aggregated rear-to-front
       cohesionContainer   CohesionContainer OPTIONAL,
       ...
}

CohesionContainer ::= SEQUENCE {
       requestedMaxSpeed                        SpeedValue,
       requestedMaxLongitudinalAcceleration     LongitudinalHdAccelerationValue
OPTIONAL,
```

```
        ...
}

LongitudinalControlContainer ::= SEQUENCE {
        currentLongitudinalAcceleration         LongitudinalHdAcceleration,
        predictedLongitudinalAcceleration       LongitudinalHdAccelerationValue,
        longitudinalSpeed                       Speed,
        roadInclination                         RoadInclination,
        grossCombinationVehicleWeight           GrossCombinationVehicleWeight,
        referenceSpeed                          Speed,
        intruderAhead                           VehicleAhead OPTIONAL,
        vehicleAhead                            VehicleAhead OPTIONAL,
        ...
}

LateralControlContainer ::= SEQUENCE {
    lateralAcceleration                 LateralAcceleration,
        yawRate                         YawRate,
        curvature                       Curvature,
        distanceToLeftLaneMarking       LaneMarkingDistance,
        distanceToRightLaneMarking      LaneMarkingDistance,
        ...
}

SplitStatus ::= SEQUENCE  {
        frontSplit          FrontSplit,
        requestBackSplit    BOOLEAN
}

FrontSplit ::= ENUMERATED {
        unpreparedForFrontSplit(0),
        preparingForFrontSplit(1),
        preparedForFrontSplit(2)
}

ReasonForSpeedOrGapAdjustment ::= ENUMERATED {
  unknown(0),
  safety(1),
  efficiency(2),
  trafficAhead(3),
  intruder(4),
  emergency(5),
  leave(6),
  cohesion(7),
  ...
}

DistanceToVehicleAhead ::= INTEGER {oneCm(1), unavailable(16383)} (0..16383)

LaneMarkingDistance ::= INTEGER {oneCm(1), unavailable(511)} (0..511)
```

```
RoadInclination ::= INTEGER {pointOnePercentUp(1), pointOnePercentDown(-1),
unavailable(128)} (-127 .. 128)

VehicleAhead ::= SEQUENCE {
    Distance DistanceToVehicleAhead,
    speed    SpeedValue
}

END
```

8.3.5. PlatoonMgmt

```
PMM-PDU-Descriptions {
 itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) en (302637) pmm
(13) version (1)
}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS
       ItsPduHeader, StationID, StationType, ReferencePosition, VehicleLength, Heading
       FROM ITS-Container {
              itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) ts
(102894) cdd (2) version (1)
       }

       LongitudinalHdAccelerationValue, PowerToMassRatio, VehicleID, PlatoonID,
PlatoonPosition, VehicleConfiguration
       FROM Platoon-Container

       GenerationDeltaTime
       FROM CAM-PDU-Descriptions

       SymmetricEncryptionKey, PublicEncryptionKey
       FROM IEEE1609dot2BaseTypes {
              iso(1) identified-organization(3) ieee(111) standards-association-
numbered-series-standards(2) wave-stds(1609) dot2(2) base(1) base-types(2) major-
version-2 (2)
       };

PMM ::=  SEQUENCE {
       Header              ItsPduHeader,
       stationType         StationType,
       referencePosition   ReferencePosition,
       heading             Heading,
       generationDeltaTime GenerationDeltaTime,
       message CHOICE {
              joinRequest  JoinRequest,
```

```
            joinResponse  JoinResponse,
            KeyUpdate     KeyUpdate,
            ...
      }
}

JoinRequest ::= SEQUENCE {
      receiver                    StationID,
      numberOfTrucks              INTEGER(1..31),
      responseKey                 PublicEncryptionKey,
      vehicleConfiguration        VehicleConfiguration,
      platooningLevel             PlatooningLevel,
      ...
}

-- KeyUpdate has been renamed to PlatoonUpdate
KeyUpdate ::= SEQUENCE {
      groupKey             SymmetricEncryptionKey,
      participantKey       SymmetricEncryptionKey,
      platoonId            PlatoonID,
      updatedPosition      PlatoonPosition
}

JoinResponse ::= SEQUENCE {
      respondingTo         StationID,
      joinResponseStatus   CHOICE {
            notAllowedToJoin    NULL,
            allowedToJoin       JoinResponseInfo
      }
}
JoinResponseInfo ::= SEQUENCE {
      groupKey                    SymmetricEncryptionKey,
      participantKey              SymmetricEncryptionKey,
      frequencyChannel            FrequencyChannel DEFAULT cch,
      platoonId                   PlatoonID,
      maxNrOfVehiclesInPlatoon    INTEGER(2..31),
      joiningAtPosition           PlatoonPosition
}


PlatooningLevel ::= ENUMERATED {
      platooning-level-A(0),                         -- does the platooning level
indicate if functions like "cohesion support" are supported?
      platooning-level-B(1),
      platooning-level-C(2),
      ...
}
FrequencyChannel ::= ENUMERATED { cch(0), sch1(1), sch2(2), sch3(3), sch4(4), sch5(5),
sch6(6) }

END
```