Towards C-V2X Enabled Collaborative Autonomous Driving

Yuankai He, Baofu Wu, Zheng Dong, *Member, IEEE*, Jian Wan, and Weisong Shi, *Fellow, IEEE*

***Abstract*—Intelligent vehicles, including autonomous vehicles and vehicles equipped with ADAS systems, are single-agent systems that navigate solely on the information collected by themselves. However, despite rapid advancements in hardware and algorithms, many accidents still occur due to the limited sensing coverage from a single-agent perception angle. These tragedies raise a critical question of whether single-agent autonomous driving is safe. Preliminary investigations on this safety issue led us to create a C-V2X-enabled collaborative autonomous driving framework (CCAD) to observe the driving circumstance from multiple perception angles. Our framework uses C-V2X technology to connect infrastructure with vehicles and vehicles with vehicles to transmit safety-critical information and to add safety redundancies. By enabling these communication channels, we connect previously independent single-agent vehicles and existing infrastructure. This paper presents a prototype of our CCAD framework with RSU and OBU as communication devices and an edge-computing device for data processing. We also present a case study of successfully implementing an infrastructure-based collaborative lane-keeping with the CCAD framework. Our case study evaluations demonstrate that the CCAD framework can transmit, in real-time, personalized lane-keeping guidance information when the vehicle cannot find the lanes. The evaluations also indicate that the CCAD framework can drastically improve the safety of single-agent intelligent vehicles and open the doors to many more collaborative autonomous driving applications.**

***Index Terms*—ADAS, Autonomous Driving, C-V2X, Collaborative Driving, Cooperative Driving, Edge Computing**

I. Introduction

Safety is the most critical feature of any vehicle. Intelligent vehicles, including autonomous vehicles and vehicles equipped with ADAS systems, are single-agent systems that safely navigate based solely on the information collected by themselves. These systems support safety-critical functions using cameras for computer vision detection and radars and LiDARs for localization and collision avoidance. To increase vehicle safety and the reliability of safety-critical functions, manufacturers have invented new hardware and designed new algorithms.

However, despite rapid advancements in hardware and algorithms, many accidents with intelligent vehicles still occur. According to the latest report by NHTSA, it has received direct crash data from manufacturers from July 2021 to May 2022, which includes 392 reports related to Level 2 ADAS and 130 reports associated with Level 3 to Level 5 Automated Driving Systems [3], [4]. From these reports, we can see that ADAS and autonomous driving are far from perfect. In fact, from the report for Level 2 ADAS systems, out of the 392 reported crashes, 78 crashes are with stationary objects, and 20 crashes are with poles or trees [3]. These crashes are often a result of miss classified objects. For example, in one of Tesla's crashes, autopilot, Tesla's autonomous driving system, mistook the side of a truck as a cloud in the sky. Dong *et al.* used two fatal Tesla accidents to demonstrate the missing capacities in current autonomous driving systems [11]. It is not difficult to see that existing solutions cannot guarantee driver safety because of the limitations of single-agent systems.

Lane keeping is the most typical autonomous driving application. Current solutions use the front camera to detect the road lines and control the vehicle. Unfortunately, existing solutions are prone to misjudgment if the lane lines become blurred or missing due to wear-and-tear or snow, as shown in Fig. 1. In the four scenarios presented, current standalone autonomous vehicle systems cannot find a safe route without the intervention of the human driver.

The key to addressing this single-agent issue is obtaining information from others. Vehicle-to-Everything (V2X) communication offers a collaborative approach where multiple agents perceive the scene cooperatively [11]. Hobert *et al.* [7] is one of the first papers to discuss the possibility of using V2X communication for cooperative autonomous driving. Hobert argues in this paper that the current design of autonomous vehicles lacks coordination among vehicles and suffers from the sensors' limited range. Hobert *et al.* also argue that V2X can overcome these drawbacks and enable cooperative maneuvering and sensing [7]. In [9]–[18], the authors proposed different ways to improve vehicle safety with V2X communications. All authors argue that V2X can significantly increase the safety of standalone autonomous vehicles. However, they did not have the opportunity to experiment with actual V2X communication devices designed for vehicles.

Our preliminary investigations, based on new trends in C-V2X and collaborative autonomous driving, led us to create a

Manuscript received XXXX XXXX, XXXXX; revised XXX, XXXX. This work was supported in XXXX. Corresponding author: Weisong Shi (email: weisong@udel.edu).

Yuankai He and Zheng Dong are with Department of Computer Science, Wayne State University, Detroit, MI 48202, USA (e-mail:william.he@wayne.edu; dong@wayne.edu)

Baofu Wu is with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China and Department of Computer Science, Wayne State University, Detroit, MI 48202, USA (e-mail:baofu.wu@hdu.edu.cn)

Jian Wan is with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China and School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China (email: wanjian@hdu.edu.cn)

Weisong Shi is with the Department of Computer Science, University of Delaware, Newark, DE, 19716, USA (e-mail: weisong@udel.edu)

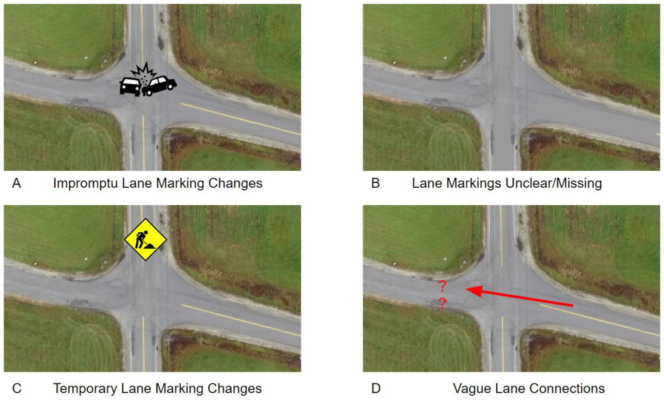


Fig. 1. Typical scenarios where current lane-keeping algorithms no longer work.

C-V2X-enabled collaborative autonomous driving framework (CCAD) to observe the driving circumstance from multiple angles. Our framework uses C-V2X technology to connect infrastructure with vehicles and vehicles with vehicles to transmit safety-critical information and to add safety redundancies. By enabling these communication channels, we connect previously isolated single-agent vehicles with each other and with existing infrastructure.

To evaluate our framework, we built a test environment inside our lab to simulate a lane-keeping scenario. We deployed cameras and off-the-shelves C-V2X devices on the roadside to analyze whether the vehicle was off-center and deployed C-V2X on vehicles to measure latency. In this lane-keeping scenario, we specifically evaluate the effectiveness of the lane-keeping application from C-V2X-enabled infrastructure.

To the best of our knowledge, our main contributions are as follows:

- We are the first to present a prototype framework for C-V2X-enabled collaborative autonomous driving, which can drastically improve the safety of single-agent intelligent vehicles and open the doors to many more collaborative autonomous driving applications.
- We implement infrastructure-based collaborative lane-keeping with the CCAD framework and demonstrate that the framework can transmit, in real-time, personalized lane-keeping guidance information when the vehicle cannot find the lanes.
- We provide a manual on how to build a C-V2X-enabled collaborative autonomous driving testbed.
- The source code will be made available via CARLab.

We will discuss Background Information in Section II; Motivations in Section III; System Design in Section IV; System Implementation in Section V; Evaluation and Results in Section VI; and Conclusion and Future Works in Section VII.

II. Background Information

In this section, we will provide a basic understanding of C-V2X, why we believe C-V2X is the most promising form of V2X, some common concerns about C-V2X, and visions for C-V2X.

A. Cellular-V2X

Cellular-V2X (C-V2X) is a communications solution for vehicles, supporting Vehicle-to-Vehicle (V2V), Vehicle-to-Pedestrian (V2P), Vehicle-to-Infrastructure (V2I), and vehicle-to-network (V2N) communications. C-V2X uses the 5.9GHz frequency band and supports direct communication through the pc5 interface and cellular communication through the Uu interface. C-V2X has the advantages of extensive coverage and low communication delay. Compared with 4G, C-V2X has a lower transmission delay; compared with WiFi, C-V2X has far more comprehensive coverage. Moreover, compared with Dedicated Short-Range Communication (DSRC), C-V2X has advantages in range and reliability. Furthermore, with cellular evolving from 4G to 5G, C-V2X technology has also evolved from LTE-V2X to NR-V2X to achieve higher bandwidth and lower communication delay. Ford and Qualcomm have compared the radio performance of C-V2X and DSRC in Ann Arbor and San Diego. Critical tests performed are shown in Table I. The results show that C-V2X has better coverage and reliability performance [5].

TABLE I
DSRC vs C-V2X radio performance comparison.

Benchmark	Experiments	Results
Congestion	Lab Cabled Congestion Control	Pass
Reliability	Lab Cabled TX and Rx Tests	C-V2X better
	Field Line-of-Sight (LOS) Range Tests	C-V2X better
	Field Non-Line-of-Sight (NLoS) Range Tests	C-V2X better
Interference	Lab Cabled Test with Simulated Co-channel Interference	C-V2X better
	Lab Cabled Near-Far Test	Pass
	Field Co-existence with WiFi 80 MHz Bandwidth in UNII-3	C-V2X better
	Field Co-existing of V2X with Adjacent DSRC Carrier	Pass

B. Reliability of C-V2X

TABLE II
DSRC vs. C-V2X Package Receptions and Losses (Single Obstacle).

Benchmark	Transmitted	Lost %
Moving Veh (C-V2X)	2997	65.03
Moving Veh (DSRC)	2325	84.69
Stationary Veh (C-V2X)	2997	57.22
Stationary Veh (DSRC)	2315	83.41

In [6], the experiment the report detailed is to predict the packet loss ratio in non-line-of-sight scenarios. For this experiment, the authors used a vehicle to act as a large metal obstacle that would prevent signals from traveling through. Therefore, the only messages the other vehicle received were from signals that took indirect paths.

[6] performed experiments for two scenarios. The report evaluates C-V2X and DSRC by measuring the Packet Reception Rate (PRR). The first scenario examined the effect of a singular stationary object obstructing the direct LOS of a vehicle-mounted antenna. For C-V2X, the PRR is 38.46%. For DSRC, the PRR is 15.95%. Next, for the second scenario, the report examined the effect of a singular stationary object

obstructing the direct LOS of a vehicle-mounted antenna moving through a narrow road with metal obstacles lining both sides. Different from the first scenario, this takes a deeper dive into how C-V2X and DSRC signals are bounced back. For C-V2X, the PRR is 79.21%; for DSRC, the PRR is 18.01%.

From Table II, we can see that C-V2X performed better than DSRC, losing a much lower percentage of packages [6].

C. Bandwidth of C-V2X

Due to bandwidth concerns, C-V2X does not allow sharing of raw data across devices. Instead, C- V2X uses the J2735 message standards to convey information. C-V2X uses it in the application layer to support the efficient interaction between V2X devices and enable tens of typical V2X applications, such as forward collision warning, blind-spot warning, red light warning, etc. Currently, there are two message set standards. The first one is TCSAE 53-2017 [2], released by China, which promotes five messages for 17 typical V2X applications. The second is J2735 2020 [1], promoted by the USA, which evolved from DSRC's message set J2735 2016, providing 17 messages for complex traffic scenarios. Both of them have BasicSafetyMessage (BSM), MapData (MAP), and SignalPhaseAndTiming Message (SPAT). BSM is used to share basic information about vehicles, such as location, speed, and heading. MAP is used for describing the traffic lanes and connections. And SPAT is designed for sharing traffic light information. The rest of the messages in the two standards are very different. T/CSAE 53-2017 defines the last two messages at a high level. RoadSideInformation (RSI) is designed to share static and dynamic road information, such as road signs, traffic crashes, and icy road alerts. Message RoadSafetyMessage (RSM) is used to share real-time traffic participant information, such as the location of vehicles and pedestrians detected by roadside cameras. J2735 2020 standard has made detailed designs according to specific needs, such as message SignalRequestMessage (SRM) for a priority signal request or a preemption signal request and message SignalStatusMessage (SSM) for publishing the current priority status of the vehicles.

D. Scaleability of C-V2X

C-V2X broadcasts messages in an area. All devices in the area will be able to receive the message. Therefore, the limitation on the scaleability of C-V2X lies with the processing power and not with the bandwidth.

E. Visions of C-V2X

In February 2020, Chen *et al.* published their paper "A Vision of C-V2X: Technologies, Field Testing, and Challenges With Chinese Development". The paper presents a vision for the possible use cases for C-V2X for autonomous driving and intelligent cities. Most importantly, the paper envisions a three-stage evolution process for C-V2X and discusses the challenges that come with it.

In the first stage, C-V2X is used for vehicle-to-vehicle and vehicle-to-road cooperative perception. In the second stage, C-V2X is used for cooperative control in closed areas with low-speed vehicles. In the third stage, C-V2X is used for cooperative control in open areas with high-speed vehicles. The three main challenges are limited perception capability, computation capability, and communication capability [10].

F. Related Works

One of the main focuses of this paper is to provide a manual for other researchers to set up their own C-V2X-enabled collaborative autonomous driving testbed. We will discuss some recent works on this subject and compare them with ours.

1) CarTest: [22]'s Hardware-in-the-loop (HIL) platform enables V2X-related testing in a HIL lab simulation environment. Their work aims to solve the inherent problems of field testing, low efficiency, coverage, controllability, and stability.

Our work sets up a similar testing environment. Different to [22]'s work, our work focuses more on the overarching framework. We include off-the-shelf C-V2X devices, a CAN-bus-enabled robotic vehicle, and additional sensors, such as cameras and, in the future, LiDAR, to support high-level applications for C-V2X-enabled autonomous vehicles.

2) TRUDI: TRUDI acts as a man-in-the-middle between the communication module and the application itself, making it possible to perform tests with the real devices and providing as an output a system ready for the road [21]. TRUDI can be used to test and validate V2X applications before they are implemented on actual vehicles.

3) [14]: builds a network-level autonomous driving simulator to better analyze the feasibility and performance of applications based on C-V2X protocol, combined with SUMO and CARLA, to evaluate road traffic and vehicle dynamics.

4) [23]: conducts an objective and independent comparison of ITS-G5 (based on IEEE 802.11p) and C-V2X (3GPP) in a real-life highway environment using off-the-shelf hardware and identical traffic conditions.

5) [24]: 's work conducts a field test of V2X device communication performance in real traffic scenarios, comparing the performance of three different devices under various parameter configurations and traffic conditions, including line of sight and non-line of sight scenarios.

6) [25]: examines two methods of LTE-based V2X for a safety application called Crash Warning Application (CWA). It compares the pros and cons of Uu-based LTE-V2X, which uses infrastructure, and PC5-based LTE-V2X, which uses D2D communication. The paper also presents quantitative performance evaluation results in terms of end-to-end latency.

These papers focus on the communication level simulation. In contrast, the CCAD framework focuses more on the overall framework, where it includes C-V2X for communication and can support camera or LiDAR for future development.

III. Motivation

This section discusses how current infrastructure or systems are insufficient to provide safety-critical information to road users. Through our investigation, we also discovered that while there are many pieces of research on V2X for collaborative autonomous driving, no paper proposed a simple framework like our CCAD framework, all of the papers involved sharing raw sensor data, and almost no framework or applications involving C-V2X.

A. *Isolated Safety-Critical Information*

As previously discussed, current intelligent vehicles are single-agent systems. They perceive safety-critical information via sensors. However, these information sources are isolated. To better organize and show how C-V2X is useful, we summarized the types of safety-critical information into three categories:

- **Infrastructure to Vehicle.** In Infrastructure to Vehicle scenarios, the infrastructure conveys safety-critical information to vehicles, such as traffic light signals, road conditions, etc. While some may argue that much of the warning information already exists as roadside signs, it is easy for a human driver to miss them either due to negligence or because another road user blocks the view. Furthermore, that are some Infrastructure to Vehicle scenarios that roadside signs cannot solve. For example, in cases where pedestrians are jaywalking, current infrastructure cannot warn the vehicles of potential danger.
- **Vehicle to Infrastructure.** In a typical Vehicle to Infrastructure scenario, an emergency vehicle needs traffic light preemption. Emergency vehicles need preemption to reduce the number of accidents between emergency vehicles and other road users. Currently, the emergency vehicle uses an IR device to send the signal to the traffic light control box when they are close to the intersection; however, the range on these devices is only 30 feet. In addition, traffic lights need at least 30 seconds before changing signals because of pedestrian walking speed. Therefore, we need a way to trigger a traffic light signal change from a greater distance, allowing other road users to finish their interaction with the intersection and restricting other road users from preempting the incoming emergency vehicle.
- **Vehicle to Vehicle.** In Vehicle to Vehicle scenarios, the vehicle may need to convey information to other vehicles, such as their speed, position, and heading. This information is then used for various ADAS functions, such as blind-spot detection. However, as mentioned previously, current intelligent vehicles collect and utilize only the data their onboard sensors collect. This leads to scenarios where if the onboard sensors cannot detect the safety-critical information, it may lead to accidents. This does not pertain to sensor failure but extends to the sensors' line of sight. For example, if a vehicle is attempting to make a lane change, but there is a speeding vehicle or a police vehicle in that lane, the blind spot sensors cannot capture this information because it is "out of sight." Other ADAS functions, such as Automatic Emergency Braking (AEB), also suffer similar issues. The current AEB function utilizes only data collected from its sensors and is prone to sensor failures.

To summarize, current intelligent vehicles require external data sources to provide additional information and redundancies for safety-critical functions. Current infrastructure requires inputs from vehicles to make the roads safer. **While concepts of V2X-enabled collaborative autonomous driving applications have been proposed, no research-friendly framework has been introduced.**

Therefore, to help other researchers build their own C-V2X-enabled autonomous driving testbed, we introduce our CCAD framework.

IV. Framework Design

In this section, we will discuss the software aspect of our framework to help researchers understand how C-V2X can be used to help single-agent autonomous driving systems. We will also discuss a lane-keeping case study to put things into perspective.

In Fig. 2, we demonstrate a high-level overview of the framework. On the smart infrastructure side, the mounted edge computing device is responsible for parsing the information received from the RSU and analyzing the collected data from the mounted sensors. The RSU receives periodic normal messages and spontaneous emergency messages from incoming vehicles. On the intelligent vehicle side, the onboard sensors collect data, and the onboard computing device analyzes the data to determine what information needs to be sent to the infrastructure or other vehicles.

A. *Vehicle to Infrastructure*

As previously mentioned, emergency vehicles need traffic light preemption. We solve this issue with the Signal Request Message (SRM) message set. The SRM message can be sent at a max range of 1km. The message is shown in Table III. The SRM message set can help emergency vehicles perform traffic light preemption because of the "requests" and "requestor" fields. The "requests" field contains a list of intersections that need traffic light preemption. The "requestor" field identifies the "id", "type", and "position" of the emergency vehicle. By combining the two fields, an emergency vehicle can use the SRM message to inform traffic lights on its path ahead of time, giving other road users at the intersection enough time for preemption.

TABLE III
Signal Request Message (SRM).

Field Name	Description	Required
`timeStamp`	Minute of the year	No
`second`	millisecond	Yes
`sequenceNumber`	MsgCount	No
`requests`	SignalRequestList	Yes
`requestor`	Requestor Description	Yes
`regional`	Regional Extension	No

B. *Infrastructure to Vehicle*

Our framework uses the TIM and MAP message set in the J2735 standards to send warnings to the vehicles. The TIM message set, as shown in Table IV uses the "dataFrames" to send warnings. The TIM message set includes the type of warnings and the geographical location of the warning zone.

For example, at a sharp curve, an RSU broadcasts the "Curve Speed Warning" with the warning zone's geographical location (latitude and longitude) to any vehicle OBUs in the broadcast radius. When the OBU receives the TIM message,

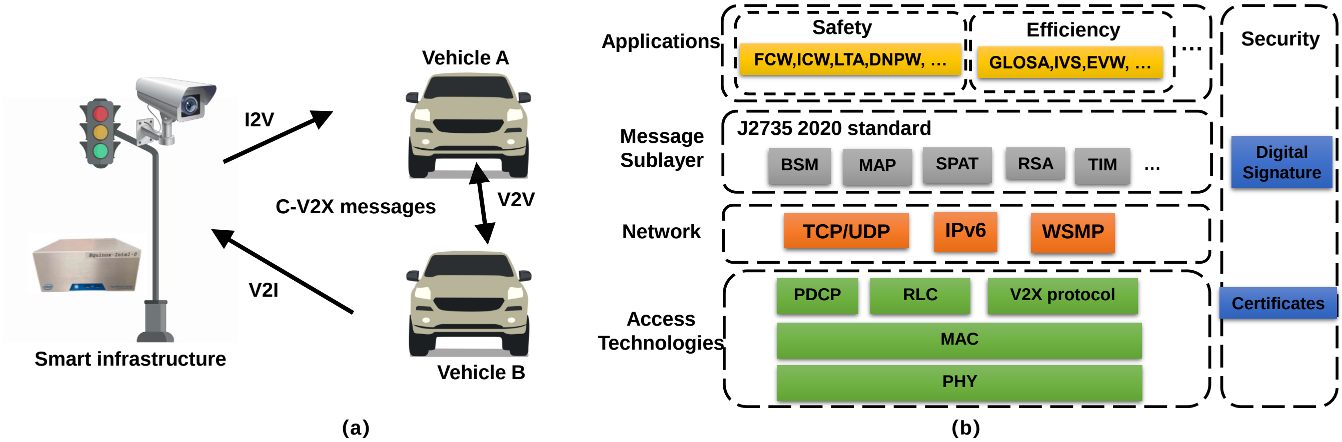


Fig. 2. CCAD framework overview. (a) C-V2X enabled a collaborative relationship between smart infrastructure and Autonomous vehicles. (b) C-V2X network stack for CCAD applications

TABLE IV
TRAVELER INFORMATION MESSAGE (TIM).

Field Name	Description	Required
`msgCnt`	Message Sequence Number	No
`timeStamp`	Time stamp of Message	No
`packetID`	Unique MSG ID	No
`urlB`	URL-Base, link to resource	No
`dataFrames`	Advisory and Road Sign Types	Yes
`regional`	Regional Extension	No

TABLE V
BASIC SAFETY MESSAGE (BSM).

Field Name	Description	Required
`coreData`	BSMcoreDat	Yes
`ID-TYPE`	id and type	Yes
`Content`	id and value	Yes
`EXT-ID`	VehicleSafety SpecialVehicle Supplement	Yes

TABLE VI
BSMCOREDATA.

Field Name	Description	Required
`msgCnt`	MsgCount	Yes
`id`	TempID	Yes
`secMark`	millisecond	Yes
`lat`	Latitude	Yes
`long`	Longitude	Yes
`elev`	Elevation	Yes
`accuracy`	PositionalAccuracy	Yes
`transmission`	TransmissionState	Yes
`speed`	Speed	Yes
`heading`	Heading	Yes
`angle`	SteeringWheelAngle	Yes
`accelSet`	AccelerationSet4Way	Yes
`brakes`	BrakeSystemStatus	Yes
`size`	VehicleSize	Yes

TABLE VII
VEHICLE SAFETY EXTENSIONS.

Field Name	Description	Required
`events`	VehicleEventFlags	No
`pathHistory`	PathHistory	No
`PathPrediction`	PathPrediction	No
`Lights`	ExteriorLights	No

the OBU stores the warning type and the geographical representation onboard. If the vehicle's GNSS sensors report that it is in the warning zone and the current vehicle speed is above the advised curve speed, the driver will be notified.

The TIM message can also be used to broadcast many other warnings, "Speed Compliance in Work Zone," "Icy Road Ahead Warning," and many more. The TIM message is superior to a simple roadside sign because when the roadside signs are blocked or ignored by the drivers, C-V2X can send these warnings straight to the driver via HMI (Human Machine Interface) in the vehicle.

C. Vehicle to Vehicle

The CCAD framework uses the Basic Safety Message(BSM) message set, as shown in Table V to send warnings or information to other vehicles. The most important fields within the BSM message set are "BSMcoreData," as shown in Table VI, and "VehicleSafetyExtensions", as shown in Table VII. BSMcoreData allows the vehicle to send information regarding its current position and heading. Broadcasting this information to other vehicles in the area allows other vehicles to know where each vehicle is. Because this information is sent through C-V2X and not collected by computer vision, vehicles will also have information about other vehicles in their blind spot and are not limited to line of sight.

Furthermore, the VehicleEventFlags field contains special event information, such as hard braking, flat tire, and many more. Previously, this information could only be observed and processed by a human driver, leaving little to no time for human drivers to react. Now, with our framework, drivers can be notified immediately when such an event occurs.

Unlike sharing raw sensor data, with the message set specified in the J2735 C-V2X message standards, we can transfer information using the least amount of data. This reduces the requirements for bandwidth, latency, and storage.

D. Priority of Messages

Some messages are more safety-critical than others. To consider this, we assign a priority level to each message. With a priority queue, higher priority messages will preempt lower priority messages and are broadcast first; and on the vehicle's HMI (Human Machine Interface), higher priority warnings will be displayed on top of lower priority warnings. If messages share the same priority level, the messages will follow FIFO.

E. Communication Safety

With C-V2X as a communication method, we have to consider communication safety. Communication safety can be broken down into several types:

- Unauthorized access.
- Distributed Denial of Service (DDoS) attacks.
- Man in the Middle.
- Privilege escalation.
- Insider Threats.

Unauthorized access occurs when the attacker send unauthenticated C-V2X messages to other devices; attackers can perform DDoS attacks by flooding the communication channel with meaningless messages; attackers can disguise themselves as a legitimate RSU or OBU to intercept and alter messages with the Man in the Middle attack; attackers can, through privilege escalation, send high priority messages even though the attacker's device does not have the authority; insider threats may happen when the attacker takes control of a legitimate device and send false information.

The CCAD framework can defend against some of these attacks by verifying the signature of RSUs and OBUs, limiting the number of messages a device can send in a given time frame, and verifying the checksum of the messages. Jung's paper "Reliability Verification Procedure of Secured V2X Communication for Autonomous Cooperation Driving" also provides three methods - certificates, hardware IDs, and public and private keys - to provide communication security [19].

F. Case Study

To demonstrate some of the functionalities above and prove that the CCAD framework can transmit safety-critical information in real-time and increase the safety of vehicles, we present Collaborative Lane Keeping (COLK), a case study of our framework.

In this case study, lane markings are partially covered, missing, or no longer trustworthy. This could result from snow, rain, construction, or accidents. As a result, the vehicle's onboard computing devices cannot make a safe decision based on the collected data. Instead, the infrastructure, knowing the ground truth of the lanes, will transmit vehicle-specific guidance messages to help the vehicles to stay in their lane.

Fig. 1 demonstrates a few typical scenarios where the COLK application is needed.

In scenario A, an accident causes the lanes to be closed. These impromptu changes in lane markings are not updated in the lane marking maps for autonomous vehicles in real-time, and vehicles with ADAS systems would transfer the control to the driver. This might cause an autonomous vehicle with an outdated lane marking map to drive into perilous situations erroneously.

In scenario B, lane markings have become unclear or missing when the lane markings have either deteriorated from years of wear and tear, covered by snow or dirt, or become unrecognizable by onboard cameras due to weather or camera conditions. While some autonomous vehicles resolve this problem by removing lane detection entirely and rely on High-Definition maps to perform localization and lane-keeping, others rely on cameras to ensure the vehicle stays in the lane. Moreover, in situations where heavy snow is covering the lane markings, drivers tend not to drive in lane but to drive in the previous vehicles' tracks. While this ensures safety, it also reduces the number of lanes available and increases traffic congestion.

In scenario C, road construction and temporary lane closure might cause the lane markings to change temporarily. However, as mentioned before, the lane marking map of an autonomous vehicle is not updated in real-time; therefore, when such a scenario occurs, there is the risk of it driving into a hazardous environment.

In scenario D, because lane connections are not always straight or drawn out, some lane connections might be confusing. Without knowing the exact lane connection in the lane marking maps, an autonomous vehicle risks driving or turning into the opposing lane, and a vehicle equipped with ADAS systems has no way of knowing where to drive when lane markings are vague.

V. Framework and Case Study Implementation

In this section, we demonstrate the setup of the CCAD framework and our infrastructure-based Collaborative Lane-Keeping (COLK) case study. By presenting this case study, we aim to demonstrate that C-V2X-enabled infrastructure has the ability to make autonomous systems safer. In addition, we aim to help researchers physically set up a testbed of their own to study and experiment with C-V2X-enabled collaborative autonomous driving.

Fig. 3 shows the setup of the CCAD framework in a downsized environment. A camera is mounted on the pole to the left. This camera looks down at the lanes drawn on the ground in black. Several boxes on or next to the lanes act as obstacles. On the right, a four-wheeled robot represents a test vehicle. It has a QR code in place of the plate number for self-identifying information. The height of the camera, size of the robotic vehicle, and width of lanes lines are shrunk on the same scale to reflect real-life sizes. The speed of the test vehicle is also scaled down.

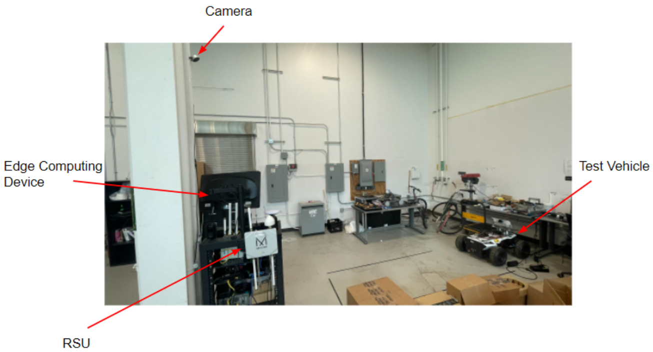


Fig. 3. Experimental grounds setup overview.

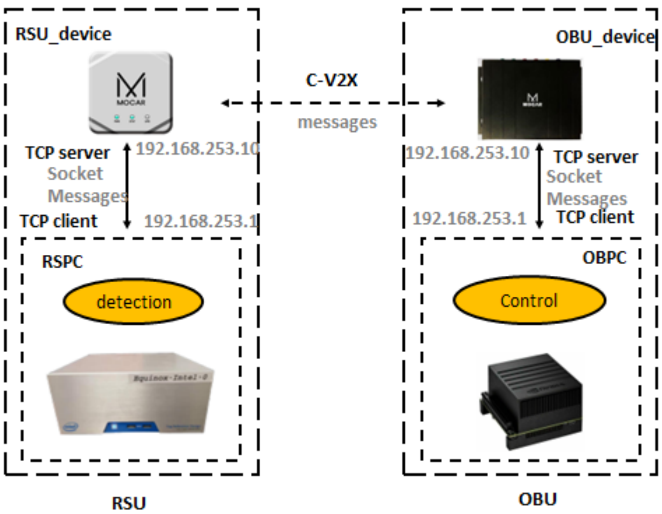


Fig. 4. CCAD message-flow framework.

Fig. 4 presents the basic communication framework of CCAD. Road-side edge Processing Computer (RSPC) detects whether the car is in the lane and sends guidance messages to the OBU with C-V2X messages. The OBU receives the messages and processes the information in OBPC and determines how to control the vehicle to stay in the lane.

The case study is designed so that when the vehicle's onboard sensors cannot find a safe path, the CCAD framework will supply additional information to guide the vehicle onto a safe path. This means that while the vehicle itself may be equipped with HD-maps, LiDAR, and cameras, these sensors cannot help the vehicle localize or perform lane following due to various reasons.

In order to accomplish this, the COLK application consists of three major components: Detection, Localization, and Communication.

Fig. 5 shows a high-level overview of the COLK application. First, the vehicle's OBU (On-Board Unit) broadcasts self-identifying information to inform the infrastructure of who it is and its appearance. Next, the camera mounted on the infrastructure pre-records and stores the lane information even if the lanes are not visible. The camera also detects, identifies, and localizes the vehicle in the scene. Next, the

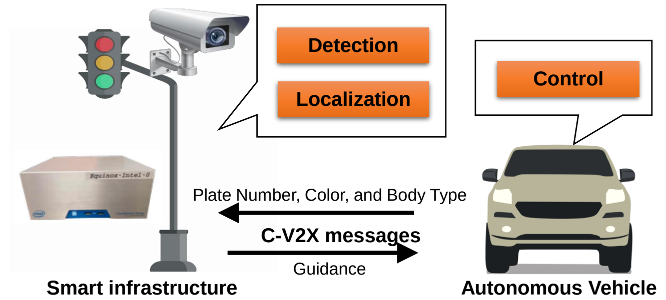


Fig. 5. COLK application design overview.

embedded edge computing device takes this information and computes whether or not the vehicle is in the lane. Finally, the infrastructure RSU (Road-Side Unit) broadcasts the vehicle-specific guidance messages.

First, we will discuss Detection. In COLK, detection is the first crucial component in our system. It is responsible for detecting, identifying, and localizing the vehicle.

To improve the run-time for YOLO, the sample space needs to be reduced. To accomplish this, background subtraction removes all the static objects in the scene. This does not affect the effectiveness of COLK because stationary objects do not require additional information to stay safe. This is the same for temporarily parked vehicles. For temporarily parked vehicles, because it has been under the COLK system's guidance up to the point when they stopped, COLK knows that the vehicle is on a safe path. When the vehicle starts moving again, the system will pick up its movement and provide guidance messages.

Once the vehicles are detected in the scene, they need to be assigned unique identifications. Each vehicle's OBU broadcasts its self-identifying messages, including plate number and vehicle description (color, body type). To also consider situations where plate numbers are not visible from the front, we add a QR code to the front of the vehicle to allow identification matching from a distance. This self-identifying information is broadcasted with the PVD message set shown in Table VIII. We will discuss the message set used in detail later in this section.

The embedded edge computing device on the infrastructure uses the camera data and the self-identifying message from the vehicles to match the detected vehicles to actual vehicles so that the COLK application can send the correct guidance to the correct vehicle.

To ensure that CLOCK matches each vehicle with the correct identification, it must fully utilize the information the vehicle broadcasts. From the infrastructure point of view, COLK has the vehicle's color, body type, and plate number. In the previous step, it had segmented vehicles from the scene. The system can use the color information to match colors first, and then it can use letter recognition to match the plate numbers. One of the challenges is matching partially obstructed plate numbers.

Based on a probability calculation, we decided to match identification information using these parameters: Plate Number, Color, and Body Type.

If a plate consists of 7 alphabets or numbers, and "O" is not allowed. Then the total number of distinct plates is $6.43e{+}10$. Let us say the last four places of the plate are partially blocked. There will be a total of $1.5e{+}6$ plates with the same partial first three numbers or alphabets. Then, picking the most popular body type, crossover, there are now $6.89e{+}5$ vehicles with the same partial plate and body type. Add in the color information and pick the most popular color, white, there are now only $1.65e{+}5$ vehicles with the same partial three numbers or alphabets, the same color, and the same body type. While this may seem large, the probability comes down to $2.56e{-}6$. This equates to about 1 in 39,083 probability of 2 or more vehicles having the same partial plate, color, and body type. The iPhone's fingerprint touch ID has a probability of 1 in 50,000 recognizing two similar fingerprints as the same. Therefore, we believe that using these three parameters, plate number, color, and body type, is sufficient to match identification information correctly.

Next, we will discuss Localization. Localization is responsible for placing the detected vehicle in the same frame as the ground truth lanes. The infrastructure can then determine whether or not the vehicle is in lane or skewing to either left or right.

As previously mentioned, the embedded computing device pre-records the lanes' ground truth. Therefore, localization of the detected vehicles involves segmenting the vehicles, finding the centroid of the segmented objects, the comparing the centroid location relative to the left and right lane boundaries

The recorded ground truth for lane markings is editable on the fly. When an accident or sudden changes occur, COLK can edit a new path into the system for vehicles to follow.

Last but not least, we will discuss Communication. Communication is responsible for communicating with the detected vehicle and guiding it to stay in its lane.

In the message list of C-V2X standards, COLK uses the following message sets to transmit the vehicle's self-identifying information and the safety-critical guidance message.

First, we will discuss the message sets in the COLK application to transmit the vehicle's self-identifying information.

TABLE VIII
PROBE VEHICLE DATA (PVD) MESSAGE SET.

Field Name	Description	Required
`timeStamp`	MinuteOfTheYear	No
`segNum`	ProbSegmentNumber	No
`probeID`	VehicleIdent	No
`StartVector`	FullPositionVector to RSU	Yes
`VehicleType`	VehicleClassification	No
`snapshots`	Snapshot	No
`regional`	RegionalExtension	No

The message set used is the PVD (ProbeVehicleData) message, as shown in Table VIII. The PVD message set is used to transmit vehicle data. Most importantly, the field required is "VehicleIdent." This field is used to provide information about a selected vehicle.

Within the "VehicleIndent" field, as shown in Table IX, COLK uses the "DescriptiveName" field. This field allows us to add descriptive information about the selected vehicle. The COLK application uses the vehicle's plate number, color, and body type in this field.

TABLE IX
VEHICLEIDENT.

Field Name	Description	Required
`name`	DescriptiveName	No
`vin`	VINstring	No
`ownerCode`	vehicle owner code	No
`id`	VehicleID, same as BSM	No
`vehicleType`	VehicleType	No
`vehicleClass`	VehicleClass	No

TABLE X
VEHICLE ID

Field Name	Description	Required
`entityID`	TemporaryID	Yes
`stationID`	StationID	Yes

The "VehicleID" field, as shown in Table X, provides personalized vehicle guidance messages. To prevent tracking, this ID is a randomly generated string that stays constant and unique for a short time.

The COLK application does not use the "VehicleType" field, shown in Table IX, to include the vehicle type because the list of vehicle types does not include the different body types for commercial vehicles; instead, this field is used to classify vehicles based on overall size.

Next, we show the message set in the COLK application to transmit safety-critical guidance messages to the vehicle.

COLK uses the BSM message to transmit the guidance message. Specifically, the angle field in BSMcoreData is used to provide guidance. This field is generally used to transmit the vehicle's steering wheel angle, ranging from -1 to 1. In real-life scenarios, drivers may never turn the steering wheel to its leftmost or rightmost angle when the vehicle is driving in a straight line. Therefore, COLK will use the values -1 and 1 to indicate if the vehicle is skewed to the left or to the right. If the vehicle is swaying to the left of the lane, COLK sends a "1"; if the vehicle is at the center of the lane, COLK does not send a guidance message to that vehicle because the message would be redundant and might create congestion in the communication channel. If the vehicle is to the right of the lane, COLK sends a "-1".

Next, we will demonstrate the computing parts of the COLK application, starting with detection.

Initially, a frame is captured by the mounted camera. Because the camera is stationary, the background subtraction method is applied to reduce the search space for moving vehicles. As explained in the previous section, only moving vehicles have the potential to move off-course. Next, applying opening morphology techniques reduces the amount of false-positive regions of interest (ROIs). Dilation morphology is also applied to reduce the amount of false negative ROIs. Next, an ROI mask is applied to the original frame to get a smaller frame that contains only the moving object. Last but not least, COLK runs an object detection algorithm (YOLOv5) to determine if there are any vehicles in the ROIs. If vehicles are in the ROIs, they are assigned unique IDs, store the vehicles' appearance information returned from the object detection algorithm, and assign the centroid information to specific IDs.

From the point of view of the edge computing device, it has collected unique IDs and vehicle appearance information from its camera.

Next, COLK uses the message set used for the vehicle to transmit its self-identifying information via its On-Board Unit (OBU). This information will match the vehicle with the information collected in the detection phase. The message set is explained in Table VIII.

Next, COLK needs to match the detected information with the received information. *dict_VehicleInfo* is a dictionary of all of the vehicle information collected by the infrastructure-mounted camera, *dict_PVDInfo* is a dictionary of all of the vehicle information transmitted by the PVD message set by each vehicle. It iterates through the contents in both dictionaries and matches the color, body type, and plate number. Once a match is found, the system assigns the transmitted vehicle-identifying information to the detected vehicle. This allows us to send vehicle-specific guidance messages. Our down-sized experiment environment uses QR codes to supplement vehicle identification information.

At this point, COLK has matched the vehicles so the system can send specific instructions. Therefore, the next step is to localize the vehicles and check to see whether the vehicle is in the center of the lane.

In Localization, we set the threshold to be 10 percent of the distance between LL and RL on the horizontal line of the vehicle's centroid. If (VehicleCentroid - CenterofLane) $>$ threshold, then the vehicle is skewed to the left. If (CenterofLane - VehicleCentroid) $>$ threshold, then the vehicle is skewed to the right. We also show this in Fig.6 to better illustrate our localization method.

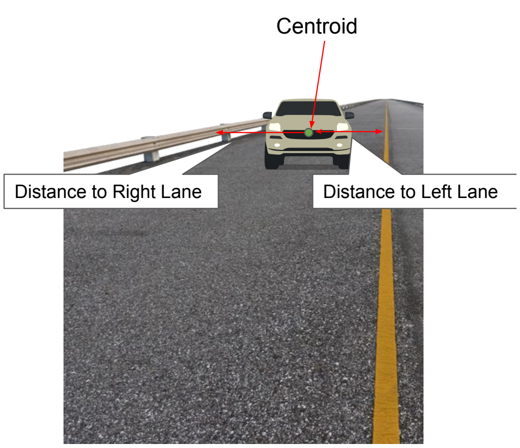


Fig. 6. Localizing a vehicle in lane.

Finally, with matching vehicle identification information and specific guidance messages, we send the guidance message to the target vehicle with the BSM message set. When the vehicle receives the guidance message, its onboard computer can compare its angle and the guidance message's angle to check its current position in the lane.

VI. Evaluation and Results

In this section, we evaluate the performance of the CCAD framework and the Collaborative Lane-Keeping (COLK) application. In this case study, we hypnosis a vehicle driving at 45 miles/h (72 km/h) on a suburban road and another vehicle driving at 20 miles/h (32 km/h) on a city road where the lane markings are not visible. The camera captures the video at 30 frames per second and has a resolution of 1080p. Due to the hardware limitation of low-cost commercial cameras, the algorithm will only identify, match, and broadcast guidance messages for vehicles that are less than 100 meters away from the camera.

A. Hardware Setup

Our hardware setup is as follows:

TABLE XI
CCAD Hardware Specifications.

	Infrastructure	Vehicle	C-V2X Device
Device Name	Intel RFD	AGX	ISW RSU / OBU
OS	ubuntu 18.04	ubuntu 18.04	Linux4.9.11
CPU	Intel Xeon E3-1275 v5	ARM v8	Cortex-A9
GPU	NONE	Volta GPU	NONE
Frequency	3.6GHz	2265MHz	800MHz
Core	4	8	2
Memory	32GB	32GB	8GB

The Intel Fog Reference Design (RFD) deploys a powerful Intel Xeon E3-1275 v5 with a frequency of 3.6GHz. The NVIDIA AGX Xavier deploys a 512-core Volta GPU with Tensor Cores and has a variety of working modes, and the frequency can be up to 2265MHz. iSmartWay Roadside Unit and Onboard Unit use the same OS, CPU, and Memory. More details are shown in Table XI.

B. Detection

Fig. 7 a, shows the original scene captured by the infrastructure's camera. The lanes are drawn in the scene with yellow lines, and several boxes are placed on or next to the lane markings to simulate a construction site. With the construction site, the lanes have shifted and are not visible. The robot vehicle acts as a vehicle and has a QR code to provide additional identifying information such as plate number, color, and body type.

Fig. 7 c, demonstrates why background subtraction without additional image processing is insufficient. The background subtraction algorithm picks up lots of small movements shown as white points.

Fig. 7 d, demonstrates the Background Mask after additional image processing. We can see that noises are removed, leaving only the region of the test vehicle.

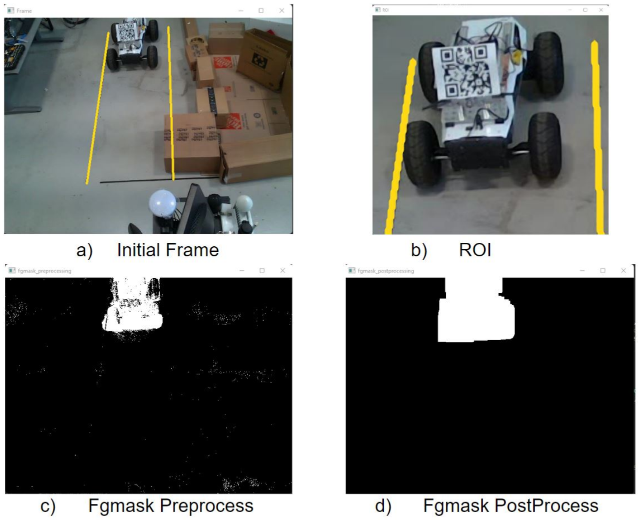


Fig. 7. Detection Pre-processing.

Next, the background mask from the previous step is applied to the original frame. The result is shown in Fig. 7 b. This step is crucial because it can significantly reduce the execution time by reducing the search space for the YOLO algorithm. This step is also important for reading the QR Code mounted on the test vehicle for this simulated environment. Because of the camera's low resolution, it is hard for the QR code reader to read it correctly. To resolve this issue, we apply super-resolution to the ROI to increase the resolution of the QR code. Having a smaller image also decreases the execution time for the super-resolution algorithm.

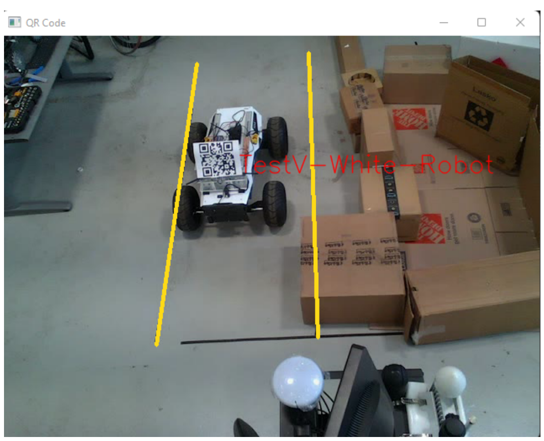


Fig. 8. Matching vehicle information.

Fig. 8, demonstrate the result of matching the information transmitted by the vehicle's OBU and the camera-captured information. This step is crucial to the COLK application because the system needs to match the vehicle with the corresponding guidance messages. Here we chose to use a QR code to represent the vehicle's color, type, and plate number. In reality, there will not be QR codes on top of vehicles; However, in testing, QR codes are usually used for indoor robots and have a similar recognition time when compared to Yolo's recognition times.

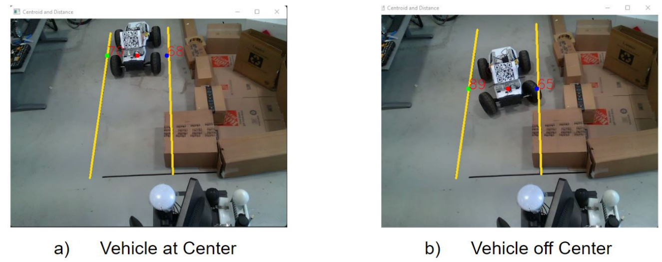


Fig. 9. Vehicle's centroid and distance to both Lanes.

After matching the vehicle information with the camera-captured information, the vehicle's centroid and the distance to the left and right lanes are calculated and compared. Fig. 9 a illustrates the scene when the test vehicle is close to the center of the lane. The red dot is the vehicle's centroid; The blue is the left lane, and the green is the right lane. The red number above the green and blue dots is the distance between the vehicle's centroid to the left and right lanes. In this case, the RSU does not send guidance messages to the vehicle because the vehicle is already centered. Fig. 9 b illustrates the centroid and distance information when the vehicle is off-center and is at risk of driving into another lane. We can see that the difference between the distance between the vehicle's centroid to the left and right lane is greater than the threshold value. In this case, in the guidance message, COLK need to tell the vehicle that it is off-center and swayed to the left.

C. End-to-End Latency

We need to verify that the CCAD framework and the COLK application can alert vehicles and drivers in safety-critical situations in real time. The end-to-end latency for COLK has three parts, the pre-processing latency, the YOLO detection latency, and the C-V2X message latency.

TABLE XII
C-V2X MESSAGE LATENCY.

Distance	0m	5m	10m	20m	50m	100m
Latency	14.43ms	17.31ms	13.53ms	15.62ms	16.75ms	15.61ms

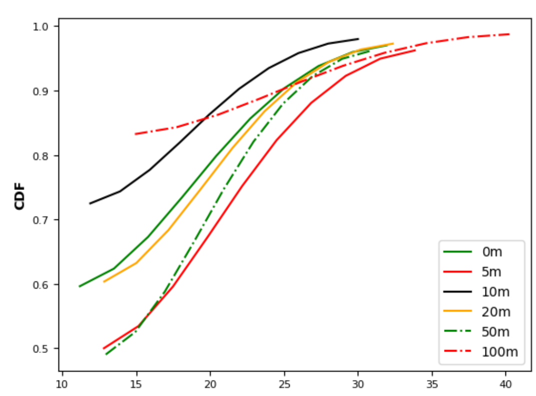


Fig. 10. Latency CDF by Distance.

C-V2X message latency is the most important part of the COLK application. Table XIIand Fig 10 present the latency of the C-V2X message latency. Our experiment demonstrates that our hardware setup's C-V2X message latency has an average delay of 15.51ms between 0 meters to 100 meters. For distances further than 100 meters, it becomes more difficult for a camera to distinguish a vehicle's plate number, color, and body type. While more powerful cameras and algorithms can increase this range, the cost and processing time may increase. Therefore we only consider a maximum distance of 100 meters.

In our experiment, the average pre-processing latency is 24.73 ms; The average YOLOv5 detection latency is 58.53 ms, and the average C-V2X message latency is 15.54 ms. Therefore, the total end-to-end latency averages 98.8 ms.

TABLE XIII
END-TO-END LATENCY.

	Environment	Speed	Delay Along the Road	Delay Perp to the Road
Vehicle A	Suburban	45mi/h (72km/h)	1.91m	1.677ft (0.511m)
Vehicle B	City	20mi/h (32km/h)	0.848m	0.745ft (0.227m)

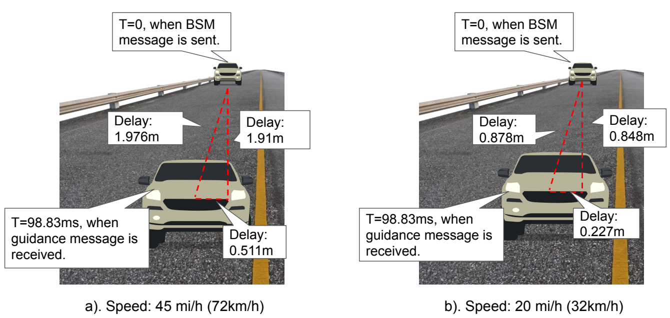


Fig. 11. Distance Delay.

In a suburban environment where the speed limit is around 45mi/h (72km/h), this latency would introduce a 1.976m delay. However, considering a vehicle's movement in the lane is mostly along the direction of the lane, this results in 0.511m of delay perpendicular to the lane if the angle between the vehicle and the lanes is 15 degrees. For a city environment where the speed limit is around 20mi/h (32km/h), this latency would introduce a 1.284m delay. If the angle between the vehicle and the lanes is 15 degrees, this would result in a 0.227m delay perpendicular to the lane. To better illustrate the latency and the effect of the latency, we show them in Table XIII and Fig. 11

D. Real-time Schedule

To examine the real-time aspect, we need to consider the detection and matching time and message period. Our detection and matching algorithm has an average execution time of 24.73 ms. YOLOv5 has an average execution time of 58.53 ms. The average message latency is 15.54 ms. This process takes around 82ms, and the BSM message is sent periodically at 100 ms.

In Figure. 12, the top timeline demonstrates when the message will be sent if there are no other applications using the BSM message. Because the total time it takes the COLK application to prepare the message is 83ms, and the BSM message is sent every 100ms, the guidance message will be sent during the period. The bottom timeline demonstrates when the message will be sent if another application blocks the COLK application from using the BSM message. If the block is long enough, it will push the guidance message into the next BSM period. As a result, the message delay would be longer. However, in cases where the message needs to be sent immediately to avoid a collision, the COLK application can also send emergency messages on demand.

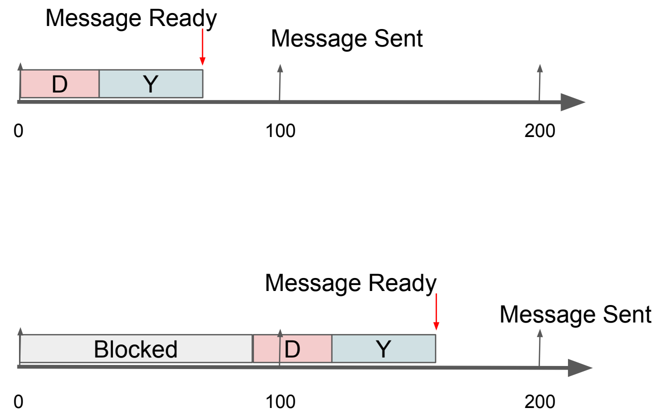


Fig. 12. Real-time message schedule.

E. Effects of Occlusion

In real-life scenarios, there might be occlusions. We concluded that three types of occlusion might happen: the road is not busy, no occlusion to plate numbers and vehicles, the road is half-full, minor occlusion to plate numbers and vehicles, the road is packed, full occlusion to plate numbers and vehicles. We will briefly discuss the effects of occlusion in each of these scenarios.

In scenarios where the road is not busy, and where there is no occlusion to the plate numbers and vehicles, we have demonstrated that our CCAD framework and COLK application work well. In scenarios where the road is half-full, and where there is minor occlusion to the plate numbers, just matching the plate number is not sufficient. As previously mentioned, matching parts of the plate number, color, and body type using YOLOv5 have a 1 in 39,083 probability of matching 2 or more vehicles at the same time. We believe this is sufficient and is able to avoid duplicate matching. Finally, in scenarios where the road is completely full, and where the plate numbers are fully occluded, the COLK application can no longer be used; however, in such a scenario, there is no need for the COLK application, because each vehicle will follow the vehicle in the front. The COLK application can still give the vehicle the correct personalized vehicle guidance message for the front of the pack.

F. Summary

In this case study of the CCAD framework, we have demonstrated that it can increase the safety of intelligent vehicles in real-time.

In the original scene, the single-agent intelligent vehicle cannot find a safe path because it cannot find the lanes. Through our CCAD framework, the COLK application, and the C-V2X standard message set defined in J2735, we achieved collaborative ADAS by successfully mapping out the lanes and sending guidance messages for the intelligent vehicle to follow. While actual vehicles and intersections were not tested, we believe this case study reinforces the vision many other researchers envisioned - a simple C-V2X framework to implement for collaborative autonomous driving.

However, in writing this paper, we realize that the CCAD framework and the current standards have some flaws. One of the most crucial flaws is that a message set was used for something it was not designed to do. This is because the standards are still only in the first stage, as defined by [10]. One other experiment we did not do was comparing our method with GNSS sensors. This is because current commercial GNSS sensors have limited accuracy. While there are more accurate GNSS sensors, localization methods, or external devices such as Real-time Kinetic (RTK) sensors, they are expensive and incur additional costs for the consumer.

VII. Future Works

In this paper, we presented our CCAD framework and demonstrated with a lane-keeping case study that our framework can increase vehicle safety.

The CCAD framework can be improved with better hardware and pattern recognition algorithms. In our case study, CCAD only broadcasts guidance messages for vehicles that are less than 100 meters away. With future technologies, it is possible for our framework to cover a much larger area.

Our work can be expanded to other safety-critical scenarios, such as forward collision warnings, blind-spot warnings, red light warnings, and many others. While many current C-V2X applications deal with these warnings, the CCAD framework can use the infrastructure's global view to capture these warnings.

For example, for forward collision and blind-spot warnings, because the infrastructure has a bird's eye view of the scene, if localization is performed correctly, then if the vertical distance is too short, the infrastructure will send a collision warning to both vehicles. For red-light warnings, because the infrastructure has knowledge of the signal time phasing, it can calculate the distance a vehicle requires to break before the light safely. If localization is done correctly, it can warn the vehicle if it will run a red light if it does not start to decelerate.

In addition, CCAD can also be used for accident detection and emergency response. [26] proposes a low-cost accident detection and notification system using V2X Communication and Edge/Cloud computing. [27] proposes an emergence response system using OBUs and Raspberry Pis. With the help of our framework, the infrastructure and other camera-and-OBU-equipped vehicles can be used for accident detection and emergency response. The testbed [27] implemented can also be easily ported to our framework.

VIII. Conclusion

Intelligent vehicles are single-agent systems. Inherently, single-agent systems have limited capabilities. As a result, accidents still occur despite rapid advancements in hardware and algorithms. V2X communication can supplement some of these missing capabilities by allowing the exchange of safety-critical information. Cellular-V2X increases the reliability, coverage, and data transmission speed.

In this article, we presented CCAD, a simple and easy-to-implement framework, and demonstrated a case study to show that the CCAD framework can transmit safety-critical information in real-time and increase intelligent vehicle safety.

Most importantly, we offer a baseline and a manual to build a C-V2X-enabled collaborative autonomous driving testbed for researchers who want to study and experiment with it.

References

[1] SAE International, "V2X Communications Message Set Dictionary_J2735202007," www.sae.org https://www.sae.org/standards/content/j2735_202007/, 2020 (accessed: June. 19, 2022).

[2] SAE China, "T/CSAE 53-2017 Cooperative intelligent transportation system; vehicular communication;application layer specification and data exchange standard (English Version)," www.codeofchina.com https://www.codeofchina.com/standard/TCSAE53-2017.html, 2017 (accessed: June. 25, 2022).

[3] NHTSA, "Summary Report: Standing General Order on Crash Reporting for Level 2 Advanced Driver Assistance Systems," www.nhtsa.gov https://www.nhtsa.gov/sites/nhtsa.gov/files/2022-06/ADAS-L2-SGO-Report-June-2022.pdf, 2022 (accessed: June. 19, 2022).

[4] NHTSA, "Summary Report: Standing General Order on Crash Reporting for Automated Driving Systems," www.nhtsa.gov https://www.nhtsa.gov/sites/nhtsa.gov/files/2022-06/ADS-SGO-Report-June-2022.pdf, 2022 (accessed: June. 19, 2022).

[5] V2X Technology Benchmark Testing - Qualcomm. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/fcc_usdot_cv2x_-_v2.14_w_video1.pdf, 2018 (accessed: June, 2022)

[6] CCAT. (2022, February 3). Reliable-V2V-Communication-Networks-Applications-in-Fuel-Efficient-Platooning-Final-Report. Center for Connected and Automated Transportation, 2022, Retrieved February 3, 2022, from https://ccat.umtri.umich.edu/

[7] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer and A. Kovacs, "Enhancements of V2X communication in support of cooperative autonomous driving," in IEEE Communications Magazine, vol. 53, no. 12, pp. 64-70, Dec. 2015, doi: 10.1109/MCOM.2015.7355568.

[8] H. Stübing et al., "simTD: a car-to-X system architecture for field operational tests [Topics in Automotive Networking]," in IEEE Communications Magazine, vol. 48, no. 5, pp. 148-154, May 2010, doi: 10.1109/MCOM.2010.5458376.

[9] R. Deng, B. Di and L. Song, "Cooperative Collision Avoidance for Overtaking Maneuvers in Cellular V2X-Based Autonomous Driving," in IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4434-4446, May 2019, doi: 10.1109/TVT.2019.2906509.

[10] S. Chen, J. Hu, Y. Shi, L. Zhao and W. Li, "A Vision of C-V2X: Technologies, Field Testing, and Challenges With Chinese Development," in IEEE Internet of Things Journal, vol. 7, no. 5, pp. 3872-3881, May 2020, doi: 10.1109/JIOT.2020.2974823.

[11] Z. Dong, W. Shi, G. Tong and K. Yang, "Collaborative Autonomous Driving: Vision and Challenges," 2020 International Conference on Connected and Autonomous Driving (MetroCAD), 2020, pp. 17-26, doi: 10.1109/MetroCAD48866.2020.00010.

[12] Z. Li, T. Yu, R. Fukatsu, G. K. Tran and K. Sakaguchi, "Towards Safe Automated Driving: Design of Software-Defined Dynamic MmWave V2X Networks and PoC Implementation," in IEEE Open Journal of Vehicular Technology, vol. 2, pp. 78-93, 2021, doi: 10.1109/OJVT.2021.3049783.

[13] H. Bagheri et al., "5G NR-V2X: Toward Connected and Cooperative Autonomous Driving," in IEEE Communications Standards Magazine, vol. 5, no. 1, pp. 48-54, March 2021, doi: 10.1109/MCOMSTD.001.2000069.

[14] S. Fu, W. Zhang and Z. Jiang, "A network-level connected autonomous driving evaluation platform implementing C-V2X technology," in China Communications, vol. 18, no. 6, pp. 77-88, June 2021, doi: 10.23919/JCC.2021.06.007.

[15] R. Yu, D. Yang and H. Zhang, "Edge-Assisted Collaborative Perception in Autonomous Driving: A Reflection on Communication Design," 2021 IEEE/ACM Symposium on Edge Computing (SEC), 2021, pp. 371-375, doi: 10.1145/3453142.3491413.

[16] H. Ma, S. Li, E. Zhang, Z. Lv, J. Hu and X. Wei, "Cooperative Autonomous Driving Oriented MEC-Aided 5G-V2X: Prototype System Design, Field Tests and AI-Based Optimization Tools," in IEEE Access, vol. 8, pp. 54288-54302, 2020, doi: 10.1109/ACCESS.2020.2981463.

[17] T. H. Dinh, V. Martinez and D. Delahaye, "Recognition of Outlying Driving Behaviors: A Data-Driven Perspective with Applications to V2X Collective Perception," 2021 IEEE Vehicular Networking Conference (VNC), 2021, pp. 52-59, doi: 10.1109/VNC52810.2021.9644627.

[18] P. Ghorai, A. Eskandarian, Y. -K. Kim and G. Mehr, "State Estimation and Motion Prediction of Vehicles and Vulnerable Road Users for Cooperative Autonomous Driving: A Survey," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2022.3160932.

[19] H. Jung et al., "Reliability Verification Procedure of Secured V2X Communication for Autonomous Cooperation Driving," 2018 International Conference on Information and Communication Technology Convergence (ICTC), 2018, pp. 1356-1360, doi: 10.1109/ICTC.2018.8539617.

[20] L. Sridhar, R. Paul, W. Paul, "Reliable V2V Communication Networks: Applications in Fuel-Efficient Platooning", Published Date: 2021-03-10, Report Number: CCAT Project Number 16, doi: http://dx.doi.org/10.7302/381

[21] M. Menarini, P. Marrancone, G. Cecchini, A. Bazzi, B. M. Masini and A. Zanella, "TRUDI: Testing Environment for Vehicular Applications Running with Devices in the Loop," 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, 2019, pp. 1-6, doi: 10.1109/ICCVE45908.2019.8965152.

[22] Wang J, Zhu Y. A Hardware-in-the-Loop V2X Simulation Framework: CarTest. Sensors (Basel). 2022;22(13):5019. Published 2022 Jul 3. doi:10.3390/s22135019

[23] V. Maglogiannis, D. Naudts, S. Hadiwardoyo, D. van den Akker, J. Marquez-Barja and I. Moerman, "Experimental V2X Evaluation for C-V2X and ITS-G5 Technologies in a Real-Life Highway Environment," in IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 1521-1538, June 2022, doi: 10.1109/TNSM.2021.3129348.

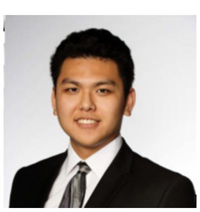
[24] J. Gao, S. Yin, Q. Yang, S. Niu and L. Lu, "Analysis for V2X Communication Device Performance Field Testing," 2021 6th International Conference on Smart Grid and Electrical Automation (ICSGEA), Kunming, China, 2021, pp. 347-351, doi: 10.1109/ICSGEA53208.2021.00085.

[25] R. Kawasaki, H. Onishi and T. Murase, "Performance evaluation on V2X communication with PC5-based and Uu-based LTE in Crash Warning Application," 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 2017, pp. 1-2, doi: 10.1109/GCCE.2017.8229434.

[26] Khaliq, Kishwer Abdul, Omer Chughtai, Abdullah Shahwani, Amir Qayyum and Jürgen Pannek. "Road Accidents Detection, Data Collection and Data Analysis Using V2X Communication and Edge/Cloud Computing." Electronics (2019): n. pag.

[27] Khaliq, Kishwer Abdul, Omer Chughtai, Abdullah Shahwani, Amir Qayyum and Jürgen Pannek. "An Emergency Response System: Construction, Validation, and Experiments for Disaster Management in a Vehicular Environment." Sensors (Basel, Switzerland) 19 (2019): n. pag.

IX. Biography



Yuankai He received his Bachelor degree in Computer Engineering at Boston University. He is currently a Ph.D student at University of Delaware and a member of the Connected and Autonomous Research Laboratory(CAR). His research interests include computer vision, simulation, LiDAR, C-V2X, connected autonomous vehicles, and machine learning.

Baofu Wu received his Master's degree in School of Computer Science of Hangzhou Dianzi University. He is also a Ph.D. candidate at Hangzhou Dianzi University and a visiting scholar in Connected and Autonomous Research Laboratory (CAR) at Wayne State University. His research interests include edge computing, connected and autonomous vehicles, communication, parallel computing, and machine learning.

Zheng Dong (Member, IEEE) received a BS degree from Wuhan University, China, in 2007, the MS degree from the University of Science and Technology of China, in 2011, and the Ph.D. degree from the University of Texas at Dallas, USA, in 2019. He is an assistant professor at the Department of Computer Science, Wayne State University, Detroit, Michigan. His research interests are in real-time embedded computer systems and connected autonomous driving systems. His current research focus is on multiprocessor scheduling theory and hardware-software co-design for real-time applications. He received the Outstanding Paper Award at the 38th IEEE RTSS. He is a member of the IEEE Computer Society.

Jian Wan received the B.S. degree in mechanical engineering, the M.S. degree in mathematics, and the Ph.D. degree in computer science from Zhejiang University, in 1990, 1993, and 1996, respectively. From 2000 to 2015, he was the Dean of the School of Computer Science and Technology, Hangzhou Dianzi University. He is currently a Professor of the School of Information Engineering, Zhejiang University of Science and Technology, Hangzhou, China, where he is also the Vice President and also the Vice Director of the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou. He has published more than 150 articles in grid and services computing, cloud computing, virtualization, and distributed systems. His research interests include distributed systems, computer networks, and big data analytics.

Weisong Shi Weisong Shi is a Professor and Chair of the Department of Computer and Information Sciences at the University of Delaware (UD), where he leads the Connected and Autonomous Research (CAR) Laboratory. Dr. Shi is an internationally renowned expert in edge computing, autonomous driving and connected health. Dr. Shi has published more than 290 articles in peer-reviewed journals and conferences and served in editorial roles for more than 10 academic journals and publications, including EIC of Smart Health, AEIC of IEEE Internet Computing Magazine. He is the founding steering committee chair of several conferences, including ACM/IEEE Symposium on Edge Computing (SEC), IEEE/ACM International Conference on Connected Health (CHASE), and IEEE International Conference on Mobility (MOST). He is a fellow of IEEE, a distinguished member of ACM.