Department of Creative Informatics
Graduate School of Information Science and Technology
THE UNIVERSITY OF TOKYO

Master's Thesis

Reliable Cooperative Perception for Connected Roadside Infrastructure

高信頼性な協調認知を可能にする協調インフラの研究

Yu Asabe
浅部 佑

Supervisor: Professor Hiroshi Esaki

January 2023

Abstract

For cooperative intelligent transport systems (C-ITS), vehicle-to-everything (V2X) communication is utilized to allow autonomous vehicles (AVs) to share critical information with each other. Collective perception enables connected autonomous vehicles (CAVs) to overcome the limitations of standalone AVs by sharing sensory information with nearby road users. The goal of this thesis is to analyze the requirements for reliable collective perception amongst multiple ITS stations and to propose methods of improving the reliability of its utilization.

We propose three methods of increasing the reliability of collective perception: (i) dual-channel hybrid delivery of sensory information, (ii) real-time packet delivery rate (PDR) monitoring, and (iii) the adaptive dual-channel delivery of sensory information using the combination of the above two methods.

In order to realize the proposed methods, we introduce AutowareV2X, an implementation of a V2X communication module that is integrated into the autonomous driving (AD) software, Autoware. AutowareV2X provides external connectivity to the entire AD stack, enabling the end-to-end (E2E) experimentation and evaluation of connected autonomous vehicles (CAV). The Collective Perception Service (CPS), which is standardized by the European Telecommunications Standard Institute (ETSI), is implemented, allowing the transmission of Collective Perception Messages (CPMs).

Functional verification in simulation-based experiments, as well as indoor experiments and outdoor field experiments, were conducted to evaluate the performance of AutowareV2X and the effectiveness of our proposed methods. Field experiments have indicated that the E2E latency of perception information is around 30 ms, and shared object data can be used by the AD software to conduct collision avoidance maneuvers. The effectiveness of the proposed methods was also confirmed, with the dual-channel delivery of CPMs enabling the CAV to dynamically select the best CPM from CPMs received from different links, depending on the freshness of their information. This enabled the reliable transmission of CPMs even where there was significant packet loss on one of the transmitting channels.

概要

自律型自動運転の研究開発と社会実装が着々と進む中、その技術的な課題や限界点も指摘され始めている．そこで、最先端の通信やネットワーク技術を活かして様々な交通システムが協調的に認知、判断、実行を担える協調型自動運転の分野が注目されている．特に、多くのコネクテッドな交通参加者が自らのセンサーで認識した物標情報を共有することで、全体の環境の認識率の向上を図る「協調認知」の活用は大きく期待されている．

本研究では、協調型自動運転車や協調インフラの高信頼性な協調認知を可能にする研究開発に着目し、新しい通信メカニズムを用いて従来の協調認知より信頼性の高い協調認知システムの考案を目指す．具体的な手法として 3 つの方法を主に提案する．(i) 二重化された通信チャンネルを用いてセンサ情報を送受信するデュアルチャネルハイブリッド配信、(ii) 送信側と受信側の両方にてパケット到達率（PDR）をリアルタイムに計測できる方式、そして (iii) 上記の 2 つの方法を組み合わせた、通信チャンネルの通信状況に応じてセンサ情報の送受信の冗長化を調節できるアダプティブデュアルチャンネル配信.

協調認知の信頼性向上につながる上記の提案手法らをを実現するために、本研究では自動運転ソフトウェア「Autoware」に統合する形で活用できる V2X 通信モジュール「AutowareV2X」の要件定義、設計、及び実装を行った．AutowareV2X は、自動運転ソフトウェアスタック全体に外部接続性を提供し、コネクテッドカーのエンドツーエンドの実験と評価を可能にする．欧州の標準化機構 ETSI によって標準化が進められている「Collective Perception Service（CPS）」を AutowareV2X のアプリケーションとして実装し、センサー情報の共有に活用できる「Collective Perception Messages（CPMs）」の送受信を可能とした.

AutowareV2X の性能、及び提案手法の有効性を評価するために、シミュレーションベースの実験、屋内実験、屋外フィールド実験を行った．フィールド実験では、協調認知のエンドツーエンドのレイテンシは約 30ms であり、共有された物標情報は自動運転車が衝突回避動作を実行するために使用できることが示された．継続的な PDR モニタリング及び CPM パケットのデュアルチャンネルハイブリッド配信が、協調認知の信頼性の向上に寄与していることも示された．冗長化された通信チャンネルのうち、1 つのチャンネルでパケットロスが大きくなってしまっている場合でも、CPM を継続的に送受信し、物標情報を低遅延で共有することができた.

Contents

List of Figures

List of Tables

List of Acronyms

AD Autonomous Driving

ASN.1 Abstract Syntax Notation One

AV Autonomous Vehicles

CAM Cooperative Awareness Message

CAV Connected Autonomous Vehicles

C-ITS Cooperative Intelligent Transport Systems

CPAM CPM Assistive Messages

CPM Collective Perception Messages

CPS Collective Perception Service

DENM Decentralized Environmental Notification Message

ETSI European Telecommunications Standard Institute

ITS Intelligent Transportation Systems

ITS-S ITS Station

ITU International Telecommunication Union

LDM Local Dynamic Map

NHTSA National Highway Traffic Safety Administration

RAT Radio Access Technology

RSU Road-Side Units

SAE Society of Automotive Engineers

VRUs Vulnerable Road Users

V2I Vehicle-to-Infrastructure

V2N Vehicle-to-Network

V2P Vehicle-to-Pedestrian

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

Chapter 1

Introduction

1.1 Background

The development and deployment of Intelligent Transportation Systems (ITS) pose as one of the most critical challenges for our modern society. With rapid advances in cyber-physical system technologies, the use of Autonomous Vehicles (AV) and Connected Autonomous Vehicles (CAV) are becoming increasingly feasible. The arrival of these new mobility technologies presents us with new opportunities to drastically improve road safety, traffic throughput, and energy efficiency.

The National Highway Traffic Safety Administration (NHTSA) reports that more than 42,000 people died in motor vehicle-related accidents in the United States (US) in the year 2021 alone [1]. Behind each of these numbers is a life tragically lost and a grieving family left behind. People living in the US also waste an extra 8.8 billion hours of time and an extra 3.3 billion gallons of fuel every year due to traffic congestion, and resulting delays [2]. The NHTSA anticipates that CAVs can reduce traffic congestion and free up nearly 50 minutes for the average commuter every day [3].

Although it is very unlikely that cities in the US will see fully autonomous environments by 2045, the penetration of AVs and CAVs will gradually progress [4]. The public's adaptability and comfort levels with these emerging vehicles, as well as the government's safety regulations and financial considerations, will greatly affect the market penetration rate. The consensus is that the global demand for transportation and mobility services is substantially increasing, along with the energy consumption levels and financial costs to support such infrastructure.

Future mobility services must deal with the limitations of available space, traffic volume, and energy resources while simultaneously fulfilling the increasing demand for a safe and efficient method of transport. To this end, next-generation transportation systems require a higher level of i) safety and dependability ii) spatial and temporal efficiency, and iii) realistic affordability. This is especially true for densely populated urban areas where the mobility demand, as well as the imposed requirements, are set high.

1.1.1 Standalone Autonomous Vehicles

Autonomous Driving (AD) technology is one of the key players in realizing all of these aspects and facilitating large-scale deployment. Empowering the AV's individual capabilities and increasing its level of autonomy can provide more optimized operation at a local level and can realize increased autonomy in wider use cases. The core functions of AVs are mainly categorized into four parts: perception, localization, planning, and control. Sensors such as the camera (image sensor), LiDAR (Light Detection and Ranging), ultrasonic Radar, and millimeter-wave Radar generate sensory input that can be processed into perception information. The sensory data can also be used in conjunction with high-definition maps of the environment to enable accurate localization of the ego-vehicle. The various sensors mounted onto AVs are shown in Fig. 1.2, while the AV architecture is shown in Fig. 1.3. After the AV is able to perceive its environment, it makes planning decisions based on the obstacles in the path and the designated goal. Finally, the AV is responsible for actuating the vehicle by outputting control commands to the steering or throttle.

The Society of Automotive Engineers (SAE) [5] provides a guideline for vehicular automation, where six levels of automation for autonomously driven vehicles are defined as shown in Fig. 1.1. The levels are defined based on the sophistication of the technology used for automation.

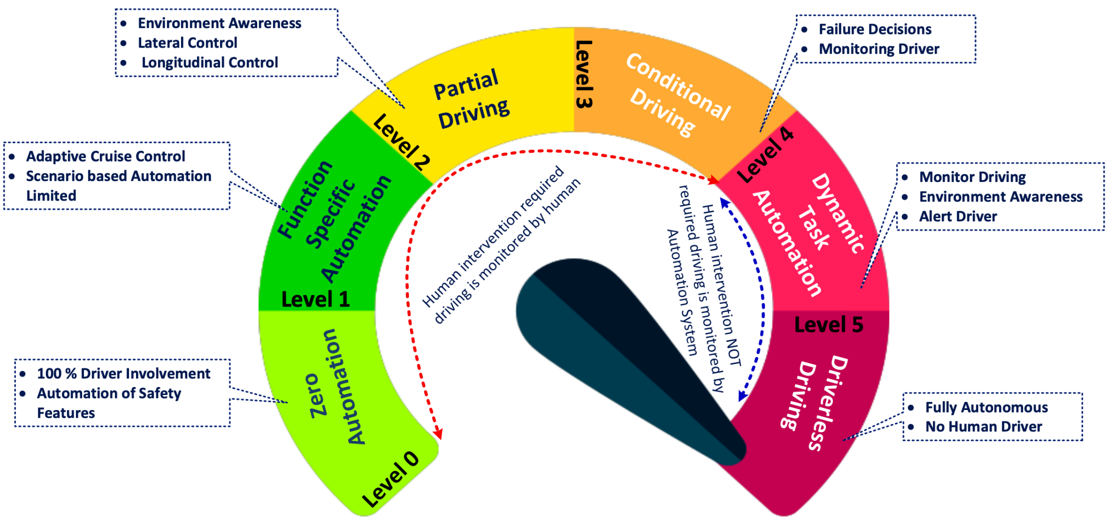

Fig. 1.1: Levels of Automation in Autonomous Vehicles [6]

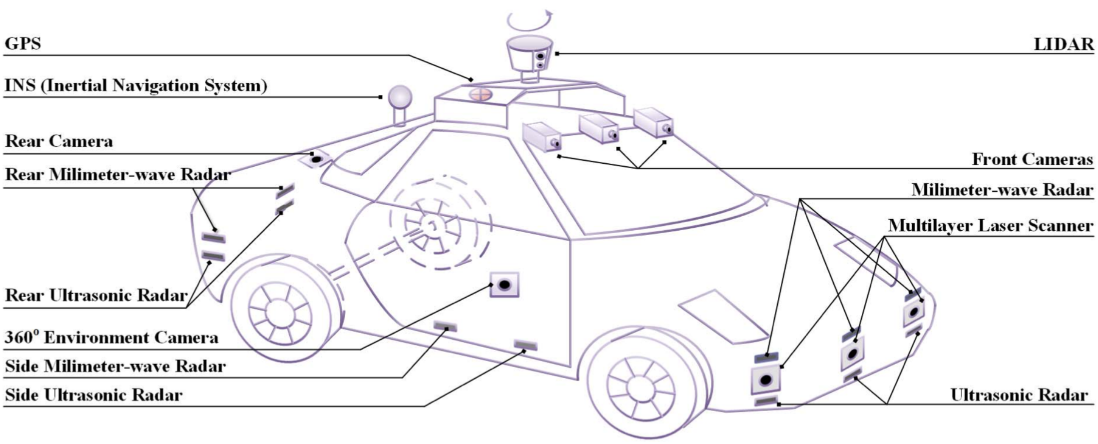

Fig. 1.2: Various Sensors equipped on Autonomous Vehicles [7]

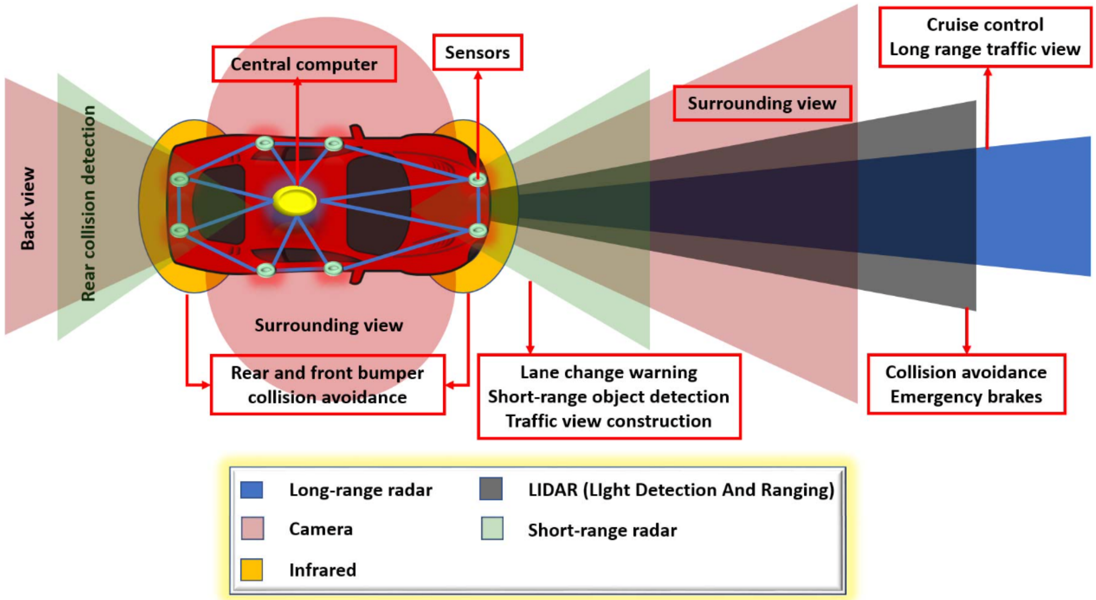

Fig. 1.3: Autonomous Vehicle Architecture [8]

1.1.2 Limitations of Standalone AVs

Stand-alone AVs only rely on their local onboard sensors to perceive their environment and are entrusted to make safety-critical decisions solely based on the information that they have. The limited perception range of these AVs poses not only a safety issue but also a problem of not being able to make optimized cooperative decisions. The quality of the perception information obtained by individual perception systems on the AVs can be decreased by the limited field of view, missing modalities, sparse sensor data, unfavorable environmental features, and other various factors. The inability to detect objects due to object occlusion is shown in Fig. 1.4, where the green ego-vehicle cannot detect the blue car due to the occlusion caused by the yellow truck. When using LiDAR point clouds as a form of sensor input, the data can become more sparse when the distance between the ego-vehicle and the objects increases. This long-range sensing issue is depicted in Fig. 1.5.

CAVs can overcome these perception issues by using V2X communication to have vehicles communicate with other nearby vehicles and infrastructure. This allows multiple ITS-Ss to be cooperatively engaged, enabling collaborative enhanced environmental awareness and multi-node decision-making. Fig. 1.6 describes the difference between standalone AVs and CAVs. This additional layer of previously exclusive information allows CAVs to achieve a level of safety and efficiency that is impossible by stand-alone AVs or human drivers.

1.1.3 Connected Autonomous Vehicles and Cooperative-ITS

Another way to approach the problem is to have road users collaborate with each other and allow them to mutually benefit from their cooperative operation. A central component to this approach would be the communication between road users, and the network effects it imposes on cooperative environmental awareness. CAV (a.k.a connected and automated

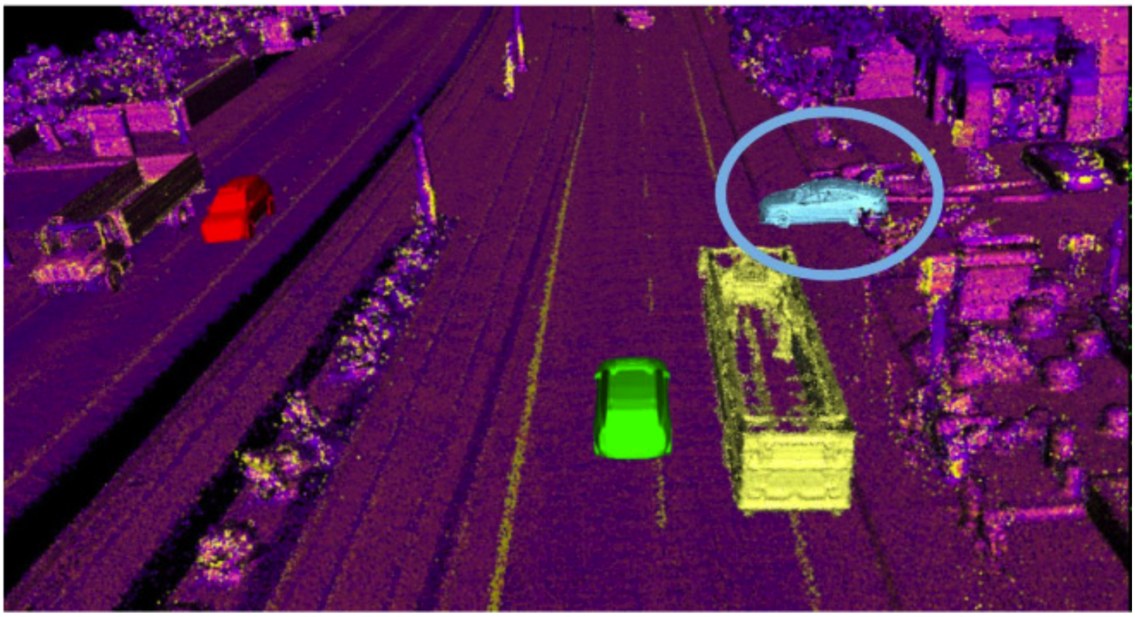

Fig. 1.4: Occlusion Issue [9]

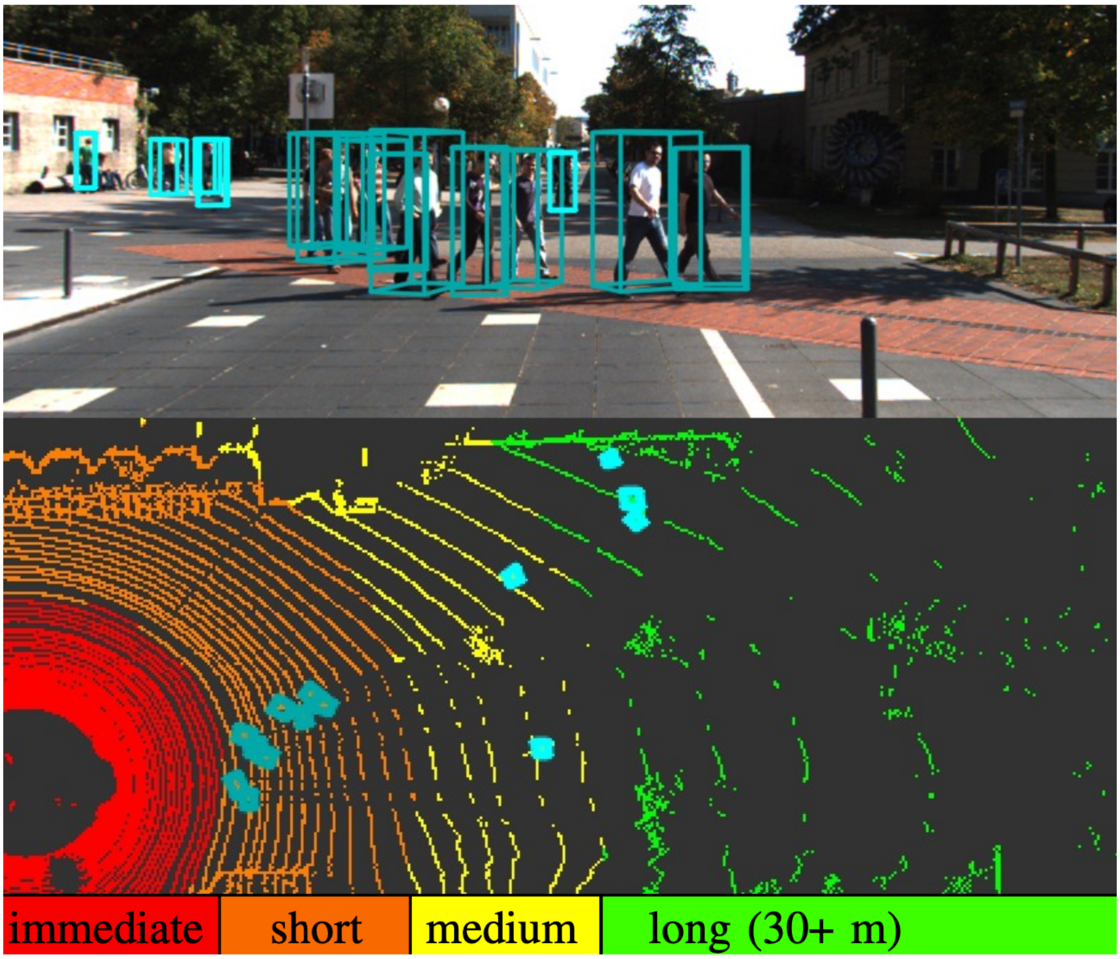

Fig. 1.5: Long Range Issue [10]

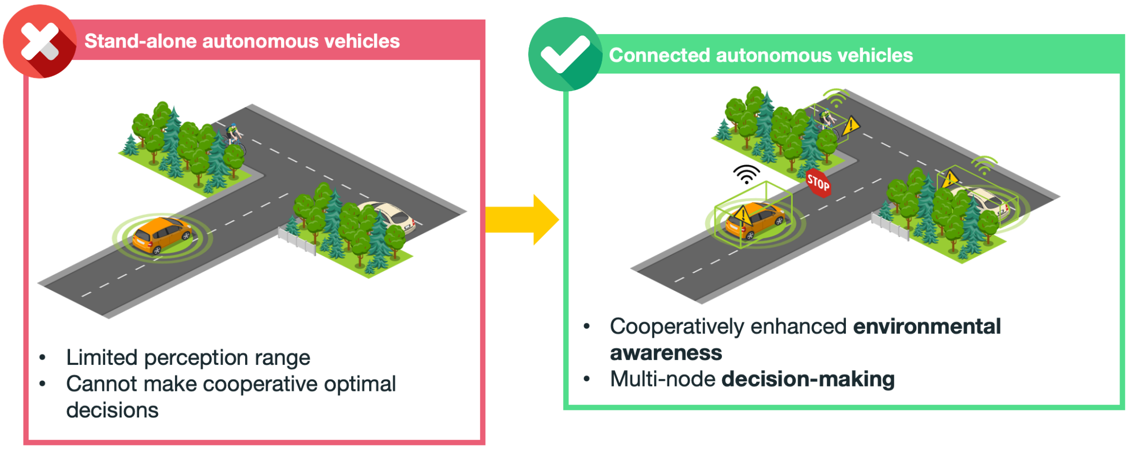

Fig. 1.6: Standalone AVs and CAVs

vehicles and driver-less cars) is one of the main participants in this road network that has great potential for reducing traffic accidents and improving the efficiency of transportation systems. When CAVs can communicate with each other and other road users through a set of standardized protocols, a variety of new ITS applications can be implemented, further benefiting all road users. The NHTSA predicts that effectively applying Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications could potentially reduce up to 80% of traffic accidents [11]. Vehicle-to-Pedestrian (V2P) and Vehicle-to-Network (V2N) communication schemes are also of considerable value, where they are collectively known as Vehicle-to-Everything (V2X) communication. The general idea of how the concepts of AVs, CAVs and the types of communications fit into one societal solution is shown in Fig. 1.7.

1.1.4 Connected Roadside Infrastructure for Smarter Mobility

In addition to connected vehicles and dynamic road users, roadside infrastructure can be deployed in public areas to act as key components in this collaboration scheme. Cooperative operation of vehicles and roadside infrastructure can complement the limitations of the vehicle-centric approach, allowing the contributions to the previously mentioned requirements in the following ways: i) Roadside infrastructure provides an extra layer of safety to standalone vehicles by providing them with safety-relevant information from other participants in the environment. The more participants there are in the vicinity, information from multiple sources can be layered on top of each other, creating a mutually beneficial dynamic environmental map. ii) Since the infrastructure will be a common resource shared by different road users, it has the potential to become a well-informed authority figure that can actively engage in the decision-making of individual vehicles. This can allow them to become a sort of traffic coordinator, overriding certain decisions made by vehicles and enhancing the global traffic flow as a result. iii) While the high initial cost of setting up and deploying a roadside infrastructure can be considered a barrier, the long-term financial benefits they can provide by publically serving a variety of road users is very promising. The centralized perception capabilities, computation resources and network resources can be shared among different road users, removing the need for individual vehicles and other road users to be equipped with the same level of expensive sensors and hardware. In order to overcome the limitations set by standalone systems, collaboration between road users and facilitating advanced cooperative schemes

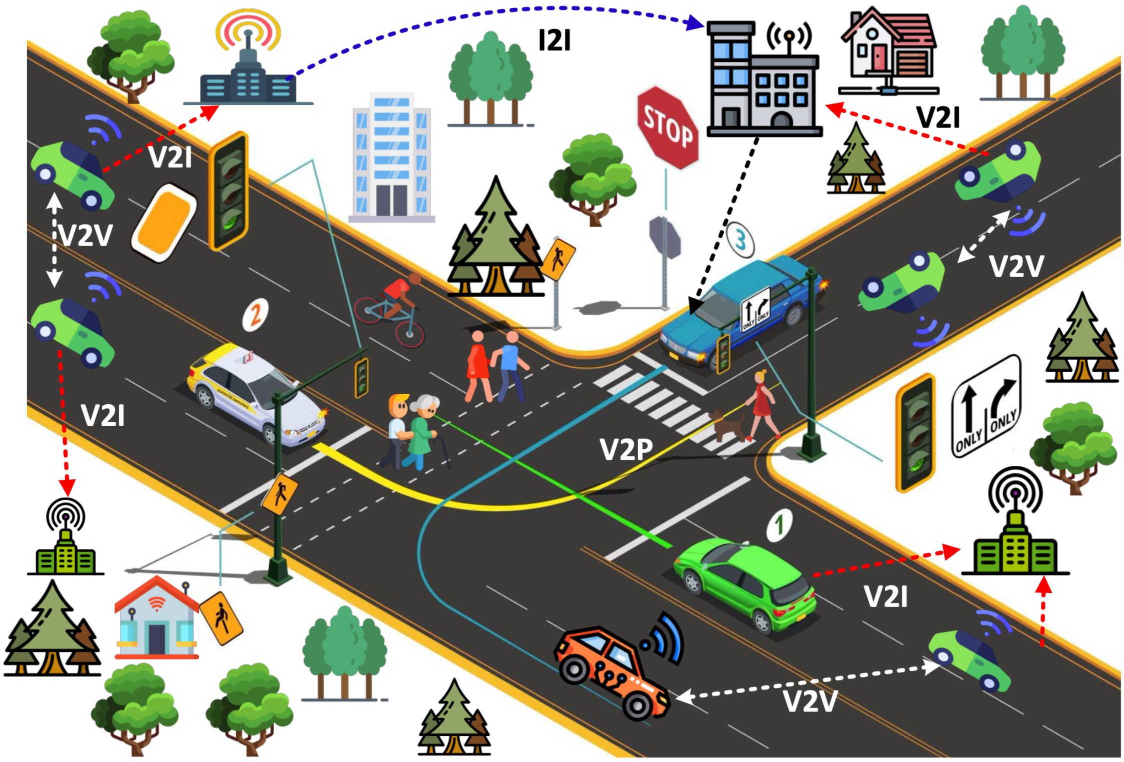

Fig. 1.7: Autonomous Vehicles, Connected Autonomous Vehicles, and its supporting communication schemes [6]

based on efficient communication technology is an enticing way forward. Entrusting the infrastructure with more advanced capabilities can also push the trend towards reducing the requirements set on vehicle autonomy, and can allow publicly shared infrastructure to take over more complex tasks such as cooperative trajectory planning, intersection management, and priority control. This requires the sensing and perception capabilities of the infrastructure to be set at a high standard since the quality and accuracy of its performance can directly influence the reliability of the cooperative decisions it makes.

1.1.5 V2X Communication Protocol Standards by ETSI

The European Telecommunications Standard Institute (ETSI) is standardizing V2X communication protocols based on the IEEE 802.11p standard called ITS-G5. In ITS-G5, nodes that are equipped with communications hardware and conduct communications-related behavior, such as connected vehicles and Road-Side Units (RSU), are called the ITS Station (ITS-S). The ITS-Station Architecture as defined by ETSI is shown in Fig. 1.8. The ITS-Station Architecture is composed of the Application layer, Facilities layer, Network and Transport layer, Access layer, Management layer, and Security layer. The architecture follows the principles of the OSI model for layered communications and extends the concept of the OSI model to include ITS-specific applications.

Examples of Facilities layer protocols standardized by ETSI are the Cooperative Awareness Message (CAM) [12], Decentralized Environmental Notification Message (DENM) [13], Local Dynamic Map (LDM) [14], and the Collective Perception Messages (CPM) [15].

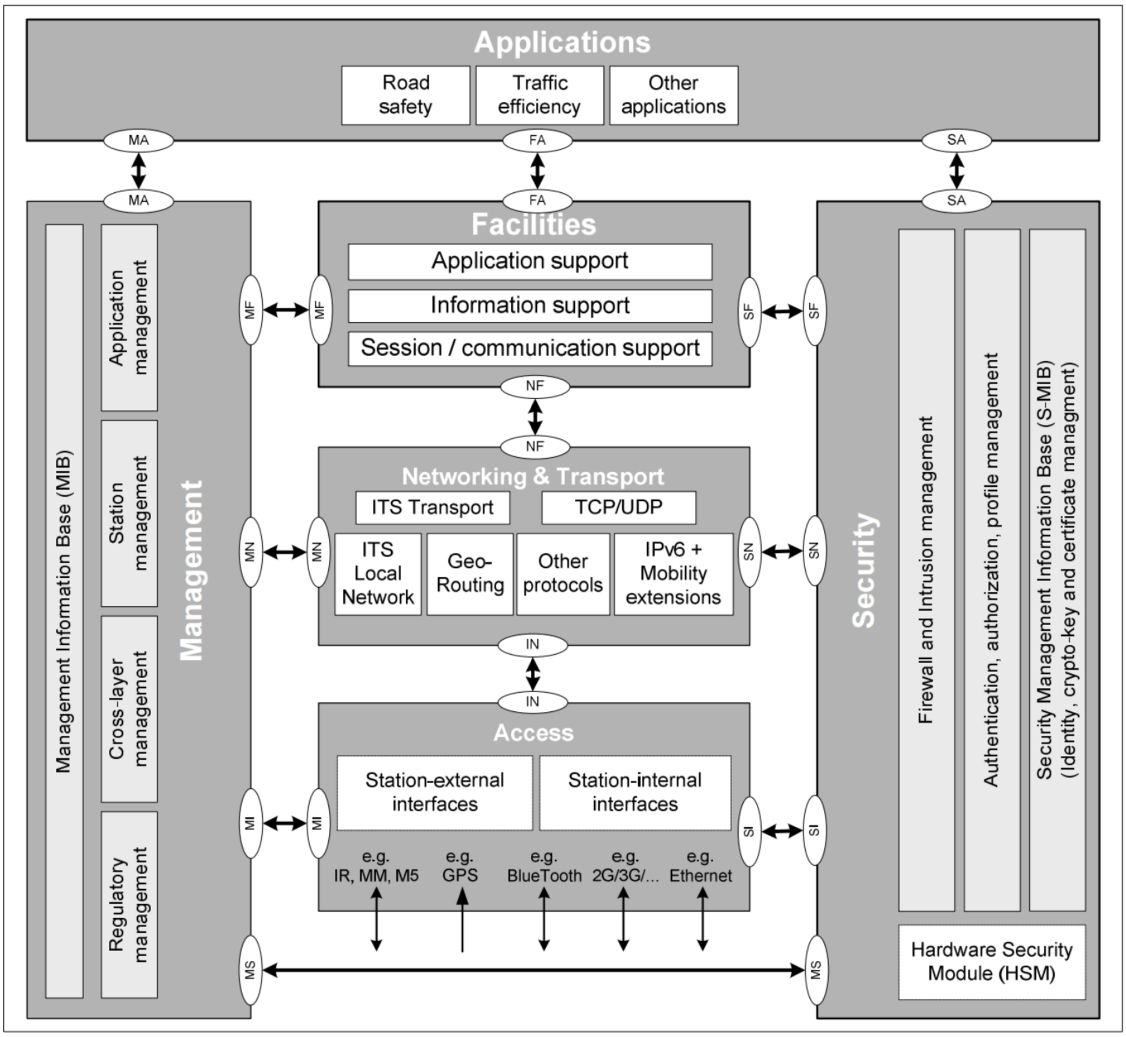

Fig. 1.8: ITS Station Architecture [16]

1.2 Motivation

For Cooperative Intelligent Transport Systems (C-ITS), V2X communication is utilized to allow AVs to share critical information with each other. Collective perception enables CAVs to overcome the limitations of standalone AVs by sharing sensory information with nearby road users. The advantages of collective perception are quite clear. However, in order to facilitate the deployment of collective perception applications and go beyond the phase of PoCs, the reliability of the information shared over the network must be sufficient. Clear metrics that describe the reliability of CPS-shared information must also be set. The goal of this thesis is to analyze the requirements for reliable collective perception amongst multiple ITS stations and to propose methods of improving the reliability of its utilization.

1.3 Organization

The rest of the dissertation is organized as follows. Chapter 2 presents related works in the field of vehicular networking and collective perception. Chapter 4 describes our contribution to designing, implementing, and evaluating a reliable V2X communication router and its collective perception application. Chapter 5 proposes novel methods of improving the reliability of collectively perceived and shared information. Chapter 6 explains the experiments that were conducted and presents findings. Chapter 7 summarizes and concludes the thesis with final thoughts and possibilities for future work.

Chapter 2

Related Works

This chapter presents the background and related work on the three topics as follows: (i) Research and Development Platforms and Tools for CAVs, (ii) Evolving vehicular network architectures, and (iii) Collective Perception. Each section of this chapter reviews relevant systems and techniques and discusses related prior work.

2.1 Research and Development Platforms and Tools for CAVs

Research and development platforms for CAVs have been widely studied, with various implementations focusing mainly on the realization of V2X communications and their application to specific use cases. Open-source platforms such as OpenC2X [17] and Vanetza [18] provide software packages that are based on the ITS Station Architecture and comply with the European V2X communication architecture standard named ITS-G5 [16]. Other proprietary V2X communication stacks exist in the market; however, they suffer from one or more of the following issues: (i) Blackbox behavior due to the lack of open-source code availability - despite the progress of open standardization by organizations such as ETSI, the software is difficult to test when the implementation details are not made clear, (ii) Lack of customizability due to slow feature updates of APIs provided by the software, (iii) Inconvenient sharing of modified code or extensions that are especially important to the research community, and (iv) the relatively expensive product fees for commercial software stacks. Table 2.1 provides a list of V2X software stacks that are both open-source and proprietary. OpenC2X realized the ETSI ITS-G5 stack and supported the CAM and DENM messages. However, its focus was limited to the design and implementation of V2X applications, leaving the integration with autonomous driving software stacks up to the works of future developers. The Cooperative Automation Research Mobility Application (CARMA) platform [19] is another open-source software stack that enables connective features on top of Autoware-based autonomous vehicles. It is based on US standards, with recent PoC for cooperative perception [20] [21]. The X-CAR experimental vehicle platform [22] is powered by CARMA but implements its capabilities in more affordable, high-quality hardware. Although this list is not exhaustive, it does give light on the fact that Vanetza is so far one of the leading open-source implementations of a working V2X communication stack. More details on Vanetza are described in Section 4.1.2.

Further works focus on specific applications or use cases of V2X-enabled scenarios and have implemented working prototypes. AutoC2X [25] extended OpenC2X by integrating it with the AD stack named Autoware [26] and enabled CAM [12]-based cooperative perception [27]. Objects that were detected by the AD stack were able to be transmitted by CAM messages to other connected road users. These projects, however, use an older

Table 2.1: Summary of V2X communication software stacks

Name	License	Description
OpenC2X	Open-source	Focused more on facilities layer applications and not the network or transport protocols. Not updated and no support community. [23]
GeoNetworking (alexvoronov)	Open-source	Basic implementation of the GeoNetworking stack No facilities layer applications Missing packet forwarding features [24]
Vanetza	Open-source	C++ library covering GeoNetworking, BTP, DCC, Security, and several Facilities. Includes demo applications for easy development. Active Development on Github [18]
CARMA	Open-source	Connectivity features on top of Autoware US-based standards Entire platform for connected solutions [19]
Commsignia	Proprietary	Full stack V2X solution running on Linux and automotive real-time OS http://www.commsignia.com/software/
ezCar2x	Proprietary	Collection of C++ libraries for prototyping and simulations https://www.esk.fraunhofer.de/en/research/projects/ezCar2X.html
Kapsch TrafficCom	Proprietary	Included in Kapsch systems and seems to run on Linux https://www.kapsch.net/ktc/downloads
Marben V2X	Proprietary	Full stack with C++ code available. OS and hardware seems to be portable http://marben- products.com/v2x/v2x-software-solution.html
Cohda Wireless SDK	Proprietary	Requires a Cohda chipset and Cohda's extensions to the Linux Kernel http://cohdawireless.com/Portals/0/MKx_SDK_10122015.pdf
NORDSYS waveBEE	Proprietary	Runs on Linux intended more for application development http://nordsys.de/en/car2x-produkte-2/280- wavebeesoftware.html

version of Autoware and do not support a wide variety of hardware or vehicles. Moreover, implementations and evaluations of the newest CPS standard were not yet done on actual hardware.

Other studies focus on only the integration of V2X communication into a specific module in the AD stack, such as the perception [28] [29] or planning [30] pipeline. [31] proposes a V2X communication-aided autonomous driving system for vehicles and presents applications for perception, planning, and control. Although this is sufficient for evaluating the effects of V2X communication on a single module or a limited scope of the AD stack, a more versatile design of AD stack integration is needed to enable the experimentation and evaluation of the effects of V2X communication on the E2E AD software as a whole.

Although each implementation shows promising results in its specific use case and deployment scenario, we are yet to see a comprehensive V2X communications router that is fully integrated with an AV software stack and capable of emitting different protocols through multi-radio access technology (RAT) interfaces.

2.2 Evolving Vehicular Network Architectures

As the demand for low-latency and reliable vehicular communication schemes increases, more focus is put on new vehicular communication standards and network architectures. For ITS-Ss to be able to communicate and share information, a common wireless communication protocol is necessary. Dedicated Short Range Communications (DSRC) [32] is a type of short-range to medium-range wireless technology specifically designed and developed for automotive platforms. DSRC utilizes the 5.9 GHz radio frequency band in most regions, with the US allocating 75 MHz of spectrum in the 5.9 GHz band for ITS use and ETSI allocating 30 MHz of spectrum in the 5.9 GHz band for ITS. DSRC consists of the physical layer, data link layer, middle layers, and message sublayers. IEEE has

Table 2.2: Comparison of standalone local vs. cooperative perception

	Local Perception (with Onboard Sensors)	Cooperative Perception (with V2X communication)
Strength	- Precise object localization within short range - Fast object detection rate - Not affected by network conditions	- Unlimited Field-of-view - Cooperatively enhanced information - Improved reliability through multi-node redundancy - Cost reduction - Detection of unconnected users
Weakness	- Limited field-of-view - Expensive sensors	- Limited accuracy (spatial and temporal) - Dependent on other CAVs or RSUs - Security issue - High throughput required - Sensitive to latency

standardized the 802.11p [33] as well.

A study [34] highlights the limitations of various existing V2X communication technology and discusses the potential for the interworking between DSRC and cellular network technologies. Ref. [35] proposes a heterogeneous LTE/DSRC approach where vehicular nodes can select its RAT based on the service requirements. Simulation-based evaluations showed that the LTE/DSRC selective algorithm outperforms purely DSRC-based communication. Another group [36] proposes a hybrid vehicular network architecture that combines DSRC and Cellular V2X and addresses radio resource management (RRM) strategies and RAT selection algorithms based on the measurements of QoS parameters and network conditions.

A common element shared amongst the different vehicular network schemes is the goal of more reliable communication through various heterogeneous technologies to provide stable high-quality information to upper-layer applications of CAVs and other ITS-S applications.

2.3 Collective Perception

One of the crucial modules of the AD stack is perception, which focuses on the task of accurately perceiving the surrounding environment and extracting information that is necessary for navigation. The whole perception pipeline includes tasks such as object detection, tracking, semantic segmentation, and so on. The perception task of a standalone AD stack is mentioned in more detail in Section 4.1.3. Although many state-of-the-art AD solutions use only their onboard sensors to provide local perception, they are faced with various limitations, many of which cannot be trivially overcome with mere improvements of individual sensors or algorithms. Collective perception (a.k.a. Collaborative or cooperative perception) enables AVs to overcome these certain limitations by sharing information with nearby vehicles and infrastructure through V2X communications. A comparison of both standalone and cooperative systems in the context of perception capabilities is described in Table 2.2.

Achieving real-time and robust collective perception in a reliable manner is a difficult task and requires addressing a multitude of challenges caused by perception accuracies, communication capacities, environmental noise, and security. There are many works that have studied the strategy of the collective perception process, including what kind of data to share, when to collaborate them, how the information should be shared, and how

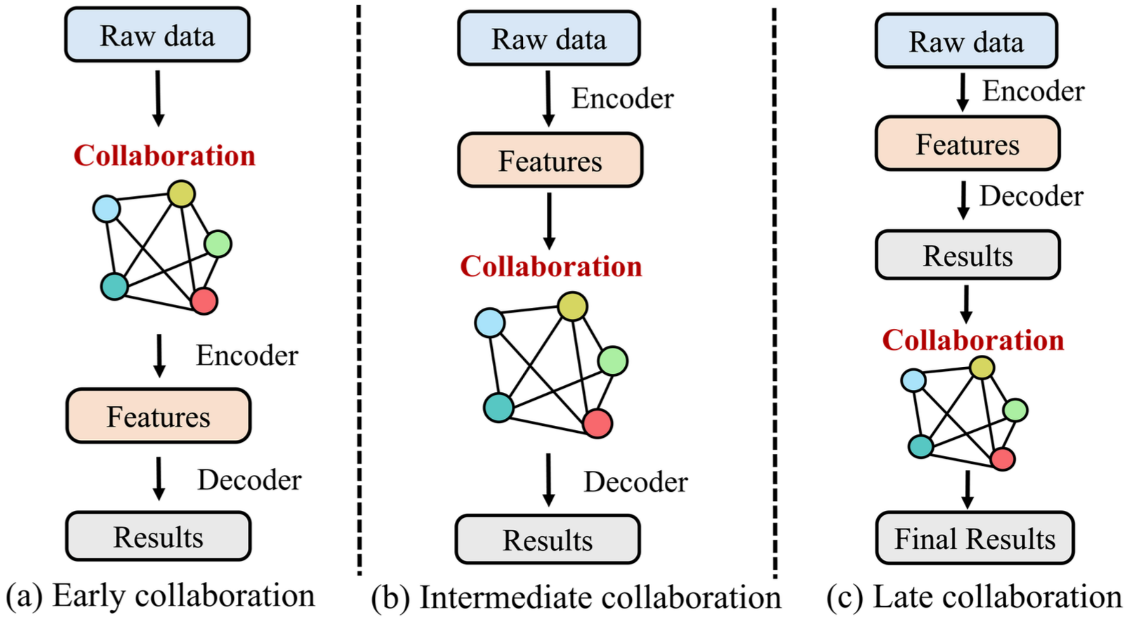

Fig. 2.1: Three Collaboration Modes of Collective Perception [37]

the shared information should be received and processed. Research in [37] defines three collaboration modes as shown in Fig. 2.1, depending on what kind of data that is being shared and when the information is being sent. Early collaboration focuses on sharing raw sensory data such as LiDAR point clouds or raw camera footage. While the abundance of information can provide a more holistic perspective for all the nodes receiving the data, the sharing of raw sensor data can require a significantly larger communication bandwidth and can easily congest the communication channel. Although emerging communication technologies can open new possibilities that utilize early collaboration schemes, current limitations impede the practical usage of this method in most cases. Late collaboration shares only the perception results of the individual ITS-Ss, such as bounding box information of objects. In intermediate collaboration, feature-level outputs from prediction models are shared amongst ITS-Ss, providing upgraded perception capabilities compared to late collaboration with significantly less communication bandwidth from early collaboration.

The Collective Perception Service (CPS) [15] is a standard being developed by ETSI WG1 that specifies how a transmitting ITS-S can inform other ITS-Ss about the kinematic and attitude dynamics and other attributes of road users and objects detected by onboard sensors such as radars, LiDARs, and cameras. The CPS is expected to increase environmental awareness among ITS-Ss by sharing information about perceived objects. In the standard, the Collective Perception Message (CPM) protocol is defined, which enables the interoperable sharing of information about detected objects and road-related perception regions. As shown in Fig. 2.2, the CPM consists of information about the disseminating ITS-S, its sensory capabilities, detected objects, and detected road-related perception regions. The generation frequency of CPMs is quasi-periodic as the generation interval is specified, but it can be controlled dynamically by underlying control layers in the form of CPM generation events.

Since CPS focuses on the sharing of perception information, the temporal accuracy of the data and the delivery latency becomes critical to its reliability. In order to understand

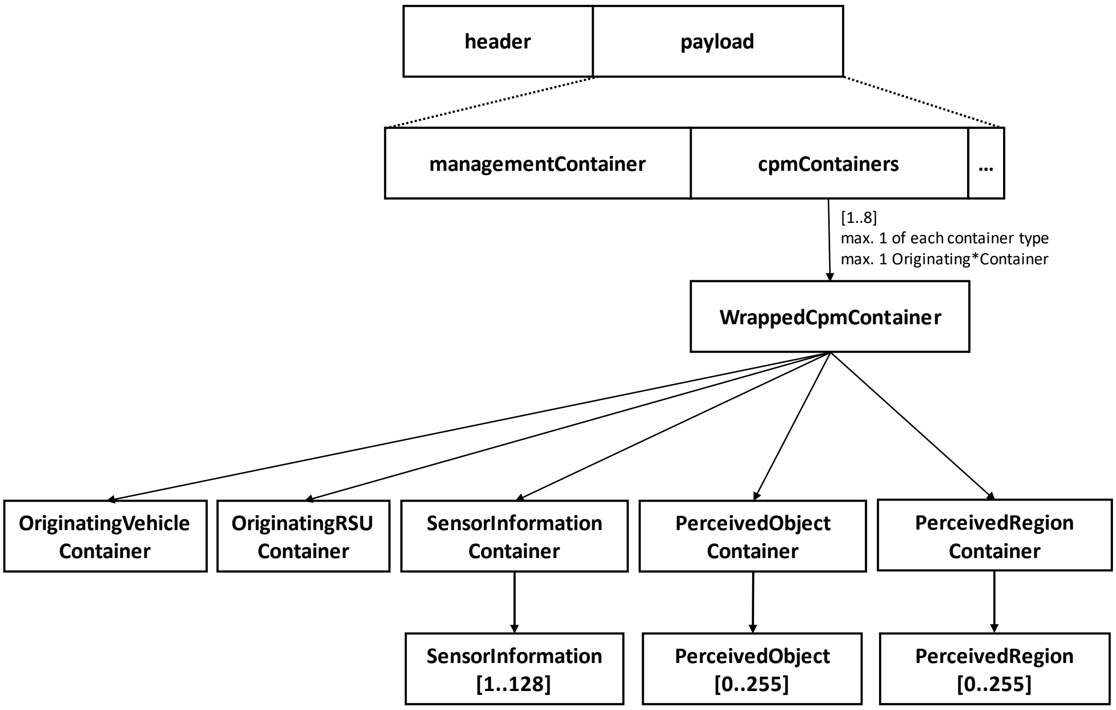

Fig. 2.2: CPM Structure [16]

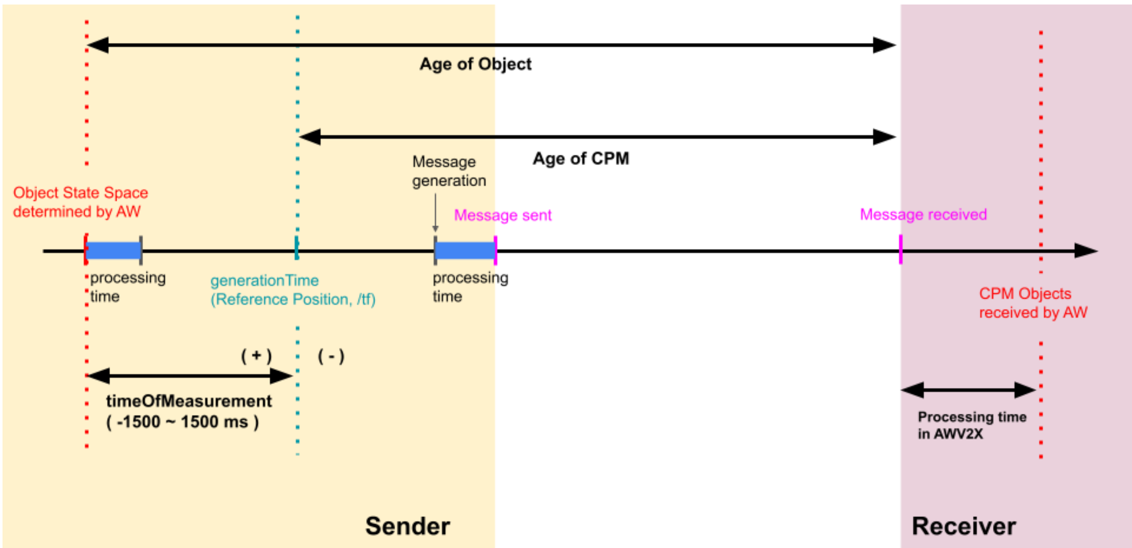

Fig. 2.3: Timeline of CPM Delivery

the different latency metrics related to the dissemination and reception of CPMs, Fig. 2.3 shows the timeline of how the objects included in a CPM are perceived, how the CPM is generated, and how the CPM is received and processed at the receiver.

A key focus of CPM-related research is information redundancy mitigation. While CPMs allows the sending of a large amount of object information, the more objects it sends, the larger the CPM packets will become. This can result in overloading the channel and creating unnecessary congestion while not providing the receiver with any more informational benefits. On the other hand, filtering too many of the objects can result in the receiver not being able to rely on the CPM-shared information. Therefore, object filtering methods for inclusion in a CPM are an important aspect of study, and the methods can affect the reliability of collective perception as a whole. Initial studies [38, 39] proposed filtering methods based on the objects' dynamics or semantic type. Studies by [40] proposed redundancy-level metrics and scores for qualitatively analyzing the different filtering rules.

Another issue in focus is how often the CPMs should be disseminated. The sending intervals of CPMs are important to the reliability of collective perception, and the right balance between bandwidth usage and the amount of information shared must be investigated. Currently, in the specification for CPS [15], the interval of CPM generation should be between 100ms and 1000ms. CPM generation can be quasi-periodic, meaning the interval between its generation can dynamically change depending on the congestion levels of the channel. The optimization of CPM generation frequency is heavily investigated, where [41] used a vision-based approach to identify connected vehicles and limit the number of generated CPMs. Message sizes of CPMs were evaluated in [42].

Development and deployment of proof-of-concept systems are also underway, many of which consider the cooperative perception of roadside infrastructure and autonomous vehicles. One group [28] developed and deployed a roadside cooperative perception system with an edge-cloud structure. Two sets of applications - traffic monitoring and road safety warning - were introduced and evaluated. In [43], the use case of collision avoidance with pedestrians in blind spots was considered, and CPMs were sent using C-V2X and ITS-G5.

The supervision of critical hotspots on the road, such as intersections, has been a major focus for infrastructure-aided cooperative schemes. Traffic lights and cameras with communication capabilities have the potential to improve road safety and traffic flow at smart intersections, and a variety of V2I applications are being implemented and evaluated in scenarios such as blind spot detection, signal phase control, and road condition monitoring. Car manufacturers are also already rolling out services that utilize such advanced schemes to provide increased safety features and information to drivers.

Chapter 3

Problem Statement

In this chapter, we first discuss existing issues with collective perception and examine how the current limitations can be overcome with new connective technology. Taking this into account, we lay out the requirements for a novel V2X communication router and Proof-of-Concept platform for experimenting and evaluating V2X-based collective perception applications. The research contributions of this thesis are finally introduced in the last section.

3.1 Existing Issues with Collective Perception

The current CPS standard [15] by ETSI focuses on broadcasting CPM packets from the sender ITS-S to arbitrary receiver ITS-Ss within the broadcast range of the wireless communication medium. This raises the concern that even when packet loss occurs on the wireless channel, both the sender nor receiver cannot be made aware of this in real-time. In order to reliably use CPM-shared information, the PDR of the CPM packets must be continuously monitored by both the sender and receiver ITS-Ss.

Furthermore, existing CPS works focus on sending CPMs on one radio access interface and do not deal with using multiple Radio Access Technology (RAT) to send CPMs over various network interfaces. More investigation is necessary into how hybrid transmission schemes can be utilized to realize a more reliable E2E exchange of collective perception information. With the emergence of communication standards such as DSRC or C-V2X, discussions on how we can take advantage of each communication method's strengths and use them in harmony are of utmost importance.

3.2 Requirements

One of the motivations behind our research is to realize fully connected and autonomous ITS-Ss and to utilize V2X communications to overcome the limitations of stand-alone systems. In order for a system to fully achieve its fullest potential, the following requirements are set, all of which must be fulfilled during the design, implementation, and deployment phases.

3.2.1 Scalability and Extensibility

In existing V2X schemes, different modules within the AD stack each had their own method of connecting to external nodes through multiple Wi-Fi or cellular interfaces. This limits the scalability of V2X utilization in the AD stack because multiple modules cannot reuse the same V2X router resources. In order to prevent this issue, the AD stack and V2X router must be decoupled, and the interface between them must be properly designed so that various applications can reuse communication interfaces and resources. In our proposed system, the V2X communication stack and the AD stack are loosely decoupled and can be placed on different hardware. The simple Ethernet connection between them will allow multiple modules within the AD stack to access and share the same V2X communication resources.

3.2.2 Compliance with Relevant Standards and Protocols

Various V2X communication standards have already been established in Europe, the US, and Japan. In order for a V2X router to function properly in a wide range of environments, the protocols used must comply with existing standards and also must be kept up-to-date with the newest changes. Protocols defined using the widely used Abstract Syntax Notation One (ASN.1) [44] standardized by the International Telecommunication Union (ITU) can be easily imported into our proposed system, enabling a wide range of supported standards and easy updates.

3.2.3 Robustness Against Packet Loss

The safety of cooperative systems that utilize CPS information is based on the reliability of the wireless network used for transmitting the CPMs. Degradation in the network condition due to issues such as congestion, radio interference, and large obstacles can lead to significant packet loss for CPMs, resulting in the CPM receiver nodes not being able to achieve full environmental awareness. Our proposed system utilizes a multi-RAT approach to provide dual-channel link redundancy to the connection used for CPM transmission. This enables CPMs to be successfully transmitted even in the event of significant packet loss on one of the channels. We also propose several methods of improving the reliability of collective perception.

3.2.4 Working Application Proof-of-Concepts (PoCs)

In order to verify the functionalities of the proposed system, a working prototype of a specific application must be implemented and evaluated in real-life scenarios. Especially in the context of collective perception, a working implementation of CPS is necessary, along with its evaluation and experimentation in field tests.

3.2.5 Open-source Development and Discussion

With the success of Linux, the power of open-source communities has shaped the way for faster development lead times and wider acceptance of new software. The proposed system must be made open-source to align with the developer ecosystem of Autoware under The Autoware Foundation [45].

3.2.6 Safety

For any AV, the safety of the passenger and Vulnerable Road Users (VRUs) is of utmost importance. Especially when the vehicles are connected with outside parties, possibly via multiple radio access technologies, security of not just the stand-alone intra-vehicle network but of external inter-connected networks becomes crucial.

3.3 Research Contributions

The goal of this thesis is to analyze the requirements for reliable cooperative perception amongst multiple ITS-Ss and to propose methods of improving the reliability of its utilization. In order to realize the end-to-end development, experimentation, and evaluation of cooperative perception between CAVs and RSUs, we discuss the issues and requirements for a new open-source implementation of a V2X router that can be fully integrated with a functional AD stack.

We propose and implement *AutowareV2X*, a V2X router that satisfies these requirements, capable of providing all the modules within an AD stack external network connectivity. AutowareV2X enables a highly flexible and extensible development and experimental platform where V2X applications could be built on top of existing AD software modules. End-to-end testing of the autonomous driving algorithms and the V2X network functionalities could be conducted, providing a more comprehensive evaluation of future connected mobility applications.

The Collective Perception Service (CPS) is implemented on top of AutowareV2X, enabling CAVs and RSUs to utilize CPM to share perception information. Three methods of improving the reliability of collective perception are proposed and explored: (i) the dual-channel hybrid delivery of CPMs, (ii) the real-time PDR monitoring of CPMs, and (iii) the adaptive dual-channel delivery of CPMs.

Evaluation of the proposed methods of improving the reliability of collective perception, along with the functional verification and performance evaluations of AutowareV2X, are done through (i) Simulation-based Experiments, (ii) Indoor Experiments, and (iii) Field Experiments in Outdoor Test Sites.

Chapter 4

Proposed System: AutowareV2X

We propose AutowareV2X, an implementation of a V2X communication module that is integrated into the autonomous driving (AD) software, Autoware, to enable the E2E development, experimentation and evaluation of CAVs. Recent studies have focused on only the prototype integration of V2X communication with a specific module in the AD stack, such as the perception or planning pipeline. In order to evaluate the E2E performance and qualities of connected autonomous driving among more advanced use cases, a more comprehensive platform is necessary.

AutowareV2X enables a highly flexible and extensible experimental platform where V2X applications can be built on top of existing autonomous driving software modules. E2E testing of both the autonomous driving algorithms and the V2X network functionalities could be conducted, providing a more comprehensive evaluation of future connected mobility applications. A high-level overview of how AutowareV2X can be utilized in a variety of ITS use-cases is shown in Fig. 4.1. AutowareV2X can be integrated easily into autonomous vehicles operating on Autoware to enable connectivity features and collection perception capabilities. CAVs that are equipped with AutowareV2X will be able to communicate with other connected road users that use the ITS-G5 communication standards in the form of V2V, V2I, V2P, or V2N communications. A variety of radio access technologies including but not limited to Wi-Fi, DSRC/802.11p, 4G, 5G cellular can be used as the underlaying network interface.

As one of the many potential applications of AutowareV2X, a Collective Perception Service (CPS) application was implemented to enable the sharing of perceived objects and to enhance environmental awareness of AVs. CPS-shared information can be utilized by AD systems to conduct cooperative collision avoidance maneuvers.

In order to realize more reliable collective perception, the mechanisms proposed in Chapter 5 were implemented as a module within AutowareV2X.

4.1 System Architecture

4.1.1 Overview

The system architecture for AutowareV2X is shown in Fig. 4.2. AutowareV2X is used as a V2X communication module integrated into Autoware. Autoware provides AD functionalities and uses sensing and HD map information in order to execute the entire AD stack from perception, localization, decision, and planning to control. It is based on a ROS2 middleware [46]; thus, all internal messages are shared through a publish-subscribe architecture. AutowareV2X is connected with Autoware through an Ethernet interface, and all relevant messages from Autoware are converted and packed into V2X messages. The implementation of basic functionalities for the ETSI ITS-G5 V2X communication

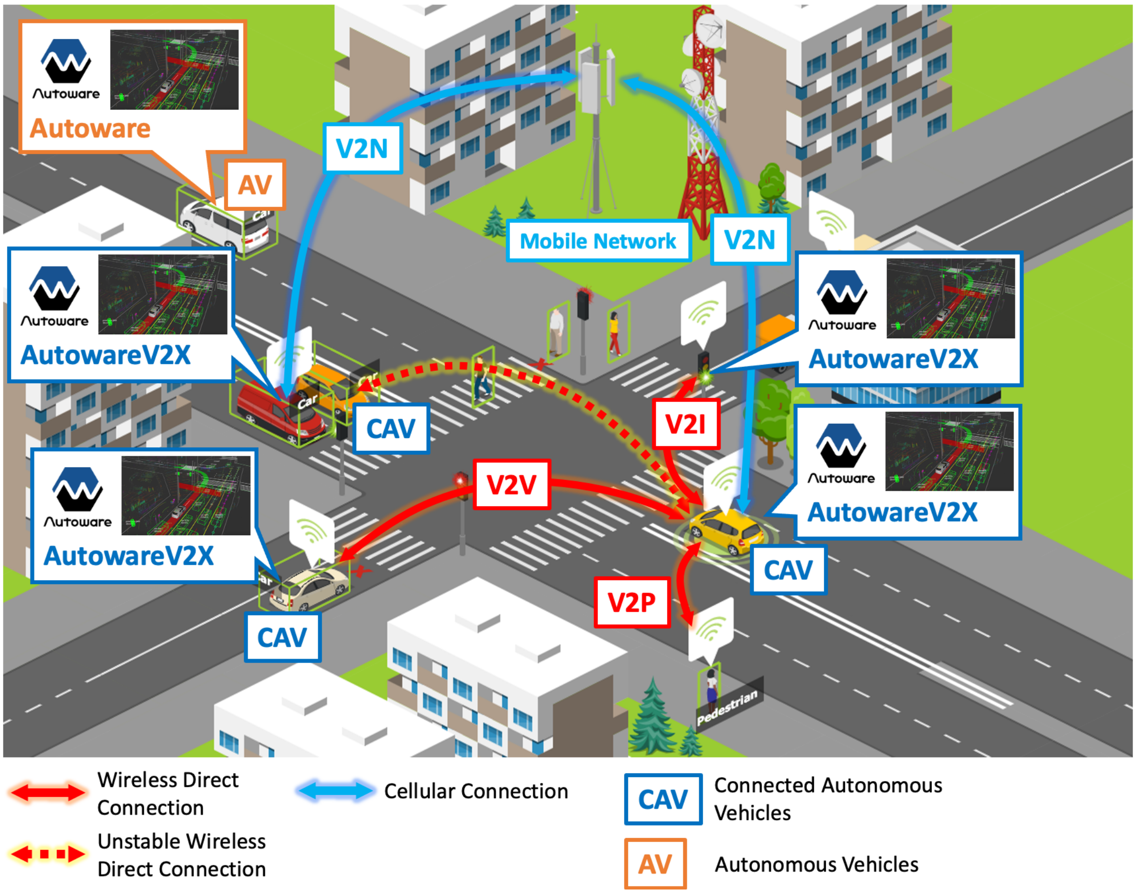

Fig. 4.1: High-level Overview of AutowareV2X Use Cases

stack is provided by Vanetza [18]. AutowareV2X extends the existing implementation of Vanetza to allow the seamless integration with Autoware, and the realization of reliable Collective Perception. Because both Autoware and AutowareV2X are only connected by a simple Ethernet interface, they can be loosely decoupled and the two components can be placed on separate hardware to accommodate for more use cases.

V2XNode functions as the interface between the V2X communication stack and the ROS2-based Autoware, while *V2XApp* is responsible for common tasks necessary for V2X communication, cross-layer network configuration, and the management of various facilities such as CAM and CPM. Specific details of the inner workings of each of the modules will be explained in the following section.

4.1.2 Vanetza

Vanetza is a software stack that provides generic ITS-G5 networking features and is designed to not be a standalone software but a library that can be linked to other application-specific programs. It implements the network and transport layers such as GeoNetworking and the Basic Transport Protocol (BTP) as well as the congestion control (DCC) and security cross-layers as shown in Fig. 4.3. The architecture of Vanetza for the realization of the ETSI ITS-G5 protocol suite is shown in Fig. 4.4. Vanetza is written in C++ and depends on very few portable libraries that can be built for almost any system where a standard-compliant C++ compiler and these dependencies (i.e. Boost, GeographicLib, Crypto++, and OpenSSL) are available. In addition to accommodating a wide variety of

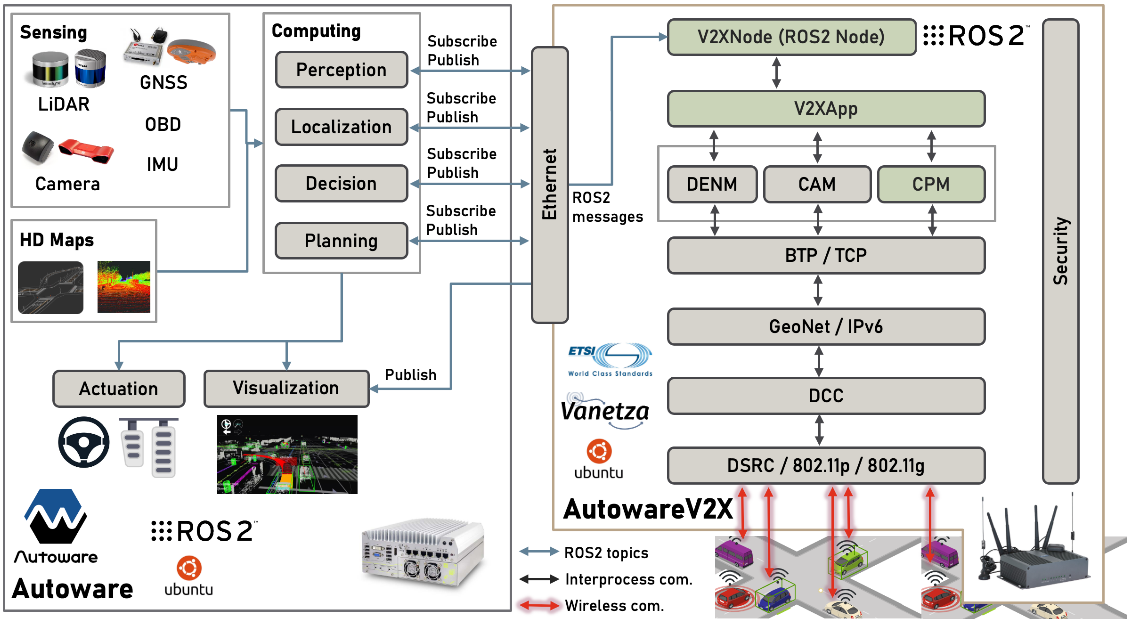

Fig. 4.2: System Design and Architecture

use cases involving embedded systems and actual hardware, Vanetza is also extensively used in VANET simulators such as Artery [47].

4.1.3 Autoware

Although there are several open-source standalone AD software stacks available [48, 26], we use Autoware [26] in the implementation of AutowareV2X, which is currently being actively developed by members of the Autoware Foundation (AWF) [45]. Autoware uses a pipeline architecture as shown in Fig. 4.5 to run the entire AD stack. It uses a high-definition point cloud map and vector map of the environment, in addition to a multitude of sensor inputs such as LiDAR point clouds, camera images, IMU values, and GPS coordinates, to perceive the environment, localize the ego-vehicle, make behavioral decisions, plan the global path and local trajectory, then control and actuate the vehicle to follow the generated path. The newest version of Autoware, Autoware.universe, is currently open-sourced on Github [49] and new features and algorithms are constantly being added. Autoware is implemented in C++ on top of a ROS2 middleware [46] that provides a publish-subscribe interface for sharing data between different modules. Each feature in Autoware, such as ego-vehicle localization is implemented with a collection of ROS2 nodes that work together to output final localization results onto a specific ROS2 topic.

4.2 Implementation Details

This section will describe the implementation details of AutowareV2X, along with how the methods introduced in Chapter 5 are realized using AutowareV2X. The basic core modules are explained in Section 4.2.1 while the implementation of the TCP/IP application used to realize dual-channel delivery is described in Section 4.2.2.

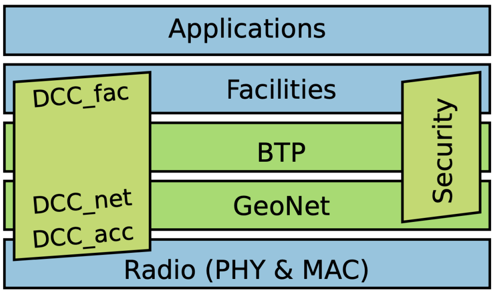

Fig. 4.3: Layers supported by Vanetza [18]

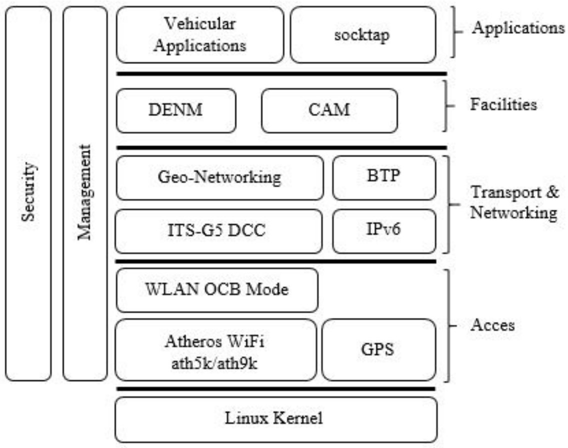

Fig. 4.4: Vanetza architecture for ITS-G5 Stack [50]

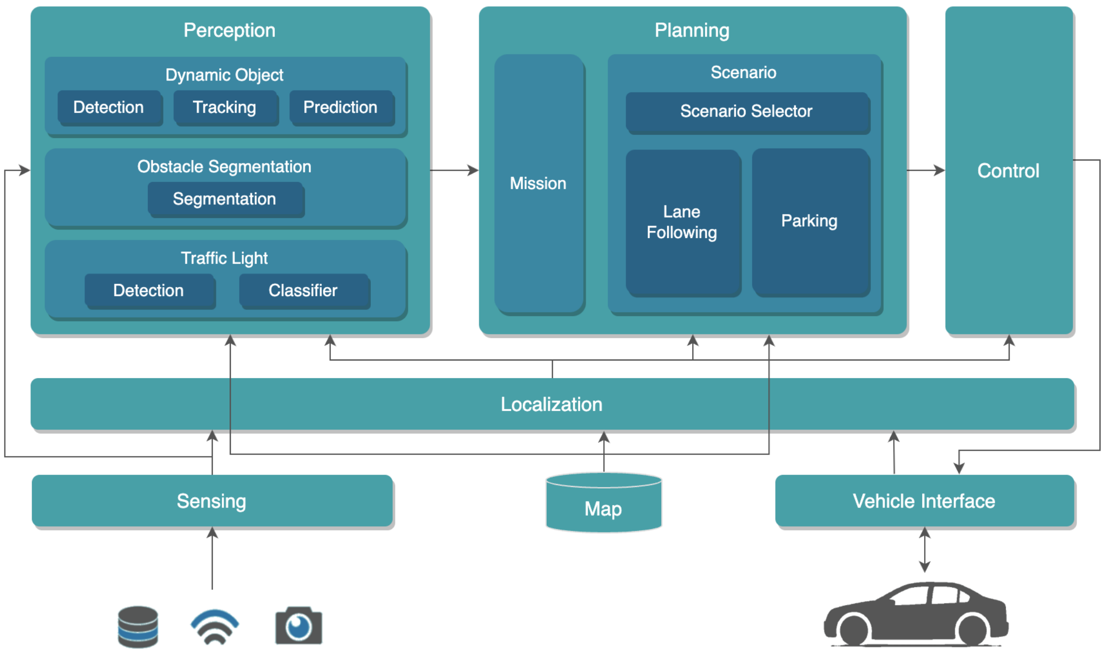

Fig. 4.5: Autoware Pipeline Architecture [49]

4.2.1 Basic Core Modules

Fig. 4.6 shows the implementation diagram of *V2XNode*, *V2XApp*, and *CPMApplication*. *V2XNode* is a ROS2 Node, and acts as the interface between AutowareV2X and Autoware. It receives information such as the objects detected by the local sensors, the ego vehicle's current position/ velocity, planned routes, etc. from Autoware in the form of ROS2 topics. For each ROS2 topic that AutowareV2X uses, a callback function must be present in the *V2XNode*, such as the *tfCallback* or *objectsCallback* functions. These callback functions will be called when there is a message published to the specified ROS2 topic. Upon the initialization of *V2XNode*, a *V2XApp* will be launched in a new thread. *V2XApp* configures the specified network interfaces to use the ETSI C-ITS protocol stack and relays the information provided by the *V2XNode* to the relevant applications. When *V2XApp* is initiated, the *start* function is initialled called to coordinate other services. The callback functions are present here as well, to redirect the necessary ROS2 topic information to each of the applications such as the *CPMApplication*.

The *CPMApplication* is an implementation of CPS and includes the following features:

- CPM generation and dissemination: The *send* function handles the generation and dissemination of the CPM packet. Once a CPM packet is generated, it is passed down to lower layers of the ITS-G5 protocol suite until it has finally broadcasted through the specified network interface.
- Scheduling of CPM dissemination: The *set_interval*, *schedule_timer*, *on_timer* functions are responsible for setting up timers with certain intervals to allow CPM generation and dissemination.
- CPM reception in both a single-channel and dual-channel configuration: The main function that handles CPM receptions is the *indicate* function. Here, the CPM packet is dissected and values are stored accordingly. Once the information of the shared objects are collected, it is published to Autoware through the *publishObjects* function in *V2XNode*.
- Selection of objects to include in the CPM: The selection of objects to include in the CPM is specified in the *updateObjectsList* and *setAllObjectsOfPersonsAnimalsToSend* functions. The flowchart for how the objects are selected are shown in Fig. 4.7. The logic used for AutowareV2X is based on the specifications laid out by ETSI in the CPS standards [15].
- Updating of relevant values necessary for CPM generation: Before a CPM can be sent, the values of its content must be regularly updated in the *CPMApplication*. The *updateXX* functions are used to update the specific values necessary for the generation of a CPM such as the ego-vehicle's MGRS coordinates, the Reference Point of the CPM, the heading angle of the vehicle, and the timestamp of when the Reference Point was decided.
- Recording of sent and received CPMs to a local database: CPMs that were sent and received is stored in a local database called *cpm_sent* and *cpm_received* to facilitate the monitoring of CPM reception quality as per mentioned in Chapter 5.

4.2.2 TCP/IP Application

In order to realize the Dual-channel Hybrid Delivery of CPMs proposed in Section 5.1, a TCP/IP application for AutowareV2X was also implemented, where the ETSI-compliant messages such as CAM, DENM, or CPM can be encapsulated in TCP/IP packets to be sent to the receiver. The implementation diagram of this TCP/IP application is shown in Fig. 4.8. Once the *V2XNode* instantiates and when the Dual-channel Hybrid Delivery is enabled, a TCP/IP Application for both the sending and receiving functions are launched on separate threads. For each sending and receiving loop, asynchronous methods are used to either connect to or accept TCP/IP connections. For the sender loop, a TCP connection is first initiated to the specified receiver, then once the connection is established, the content to be delivered such as a CPM packet is encapsulated in a TCP/IP packet before being sent on the specified network interface. On the receiver loop, after accepting a TCP connection on a specified port, the contents are received then decapsulated from the TCP packet into the original CPM packet. The CPM packet is then sent to the *CPMApplication* to be utilized as normal CPMs.

The development of AutowareV2X is an ongoing project and the source code can be

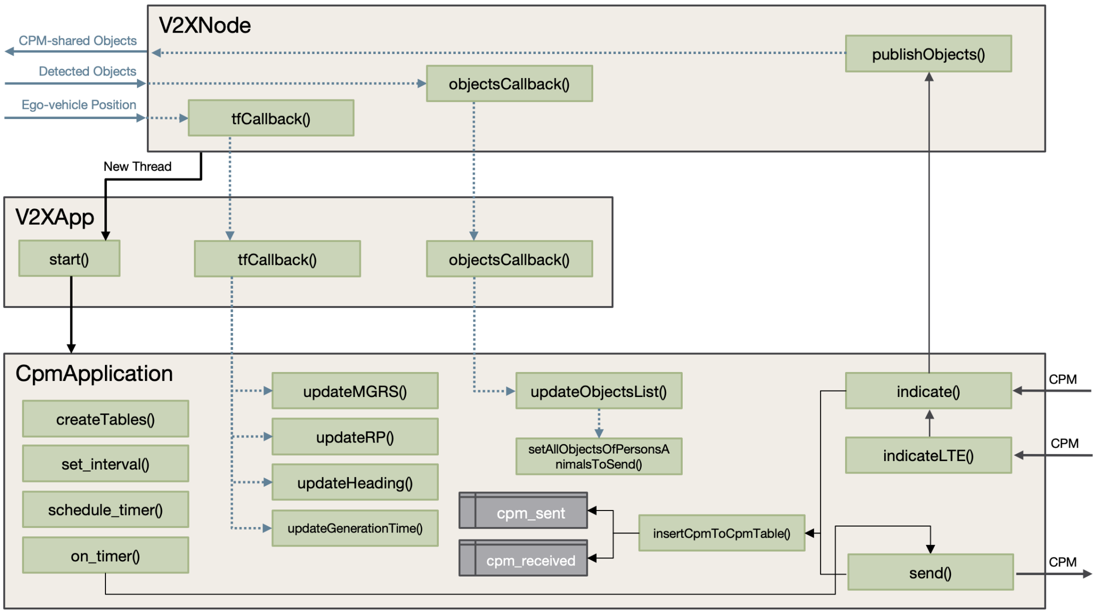

Fig. 4.6: Implementation Diagram of V2XNode, V2XApp, and CPMApplication

found at https://github.com/tlab-wide/AutowareV2X, while the documentation and tutorials can be found at https://tlab-wide.github.io/AutowareV2X/latest/.

4.2.3 CPM Assistive Messages (CPAM) Manager

In order to realize the (i) real-time PDR monitoring on the sender and receiver ITS-Ss and (ii) the adaptive dual-channel CPM delivery described in Chapter 5, the implementation of a module that can send and receive CPM Assistive Messages (CPAM) is necessary. The *CPAM Manager* that is responsible for this task is implemented as a module that can be integrated into the core AutowareV2X. *CPAM Manager* can be set up in two modes: the RSU mode and the Vehicle mode, depending on whether it is the CPM sender or receiver. The implementation diagram for the *CPAM Manager* in each of the two modes is shown in Fig. 4.9 and Fig. 4.10. In both modes, the *CPAM Manager* launches one CPAM Receiver module and multiple CPAM sender modules depending on the number of vehicles or RSUs it must communicate with. In the RSU mode, the CPAM Sender initiates a TCP/IP connection with all vehicles in the vicinity and periodically sends the *CPAM Type 1* packet that includes information about how many CPMs the RSU has sent in a given time window. Simultaneously, the CPAM Receiver accepts any connections from nearby vehicles and receives the *CPAM Type 2* packet that includes the PDR information of that specific vehicle. The PDR value is then alerted to the core AutowareV2X modules so that AutowareV2X can decide whether or not to initiate or stop the dual-channel hybrid delivery of CPMs. In the vehicle mode, the CPAM Sender sends the *CPAM Type 2* packet to RSUs in the vicinity in order to alert them of the current PDR value of the vehicle. At the same time, the CPAM Receiver receives the *CPAM Type 1* packet that includes the number of CPMs that were sent from an RSU, so that the vehicle can calculate its current PDR of CPMs received from a specific RSU. If the PDR is significantly degraded for a certain RSU, the core AutowareV2X can anticipate incoming CPMs from a different channel or CPMs from other RSUs with better communication conditions.

In this way, the *CPAM Manager* is responsible for the sending and receiving of CPAMs

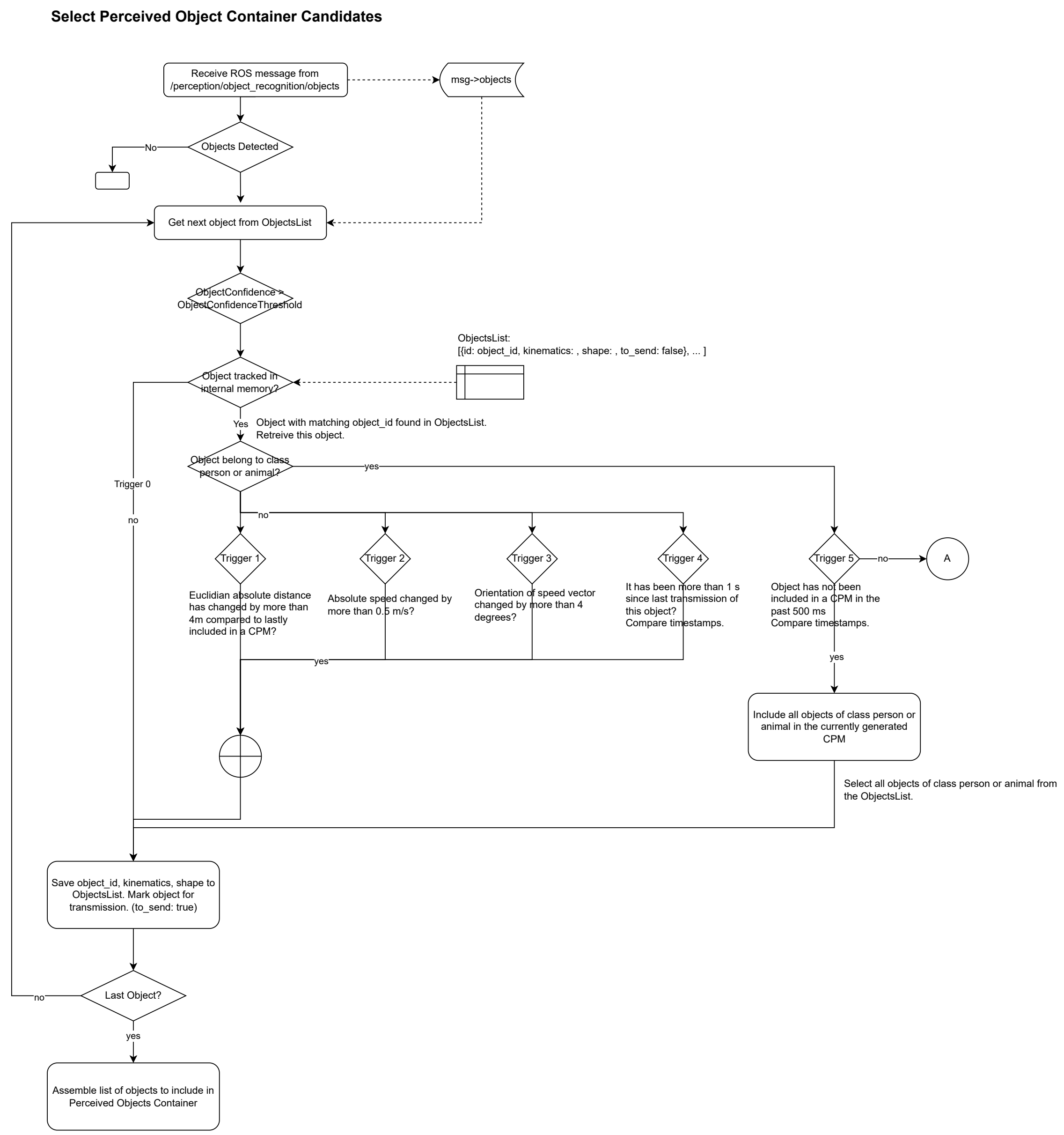

Fig. 4.7: Perceived Objects Selection Logic

on the more reliable 4G channel, and the mechanism provided here allows the realization of our proposed methods.

4.2.4 Service Discovery of RSUs

In this research, we mainly consider the scenario of V2I communication where RSUs share information with CAVs to provide better collective perception. A dual-channel communication medium is created between the RSUs and CAVs, and the transmission of Wi-Fi-based CPMs and LTE-based CPMs are conducted. While the Wi-Fi-based CPMs are transmitted based on the ETSI ITS-G5 protocol suite in the form of single-hop broadcasts, the LTE-based CPMs are transmitted in a unicast style on top of an ordinary IP network supported by existing 4G backbone infrastructure. Therefore, in order for the CAVs to be able to dynamically figure out which IP address to send the TCP-encapsulated CPMs

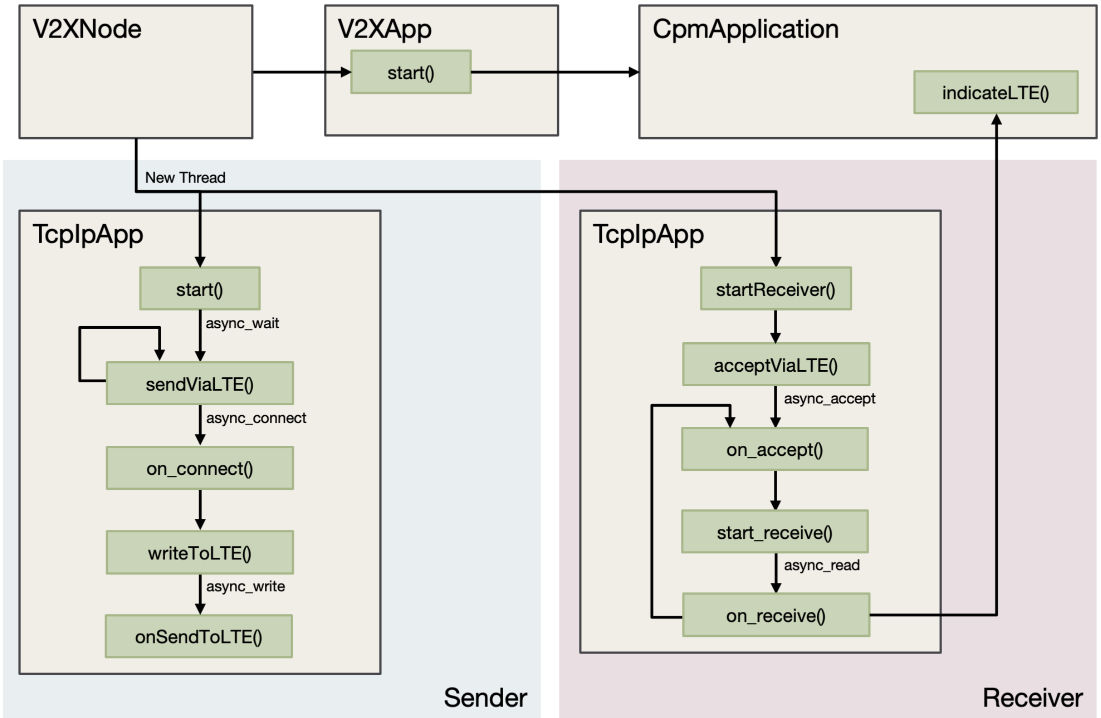

Fig. 4.8: Implementation Diagram of TCP/IP Application

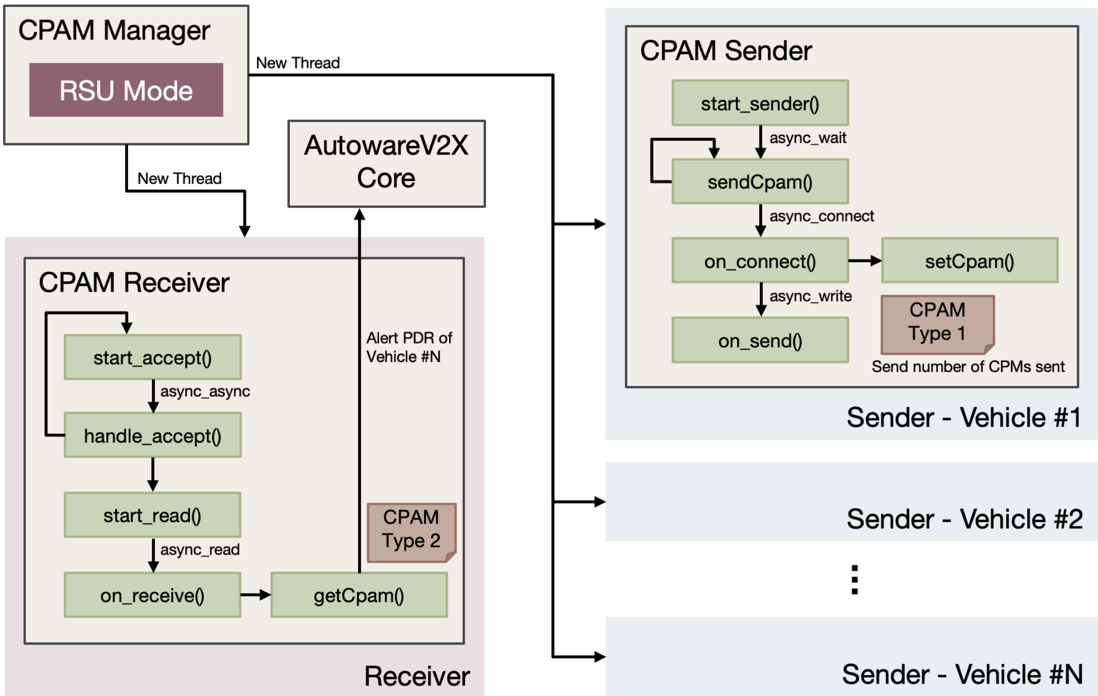

Fig. 4.9: CPAM Manager for RSU

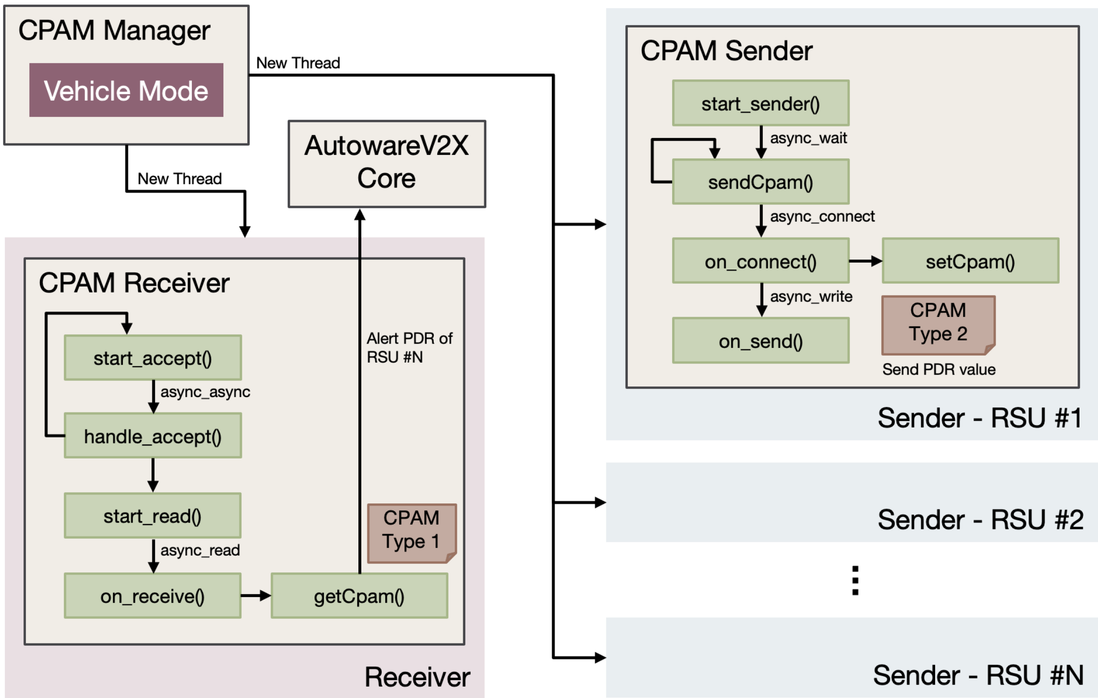

Fig. 4.10: CPAM Manager for Vehicle

to, they must conduct an "RSU Service Discovery" step to find nearby RSUs and retrieve the IP addresses of the RSUs. Although the investigation of the "Service Discovery" related to dynamic CAVs and RSUs is out of the scope of this thesis, we have considered this issue and have concluded that with the right design and specification of a "Service Discovery Protocol" for RSUs and CAVs, their IP addresses can be shared between each other. Therefore, with this in mind, in this research, we assume that both the RSU and CAV are aware of each other's IP addresses.

Chapter 5

Improving Reliability of Collective Perception

In this chapter, we propose methods of improving the reliability of collective perception by providing redundant CPM delivery mechanisms. All of the methods here are implemented and integrated into our proposed system of AutowareV2X.

Section 5.1 proposes a method of dual-channel delivery of CPMs using a hybrid radio access technology (RAT) architecture. Section 5.2 then uses this dual-channel approach to propose a method for real-time PDR monitoring of CPM packets on both the sender and receiver ITS-Ss. Section 5.3 then discusses how the real-time PDR monitoring scheme and the dual-channel hybrid delivery mechanism can realize an adaptive CPM dissemination scheme that can both improve the reliability of Collective Perception while limiting the network resources used.

5.1 Dual-Channel Hybrid Delivery of CPMs

For a more reliable realization of CPS, we propose a multi-Radio Access Technology (RAT) approach to provide dual-channel link redundancy to the connection used for CPM transmission. The proposed concept, namely the "**Dual-Channel Hybrid Delivery System**", is shown in Fig. 5.1. We consider a scheme where RSUs send their perception information in the form of CPMs to nearby connected vehicles. The RSUs communicate with the surrounding vehicles over Channel 1 with Vehicle-to-Infrastructure (V2I) communications. Access technologies used here are direct communication mediums such as ITS-G5, 802.11p, or normal Wi-Fi-based solutions such as 802.11g. In parallel to CPM dissemination on this main direct V2I communications channel (Channel 1), the same CPMs are also sent from the RSUs to mobile networks through the Internet and are disseminated to the vehicles on a more stable secondary communications channel (Channel 2) such as 4G. This communication structure used here is based on the Vehicle-to-Network (V2N) communications scheme and cellular technologies can be utilized.

The main advantage of the Dual-Channel Hybrid Delivery System is that even if there is significant packet loss on the direct V2I communication interface between the RSU and the vehicles, the same information can be reliably delivered to the vehicles via the redundant route based on the V2N communication scheme over the mobile cellular network. This is based on the assumption that V2N communication medium (Channel 2) has significantly smaller packet loss rates compared to Channel 1, although it may have longer latency times.

The algorithm for the dual-channel delivery of CPMs is described in Algorithm 1. Using the method proposed in Section 5.2 with a newly proposed type of message called the

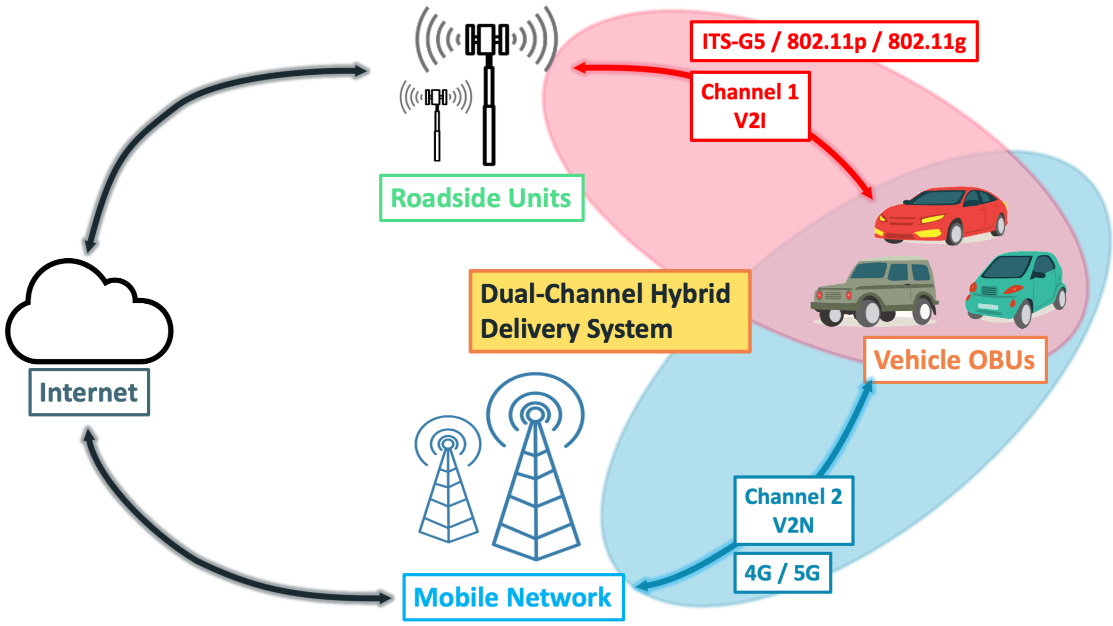

Fig. 5.1: Dual-Channel Hybrid Delivery System

Algorithm 1 Sender Dual-channel CPM Delivery

1: $\theta \leftarrow$ *PDR Threshold for dual-channel CPM Delivery*
2: $\tau \leftarrow$ *Interval for CPM delivery on second channel*
3: **while** $CPAM_i$ *is received* **do**
4: $\quad PDR_\omega \leftarrow$ *PDR of* $CPAM_i$ *sender*
5: $\quad$ **if** $PDR_\omega < \theta$ **then**
6: $\quad\quad$ *Start CPM delivery at* τ *interval on second channel*
7: $\quad$ **else**
8: $\quad\quad$ *Stop dual-channel CPM delivery*

CPAM, the PDR of CPMs can be monitored on both the sender and receiver ITS-Ss. The dual-channel delivery algorithm utilizes the PDR value on the sender ITS-S, where if the PDR value of a certain receiver ITS-S goes below a certain threshold, the dual-channel delivery of the CPMs is initiated. When network conditions improve, and the PDR for Channel 1 goes above the threshold, then the dual-channel delivery of CPMs is stopped. This method is proposed in more detail in Section 5.3 as the "Adaptive Dual-Channel CPM Delivery Method".

When the same CPMs are sent from the sender ITS-S to the receiver ITS-S over multiple channels, the receiver ITS-S will have to decide which CPMs to accept into its system and use, and which CPMs to discard and ignore. Thus, we introduce a new concept for the use of CPMs in the receiver ITS-S, which is the **Acceptance** and **Rejection** of CPMs. In conventional methods of using CPMs, the receiver ITS-S will typically always **Receive** and **Accept** a CPM. However, when considering multiple channels, we must consider scenarios where the receiver ITS-S will **Receive** but **Reject** certain CPMs.

Fig. 5.2 shows the mechanism of how the acceptance and rejection of CPMs are decided in this proposed method. The CPMs sent over the V2I communications channel (Channel 1) are depicted with green arrows, and the CPMs sent over the secondary cellular channel (Channel 2) is depicted with red arrows. It can be seen that the red arrows are more

Algorithm 2 CPM Acceptance and Rejection at Receiver ITS-S

$T_{GT} \leftarrow$ *generationTime of most recently accepted CPM*
while *CPM is received* **do**
  $T'_{GT} \leftarrow$ *generationTime of newly received CPM*
  $GTD = T'_{GT} - T_{GT}$
  if $GTD > 0$ **then**
    Accept(CPM)
    $T_{GT} = T'_{GT}$
  else
    Reject(CPM)

slanted than the green arrows, representing the fact that Channel 2 has more delivery latency when it uses 4G connections. Some of the green arrows are also stopping in the middle, representing a packet loss on Channel 1.

In order to decide which CPMs to accept into the system, the *generationTime* value inside the CPM is used. The *generationTime* value is included in all CPMs, and it is defined as the timestamp for when the CPM's sender ITS-S's position was decided. A larger *generationTime* value means that the CPM was generated later in time, so by comparing the *generationTime* values, we can identify which CPMs were more newly generated. Therefore, when CPMs from multiple channels are being received by a receiver ITS-S, the receiver ITS-S can decide on the acceptance and rejection of a CPM based on the difference of the *generationTime* values of CPMs.

Here, we define the *Generation Time Difference* denoted as GTD given by

$$GTD = T'_{GT} - T_{GT} \tag{5.1}$$

where T'_{GT} is the *generationTime* for the newly received CPM while T_{GT} is the *generationTime* for the most recently accepted CPM. A positive GTD would signify that the newly received CPM includes fresher information compared to the previously accepted CPM, while a negative GTD would mean that it includes older information than a previously accepted CPM.

In Fig. 5.2, the blue d values represent a positive GTD value, while the orange d values represent a negative GTD value. For each CPM that reaches the Vehicle, which is the receiver ITS-S in this case, the GTD is calculated, and if that value is positive, then the CPM is accepted (as portrayed with the green checkmark icon).

Using the GTD value, the algorithm for the acceptance and rejection of CPMs in the receiver ITS-S is described in Algorithm 2.

When a new CPM is received from either of the dual channels, the GTD is calculated by finding the difference between the *generationTime* included in the received CPM and the *generationTime* of the previously accepted CPM, as stored in the system. If the GTD value is positive, then the new CPM is accepted, and the *generationTime* of the most recently accepted CPM is updated. Otherwise, the CPM is discarded and ignored.

5.2 Real-time Packet Delivery Rate (PDR) Monitoring

After an ITS-S generates a CPM, the packet is delivered to nearby vehicles and infrastructure through a single-hop broadcast (SHB) communication scheme. Fig. 5.3 shows this process where an RSU is responsible for disseminating CPMs, and Cars 1, 2, and A receive the CPMs. Since the CPMs are broadcasted once, and there are no retransmission

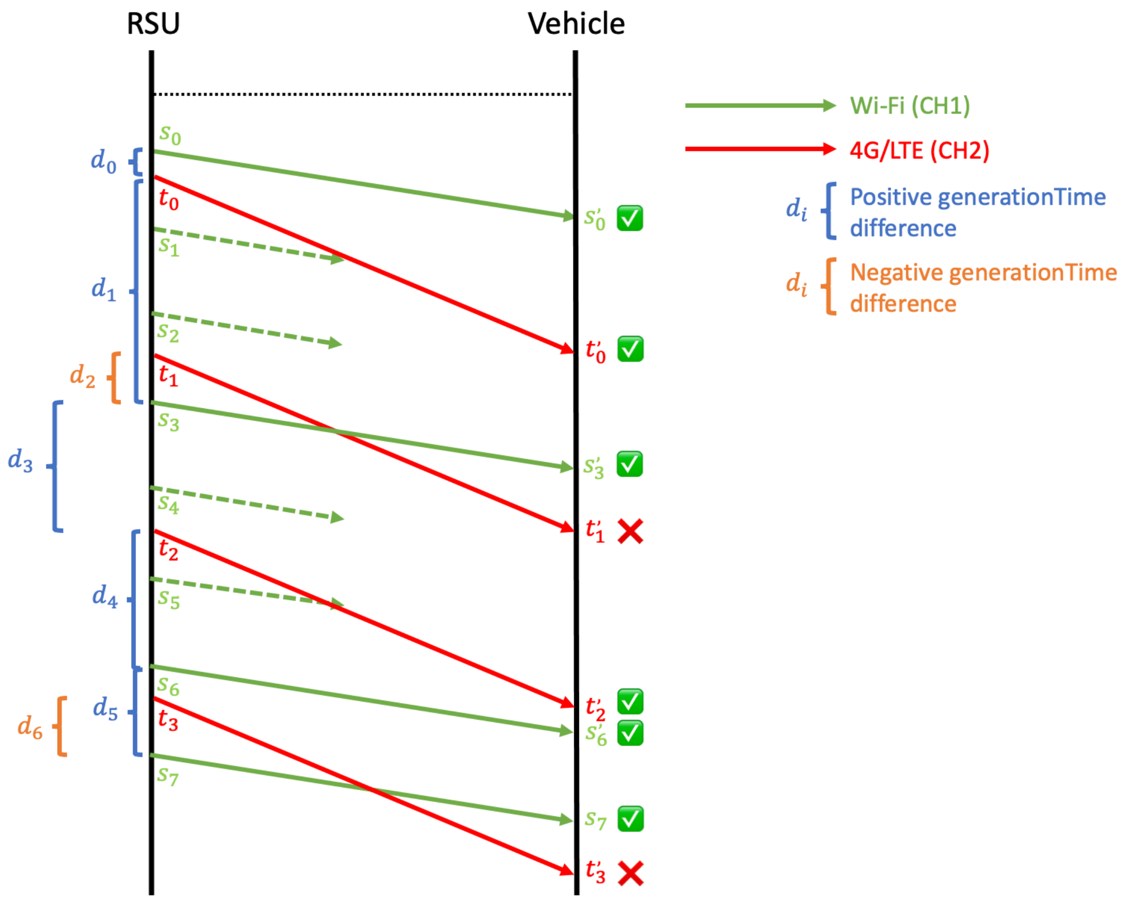

Fig. 5.2: Time diagram of CPM acceptance/rejection using *generationTime difference*

schemes, regardless of the condition of the communication channel or the successful transmission of the CPM to a receiver, the sender of the CPM has no way of knowing whether the CPM was actually received by important ITS-Ss in the vicinity. In other words, the sender of the CPM cannot monitor the packet delivery rate (PDR) of the CPMs it has sent. Similarly, the receiver ITS-S cannot calculate the PDR of CPM delivery as well since it does not know how many CPMs were sent in a given time interval in the first place.

In order to realize reliable collective perception, the PDR of the CPMs must be monitored continuously, and both the receiver and sender ITS-Ss must be made aware of this value. In order to achieve this, the more reliable cellular channel (Channel 2 in Fig. 5.1) is used as a control plane to send information necessary for the calculation of the PDR.

As a way of sending the necessary information, we propose the **CPM Assistive Messages (CPAMs)**. Two types of CPAMs are defined, with each of their purposes and the data they include described in Table 5.1. The packet delivery unit of CPAMs is shown in Fig. 5.4. There are two types of CPAMs, and both of them are used in order to allow the sender and receiver ITS-Ss to be able to continuously monitor the PDR of a given channel.

Algorithm 3 describes the real-time PDR monitoring mechanism in detail. The CPM sender ITS-S is responsible for not only sending the CPMs on the main V2X channel (Channel 1) but for periodically sending the *CPAM Type 1* packets on the secondary

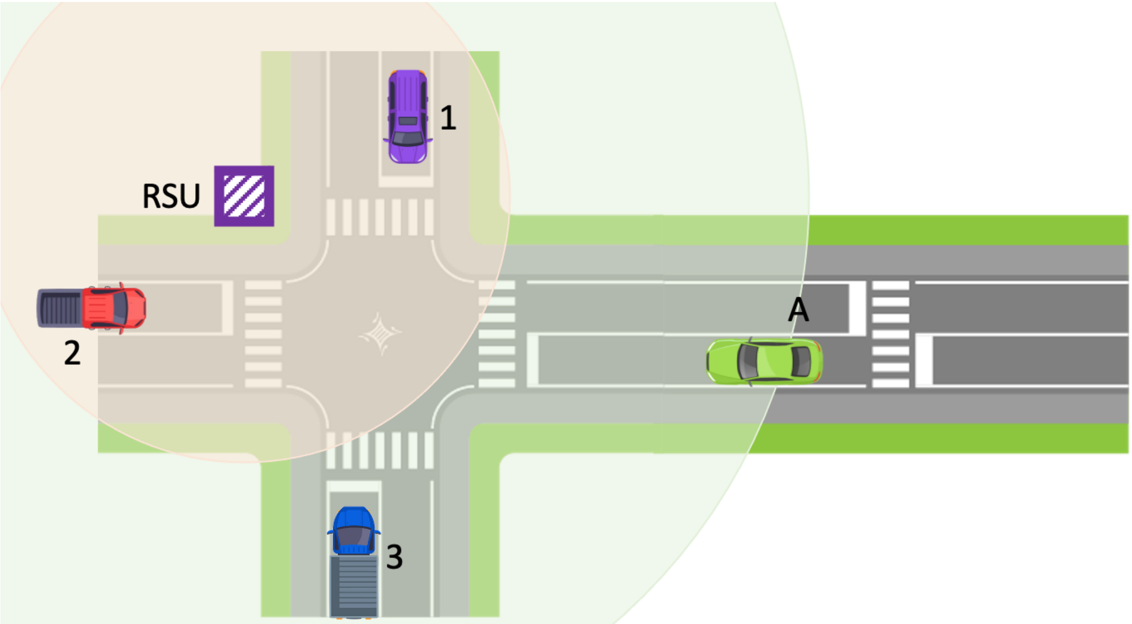

Fig. 5.3: Delivery Range of CPMs

Table 5.1: CPAM Types and Purpose

CPAM Type	Purpose	How it is sent	Included Data
Type 1	PDR calculation on receiver ITS-S	CPM sender ITS-S to CPM receiver ITS-S	Number of CPMs sent from the sender ITS-S in a given time interval
Type 2	Relay the PDR value at the receiver ITS-S to the sender ITS-S	CPM receiver ITS-S to CPM sender ITS-S	PDR value at the receiver ITS-S

Algorithm 3 PDR Calculation and CPAM Generation at the Receiver ITS-S

```
1: τ ← Interval for CPAM delivery on second channel
2: ω ← Window size for PDR calculation
3: while CPAM(count(CPMs, ω)) is received do
4:     PDR_ω ← Calculate PDR for time ω
5: for every τ milliseconds do
6:     Generate new CPAM(PDR_ω)
7:     Send CPAM(PDR_ω) to cpm_sender
```

cellular channel (Channel 2). The *CPAM Type 1* packet is sent from the CPM sender ITS-S to the CPM receiver ITS-S and includes information about how many CPMs were sent within the time intervals specified in the CPAM PDU. Upon receiving this *CPAM Type 1* packet, the CPM receiver ITS-S can calculate how many CPMs it has actually received within the specified time interval, and by comparing that with the number of CPMs that were sent, the PDR for that time window can be calculated. Then once the PDR for a specific time window is calculated, the CPM receiver ITS-S then generates a *CPAM Type 2* packet that includes the calculated PDR value and proceeds to send the packet back to the CPM sender ITS-S. In this way, the CPM sender ITS-S is also able to be made aware of the PDR for a specific CPM receiver ITS-S. This step is denoted as Step 1. "Sending of CPAM(PDR=70%)" in Fig. 5.5.

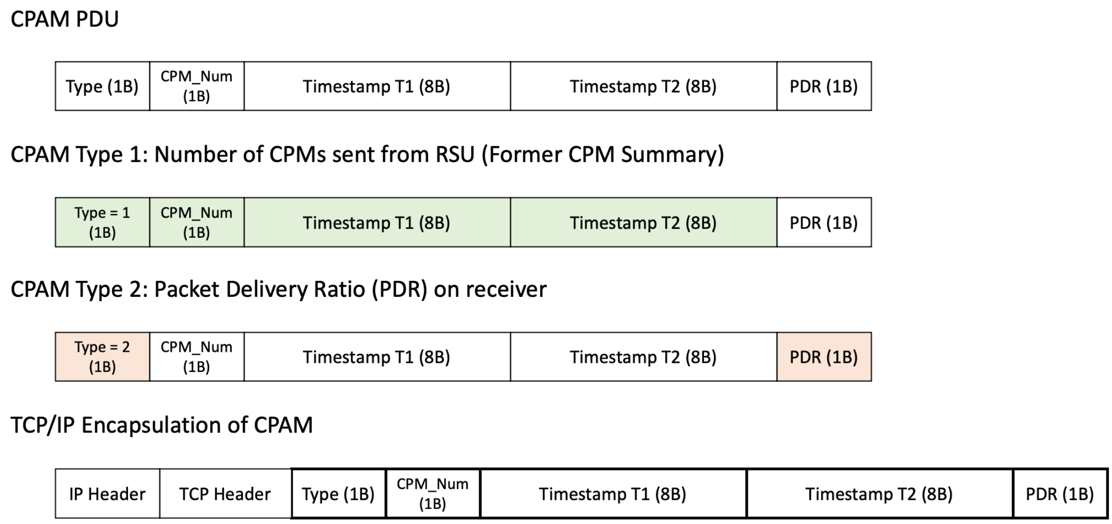

Fig. 5.4: Packet Delivery Unit (PDU) of CPM Assistive Messages (CPAM)

5.3 Adaptive Dual-Channel CPM Delivery

With the methods proposed in the previous two sections, the dual-channel delivery of CPMs as well as the real-time PDR monitoring of CPMs can be conducted. By combining the two methods, we can further investigate a scheme where the dual-channel delivery of CPMs can be turned on and off depending on the real-time PDR values. Fig. 5.5 shows the time diagram of how this scheme can play out. In Step 1, the *CPAM Type 2* with the PDR value for the receiver ITS-S is sent from the CPM receiver ITS-S to the CPM sender ITS-S. Once the PDR is received on the CPM sender ITS-S, it can compare the PDR to a predefined threshold and depending on the value, it can dynamically turn the dual-channel delivery of CPMs on and off. In this way, bandwidth on the secondary cellular channel can be conserved and dual-channel delivery of CPMs will not be conducted when it will not benefit the CPM receiver ITS-S as much.

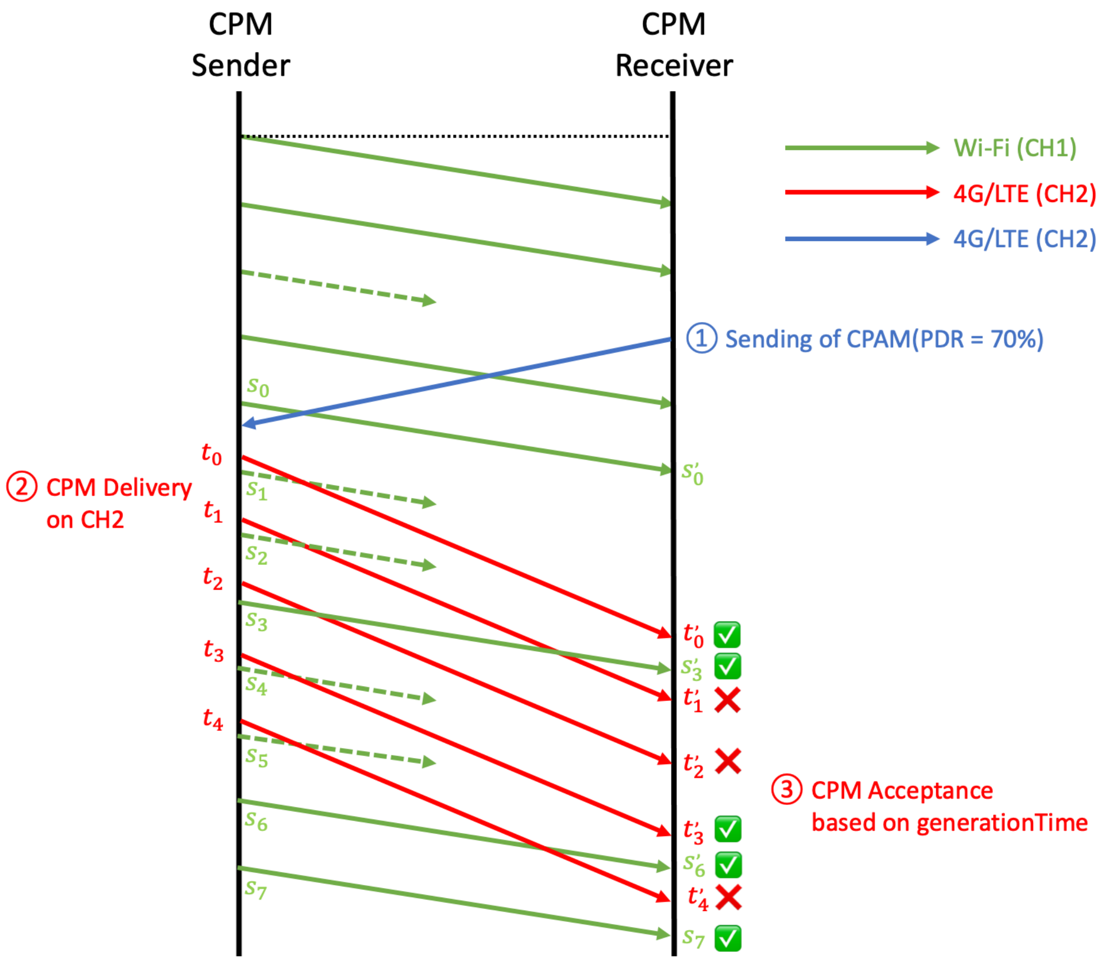

Fig. 5.5: Time Diagram of Adaptive Dual-Channel CPM Delivery

Chapter 6

Experiments and Evaluation

6.1 Overview

The various features of AutowareV2X, including the dissemination and reception of ETSI-compliant CPMs and implementation of the proposed methods of Chapter 5 were evaluated through (i) Functional Verification in Simulation-based Environments, (ii) Indoor Experiments using actual hardware in Indoor scenarios, and (iii) Field Test Experiments in outdoor testing facilities. Section 6.2 describes the functional verification of all the implemented modules in a simulation-based environment. Here, we confirmed that all the modules and features including our proposed methods were working properly. Section 6.3 introduces the evaluation metrics used in analyzing the results for the later sections. We propose some novel metrics used for evaluating the dual-channel delivery of CPMs. Section 6.4 then describes some preliminary experiments using actual hardware in an indoor environment. Some important metrics were gathered here and evaluated. Finally, in Section 6.5, we conduct our main evaluation experiments in an outdoor test field and analyze all of the gathered data.

The following sections dive into each of the experiments and evaluation methods.

6.2 Functional Verification in Simulation-based Environments

Functional verification of AutowareV2X was first conducted in simulation-based environments using Docker-based containers and Autoware's Planning Simulator [51]. An example setup of two ITS-Ss is shown in the diagram in Fig. 6.1. The sender and receiver ITS-Ss are composed of Docker containers running the newest Autoware AD software stack and AutowareV2X. They are each placed in a virtual network called the *Sender Net* and *Receiver Net*, respectively. The AutowareV2X containers for both the sender and receiver ITS-Ss are placed in a separate virtual network called the *V2X Net*, where the network conditions for the direct V2X communications based on ITS-G5 are simulated. Similarly, they are also placed in the *LTE Net* virtual network, where the 4G network conditions are simulated. The *V2X Net* is primarily used for CPM dissemination and reception, and the *LTE Net* is used for the dual-channel delivery of CPMs over TCP/IP encapsulation and the dissemination of CPAMs.

A Wireshark capture of the CPMs generated in the simulation-based environment is shown in Fig. 6.2. The CPM packet is being successfully dissected by Wireshark, along with the GeoNetworking and BTP frames. Inside the *Intelligent Transport Systems* PDU, the *ItsPduHeader* includes the protocol version, message ID, and station ID. This is followed by the *CollectivePerceptionMessage* PDU, which includes the values for the perceived objects.

Similarly, a Wireshark capture of the TCP/IP encapsulated LTE-based CPMs are shown

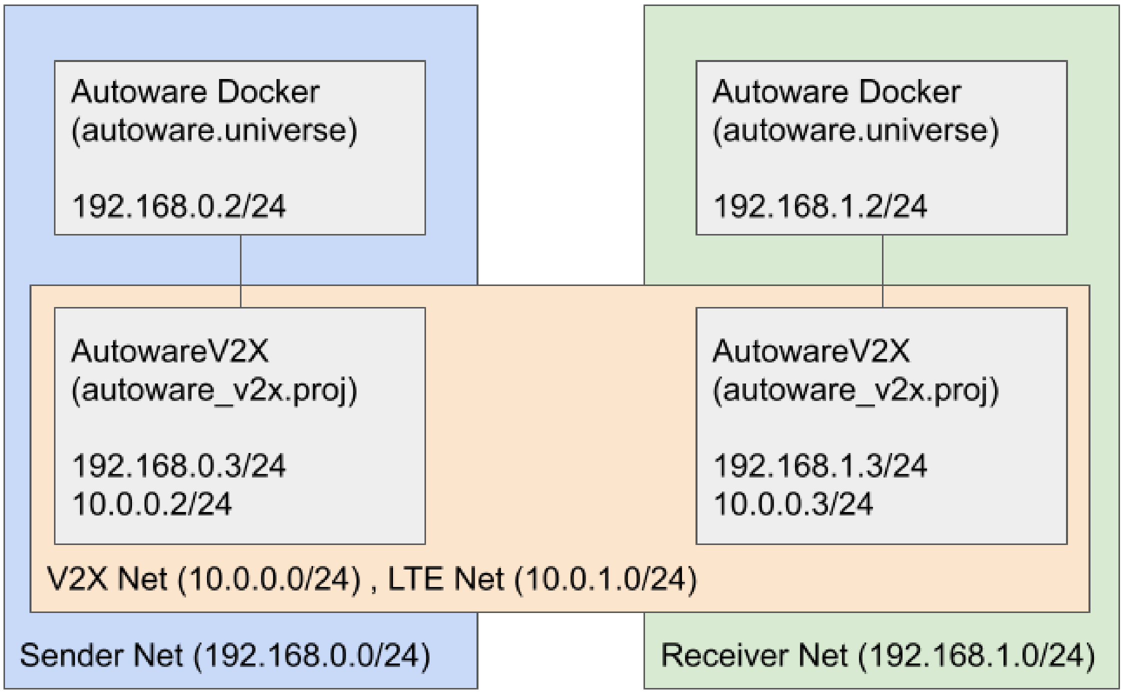

Fig. 6.1: Docker-based environment for AutowareV2X Testing

in Fig. 6.3. The *Intelligent Transport Systems* PDU is the same as Wi-Fi-based CPMs, but the whole PDU is encapsulated in a TCP/IP packet. In this way, a CPM packet can be sent over IP to a specified IP address over an IP-based 4G connection.

The use of CPAMs on the *LTE Net* is also allowing the CPM Receiver to continuously measure the PDR of the CPMs sent from the CPM Sender. In Fig. 6.4, the value of the PDR can be seen in the terminal window, with outputs being made continuously.

6.3 Evaluation Metrics

AutowareV2X was primarily evaluated through the packet delivery ratio (PDR) between the sender and receiver routers and the E2E latency T, which we define as the time taken for the CPM-based perception information to reach from a_1 to a_2. Latency T is given by

$$T = T_{r_1} + T_{r_1 r_2} + T_{r_2} \tag{6.1}$$

where T_{r_1} and T_{r_2} are the execution time at routers r_1 and r_2, respectively, and $T_{r_1 r_2}$ is the communication latency between r_1 and r_2. $T_{r_1 r_2}$ is calculated by considering half of the round-trip time of the wireless link. For latency measurement, we extended AutowareV2X so that CPM receivers would reflect CPM packets back to the sender upon reception. Fig. 6.6 shows a timing diagram of how perceived objects information flows from the original sender Autoware host A1 to the receiver Autoware host A2.

For the evaluation of the dual-channel delivery of CPMs proposed in Chapter 5, we will use the *Generation Time Difference* GTD defined in Equation 5.1 to see if the CPMs sent on the secondary cellular channel (LTE-based) CPMs are achieving a positive GTD value.

We will also investigate the *Acceptance Rate* of LTE-based CPMs, namely the *LTE-based Acceptance Rate (LAR)*. The LAR value is defined as the ratio of the number of

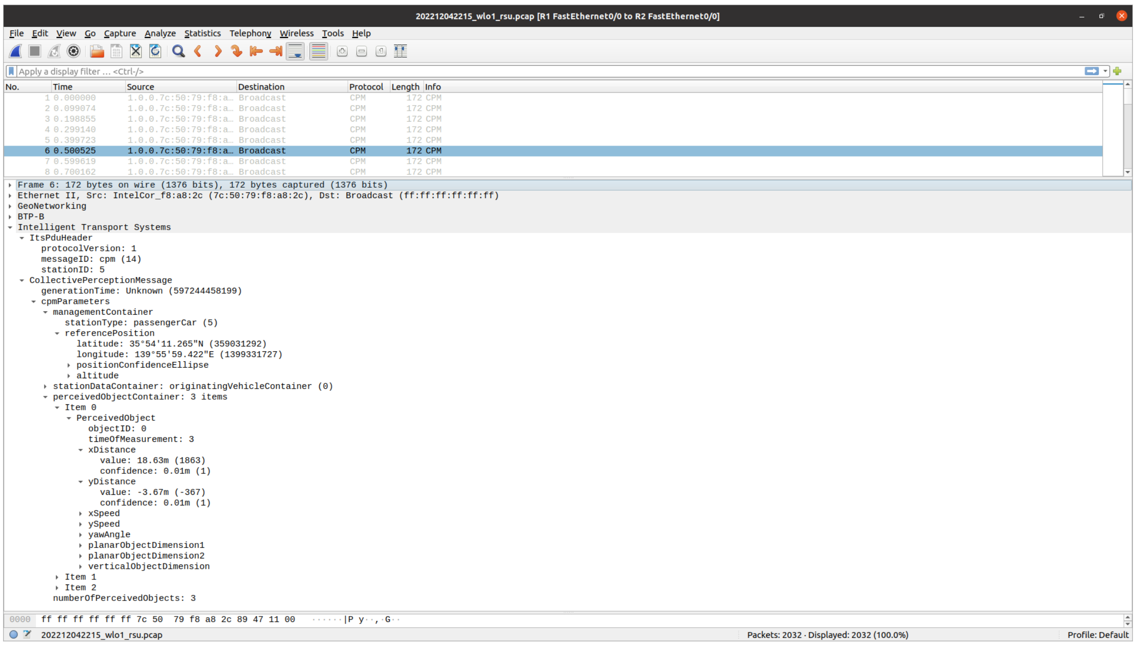

Fig. 6.2: Packet Capture of Wi-Fi-based CPMs on Wireshark

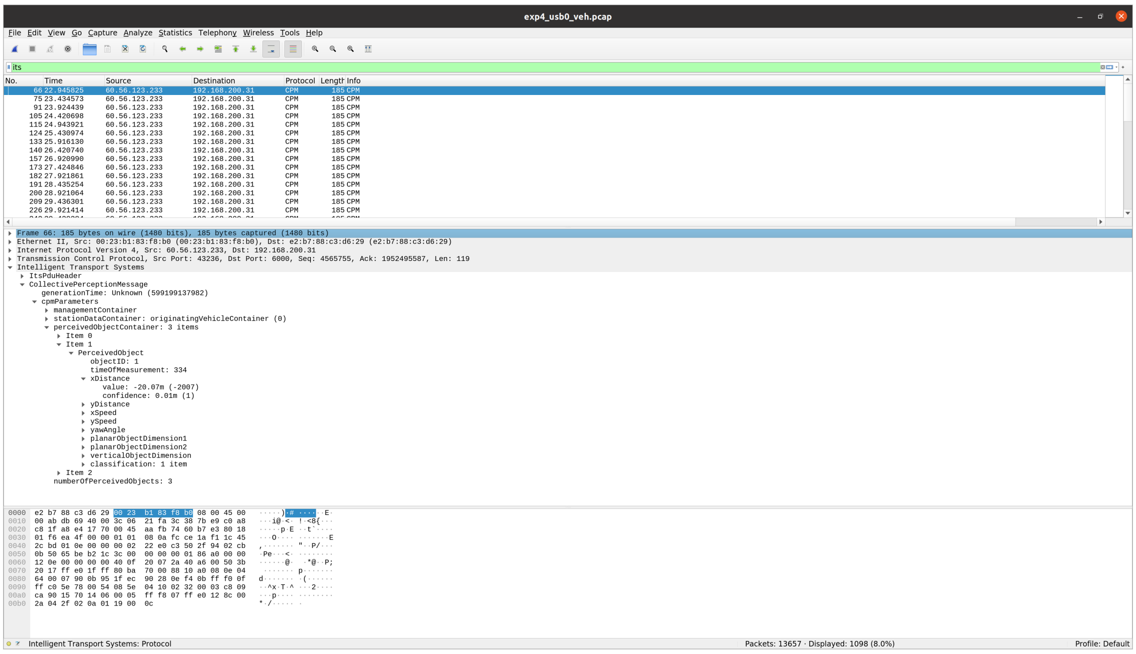

Fig. 6.3: Packet Capture of TCP/IP encapsulated LTE-based CPMs on Wireshark

LTE-based CPMs that were accepted in a specific time window to the number of all CPMs that were accepted in the same time window, including the Wi-Fi-based CPMs. A higher *LAR* value would signify that the LTE-based CPMs in the dual-channel delivery scheme are having an increased impact on the cooperative perception reliability in the receiver ITS-S.

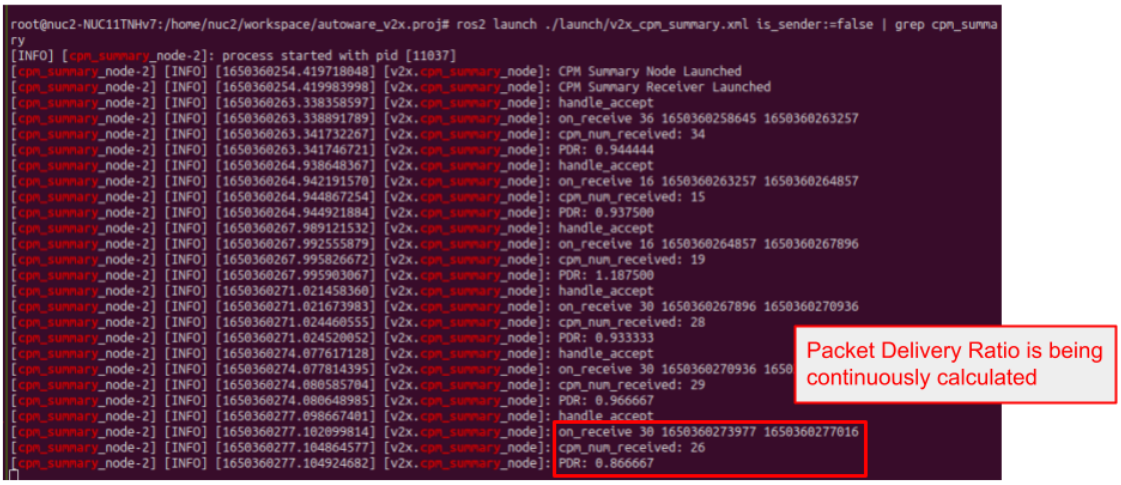

Fig. 6.4: Continuous Calculation of PDR at the CPM Reciever

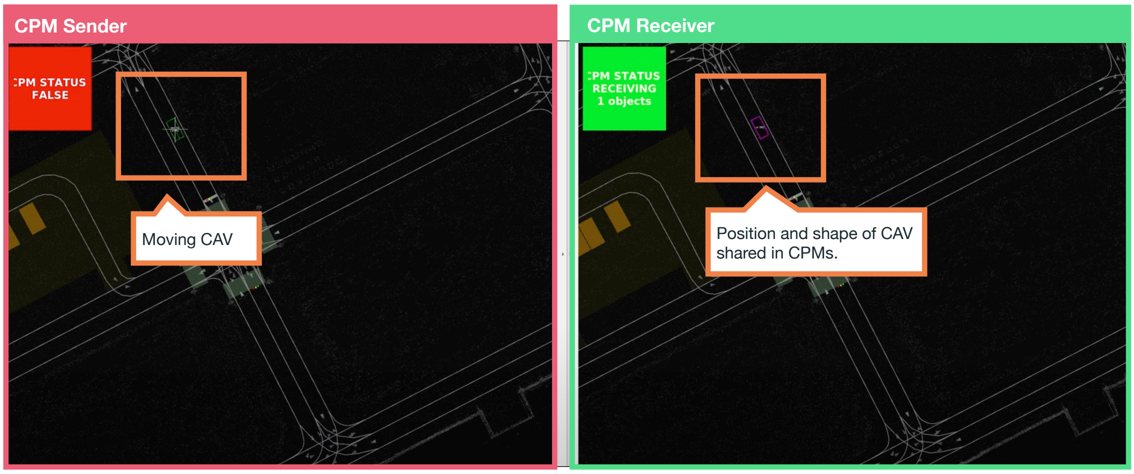

Fig. 6.5: CPM-shared Object shown in the Planning Simulator

6.4 Indoor Experiments

The performance metrics of AutowareV2X were evaluated on actual hardware in the form of indoor experiments. The setup of the indoor experiments is shown in Fig. 6.7. Autoware is run on hosts a_1 and a_2, while the V2X communication router is executed on routers r_1 and r_2. Host a_1 and router r_1 are connected by an Ethernet interface, and these two hosts comprise the AutowareV2X system. Also, host a_2 and router r_2 are set up in the same configuration.

A recording of real-life sensor information in the form of a ROS2 rosbag is fed into Autoware that is running on host a_1. The perception module of Autoware detects nearby objects, and sends the perception information to V2X communication router r_1 over the Ethernet connection. The V2X communication router then converts the information received from Autoware into the form of a CPM, before sending the CPM out of a Wi-Fi interface. Router r_2 then receives the CPM and converts the CPM data into an Autoware-compatible format before forwarding the perception information to Autoware running on host a_2. Routers r_1 and r_2 communicate with each other using the 802.11g adhoc mode,

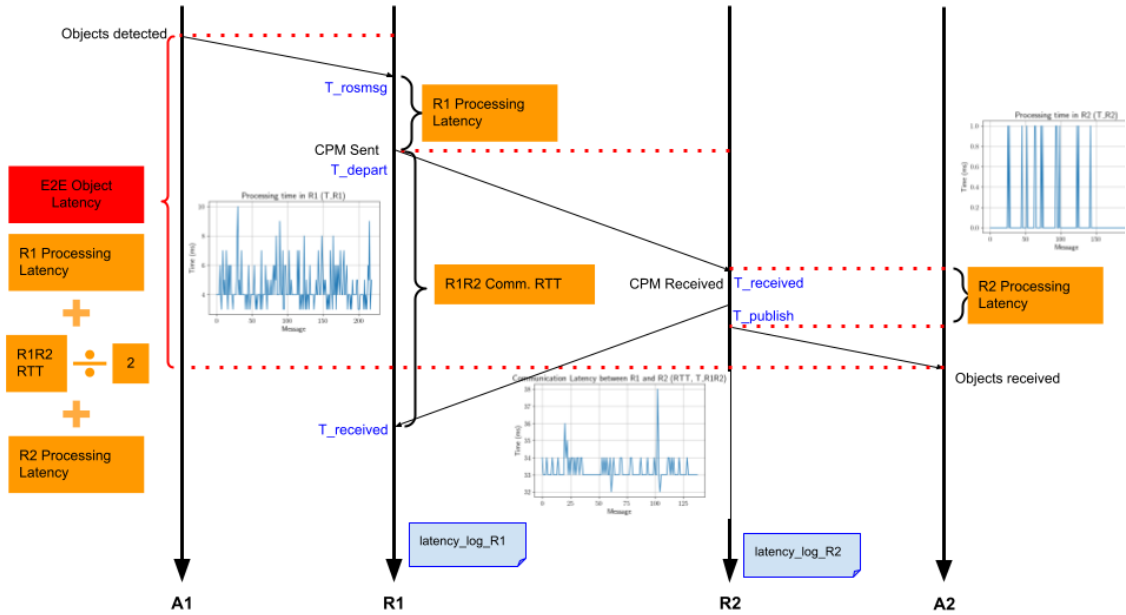

Fig. 6.6: Latency Calculation for AutowareV2X

and the distance between them, d, is set to 5 m.

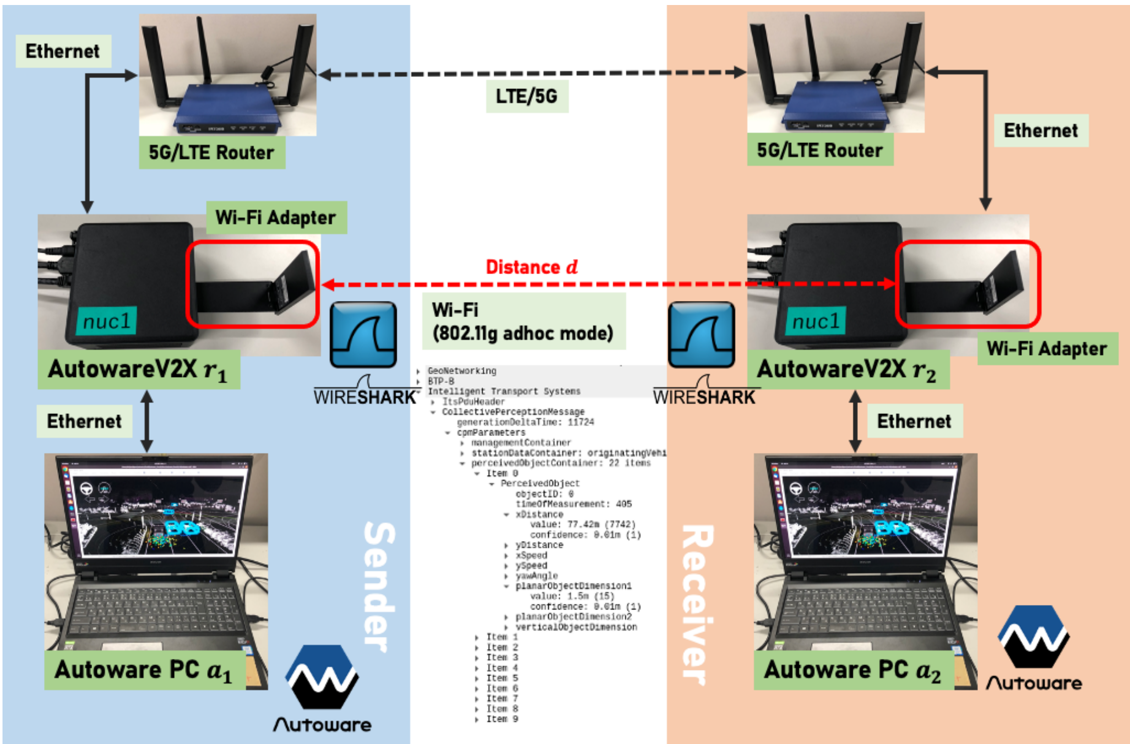

Fig. 6.7: Indoor Experiments Setup

The E2E total latency and its breakdown are shown in Fig. ??. Perception information can be delivered from the sender AD stack to the receiver AD stack in around 25 ms. The latency breakdown reveals that the communication latency between the two routers $T_{r_1 r_2}$

is about 12 ms, while the execution time at both routers accounts for less than 10 ms.

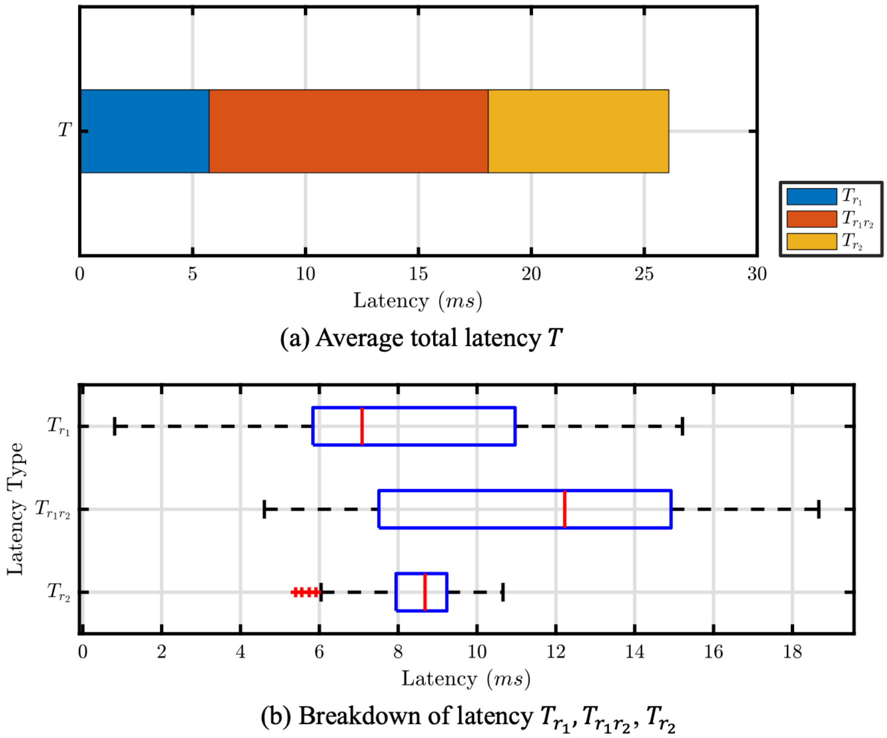

Fig. 6.8: Total latency and its breakdown

6.5 Outdoor Field Experiments

Further experiments to evaluate the performance of AutowareV2X and the effectiveness of the methods described in Chapter 5 were conducted at the Kashiwa ITS R&R Test Field in The University of Tokyo, Japan. This test field is run by the Advanced Mobility Research Center in the Institute of Industrial Science, The University of Tokyo [52], and features multiple intersections, controllable traffic lights, infrastructure-mounted LiDARs/cameras, V2X communication modules, and a railroad crossing (Fig. 6.9). A birds-eye view of the test field is shown in Fig. 6.10 with the experimental area used in Section 6.5.2 and Section 6.5.3 depicted with the yellow and red rectangles respectively.

6.5.1 Experimental Setup

The experimental setup is shown in Fig. 6.11. In our experiments, the RSU acts as the CPM sender, while the CAV acts as the CPM receiver. The hardware used for the CAV and RSU is described in Table 6.1 and Table 6.2, respectively.

Autoware is run on hosts a_1 and a_2, while AutowareV2X is executed on routers r_1 and r_2. Host a_1 and router r_1 are connected by an Ethernet interface, and these two hosts comprise the AutowareV2X system for the RSU. The sensing component of the RSU is a Velodyne VLP-16 LiDAR mounted on top of a tripod. The point cloud packets generated by the LiDAR are sent to host a_1 through an Ethernet interface. Host a_2 and router r_2 are set up in the same configuration and are set up on the CAV. The CAV is equipped with a Velodyne VLP-16 LiDAR on the top and can run the complete AD stack to realize autonomous driving. Routers r_1 and r_2 are equipped with both an 802.11g ad-hoc mode

Fig. 6.9: Overview of the Kashiwa ITS R&R Test Field

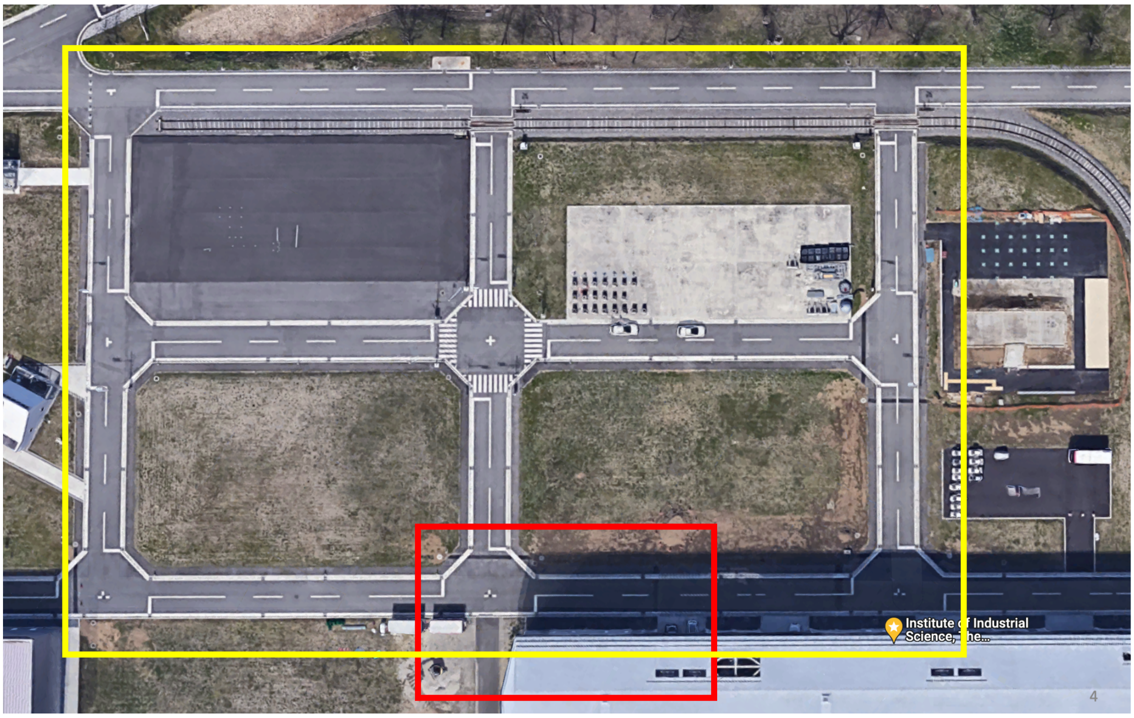

Fig. 6.10: Birds-eye view of the Kashiwa ITS R&R Test Field

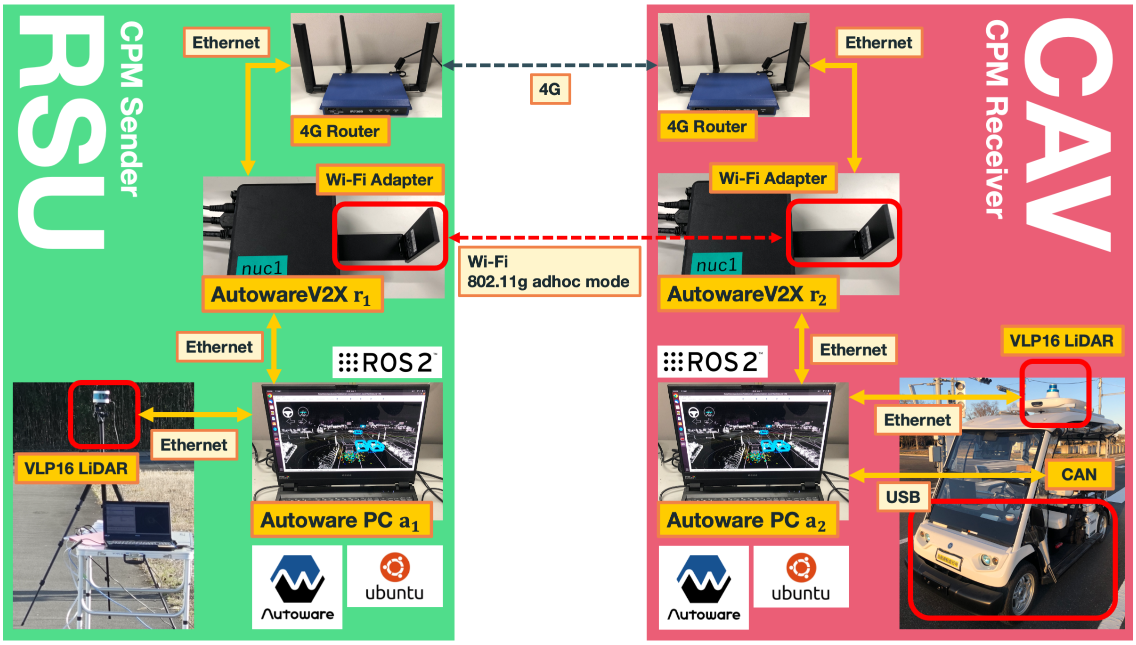

Fig. 6.11: Experimental Setup

Table 6.1: Hardware for CAV

Device	Specifications
Intel NUC Model: 11Pro	OS: Ubuntu 20.04, ROS: Galactic, Software: AutowareV2X
Netgear Model: A6210	Dual-band: 802.11b/g/n (2.4GHz), 802.11a/n/ac (5GHz)
IDY IoM 5G Gateway Model: iR730B	Bands: 3G, 4G, 5G, nano PSIM, 4x high-performance active antennas
Gigabyte PC Model: AERO-15	OS: Ubuntu 20.04, ROS: Galactic, Software: Autoware.universe
Vehicle Model: Yamaha G30Es-Li	Speed: $\leq$ 20 km/h, Sensor: VLP16 3D LiDAR (Rooftop)

Wi-Fi interface and a 4G interface. For the methods described in Chapter 5, the Wi-Fi interface is used as CH1, and the 4G interface is used as CH2.

6.5.2 Basic Scenario

For the first scenario, we considered a simple scenario where the RSU is placed in one corner of the test field, and the CAV circles around the outer perimeter (as shown in Fig. 6.12). The RSU is constantly detecting nearby objects with its onboard LiDAR and

Table 6.2: Hardware for RSU

Device	Specifications
VLP16 3D LiDAR	Range: 100 m, Accuracy: +/- 3 cm, Rotation rate: 5-20 Hz, 16 channels
Intel NUC Model: 11Pro	OS: Ubuntu 20.04, ROS: Galactic, Software: AutowareV2X
Netgear Model: A6210	Dual-band: 802.11b/g/n (2.4GHz), 802.11a/n/ac (5GHz)
IDY IoM 5G Gateway Model: iR730B	4G, 5G, nano pSIM, 4x high-performance active antennas
Gtune PC Model: H5	OS: Ubuntu 20.04, ROS: Galactic, Software: Autoware.universe

is generating CPMs with the perceived object information. The CAV is also constantly receiving CPMs sent by the RSU. The CPMs are being delivered using the dual-channel hybrid delivery system, on both the Wi-Fi and LTE channels. The placement of the LiDAR sensor for the RSU and the AutowareV2X PC is shown in Fig. 6.13. The LiDAR sensor is placed on top of a tripod to act as a temporary RSU sensor component, while the AutowareV2X PC with its network interfaces is placed in a higher location on top of the pillar.

The received signal strength indicator (RSSI) values for the CPMs sent from the RSU to the CAV are mapped on a heatmap in Fig. 6.14. It can be seen that the RSSI values are significantly higher when the CAV is closer to the RSU, with values ranging in the -50 to -60 dBm range. When the CAV goes further away from the RSU, the RSSI values drop to below -80 dBm. The PDR values for the CAV in the same scenario are shown in Fig. 6.15. It can be seen that despite the low RSSI in some areas, the PDR of the CPMs is nominal and high for most areas. However, in some places near the top-left corner and the bottom edge, the PDR has dropped to about 60%. This can be due to interference in the Wi-Fi channel degrading the network conditions significantly. Especially in the south edge of the perimeter, the PDR is leveling at around 70% until the CAV returns closer to the RSU.

Each of the latency components for the E2E latency of the Wi-Fi-based CPMs is shown in Fig. 6.16. The processing time at the RSU's AD stack T_{r1} is around 15 ms, while the wireless communication latency between the RSU and CAV T_{r1r2} is only around 9 ms. The processing time at the CAV T_{r2} is less than 1 ms because the receiver side of CPMs only needs to extract information from the CPM and publish that as a ROS2 topic to the AD stack. In total, the objects perceived by the RSU's AD stack were delivered from the RSU to the CAV's AD stack in less than 30 ms.

The *Generation Time Difference* GTD values for all the Dual-Channel CPMs received on both Channel 1 (Wi-Fi-based CPMs) and Channel 2 (LTE-based CPMs) are shown in Fig. 6.17. The change in PDR for the W-Fi-based CPMs is also depicted with the red line. Any CPM with a positive GTD was accepted by the system. When the PDR

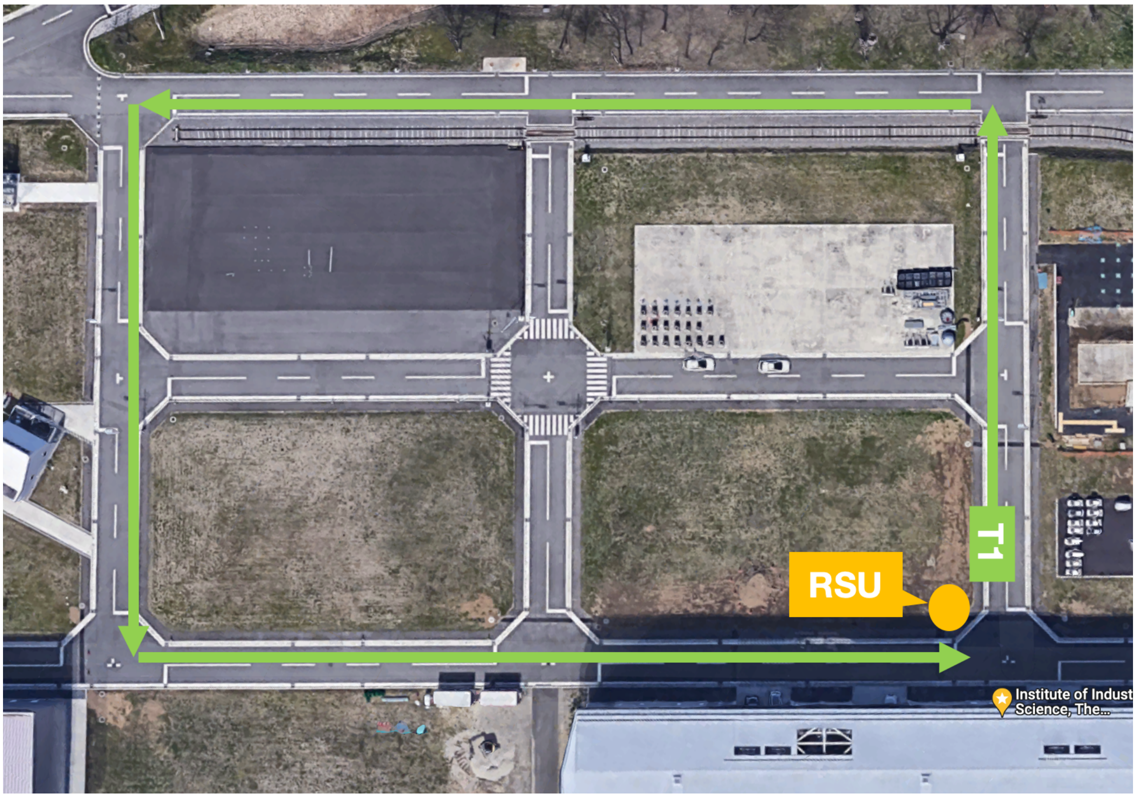

Fig. 6.12: Basic Scenario

Fig. 6.13: Experimental Setup for Basic Scenario

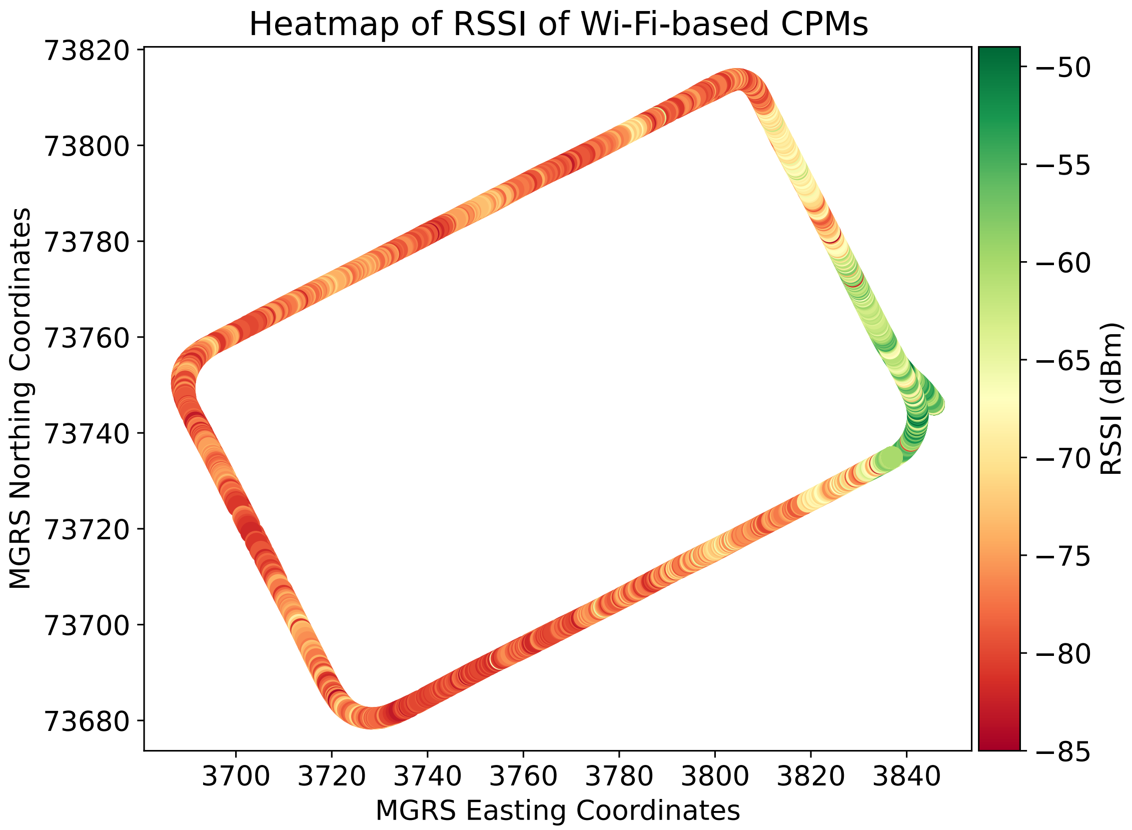

Fig. 6.14: Map of RSSI for Wi-Fi-based CPMs

for Wi-Fi-based CPMs decreases, the CPMs sent over 4G (denoted in orange) tend to show positive GTD values. This signifies that when the CPMs sent over Wi-Fi endure higher packet loss, the CPMs sent via 4G are successfully being accepted by the system as CPMs with newer information. Therefore, even in the case of high packet loss in the Wi-Fi channel, the 4G channel is successfully making up for the loss and is able to provide the receiver with new object information.

The CDF plot for the *Generation Time Difference GTD* is shown in Fig. 6.18. It can be seen that, for the most part, the CPMs sent over Wi-Fi have a positive GTD value. Meanwhile, LTE-based CPMs mostly have a GTD value that is negative or close to zero. However, the CPMs sent over the 4G channel yield higher GTD values over 200 ms. This can include the effects of packet loss on the Wi-Fi channel, and the CPMs sent over the 4G channel providing significantly newer information when many CPMs are dropped on the Wi-Fi channel consecutively.

We also analyze the Acceptance Rate for LTE-based CPMs (LAR) as defined in Section 6.3. Fig. 6.19 shows the LAR values along with the PDR for Wi-Fi-based CPMs at the same time. Although the LAR values are less than 25% and not necessarily high, when the PDR for Wi-Fi-based CPMs increase, LAR increases significantly. This signifies the fact that when the PDR for Wi-Fi-based CPMs increase and the conditions for the Wi-Fi-based CPMs are not well, the LTE-based CPMs are being able to provide newer information to the receiver ITS-S better than the Wi-Fi-based counterparts. The interval for the LTE-based CPMs' dissemination is set to 500 ms in this scenario. By changing this interval to be more frequent, we may see an increase in the LAR values.

A heatmap of the LAR values, as well as the PDR for Wi-Fi-based CPMs are plotted

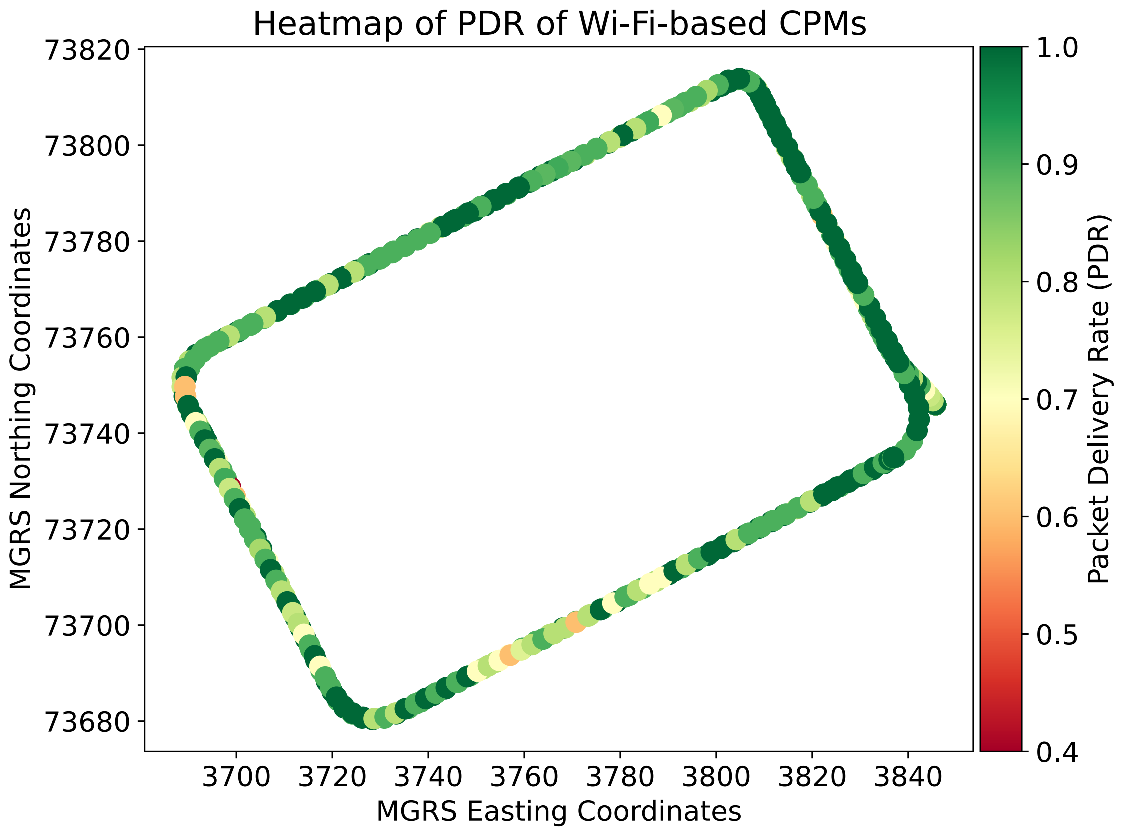

Fig. 6.15: Map of PDR for Wi-Fi-based CPMs

in Fig. 6.20 to discuss the geographic features of the results. Do note that the CAV paths were slightly different in this experiment. It can be seen from the figure that in the areas where the PDR for the Wi-Fi-based CPMs deteriorates, the LAR values are increasing. Especially in the top left corner of the test field, this effect is apparent. In areas where Wi-Fi-based CPMs are able to be utilized with high PDR, the usage of LTE-based CPMs is extremely low.

Fig. 6.21 shows the PDR for Wi-Fi-based single-channel CPM delivery and the PDR for the proposed method of dual-channel CPM delivery. It can be seen that the PDR for the proposed method increases compared to the single-channel CPM delivery, signifying that the receiver ITS-S is able to receive more of the CPMs sent from the sender ITS-S. Especially in cases where the wireless radio conditions deteriorate and the PDR decreases, the extra boost of packet delivery we can obtain from the dual-channel method can increase CPM usability and reliability.

6.5.3 Collective Perception in Blindspot Scenario

We considered a scenario where an RSU located in an intersection perceived nearby objects and broadcasted the information to an approaching CAV in the form of CPMs. While the CAV approaches the intersection from the left, it cannot locally detect the two pedestrians and vehicle approaching the intersection from the right since they are in a blind spot behind the building wall. However, the RSU can detect the objects in the blind spot and transmit their information to the CAV through CPMs. This blindspot scenario is portrayed in Fig. 6.22.

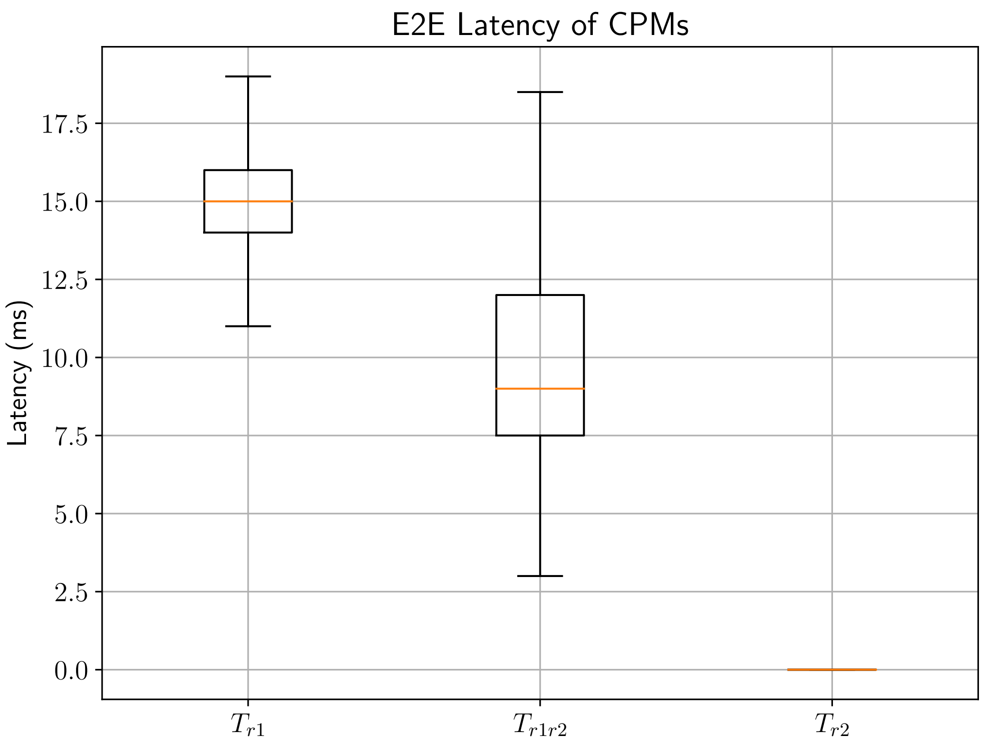

Fig. 6.16: E2E latency for Wi-Fi-based CPMs

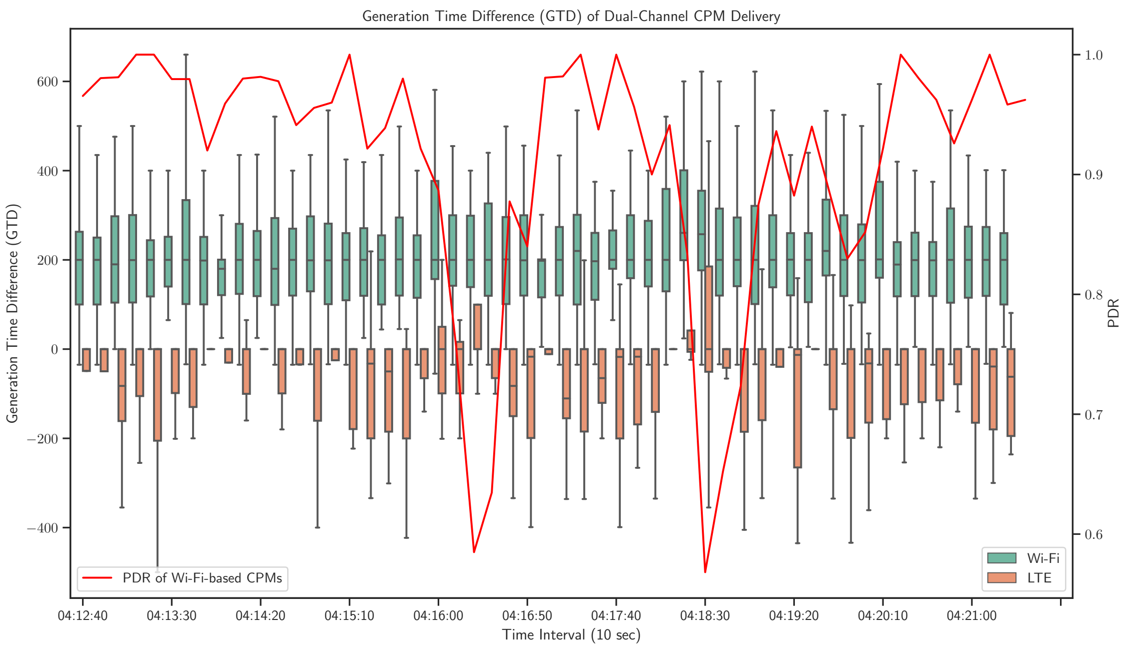

Fig. 6.17: "Generation Time Difference" GTD for the Dual-Channel CPMs

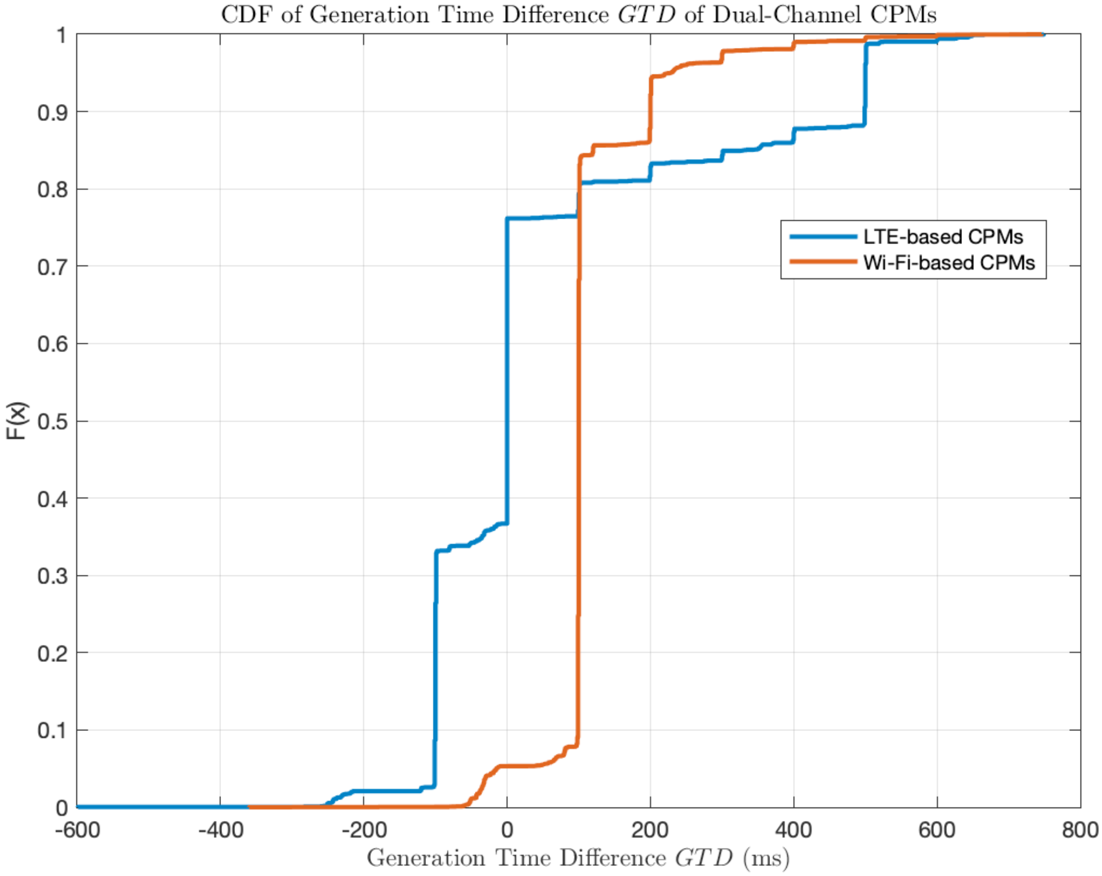

Fig. 6.18: CDF for the "Generation Time Difference" GTD of Dual-Channel CPMs

In this way, the CAV was able to perceive a wider area of its surroundings (i.e., including previously unknown places such as blind spots) through both its local onboard sensors and the object information shared by CPMs, as shown in Fig. 6.23. A video of this scenario in action is provided as well. [*1]

The packet size of the CPMs and the number of perceived objects at the RSU is shown in Fig. 6.24. As pedestrians and vehicles approach the RSU, the number of perceived objects increases, and the packet size of the CPM grows respectively as well.

The speed and position of the CAV for the blind spot scenario in the field experiment are shown in Fig. 6.25. The red line depicts the speed of the CAV when no CPMs are sent from the RSU. The CAV is unaware of the objects in the blind spot, therefore, continues to proceed through the intersection with no deceleration. The blue line depicts the speed of the CAV when CPMs are sent from the RSU, and the CAV is able to perceive the objects behind the blind spot. Through the information shared by AutowareV2X in the form of CPMs, the CAV is able to infer that the pedestrians and vehicle behind the blind spot are approaching the intersection. Therefore, it can decelerate before entering the intersection and come to a complete stop before slowly starting again once it can fully confirm the safety.

[*1] Blindspot Scenario PoC Video: `https://youtu.be/57fx3-gUNxU`

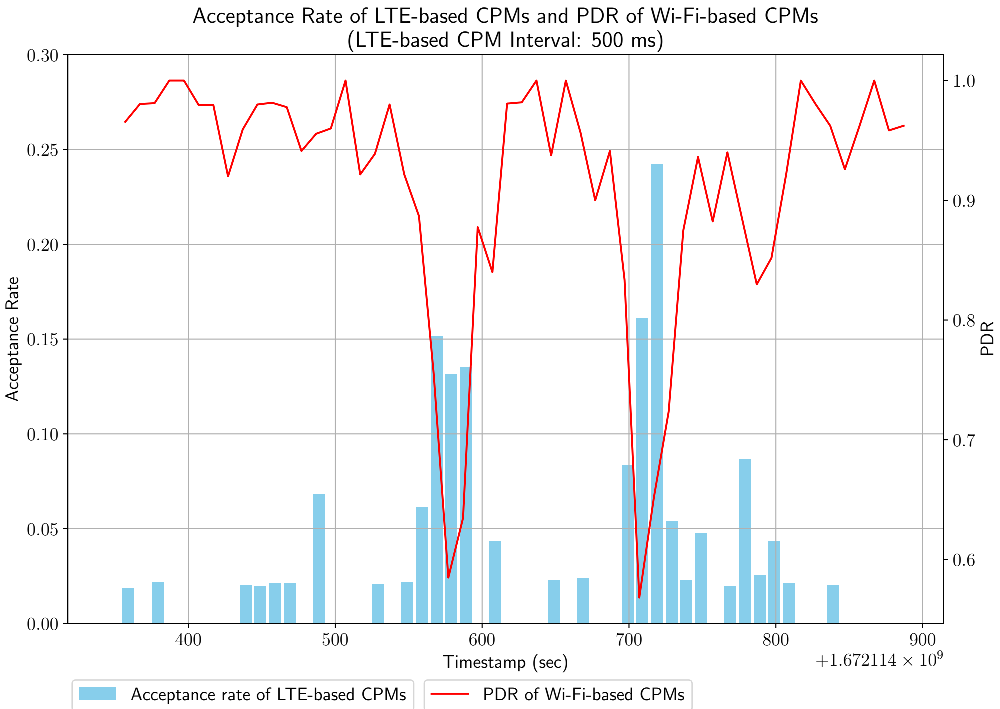

Fig. 6.19: Acceptance Rate for LTE-based CPMs (LAR)

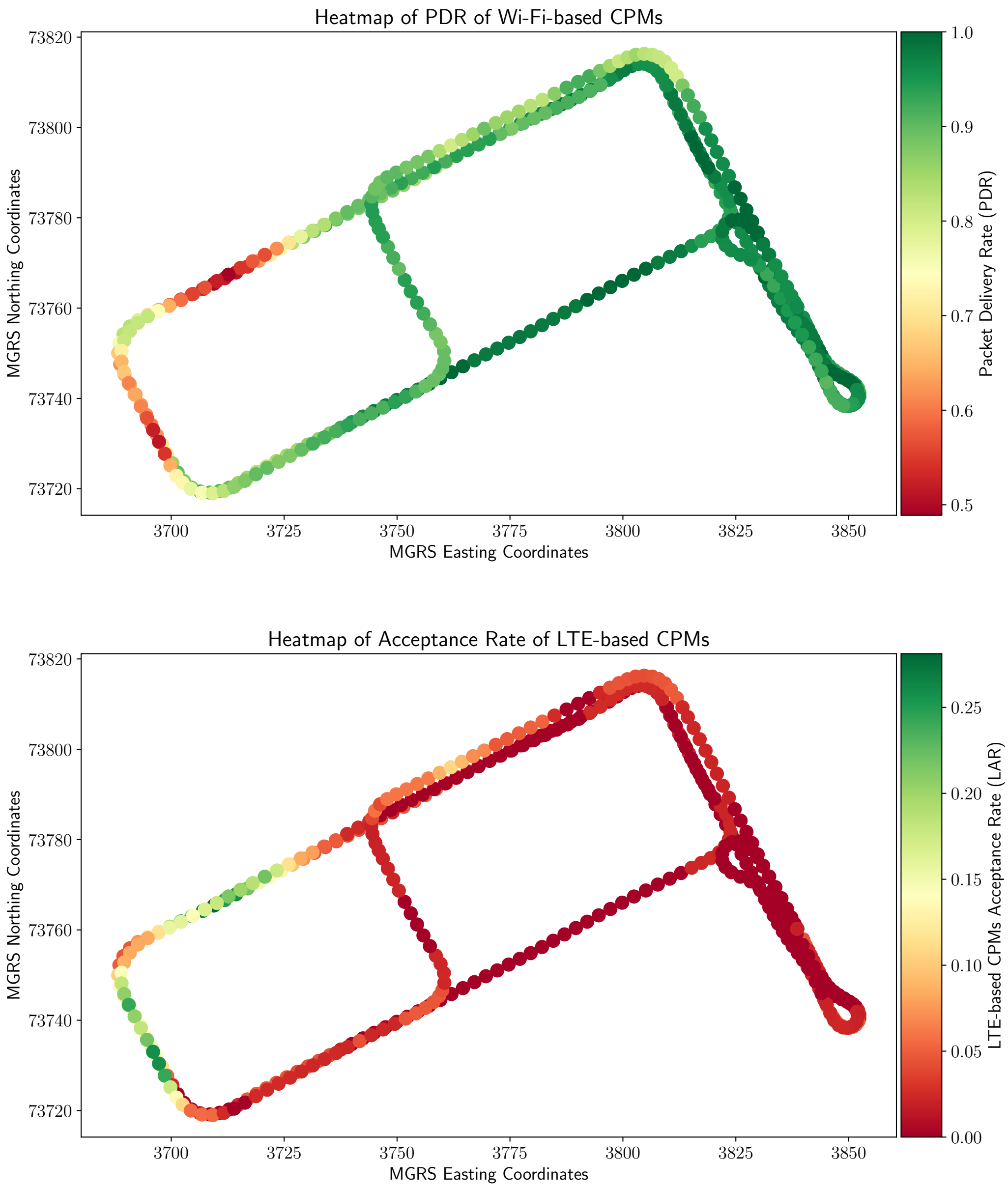

Fig. 6.20: Heatmap of PDR for Wi-Fi-based CPMs and Acceptance Rate for LTE-based CPMs (LAR)

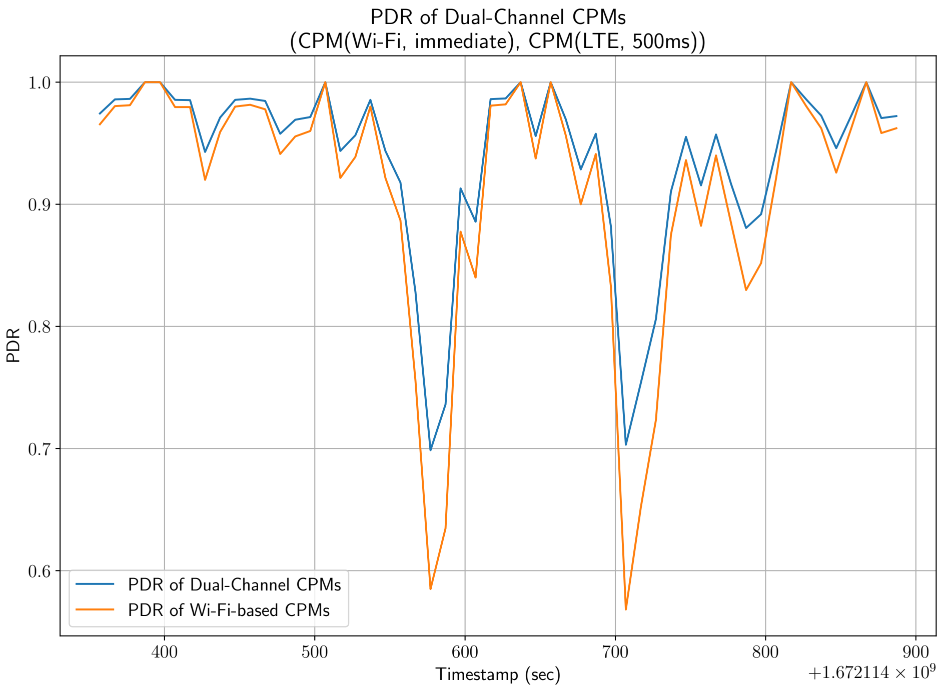

Fig. 6.21: PDR for Dual-Channel CPMs

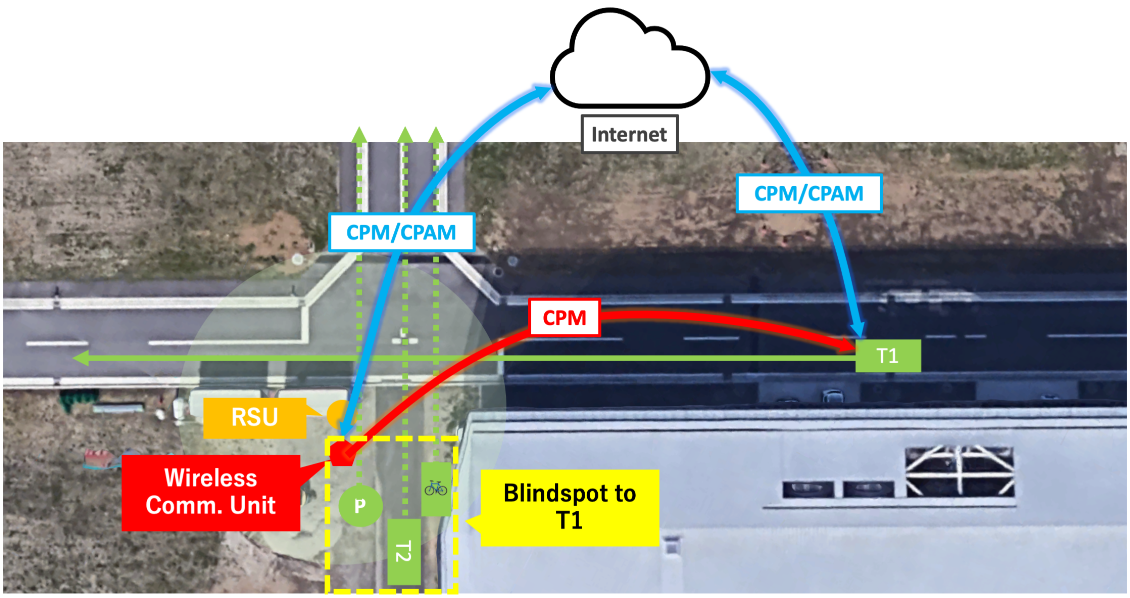

Fig. 6.22: Blindspot Scenario

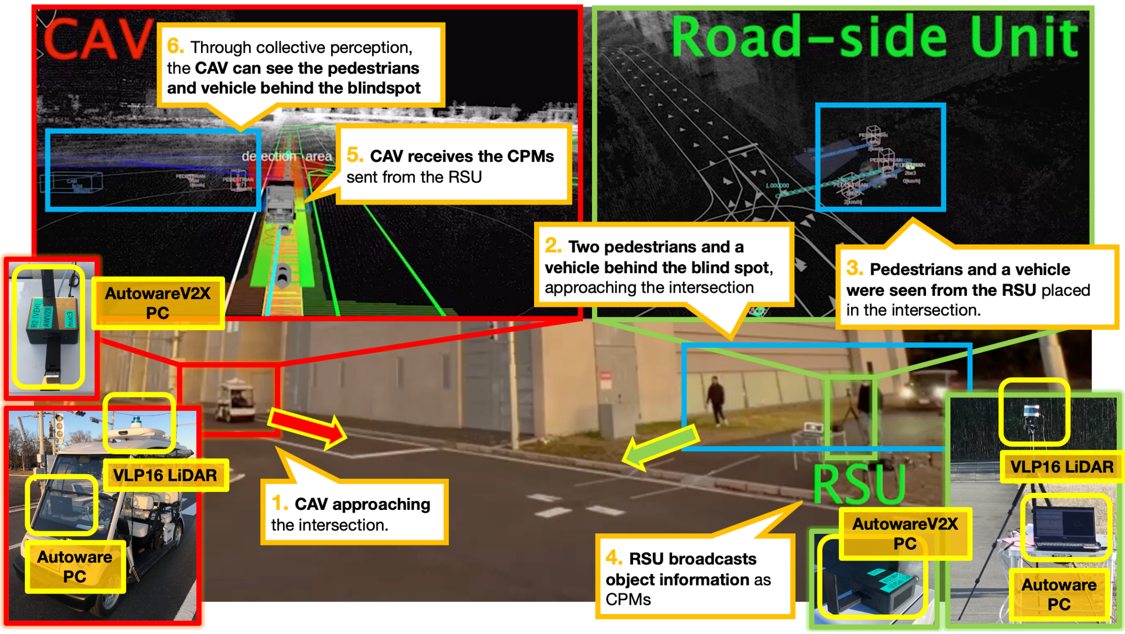

Fig. 6.23: Collective perception in action with blindspot scenario

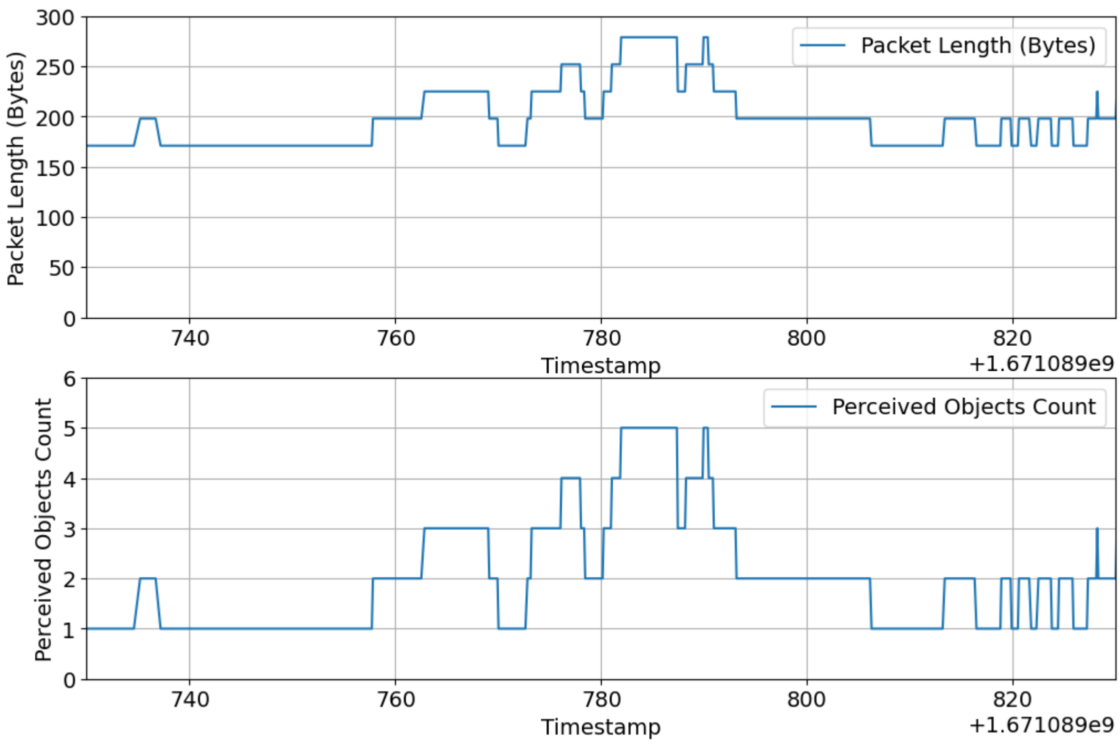

Fig. 6.24: CPM Packet Size and Perceived Object Count

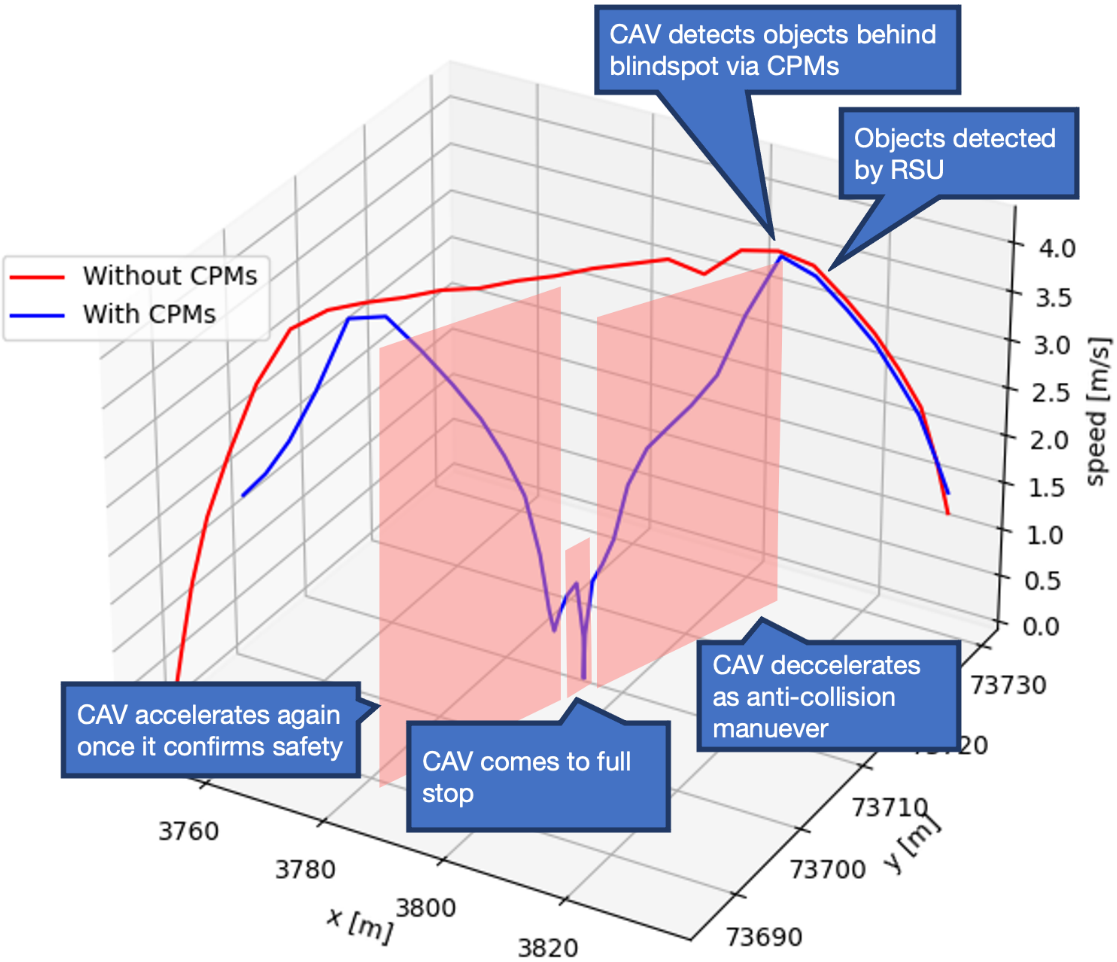

Fig. 6.25: Speed and position of CAV for Blindspot Scenario

Chapter 7

Conclusion

For cooperative intelligent transport systems (C-ITS), vehicle-to-everything (V2X) communication is utilized to allow autonomous vehicles to share critical information with each other. Collective perception enables connected autonomous vehicles (CAVs) to overcome the limitations of standalone AVs by sharing sensory information with nearby road users. The goal of this thesis is to analyze the requirements for reliable collective perception amongst multiple ITS stations and to propose methods of improving the reliability of its utilization.

We propose three methods of increasing the reliability of collective perception: (i) dual-channel hybrid delivery of sensory information, (ii) real-time packet delivery rate (PDR) monitoring, and (iii) the adaptive dual-channel delivery of sensory information using the combination of the above two methods.

In order to realize the proposed methods, we introduce AutowareV2X, an implementation of a V2X communication module that is integrated into the autonomous driving (AD) software, Autoware. AutowareV2X provides external connectivity to the entire AD stack, enabling the end-to-end (E2E) experimentation and evaluation of connected autonomous vehicles (CAV). The Collective Perception Service (CPS), which is standardized by the European Telecommunications Standard Institute (ETSI), is implemented, allowing the transmission of Collective Perception Messages (CPMs).

Functional verification in simulation-based experiments, as well as indoor experiments and outdoor field experiments, were conducted to evaluate the performance of AutowareV2X and the effectiveness of our proposed methods. Field experiments have indicated that the E2E latency of perception information is around 30 ms, and shared object data can be used by the AD software to conduct collision avoidance maneuvers. The effectiveness of the proposed methods was also confirmed, with the dual-channel delivery of CPMs enabling the CAV to dynamically select the best CPM from CPMs received from different links, depending on the freshness of their information. This enabled the reliable transmission of CPMs even where there is significant packet loss on one of the transmitting channels.

References

[1] National Highway Traffic Safety Administration, U.S. Department of Transportation. Early estimates of motor vehicle traffic fatalities and fatality rate by Sub-Categories in 2021.pdf. Technical Report DOT HS 813 298, National Highway Traffic Safety Administration, May 2022.

[2] American Society of Civil Engineers (ASCE) Foundation. ASCE 2021 INFRASTRUCTURE REPORT CARD. Technical report, ASCE, 2021.

[3] Shunsuke Aoki. *Towards Cooperative and Energy-Efficient Connected and Automated Vehicles.* PhD thesis, 09 2020.

[4] Prateek Bansal and Kara M. Kockelman. Forecasting Americans' long-term adoption of connected and autonomous vehicle technologies. *Transportation Research Part A: Policy and Practice*, 95(C):49–63, 2017.

[5] S. Taxonomy. Definitions for terms related to driving automation systems for on-road motor vehicles (j3016).

[6] Saqib Hakak, Thippa Reddy Gadekallu, Swarna Priya Ramu, M Parimala, Praveen Kumar Reddy Maddikunta, Chamitha de Alwis, and Madhusanka Liyanage. Autonomous vehicles in 5G and beyond: A survey. July 2022.

[7] J Wang, J Liu, and N Kato. Networking and Communications in Autonomous Driving: A Survey. *IEEE Communications Surveys Tutorials*, 21(2):1243–1274, 2019.

[8] R Hussain and S Zeadally. Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys Tutorials*, 21(2):1275–1313, 2019.

[9] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2VNet: Vehicle-to-Vehicle communication for joint perception and prediction. In *Computer Vision – ECCV 2020*, pages 605–621. Springer International Publishing, 2020.

[10] Michael Fürst, Oliver Wasenmüller, and Didier Stricker. Lrpd: Long range 3d pedestrian detection leveraging specific strengths of lidar and rgb, 2020.

[11] National Highway Traffic Safety Administration, U.S. Department of Transportation. U.s. dot advances deployment of connected vehicle technology to prevent hundreds of thousands of crashes. Technical Report DOT HS 813 298, National Highway Traffic Safety Administration (NHTSA), 2016. Accessed: 2023-01-10.

[12] ETSI. Itelligent transport systems (ITS); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service; EN 302 637-2 - v1.3.1, 2017.

[13] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification (DENM) Basic Servic. Technical report, ETSI-TC-ITS, September 2010.

[14] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM). Technical report, September 2014.

[15] ETSI. Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Collective Perception Service; Release 2; TS 103 324 - V0.0.52 , 2022.

[16] ETSI. ETSI EN 302 665 v1.1.1 Intelligent Transport Systems (ITS); Communications Architecture. Technical report, September 2010.

[17] Sven Laux, Gurjashan Singh Pannu, Stefan Schneider, Jan Tiemann, Florian Klingler, Christoph Sommer, and Falko Dressler. Demo: OpenC2X — an open source experimental and prototyping platform supporting ETSI ITS-G5. In *2016 IEEE Vehicular Networking Conference (VNC)*, pages 1–2, December 2016.

[18] Raphael Riebl, Christina Obermaier, Stefan Neumeier, and Christian Facchi. Vanetza: Boosting research on inter-vehicle communication. *Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)*, pages 37–40, 2017.

[19] U.S. Department of Transportation Federal Highway Administration. CARMA platform detailed design documentation. `https://usdot-carma.atlassian.net/wiki/spaces/CRMPLT/pages/196182019/CARMA+Platform+Detailed+Design+Documentation`, February 2021.

[20] Yingyan Lou, Amir Ghiasi, Mafruhatul Jannat, Sujith Racha, David Hale, Wade Goforth, and Pavle Bujanovic. Cooperative automation research: CARMA proof-of-concept TSMO use case testing: CARMA cooperative perception concept of operations. Technical Report FHWA-HRT-22-062, April 2022.

[21] Yingyan Lou, David Hale, Michael McConnel, Amir Ghiasi, Mafruhatul Jannat, Sujith Racha, and Leidos, Inc. FHWA cooperative automation research: CARMA proof-of-concept TSMO use case testing: CARMA cooperative perception low-level concept of operations. Technical Report FHWA-HRT-22-116, November 2022.

[22] Goodarz Mehr, Prasenjit Ghorai, Ce Zhang, Anshul Nayak, Darshit Patel, Shathushan Sivashangaran, and Azim Eskandarian. X-CAR: An experimental vehicle platform for connected autonomy research powered by CARMA. April 2022.

[23] OpenC2X - distributed embedded systems. `https://www.ccs-labs.org/software/openc2x/`. Accessed: 2022-12-20.

[24] alexvoronov. Etsi its g5 geonetworking stack.

[25] Manabu Tsukada, Takaharu Oi, Akihide Ito, Mai Hirata, and Hiroshi Esaki. AutoC2X: Open-source software to realize V2X cooperative perception among autonomous vehicles. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–6, November 2020.

[26] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296, April 2018.

[27] Manabu Tsukada, Takaharu Oi, Masahiro Kitazawa, and Hiroshi Esaki. Networked roadside perception units for autonomous driving. *Sensors*, 20(18), September 2020.

[28] Rusheng Zhang, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Design, implementation, and evaluation of a roadside cooperative perception system. *Transp. Res. Rec.*, page 03611981221092402, June 2022.

[29] F A Schiegg, I Llatser, H Tchouankem, T Frye, F Wildschutte, and F Hofmann. Removing blind spots – infrastructure-assisted collective perception. In *ELIV 2021*, pages 29–44. VDI Verlag, October 2021.

[30] Masaya Mizutani, Manabu Tsukada, and Hiroshi Esaki. AutoMCM: Maneuver coordination service with abstracted functions for autonomous driving. In *IEEE International Conference on Intelligent Transportation (ITSC) 2021*. unknown, September 2021.

[31] Chanyoung Jung, Daegyu Lee, Seungwook Lee, and David Hyunchul Shim. V2X-Communication-Aided autonomous driving: System design and experimental validation. *Sensors*, 20(10), May 2020.

[32] John B. Kenney. Dedicated short-range communications (dsrc) standards in the

united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[33] Fabio Arena, Giovanni Pau, and Alessandro Severino. A review on ieee 802.11p for intelligent transportation systems. *Journal of Sensor and Actuator Networks*, 9(2):22, Apr 2020.

[34] Khadige Abboud, Hassan Aboubakr Omar, and Weihua Zhuang. Interworking of DSRC and cellular network technologies for V2X communications: A survey. *IEEE Trans. Veh. Technol.*, 65(12):9457–9470, December 2016.

[35] Xiaoman Shen, Jun Li, Lei Chen, Jiajia Chen, and Sailing He. Heterogeneous LTE/DSRC approach to support real-time vehicular communications. In *2018 10th International Conference on Advanced Infocomm Technology (ICAIT)*, pages 122–127, August 2018.

[36] Zeeshan Hameed Mir, Jamal Toutouh, Fethi Filali, and Young-Bae Ko. Enabling DSRC and C-V2X integrated hybrid vehicular networks: Architecture and protocol. *IEEE Access*, 8:180909–180927, 2020.

[37] Shunli Ren, Siheng Chen, and Wenjun Zhang. Collaborative perception for autonomous driving: Current status and future trend. In *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, Lecture notes in electrical engineering, pages 682–692. Springer Nature Singapore, Singapore, 2023.

[38] ETSI. Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2; TR 103 562 - V2.1.1 , 2019.

[39] G Thandavarayan, M Sepulcre, and J Gozalvez. Analysis of Message Generation Rules for Collective Perception in Connected and Automated Driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 134–139, June 2019.

[40] Quentin Delooz, Alexander Willecke, Keno Garlichs, Andreas-Christian Hagau, Lars Wolf, Alexey Vinel, and Andreas Festag. Analysis and evaluation of information redundancy mitigation for V2X collective perception. *IEEE Access*, pages 1–1, 2022.

[41] Hidetaka Masuda, Oussama El Marai, Manabu Tsukada, Tarik Taleb, and Hiroshi Esaki. Feature-based vehicle identification framework for optimization of collective perception messages in vehicular networks, 2022.

[42] Michael Klöppel-Gersdorf and Thomas Otto. Evaluating message size of the collective perception message in real live settings. In *VEHITS*, pages 554–561, 2021.

[43] F A Schiegg, D Bischoff, J R Krost, and I Llatser. Analytical performance evaluation of the collective perception service in IEEE 802.11p networks. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, May 2020.

[44] X Series. Data networks and open system communications. *OSI networking and system aspects—Abstract Syntax Notation One (ASN. 1), International Telecommunication Union, ITU-T Recommendation X.*

[45] The autoware foundation. `https://www.autoware.org/`. Accessed: 2023-01-10.

[46] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Sci Robot*, 7(66):eabm6074, May 2022.

[47] R Riebl, H Günther, C Facchi, and L Wolf. Artery: Extending veins for VANET applications. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 450–456, June 2015.

[48] Baidu apollo. `https://developer.apollo.auto/index.html`. Accessed: 2023-1-1.

[49] Github: autowarefoundation/autoware. `https://github.com/autowarefoundation/autoware`. Accessed: 2023-01-10.

[50] Adrian Abunei, Ciprian-Romeo Comşa, and Ion Bogdan. Implementation of ETSI ITS-G5 based inter-vehicle communication embedded system. In *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4, July 2017.

[51] Autoware documentation - planning simulation. https://autowarefoundation.github.io/autoware-documentation/main/tutorials/ad-hoc-simulation/planning-simulation/. Accessed: 2023-01-10.

[52] The University of Tokyo Advanced Mobility Research Center (ITS Center), Institute of Industrial Science. Its r&r exp. field - advanced mobility research center (its center), institute of industrial science, the university of tokyo.

Acknowledgements

Throughout my master's degree program, I have received tremendous support and assistance from many different people. I would first like to express my deepest gratitude to my supervisors, Professor Hiroshi Esaki and Professor Manabu Tsukada, who have been continuously supporting me since my undergraduate years and always providing me with invaluable expertise and feedback on research methodology, experiments, and academic writing. I will also like to thank Professor Ehsan Javanmardi, Professor Jin Nakazato, and Dr. Asad Muhammad for their immense support in all aspects of my academic experience. I am very grateful for all my professors and supervisors who have constantly provided me with insightful suggestions and guidance throughout our many discussions. I was also privileged to join various programs both internationally and domestically, and the professors constantly pushed me to do better and take on new opportunities.

I will also like to thank my incredible colleagues from the C-ITS team: Koichi Kambara, Vishal Chauhan, Yosodipuro Nicholaus Danispadmanaba, Pengfei Lin, Tao Ye, Yuze Jiang, Kazuto Matsumoto, Yiwei Cheng, Zekun Wang, Hu Dou, and others for their constant feedback on my research in the C-ITS Meeting as well as refreshing fun times spent outdoors.

I would also like to acknowledge my supervisor from my working experience at TIER IV, INC., Yuki Iida, who has generously and constantly given his time to provide me with extensive implementation support and engaging learning experiences.

I would also like to thank the laboratory secretaries, Fumi Takahashi and Aiko Iwai, for their dedicated continual assistance with all sorts of administrative tasks. Especially when participating in international conferences or conducting week-long experiments in domestic test fields, they have supported me in creating the correct documents and making sure everything went smoothly so I can focus on my research.

My experiences with the team gave me a holistic view of future mobility possibilities, and lead me to take interest in this field not only academically but from an industry perspective as well. My time spent with the amazing people here was not just an academic experience, but quite literally a life-changing adventure.

I would also like to thank my parents, Hiromi Asabe and Shinichi Asabe, as well as my sister, Mao Asabe, for their undying support in every aspect of my life. They were always there for me and supported my every decision. Finally, a big thank you to my fiance Liz Bauer, for all her love and support even in the hardest of times.

Research Activities

[1] Organizing chair in the program committee for the WIDE Meeting December 2021. December 2021.
[2] Shownet Team Member for Interop Tokyo 2022. June 2022.
[3] Third place in the Japan Automotive AI Challenge 2022. June 2022.
[4] Visiting student at the Cooperative Interactive Vehicles Summer School 2022 held in California, USA. August 2022.

Publications

[1] Yu Asabe, Ehsan Javanmardi, Jin Nakazato, Manabu Tsukada, Hiroshi Esaki, "AutowareV2X: Enabling V2X Communication and Collective Perception for Autonomous Driving", Asian Internet Engineering Conference (AINTEC) 2022 Poster, December 2022, (**Best Poster Award**).

[2] Yu Asabe, Ehsan Javanmardi, Jin Nakazato, Manabu Tsukada, Hiroshi Esaki, "AutowareV2X: Enabling V2X Communication and Collective Perception for Autonomous Driving", IECIE Technical Committee on Intelligent Transport Systems Technology (ITS), Feburary 2023.

AutowareV2X: Enabling V2X Communication and Collective Perception for Autonomous Driving

Tsukada Lab. Computer Network Group
東京大学 THE UNIVERSITY OF TOKYO

Yu Asabe, Ehsan Javanmardi, Jin Nakazato, Manabu Tsukada, Hiroshi Esaki
The University of Tokyo, Graduate School of Information Science and Technology

I. Introduction

- For cooperative intelligent transport systems (C-ITS), autonomous vehicles **utilize vehicle-to-everything (V2X) communication** to share critical information.
- **Overcomes the limitations** of stand-alone autonomous driving (AD) systems.
- Connected autonomous vehicles and infrastructure enable **cooperatively enhanced environmental awareness and decision-making**.

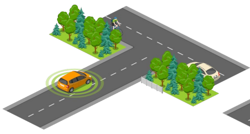

Fig. 1 Standalone AD

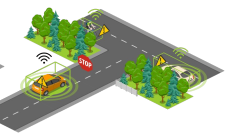

Fig. 2 Connected and Cooperative ITS

II. Problem

- Many studies focus on the **effect of V2X communication on a specific module in the autonomous driving (AD) stack**, such as the perception [1] or planning [2] pipeline, **but not the entire stack**.
- A more comprehensive **AD-integrated V2X router is necessary** to enable the experimentation and evaluation of the **effects of V2X communication on the end-to-end (E2E) AD software** as a whole.

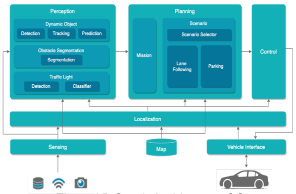

Fig. 3 AD Stack Architecture [3]

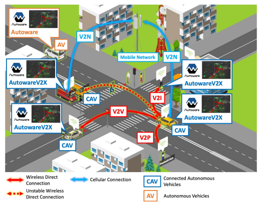

Fig. 4 V2X Communication

III. Proposed System – AutowareV2X

- **AutowareV2X** is a V2X communication module that can be integrated into the widely used open-source AD software stack, **Autoware** [3], to realize connected autonomous vehicles.

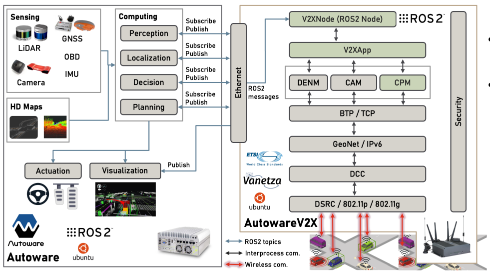

Fig. 5 System Design and Architecture

- **V2X Node** is a Robot Operating System (ROS2) node that functions as the interface between the V2X communication stack and the ROS2-based Autoware.
- **V2X Application** is responsible for common tasks necessary for V2X communication, coordinate transformations, cross-layer network configuration, and the management of various ITS facilities.
- AutowareV2X is able to **generate Collective Perception Messages (CPMs)** [4] with the objects detected in Autoware.
- Autoware and AutowareV2X **can be placed on separate hardware** to accommodate more use cases.

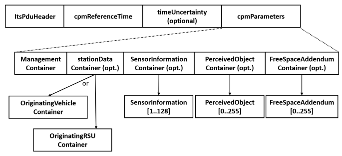

Fig. 6 Collective Perception Messages (CPM) Structure [4]

IV. Evaluation

- Performance metrics were evaluated on **actual hardware in the form of indoor experiments.**
- A **recording of real-life sensor information** in the form of a ROS2 rosbag is fed into Autoware that runs on host $a1$.
- CPMs are sent from $r1$ to $r2$ over **a Wi-Fi (802.11g) adhoc mode network**.

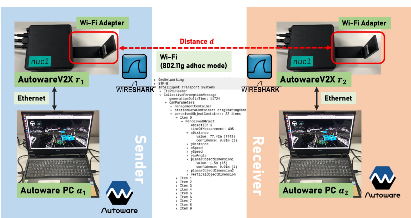

Fig. 7 Indoor Experiments Setup

V. Results

- **Generate Collective Perception Messages (CPMs)** based on the objects detected in Autoware.
- Total E2E latency T : **25 ms**
- Wireless communication latency T_{r1r2} : **12 ms**
- Processing time T_{r1} in $r1$: **7 ms**
- Processing time T_{r2} in $r2$: **9 ms**

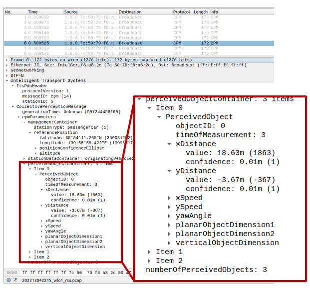

Fig. 8 Wireshark of CPM

$$T = T_{r1} + T_{r1r2} + T_{r2}$$

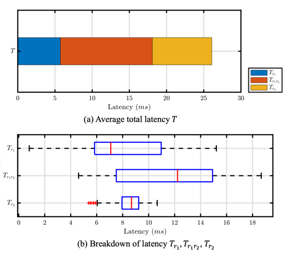

Fig. 9 Latency analysis

[1] H. Masuda et al. 2022. Feature-based Vehicle Identification Framework for Optimization of Collective Perception Messages in Vehicular Networks. , 11 pages.
[2] M. Mizutani et al. 2021. AutoMCM: Maneuver Coordination Service with Abstracted Functions for Autonomous Driving. In IEEE International Conference on Intelligent Transportation (ITSC) 2021.
[3] S. Kato et al. 2018. Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems. In 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS). 287–296.
[4] ETSI. 2019. ITS; Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2; TR 103 562 - V2.1.1 .

Acknowledgment
This research was supported in part by JSPS KAKENHI under Grant JP19K0281 and Grant JP21H03423; and in part by the National Institute of Information and Communications Technology (NICT) in Japan under Grant #01101 for Beyond 5G.

Fig. 1: Poster for AINTEC 2022