2024

V2I-BASED ADAPTIVE COLLISION AVOIDANCE FOR SAFETY AND TRAFFIC EFFICIENCY

Vaishnavi Balambeed
Michigan Technological University, vhbalamb@mtu.edu

V2I-BASED ADAPTIVE COLLISION AVOIDANCE FOR SAFETY AND TRAFFIC EFFICIENCY

By

Vaishnavi Balambeed

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical and Computer Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2024

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Electrical and Computer Engineering.

Department of Electrical and Computer Engineering

Thesis Advisor:	*Aurenice Oliviera*
Committee Member:	*K. C. Dukka*
Committee Member:	*Anthony J. Pinar*
Department Chair:	*Jin W. Choi*

Table of Contents

List of Figures

List of Tables

Acknowledgements

This thesis is completed through the shared efforts and unwavering support of numerous individuals, whose invaluable contributions encompassed not only academic mentorship but also personal motivation and encouragement.

At the forefront, I extend my sincerest appreciation to my advisor, Dr. Aurenice Oliveira, for her steadfast support, enduring patience, and priceless wisdom. Her guidance was the compass navigating me through the challenges of research and composition, and her mentorship transcended the boundaries of academia. Mirroring the sacred verse ingrained in me from childhood — "Guru Brahma Guru Vishnu Guru Devo Maheshwara, Guru Sakshat Parambrahma Tasmai Shri Gurave Namah," which epitomizes the teacher as an embodiment of knowledge, discipline, and the catalyst of joyous triumph — Dr. Oliveira has personified this in every aspect of my growth over the past two years. For her dedication and unwavering belief in my potential, I am eternally grateful.

I also owe a debt of gratitude to the creators of "An Edge-Based Framework for Enhanced Road Safety of Connected Cars" and the msvan3t project collaborators. Their readiness to incorporate my thesis project code into their repository has generously enriched my research.

Special thanks are due to my committee members, Dr. KC Dukka and Dr. Tony Pinar, for their willingness to accept to serve as committee members for my defense and for providing valuable feedback to improve my thesis.

On a personal note, it is to my mother that I attribute my resolve and commitment. The embodiment of strength and integrity, she has sculpted the individual I have become. The sacrifices she has made do not go unrecognized; to her love and constant encouragement, I am profoundly obliged. This thesis is dedicated to her — a tribute to her ceaseless faith in me and the sacrifices she has made for her children.

Lastly, I express my genuine gratitude to my friends and colleagues for the respite they provided from the demands of thesis composition through their companionship and levity. They served as a reminder that balance is essential in all successful endeavors.

Definitions

1. **Vehicular Ad-Hoc Network (VANET):** A network that uses cars as mobile nodes in a MANET to create a mobile network. VANETs turn every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of each other to connect and, in turn, create a network with a wide range.
2. **Simulation of Urban MObility (SUMO):** An open-source traffic simulation package that allows the modeling of intermodal traffic systems including road vehicles, public transport, and pedestrians. It supports the simulation of large networks and various traffic demand models.
3. **Network Simulator 3 (NS3):** A discrete-event network simulator, widely used in academia and industry to simulate networking protocols and mechanisms for Internet systems.
4. **LTE-EPC Network Simulator (LENA):** A module within NS3 that supports the simulation of 4G LTE networks, including the Evolved Packet Core (EPC), providing tools to study the performance of LTE networks.
5. **Cooperative Awareness Message (CAM):** A message format defined in ITS (Intelligent Transport Systems) standards used for periodic broadcasting of vehicle status information (like position, speed, direction) to nearby vehicles for safety and traffic efficiency purposes.
6. **Decentralized Environmental Notification Message (DENM):** A notification system within ITS that allows vehicles to disseminate information about hazardous situations or events on the road to nearby vehicles and infrastructure.
7. **Vehicle-to-Vehicle (V2V):** A wireless communication system where vehicles share information about their speed, location, and heading with nearby vehicles to prevent accidents and improve traffic flow.
8. **Vehicle-to-Infrastructure (V2I):** A communication framework where vehicles exchange information with road infrastructure, such as traffic lights and sensors, to improve road safety and traffic efficiency.
9. **European Telecommunications Standards Institute (ETSI):** A non-profit organization that establishes standards for information and communication technologies within Europe, including standards for vehicular communications and ITS.
10. **Long-Term Evolution (LTE):** A standard for wireless broadband communication for mobile devices and data terminals, focusing on increasing the capacity and speed of wireless data networks.
11. **Ad hoc On-Demand Distance Vector (AODV):** A routing protocol for ad hoc mobile networks which is capable of both unicast and multicast routing. It is designed for use in networks where the network topology may change frequently.

12. **Road-Side Unit (RSU):** Infrastructure components in VANETs that facilitate wireless communication with vehicles passing by, supporting various applications from traffic management to safety warnings.
13. **M/G/1 Queue:** A single-server queueing model where arrivals follow a Poisson process, service times have a general distribution, and there is only one server, commonly used to model service systems.
14. **MX/G/1 Queue:** A generalization of the M/G/1 queue model that allows batch arrivals of units to be serviced, providing a more flexible framework for modeling complex arrival processes.
15. **Wireless Access in Vehicular Environments (WAVE):** A protocol standard designed to enable vehicular communication systems, including both V2V and V2I communications, aimed at improving road safety and traffic efficiency.
16. **On-Board Unit (OBU):** A device mounted in vehicles that enables communication with other vehicles and roadside infrastructure within a vehicular communication system.
17. **Global Positioning System (GPS):** A satellite-based navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth.
18. **Advanced Driver-Assistance Systems (ADAS):** Technologies that enhance vehicle safety and driving, including automated lighting, adaptive cruise control, automated braking, GPS/navigation, and more, by reducing human error.
19. **Enhanced Collision Avoidance (eCA):** An advanced system designed to prevent or reduce the severity of a collision by utilizing sensors and other technologies to detect imminent crashes and take appropriate preventive actions.
20. **Collision Avoidance Algorithm (CAA):** A set of algorithmic strategies implemented within collision avoidance systems to detect potential collisions and execute maneuvers to avoid them, often based on sensor input and predictive modeling.
21. **Collision Avoidance Strategy (CAS):** The tactical approach or methods used by collision avoidance systems to prevent accidents, including altering vehicle speed, direction, or both, based on real-time analysis of the vehicle's environment.
22. **Graphical User Interface (GUI):** A user interface that allows users to interact with electronic devices through graphical icons and visual indicators as opposed to text-based interfaces, typed command labels, or text navigation.

List of Abbreviations

1. **VANET** - Vehicular Ad-Hoc Network
2. **SUMO** - Simulation of Urban MObility
3. **NS3** - Network Simulator 3
4. **LENA** - LTE-EPC Network Simulator
5. **CAM** - Cooperative Awareness Message
6. **DENM** - Decentralized Environmental Notification Message
7. **V2V** - Vehicle-to-Vehicle
8. **V2I** - Vehicle-to-Infrastructure
9. **ETSI** - European Telecommunications Standards Institute
10. **LTE** - Long-Term Evolution
11. **AODV** - Ad hoc On-Demand Distance Vector
12. **RSU** - Road-Side Unit
13. **WAVE** - Wireless Access in Vehicular Environments
14. **OBU** - On-Board Unit
15. **GPS** - Global Positioning System
16. **ADAS** - Advanced Driver-Assistance Systems
17. **eCA** - Enhanced Collision Avoidance service
18. **CAA** - Collision Avoidance Algorithm
19. **CAS** - Collision Avoidance Strategy
20. **GUI** - Graphical User Interface

Abstract

To leverage the growing communication and connectivity among modern vehicles, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) systems are increasingly being used to implement active safety applications. However, current research often overlooks the impact of algorithms such as collision avoidance on traffic flow efficiency. This work investigates the adaptation of a collision avoidance algorithm implemented in V2I to incorporate a variable time headway and spacing control strategy. The proposed approach aims to maintain higher average speeds among vehicles, lower individual vehicle's waiting, and travel times in the vicinity of the infrastructural unit while simultaneously avoiding collisions; thereby enhancing both safety and traffic flow efficiency. A simulation framework based on the Network Simulator 3 (NS-3) and the Simulation of Urban Mobility (SUMO) traffic simulator is developed to evaluate the performance of this concept, enabling the study and analysis of the system's results. The results demonstrate the effectiveness of the adapted algorithm in striking a balance between collision avoidance and traffic flow optimization, paving the way for more comprehensive safety applications in connected vehicles.

1 Introduction

Advancements in connected mobility has been on the forefront of innovation and research for the past few years and has drastically accelerated the need to explore efficient technology as well as gathered attention among the researchers, policy makers and industry to largely make these systems increasingly reliable, safe, and less expensive. With an increase in this demand, there is also a rising requirement for these systems to be more holistically designed to optimize resources used to make the vehicles autonomous.

In the current automotive industry, two of the major areas of research are active safety components and vehicle connectivity of Infrastructure and Vehicular entities. While the safety components also form the basis for intelligence and efficiency in these vehicles, connectivity allows for these vehicles to be in the forefront of their environment's visualization and awareness. All modern vehicles in the current automotive scenario are being outlined with autonomous features with varying levels of advancements in their perception capabilities, connectivity and driving capabilities. These automation features have been classified by Society of Automotive Engineers [1] and are adopted worldwide.

To better understand the progression and capabilities of modern vehicles, it is essential to define the different levels of automation. The levels, as outlined by the National Highway Traffic Safety Administration (NHTSA)[2], provide a framework for categorizing the technological advancements and functionalities of autonomous vehicles at various stages of development. By clearly defining these levels, we can better understand the importance of integrating Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications into modern vehicles with varying levels of vehicular perception and driving capabilities such as Levels 2 and 3, where the human driver still plays a significant role in the driving task. The NHTSA [2] classifies the different levels of autonomous vehicles, based on the extent of human driver's necessity and system autonomation capabilities.
These different levels are:

- Level 0 (No Automation): The human driver performs all driving tasks without any system assistance.
- Level 1 (Driver Assistance): The vehicle can control either steering or acceleration/deceleration, but not both simultaneously. The human driver performs all other tasks.
- Level 2 (Partial Automation): The vehicle has combined automated functions like acceleration and steering, but the human must remain engaged with the driving task and always monitor the environment.
- Level 3 (Conditional Automation): The vehicle can perform all aspects of the driving task under certain conditions. The human driver must be ready to take back control when the system requests.

- Level 4 (High Automation): The vehicle can perform all driving tasks and monitor the environment in certain conditions. Human interaction is not required in those conditions.
- Level 5 (Full Automation): The vehicle can perform all driving tasks under all conditions. No human intervention is required.

These levels define the gradual shift from manual to fully autonomous driving, highlighting the technological advancements and functionalities at each stage. From basic assistance features to complete driving automation, these levels illustrate the vehicle's capability to take over driving tasks, the necessity for human drivers to monitor and intervene, and the eventual elimination of human intervention in driving.

As the levels of automation increase, the vehicle's ability to perform driving tasks independently becomes more sophisticated, reducing the need for human intervention. This progression also highlights the necessity of developing robust safety applications and connectivity solutions to ensure the reliable and efficient operation of autonomous vehicles at higher levels of automation.

Integrating V2V and V2Icommunications into autonomous vehicles at Levels 2 and 3 is a crucial step towards enhancing road safety and reducing the occurrence of crashes. This approach significantly advances the capabilities of autonomous vehicles, allowing them to communicate key information such as speed, direction, and positional data in real-time, with each other and other infrastructural entities. In the subsequent section, we explore the merits of V2V and V2I communication in autonomous driving by defining their architecture and introducing their role in implementing active safety algorithms that help in preventing fatal crashes or collisions within traffic ecosystem.

1.2 V2V and V2I in autonomous driving

Connected vehicles are vehicles equipped with advanced technologies that enable them to communicate and exchange information wirelessly with other vehicles (V2V), infrastructure (V2I), and other road users (V2X). This communication capability allows connected vehicles to share critical data, such as their location, speed, heading, and other relevant parameters, in real-time.

The concept of connected vehicles relies on the use of wireless technologies that operate within a range of approximately 300 meters or more. These devices enable vehicles to constantly monitor and exchange data with nearby vehicles and infrastructure, creating a dynamic, rapidly moving environment where potential hazards can be identified and communicated to drivers or autonomous systems.

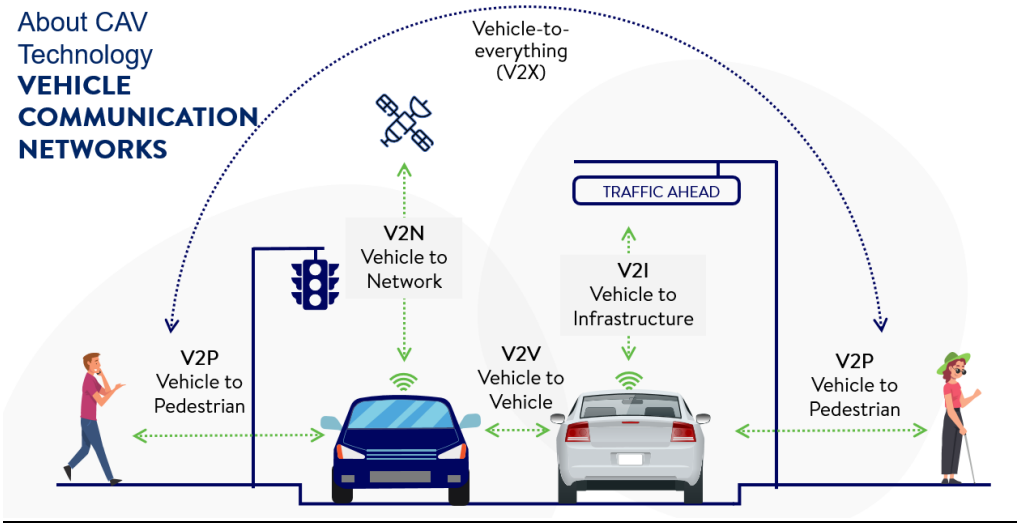

Figure 1 Connected Vehicles in Traffic Ecosystem [3].

Figure 1 illustrates the connected vehicle technology in an ecosystem with all the possible entities that a connected vehicle can establish communication with. The figure illustrates Vehicle to Vehicle, Vehicle – to – infrastructure, Vehicle-to-pedestrian, and Vehicle-to-Network communications. Vehicle-to-Vehicle (V2V) communication is a wireless technology that allows vehicles to share real-time information with each other, creating a network of connected vehicles on the road. V2V systems utilize wireless technologies (such as 5G/6G) to exchange critical data, such as a vehicle's speed, position, heading, and brake status, with other nearby vehicles. This information exchange occurs regardless of whether the vehicles are in direct line of sight or obscured by obstacles like buildings or other vehicles. By enabling vehicles to communicate with each other, V2V technology forms the foundation for the development of advanced safety features, such as collision avoidance systems, which can alert drivers about potential hazards and help prevent accidents. V2V communication has the potential to significantly reduce crashes, injuries, and fatalities on our roads by providing drivers with a more comprehensive understanding of their surroundings and allowing them to make informed decisions.

Vehicle-to-Infrastructure (V2I) communication, on the other hand, refers to the wireless exchange of information between vehicles and roadside infrastructure elements, such as traffic lights, road signs, and roadside units (RSUs). V2I technology enables vehicles to receive real-time data about traffic conditions, road hazards, construction zones, and weather updates from the infrastructure. This information can be used to optimize traffic flow, minimize congestion, and improve overall road safety. For instance, V2I communication can facilitate intelligent traffic signal coordination, where traffic lights adapt their timing based on real-time vehicle data to reduce stops and delays. Furthermore, V2I systems can warn drivers about upcoming road work, accidents, or other incidents, giving them ample time to take alternative routes or adjust their driving accordingly. By establishing a bidirectional communication channel between vehicles

and infrastructure, V2I technology has the potential to revolutionize the way we manage and operate our transportation systems, making them smarter, safer, and more efficient.

Merging Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications into autonomous vehicles at Levels 2 and 3, is a critical step towards enhancing road safety and reducing the occurrence of crashes. Such interconnectedness is pivotal for the development of advanced safety applications that can preemptively identify potential hazards and mitigate risks effectively. As detailed within USDOT's definition of Connected Vehicle [4] the primary application of V2V and V2I communication is to enable safety applications that could potentially address approximately 75 percent of crashes involving all vehicle types, underscoring the vast potential of this technology in improving roadway safety. The necessity of a publicly supported infrastructure for the successful implementation and security of V2V and V2I systems is emphasized, alongside the evaluation of benefits derived from infrastructure-based communications used to support V2I applications.

The current research and implementation efforts by the connected vehicle's research community are intended to Transition from DSRC (Dedicated Short-Range Communications) at 5.9 GHz to more modern communication technologies like WiFi 6 and 5G - LTE offers a promising avenue for supporting active safety applications in connected vehicles[5]. This shift acknowledges the evolving landscape of vehicular communication, where the integration of WiFi 6 and 5G-LTE could offer enhanced data transmission rates, broader coverage, and more reliable connectivity. Such advancements are crucial for ensuring that vehicles can seamlessly communicate key information, including speed, direction, and positional data, in real-time. To facilitate a smooth transition and preserve the investments in existing DSRC-equipped vehicles, it is essential to develop interoperability solutions. These solutions would enable backward compatibility, allowing formerly DSRC equipped vehicles to interact effectively with those using WiFi- 6 or 5G- LTE for V2V and V2I (communications.

The research outlined by the USDOT's Connected Vehicle study in [4] includes several major research tracks aimed at accelerating the deployment of V2V and V2I-based safety systems. These tracks include the development of crash scenario frameworks, ensuring interoperability across all vehicles makes and models, assessing the benefits of V2V and V2I applications, and addressing driver and policy issues related to the implementation of these technologies. Each track is designed to contribute to the overall goal of establishing an effective framework for V2V and V2I communication, enhancing crash avoidance capabilities, and ensuring the safety and efficiency of the transportation system.

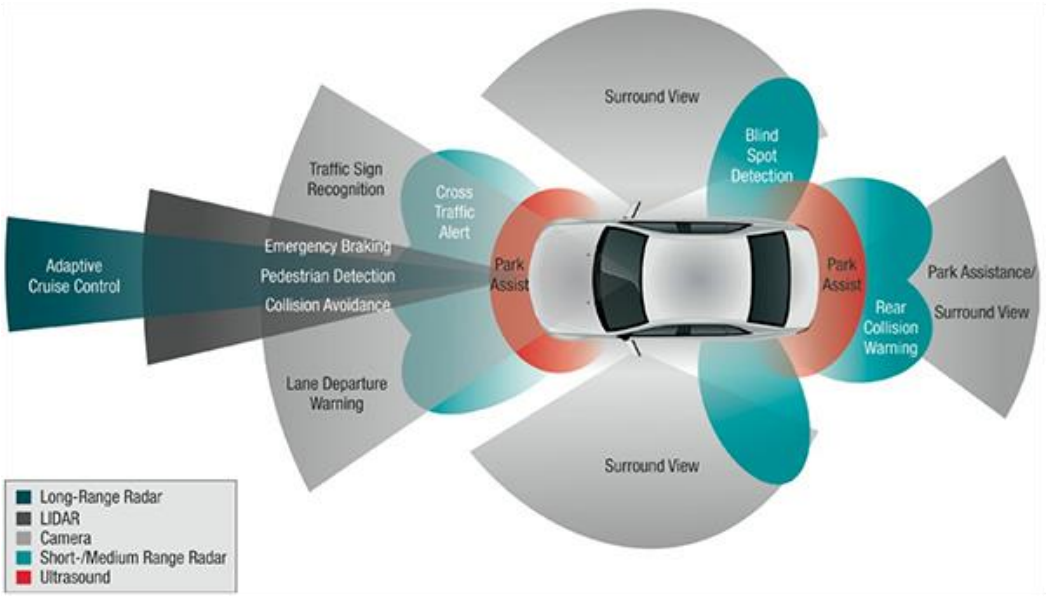

Figure 2 In-Vehicle Sensor systems in Autonomous Vehicles [6].

The in-vehicle sensor systems in the autonomous vehicles that assist the vehicles in perception are mostly equipped with sensors such as Radar, Camera, Lidar, and ultra-sonic sensors such as illustrated in Figure 2. The sensors illustrated in this figure have limited viewing range (within 300-500 m) around the vehicle. Thus, their capability of gauging and adapting the vehicle's surrounding with its changing environment is limited. These sensors also have a predefined field of view, beyond which their perception ability is diminished. The perception capabilities of these vehicles are also largely affected by blind spots which are beyond the restricted field of view of these sensors and the obstacles which stunt their field of view by a great degree. These shortcomings can be expensive on the vehicle's safety aspect since perception forms an integral part of the implementation of safety algorithms that prevent the vehicle from crashing or getting into any fatal incidents.

V2V and V2I technologies have a relatively superior field of operation as compared to these sensors. Their communication capabilities are beyond the physical limitations since the wireless technology can operate independent of field of view, weather dependencies or obstacle in the path that vehicle is traversing. Therefore, the integration of V2V and V2I communication into autonomous driving levels 2 and 3 will significantly enhance the vehicles' ability to avoid collisions and crashes when combined with advanced warning systems and automated responses to potential hazards. This includes applications such as Emergency Stop Lamp Warning, Forward Collision Warning, Intersection Movement Assist, Blind Spot and Lane Change Warning, Do Not Pass Warning, and Control Loss Warning, each designed to address specific scenarios that could lead to accidents[7].

By providing vehicles with the capability to communicate with each other and with infrastructure, V2V and V2I technology complements other vehicle-based safety technologies that include the sensor-based systems as illustrated in Figure 2, creating a more comprehensive safety net for drivers and passengers alike. The ongoing research and development efforts by NHTSA and other stakeholders are crucial in furthering the adoption of connected vehicles, making roads safer for everyone. This comprehensive approach to enhancing road safety through connected vehicle technologies also illustrates the importance of collaboration between government agencies, industry stakeholders, and the research community. It is a testament to the potential of V2V and V2I communications in revolutionizing transportation safety and efficiency, paving the way for a future where autonomous vehicles can navigate roads with minimal human intervention and maximum safety. Thus, moving away from the solely depending on isolated in-vehicle sensor systems that are wholly responsible for supporting the autonomous functions is beneficial.

The current landscape of creating autonomous or semi-autonomous vehicles heavily relies on collision avoidance and detection algorithms situated within the vehicle's onboard units. An OBU (on-board unit) is an electronic device that functions alongside the electronic control unit installed in a vehicle. The OBU is responsible for recording the traffic and driving data, and to connect to roadside and satellite navigation systems. These OBUs are equipped to communicate with other OBUs of various vehicles through broadcasting safety messages for sharing their ground truth driving data such as speed, position or heading. The onboard units can also draw upon data from various sensors, including radar, lidar, and cameras, which are pivotal in enabling driving aids such as adaptive cruise control and automatic emergency braking. While the sensor-based systems offer high accuracy in numerous situations, each sensor type presents its own set of drawbacks. To mitigate these shortcomings ,such as limited line-of-sight range, inadequate night vision, and object detection precision, vehicles often employ a combination of sensors. This compensatory approach, however, demands substantial processing power from the onboard units.

Integrating V2V or V2I communication strategies into Advanced Driver-Assistance Systems (ADAS) not only conserves processing time by distributing computational tasks to the control units present in infrastructural sites or networks (in case of Vehicle-to-Network communication) but also allows for more refined development and implementation owing to the accuracy of real time data. This addition in system design aims to increase accuracy in detection and diminish latency in vehicle’s reaction to hazard by informing the vehicle ahead of time about the possible hazard, thus presenting a significant improvement in the implementation of the safety algorithms when integrated alongside traditional sensor-only dependent model. Therefore, transitioning towards integration of communication-based approaches facilitates a more interconnected and cooperative vehicular environment, which can adapt more dynamically to real-world conditions and enhance the overall safety and efficiency of autonomous vehicle operations.

1.3 Active Safety Algorithms

Active safety components are pivotal for autonomous driving, serving as the foundational layer for ensuring vehicle safety and operational integrity in a world increasingly reliant on self-driving technology. As outlined by European Automobile Manufacturers Association's research [8] active safety systems like Anti-lock Braking Systems (ABS) and Electronic Stability Control (ESC) offer real-time preventive measures to avoid accidents by enhancing vehicle control during critical moments. The advent of more sophisticated systems, such as Autonomous Emergency Braking (AEB) and Lane Keeping Assistance (LKA), represents a significant leap forward. These technologies, powered by on-board sensors, radar, cameras, GPS, and lasers, enable vehicles to autonomously detect and respond to potential hazards with little to no human intervention.

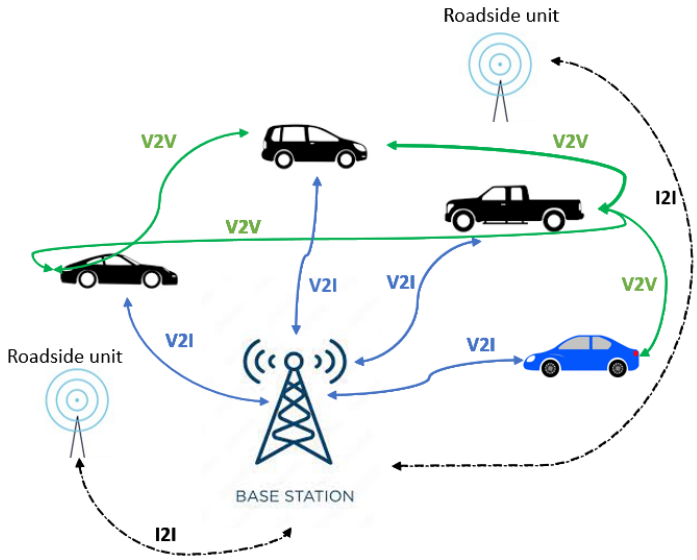

Figure 3 Vehicular Communication Networks (source: author).

Incorporating V2V and V2I communication into active safety systems further amplifies their effectiveness. V2V and V2I technologies facilitate seamless communication between vehicles and their surroundings, extending the range of perception beyond the direct line of sight provided by conventional sensors. This networked communication layer allows autonomous vehicles to anticipate and respond to dynamic road conditions, traffic patterns, and potential threats with enhanced precision, thereby significantly reducing the likelihood of collisions and improving road safety for all users.

Collision avoidance is the linchpin of active safety technology, crucial for mitigating accidents and safeguarding lives on the road. As described in [8], collision avoidance systems actively intervene before an accident occurs, utilizing technologies such as AEB, LDW, and LKA to detect imminent threats and take corrective action. These systems are integral to preventing accidents by automatically braking or providing steering assistance

to avoid potential collisions, thereby playing a vital role in reducing the incidence and severity of accidents.

The integration of V2V and V2I communication enhances collision avoidance capabilities by enabling vehicles to share and receive critical information about their environment beyond their line of sights. This interconnected approach allows vehicles to make informed decisions based on a comprehensive understanding of the traffic scenario, including hazards that are beyond the immediate field of view of on-board sensors. By facilitating a cooperative driving environment, V2V and V2I technologies augment the effectiveness of collision avoidance systems, making autonomous vehicles safer and more reliable.

The importance of collision avoidance in enhancing road safety cannot be overstated. Active safety systems equipped with collision avoidance technologies represent a proactive approach to accident prevention, as highlighted in [8]. By identifying and addressing potential hazards before they escalate into accidents, these systems significantly reduce the risk of collisions, thereby protecting not only the occupants of the vehicle but also other road users.

Moreover, the implementation of V2V and V2I communication in collision avoidance strategies elevates the potential for accident prevention to new heights. This networked approach ensures that vehicles are not only aware of their immediate surroundings but also of the broader traffic environment, including upcoming hazards communicated by other vehicles and infrastructure. Consequently, V2V and V2I technologies enable a more anticipatory form of driving, where vehicles can adapt their behavior based on real-time traffic information, thus significantly enhancing the efficacy of collision avoidance systems, and advancing the vision of a safer, accident-free future on our roads.

This Master research work focuses on implementation of collision avoidance algorithms using V2I infrastructure. The Edge-based enhanced Collision Avoidance (enhanced Collision Avoidance) services entailed in the research work [9] forms the basis of this research implementation. The crux of this work it to leverage edge computing in V2V and V2I using enhanced collision avoidance service in a multi-stack Vehicular Ad-Hoc Network framework. Thus, enabling the use cases of Collision Avoidance Assist (CAAs)[10] and Collision Avoidance Strategy (CAS)[9] to experiment with different control strategy explore their efficiency in both active safety and traffic flow efficacy.

The primary basis is to recognize vehicles within a given road infrastructure where the vehicles are identified as connected vehicles and to evaluate which of them are in a probable collision course, monitored specifically at the intersections, then mitigating their potential collision and consequences by mainly monitoring and improvising the control factors: Time to Collision and Space to Collision. The control theory is extended to utilize the variable time headway and spacing strategy used in [11] which entails the advantage that using a variable spacing control strategy has over a constant one.

Applying the Variable Time Headway and Spacing Control Strategy extends the works [9] and [10] to not only be applied as a collision avoidance algorithm but also to enable a possibility of traffic flow efficiency in the given road infrastructure.

This thesis is structured into six chapters, each designed to progressively delve into the intricacies of utilizing V2Vand V2I communications for enhancing road safety through active safety applications, with a focus on collision avoidance and traffic flow efficiency. In this first chapter we introduced the realm of autonomous vehicles, highlighting the necessity for V2V and V2I communications for active safety applications, emphasizing their pivotal role in advancing collision avoidance capabilities.

Chapter 2 lays out the overall framework, spotlighting the significance of the system architecture of Enhanced Collision Avoidance's edge-based service using V2I. It defines key terminologies and explores the Variable Time Headway strategy, providing a foundation for understanding the sophisticated mechanisms behind the proposed safety enhancements.

Chapter 3 presents a comprehensive literature review encompassing V2V and V2I communications, braking strategies, traffic flow theory, and the variables of time-to-collision (T2C) and space-to-collision (S2C). This chapter aims to encapsulate the breadth of research relevant to the thesis objectives, offering insights into existing knowledge and identifying gaps that the current work seeks to address. Chapter 4 details the methodology and implementation of the collision avoidance assist and collision avoidance strategy. It delves into the Variable Time Headway implementation and the integration of this architecture with a multi-stack, ETSI-compliant V2X framework for ns-3 known as msvan3t, demonstrating the application of these concepts in a simulated environment.

Chapter 5 focuses on simulation and results, employing tools such as SUMO [12] and NS-3 LENA LTE[13] to conduct experiments . This chapter aims to showcase improved outcomes from the foundational paper "An Edge-Based Framework for Enhanced Road Safety of Connected Cars[9]," critically analyzing the data to underscore the efficacy of the implemented collision avoidance strategies.

Chapter 6 discusses the conclusions drawn from evaluating collision avoidance system and its applications. We also discuss how the novel introduction of traffic monitoring factors have affected the traffic flow efficiency. This final chapter reflects on the findings, discussing their significance in the broader context of road safety and intelligent transportation systems. It also outlines potential avenues for future research, indicating how this work can serve as a steppingstone for further advancements in the field.

Through this structured approach, this work aims to contribute meaningful insights and advancements to the domain of autonomous driving and connected vehicle technologies, focusing on the critical aspect of enhancing road safety through innovative V2I applications.

2 Framework and Terminologies

Vehicle to Infrastructure communication system utilizes communication technologies to facilitate active, real time and secure periodic communication between vehicles and roadside units that are defined as infrastructure. Thus, essentially forming Vehicular Ad-Hoc Networks (VANETS) that facilitate the implementation of traffic flow regulation and safety by monitoring the traffic ecosystem continuously.

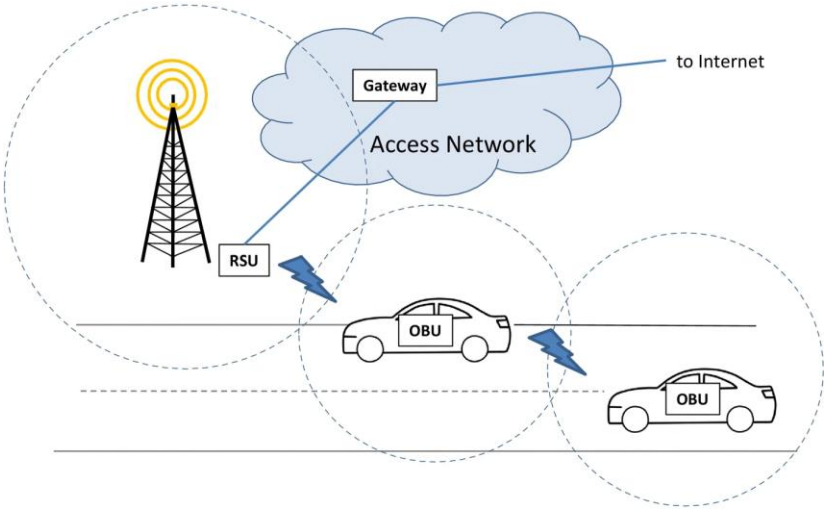

Figure 4 VANET architecture [14].

An example of VANET , as illustrated in Figure 4, which is a type of network that enables wireless communication between vehicles (V2V), as well as between vehicles and roadside infrastructure (V2I). The main goal of VANETs is to enhance road safety, improve traffic efficiency, and provide various services to road users. In a VANET, each vehicle is equipped with an On-Board Unit (OBU) that allows it to send, receive, and relay messages to other vehicles or Roadside Units (RSUs). This communication is typically based on short-range wireless technologies such as DSRC and long-range communication such as 5G-LTE and Wi-Fi – 6G.

The structure and architecture of a VANET consists of several key components that enable communication between vehicles and infrastructure as shown in Figure 4.
On-Board Unit (OBU) is a device installed in a vehicle that enables it to communicate with other vehicles and RSUs. It consists of a processor, memory, and a wireless communication module (e.g., IEEE 802.11p, LTE, or 5G). The OBU is responsible for sending and receiving messages, processing data, and running various applications. It is also connected to the vehicle's sensors, which provide information about the vehicle's state, such as speed, position, and direction.
Roadside Unit (RSU) is a stationary device deployed along the roadside that acts as an access point for vehicles to communicate with the infrastructure. RSUs are typically equipped with a processor, memory, and multiple wireless communication modules (e.g., IEEE 802.11p, LTE, or LTE-5G) to support different types of communication.

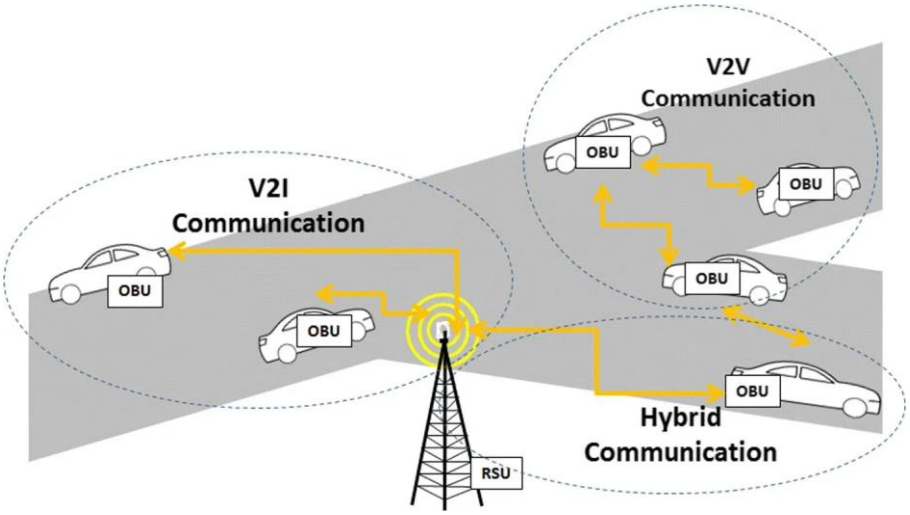

Figure 5 VANET communication architecture [14].

The three main types of VANET communication systems are Vehicle-to-Vehicle, Vehicle-to-Infrastructure and a hybrid approach that entails both the former mentioned systems.

In Vehicle-to-Vehicle (V2V) communication, vehicles can communicate directly with each other using wireless connectivity. This allows vehicles to exchange safety messages, traffic information, or cooperate in various applications. Vehicle-to-Infrastructure (V2I) communication involves the exchange of data between vehicles and RSUs. Vehicles can send information to RSUs, which can then relay it to other vehicles or to a central traffic management system. RSUs can also broadcast messages to vehicles in their vicinity or defined radius of operation, providing them with relevant traffic or safety information. The hybrid approach combines both V2V and V2I communication. This allows for a more flexible and robust communication network, where vehicles can communicate with each other directly and with the infrastructure when necessary.

For the wireless communication within Intelligent Traffic System (ITS) the possibility of holistic connectivity and networking capabilities have seen the light in recent years. The VANETs now allow for V2V, V2I and V2X (Vehicle to Everything) technology implementation. In all cases, they support active safety and traffic management functions.

The safety applications rely significantly on short-message broadcasts in a vehicle's environment, event- driven short and long message transmission from infrastructure (Roadside Units) to the vehicles. The traffic efficiency monitoring applications utilize the infrastructure capabilities to measure, monitor and regulate the traffic density, flow, and mobility of specific entities such as those of emergency utilities, to enhance the Road's traffic ecosystem, thus furthering its connectivity and intelligence.

The safety and traffic applications previously cited require the following components for their effective operation:

- **Wireless Connectivity** that can support the communication primitives which comprise within the spatial and temporal scope of the ITS.
- **Messaging protocols** that allow information to be transmitted to and from the vehicles and infrastructural units, which can also be packaged and facilitated to be sent securely without a large overhead on the communication systems.
- **Traffic Monitoring and Safety Systems deployed on the Vehicle's On Board units (OBUs) and infrastructural units** which allow the active safety and traffic flow efficiency monitoring algorithms to be implemented, as explained in the context of VANETs before.

The wireless connectivity that we have implemented in the V2I system entailed in this research work focuses on the usage of 5G-LTE communication technology. The Messaging protocols are European Telecommunications Standard Institute (ETSI) complaint broadcasting messages called Cooperative Awareness Messages(CAM) and event triggered messages called Decentralized Environmental Notification Message(DENM) which carry information between the vehicles and infrastructural units [15]. Lastly, the applications implemented in this research work are part of the system called "Collision Avoidance system with variable time headway and spacing control strategy". This system comprises of applications that interact in a server-client fashion to for an edge-based computing network where the client application are installed within OBUs of vehicles and the server applications are installed in the infrastructural roadside units. This chapter entails the first three aspects discussed above, along with their scope, requirements, definitions of the concepts and techniques utilized, their limitations. This chapter aims to describe in detail the motivation and ideology behind the utilization of the framework used in the accomplishment of this research work.

2.1 Wireless Connectivity

The enhanced Collision Avoidance service is defined in [9] as a service that aims to mitigate vehicular collisions at the road intersection, through vehicular connectivity provided by wireless communications. Essentially, enhanced Collision Avoidance utilizes V2I based network architecture and leverage edge-computing in which the edge is defined within the infrastructural unit to monitor a predetermined area of action. To apply Collision Avoidance Algorithm (CAA) and Collision Avoidance Strategy (CAS) on the vehicles in a given area of action, a communication is established between vehicles and infrastructural road side units, using periodic broadcasting messaging protocols and event-driven specific messaging protocols.

This work is based on the use of LTE (Long Term Evolution) networking facilitating V2V and V2I connectivity. This -briefly explain LTE's distinctive advantages, it's developmental trajectory that allows for it to be ideally used in enhanced Collision Avoidance for V2I connectivity [7]; as well as the IEEE 802.11 standard. Thus delineating the implementation of enhanced Collision Avoidance using both these technologies.

IEEE 802.11p, part of the Wireless Access in Vehicular Environments (WAVE) standard, facilitates V2V and V2I communications in an ad-hoc manner. While it allows for easy deployment and direct V2V communications without infrastructure, its performance suffers in high-density scenarios due to scalability issues and cannot guarantee Quality of Service (QoS)[16]. Its limited radio range and reliance on roadside units for V2I communications result in fragmented connectivity, which can be inadequate for the needs of dynamic vehicular networks.

LTE presents a significant leap forward in supporting vehicular communications. Unlike IEEE 802.11p, which has limitations in scalability, unbounded delays, and lacks deterministic QoS guarantees, LTE offers high data rates, low latency, and extensive coverage. LTE's all-IP flat architecture simplifies network deployment, ensuring efficient, real-time data, voice, and signaling transmissions critical for delay-sensitive vehicular applications. The centralized management of radio resources by eNodeBs which are identified as base stations [5] in LTE facilitates improve QoS, satisfying the stringent requirements of vehicular applications by offering reliable and consistent communication over wide areas. This comprehensive coverage is vital for V2I communications, ensuring continuous connectivity even at high speeds, thereby addressing the intermittent connectivity challenges posed by IEEE 802.11p.

The transition to LTE represents an improvement in vehicular networking by addressing the limitations of previous technologies. LTE's support for high mobility, coupled with its wide area coverage, solves the problem of poor, intermittent V2I connectivity seen in IEEE 802.11p networks. With higher penetration rates anticipated due to the integration of LTE interfaces in common devices like smartphones, LTE is poised to achieve a broader adoption in vehicular environments. Its capacity to handle high downlink and uplink speeds supports a larger number of vehicles per cell than IEEE 802.11p, making it a more scalable solution for vehicular communications. Moreover, LTE's multicast and broadcast capabilities through Multimedia Broadcast Multicast Services (eMBMS) [17] enhance the efficiency of transmitting data to multiple users, further solidifying its role in advanced vehicular applications.

Enhanced Collision Avoidance systems can operate effectively on both LTE and IEEE 802.11, leveraging the edge-based framework for optimal performance. The framework we are discussing in this thesis utilizes the strengths of both technologies, employing LTE for its wide coverage and reliable QoS for V2I communications, while using IEEE 802.11p for direct V2V interactions in proximity scenarios. Since this thesis focuses on V2I based collision avoidance system and algorithms that support V2I communication for detecting and eliminating risks of intersectional collisions, we are utilizing LTE wireless technology for the implementation of this study with the network framework of NS-3’s LENA module. Furthermore, the edge-based service architecture integrates seamlessly with LTE's infrastructure, ensuring low-latency communication essential for real-time safety applications.

By integrating LTE's broad coverage in V2I communication and reliable data transmission with IEEE 802.11p's capability for direct V2V exchanges(if integrated with

this framework), the edge-based framework for collision avoidance algorithm ensures comprehensive coverage and responsiveness. This dual-technology approach facilitates real-time data sharing among vehicles and infrastructure.

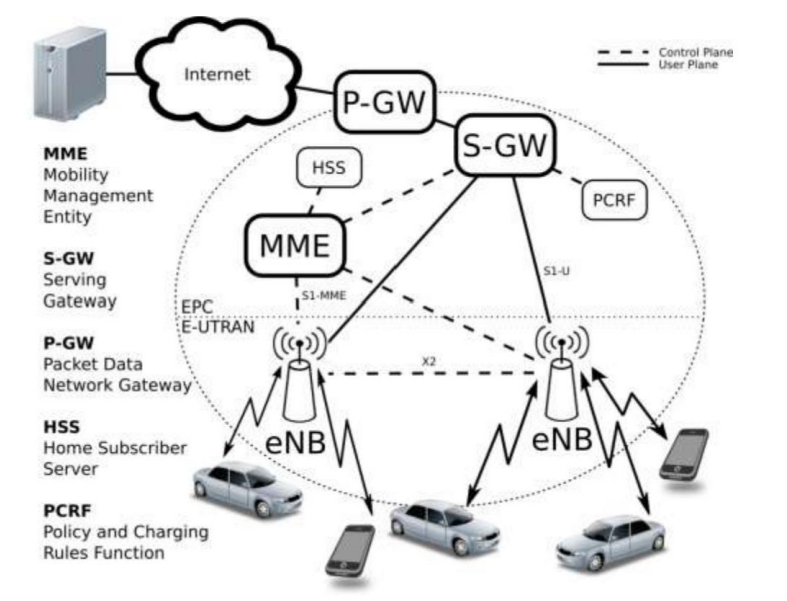

Figure 6 LTE architecture: access network and core network (EPC) entities [5].

Figure 6 illustrates the LTE architecture that we are leveraging for this study which consists of the access network and core network (EPC) entities. The LTE system is characterized by a flat all-IP architecture with a reduced number of network devices. IP-based data, voice, and signaling transmissions allow for greater deployment feasibility and extendibility compared to previous cellular networks. The access network comprises eNodeBs, which manage radio resources and handover events, while the core network consists of the MME (responsible for control procedures), S-GW (responsible for routing and data forwarding), and P-GW (the outgoing entity for communication with IP and circuit-switched networks). This architecture enables LTE to provide low latency and efficient support for vehicular communications [14].

Following the exploration of wireless technologies in vehicular communications, we now delve into the messaging protocols critical to the operation of Intelligent Transportation Systems (ITS), particularly focusing on Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs). These protocols are fundamental to V2V and V2Icommunications, enabling a wide array of safety and efficiency applications within vehicular networks. The roles and communication processes of CAMs and DENMs highlighting their distinct functionalities and contributions to ITS are outlined in [7].

2.2 Messaging Protocols: CAMs and DENMs

Cooperative Awareness Messages (CAMs) are periodically broadcast messages intended to provide continuous situational awareness within vehicular networks. CAMs contain dynamic information about the vehicle, such as its position, speed, heading, and acceleration, facilitating real-time awareness among nearby vehicles and infrastructure. These messages are critical for maintaining an updated view of the vehicular environment, enabling effective V2V and V2I interactions essential for collision avoidance and efficient traffic management.

Decentralized Environmental Notification Messages (DENMs), on the other hand, are event-triggered messages designed to inform vehicles and infrastructure about specific events or hazards on the road, such as accidents, roadworks, or slippery surfaces. Unlike CAMs, which are broadcasted periodically, DENMs are generated and disseminated in response to incidents, offering timely and targeted warnings to potentially affected vehicles, thereby enhancing road safety and aiding in emergency response efforts.

The primary distinction between CAMs and DENMs lies in their trigger mechanisms and content specificity. While CAMs are exchanged periodically to maintain a constant flow of situational data among vehicles and infrastructure, DENMs are disseminated in response to specific events, providing focused alerts and warnings about unusual or hazardous conditions. This fundamental difference underscores the complementary roles of CAMs and DENMs in ITS, with CAMs ensuring ongoing vehicular awareness and DENMs facilitating acute responses to emergent situations.

CAMs enable vehicles and infrastructure to continuously update and share their status, creating a dynamic map of the vehicular environment that supports a wide range of safety and efficiency applications. DENMs complement this by offering a mechanism to rapidly disseminate critical information about road hazards or emergencies, allowing vehicles and drivers to take proactive measures to avoid potential dangers.

The communication process for CAMs and DENMs in LTE networks involves several key steps, as detailed in [18]. CAMs are transmitted from vehicles to infrastructure nodes (e.g., eNodeBs in LTE networks), where they are processed and then distributed to relevant vehicles within the vicinity. This ensures that vehicles receive timely updates about their immediate environment. DENMs follow a similar path but are triggered by specific events. Upon detection of an event warranting a DENM, the message is sent to a backend server, which then processes and redistributes the message to concerned vehicles in the affected geographical area. Both types of messages can utilize unicast for uplink transmissions, with the option of unicast or broadcast modes on the downlink, leveraging LTE's Multimedia Broadcast Multicast Services (MBMS) for efficient distribution.

2.3 Application: Collision Avoidance System

The architecture of the collision avoidance algorithm that is implemented in this thesis work is based on the architecture that is designed for enhanced Collision Avoidance in [9], illustrated in Figure 7, which consists of several key components that work together to enable effective collision avoidance algorithm.

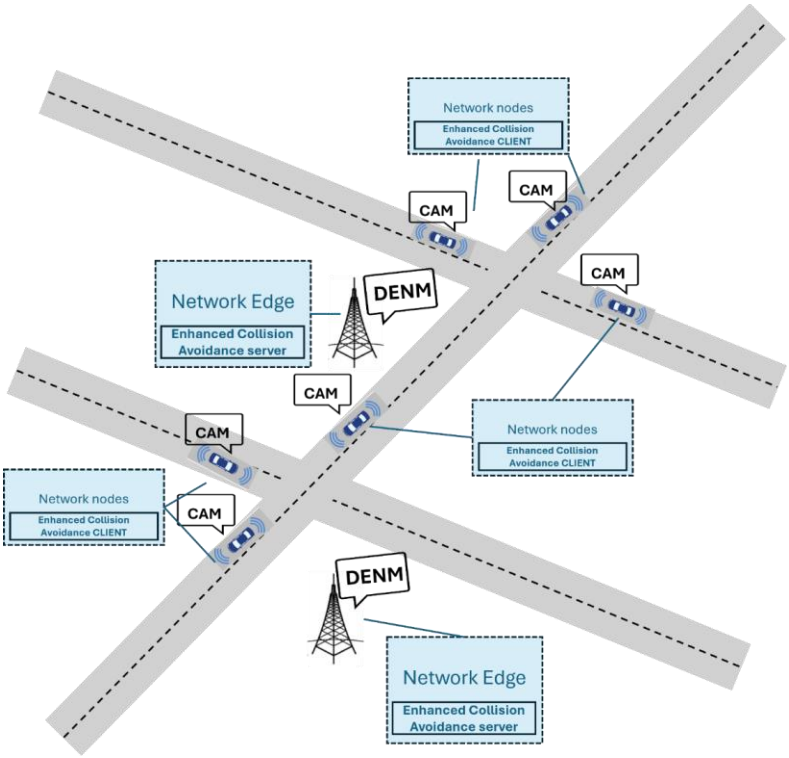

Figure 7 Collision Avoidance System Architecture (source: author).

In the eCA architecture, the "Network Edge" represents the edge computing infrastructure, which includes the application "Enhanced Collision Avoidance server." This server is responsible for processing data and executing the collision avoidance algorithms. The "Network nodes" represent the vehicles equipped with on-board units (OBUs) that enable communication with the infrastructure and other vehicles. These nodes periodically transmit Cooperative Awareness Messages (CAMs) containing information about their status, such as position, speed, and heading, to the "Enhanced Collision Avoidance CLIENT" component within the OBU.

The "Enhanced Collision Avoidance CLIENT" forwards the received CAMs to the "Enhanced Collision Avoidance server" at the network edge. The server utilizes the Collision Avoidance Assist (CAA) algorithm to process the received CAMs and project future trajectories of the vehicles. The CAA algorithm introduced in [10] uses the information contained in the CAMs, such as position, speed, heading, and vehicle size, to predict potential collision risks.

The CAA algorithm employs different trajectory prediction methods based on the status of the vehicle's turn signals. If the turn signals are off, the algorithm projects the future position using a straight-line trajectory. However, if the turn signals are active, indicating an intended turn, the algorithm projects the trajectory using multiple segments to account for the curvature of the intersection. By analyzing these projected trajectories, the CAA algorithm identifies pairs of vehicles that are on a potential collision course.

Once the CAA algorithm detects a potential collision, the Collision Avoidance Strategy (CAS) application of the "Enhanced Collision Avoidance server" comes into play. The CAS utilizes the information provided by the CAA algorithm to make decisions on which vehicle should yield and which vehicle should have the right of way. These decisions are based on factors such as the time to collision (TTC) and the space to collision (STC), which consider the speeds and sizes of the vehicles involved. These two factors are discussed in detail in the coming sections and their implementation along with their implications are discussed in fourth chapter.

The CAS generates Decentralized Environmental Notification Messages (DENMs) to communicate its decisions to the vehicles. These DENMs contain a "precedence" field that indicates whether a vehicle should yield or proceed. If the precedence is set to 1, the vehicle is granted the right of way and can continue its path through the intersection. On the other hand, if the precedence is set to 0, the vehicle is instructed to yield or stop.

The "Enhanced Collision Avoidance CLIENT" application in the vehicle's OBU receives the DENMs and interprets the precedence field. Based on the received instructions, the OBU communicates with the vehicle's control units to execute the appropriate actions, such as applying brakes or allowing the vehicle to proceed.

By leveraging the edge-based architecture and the collaborative exchange of CAMs and DENMs, the eCA service enables real-time collision detection and avoidance at road intersections. The CAA algorithm's trajectory prediction and the CAS's decision-making capabilities work together to enhance road safety and prevent potential collisions.

2.4 Traffic flow regulation: Variable Time Headway and Spacing Control Strategy

In this thesis, we aim to enhance the existing work in [9] by exploring the concept of variable time headway (VTH) and a dynamic spacing control algorithm applied to the vehicular nodes in a V2I setup so as to evaluate and enhance the impact of safety algorithm such as Collision Avoidance System on traffic flow efficiency and vehicle safety. The spacing strategy selection is crucial as it determines the safety gaps used as inputs to the control algorithms for applications such as collision avoidance and cooperative adaptive cruise control (CACC).

The traditional constant time headway (CTH) strategy, where the time gap between vehicles remains fixed, has limitations in adapting to complex traffic conditions and ensuring string stability [19]. In contrast, the VTH strategy proposed in [11] allows the time headway to vary based on factors such as vehicle speed, relative speed, and

preceding vehicle acceleration. By considering these additional parameters, the VTH strategy can better adapt to changing traffic conditions and improve overall traffic flow and safety.

In [11], Chen et al. propose an improved VTH spacing strategy that considers the following factors:
1. Vehicle speed: The time headway is adjusted based on the speed of the considered vehicle, allowing for larger gaps at higher speeds to ensure safety.
2. Relative speed: The time headway is reduced when the relative speed between the considered vehicle and the preceding vehicle increases, helping to minimize transient errors and maintain smaller gaps.
3. Preceding vehicle acceleration: The time headway is modified based on the acceleration of the preceding vehicle, providing a larger gap when the preceding vehicle is decelerating to avoid rear-end collisions.
4. Traffic flow parameters: The jam density and free-flow speed are incorporated into the VTH strategy to adapt the time headway to the overall traffic conditions.

By incorporating these factors, the improved VTH strategy aims to strike a balance between traffic efficiency and vehicle safety. The authors also propose a saturation function to limit the time headway within a specific range, ensuring that the gap remains within acceptable boundaries.

In this research, we apply the improved VTH strategy to the enhanced Collision Avoidance (eCA) system in a server-client application manner. The collision avoidance system leverages V2I communication and edge computing to monitor a predetermined area of action and apply collision avoidance algorithm (CAA) [10] and collision avoidance strategy (CAS) [9] algorithms to the vehicles within that area.

By integrating the VTH strategy into the eCA system, we aim to demonstrate its effectiveness in improving safety and traffic management under various speeds and vehicle densities. The server-client architecture allows for centralized decision-making and coordination among vehicles, enabling the implementation of the VTH strategy in a cooperative manner. Through simulations and experimental results, we evaluate the impact of the VTH strategy on key performance metrics such as the vehicles' travel times, waiting times, average speeds, and collision avoidance.

By comparing the results with the collision avoidance system presented in [9], we aim to validate the benefits of the improved VTH strategy in enhancing traffic flow efficiency while ensuring vehicle safety. The findings of this research contribute to the understanding of spacing strategies and their role in the development of advanced driver assistance systems (ADAS) and Cooperative Intelligent Transportation Systems (C-ITS).

3 Literature Review

The previous chapters introduced the field of connected vehicle technologies and the significant progress witnessed in recent years, enabling the development of innovative safety applications that leverage vehicle-to-everything (V2X) communication to enhance road safety and traffic efficiency. With this progress and advancement, there has been a significant rise of interest that researchers from various backgrounds such as civil engineering, computing, data science, electrical and mechanical engineering have demonstrated in furthering the development of innovative technologies in this field of vehicular application. In this chapter, we will discuss various research works that have formed the basis of conducting and implementing the collision avoidance system that is discussed further in this master's thesis. We will also compare the parallel research works to gain knowledge on the direction in which this research is carried out.

The enhanced Collision Avoidance (eCA) service presented in [3], which utilizes the cellular network to exchange information between vehicles and infrastructure to prevent collisions at intersections, stands out as a prominent example of V2I-based safety regulation applications developed in recent research. The eCA service is deployed at the edge of the network to reduce communication latency, a concept that aligns with the approach taken in this research work. By deploying the eCA service at the edge of the network, the communication latency is reduced as the processing responsibilities are shifted closer to the vehicles, thus minimizing the distance and time required for data transmission between the vehicles and the infrastructure. This edge computing approach significantly enhances the responsiveness of the system by decreasing the communication latency between vehicles. Several research studies investigate these type of applications, including the use of Multi-access Edge Computing (MEC) platforms for supporting safety services, such as vehicle collision avoidance [20],and the use of LTE for vehicular applications [18],[16]. These studies provided a solid foundation for the work presented in [9] and highlighted the importance of collision avoidance as an automotive application as detailed in [10].

While the work in [9] focuses on deploying the Collision Avoidance Algorithm (CAA) in different scenarios and studying collision avoidance involving vulnerable users, the research work presented in this thesis aims to extend the functionality of the collision avoidance algorithm to improve traffic flow efficiency. The key concepts discussed in [3], such as the eCA service architecture, consisting of a Collision Avoidance Algorithm (CAA) for predicting vehicle trajectories and detecting potential collisions, and a Collision Avoidance Strategy (CAS) for deciding which vehicle should yield in case of a potential collision, serve as a starting point for the enhancements proposed in this thesis.

The methodology employed in [9] involves developing the eCA application in NS-3 over the LENA framework [13] and managing vehicle mobility using SUMO [12] through the TraCI interface. While this approach ensures realistic mobility patterns and dynamics, this thesis takes a step further by implementing the enhanced collision avoidance algorithm in the multi-stack simulation framework for vehicular applications testing [21].

This framework allows for a more comprehensive evaluation of the proposed enhancements, considering various performance metrics and scenarios.

The main conclusions of [9] highlight the effectiveness of the eCA service in reducing road accidents at crossroad intersections and increasing the average vehicle speed compared to traffic-light regulated intersections. However, the authors acknowledge that the performance of the eCA service in terms of avoided collisions depends on vehicle speed and density, with higher speeds and densities leading to a lower percentage of avoided collisions. In contrast, this thesis aims to address these limitations by enhancing the collision avoidance algorithm to adapt to varying traffic conditions and optimize traffic flow efficiency.

Furthermore, while Malinverno et al. [9] compare the performance of the Single-Event CAS (SE-CAS) and Multiple-Event CAS (ME-CAS) strategies. The ME-CAS strategy outperforms the SE-CAS strategy in terms of avoided collisions and downlink traffic load, as it has a wider perception of the intersection status by considering multiple collision events simultaneously. V2X technologies are key enablers for efficient vehicle traffic management, as demonstrated by the improved average speed when using the Collision Avoidance service compared to traffic-light regulated intersections. Malinverno et al. highlight the importance of their open-source simulation framework, which closely mimics real-world conditions and dynamics, and can be used by other researchers to enhance the application or develop new road-safety applications. They also emphasize the flexibility of their framework, which allows for the investigation and selection of strategies to avoid vehicle collisions and increase road safety at intersections. This thesis focuses on developing a more advanced strategy that considers not only collision avoidance but also traffic flow optimization. By incorporating traffic flow efficiency as a key objective, the proposed enhanced collision avoidance algorithm aims to outperform the strategies presented in [9].

3.1 Open-Source Simulation Framework: MS-VAN3T

The work in [21] by Malinverno et al. presents an open-source simulation framework for testing V2X applications using NS-3 and SUMO. The framework supports different communication protocols (IEEE 802.11p, 3GPP C-V2X, and 3GPP LTE) and provides tools for setting up basic vehicular communication scenarios. While the focus of the MSVAN3T is not specifically on collision avoidance, it discusses parallel research studies comparing 802.11p and C-V2X in terms of performance and maturity as well as other related works such as OpenC2X.

The key concepts discussed in Malinverno's work [21] include the composition of the simulation framework, which consists of SUMO v1.6.0 [12] for mobility modeling and ns-3 v3.29 [13] for network simulation. The framework supports V2I/V2N communication models using LTE (with LENA [13]) and 802.11p (with WAVE), as well as V2V communication models using C-V2X Mode 4 and 802.11p. Additionally, the framework implements ETSI-compliant features of the Facilities layer, including the CA and DEN Basic Services for handling CAM and DENM messages [15], [22].

The architecture of the simulation framework is designed to be modular and extensible, allowing users to easily configure and switch between different communication stacks and network configurations. SUMO provides a graphical user interface (SUMO-GUI) for visualizing and interacting with simulations, while NS-3 integrates several modules to support different communication technologies. The interaction between SUMO and NS-3 is managed through the TraCI interface, enabling bidirectional coupling between the two simulators.

The methodology used in the implementation of the multi stack vehicular framework involves combining SUMO for mobility modeling and NS-3 for network simulation. V2I/V2N communication is modeled using LTE with the LENA framework and 802.11p with the WAVE model, while V2V communication is modeled using C-V2X Mode 4 and 802.11p with the WAVE stack. ETSI-compliant message encoding and decoding are implemented using the Asn1c tool [23]. The performance of the sample applications is evaluated using metrics such as the number of collisions and the average speed of emergency vehicles.

The main conclusions of [21] highlight the flexibility and potential of the open-source simulation framework for testing V2X applications. The framework includes state-of-the-art models for V2X communications and standard-compliant vehicular message dissemination stacks (ETSI CA and DEN Basic Services) [15]. The sample applications demonstrate the framework's ability to evaluate V2X applications, with the area speed advisory application significantly reducing the number of collisions at intersections and the emergency vehicle alert application allowing emergency vehicles to maintain a higher average speed. The authors emphasize the framework's interaction with SUMO-GUI and its integration with up-to-date NS-3 models, making it a solid base for V2X application testing and validation. Future work discussed in the research work [21] is planned to focus on developing security modules and other parts of the ETSI stack, such as GeoNetworking and the Basic Transport Protocol (BTP).

The Enhanced Collision Avoidance Algorithm service [9] mainly focuses on the control strategy to monitor and avoid collisions at intersections of the road infrastructure by manipulating the vehicles' movements and actions based on a dynamic Time to Collision and Space to Collision parameters. Owing to a higher risk of collisions at the intersections as demonstrated in [16], the research carried out in this thesis work aligns with the aim of enhanced collision avoidance algorithm in [9], [10], [24] to mitigate the collisions at the intersections. The time to collision and space to collision parameters form the foundation of decision-making factors, enabling safety algorithms like Collision Avoidance to significantly influence the overall speed of the traffic fleet, which in turn affects traffic flow management efficiency in and around these intersections. Studying traffic fleet speed efficiency allows for the analysis of road occupancy, which will become a crucial aspect of the Vehicle to Infrastructure (V2I) communication ecosystem in the coming years.

To enhance the control strategy in enhanced Collision Avoidance service, this thesis research work introduces a variable time headway and spacing control strategy that

facilitates the monitoring of traffic flow efficiency while simultaneously preventing collisions at intersections within the given traffic ecosystem. This ecosystem comprises infrastructural units housing the roadside units and vehicles equipped with onboard units.

The following section reviews relevant works on time headway and spacing strategies used in vehicle control, particularly for cruise control applications. It is also crucial to discuss the formal definitions of time headway, spacing control, occupancy maps, and related concepts to establish a clear understanding of these key elements.

3.2 Variable Time Headway and Spacing Control Strategy

In the context of adaptive cruise control (ACC) systems, the spacing strategy plays a crucial role in allocating vehicle spacing according to the driving environment while ensuring safety and improving road utilization. Traditional fixed-spacing strategies, as discussed by Zhou et al. (2005) [25], have limitations in adapting to complex driving situations and achieving multi-target vehicle control, as pointed out by Swaroop et al. (1999) [26]. To address these shortcomings, variable spacing strategies have gained prominence, with the time headway-based approach being the most extensively studied.

Time headway, as defined by [27], refers to the time interval between the ends of the heads of two consecutive vehicles passing through a section in the queue of vehicles in the same lane. Various researchers have proposed different time headway models, such as the constant time headway model by Swaroop et al (2017) [26], the headway model related to the vehicle's own speed by Broqua et al. (1991) [28], and the headway model considering the relative speed of two cars by Yanakiev et al. (1998) [29].

In [30], Jiang et al. present a variable time headway (VTH) strategy based on Luo's model. The proposed time headway function is described as

$$th = sat\{th0 \pm k_{v1}v_r 2 - k_{v0}v_r - k_{v1}\alpha_1\} \quad \text{Eq (3.1)}$$

where t_h is the time headway, t_{h0} is the basic time headway, k_{v1}, k_{v0}, and ka are parameters relevant to the time headway, v_r is the relative speed of the two vehicles, α_l is the acceleration of the leading vehicle, and $sat(\cdot)$ is a saturation function is applied over this equation that limits the time headway within a predefined range [$t_h min$, $t_h max$].

The desired spacing between two vehicles is expressed as

$$d_{des} = t_h \cdot v_f + d_0 \quad \text{Eq (3.2)}$$

where d_{des} is the desired safe distance, v_f is the speed of the following vehicle, and d_0 is the minimum safe distance between the two vehicles when they come to a halt. However, this research doesn't account the traffic fleet dynamics or spatial density for evaluating time headway and spacing control strategy, we only borrow from it the dynamics designed to evaluated desired spacing control utilizing the time headway strategy.

Jiang et al. (2019) [30] prove the convergence of spacing error and relative velocity using mathematical analysis. The spacing error (ε) is defined as the difference between the actual distance (d) and the desired distance (d_{des}), and its dynamics are derived for two cases: when the leading vehicle's speed is greater than the following vehicle's speed ($vl > v_f$) and when the leading vehicle's speed is less than the following vehicle's speed ($v_l < v_f$). Under the convergence condition ($\alpha_l = 0$), the spacing error dynamics can be simplified to,

$$\dot{\varepsilon} = -(m/n)\varepsilon \qquad \text{Eq (3.3)}$$

Equation 3.3 proves that the spacing error converges to zero in both cases.

The effectiveness of the improved VTH strategy is demonstrated through simulation results comparing it with Luo's VTH strategy under five classic scenarios: car following, cut in, cut out, approaching, and hard braking. The improved VTH proposed in [30] strategy exhibits better performance in terms of smaller speed fluctuations, better stability in maintaining the desired distance, faster adaptation to changing environments, and smoother driving experiences. These results highlight the potential of the improved VTH strategy in enhancing traffic flow efficiency and road occupancy management, which are essential aspects of the Vehicle to Infrastructure (V2I) communication ecosystem.

To further enhance our research on V2I-based enhanced Collision Avoidance presented in [9] and extend its capabilities beyond safety applications, we draw inspiration from the work titled "Effects of ACC and CACC vehicles on traffic flow based on an improved variable time headway spacing strategy [11].” This study introduces a variable time headway and variable spacing control strategy specifically designed for vehicles equipped with Cooperative Adaptive Cruise Control (CACC). CACC is an advanced form of Adaptive Cruise Control (ACC) that leverages V2V and V2I communication technologies to enable closer and more coordinated vehicle spacing, ultimately improving traffic flow management.

By applying the principles of variable time headway and spacing control strategy from [11] to our V2I-based enhanced Collision Avoidance system, we aim to transform it from a purely safety-focused application into a comprehensive solution that allows for the study, monitoring, and optimization of traffic flow within a fleet operating in a defined infrastructure-controlled area. This integration will enable our enhanced Collision Avoidance system to dynamically adjust the desired spacing between vehicles based on real-time traffic conditions, vehicle speeds, and relative positions, resulting in a more efficient and safer traffic flow while minimizing the risk of collisions.

The research work in [11] focuses on the impact of Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC) vehicles on traffic flow using an improved variable time headway spacing strategy. The motivation behind the research [11] is to enhance traffic flow efficiency and safety by leveraging advanced vehicle control systems and communication technologies. The authors in [11] build upon previous

works, such as the constant time headway spacing strategy proposed in [26] and the variable time headway strategy introduced in [29].

The new variable time headway strategy [11] is particularly relevant for CACC, as it highlights the importance of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in improving traffic flow. CACC extends the capabilities of ACC by allowing vehicles to exchange information with each other and with the infrastructure, enabling closer and more coordinated vehicle spacing. The work in [31] extends the constant time headway theory to CACC along with the multi-anticipated CACC control algorithm which takes into consideration the speed difference of multiple vehicles leading the Ego Vehicle.

Furthermore, the research and results produced by Amoozadeh et al [32] entails three modes of operation which are speed control, gap control and collision avoidance mode. These are implemented within multi-scene mixed traffic simulations on VENTOS which is a vehicle network open simulator, like SUMO. In the work presented in [33] Yan-Yan et al propose a macroscopic heterogeneous traffic flow model considering multiple vehicle classes, each with homogeneous car-following model.

The research [33] also introduces concepts of road capacity split and perceived equivalent density to model lateral and longitudinal cross-class interactions while establishing a practical interaction rule based on the allocation of vehicles rather than complex analogies. The main motivation is to develop a generalized and parsimonious multi-class flow model that is data driven, computationally efficient and able to reproduce traffic phenomena. Thus, the methodology presented in [27] has form the basis for formulating and adapting the concepts of Time headway and variable spacing strategies in collision avoidance to study the traffic flow in a more empirical manner. It allows this thesis research work to be extended to study the traffic flow in a macroscopic manner capturing the key traffic phenomena in a large network.

The authors in [11] propose an improved variable time headway spacing strategy for CACC vehicles , which is described by the following equations:

1. Desired time headway:

$$t_h = C_k * \frac{1}{\left(Rho_{jam} * (V_{free} - V_f)\right)} - C_v * (V_p - V_f) - C_a * a_p \qquad \text{Eq. (3.4)}$$

2. Desired spacing:

$$\Delta x_{des} = t_h * v_f + \Delta x_0 \qquad \text{Eq. (3.5)}$$

Where t_h is the time headway, v_f is the velocity of the Ego vehicle for which the time headway is being calculated, C_k, C_v and C_a are positive coefficients. a_p is the acceleration of the preceding or leading vehicle (the vehicle in front of the Ego vehicle). Rho_{jam} is the traffic jam density and V_{free} is the free allowable maximum speed. The main idea behind this strategy is that the time headway decreases as the relative speed increases, allowing the platoon to maintain a smaller spacing compared to the Constant Time Headway (CTH) strategy [26]. This approach helps to reduce transient errors and improve the overall efficiency of the platoon.

To address these limitations, an improved VTH strategy is proposed in Equation (1) by the authors in [11] . This strategy incorporates the speed change trend of the preceding vehicle, specifically the influence of its acceleration (a_p) on the desired safety gap (Δx_{des}). Moreover, the improved strategy considers the traffic jam density (ρ_{jam}) and the free flow speed (v_{free}) to better adapt to different traffic conditions. The desired safety gap is calculated as a function of the time headway (t_h) and a constant minimum spacing (Δx_0). The improved VTH strategy is particularly beneficial for connected and autonomous vehicles (CACC) that utilize wireless communication. These vehicles can leverage their shorter response times to further reduce the headway while maintaining safety. To account for this, the coefficient ck can be appropriately reduced for CACC vehicles compared to traditional ACC (Adaptive Cruise Control) vehicles.

In summary, the improved VTH strategy builds upon the original VTH strategy by incorporating additional factors such as the acceleration of the preceding vehicle, jam density, and free flow speed. By considering these parameters and adapting the time headway accordingly, the improved strategy aims to enhance the performance, efficiency, and safety of automatic platoons, especially in the context of connected and autonomous vehicles.

We now delve into the aspect of braking reaction time, which is a critical parameter in our collision avoidance algorithm for evaluating safety distance and spacing control strategy in autonomous driving systems. In [9], the authors emphasize that the reaction time of an autonomous driving system is determined by two key factors: frame rate and processing latency. It is emphasized that to provide better safety, autonomous driving systems should be able to react faster than human drivers, suggesting that the processing latency for traffic conditions should be within 100 ms.

Furthermore, Lin et al. [34] discuss the reaction time in the context of automatic braking systems, stating that a reaction time of 0.05 s accounts for the interval from the moment a Decentralized Environmental Notification Message (DENM) is received at the application layer to the instant when the vehicle's actuator starts the deceleration. This value aligns with the relevant literature [34], [35], and is the value we consider in our thesis algorithm for evaluating safety distance and spacing control strategy.

Johansson and Rumar [36] investigate drivers' brake reaction times and report that the fastest possible action by a human driver takes 100–150 ms. This finding supports the notion that autonomous driving systems should aim for a reaction time faster than human

drivers to enhance safety. The research in [35] is a result of study conducted on driver reaction time in crash avoidance research, validating a driving simulator study on a test track. They report a perception-reaction time of 1.28 s for drivers on a track, with an additional 0.3 s movement time, resulting in a total reaction time of 1.58 s. These studies provide valuable insights into human driver reaction times and serve as a benchmark for autonomous driving systems to surpass.

By incorporating the braking reaction time of 0.05 s, as discussed in [9] and supported by the literature [34], [35], [36] , our collision avoidance algorithm aims to optimize safety distance and spacing control strategy in autonomous driving systems. This approach ensures that the system can react quickly to dynamic traffic conditions and maintain a safe distance from surrounding vehicles, ultimately enhancing overall road safety.

By applying the concepts and equations from research work described in [11] to our research on V2I-based enhanced Collision Avoidance (enhanced Collision Avoidance), we can extend the capabilities of enhanced Collision Avoidance beyond safety and towards improved traffic flow management. Incorporating the variable time headway spacing strategy into enhanced Collision Avoidance can allow the system to dynamically adjust the desired spacing between vehicles based on real-time traffic conditions, vehicle speeds, and relative positions. This can result in a more efficient and safer traffic flow, as vehicles can maintain optimal spacing while minimizing the risk of collisions.

Furthermore, by leveraging the V2I communication capabilities of enhanced Collision Avoidance, the system can collect and analyze data from the vehicles and infrastructure to monitor traffic flow patterns and make informed decisions on spacing strategies. This can enable a more proactive approach to traffic management, where the enhanced Collision Avoidance system can anticipate potential congestion or safety issues and adjust the spacing strategies accordingly.

In conclusion, integrating the variable time headway spacing strategy [5] into the V2I-based Collision Avoidance research can significantly enhance the system's ability to improve traffic flow efficiency and safety. By extending enhanced Collision Avoidance's capabilities beyond collision avoidance and towards active traffic management, we can create a more comprehensive and effective solution for the challenges faced in modern transportation systems.

4 Methodology

The preceding chapter provided a comprehensive literature review of the concepts and system architecture employed in this thesis project, which focuses on the development of an enhanced Collision Avoidance system with variable time headway (VTH) and spacing control strategies. The implementation of the enhanced Collision Avoidance system was carried out using the NS3 and SUMO simulators. In this chapter, we delve deeper into the research methodology by presenting a detailed quantitative analysis of the theory, concept analysis, control equation studied, their implementation, and software configurations and tools that were used to generate the results discussed in the subsequent chapter.

The structure of this chapter is designed to provide a logical and systematic flow of information, beginning with a clear statement of the problem and the formulation of the research question that underpins this thesis. Subsequently, we establish the fundamental knowledge of the enhanced Collision Avoidance implementation and its network topology, focusing on the Vehicle-to-Infrastructure (V2I) driver client-based code implementations within the NS3 simulator. We also explore the multi-stack simulator that encompasses the enhanced Collision Avoidance system implementation, providing a comprehensive understanding of the simulation environment.

Next, we examine the control equations of the VTH and spacing strategy, discussing their implementation within the enhanced Collision Avoidance system. We justify the coding flow implemented by providing numerical evidence and detailed explanations, ensuring a thorough understanding of the system's inner workings. This section forms a crucial part of the methodology, as it highlights the logical aspects of the Collision Avoidance system and its integration with the VTH and spacing control strategies.
Moving forward, we discuss the quality and quantity of the data simulated by the setup described in the previous chapter, which focuses on the simulation environment. This discussion provides insights into the robustness and reliability of the data used for analysis and validation of the enhanced Collision Avoidance system's performance.

We present the experiments conducted to establish a relationship between collision avoidance and traffic flow on a macroscopic level, providing a holistic view of the system's impact on the overall traffic dynamics. This section showcases the innovative approach taken in this thesis, integrating RL techniques with the enhanced Collision Avoidance system to optimize its performance and adaptability.
Finally, we conclude the chapter by summarizing the basis of the study and the evidence provided to validate the research conducted, tying together the various aspects of the methodology, highlighting the significance of each component in addressing the research question and achieving the objectives of the thesis.

By presenting a well-structured and detailed methodology, this chapter aims to provide a solid foundation for the analysis and interpretation of the results in the forthcoming chapter. The in-depth explanations and justifications of the methods employed demonstrate the rigor and validity of the research, ensuring that the findings are reliable

and contribute meaningfully to the field of intelligent transportation systems and collision avoidance.

4.1 Establishing the Research Questions

The crux of the enhanced Collision Avoidance presented in [3] system lies on its ability to gather information from vehicles or nodes and detect probable collisions at road intersections. While enhanced Collision Avoidance has shown promise in reducing collision risks, there is a lack of research and evidence on its ability to optimize and monitor traffic flow, especially in adapting to different traffic conditions and maintaining performance with increasing or decreasing traffic density.

The main reason that this feature is needed to enhance the collision avoidance system is because safety applications such as collision avoidance system or emergency braking systems affect the individual vehicle's average speeds through their trip, thus also affecting the traffic fleet's ability to maintain a traffic flow efficiency within minimum congestions, collisions and maximizing the road utilization.

This research aims to address this gap by integrating variable time headway (VTH) and spacing control strategies within the enhanced Collision Avoidance system particularly using the Single Event Collision avoidance strategy (SE-CAS). By simulating enhanced Collision Avoidance within the multi-stack framework for data collection, this study seeks to enhance the system's performance and adaptability in optimizing traffic flow efficiency alongside its primary function of collision avoidance.

The proposed research question effectively encapsulates the identified gap and the novel approach taken to address it:

"How can the integration of variable time headway and spacing control strategies, implemented within the multi-stack framework for data collection, enhance the performance and adaptability of the Collision Avoidance system in optimizing traffic flow as well as reducing collision risks in vehicular networks?"

This question highlights the limitations of the current V2I based collision avoidance system as in [9] for optimizing and monitoring traffic flow and presents the integration of VTH, spacing control strategies and data collection methods through the multi-stack framework. By investigating this research question, -our study aims to contribute to the development of a more effective and adaptive enhanced Collision Avoidance system that not only reduces collision risks but also optimizes traffic flow in diverse vehicular network scenarios.

4.2 Collision Avoidance System with Variable Time headway And Spacing Strategy - Working:

As described in the system architecture explained in Chapter 2, the Collision Avoidance system considers road topology with intersections and is independent of the fact that these road intersections are regulated or unregulated by Intelligent Traffic lights. Reviewing the architecture before diving into the working of enhanced Collision Avoidance, it is essential to establish that the roadside units (RSUs) are equipped with processing power and bandwidth to provide connectivity over a defined geographical area. This area is subject to the network connectivity limitations of the RSU. In this work, we consider this to be a radius of 300 m around the RSU.

RSUs are equipped to ensure that data connectivity and communication occurs with the passing by vehicles. These vehicles are also considered to be carrying on board units (OBUs). Thus, all the vehicles considered in this discussion and that are designed within the simulation environment for this thesis are assumed to be carrying their on-board units that help to achieve the communication of data to and from the server/ base stations. These vehicles are also assumed to be equipped with automatic braking modules which have a degenerative braking time of 0.05s (as borrowed from the study in [26], [27]).

The Collision Avoidance system is an application that is installed at the roadside units in the road infrastructure topology. It is developed to communicate and work in a client-server fashion, where the vehicles are the clients, and the server is the one that is located within Road Side Units(RSUs). Thus, forming an edge-computing network where the server acts as the edge and the clients act as the distributed nodes that are all connected with the server through the wireless technology discussed before. The architecture that we have developed can also be extended for the distributed nodes to interact with each other, thus essentially forming vehicle to vehicle communication.

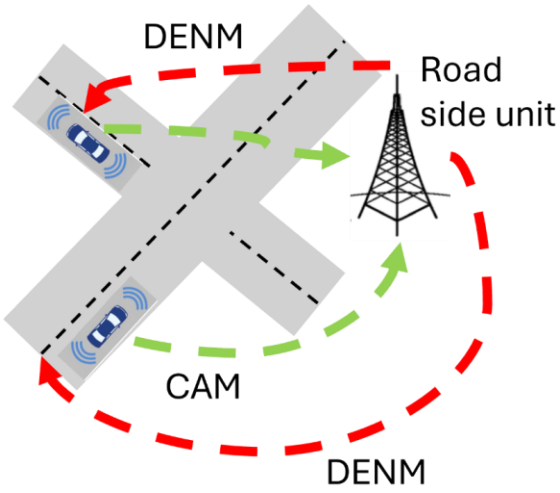

Figure 8 Edge Computing V2I architecture (source: author).

Figure 8 illustrates the edge computing architecture in which the application planted at RSUs receives the broadcasted Cooperative Awareness Messages from vehicular nodes, and it sends event triggered DENMs to the vehicles as a warning to avoid collisions.

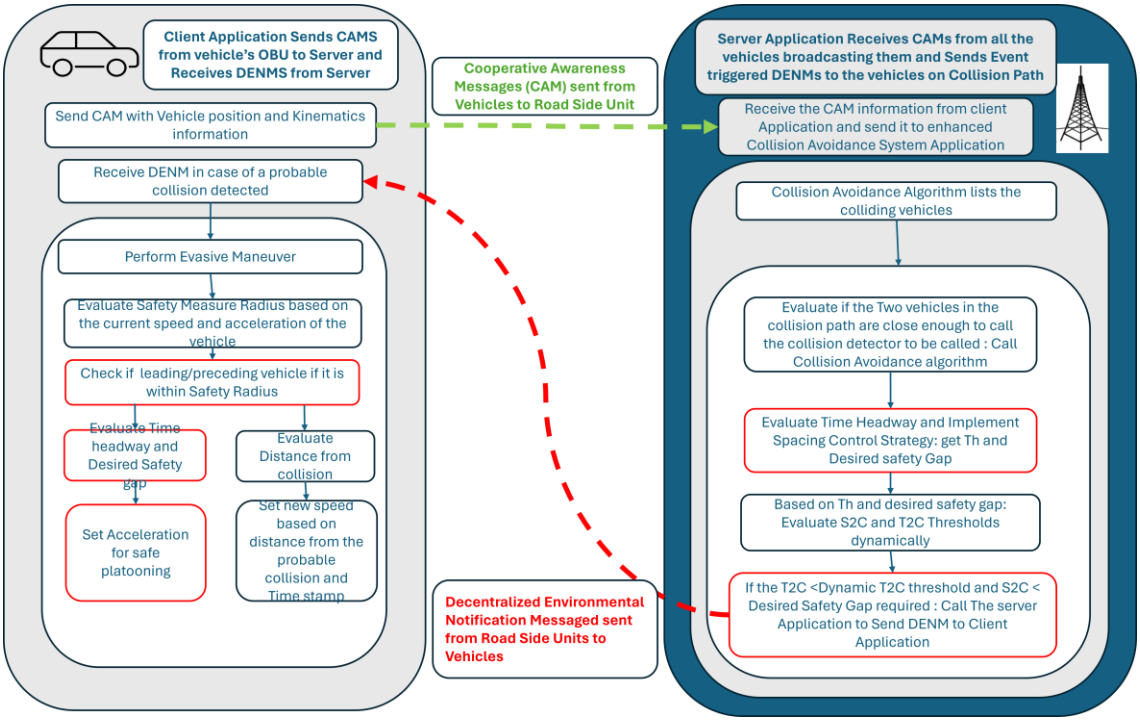

Figure 9 Logical analysis of Collision Avoidance algorithm: Server and Client Application (source: author).

Figure 9 shows the overall Logical analysis of working of Collision Avoidance system and its communication with Client Application. In [9], the authors consider that such an application is located on a server that can be reached out through a cellular network by the base station, and the base station receives the vehicular information first to pass it on to the server which is located at a remote location or cloud-based entity. However, in this thesis, we consider that this application is deployed on the Roadside units which has defined network coverage area of 1km to accommodate the vehicles' information around them. These detection zones can be explicitly deployed within this application and are subject to the network connectivity and its coverage requirements.

The Collision Avoidance system that is implemented is built upon the enhanced Collision avoidance service and Collision avoidance algorithms introduced in [9] and [10] respectively. The crux of their study and implementation is for enabling these algorithms to extract the information from the vehicular nodes to predict the vehicles' future position and their driving trajectories. They use a combination of inbuilt data tables and Cooperative Information Management Database tools to store these data. In the Collision Avoidance application deployed on the server application in our case, we utilize only the

dynamically and periodically updated tables as the newer vehicles enter the detection zone and leave the zone through the running of their travel times.

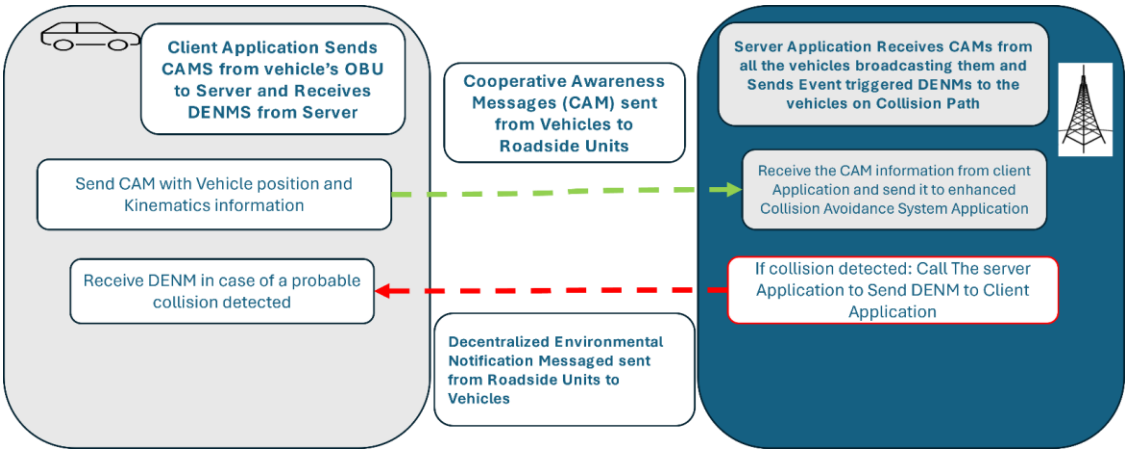

Figure 10 Client - Server communication through CAMs and DENMs (source: author).

Figure 10 illustrates the client and server communication that takes place in the application layer. The server application receives the CAM message from the client application housed in Vehicle's OBU. The server application then pushes these CAM messages to the Collision Avoidance algorithm implemented in the server application. The server application doesn't receive the CAM messaged from the vehicles synchronously, and thus there is very little probability that the CAM messages being received from two or more vehicles must be processed by the server application simultaneously or at the same instance. The CAMs are sent to the Collision Avoidance application in the order that they are received by the server application. Thus, the server application plays a vital role in scheduling and sending these CAM messages to the enhanced Collision Avoidance server using the timestamp of when they are received from the vehicles.

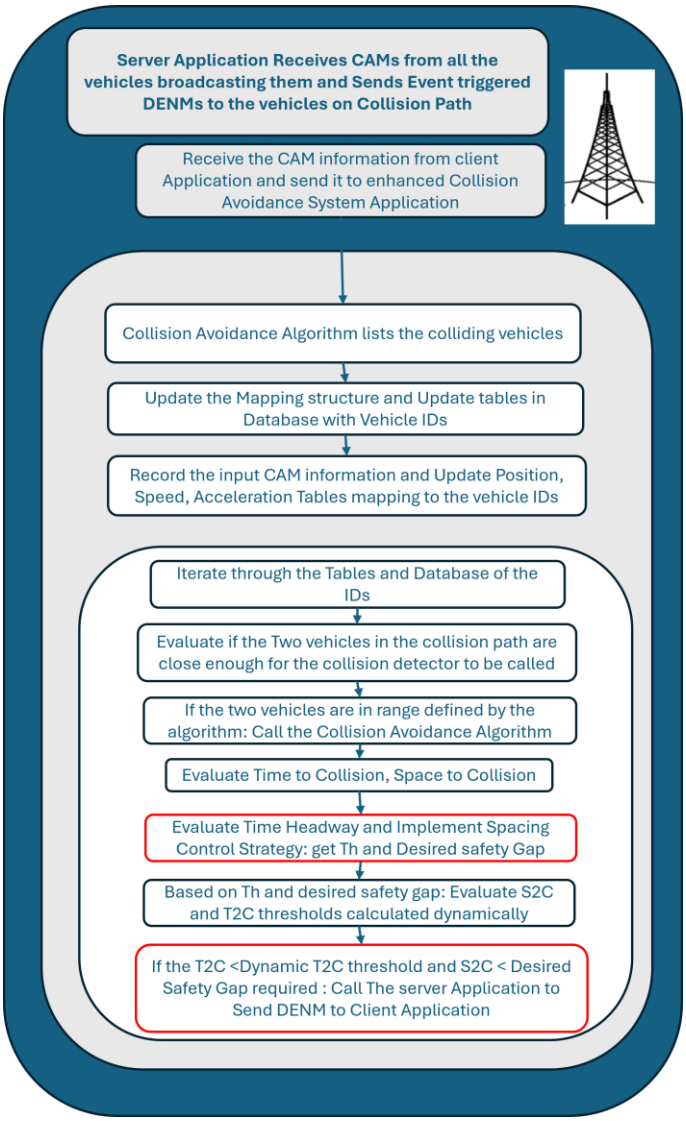

Figure 11 Server application and Collision avoidance application implemented at RSU (source: author).

In Figure 11 -illustrates how the information is passed on from the server application to the collision avoidance application which is called by the server application and is implemented on the RSU. Once enhanced Collision Avoidance applictation is fed with two or more vehicle's CAM messages, it saves this information by forming pairs of vehicles that might be on a probable collision course. This is determined by drawing a future projected trajectory of these vehicles, either as straight lines or curved segments while the vehicles are approaching the intersections of the road topology. As explained in [9], the Collision Avoidance Algorithm and Collision Avoidance Strategy predict future trajectory by checking if the vehicles will travel close enough to each other. If they do, the collision detector is called. The Future Position(FP) for each vehicle is calculated taking into account the Current Position(CP) of the vehicle, it's speed and a threshold defined as Time to Collision Threshold, as in the Equation 4.1.

$$FP(x,y) = CP(x,y) + Speed(x,y) * 1.5 * T2CThreshold \quad \text{Eq (4.1)}$$

Here the *CP(x,y)* and *Speed(x,y)* is the current position of the Vehicle and current speed of the vehicle which is delivered to the server from the vehicle through a CAM, and T2C threshold is the time to collision threshold that we are further in an optimal manner as explained further in this section The *x* and *y* components of Current Position and Speed are the vector components of both the position and speed of the vehicles that are evaluated for accuracy and precision. In the proposed algorithm, Collision Avoidance system with Variable Time Headway and Spacing Control strategy, we evaluate this trajectory by including a speed difference component evaluated as the difference in speed between the current speed of the vehicle and the maximum allowable free speed of the lane that the vehicle is traveling in. This factor allows us to study about the traffic fleet's speed in the given road topology, that is if the fleet is in a congestion or if they are moving with at an expected overall fleet speed. In [37] the authors, through experimentation highlight the importance of setting appropriate values for the time to collision threshold (t2ct) and space to collision threshold (s2ct) parameters, as they significantly impact the system's accuracy and efficiency in detecting collisions. For vehicles traveling at or around 50 km/h, the authors emphasize that the t2ct value should be at least 4 seconds to allow the driver sufficient time to react and avoid the collision, considering factors such as processing time, human reaction time, and braking time.

In this thesis, the 1.5 threshold coefficient is introduced to accommodate the minimum required t2ct value of 3 seconds, as mentioned in [37], for the range of speeds considered in the study. By multiplying the T2CThreshold value obtained from the rules-based logic approach by this coefficient, the thesis ensures that the threshold is appropriately scaled to provide a safety margin and account for varying range of vehicle speeds considered in this thesis. This coefficient provides a safety buffer and allows for the dynamic adjustment of the T2C threshold based on real-time traffic conditions, enhancing the reliability and effectiveness of the collision avoidance system.

For example, if the speed of individual vehicles is way less than the allowable speed set for the lane, there is probable congestion in the traffic or a possibility of an accident that has occurred, and hence the threshold of T2C that could be set here can be low. This is because if the speed of the vehicle is low and the automated the degenerative braking time for the vehicle to come to a full stop is 0.05s as discussed before[35], time that it will take to come to a full halt is lesser that if the vehicle was at high speed. Therefore, the threshold for T2C can be dynamically optimized based on the difference between the current speed of the vehicle and the maximum allowable speed of the lane that the vehicle is travelling within. We hypothesize that this will eliminate unnecessary braking and enhance the road's free space utilization while maintaining a safe gap between the vehicles so as to avoid any accidents. To estimate the traffic congestion, we define **VDiff** which is evaluated as a difference of velocity between the current vehicle's speed and the maximum drivable speed set for the lane that the vehicle is traversing in.

To optimize the system, we use a rule-based fuzzy logic control which can help us differentiate 4 different membership functions based on this speed difference. Fuzzy

control logic is a method used in systems where inputs can take on a range of values and outputs are derived based on linguistic (descriptive) terms rather than precise numerical computations[38]. The crux of the rule-based fuzzy logic is that it provides decision making framework pertaining to certain set of rules when the data being assessed or dealt with is large and non-linear. This decision-making framework is developed in a way that it's reasoning is similar to that of a human decision-making instinct. Here the rule-based fuzzy control logic is applied to adjust the Time to Collision (T2C) thresholds based on different traffic conditions inferred from the velocity difference (VDiff).

As a next step it is important to clarify that the approach used in this thesis is more accurately described as a rules-based logic approach rather than fuzzy logic in the strict sense. Fuzzy logic is a mathematical framework that deals with reasoning based on degrees of truth rather than the conventional binary logic of true or false. It involves defining membership functions and using linguistic variables to represent the degree to which an element belongs to a set. In contrast, a rules-based logic approach relies on a set of predefined rules or conditions to make decisions based on specific inputs or scenarios.

In the collision avoidance system presented in this thesis, the decision-making process is based on a set of predefined rules and thresholds. These rules are used to evaluate the traffic congestion state and determine the appropriate time-to-collision (T2C) threshold and minimum safety gap which is explained further in the next sub-section. The system does not employ fuzzy membership functions or linguistic variables to represent degrees of truth. Instead, it uses crisp values and specific conditions to make decisions.

Therefore, to maintain accuracy and clarity, the approach used in this thesis is referred to as a rules-based logic approach rather than fuzzy logic. The rules-based approach allows for effective decision-making based on predefined conditions and thresholds, enabling the collision avoidance system to adapt to different traffic scenarios and optimize safety and traffic flow efficiency.

Membership functions in this logic define how each point in the input space is mapped to a membership value between lowest value and highest value. These functions are key to defining the rule sets used in the fuzzy logic system. In this case, the membership functions can be parameterized to represent different traffic congestion states based on the **VDiff**. The importance of rule based fuzzy control logic here is to evaluate the T2C threshold based on the judgement of VDiff in the similar way that a human driver would have judged a possible congestion in traffic in her environment and decided when to decelerate from the current speed to avoid a probable collision.

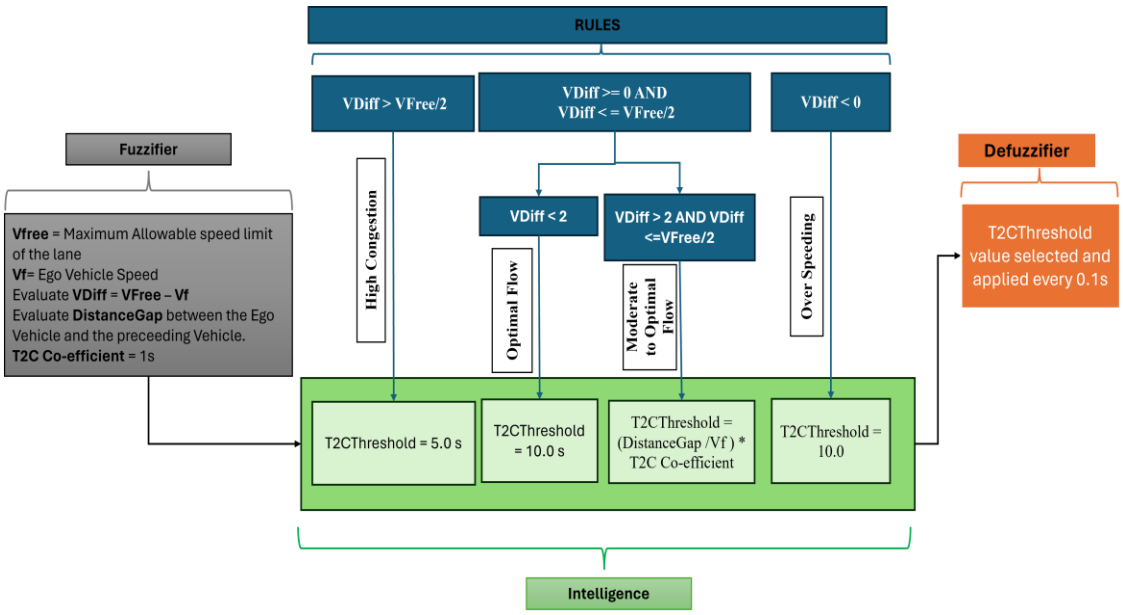

Figure 12 Rule-Based Fuzzy Control Logic to Evaluate T2C threshold value (source: author).

Figure 12 illustrates the Rule-Based Fuzzy logic control designed to evaluate and apply the T2C threshold value. The main components of a fuzzy controller are Fuzzifier, Rules, Intelligence and Defuzzifier [38]. Fuzzifier is the component that converts the raw values of data into fuzzy values. In this case, the fuzzifier is evaluating VDiff values by subtracting each vehicle's current speed from the maximum drivable speed value set for the lane that the vehicle is traversing in. Fuzzifier also calculates the distance between the vehicles based on the vehicles' information received from different vehicles using the CAMs that they broadcast. Here, we also set a T2C Co-efficient that can be further enhanced in future studies based on the desired results and vehicle's high-level controller requirements. We set this value to 1.

Next component is the Rule set which contains all the rules and the if-then conditions used to control the decision-making system. In our case, this rule set is defined by 4 rules which represent varying traffic congestion conditions as detailed in Table 1.

Table 1 Fuzzy Control Logic Rule Set and Intelligence.

Rule Set	**Traffic Fleet Congestion Significance**	**Intelligence: Time to Collision Threshold evaluation**
High Congestion: VDiff > VFree/2	High possibility of congestion, vehicle speed significantly lower than the speed Limit,	T2CThreshold: 5.0 reflects shorter required stopping distance due to lower speed.

Optimal Flow: VDiff < 2m/s AND close to VFree	Vehicles are nearly at maximum allowable speed	Threshold: 10.0 (to provide a safety margin at higher speeds).
Moderate to Low Congestion: VDiff > 2 AND VDiff < = VFree/2	Indicates moderate traffic flow where vehicles are moving but not at peak possible speeds.	Threshold = DistanceGap /vehicle's speed * T2C Co-efficient
Over Speeding: VDiff < 0	Vehicle speed exceeds the maximum allowable speed.	Threshold=10.0 (higher potential for severe consequences in collisions).

The rule sets in Table 1 represent different traffic fleet congestion estimate based on the comparison between evaluated VDiff and VFree. This evaluation is done within the collision avoidance system for every vehicle that is detected by the server application on RSU. Based on the Rule sets, fuzzy control logic defines intelligence within the fuzzy framework. This intelligence comprises of decision-making logic that defines what action is pertinent for each rule set. In the Fuzzy logic we have defined, the intelligence details what the T2C threshold is set to. The Varying T2C thresholds with different congestion states of traffic fleet is given in the Table 1 under the column 'Intelligence'.

T2C threshold is a value that determines when the vehicle should start decelerating and when the collision detector should enact on the probable collision. If the evaluated T2C is less than T2C threshold, the OBU indicates to the vehicle's controller that the deceleration should begin. The T2C threshold is set to a low value when the congestion is high, this is because the vehicle is at a lower speed, which means it need not require a larger distance for stopping in a short interval of time. Thus, relatively lower time to collision is safe for the vehicle to maintain at its current driving speed. When the congestion is low, and the vehicle is driving at an optimal speed where it is trying to gain a speed equal to the maximum speed set for the lane, the T2C threshold is set to a higher value of 10.0. This is to avoid the collision at high vehicle speed, as at this high speed the vehicle will take longer distance to come to a halt. Thus, we set a higher time to collision threshold to maintain that are a relatively higher evaluated value of time to collision, the collision detector must be called.

At a moderate to low congestion condition, the T2C threshold is evaluated as a convolution of the Gap distance between the vehicle for whom the T2C being calculated (referred to as Ego Vehicle) and the vehicle directly in front of it in the same lane(referred to as preceding vehicle), and the current speed of the vehicle. We also factor in a co-efficient here, currently considered as 1. This co-efficient is signified to also consider the traffic density, type of road (highway, urban, countryside etc.) and other factors that take into consideration the environment of the vehicle. This co-efficient can be further tuned and experimented in order to keep the T2C threshold's value to be

evaluated and changed with the changing environment that the vehicle is driving with. Our framework allows this flexibility for further experimentation.

In case of over speeding, the T2C threshold is set to a higher value to avoid collision with a larger safety distance, however we do not consider these conditions to evaluate the performance of our framework. Since over speeding directly translates to a regulatory issue and breaking of traffic rules, this situation is beyond the scope of our study. This case is only considered to not leave this rule set open ended.

Once the trajectory is generated, the trajectories of different vehicles are analyzed and compared, thus, to determine which pair of vehicles are set on collision course. For this, we are building upon the two parameters which are defined for CAA in [10] and enhanced Collision Avoidance in [9] that is Time to Collision and Space to Collision, by introducing Time headway and Desired Safety Gap like the ones defined in [11]. The implementation of this concept is explained in the subsequent section.

In [37] the authors discuss the impact of the time to collision threshold and space to collision threshold parameters on the number of false positive alerts generated by the collision avoidance system. False positive alerts refer to warning messages sent to vehicles in situations where the risk of collision is low or non-existent. A high number of false positive alerts can desensitize drivers, leading them to ignore future warnings and potentially compromising the effectiveness of the collision avoidance system. This hypothesis highlights that setting the Time to collision threshold and space to collision thresholds too high can result in an increased number of false positive alerts. Conversely, setting these thresholds too low can lead to a higher percentage of undetected or late-detected collisions. The authors emphasize the importance of finding an optimal balance between minimizing false positives and ensuring the timely detection of potential collisions.

In the thesis, the incorporation of the rules-based logic approach and the 1.5 threshold coefficient in Equation 4.1 addresses the challenge of reducing false positive alerts while maintaining a high level of collision detection accuracy. By dynamically adjusting the T2C threshold based on real-time traffic conditions and vehicle speeds, the proposed collision avoidance system can optimize its performance and minimize unnecessary warnings. The rules-based logic approach allows for the adaptive setting of the T2C threshold based on the estimated traffic congestion level, derived from the difference between the vehicle's current speed and the maximum allowable lane speed (VDiff). This dynamic adjustment ensures that the threshold is appropriately scaled to provide a safety margin and account for varying vehicle speeds, reducing the likelihood of false positive alerts in different traffic scenarios.

The motivation behind introducing these factors in the algorithms is that, when the trajectories are compared and a pair of vehicles are detected to be in the course of collision, the event triggered DENM that is sent to the vehicles from the RSU’s server application aims to either stop one of the vehicles and let the other vehicle pass through as planned by the trajectory evaluation, or stop both the vehicles. This is defined by a

definitive action space as explained in the Single-Event Collision Avoidance Strategy implemented in [3]. Both of these actions are aimed at reducing the speed of the vehicles involved in the collision pair.

Single Event Collision Avoidance Strategy (SE-CAS) defines four actions in its configurations which are described in Table 2. The actions are defined to decide on which vehicle is sent an event triggered DENM by setting the *precedence* value within DENM message to 0. This is decoded by the vehicle's OBU to command the vehicle's OBU's Client application to start the braking phase.

Table 2 Single Event Collision Avoidance Strategy: Predefined Actions [9].

Actions	**DENM Information**
Both Vehicles are stopped	DENMs with precedence = 0 is sent to both the vehicles
Only the Vehicle coming from the left is stopped, right of the way is given to the vehicle coming from Right	DENM with precedence = 0 is only sent to the vehicle coming from left.
Vehicle which is the slowest will be stopped, this is done so the vehicle which can stop in the shortest time is stopped	DENM with precedence =0 is sent to the slower vehicle in the pair of vehicles detected.
Vehicle which is the farthest from the intersection is stopped	DENM with precedence = 0 is sent to the vehicle which has to travel the longest to arrive at the intersection is stopped.

SE-CAS works on predefined actions, which is advantageous from a perspective of it being included in a safety application but reduces the overall efficiency of traffic in case of increased traffic density. In [9], the authors highlighted how this issue could lead to a possible deadlock, and its ineffectiveness can be worked upon due to the flexibility of the framework. Thus, we have harnessed upon this flexibility to further enhance the collision avoidance algorithm to an algorithm which, along with being a safety algorithm, also works towards enhancing traffic efficiency capabilities with adapting traffic density.

In the next section, we first discuss the variable time headway and spacing control strategy, then explain how the SE-CAS strategy is built upon by additionally factoring in these parameters which resolves the deadlocks, enhancing traffic efficiency with different traffic densities and factors in parameters such as waiting time and travel time to enhance the traffic flow with the road infrastructure.

4.3 Variable Time headway and spacing strategy

As discussed in the previous section, in this thesis work we are applying the time headway and spacing strategy to collision avoidance system to enhance its capabilities from being a safety application alone to an application that modulates the traffic flow efficiency while detecting and avoiding the probable collisions.

In this section, we discuss how this strategy is implemented in Enhanced Collision Avoidance defined in [9] and define the parameters based on which the traffic flow management is assessed and analyzed. The collision avoidance system we have implemented with the Variable time headway and spacing control strategy utilizes the Server-Client format of computing as explained in the section before which is similar to the system established in [9]. In evaluating time headway strategy, there are two widely known approaches as discussed in the literature review section, which are constant time headway strategies[26] and variable time headway strategy[30]. Due to the constant time headway strategy's inability to handle complex and dynamic traffic conditions, such as frequent acceleration and deceleration of the preceding vehicle, or changing spatial density, we move to a Variable Time headway strategy like that defined in the New Variable Time Headway and Spacing control strategy in [11].

4.3.1 Formal Definitions

Time headway is defined by Katja et al [17] as the time that is elapsed between the front of the lead vehicle passing a point on the roadway and the front of the following vehicle passing the same point.

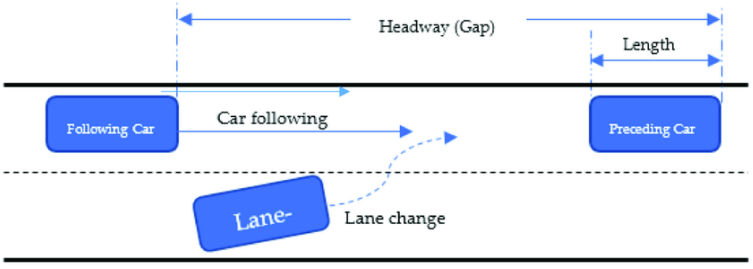

Figure 13 Preceding and Following car definition[39].

In the Figure 13, the following car is the one which is behind the preceding car or the leading car. In this thesis report, we have used the terminology preceding and leading cars interchangeably. The following car is also referred to as the Ego car since that is the vehicle, we are applying our actions or decisions to. Collision avoidance system's algorithm utilizes two main components: time to collision and space to collision to

evaluate what would be the deciding factor to qualify a particular scenario as a collision. It utilizes these factors to also predict a probable collision between two entities or vehicles. In the algorithm that we are studying and implementing in this thesis, we have considered the collisions that are or are probably going to occur at the intersections only.

Time to collision is defined as the time taken to collide if two vehicles continue to run at their present speed and on the same path (as defined in [19]). This is often referred to as a surrogate safety measure which is evaluated considering the kinematics of a conflict in traffic microsimulation[40]. T2C is measured for each timestamp and a threshold is used to determine whether a collision will happen if the current speed and direction are maintained. T2C evaluation and its modification helps in identifying the relationship between the reason why the conflicts/collisions occur and observed crashes within a simulated environment such as the one that we are experimenting with for this project.

Another factor that collision avoidance algorithms utilize is the Space to Collision, which is defined as the minimum safety distance that must be achieved between the vehicles in order to avoid collision. This factor is evaluated based on the current position of the vehicles, their speeds and the time to collision that is described before.

4.3.2 How is Variable Time Headway and Spacing Control Strategy Implemented?

Once the pair of vehicles, which are falling within the range of collisions which is evaluated based on their future trajectories as defined with equation 4.1, the positions of these vehicles is assessed to qualify them as a leading or preceding vehicles. In the context of intersectional collision scenario that is being evaluated in enhanced Collision Avoidance, amongst the pair of vehicles in the set collision path, a preceding vehicle is defined as the one that reaches a particular point at intersection before the other vehicle in consideration.

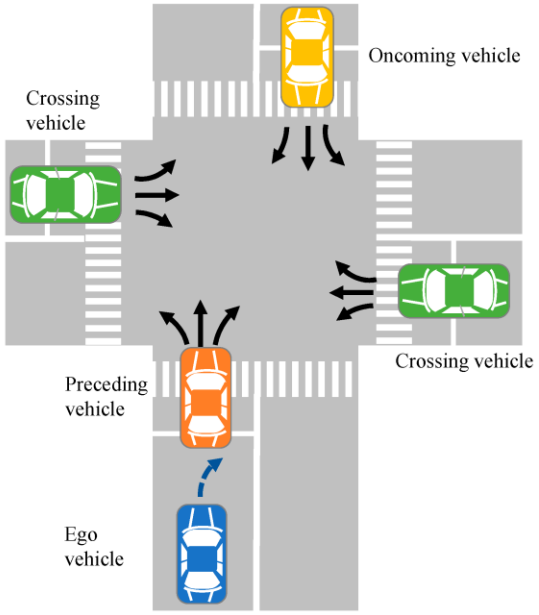

Figure 14 Terminologies for types of vehicles definition[41].

In Figure 14, we can see that preceding vehicle is the one denoted in orange color. This vehicle is set to reach a particular point before the following or Ego vehicle(represented by blue vehicle). Both the references to these vehicles are tagged as “Preceding” and “Following”, respectively, for the ease and continuity of the overall time headway and variable spacing control calculations.

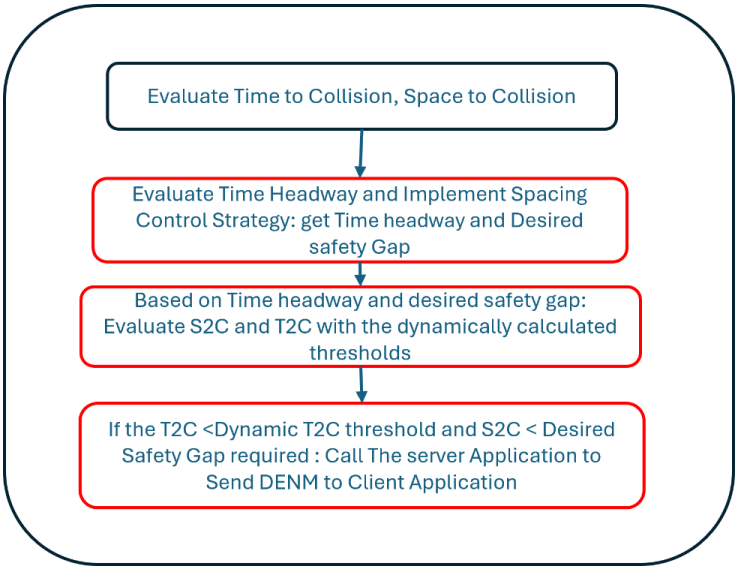

Figure 15 Evaluation of Time headway and implementing spacing control (source: author).

Figure 15 illustrates the implementation of time headway and spacing control within the collision avoidance algorithm. The blocks outlined in red indicate the novel approach introduced in the collision avoidance system with the intension for the application to regulate traffic flow efficiency.

On exploiting the information contained in the CAMs being sent from the vehicle's client application planted in their OBUs, the vehicles are tagged as following or preceeding using their relative positions on the lane and their unique IDs. The preceding and following vehicle's ground truth values relating to their velocities, acceleration and positions are updated. We utilize the equations to evaluate time headway as defined in [11] for the Following/Ego vehicle as denoted in the equation below.

$$\mathrm{th} = \mathrm{ck} * 1/(\rho\mathrm{jam} * (\mathrm{vfree} - \mathrm{vf})) - \mathrm{cv} * (\mathrm{vp} - \mathrm{vf}) - \mathrm{ca} * \mathrm{ap} \qquad \text{Eq (4.2)}$$

Here **th** is the time headway, **ck** is positive coefficient, **ap** is the preceding vehicle acceleration and **ca** is the positive weight coefficient of the preceding vehicle acceleration, **vf** is the velocity of the following car and **vp** is the velocity of the preceding car. The positive weight coefficient of the relative speed is considered as **cv** and **vfree** as defined earlier is the free flow speed. **ρjam** is the jam density that is defined as vehicles present on the road per kilometer. Here, moving away from the constant time headway strategy as defined in [19], which only considers the relative speed, we implement a variable time headway variation in the above equation which considers the speed change trend of the preceding vehicle which is encapsulated with the preceding vehicle's acceleration on a desired safety gap as in [11].

The spacing strategy forms the basis of our controller that modifies Collision Avoidance system's algorithm to ensure smooth and safe platooning while dynamically adjusting the desired safety gap between the vehicles through the entirety of their running and when they are on a collision course. By doing so, the algorithm ensures that the desired safety space and time headway are updated and evaluate to minimize the speed changes occurring in vehicles, specifically those for which a collision is avoided using enhanced Collision Avoidance by adapting the action space discussed in the SE-CAS strategy discussed before. The space control strategy mimics the behavior close to controller of a typical Cooperative Adaptive Cruise Control strategy, as in [11], in order to adjust the space between the preceding and following vehicle by capitalizing on the positional and kinematic information of the vehicles periodically.

For this spacing strategy, within Collision Avoidance system's algorithm, we next implement Spacing Control function that evaluates a desired safe space which translates to the Desired safety gap illustrated in Figure 15, that has to be maintained by each vehicle in order to avoid collisions with other vehicles in their close vicinity and in order to maintain a desirable speed which can enhance the efficiency of the macroscopic traffic flow of all the vehicles in the range of the Road side unit's coverage by aiming to increase their average speeds.

Amongst the pair of vehicles considered by the SE-CAS strategy[9], the desired safety space that should be maintained by them is evaluated by considering the time headway given by Equation 4.1 and a safety threshold value.

$$\Delta X_{des} = t_h * v_f + \Delta x_0 \qquad \text{Eq (4.3)}$$

ΔXdes is the desired safety distance taking into account a time headway factor and a minimum safety gap factor. The time headway factor is evaluated for every iteration by convoluting the following vehicle's speed vectors with the time headway evaluated in the previous equation (4.2). $\Delta \mathbf{X_0}$ denotes the minimum safety gap factor that accounts to the minimum distance that must be maintained by the following car with the preceding car when both the vehicles are completely stopped. For this factor, we have designed another fuzzy control logical block that defines this safety parameter by considering the braking reaction phase as defined for an automatic braking system [31].

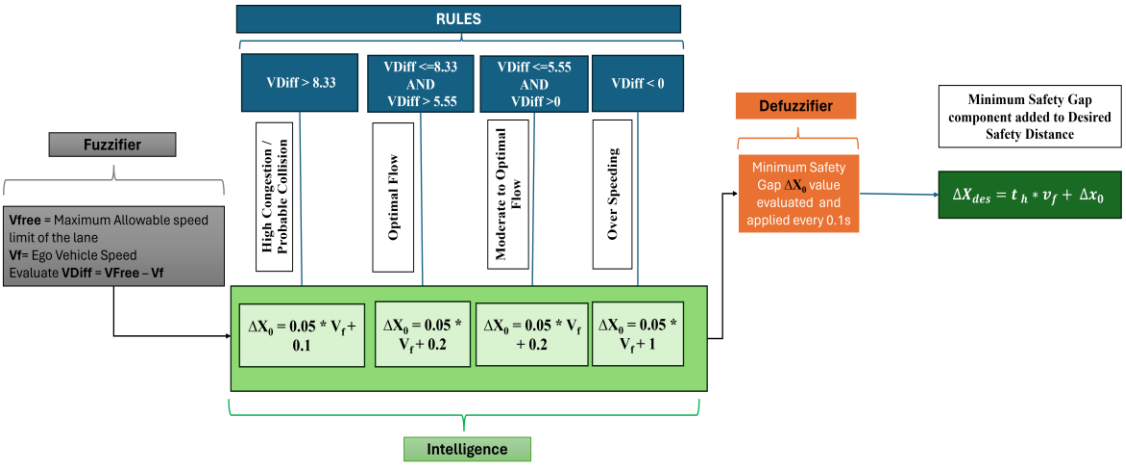

Figure 16 Fuzzy Control Logic for Minimum Safety Gap evaluation (source: author).

Figure 16 illustrates the fuzzy control logic which is developed for evaluating the minimum safety distance that is a component used in estimating the Desired Safety Gap in Equation 4.3. Unlike the value of this minimum safety gap that is set to 0 in [11], we have developed a dynamically evaluated minimum safety gap for each vehicle based on the congestion states considered by the fuzzy control logic. The fuzzy control, as explained in the earlier sub section consists of Fuzzifiers, Rule sets, Intelligence and the Defuzzifier. The Fuzzifier here illustrates the Velocity Difference (**VDiff)** that is evaluated to gauge the estimate of the traffic congestion based on the difference between vehicle's current speed and the free allowable maximum speed set for the lane. The rule sets are defined to consider different traffic congestion states based on this Velocity Difference. For fuzzy control, within this rule set, the congestion within the traffic ecosystem at the present time is compared with varying thresholds of difference in the speed between the vehicle's current speed (**Vf)** and the maximum allowable speed for the lane, to provide a safety which is adaptable with the changing traffic fleet's speeding

trends. The rule set and their significance with the decision- making logic embedded in the intelligence of the fuzzy control logic is represented in the table below.

Table 3 Fuzzy Control Logic Rule set and Decision-Making Intelligence for minimum safety gap.

Rule Set	**Significance**	**Intelligence: Evaluating Minimum safety distance ΔX_0**
High Congestion or Probable Collision: VDiff >8.33 m/s	In case of high congestion or probable collision when the Ego vehicle's speed is less, the minimum gap can be evaluated with a lower co-efficient of 0.1. This is indicative of lower required minimum gap since distance needed to be covered for the vehicle to halt completely is less with a constant degenerative braking time = 0.05s	$\Delta X0 = 0.05 * Vf + 0.1$
Moderate to Optimal traffic flow: VDiff <=8.33 m/s AND VDiff > 5.55 m/s	In case of moderate to optimal flow, the ego vehicle is trying to gain a higher speed to close the gap between its speed and the maximum speed set. This indicates a relatively higher speed than first case, hence we are factoring in a slightly higher coefficient value of 0.2m to give a higher minimum gap for the vehicle to maintain.	$\Delta X0 = 0.05 * Vf + 0.2$
Optimal traffic flow: VDiff <=5.55m/s && VDiff > 0	In case of optimal flow when the speed of the traffic fleet is higher and the Vdiff becomes very less, the time headway can lead to an infinite value. Hence, we contain the Desired gap by factoring in a higher value of 0.5. This also allows a higher minimum safety gap for the vehicles.	$\Delta X0 = 0.05 * Vf + 0.5$

Over speeding: VDiff < 0	This indicates that the speed of the traffic fleet is very high or that of the following vehicle is very high(it is overspeeding), therefore to be on a safer side, we factor in a higher coefficient value so as to maintain a minimum of 1m distance excluding the distance considered during the braking phase of 0.05s	$\Delta X0 = 0.05 * Vf + 1$

Table 3 illustrates the Rule set defined in the fuzzy control logic and the minimum safety gap calculations. We are considering the VDiff thresholds in m/s to be more precise since the minimum safety gap calculation is considering the braking reaction time as defined earlier as 0.05s [35]. We define braking reaction time of the vehicle which is the interval between the DENM reception by the vehicle's OBU at the application layer and the starting of deceleration phase. This value denotes the instant when the DENM is received by the Client Application and the trigger of the deceleration by the vehicle's actuator. The 0.05s value is considered for automatic braking controllers on the vehicles. In this research work and its simulation framework, all the vehicles we have defined are equipped with automatic braking controllers. However, our algorithms framework allows the flexibility to include different variety of vehicle's controller models and subsequently, different values of braking times. As in the equation 4.3, we multiply this value by the vehicle's current speed and factor in different coefficients as defined in the Table 3. The significance of these coefficients is to concatenate the distance that is needed for the vehicle to cover in its braking phase with that of a safety threshold factor to determine the safety gap which is neither too high that can lead to inefficient road occupancy, nor too low to lead into a collision with the preceding vehicle.

Factoring in the Time headway, Desired Safety gap and implementing the fuzzy control over time to collision threshold and Minimum safety gap, we introduce into enhanced Collision Avoidance[9] a traffic management system that monitors and regulates the traffic fleet's average speed along with ensuring the safety and avoiding collisions in a given traffic scenario

Next, the collision detector is called to evaluate if the collision is going to happen in case the vehicles maintain the current speeds and lane of traversal. This decision is factored in by considering the Desired safety gap that we evaluated in the Equation 4.3. The collision detector calls for evaluation of the Time to Collision. Traditionally, Time to Collision is evaluated as a ratio of the distance between the two vehicles in the collision course and their relative speed.

$$Time\ to\ Collision = \frac{Distance\ between\ following\ and\ preceding\ vehicle}{Velocity\ of\ Following\ Vehicle - Velocity\ of\ Preceding\ Vehicle} \quad \text{Eq (4.4)}$$

For an intersectional collision, if the velocity of the following vehicle is greater than the preceding vehicle, the relative velocity is positive; if the velocity of the preceding vehicle is greater than the following vehicle, then the time to collision runs into negative value. Thus, we use a saturation function to maintain this value with a minimum and maximum defined by the headway distance.

We factor in a minimum headway distance parameter using the time headway evaluated in the Equation 4.2. The minimum headway distance is the shortest distance achievable by a system without a reduction in the speed of vehicles as illustrated in Equation 4.5.

$$Minimum\ Headway\ Distance = Ego\ vehicle\ speed * th \quad \text{Eq (4.5)}$$

The minimum headway distance is evaluated by the time headway in a way that we calculate it for the Ego/following car as the convolution of the time headway between the Ego/following car and the preceding car evaluated as in Equation 4.2 and speed vector of the Ego/following car.

For evaluation of updated T2C we take the ratio of this minimum headway distance by the current relative speed between the Ego/following and the preceding vehicles.

$$newT2C = \frac{Minimum\ headway\ distance}{Velocity\ of\ Following\ Vehicle - Velocity\ of\ Preceding\ Vehicle} \quad \text{Eq (4.6)}$$

We utilize this new T2C as the time to collision evaluated by the server for each of the Ego vehicles based on their kinematic information sent by these vehicles via CAM messages.

Once the collision detector is called it qualifies a scenario as a probable case of collisions that might occur which is decided upon evaluating the future trajectories, time headway, spacing control strategy new time to collision as explained in this section earlier. The new time to collision is compared with the T2C threshold evaluated earlier by the fuzzy control logic for every vehicle as a CAM from the vehicle is received. Using their Kinematic information encoded within the CAM messages, the spacing control strategy is evaluated, and this information is made available for the client application which decides the vehicle's heading and acceleration (explained further). If the evaluated new time to collision is less than the evaluated T2C threshold based on the fuzzy control logic, the Collision Avoidance application triggers the Server Application to send a Decentralized Environment Notification Message (DENM) to the vehicles involved in this collision scenario. While this evaluation is being conducted by the server based on the CAMs it receives from the vehicles, the vehicle's On-Board Unit runs a Client Application as we mentioned before. In the next sub section, we will study the working of this application and delve into the implementation of the additional safety platooning built upon the CAA[10] and CAS's[9] client application implementation.

4.3.3 Spacing control and acceleration setting for Client Application

The Client Application's functionality plays a vital role in the how efficient the traffic fleets speed performance is and how accurate the collision avoidance applications is. The application is equipped with the API that connect the data communications between the server and client application. This application is also capable of gauging the presence of a leader vehicle or a preceding vehicle. This capability is assumed and implemented to be a wireless connectivity between the traffic fleet in this thesis implementation, however it can be extended to utilize the in-vehicle sensors such as radar, camera and lidar if the vehicles are already equipped with these. The Client application designed and implemented here can adapt to different input channels from the vehicle's Electronic Control Unit (ECU) or any other distributed network that the vehicle is connected to in order to enhance its perception capabilities.

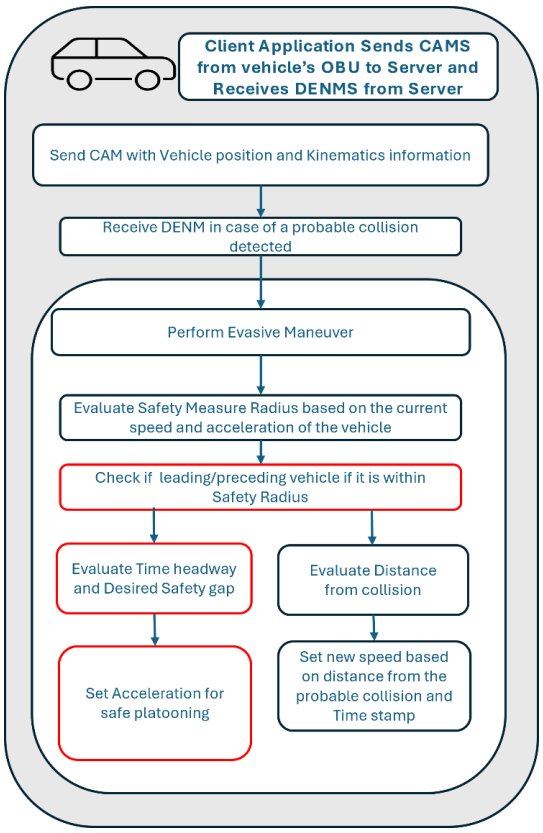

Figure 17 Client Application: Logical analysis (source: author).

Figure 17 refers to the client application's implementation in which we consider that the CAMs being sent by the vehicle's client application is also capable of receiving the CAMs from other vehicles, within a defined radius of operation. This radius keeps changing as the vehicle that is receiving these CAMs keeps traversing through the road topology. The client application is also capable of being installed and implemented in different types of User Equipment's (UEs) or on-board units. The client application can be installed in a pedestrian's mobile phone, a connected bicycle's on-board unit or any

vehicle with the capability of being connected in an LTE-5G network. The scope of this project is only restricted to implementing this application in a Car or a truck equipped with the V2I communication compatible On-Board Units, since the kinematics of these vehicles are inputted additionally for the evaluation of control strategies in the Collision Avoidance System's algorithm.

An essential part of the client algorithm is the Evasive Maneuvering which is defined for the Ego/Following vehicles to gauge the presence of the preceding vehicle in its safety radius and set its speed factoring in the time headway and spacing control. When a collision in detected, the Evasive Maneuvering is designed to detect the distance from the probable collision and adjust its speed factoring in a speed coefficient which is the ratio of the distance from collision and the evaluated safety radius. In the Figure 17, the red outlined blocks represent the introduction of this novel approach in client application that is implemented in the research work.

The safety radius(SRad) is evaluated periodically with the frequency of 10 Hz which is the rate at which the CAMs are disseminated and it expressed as,

$$SRad = Ego\_Speed * T_d + 0.5 * Ego\_Accel * {T_d}^2 \qquad \text{Eq (4.7)}$$

It is essential to note that, as discussed before, the client application runs for every vehicular node independent of the presence of the other applications and their proximity. Hence, every vehicle has its own safety radius evaluated with the equation above. *Ego_Speed* is the Ego/following vehicle's speed and *Ego*_Accel is the Ego/following vehicle's acceleration that is obtained from the vehicle's speed sensors and ground truth measurements, and time duration T_d is the reciprocal of the frequency at which the application updates the speed and position of the vehicles. In the case of the simulations run for this thesis experimentation, this value is equivalent to 0.01s which is the value at which the traffic simulator updates the vehicular information. This is discussed in detail further in the next chapter.

There are two approaches that are being illustrated in each of the vehicle's application which is based on if there is a preceding vehicle(that is also called the leader vehicle) in the vehicle's proximity, that is within the safety radius of the vehicle (client) as evaluated in Equation 4.7. From Figure 17, we can infer that if there is a leader or a preceding vehicle that is present within the safety radius, then the time headway and desired safety gap is evaluated taking into consideration the leader or preceding vehicle's position, speed and acceleration. This information is received from the server application enclosed within the DENM that received from the server as and when requested for it by the client application. In this thesis, we assume that the client requires this information in a broadcasted fashion from the infrastructure and hence we allow the server to send this information to individual vehicles every 0.01s. The client application selects a leader/preceeding vehicle that is within its radius and the server application, using this vehicle's ID evaluates and gauges it's kinematic attributes. This is also conditional to both the ego and leader vehicle's being in the safety radius of the RSU's server application. The safety radius calculation ensures this is always true as the safety radius is

always less than the overall network coverage of the server(1km) in our case. We also get from the server application the evaluated values of time headway and desired safety gap using the equations 4.1 and 4.2.

In the client application, we introduce a desired acceleration value within the Variable Space Control strategy with the equation below,

$$a_{sc} = K_a * a_p + K_v * (\mathrm{v}_p - \mathrm{v}_f) + \mathrm{K}_s * (\mathrm{s} - s_d) \quad \text{Eq (4.8)}$$

$$s_d = \Delta X_{des} - L_p \quad \text{Eq (4.9)}$$

$$\Delta X_{des} = t_h v_f + \Delta x_0 \quad \text{Eq (4.10)}$$

The acceleration of the vehicle is evaluated taking into consideration $\mathbf{K_a}$ which is the feedback gain of the acceleration term, $\mathbf{K_v}$ and $\mathbf{K_s}$ are the positive gains of the speed term and the spacing term respectively. $\mathbf{S_d}$ is defined as the desired safety gap between the following vehicle's front end of the bumper and the preceding vehicles rear end, this is obtained from server applications calculation of desired safety gap from Equation 4.2. From the desired Safety gap evaluated in 4.2 we minus the $\mathbf{L_p}$ term. $\mathbf{L_p}$ is the length of the preceding vehicle, this information in encoded in the CAM messages that are broadcasted by the vehicles to the server application. However, in case the in-vehicle sensors which are detecting and evaluating the target objects, the perception algorithms therein can detect the dimensions of the target objects/ vehicles in their environment of traffic ecosystems which can be utilized by the client application. In our experimentation, we rely solely on the information encoded by the on-board units of individual vehicles within the CAMs sent by these vehicles respectively.

Thus, the acceleration term is defined in a similar way as that defined for CACC vehicles in [11] and [30]. To further evaluate and limit this term with the consideration of the driver's comfort and braking reaction time, the deceleration and acceleration in the space control mode limits this value to a minimum deceleration value given by $\mathbf{a_{min}}$ and maximum acceleration by $\mathbf{a_{max}}$ in such a way that these values are saturated in the following way,

$$a_{sc} = \max(a_{min}, \min(\mathrm{a}_{sc}, \mathrm{a}_{max})) \quad \text{Eq (4.11)}$$

For the client application to adjust its speed, we then evaluate the speed that the vehicle would attain if it were to accelerate by the value $\mathbf{a_{sc}}$ as calculated in equation 4.10. We use the relationship between acceleration and speed given by the equation,

$$v = u + a * t \quad \text{Eq (4.12)}$$

where $\boldsymbol{v}$ is the desired speed to be obtained by an entity which is traveling with an initial speed of $\boldsymbol{u}$ and is accelerating by a value $\boldsymbol{a}$ in $\boldsymbol{t}$ time. Hence the new speed for the vehicle to obtain would be given by,

$$newSpeed = currentSpeed - a_{sc} * 0.01 \quad \text{Eq (4.13)}$$

Where the *currentSpeed* is the speed with which the vehicle is travelling at the time of this evaluation, $\mathbf{a_{sc}}$ is the required acceleration evaluated for the spacing control strategy and 0.01 in our case is the time gap between the observations accounted for the vehicle's kinematic values.

Thus, the *newSpeed* is set by the vehicle's on-board units by triggering the high-level controller present in the vehicles. This in totality takes into consideration the time headway, desired safety gap and the acceleration to maintain a higher speed in order gain traffic fleet efficiency in an adaptable manner while avoiding collisions at intersections. The aim of inclusion of these concepts with enhanced Collision Avoidance is to make it more adaptable to the changing traffic ecosystem and enhance traffic efficiency. Before discussing the simulation setup and the results obtained from experimenting with different traffic networks and density, we will establish the expected enhancements that we are aiming to look for by adapting enhanced Collision Avoidance to be a traffic flow enhancement algorithm.

4.3.4 Expected behavior after variable time headway implementation.

Enhanced collision avoidance algorithm, as discussed in the sections earlier and as described within the scope of [9] and [10] is restricted to function as a safety algorithm alone. By implementing variable time headway and spacing control strategy, in this research work, we further this algorithm to enhance the traffic flow efficiency of the traffic fleet at a macroscopic level and enhance the co-relation between the average speed of vehicles and the collisions.

The first stage of implementation is described by the Time headway and desired safety gap implementation in the Single-Event Collision Avoidance Strategy described in the previous subsection. By introducing these two factors, we expect for the average speed of the vehicles to increase for varying traffic density due to the optimization of time to collision parameter and the safety gap parameter. The time to collision parameter is a key component in decision-making of if a scenario is qualifying as collision, thus practically influencing a change in the speeds of the vehicles which are involved in this scenario. This factor does not consider the optimal safety space which should be evaluated if the collision is to occur, or the optimal time headway that should be calculated to maintain the speeds of the vehicles in case the detected collision is a false positive event. A false positive event is defined as an event which has been qualified as a probable collision warning but in reality, it is due to the miscalculation or erroneous evaluation of the traffic parameters that this event is trigger.

The time headway and spacing control strategy consider several key factors which help in getting an overall accurate environmental information of the traffic ecosystem that the car is driving in. For example, traffic density gives us information about the congestion on the lane that the vehicle is traversing in. Factoring this in, while maintaining a time headway and spacing between vehicles to promote safe platooning, not only makes the traversal safe by avoiding collisions but also adaptable to the changing traffic conditions. Thus, with changing densities, enhanced Collision Avoidance is expected to corelate the collisions occurring with the average speed trends in a uniform and adaptable manner.

The second stage of implementation that allows for adaptability and enhancement road occupancy is the fuzzy control drawn over the minimum safety gap that should be maintained by the vehicles when they come to a complete halt. This factor forms the basis of desired safety gap implementation in Spacing control strategy. For this fuzzy control, we consider the speed difference between maximum free allowable speed allotted to the lane that the vehicles are traveling in and the current speed that the vehicle is running with. This enables the algorithm to consider the required increase in the speed for the vehicle to reach it maximum potential, thus enhancing the traffic flow efficiency of the overall fleet. Based on this, we consider the braking phase reaction time and the current speed with which the vehicle is running to evaluate the distance the vehicle will travel to come to a complete halt. Such an implementation makes the algorithm efficient with the adapting traffic to make the best utilization of the lane space. That is, if the traffic is congested and the speed of the vehicle is low, the minimum desired distance would be less than if the traffic fleet is free flowing and the speed of the vehicles in traffic is high, which accounts to larger space availability for the vehicles to come to halt. This aims at decreasing wait times of the vehicles in the intersections if the traffic is free flowing. We expect to see a reciprocal trend between the wait times of the vehicles and the traffic density.

The third stage is the implementation of this strategy within client applications in the vehicular nodes of the network. It enhances the vehicular speed by a significant degree while maintaining safe and efficient platooning with the other vehicles. This was not taken into consideration with the enhanced Collision Avoidance and Collision Avoidance Algorithms earlier. One of the essential reasons to include this in a safety algorithm such as Collision Avoidance is because of the dependency of the average speeds of the traffic fleet on the collisions detected. The traditional methods and theory of Collision Avoidance calls for reduction of the speeds of vehicles involved in the probable collisional event, without considering the enhancement of the speeds once the collision is avoided or avoiding this speed reduction in case of false positives. Thus, we expect to see a lower number of collisions occurring due to the elimination of false positives and an increase in the average speeds due to the enhanced variable platooning techniques employed for adaptable traffic conditions.

Largely, performance network parameters such as Packet Drop Ratio, Traffic load efficiency and the data transmission latency remain unaffected. However, there can be a slight drop in the packet drop ratio since the number of DENMs being sent from server to the clients is reduced due to elimination of some false positive events. These factors are

defined and studied in detail in the next chapter that entails the simulation setup that was established to develop these algorithms and the results of these experimentations carried out for varying traffic conditions are illustrated with appropriate arguments to justify the behavior of the system. Appendix B is dedicated to the pseudo code analysis of the above-mentioned algorithms.

4.3.5 Data for analysis

The data synthesized for the experimentation of the Collision Avoidance system is simulated by implementing the application in a multi-stack vehicular ad-hoc network framework. This framework provides flexibility in configuring various software settings and networking conditions, allowing for comprehensive testing of the system's performance and robustness under different scenarios.

The simulated road network consists of several lanes with varying speed limits, and the enhanced Collision Avoidance algorithm, along with the variable time headway and spacing control strategy, takes these speeds into account while evaluating the individual vehicles' kinematics. To collect representative data, the simulation setup is designed to vary the maximum drivable speeds (V_{Free}) and study the system's performance on road networks with physical parameters resembling real-world road lanes and networks.

Since the focus of this thesis is on intersectional collisions, the data collection primarily concentrates on lower speed limits to facilitate comparison with the results of the collision avoidance algorithm [9] and collision avoidance strategy presented in [10]. The data collection methods are consistent with those used in the research works to ensure continuity and enable meaningful comparisons between the experimental outcomes.

For each speed limit, two sets of data are collected with different vehicular traffic spatial densities. This approach allows for the analysis of the system's performance under varying traffic conditions and provides insights into its robustness and adaptability to different traffic scenarios.

Moreover, the simulation framework generates data on various network performance metrics, such as traffic load, latency, and packet delivery ratio (PDR). These metrics help assess the communication performance of the Collision Avoidance system and its ability to efficiently disseminate messages between vehicles and the edge server. The application-related metrics, including the percentage of collisions avoided and average vehicle speed, are also collected to evaluate the effectiveness of the enhanced Collision Avoidance system in enhancing road safety and maintaining traffic flow efficiency.

By synthesizing data across a range of maximum drivable speeds and spatial densities, the experimental setup enables a comprehensive evaluation of the enhanced Collision Avoidance system's performance, robustness, and adaptability to different road networks and traffic conditions. The consistent data collection methodology ensures comparability with previous research works and allows for meaningful conclusions to be drawn regarding the effectiveness of the proposed system in real-world scenarios.

5 Simulation Setup and Parameters

In the preceding chapters, we have established the theoretical foundation, literature review, and architectural aspects of the vehicular communication system employed in this research project. Specifically, we have explored the Multi Stack Vehicular Ad-hoc Network (VANET) setup, which incorporates the NS-3 modules for building and simulating ETSI-compliant VANET (V2X) applications using SUMO and NS-3. For this research endeavor, we have implemented the proposed algorithms within the msvan3t simulation framework on Ubuntu 22.04. This chapter aims to provide a comprehensive guide to the simulation setup, including detailed explanations, relevant figures, and diagrams elucidating the simulation architecture and its significance.

We delve into a detailed discussion of how the proposed algorithms were built and simulated within MS-VAN3T[21].Additionally, we explore the SUMO network topology considered in this project for running simulations and collecting data. We conclude by emphasizing the network and application parameters used to evaluate the performance of this research work..

MS-VAN3T is an open-source simulation framework designed for testing vehicular applications[21]. It combines the capabilities of the NS-3 network simulator and the Simulation of Urban MObility (SUMO) tool, offering state-of-the-art vehicular communication models and ETSI-compliant message dissemination stacks. This framework provides flexibility in configuring the communication technology and network architecture, allowing for simulations of both centralized (V2I/V2N) and distributed (V2V) vehicular network architectures. One of the key strengths of MS-VAN3T lies in its integration of the ETSI Facilities layer, which includes the Cooperative Awareness (CA) and Decentralized Environmental Notification (DEN) Basic Services. These services facilitate the encoding, decoding, and management of standardized ETSI messages, such as Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs), ensuring compliance with industry standards.

To evaluate the performance of our algorithms, we have constructed a specific network topology and varying traffic fleet scenarios within SUMO. This topology represents an urban environment, consisting of multiple intersections, two-way streets, and with different speed limits on the streets. By mimicking real-world conditions, this scenario allows us to assess the effectiveness of our algorithms in realistic traffic situations.

To comprehensively analyze the performance of our proposed algorithms, we have identified a set of key parameters and metrics. These parameters will be used to configure and run the simulations, while the metrics will serve as quantitative measures to evaluate the effectiveness of our algorithms. The simulation parameters encompass various aspects, such as the number of vehicles, vehicle types (passenger cars), communication technology (LTE-5G), transmission rates and ranges, packet sizes and rates, mobility models, and vehicle trajectories.

The performance metrics encompass various aspects, including safety, efficiency, and reliability, to comprehensively evaluate the algorithms' impact on vehicular communication and applications. Key metrics include collision avoidance rate, average vehicle speed, waiting times and travel times of individual vehicles, end-to-end latency, and packet delivery ratio. We will provide detailed explanations of these metrics, their relevance to the research objectives, and the methods employed for their calculation and analysis.

5.1 Simulation Framework

The simulation framework presented in "An Edge-Based Framework for Enhanced Road Safety of Connected Cars" [9] aims to evaluate the performance of a collision avoidance system for connected cars in different scenarios. To implement the collision avoidance algorithm, we retain this framework. This framework combines a traffic simulator, a wireless communication model, and a collision avoidance algorithm to create a realistic environment for testing the system. The simulation scenario consists of two cars approaching an intersection from different directions, and the goal is to avoid a collision by exchanging safety messages through V2I communication. The framework evaluates the system's performance in terms of safety metrics, such as collision rate, average traffic fleet and false positive rate, and communication metrics, such as packet delivery ratio and end-to-end delay. Additionally, the novel approach of utilizing this framework for traffic flow efficiency monitoring and enhancements provide us with two more metrics to monitor: Travel time and overall waiting time of individual vehicles. The overall framework allows us determine the traffic fleet’s trip information with details such as individual vehicle parameters including average speeds, travel time, waiting times, etc.) and traffic fleet’s average speed recorded over varying spatial density by increasing and decreasing the number of traffic participants(vehicles, in our case) within the simulation setup.

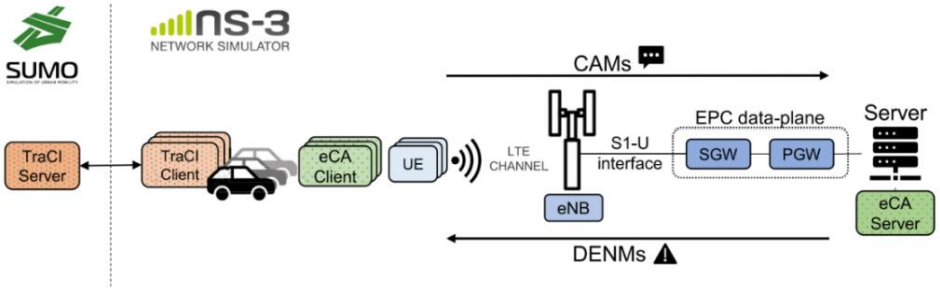

Figure 18 High level architecture of the simulation framework and data plane [9].

The high-level architecture explained in Figure 18 shows interoperability between SUMO and NS-3 simulator, where SUMO serves as traffic simulator that houses TraCI server. We establish vehicles in SUMO and configure them as client nodes using TraCI client. The TraCI client establishes a data flow pipeline between the client application described in the earlier chapter. Through this pipeline, we can source the real-time vehicular

parameters such as the vehicle's position, current speed, steering angle, heading, etc, from the SUMO simulator.

Network Simulator 3 (NS-3)

The NS-3 simulator provides a flexible and customizable simulation environment, allowing researchers to configure various network parameters such as network topology, routing protocols, and channel models[42]. In this study, NS-3 version 3.35 is leveraged and utilized along with the LENA framework for modeling the LTE stack. The LENA (LTE/EPC Network Emulator for NS-3) framework is an extension to the NS-3 simulator that provides a realistic and flexible simulation environment for simulating LTE (Long-Term Evolution) and

EPC (Evolved Packet Core) networks. It allows testing to be simulated with the advantages of LTE/EPC networks under various scenarios and evaluate the performance of network protocols and applications.

- The LTE module is responsible for implementing the LTE protocol stack, including the PHY layer, MAC layer, RLC layer, PDCP layer, RRC layer, and NAS layer. It supports multiple carrier aggregation, MIMO antennas, and different channel models.
- The EPC module implements the EPC network, including the MME, SGW, PGW, and HSS. It supports mobility management, packet routing, and authentication.
- The Traffic model module provides support for different traffic models and traffic classes, while the Application module implements different applications and supports different transport protocols.

Traffic Simulator

This thesis research work use SUMO (Simulation of Urban Mobility) as a mobility generator to model the real-world traffic infrastructure scenario [12]. This framework uses microscopic simulation of vehicle traffic and can simulate various aspects of vehicular traffic such as lane change, collision detection and vehicle movement. The road network is first created using the Open Street Map tool and then imported into the SUMO framework. The SUMO then generates the traffic and inputs the required aspects of the vehicular node information onto the NS-3 module for simulation. Thus, SUMO helps in studying and tuning the algorithm according to the realistic representation of the traffic condition.

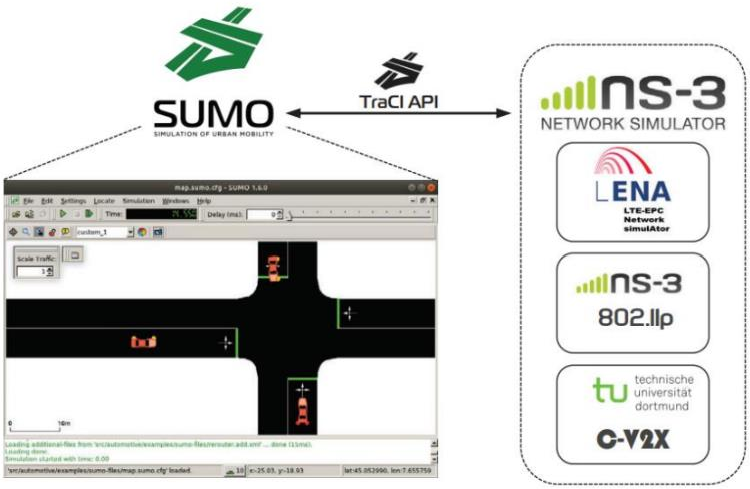

Figure 19 Main components of the simulation framework [9]

This thesis research work utilizes the simulation framework illustrated in Figure 16 that evaluates the performance of a communication system for connected and autonomous vehicles (CAVs). For our study, the applications in enhanced Collision Avoidance are integrated with the Multi stack framework introduced in [10] to evaluate the Collision Avoidance Strategy algorithm. The integration details and simulation parameters which are set have been described in detail below.

5.2 Multi-Stack Vehicular Ad-Hoc Network (MS-VANET)

MS-VAN3T is a multi-stack simulation framework that provides a novel V2X tool for developing vehicular communication-based applications and conducting virtual validation. By combining ns-3 and SUMO, it supports the integration of hardware-in-the-loop while merging both mobility and connectivity of vehicles. The integration of enhanced Collision Avoidance within MS-VAN3T is particularly beneficial, as MS-VAN3T enables simulation offered by a full ETSI C-ITS compliant stack.

MS-VAN3T leverages the capabilities of ns-3, which supports CAM (Cooperative Awareness Message) and DENM (Decentralized Environmental Notification Message) for deriving information from vehicles and sending event-based triggers to the vehicles, respectively. CAMs are periodically sent by vehicles to share their status and behavior, while DENMs are used to exchange information about local environmental conditions and events.

One of the key advantages of MS-VAN3T that makes it an ideal simulation tool for our use case is its ability to switch between communication stacks, perform large-scale simulations by scaling the number of traffic participants, and its fully open-source nature, which allows for experimentation with various variations of the applications. Integrating *Collision Avoidance system* into MS-VAN3T is done by utilizing the automotive module of ns-3, which is the main module of MS-VAN3T. This module contains sample

applications for both V2I and V2V ETSI C-ITS sub-modules that help in establishing network and communication in the road infrastructure.

The enhanced Collision Avoidance system consists of a Collision Avoidance Algorithm (CAA) and a Collision Avoidance Strategy (CAS). By integrating enhanced Collision Avoidance into MS-VANET, we can leverage the multi-stack communication capabilities and the ETSI C-ITS compliant stack to simulate and evaluate the performance of the collision avoidance system in various scenarios. In the next section, we will provide a more detailed explanation of the NS-3 modules utilized, mainly focusing on the automotive module in MS-VAN3T to offer clarity and insight into the implementation of enhanced Collision Avoidance within this simulation framework.

5.2.1 Network Simulator 3 – Lena

LENA (LTE/EPC Network Simulator) and LENA-5G (5G-LENA) are open-source simulation modules developed by the Open Sim Research Unit (RU) at the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) in collaboration with industry partners. These modules enable researchers and developers to model and simulate 4G and 5G Long-Term Evolution (LTE) and 5G New Radio (NR) networks, respectively, within the NS-3 (network simulator 3) ecosystem[13] .

LENA and LENA-5G are tightly integrated with NS-3, a discrete-event network simulator widely used in academic and industrial research[43]. By leveraging the core functionalities and modular architecture of ns 3, these modules provide a comprehensive implementation of the LTE and NR protocol stacks, closely adhering to the 3rd Generation Partnership Project (3GPP) specifications. This integration facilitates the creation of end-to-end simulations encompassing the complete protocol stack, from the physical layer to the application layer, enabling the simulation of complex, multi-radio access technology (multi-RAT) scenarios where LTE, NR, and other technologies like Wi-Fi can coexist and interact.

The architecture of LENA and LENA-5G follows a modular design that mirrors the respective 3GPP LTE and NR standards. Key architectural components include the Physical Layer (PHY), which implements channel models, modulation schemes, and coding techniques conforming to the 3GPP specifications. The Medium Access Control (MAC) layer handles resource allocation, scheduling, and other MAC-related functions. The Radio Link Control (RLC) layer ensures reliable data delivery through segmentation, concatenation, and retransmission mechanisms. The Radio Resource Control (RRC) layer manages control plane signaling, handling functions such as connection establishment, handover, and radio bearer configuration. Additionally, the Packet Data Convergence Protocol (PDCP) layer handles header compression, ciphering, and other packet-related operations. Furthermore, LENA and LENA-5G include models for the Evolved Packet Core (EPC) and the NextGen Core, respectively, enabling simulations of end-to-end network scenarios.

In our simulation framework, we use EPC to establish the data plane as shown in Figure16. The EPC is deployed at the edge of the network at the server application. The

client applications which are implemented on the OBUs of the vehicle are representing the LTE's User Equipment (UEs). In the data plane, EPC houses serving gateway (SGW) and a Packet data Network Gateway (PGW). The connectivity for the node running in the client application is given by PGW, which helps in gathering information from the vehicles which are sending in CAMs and PGW also generates unicast DENM messages from the server to the client nodes. Therefore, enabling an end-to-end communication between client and server applications.

For all the applications demonstrated in this study, the ns-3 node of each vehicle consists of two nodes:

1. TraCI client: TraCI is a client-server interface that allows SUMO to communicate with external programs, such as ns-3, in real-time. It provides a set of commands that can be used to retrieve information about the simulation state, such as vehicle positions, speeds, and accelerations. The traCI client is implemented in the ns-3 simulation environment and is responsible for connecting to the TraCI server, which is running inside the SUMO simulation environment. Once the connection is established, the TraCI client can send commands to the TraCI server to retrieve information about the simulation state, such as the location and speed of each vehicle.

2. Collision Avoidance client: This is used in the results and analysis of the Enhanced Collision avoidance study. This client application manages the CAM dissemination and is used to take maneuvering decisions when the DENM is received.

5.2.2 Module Overview- Automotive Module

ms-van3t is developed on top of the ns-3 network simulator and the Simulation of Urban MObility (SUMO) tool. It is designed to facilitate the simulation and evaluation of various aspects of vehicular networks, including communication technologies, routing protocols, and applications.

At the core of ms-van3t lies the automotive module, which comprises four main components: nodes and mobility, network devices, routing and networking, and applications. These components work together to create a comprehensive simulation environment for vehicular networks.

Nodes and Mobility: The nodes and mobility component handle the creation and management of nodes within the simulation environment. In ms-van3t, nodes are created in the ns-3 simulation as vehicles enter the SUMO simulation. The number of vehicles is determined by parsing the mobility trace file (cars.rou.xml), which contains information about the vehicles' movements and trajectories. The vehicles can be categorized into different types, such as passenger vehicles and emergency vehicles.

Each vehicle in the simulation starts sending Cooperative Awareness Messages (CAMs) with a frequency between 1 Hz and 10 Hz, following the ETSI standards. The mobility of the vehicles is managed by SUMO through the TraCI interface, which provides a bidirectional coupling between ns-3 and SUMO.

Network Devices: ms-van3t supports both LTE and 802.11p communication technologies for vehicular networking scenarios, providing a multi-stack architecture that allows users to switch between these communication stacks easily.

For LTE-based scenarios, ms-van3t utilizes the LENA (LTE/EPC Network Simulator) module of ns-3. The network topology consists of User Equipments (UEs), representing vehicles, connected to an eNodeB (eNB), which is further connected to the Evolved Packet Core (EPC) network.

For 802.11p-based scenarios, ms-van3t utilizes the WAVE (Wireless Access in Vehicular Environment) model of ns-3. Vehicles, equipped with On-Board Units (OBUs), communicate directly with a Road Side Unit (RSU). Vehicles broadcast periodic CAM messages, and the RSU periodically broadcasts DENM messages to inform vehicles traveling in specific areas to adjust their speed.

Routing and Networking: ms-van3t supports various routing and networking protocols to facilitate communication within the vehicular network environment. It includes support for IPv4 and IPv6 networking protocols, as well as broadcast communication for disseminating CAMs and DENMs using the underlying communication technology (e.g., IEEE 802.11p or LTE).

Applications: ms-van3t provides support for various applications, both specific to Intelligent Transportation Systems (ITS) and generic networking applications. Researchers and developers can develop and integrate their own custom applications into ms-van3t, leveraging the provided APIs and interfaces. This flexibility allows for the simulation and evaluation of new protocols, algorithms, and applications in the context of vehicular networks.

5.2.3 CAM and DENM structure

In the simulation architecture designed for the enhanced Collision Avoidance system, the generation and exchange of safety-critical messages between vehicular nodes and the infrastructure play a vital role. These messages, namely Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs), are essential components of the vehicular communication protocol. They enable the timely and accurate sharing of crucial information about the state of vehicles and the surrounding environment, which is paramount for ensuring road safety and enabling advanced driver assistance systems like enhanced Collision Avoidance.

CAMs and DENMs are time-critical messages that carry ground truth information about vehicular and infrastructure nodes. In the context of the Collision Avoidance system, these messages are employed to detect and mitigate potential collision risks. The timely and correct delivery of these messages between vehicles and infrastructure can significantly influence the overall performance and effectiveness of the Collision Avoidance algorithm. To gain a deeper understanding of how these messages contribute to the functioning of the Collision Avoidance system, we will first explore the CAM generation architecture and its significance within our simulation framework. We will

delve into the details of how CAMs are constructed, transmitted, and processed by the various entities in the vehicular network.

Following the discussion on CAMs, we will shift our focus to DENMs and examine their critical role in the enhanced Collision Avoidance system. We will investigate how DENMs are triggered, generated, and disseminated to alert vehicles about potential hazards or adverse road conditions. Furthermore, we will analyze how CAMs and DENMs are encoded and decoded within the multi-stack vehicular ad-hoc network (VANET) framework. This will provide insights into the practical aspects of message exchange between vehicles and infrastructure, enabling us to emulate real-world scenarios and evaluate the performance of the enhanced Collision Avoidance system under various conditions.

By examining the CAM and DENM architectures in detail, we aim to highlight their importance in facilitating effective communication and coordination among vehicular nodes. This understanding will serve as a foundation for assessing the impact of these messages on the overall functionality and reliability of the enhanced Collision Avoidance system in enhancing road safety.

Cooperative Awareness Message (CAM) Architecture

Cooperative Awareness Messages (CAMs) are a type of facility standardized by the European Telecommunications Standards Institute (ETSI) to support vehicular safety and traffic efficiency applications. CAMs provide a basic awareness service in cooperative Intelligent Transportation System (ITS) networks by enabling periodic exchange of status information among neighboring nodes. Each ITS entity (vehicle, in our case) periodically broadcasts CAM messages containing its presence, position, and status to other stations within its single-hop communication range. By receiving and processing CAM messages the server gains awareness about other vehicular nodes in its vicinity, including their positions, dynamics, and other relevant attributes. This awareness is crucial for various ITS applications that rely on continuous tracking and monitoring of vehicles in the network.

In the reference architecture presented in the research papers [3], [4], which is consistent with the architecture used in this research work, CAM messages are generated by the Vehicle ITS Station Mobile Router (Vehicle ITS-S MR) and transmitted via the LTE-5G wireless interface, which is specifically designed for vehicular environments. Roadside ITS Station Access Routers (Roadside ITS-S ARs) within the communication range of the vehicles receive these CAM messages and forward them to the Central ITS Station Application Server (Central ITS-S AS). The Central ITS-S AS then decodes the received CAM messages and makes the extracted vehicle status information available to the upper-layer ITS applications for further processing and utilization. The Vehicle ITS-S MR is responsible for generating CAM messages based on the vehicle's status and dynamics. The CAM message structure, as illustrated in research paper [31], is compliant with the ETSI EN 302 637-2 standard. It consists of a header and a body, where the header contains fields such as protocol version, message identifier, and generation

timestamp, while the body carries the core information about the originating ITS station.

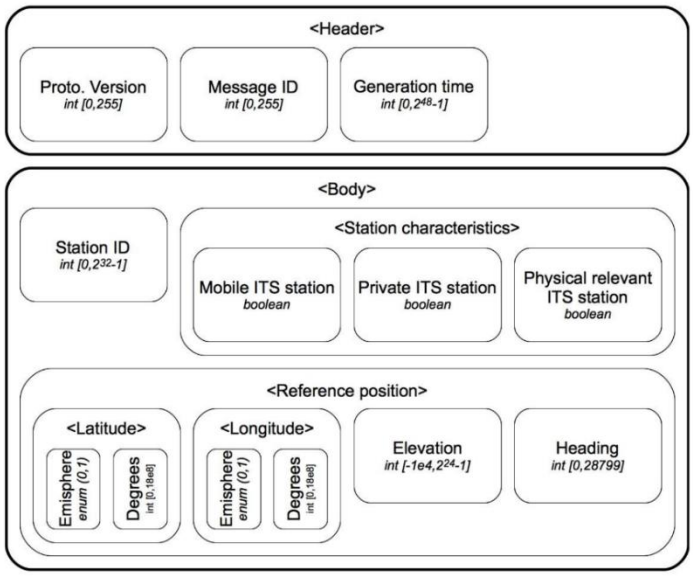

Figure 20: CAM structure [15].

Key fields in the CAM message body include:
Station ID: A unique identifier for the originating ITS station.
Station Type: Indicates the type of the ITS station (e.g., vehicle, roadside unit, public transport, etc.).
Reference Position: Specifies the geographical position of the ITS station in terms of latitude, longitude, and altitude.
Heading: Represents the current heading of the vehicle.
Speed: Indicates the vehicle's current speed (optional field, not included in the paper's implementation).

The Vehicle ITS-S MR periodically generates CAM messages based on the vehicle's movement and a set of predefined triggering conditions. These conditions, as described in Algorithm 1 shown in the figure below from the research paper [31], are evaluated each time a new GPS position update is received, or a maximum time interval has elapsed

since the last CAM transmission.

Algorithm 1: CAM message generation algorithm

Input: A new position read *p*, if any; a record of previous positions *pHist*; the last CAM message sent *lastCam*

Output: A new CAM message is sent and *lastCam* is updated, if applicable; *pHist* is updated with *p*, if applicable

```
1  while true do
2      time = System.getTime()
3      heading = calcHeading(pHist, p)
4      lastPos = lastPosition(pHist)
5      lastHist = pHist \ lastPos
6      lastHead = calcHeading(lastHist, lastPos)
7      speed = calcSpeed(pHist, p)
8      if p ≠ null then
9          lastSpeed = calcSpeed(lastHist, lastPos)
10         if distance(p, lastCam.pos) ≥ D_THRESHOLD or
11         |heading - lastCam.heading| ≥ H_THRESHOLD or
12         |speed - lastCam.speed| ≥ S_THRESHOLD then
13             cam = newCam(time, p, heading, speed)
14             sendCam(cam)
15             lastCam = cam
16         pHist = pHist ∪ p
17     else
18         p = lastPos
19         heading = lastHead
20     if time - lastCam.time ≥ T_THRESHOLD then
21         cam = newCam(time, p, heading, speed)
22         sendCam(cam)
23         lastCam = cam
24     System.wait(CHECK_PERIOD)
```

Figure 21: CAM generation algorithm [15].

The triggering conditions are as follows:
Distance Threshold: If the distance between the current position and the position in the last transmitted CAM exceeds a certain threshold (4 meters), a new CAM is triggered.
Heading Threshold: If the change in the vehicle's heading since the last CAM transmission exceeds a predefined threshold (4 degrees), a new CAM is triggered.
Speed Threshold: If the change in the vehicle's speed since the last CAM transmission exceeds a certain threshold (0.5 m/s), a new CAM is triggered.
Maximum Time Interval: If the time elapsed since the last CAM transmission exceeds a maximum interval (1 second), a new CAM is triggered regardless of the other conditions.
When any of these triggering conditions is satisfied, the Vehicle ITS-S MR generates a new CAM message, populates its fields with the current vehicle status information, and broadcasts it via the LTE-5G interface.

Decentralized Environmental Notification Messages (DENMs)

Decentralized Environmental Notification Messages (DENMs) are another type of facility standardized by ETSI to support vehicular safety applications. DENMs provide an event-driven notification service about road conditions and hazards. While the primary

focus of DENMs is on road safety applications, they can be extended to support any ITS application that requires information about road traffic events and conditions.

DENMs are critical and important in vehicular networks because they enable the timely dissemination of road hazard warnings and traffic incident notifications to vehicles and infrastructure nodes. By providing real-time information about adverse road conditions, accidents, or other events that may impact traffic safety and efficiency, DENMs help drivers and autonomous vehicles make informed decisions and take appropriate actions to avoid or mitigate potential risks. This can significantly enhance road safety, reduce accidents, and improve overall traffic management.

DENM Generation and Dissemination

In the V2I communication scenario, DENMs are generated by the Central ITS Station Application Server (Central ITS-S AS) in response to a request from an ITS application (the collision avoidance system in our case) that detects a relevant road event or condition. The application provides the necessary information about the event to the DENM facility, such as the event type, position, detection time, expiry time, destination area, and transmission frequency.

Upon receiving the event information, the DENM facility at the Central ITS-S AS creates a DENM message and determines the Roadside ITS Stations (Roadside ITS-S) located within the event's destination area. The DENM message is then distributed to the selected Roadside ITS-Ss, which forward the message to vehicles within their communication range using IPv6 multicast.

The Vehicle ITS Station Mobile Router (Vehicle ITS-S MR) receives the DENM message from the Roadside ITS-S and forwards it to the Vehicle ITS Station Host (Vehicle ITS-S Host), where the actual ITS application resides. The application processes the received DENM, evaluates the relevance of the event information, and takes appropriate actions, such as notifying the driver or triggering an autonomous vehicle maneuver.

In the research papers [3], [4] and in this research work, DENMs are used to implement a road incidence notification service. The Central ITS-S AS generates DENM messages based on event information received from the ITS server application and distributes them to the relevant Roadside ITS-Ss. The Roadside ITS-Ss then broadcast the DENMs to vehicles in their vicinity. At the vehicle side, a reference application is developed to process the received DENMs and display the event information to the driver through a graphical interface.

DENM Message Structure

The structure of the DENM message, as defined in the ETSI EN 302 637-3 standard and depicted in Figure 20 below consists of a header and a body.

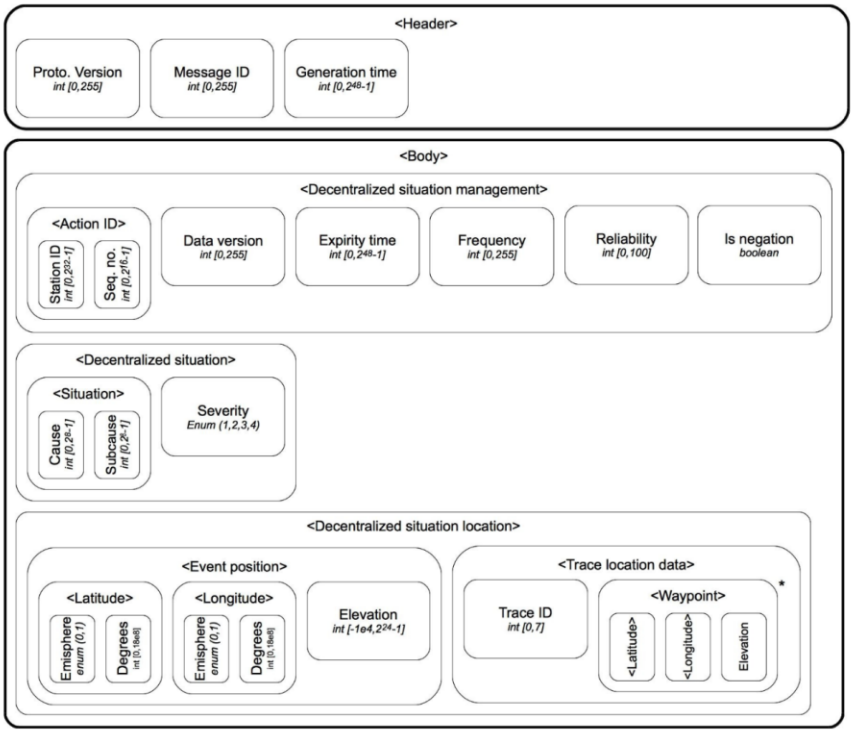

Figure 22 DENM message structure [15].

The header contains fields such as protocol version, message identifier, and generation timestamp. The body carries the event-specific information, organized into three categories:
Management Container: Includes general information about the event, such as the event identifier, version, expiry time, transmission frequency, and reliability.
Situation Container: Contains details about the event, including the cause, sub-cause, and severity.
Location Container: Specifies the geographical location of the event, including the event position and trace.

The DENM message structure is designed to convey comprehensive information about road events, enabling receiving ITS stations to interpret and act upon the notifications effectively. The standardized format ensures interoperability among different ITS stations and applications.

DENM Generation Algorithm

The research [32] presents a detailed algorithm (Algorithm 2) for DENM message generation at the Central ITS-S AS, shown in the Figure 21.

Algorithm 2: DENM message generation algorithm

Input: A new event *e* is received from one of the applications; a record of the previously generated DENMs *dHist*; a record of the previously generated event traces *tHist*

Output: A new DENM message is generated and sent to involved Roadside ITS-Ss, if applicable; *dHist* is updated with a new DENM message, if applicable

```
1  sequenceNo = generateSequenceNo()
2  previousDenm = findDenm(dHist, e)
3  dataVersion = 1
4  if previousDenm ≠ null then
5      if e.termination then
6          dataVersion = EVENT_END
7      else
8          dataVersion = (previousDenm.dataVersion + 1) % 256
9      delDenm(dHist, previousDenm)
10 mapSituationCause(e.type, ref cause, ref subcause)
11 mapSeverity(e.gravity, ref severity)
12 trace = findTrace(tHist, e.waypositions)
13 if trace == null then
14     trace = newTrace(e.waypositions)
15     tHist = tHist ∪ trace
16 denm = newDenm(INVALID_TIME, STATION_ID, sequenceNo, dataVersion,
   e.expiryTime, e.frequency, e.reliability, cause, subcause, severity, e.pos, trace)
17 rsus = findRsusInEvent(e.pos, e.dissZone)
18 sendDenm(denm, rsus)
19 if previousDenm == null then
20     dHist = dHist ∪ denm
```

Figure 23 DENM generation algorithm [15].

The key steps of the algorithm start with the algorithm, upon receiving a new event from an ITS application, checks if a previous DENM was generated for the same event. If a previous DENM exists and the event is being terminated, the data version field is set to indicate the event's end (255). Otherwise, the data version is incremented. The event cause, sub-cause, and severity are mapped based on the information provided by the ITS application. If trace waypoints are provided, a new trace is created or an existing trace is updated. The DENM message is constructed with the appropriate fields, including the event information, expiry time, transmission frequency, and reliability. The Roadside ITS-Ss within the event's destination area are identified. The DENM message is sent to the selected Roadside ITS-Ss for further dissemination to vehicles. The Roadside ITS-S, upon receiving a DENM from the Central ITS-S AS, executes the DENM forwarding algorithm. This algorithm ensures that the DENM is retransmitted to vehicles at the specified frequency until the event expires or is explicitly terminated.

DENM in Enhanced Collision Avoidance

In the context of the Collision Avoidance service, DENMs play a crucial role in notifying vehicles about potential collision risks and providing instructions for collision avoidance maneuvers. When the enhanced Collision Avoidance service detects a potential collision situation, it generates a DENM message containing information about the detected hazard, such as the type of hazard, detection time, and predicted collision point. The DENM is then disseminated to the vehicles involved in the potential collision through the Roadside ITS-Ss. Upon receiving the DENM, the vehicle's onboard Collision Avoidance application processes the message and takes appropriate actions based on the received instructions. This may include warning the driver, triggering an automatic braking system, or initiating a coordinated collision avoidance maneuver in cooperation with other vehicles. By leveraging DENMs, the Collision Avoidance service can effectively communicate collision warnings and avoidance instructions to vehicles, enabling them to respond promptly and avoid or mitigate potential accidents. The standardized DENM format and dissemination mechanisms ensure reliable and timely delivery of critical safety information in the context of the Collision Avoidance application.

In summary, the DENM architecture in the presented simulation framework enables the event-driven dissemination of road hazard and traffic incident notifications in vehicular networks. DENMs are generated by the Central ITS-S AS based on event information received from ITS applications and distributed to relevant Roadside ITS-Ss for broadcasting to vehicles. The standardized DENM message structure and generation algorithms ensure effective communication of critical safety information. In the context of the enhanced Collision Avoidance service, DENMs are used to notify vehicles about potential collision risks and provide instructions for collision avoidance maneuvers, enhancing road safety in connected vehicle environments.

5.2.4 ASN encoding

ASN.1 (Abstract Syntax Notation One) is a widely adopted standard for defining the structure and encoding rules of data exchanged in telecommunications and computer networking[23]. In the context of Intelligent Transport Systems (ITS), ASN.1 is used to encode and decode Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs), as standardized by the European Telecommunications Standards Institute (ETSI).

In this thesis work, we utilize ASN.1 encoding and decoding for the transmission of CAMs and DENMs within the Collision Avoidance algorithm implemented in the Multi-Stack Vehicular Ad-hoc Network (MSVAN3T) simulation framework. The enhanced Collision Avoidance algorithm relies on the exchange of these standardized messages to enable vehicles to share their status, position, and environmental information, allowing for collision detection and avoidance. The ASN.1 encoders and decoders for CAMs and DENMs are deployed within the ns-3 (Network Simulator 3) environment. ns-3 is a discrete-event network simulator that provides a comprehensive framework for modeling and simulating various networking protocols and systems. By integrating the ASN.1

encoders and decoders into NS-3, we ensure that the generated CAMs and DENMs adhere to the ETSI ITS standards.

In our implementation, the size of CAMs and DENMs is set to 83 bytes, which is sufficient to accommodate the necessary information required for the enhanced Collision Avoidance algorithm. However, the size of these messages can be easily modified to cater to the specific requirements of different applications or scenarios.

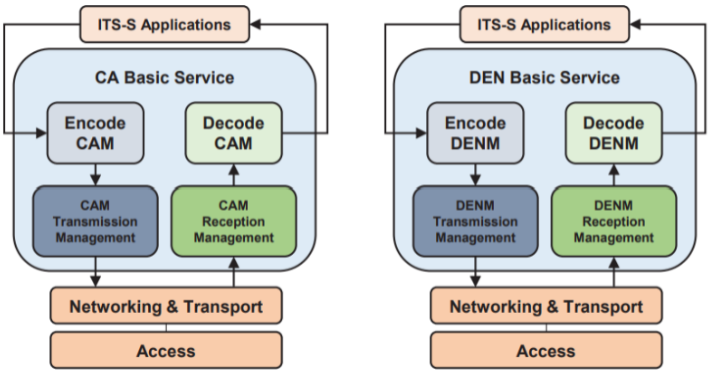

Figure 24 ASN.1 encoding and decoding for CAM and DENM messages at the server edge [21].

Referring to the diagram in Figure 24, the ITS-S Applications layer includes the CA Basic Service and DEN Basic Service, which are responsible for handling the encoding and decoding of CAMs and DENMs, respectively. The CA Basic Service consists of modules for encoding and decoding CAMs, as well as managing their transmission and reception. Similarly, the DEN Basic Service includes modules for encoding and decoding DENMs, along with managing their transmission and reception.

The structure of the ASN.1 encoders for CAMs and DENMs defined in Figure 24 illustrates the format and content of the messages, ensuring that they are encoded and decoded consistently across different vehicles and communication systems. The encoders take the relevant information, such as vehicle position, speed, and environmental data, and encode it into the specified ASN.1 format. On the receiving end, the decoders extract the encoded information and make it available to the enhanced Collision Avoidance algorithm for further processing. In the enhanced Collision Avoidance algorithm, vehicles periodically broadcast CAMs containing their status information. These CAMs are encoded using the ASN.1 encoder and transmitted through the lower layers of the communication stack (Networking & Transport and Access layers)[23]. Upon receiving a CAM, the server's CA Basic Service utilizes the ASN.1 decoder to extract the relevant information. This decoded information is then used by the Collision Avoidance algorithm to track the positions and status of vehicles and detect potential collision risks.

When a potential collision is detected, the Collision Avoidance system's algorithm generates a DENM to alert the involved vehicles. The DENM, containing details about

the detected collision risk, is encoded using the ASN.1 encoder within the DEN Basic Service at the server. The encoded DENM is then transmitted through the communication stack to the intended vehicular nodes. Upon receiving a DENM, the receiving vehicle's DEN Basic Service employs the ASN.1 decoder to extract the collision-related information, which is then utilized by the vehicle's Collision Avoidance system's client application algorithm to take appropriate actions (defined in earlier chapters).

By leveraging ASN.1 encoding and decoding, the Collision Avoidance System algorithm implemented in the MSVAN3T[21] simulation framework ensures that the exchanged CAMs and DENMs are structured, encoded, and decoded in a standardized manner. This approach enhances the reliability and effectiveness of the collision avoidance system, enabling vehicles to accurately interpret and act upon the received information in real-time. The integration of ASN.1 encoders and decoders within the ns-3 environment allows for the realistic modeling and testing of the Collision Avoidance System's algorithms both at server and client applications, and its reliance on the exchange of standardized messages. The flexibility to adjust the size of CAMs and DENMs further enhances the adaptability of the system to accommodate the requirements of different applications or scenarios.

In summary, the utilization of ASN.1 encoding and decoding for CAMs and DENMs, as standardized by ETSI, is a crucial aspect of the Collision Avoidance System's algorithm implemented in the MSVAN3T [21]- simulation framework. The deployment of ASN.1 encoders and decoders within ns-3 ensures adherence to the ITS standards, enabling seamless communication and interoperability between vehicles. By exchanging standardized messages, the enhanced Collision Avoidance algorithm can effectively detect and mitigate collision risks, thereby enhancing road safety in intelligent transportation systems.

The reaction time is defined as the interval between the time when DENM is received at application layer and when the braking of the vehicle starts because of the DENM action decoded by the actuator of the vehicle. When a human driver is driving in semi-autonomous or manually driven car, the braking phase is defined as 1s. While the braking phase is defined as 0.05s in the case of a vehicle with automatic braking system. In our experimentation and simulation framework, we have only considered all the vehicles that are equipped with automatic braking system modules; hence we have used the reaction time to be 0.05s.

Table 4 Communication Network Parameters

Parameter	**Value**	**Description**
Application Message format (CAM and DENM)	ASN.1	ASN.1 decoder and encoder implemented in ns3
CAM Frequency	10Hz	Frequency of CAMs being sent
CAM Size	83 Bytes	Packet size at Physical layer

Parameter	Value	Description
DENM Size	83 Bytes	Packet size at Physical layer
Reaction time	0.05 s [automatic braking system]	Time between the DENM received and start of the actuator reaction
Transport Network layer	UDP-IP	UDP is used to create communication sockets
Fading Model	Trace-based Fading Model	Fading calculation to reduce computational complexity
UE Transmission power	23 dBm	Transmission Power for UEs
eNB transmission power	45 dBm	Transmission power for eNB
Bandwidth	5Mhz	LTE channel bandwidth

Table 4 details the communication network parameters. This table summarizes our discussion on the CAM and DENM protocol front. The Trace based fading model is an inbuilt module used in NS-3 that we leverage[44].

5.2.5 SUMO – Creating Simulation Scenario

To demonstrate the working philosophy and simulate results for the performance metric parameter evaluation, the following sumo network is constructed.

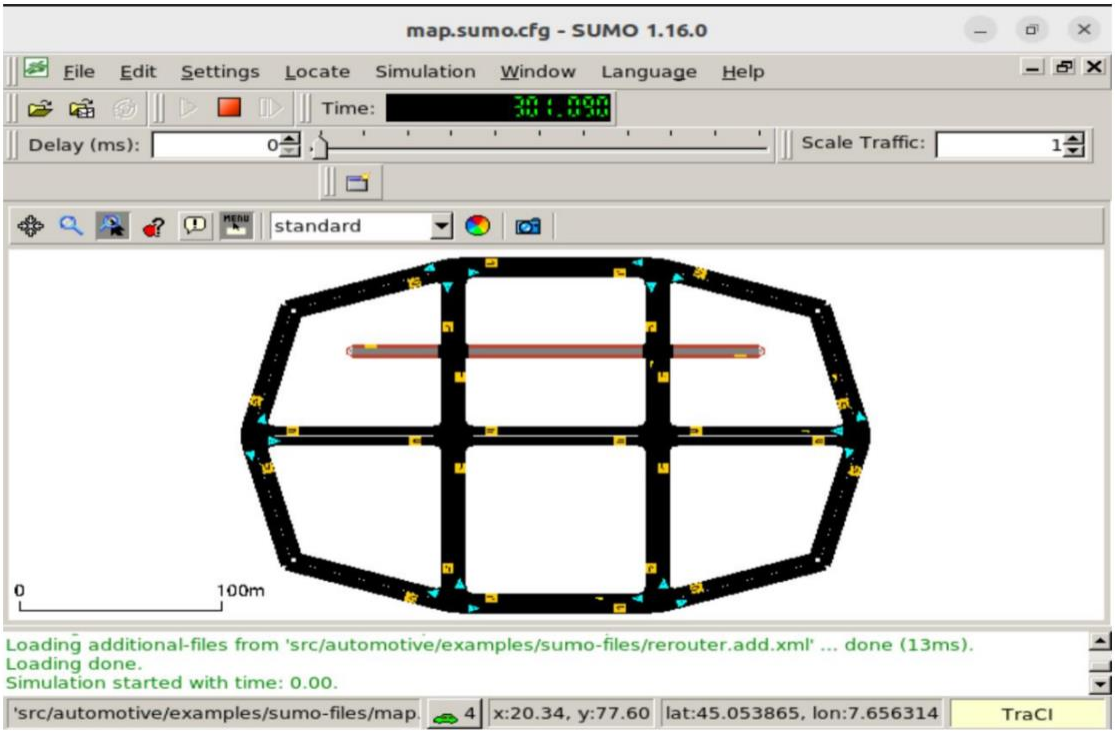

Figure 25 Sumo Road Topology (source: author).

This SUMO network consists of two vertical roads which are 400 meters long and one horizontal road which is a total of 700 meter. The vehicles on both these roads are allowed to travel in both directions. There are a total of 6 entry lanes and two intersections. The acceleration and deceleration values for maximum evaluation are fixed. The spatial density and maximum speed each vehicle can take during the course of their travel is varied according to the scenario and test cases. The vehicles enter from one of the six defined lanes and are generated with a Poisson's distribution of rate = 0.7. The mobility parameters are mentioned in the table below. These parameters are set to allow the experimentations and simulations to be carried out for testing urban and sub-urban

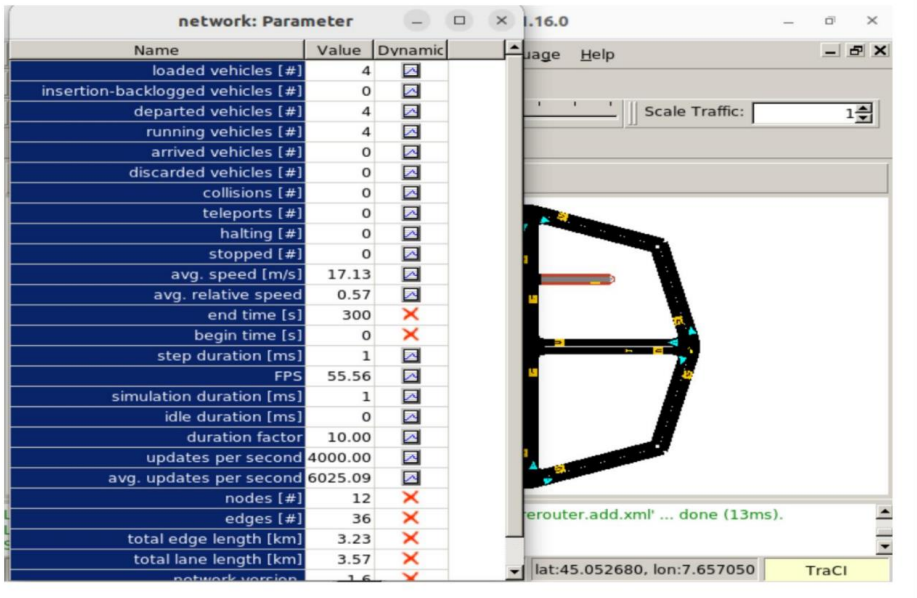

Figure 26 Vehicle and network parameters set on the SUMO and TraCI client setup (source: author).

Figure26 shows the network parameters set which comprises of number of vehicles loaded onto the simulation setup (here, 4 vehicles), average speed of all the vehicles in the topology, number of edges and nodes, and finally the total length of the lane in the topology. For the Collision Avoidance system's simulations, 2.22km has been considered as drivable path.

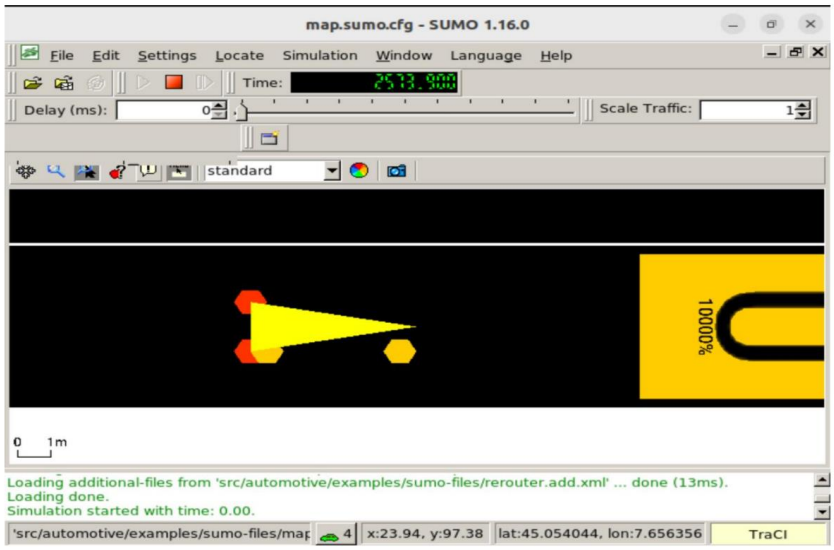

Figure 27 Tracking of individual vehicles which can be traced over the duration of the simulation to study their behavior or targeted study of one of the vehicles in the system (source: author).

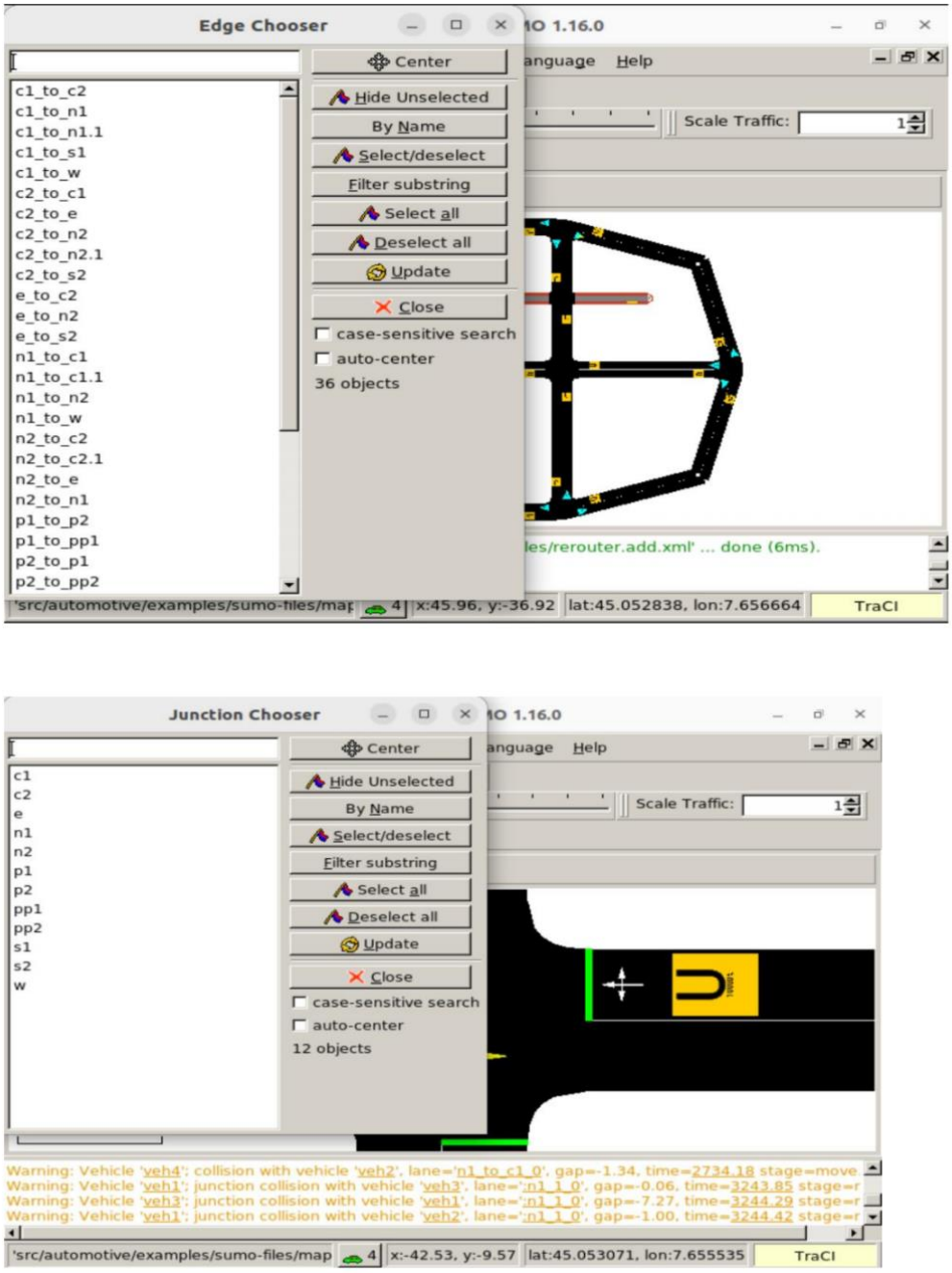

Figure 28: Road edges on SUMO simulator (source: author)

The topology's edges and nodes are specified and monitored as shown in the Figure 28. The junctions here are the intersections of the road topology where the Collision Avoidance service is deployed to monitor the junctional collisions occurring. These functions can be modified through the simulation setup.

Table 5 describes the mobility parameters which are set for the experimental setup for this study.

Table 5 Mobility and simulation parameters

Parameter	**Value**	**Description**
Vehicle Density	[2,4] vehicle/km	Number of vehicles per km
Vehicle generation distribution	Poisson with Lamda = 0.7	Vehicle generation distribution
Vehicle dimensions	Width: 1.8m; Length 4.3m	Width and length of each vehicle generated
Maximum drivable lane Speed	30,50,75,100,130 km/h	Maximum Vehicle speed
Vehicle Acceleration	4 m/s^2	Maximum Acceleration
Vehicle Deceleration	7.5m/s^2	Maximum deceleration
Map size	2.22Km	Total map length for 4 veh/km and 2 veh/km spatial density
Simulation Step Length	0.01s	Mobility updates

5.3 Performance Metrics

In order to define the performance metrics, we first look at the various conditions under which we evaluate the collision avoidance system. These conditions are designed to test the working of the collision avoidance system with varying traffic density and standard maximum allowable speed limits. The Traffic density factor is chosen to compare the overall performance of the system with varying fleet capacity and their behavior under different speed conditions. Building upon the knowledge that as the speeds of the fleet increases, the risk of collision is higher due to the vehicle's reaction time and braking time contributions towards the vehicle coming to a complete halt state. When the maximum allowable speeds are increased, the vehicles will try to achieve a higher average fleet speed hence will travel at a higher speed individually, which enhances their tendency to collide with other vehicles with similar behavior. In this condition, it is harder for the vehicles to stop in a short duration due to their high driving speeds and the algorithm tends to inform the vehicles too late of a collision warning due to a standard braking reaction time considered (0.05s).

The traffic management of the traffic fleet is measured in terms of the average speeds that the vehicles are keeping up with to achieve an efficient higher average speed which is below the maximum allowable speeds for the given topology. Furthermore, these speed values will help us draw a relationship between the average waiting time and travelling time of the vehicles, to analyze how to minimize both of these without compromising on the fleet's speed efficiency. This study leads us to study the impact of a safety algorithm such as the collision avoidance algorithm discussed here has on the overall moving traffic efficiency with parameters such as time headway, waiting time, travelling time and average speed of the vehicle. We compare some of these results with the performance of the enhanced collision avoidance algorithm as described in [9]. The algorithm and the results of this research work have been reproduced for this comparison to be as accurate as possible.

5.3.1 Collision Avoidance System

The core of the collision avoidance system works on assessing the probable collisions based on a predicted future trajectory of the vehicles nearing an intersection and avoiding them by applying preconditioned actions, which are deceleration, or braking events triggered, as described in Chapter 4. In this discussion, we will consider the parameters relating to collision avoidance as Application-Related Metrics. By studying these metrics, we aim to understand the effectiveness of the enhanced collision avoidance algorithm on the overall traffic conditions, its performance with varying spatial traffic density, and compare the results with those presented for the enhanced collision avoidance strategy algorithm in [9] to justify how the implementation of the Variable Time Headway and Spacing strategy has impacted the application performance.

The first metric that we define is the number of collisions that have occurred despite the collision avoidance algorithm being engaged in the traffic network. This metric's performance and values are subject to various traffic parameters such as the overall average speed of the vehicles, the maximum allowable speed in the given road

topography, the reaction time defined for the vehicles, and the spatial density in the traffic fleet. Along with this, we also discuss the percentage of **collisions avoided** which is defined as the percentage of number of collisions that are avoided when collision avoidance system is active, compared to the overall number of collisions that have occurred during the duration of simulation run.

The second metric that we study is the average vehicle speed that the traffic fleet can attain given a particular maximum allowable speed for the lane in which the vehicle is traveling. In our thesis work, we are only considering intersectional collisions because, as discussed before, the NHTSA report claims that the maximum number of collisions that occur in unregulated or regulated traffic conditions are attributable to intersectional collisions. Having established the traffic architecture that, we are considering for these simulations, the road topology has several intersections with both 3-way T-junctions and 4-way intersections. Thus, we study how many of these collisions take place at these intersections at a given maximum speed for the vehicles to traverse. Then, we draw a correlation to study the trend of average speed kept by these vehicles and the collisions that are occurring among them during the entire run time. We hypothesize that an increase in the spatial density corresponds to a higher number of vehicles waiting in the queues created at intersections. This will lead to vehicles taking longer to restart and thus result in decreasing the average speeds of the vehicles.

Lastly, we compare both these factors with increasing and decreasing spatial density. In one case, we consider a spatial density of 2 vehicles per kilometer, and in the second case, we consider 4 vehicles per kilometer. For the varying spatial densities, we compare the average speed at which the vehicles are driving with the number of collisions that are occurring when the average speed of all the vehicles is a certain value. With this, we establish how the number of collisions is a factor of the average speed at which the vehicles are trying to run.

5.3.2 Traffic Management

In this thesis, we evaluate the performance of the enhanced collision avoidance algorithm by analyzing key traffic management parameters, focusing on the impact of the time headway and spacing control strategy. Through simulations, we collect data to assess the waiting times of vehicles at intersections and the total travel times of the fleet under varying spatial densities and average speeds.

Relationship between Waiting Times and Spatial Density:

One of the primary objectives is to establish a relationship between the waiting times of vehicles and the spatial density of the fleet. We hypothesize that as the average speed of the fleet increases and the spatial density is high, the waiting time of vehicles will change linearly with the average speed. This is due to the increased likelihood of congestion at intersections when the collision avoidance algorithm detects potential collisions and initiates vehicle stoppage. Once a collision is detected, the involved vehicles decelerate and perform appropriate maneuvering actions before regaining their desired speed. The time taken for this process constitutes the waiting time of the vehicle.

We expect that the average waiting time of the fleet will decrease with an increase in the average speed when the spatial density is reduced. With fewer vehicles on the road, the probability of congestion caused by detected collisions is lower compared to scenarios with higher spatial density. As a result, vehicles can reach their maximum speed limits more quickly after deceleration due to fewer obstacles in their path.

Total Travel Time Analysis:

The total travel time of the fleet is a function of both spatial density and vehicle speed. It represents the sum of the time taken by each vehicle to complete its journey from origin to destination, including waiting times at intersections due to collision avoidance maneuvers and the time spent traveling at the desired speed.

As spatial density increases, the number of vehicles on the road network rises, leading to a higher probability of congestion and collisions. Consequently, the collision avoidance algorithm will detect more potential collisions, causing vehicles to decelerate and perform avoidance maneuvers more frequently. This increase in the number of maneuvers results in longer waiting times at intersections, ultimately increasing the total travel time of the fleet.

Conversely, as the average speed of the vehicles increases, the time spent traveling between intersections decreases. However, higher speeds also increase the likelihood of collisions, particularly at intersections, which may lead to more frequent collision avoidance maneuvers and longer waiting times.

To gain a deeper understanding of the relationship between spatial density, average speed, and the resulting waiting times and total travel times, we will analyze the simulation results under various conditions. By comparing the performance metrics for

different scenarios with respect to the traffic fleet density, we aim to determine the optimal balance between spatial density and average speed that minimizes the total travel time while ensuring safety through the enhanced collision avoidance algorithm.

The simulation results will provide valuable insights into how the time headway and spacing control strategy employed in the enhanced collision avoidance algorithm affects the overall performance of the traffic network. By examining the trends in waiting times and total travel times, we can assess the effectiveness of the algorithm in managing traffic flow and mitigating congestion while prioritizing vehicle safety.

5.3.3 Network Metrics

The NS-3 LENA setup is designed to emulate a real-world scenario of a Vehicle-to-Infrastructure (V2I) communication network through wireless 5G connectivity. This setup allows us to study and analyze the network parameters involved in the data communication between the vehicle's client application and infrastructural server applications through the simulation setup discussed earlier. The network metrics that we mainly focus on in this thesis are built upon the network metrics that were quoted and analyzed while studying the enhanced collision avoidance algorithm in [9]. These network metrics are traffic load, latency, and packet delivery ratio. In this section, we define these metrics further and explore their correlation with varying spatial density to establish the expected data analysis pattern from the data obtained through the simulations run for the experimentation in this research work.

Traffic load is defined as the network traffic generated by the eCA service, which is denoted in kb/s. Traffic load is measured and analyzed for both uplink and downlink traffic. The uplink traffic load is measured and analyzed for the CAM messages that are broadcasted by the vehicular nodes to the client edge server node. The uplink traffic load is evaluated only based on the CAM messages, which in our study have been generated at a fixed dissemination rate of 10 Hz. This value is independent of factors such as the speed of the traffic fleet, braking reaction time, or DENM received. The only factor it depends on is the number of vehicles in the simulation, in other words, the spatial traffic density. Thus, we expect that with an increase in spatial density, there will be an increase in the uplink traffic load. We study this parameter by evaluating the number of CAM messages disseminated per second throughout the traveling time of the vehicles.

The downlink traffic load is evaluated in a similar way but for the DENMs sent by the server application on the event of a probable collision occurring. Thus, the downlink traffic is always lower (by one or two degrees) than the uplink traffic since the DENM messages are only generated upon the occurrence of an event (collision). We compare the downlink traffic for different spatial densities since an increase in spatial density translates to an increase in the probability of a collision occurring, and hence, this means that there is an increase in the number of DENMs being transmitted by the server application to the vehicular node to avoid or mitigate a collision. Thus, we plot the data for different spatial densities based on the varying allowable maximum drivable speed for the vehicles, since this also directly affects the collision occurrence probability. Therefore, with an increase in the number of vehicles and the maximum speed, there is a high likelihood that dangerous situations can occur, leading the eCA service to react with a DENM.

The next parameter that we study is latency. Latency is defined as the difference between the time at which a CAM is sent from a vehicular node to the moment at which the corresponding DENM is received. The value is evaluated considering the timestamp that is included in the DENM and CAM structure. Thus, this factor accounts for the delay that the service is experiencing in providing an appropriate reaction to the vehicular nodes in case of a collision occurrence.

Ideally, we expect the latency to be minimal to none; however, due to network factors such as *fading, Doppler effect*, ([9], [44]) and unforeseen false negatives wherein a probable collision is not detected by the algorithm or is detected with a delay, latency may occur. Latency is of two types: uplink and downlink latency. Uplink latency is evaluated as a client-to-server latency, where the server uses the timestamp in the CAM message structure every time it is received to calculate the latency between that timestamp and the timestamp at which the DENM is sent to the client from the server, if the DENM is triggered for the information being evaluated in that CAM message.

Downlink latency is evaluated on the client application in a similar way, that is, by evaluating the gap between the timestamp at which the CAM was sent and the moment at which the DENM was received from the server application. These values help us study the delay that is introduced in both uplink and downlink traffic. These values are independent of speed, spatial density, or the collisions that are occurring. They are specific to network dependencies such as the communication channel used or the processor delay both on the server side and on-board units of the vehicular nodes.

The third metric that we consider for the evaluation of the collision avoidance system is the Packet Delivery Ratio (PDR). PDR is defined as the ratio between the number of packets sent by an entity to the number of packets it receives. For uplink PDR, the ratio is evaluated as the number of CAM messages that are sent by the vehicle to the number of CAM messages which are received by the server. This metric is useful to study the packet drop tendency of the network. The downlink PDR is evaluated by calculating the number of DENMs sent by the server during a simulation cycle to the sum of the number of DENMs received by all the vehicles in the simulation, within the radius of operation of the server. This metric does not depend upon maximum speed or the number of collisions occurring. Hence, not many variations of this data are analyzed to justify its dependency on the factors mentioned before.

In the next subsection, we will dive into the results of the simulation experiments carried out to study these performance parameters and enable the collision avoidance algorithm to be implemented with the variable time headway strategies. In the consequent chapters, we will discuss these results and justify the analysis based on the ideal or expected behavior of the performance parameters as discussed in this section. We will also discuss how, in future work, we can identify and correct these performance parameters to tune them for near-ideal performance of the collision avoidance system.

5.4 Simulation Results

The experimentation for simulating and synthesizing the results presented in this section is carried out by obtaining data from 50 simulation runs, each lasting 3600 seconds. The CAM dissemination rate is kept constant at 10 Hz throughout the simulations. Within the SUMO routing files, the lanes and road architectures are kept consistent, as explained in the previous sections. The spatial density is varied by adjusting the number of vehicles simulated on the road topology. The results presented in this section are divided into three subsections, correlating to the performance parameters discussed earlier: Collision Avoidance, Traffic Management, and Network Parameters.

The Collision Avoidance subsection presents data plots and results from simulations run for different maximum drivable speeds and spatial densities. The data is recorded per simulation step, which is defined by SUMO's simulator as 0.01s. A total of 6 variant cycles of data are recorded for each maximum drivable lane speed set for the road topology. This process is repeated twice to obtain comparable results for the system running with spatial densities of 4 vehicles/km and 2 vehicles/km. To provide a comprehensive analysis, we have also reproduced the results of the enhanced Collision Avoidance system as implemented in [9], allowing for a direct comparison with the enhanced results obtained from our implementation.

The Traffic Management parameters are a novel addition to the network and application parameters. These are evaluated for each vehicle individually to provide a microscopic analysis of each vehicle's travel and waiting time. A macroscopic analysis is also conducted by evaluating and analyzing the overall performance of the vehicles with respect to their travel times and waiting times, considering varying maximum speeds, average fleet speeds, and spatial density.

The Network Parameters are evaluated based on the theoretical knowledge obtained from the literature review of networking constraints and their importance in a system such as Collision Avoidance. The data is obtained in a similar manner as described above, enabling us to study and analyze the role that networks, and their accuracy play in safety algorithms like Collision Avoidance.

The results which are represented as curves, or trend-line plots are presented with the Standard Error Mean (SEM) values evaluated for the data collected during the above explained simulations. The SEM is a measure of the precision of the mean estimate, providing information about the variability of the sample means.
The SEM is calculated by dividing the standard deviation of the sample by the square root of the sample size. This calculation takes into account the variability of the data and the sample size, providing a more accurate representation of the true mean of the population. By plotting the mean values along with their corresponding SEM values, we can visualize the confidence intervals around the mean estimates. In the plots presented, the error bars represent the SEM values, indicating the range within which the true mean is likely to fall. Smaller error bars suggest higher precision and reliability of the mean estimates, while larger error bars indicate greater variability and uncertainty. By considering the SEM values, we can make more informed conclusions about the significance of the observed differences and trends in the data.

By examining the results from multiple perspectives, including collision avoidance, traffic management, and network parameters, we aim to present a holistic evaluation of the system's effectiveness and identify potential areas for further optimization.

5.4.1 Collision Avoidance

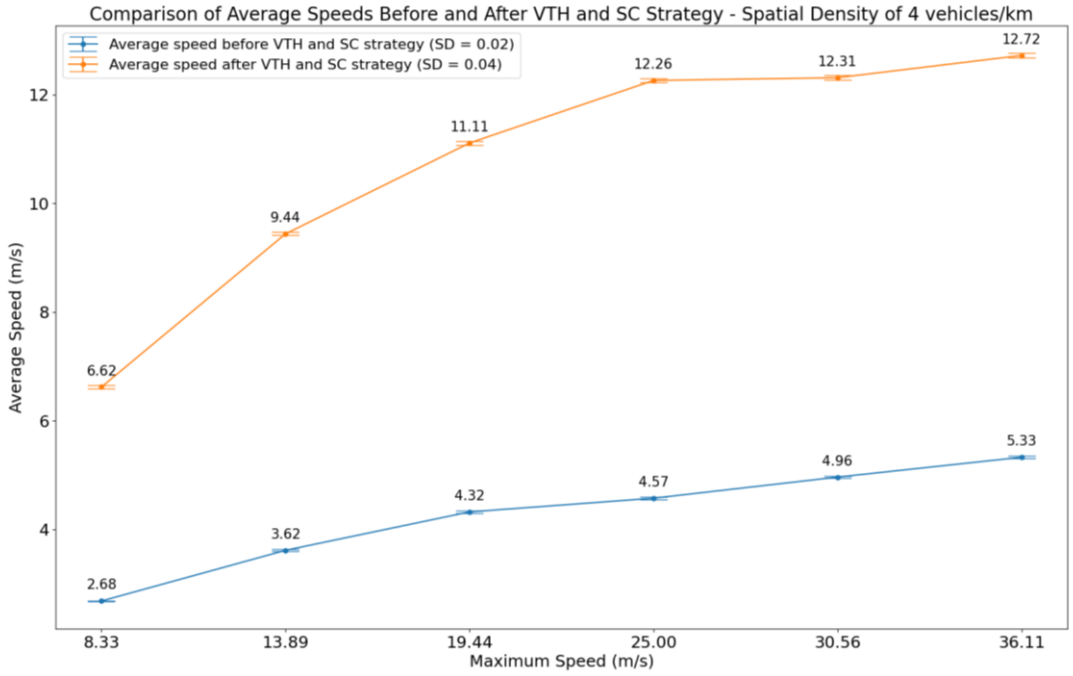

Figure 29 Average speed maintained by vehicles show an improvement with implementation of Variable time headway and spacing control strategy (Spatial density of 4 vehicles/km).

In Figure 29, the average speeds of vehicles are plotted against the maximum allowable speed limits set for the road topology through which the vehicles are traversing. This figure shows a comparison between the average speeds maintained by the vehicles before the Collision Avoidance system was implemented with the Variable Time Headway strategy and after its implementation. The spatial density considered in the Figure 29 is 4 vehicles/km. Upon examining the graph, we can observe a linear relationship between the average speeds maintained by the vehicles and the maximum allowable speed limits in both cases. As the maximum speed limit increases, the average speed of the vehicles also increases proportionally.

However, a notable difference can be seen in the average speeds maintained by the vehicles after the implementation of the Collision Avoidance with the variable time headway strategy compared to the speeds before its implementation. At lower maximum speed limits, such as 8.33 m/s and 13.89 m/s, the difference in average speeds is relatively lesser than at higher maximum speeds. For instance, at a maximum speed limit of 8.33 m/s, the average speed before the implementation is 2.68 m/s, while after the implementation, it is 6.62 m/s, showing an improvement.

As the maximum speed limits increase to higher values, such as 19.44 m/s, 25.00 m/s, and 27.78 m/s, the difference in average speeds becomes more significant. At a maximum

speed limit of 19.44 m/s, the average speed before the implementation is 4.32 m/s, while after the implementation, it increases to 11.11 m/s, indicating a substantial improvement in the average speed maintained by the vehicles.

This trend continues for the highest maximum speed limits considered in the plot. At a maximum speed limit of 36.11 m/s, the average speed before the implementation is 5.33 m/s, while after the implementation, it reaches 12.72 m/s, showcasing a notable increase in the average speed of the traffic fleet.

The improved performance in terms of average speeds after the implementation of the Collision Avoidance system with the variable time headway strategy can be attributed to the optimization of the time headway between vehicles. The variable time headway and spacing control based strategy which attributes to the adaptive acceleration of individual vehicles to be evaluated allows the collision avoidance system's client application to consistently maintain a higher speed limit in the traffic fleet, thus reducing unnecessary false braking, optimizing the spacing between the vehicles in the fleet and increasing the fleet speed. Hence, by dynamically adjusting the time headway based on the traffic conditions and the maximum speed limits, the Collision Avoidance system allows vehicles to maintain higher average speeds while ensuring safety.

This plot demonstrates the effectiveness of the Collision Avoidance system with the variable time headway strategy in improving traffic flow efficiency and enabling vehicles to maintain higher average speeds, especially at higher maximum speed limits. The results suggest that the implementation of this strategy can lead to a more efficient and faster-moving traffic fleet under the given conditions of speed limits and spatial density.

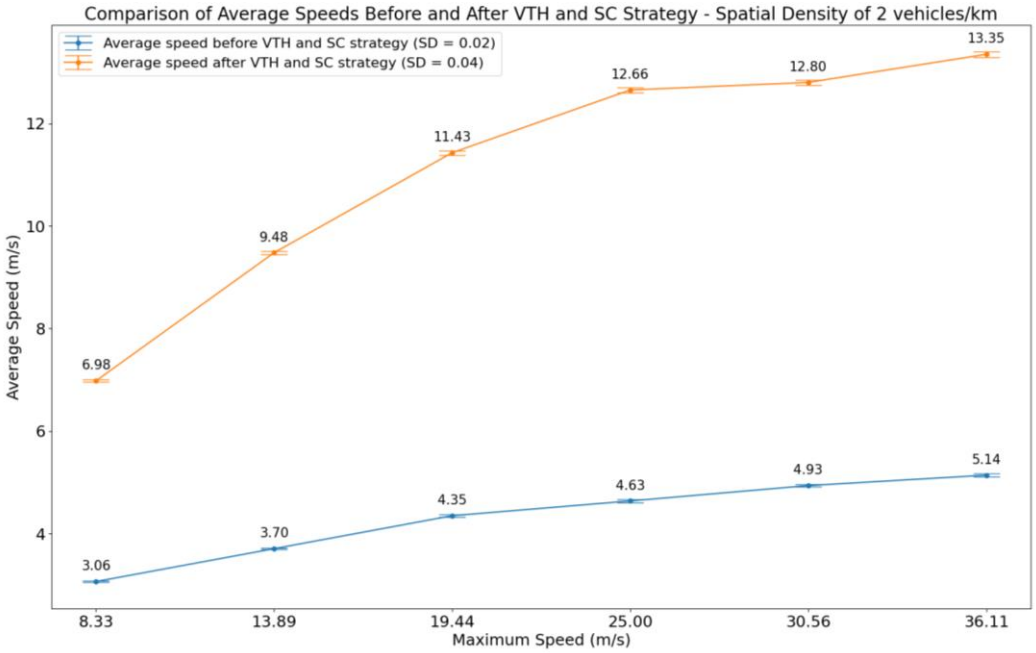

Figure 30 Average speed maintained by vehicles show an improvement with implementation of Variable time headway and spacing control strategy (Spatial density of 2 vehicles/km).

In Figure 30, the average speed of the traffic fleet with the density of 2 vehicles/km are plotted against the maximum allowable speed limits set for the road topology through which the vehicles are traversing, like Figure 29. The comparison is made between the average speeds maintained by the vehicles before and after the implementation of the Collision Avoidance system with variable time headway strategy.

As observed in the previous figure (Figure 29), there is a linear relationship between the average speeds maintained by the vehicles and the maximum allowable speed limits in both cases. The average speeds increase as the maximum speed limits increase. Notably, the difference in average speeds before and after the implementation of the enhanced Collision Avoidance with variable time headway strategy is more pronounced in this scenario with lower spatial density.

At lower maximum speeds of 8.33 m/s and 13.89 m/s, the difference in average speeds is relatively lower. However, as the maximum speed limits increase to 19.44 m/s, 25.00 m/s, and 30.56 m/s, the gap between the average speeds before and after the implementation of the enhanced Collision Avoidance with variable time headway strategy widens significantly. This indicates that the improved Collision Avoidance system is more effective in maintaining higher average speeds, especially at higher maximum speed limits, when the spatial density is lower.

The lower spatial density of 2 vehicles/km reduces the likelihood of congestion and allows vehicles to maintain higher average speeds compared to the scenario with 4

vehicles/km. The implementation of the Collision Avoidance system with variable time headway strategy further enhances the traffic flow efficiency, enabling vehicles to travel at higher average speeds while maintaining safety.

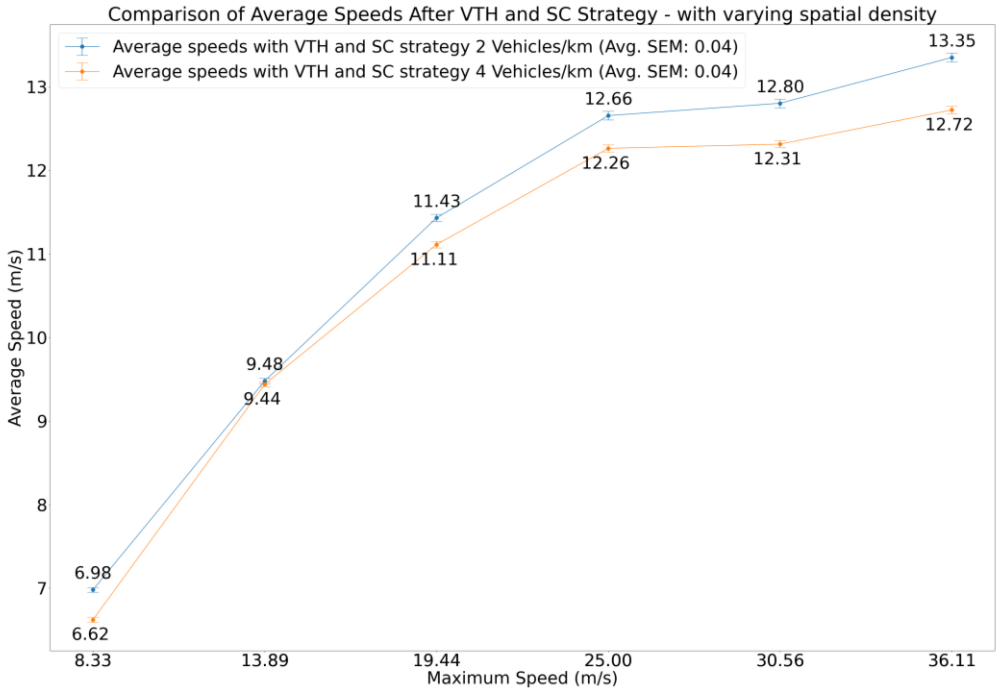

Figure 31 Comparison of Average speed maintained by vehicles show an improvement with decrease Spatial density from 4 and 2 vehicles/km.

Figure 31 presents a comparison of the average speeds maintained by vehicles for two different spatial densities: 4 vehicles/km and 2 vehicles/km. The plot shows the average speeds achieved by the vehicles after the implementation of the Collision Avoidance with variable time headway strategy for both spatial densities across various maximum speed limits.

As expected, the average speeds maintained by vehicles are consistently higher for the lower spatial density of 2 vehicles/km compared to 4 vehicles/km. This trend is observed across all maximum speed limits, from 8.33 m/s to 36.11 m/s. At lower maximum speed limits, such as 8.33 m/s and 13.89 m/s, the difference in average speeds between the two spatial densities is relatively small. However, as the maximum speed limits increase, the gap between the average speeds for 4 vehicles/km and 2 vehicles/km becomes more significant.

For example, at a maximum speed limit of 19.44 m/s, the average speed for 4 vehicles/km is approximately 11.11 m/s, while for 2 vehicles/km, it is around 11.43 m/s. This difference becomes even more pronounced at higher maximum speed limits, such as 25.00 m/s and 36.11 m/s. The higher average speeds observed for the lower spatial density can be attributed to reduced traffic congestion and fewer interactions between

vehicles. With fewer vehicles on the road, there is less likelihood of collisions, allowing vehicles to maintain higher speeds. The Collision Avoidance with variable time headway strategy further optimizes the traffic flow, enabling vehicles to travel at higher average speeds while ensuring safety. This comparison highlights the impact of spatial density on the average speeds maintained by vehicles and demonstrates the effectiveness of the Collision Avoidance with variable time headway strategy in improving traffic flow efficiency, particularly at lower spatial densities and higher maximum speed limits.

Even though the maximum allowable speeds are as high as 36.11m/s, it should be noted here that the average speeds remain within 14m/s for both the spatial densities experimented over the road topology with 10 intersections. These lower average speeds signify that even though we are aiming at attaining a higher speed, this is not at the cost of safety. The repeated stoppages and breaking when collision detector warn the vehicles involved in the probable collisions stand as testament to these reduced speeds of the traffic fleet overall. Thus, prioritizing safety over the traffic fleet's speed efficiency with the given maximum allowable speeds.

In the average speeds resulting from the demonstration of both the spatial densities, we can observe that the randomness of the average speed data pertaining to specific maximum speeds is lower and the standard deviation is minimal in both the cases. This is due to the fixed road infrastructure that we are simulating these results with, keeping the lane speed limits constant and the vehicles behave very uniformly in these lanes. This also goes to say that the algorithm can be scaled to different road topologies, and it can adapt to varying speed limits with maximum accuracy and less entropy in the average speeds maintained by all the vehicles.

Next, to study the number of collisions that occur at varying maximum speed limits, we examine the data under similar conditions of varying spatial densities of 4 vehicles per km and 2 vehicles per km. This analysis allows us to understand the effect that the maximum speed limit has on the number of collisions occurring at intersections, while the vehicles are trying to traverse with an optimal traffic flow relative to the maximum speed limits set. For this study, we plot histograms where each bar shows the data values collected during the each simulation relative to the maximum drivable speed. The number of collisions that occur vary from simulation to simulation due to the random generation and traversal of the vehicles in the SUMO simulated road architecture, hence plotting these collisions in a bar plot is more beneficial to study an overall macroscopic trend of increase or decrease in the number of collisions for varying maximum speeds and spatial densities.

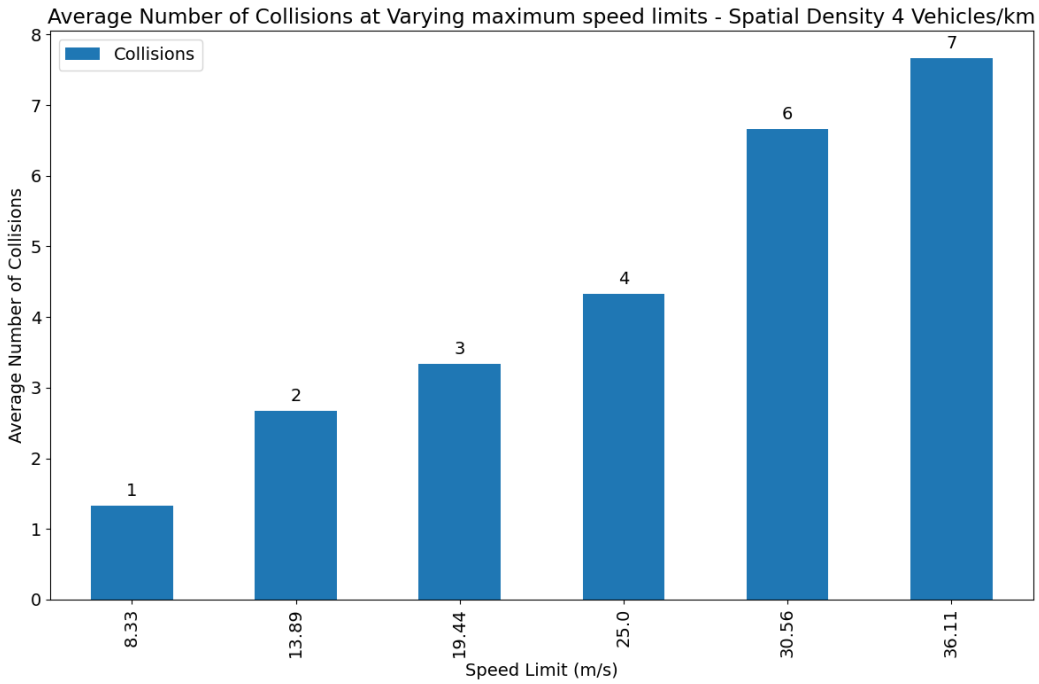

Figure 32 Number of collisions at varying maximum drivable speed limit for 4 vehicles/km.

The bar graph in Figure 32 illustrates the number of collisions that occurred at different maximum drivable speeds for a spatial density of 4 vehicles/km. As the speed limit increases from 8.33 m/s to 36.11m/s m/s, there is a notable upward trend in the number of collisions. At the lowest speed limit of 8.33 m/s, there were 1 collision recorded. However, when the speed limit increases to 25.0 m/s, the number of collisions surges to 4, highlighting the significant impact that even a moderate increase in allowable speed can have on collision frequency at this spatial density. The collision count remains high at 7 for the 36.11 m/s speed limit. An average of 7 collisions occurring through the simulations 50 carried out at 36.11 m/s maximum speed limit are still considerably higher than the number observed at the lowest speed limit of 8.33 m/s. This graph clearly demonstrates a strong positive correlation between higher speed limits and increased collisions at a spatial density of 4 vehicles/km. By increasing the vehicle speeds, it becomes harder for the vehicles to stop within a short duration, and the algorithm takes longer to inform the vehicles for deceleration. The high collision counts at faster speeds suggest that the collision avoidance algorithm may require further tuning to effectively handle this density of traffic in this aspect.

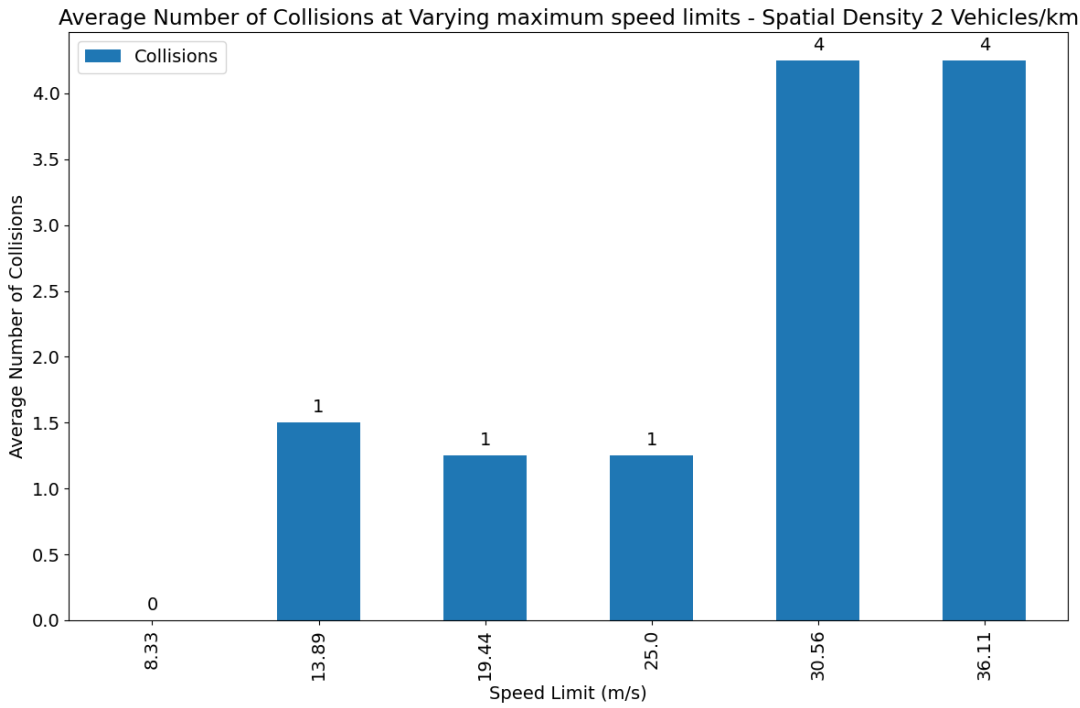

Figure 33 Number of collisions at varying maximum drivable speed limit for 2 vehicles/km

In Figure 33, the graph displays collision data for a lower spatial density of 2 vehicles/km across the same range of speed limits as Figure 32. Although there is still a general upward trend in collisions as the speed limit increases, the total number of collisions is substantially lower compared to the 4 vehicles/km density scenario. At the minimum speed limit of 8.33 m/s, no collisions were recorded, and only average of 1 (~1.5) collision occurred at the 13.89 m/s and 19.44 m/s limit. The number of collisions increased to 4 at both the 30.56 m/s and 36.11 m/s speed limits. While the trend is like Figure 33, with higher speed limits resulting in more collisions, the lower spatial density evidently reduces the overall collision occurrence, even at high speeds. This suggests that the collision avoidance algorithm performs better at this reduced vehicle density compared to the higher density scenario. With a decrease in the spatial density, there is a decrease in the number of collisions since crowding in the intersections is lower as compared to that in higher spatial density.

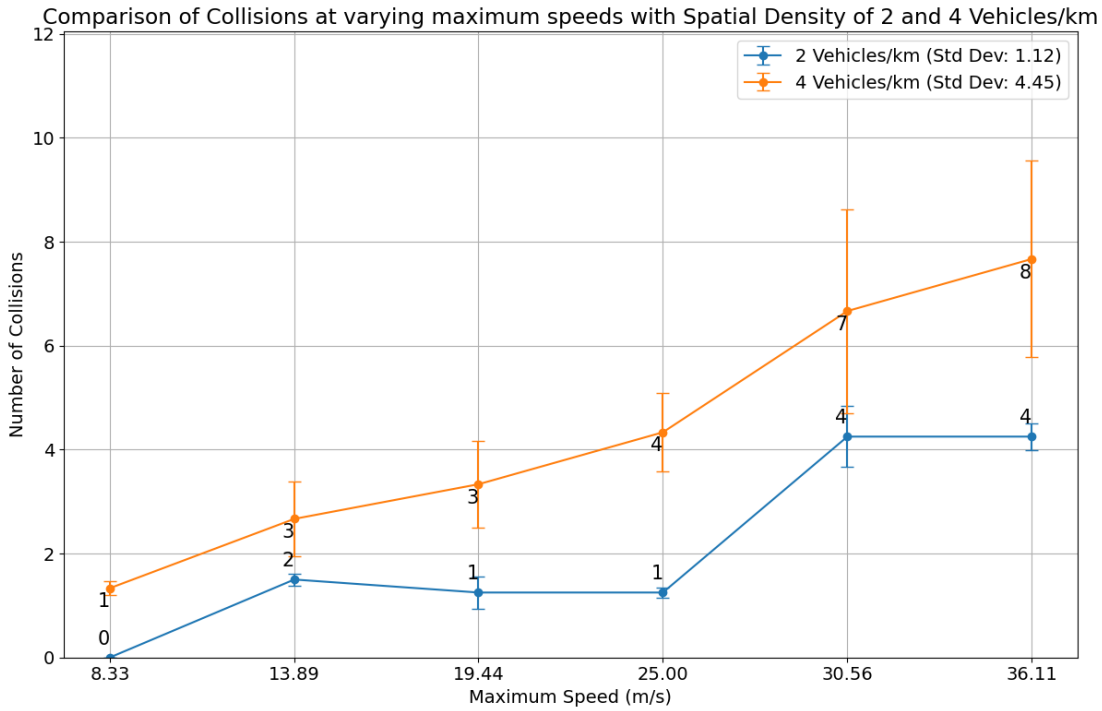

Figure 34 Comparison of Collision of 4 vehicles/km and 2 vehicles/km.

Figure 34 features a line graph with standard error mean bar that directly compares the collision counts at each speed limit for spatial densities of 4 vehicles/km and 2 vehicles/km. This graph clearly illustrates the consistently higher number of collisions that occur with denser traffic across all speed limits. At the lower speeds of 8.33 m/s and 13.89 m/s, there is a small difference, with an average of 1 to 3 collisions at 4 vehicles/km and 0 to 2 collisions at 2 vehicles/km. However, the gap widens significantly at 25.00 m/s, with over 4 collisions for the higher density compared to just 1 for the lower density.

The Standard Error Mean value demonstrates a higher error bar value along the collisions with 4 vehicles/km for all the maximum speed scenarios. The large Standard deviation value for higher spatial density is attributed to the large difference in the number of collisions occurring from simulation to simulation; however, it should be noted that even then the number of collisions occurring at higher spatial density is relatively more than that occurring at lower densities. The collision counts remain notably higher for 4 vehicles/km at the remaining speed limits, although the difference increases at the highest speed limit of 27.78 m/s. Nevertheless, there were still over 1.5 times more collisions with the denser traffic, even at this maximum speed limit. This comparison emphasizes that not only does increasing the maximum speed limit lead to more collisions, but the effect is amplified at higher traffic densities. The collision avoidance algorithm appears to handle sparser traffic reasonably well but struggles once traffic becomes denser, particularly at higher speeds. Further algorithm tuning focusing on these higher density scenarios seems necessary to mitigate collisions effectively.

Here, it should also be noted that all the collisions recorded are intersectional collisions, all these comparisons are made with same mobility traces with the same number of intersections. All three figures (Figure 32,33,34) above demonstrate that raising maximum speed limits increases the frequency of collisions, with the effect being much more pronounced for denser traffic. The collision avoidance system, in its current form, has room for improvement, especially in handling scenarios with a high density of fast-moving vehicles. Tuning the algorithm to account for higher spatial density could help reduce collisions at the 4 vehicles/km to be more in line with the results seen at 2 vehicles/km. This analysis provides valuable insights into the performance of the collision avoidance system and highlights the need for further optimization to ensure safety across a wide range of traffic conditions.

Next, we account for the percentage of collisions that were avoided. The trend allows us to study the performance of the collision avoidance system for it's functionality of how many collisions were avoided. These results are also comparable between the spatial densities of 4 vehicles/km and 2 vehicles/km. The percentages are individually evaluated for every case of maximum drivable speed limit that is set and for both these spatial densities with respect to the total number of collisions occurred during the simulations. The total number of collisions that occurred is significantly higher as recorded in the simulations run with spatial density of 4 vehicles/km in the road network as compared to lower spatial density. However, we observe a linear and similar trend of percentage of collisions avoided with varying maximum speeds in both these cases where the percentage of collisions avoided is the most at lower average and maximum drivable speeds as opposed to the least percentage of collisions avoided at higher speeds.

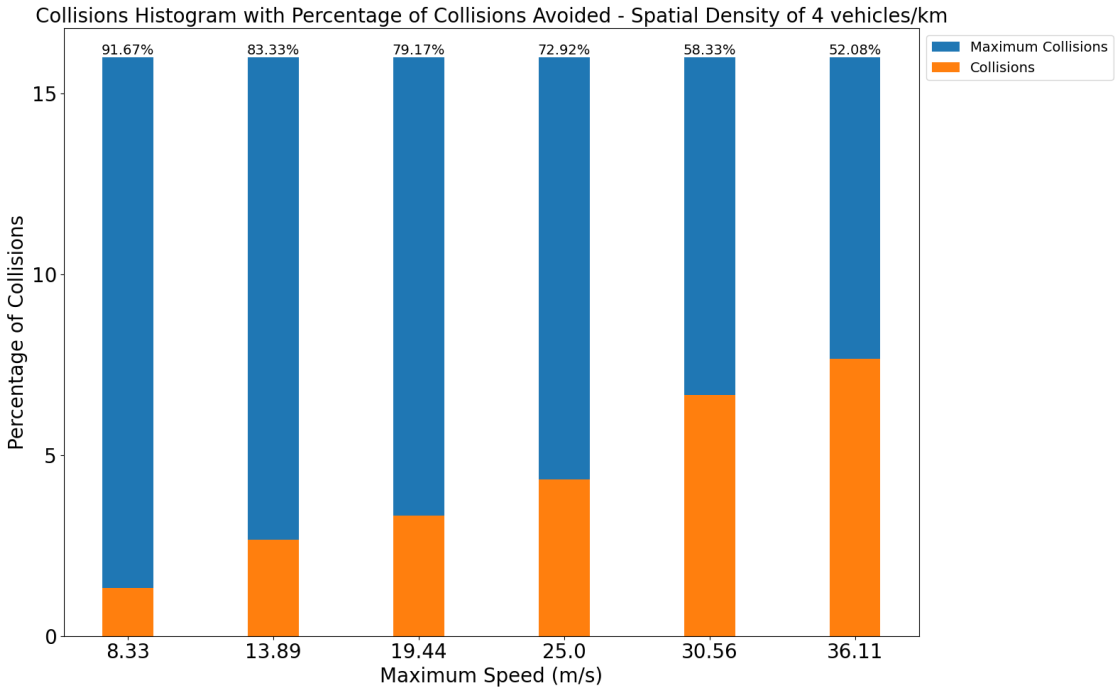

Figure 35 Percentage of Collisions avoided with spatial density of 4 Vehicles/km

With a spatial density of 4 vehicles/km, the percentage of collisions avoided is the most at lowest speed of 8.33m/s with 91.67% of collisions avoided. The number significantly and linearly decreases following the study and explanation pertaining to linear relationship between the number of collisions occurring, average fleet speed and maximum drivable speeds. The least percentage of collisions that are avoided is at 36.11m/s with only 52.08% of collisions being avoided.

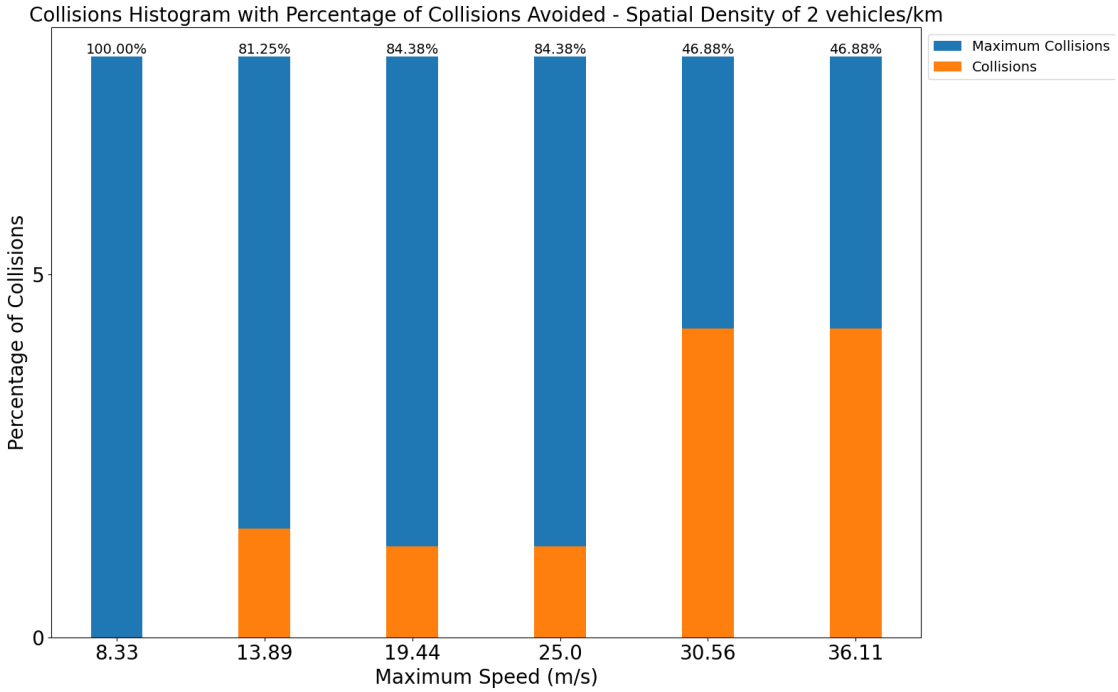

Figure 36 Percentage of Collisions avoided with Spatial Density of 2 Vehicles/km

The trend seen in Figure 35 also applies to Figure 36, where the percentage of collisions avoided is the most at lower maximum speeds. However, there is more scope of improvement on the higher maximum speeds in both the cases.

In the context of studying the performance of collision avoidance algorithms in connected vehicles, it is crucial to understand the relationship between the average speed of the vehicle fleet and the occurrence of collisions under different maximum drivable speed limits. The results section of the research work [9] and [10] offer valuable insights into this correlation, which can be further analyzed through visual representations of the data that we have collected.

To better illustrate the relationship between average speed and collisions, we present two plots that showcase the trends observed in the simulation results. These plots depict the average speed of the vehicle fleet and the average number of collisions occurring at different maximum drivable speeds, allowing for a clear visualization of the correlation between these variables. By examining the plots, we can gain a deeper understanding of how the collision avoidance algorithm performs under various traffic conditions and speed limits.

The relationship between average speed and collisions is generally found to be linear, meaning that the probability of collisions increases proportionally with the average speed of the traffic fleet. This correlation can be attributed to the increased risk associated with higher vehicle speeds, as faster-moving vehicles have shorter reaction times (0.05s in our case) and require greater stopping distances to avoid collisions.

Now, to draw a correlation between the average speed of the fleet and the number of collisions occurring through the maximum drivable speed that the simulation is limited to, we plot the average speed line against the maximum drivable speed and the number of collisions line to compare the with average speed line. This approach allows us to visually assess the impact of varying speed limits on both the average speed and collision frequency, providing valuable insights into the performance of the collision avoidance algorithm.

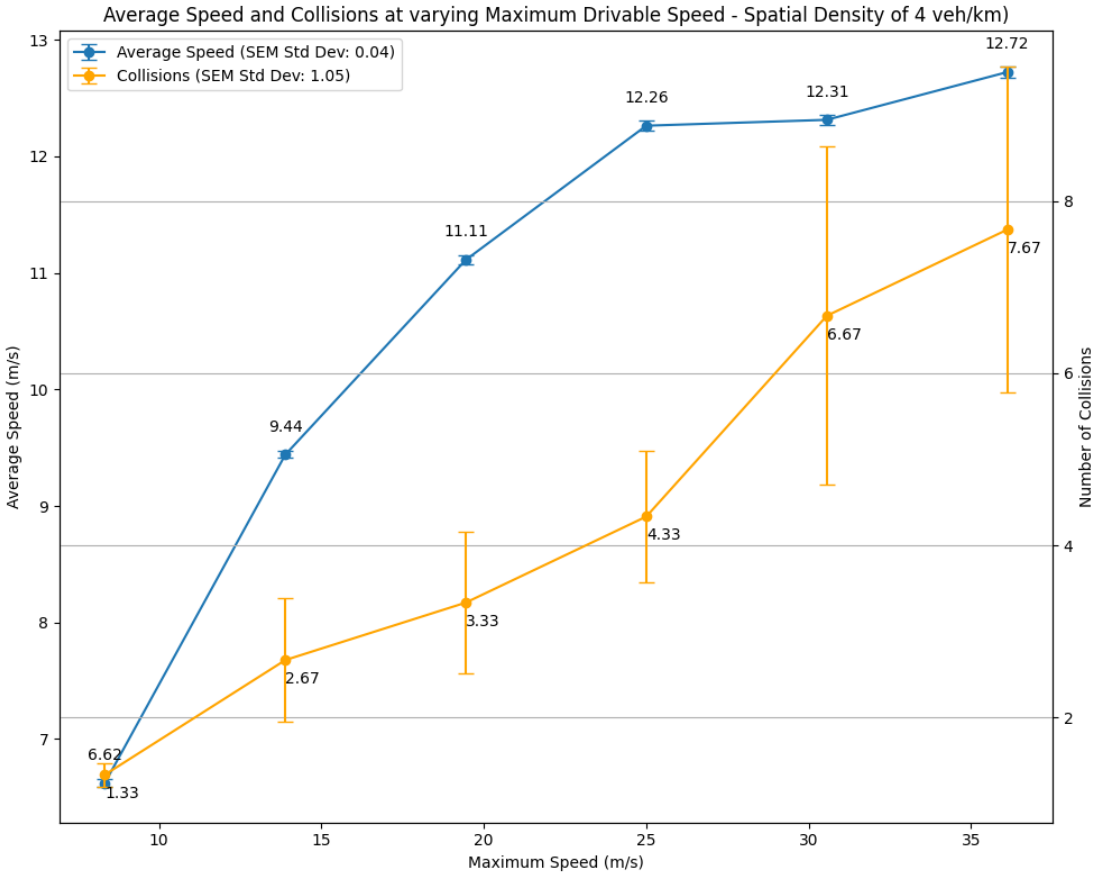

Figure 37 comparing number of collisions occurred with average speed for the spatial density of 4 vehicles/km.

Figure 37 shows the comparison of average speed and collisions for a spatial density of 4 vehicles/km. As the maximum drivable speed increases from 8.33 m/s to 36.11 m/s, we observe a steady increase in the average speed of the vehicle fleet. However, this increase in average speed is accompanied by a corresponding rise in the number of collisions. The plot reveals a strong positive correlation between the average speed and the collision count, with both metrics exhibiting a similar upward trend as the maximum drivable speed increases.

The trend in Figure 37 aligns with the findings discussed in the enhanced Collision Avoidance explored in [9] and their results. Hence, we note that raising the maximum

speed limit leads to a higher frequency of collisions, and this effect is more pronounced for denser traffic scenarios. The high collision counts at faster speeds suggest that the collision avoidance algorithm may require further tuning to effectively handle situations with a higher density of fast-moving vehicles.

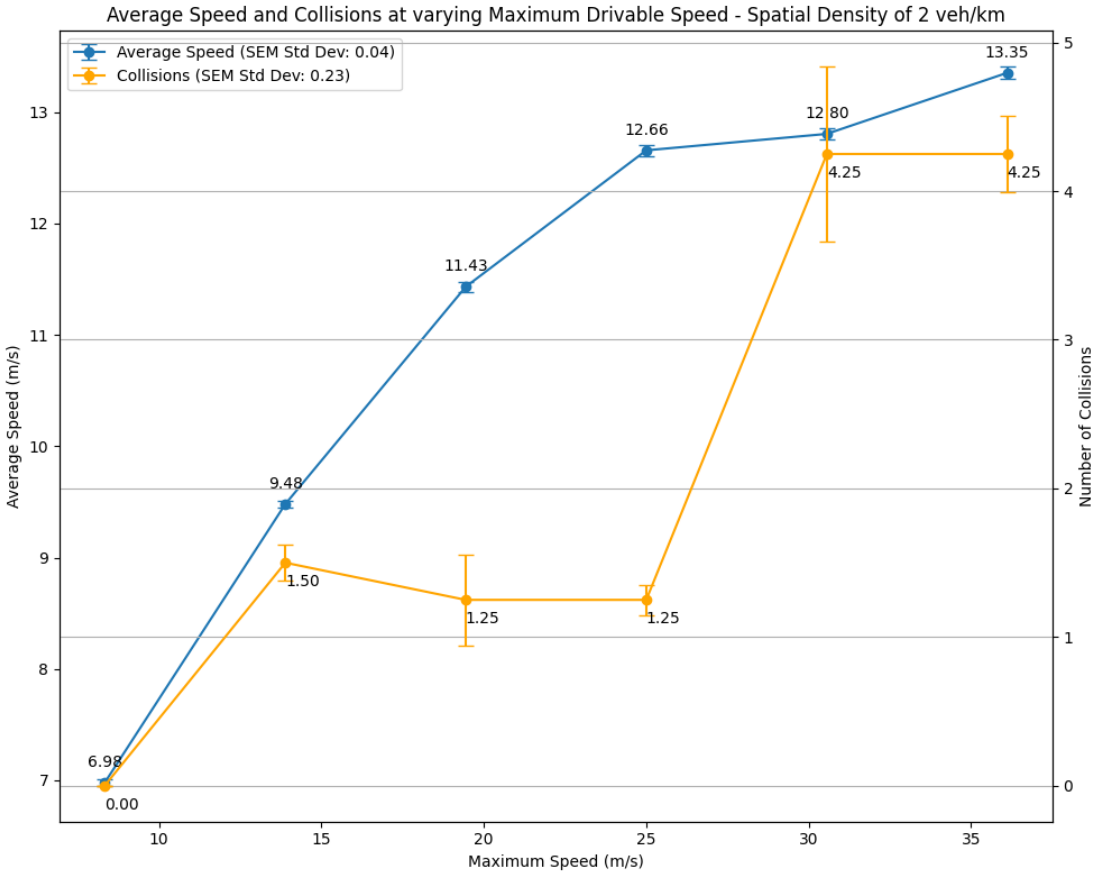

Figure 38 Comparison between average speed and collisions at 2 vehicles/km.

Figure 38 presents a similar comparison of average speed and collisions, but for a lower spatial density of 2 vehicles/km. While the average speed still increases with the maximum drivable speed, the number of collisions remains relatively low compared to the 4 vehicles/km scenario. The collision count exhibits a more gradual increase, indicating that the collision avoidance system performs better in less dense traffic conditions.

The results in Figure 38 are also consistent with the observations made in [9] and demonstrates the collision avoidance algorithm can handle sparser traffic more effectively, even at higher speeds. The lower spatial density reduces the overall collision occurrence, as there are fewer interactions between vehicles.

The comparative analysis of Figure 37 and Figure 38 highlights the significant impact of spatial density on the relationship between average speed and collisions. Higher traffic

density amplifies the effect of increasing the maximum speed limit, resulting in a greater number of collisions. This finding underscores the need for the collision avoidance system to be optimized for various traffic scenarios, particularly those involving dense, fast-moving vehicles.

These results can also demonstrate that comparison between the average speeds and collisions reveal that although the algorithm is exercising speed boosting techniques for the traffic fleet, this is not done at the cost of safety. That is, when the required warnings are issued by collision detectors to the vehicles which take actions such as decreasing the speed or stopping for the other vehicle involved in the collision to pass, the speed does not drastically increase. This can be better understood by the microscopic analysis in the following subsection.

In summary, the plots demonstrate a clear linear correlation between the average speed of the vehicle fleet and the number of collisions across different maximum drivable speeds. The effect of this correlation is more pronounced in higher density traffic, as evidenced by the steeper increase in collisions for the 4 vehicles/km scenario compared to the 2 vehicles/km case. These results emphasize the importance of tuning the collision avoidance algorithm to account for varying spatial densities and speed limits to ensure optimal performance and safety.

5.4.1.1 Microscopic Analysis

In the realm of connected vehicles and intelligent transportation systems, understanding the overall performance and effectiveness of collision avoidance algorithms is crucial for ensuring the safety and efficiency of the entire traffic fleet’s flow. While macroscopic analysis provides valuable insights into the overall behavior of the traffic fleet, it is equally important to delve into the microscopic level to examine the individual behavior of each vehicle. Microscopic analysis allows us to assess the performance of the client application in the Variable Time Headway (VTH) and Spacing Control (SC) strategies, at a granular level. By investigating how individual vehicles respond to these strategies under different maximum drivable speeds, we can validate the macroscopic trends observed and gain a deeper understanding of the system's dynamics.

The importance of microscopic analysis lies in its ability to uncover the nuances and variations in vehicle behavior that may be overlooked in macroscopic studies. By examining the average speeds of individual vehicles before and after the implementation of the VTH and SC strategies, we can identify patterns, outliers, and specific instances where the client application's performance excels or requires further optimization. This level of detail is essential for fine-tuning the collision avoidance algorithms and ensuring that they can effectively handle a wide range of traffic scenarios.

Moreover, microscopic analysis allows us to validate the macroscopic trends observed in the context of different spatial densities. In the case of the 4 vehicles/km spatial density scenario, the macroscopic analysis reveals a positive correlation between the average speed of the traffic fleet and the maximum drivable speed, with the implementation of the VTH and SC strategies leading to improved average speeds. By examining the

microscopic behavior of individual vehicles, we can confirm whether this trend holds true at the vehicle level and identify any deviations or anomalies that may require further investigation. To conduct a comprehensive microscopic analysis, we will examine a set of data that depict the average speeds of individual vehicles before and after the implementation of the VTH and SC strategies at different maximum drivable speeds.

Furthermore, the microscopic analysis will enable us to identify any limitations or challenges faced by the client application in specific scenarios. For example, if certain vehicles show minimal improvement in average speed after the implementation of the VTH and SC strategies, it may indicate that the algorithms need to be refined to better accommodate the characteristics of those vehicles or the traffic conditions they encounter. By pinpointing these issues at the microscopic level, we can focus our efforts on optimizing the client application to ensure robust performance across a diverse range of vehicles and traffic situations.

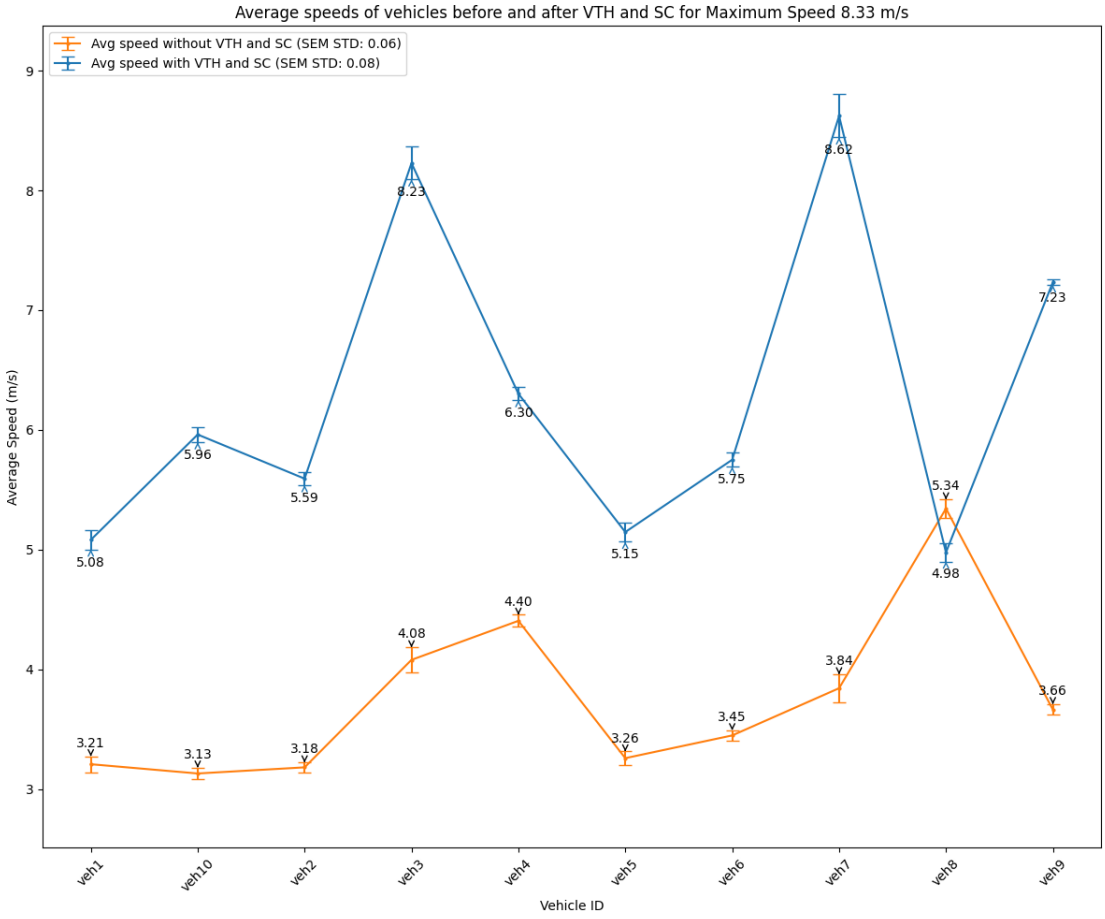

Figure 39 Average speed before variable time headway and average speed after variable time headway and spacing control strategy at the maximum drivable speed of 8.33 m/s for 10 vehicles in the simulation with spatial density 4 vehicles/km.

Figure 39 shows the average speeds of individual vehicles before and after implementing the Variable Time Headway (VTH) and Spacing Control (SC) strategies at a maximum drivable speed of 8.33 m/s. Most vehicles maintain higher average speeds after the VTH and SC implementation compared to their average speeds before. However, some vehicles show lesser changes in their behavior.

Particularly the average speed of the vehicle with ID veh8 after VTH and SC implementation does not show much improvement. The behavior is varied over 50 simulations that were taken; the average speed does not show improvement for this one vehicle. On further looking through the simulations, it is evident that this vehicle also showed the most waiting time at intersections, which is attributed to the random routing chosen by the simulation software. The vehicle with ID veh8 showed the *timeloss* to be the highest. Time loss is a factor which is defined as the time lost due to driving below the ideal speed or the maximum set speed[45]. This is contributed by the slowdowns due to intersections and the warnings that are issued by the collision avoidance system for this vehicle to stop or reduce it's speed in the scenario where a probable collision is detected between this vehicle and other vehicles in the fleet. In this case, we observe that safety of the vehicle is prioritized over the speeding action to contribute towards overall traffic fleet's speed efficiency.

However, the overall trend of the average speeds of majority of the vehicles supports the hypothesis that the average speeds of the vehicles is greater after the VTH and SC is implemented to collision avoidance system. This microscopic analysis aligns with the macroscopic trend observed for the 4 vehicles/km spatial density, where the average speed of the entire fleet improves slightly after implementing the VTH and SC strategies at this maximum drivable speed.

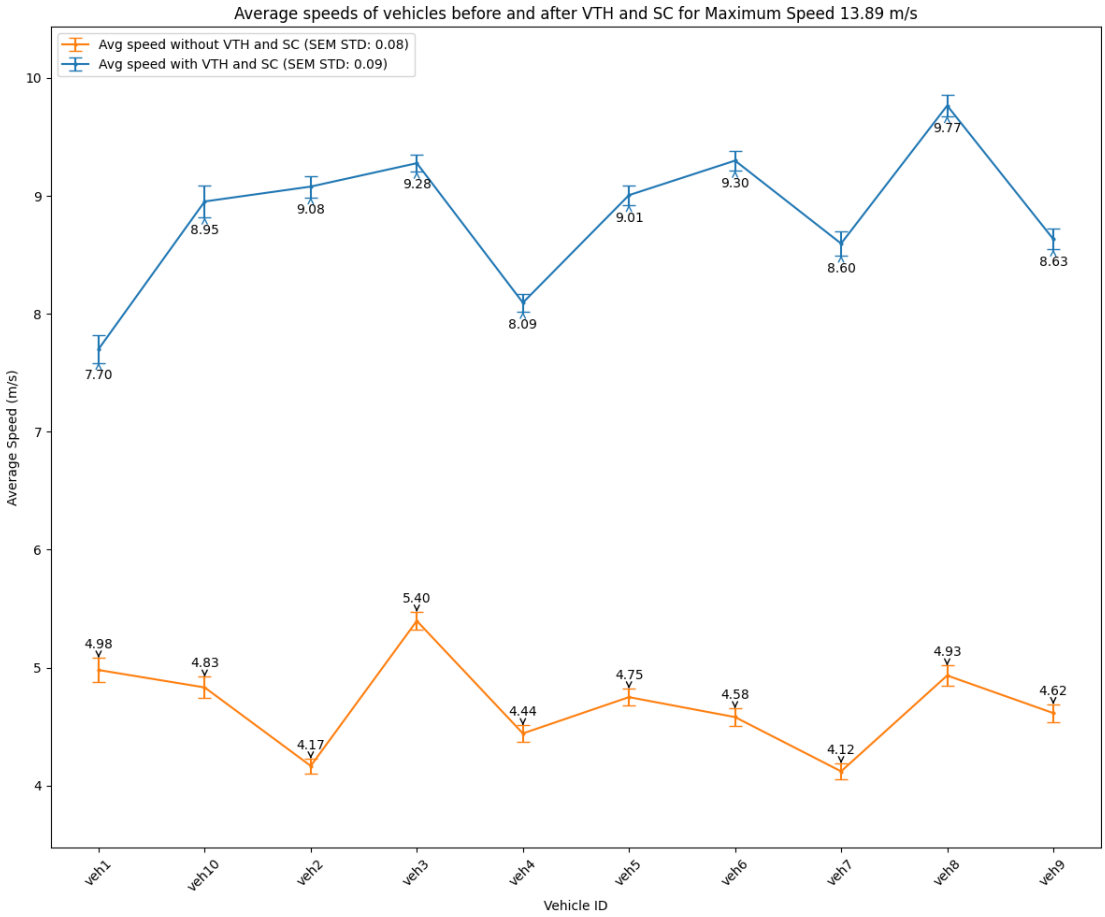

Figure 40 Average speed before variable time headway and average speed after variable time headway and spacing control strategy at the maximum drivable speed of 13.89 m/s for 10 vehicles in the simulation with spatial density 4 vehicles/km.

At a maximum drivable speed of 13.89 m/s, as shown in Figure 40, the microscopic analysis reveals that most vehicles experience an increase in average speed after implementing the VTH and SC strategies. The improvement in individual vehicle speeds is more pronounced compared to the 8.33 m/s scenario. This observation is consistent with the macroscopic trend for the 4 vehicles/km spatial density, where the average speed of the fleet shows a more significant increase after the implementation of the VTH and SC strategies at this higher maximum drivable speed.

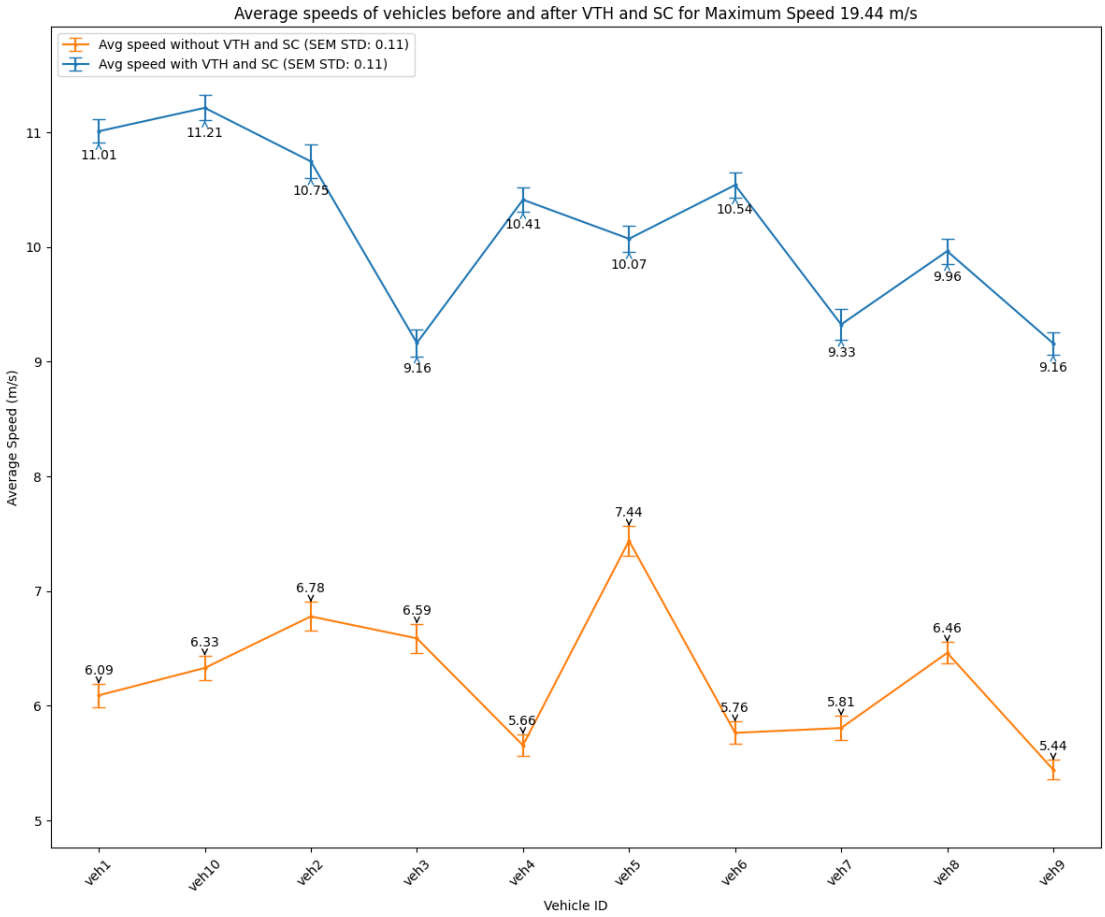

Figure 41 Average speed before variable time headway and average speed after variable time headway and spacing control strategy at the maximum drivable speed of 19.44 m/s for 10 vehicles in the simulation with spatial density 4 vehicles/km

In Figure 41, the microscopic analysis at a maximum drivable speed of 19.44 m/s demonstrates that the majority of vehicles maintain higher average speeds after the VTH and SC implementation compared to their speeds before. The improvement in individual vehicle speeds is even more evident than in the previous scenarios. This finding supports the macroscopic trend observed for the 4 vehicles/km spatial density, where the average speed of the entire fleet continues to increase after implementing the VTH and SC strategies at this maximum drivable speed. The large error difference between lower and upper indicates that each vehicle at certain times during the simulation tried to achieve a higher speed, consistently through the simulations.

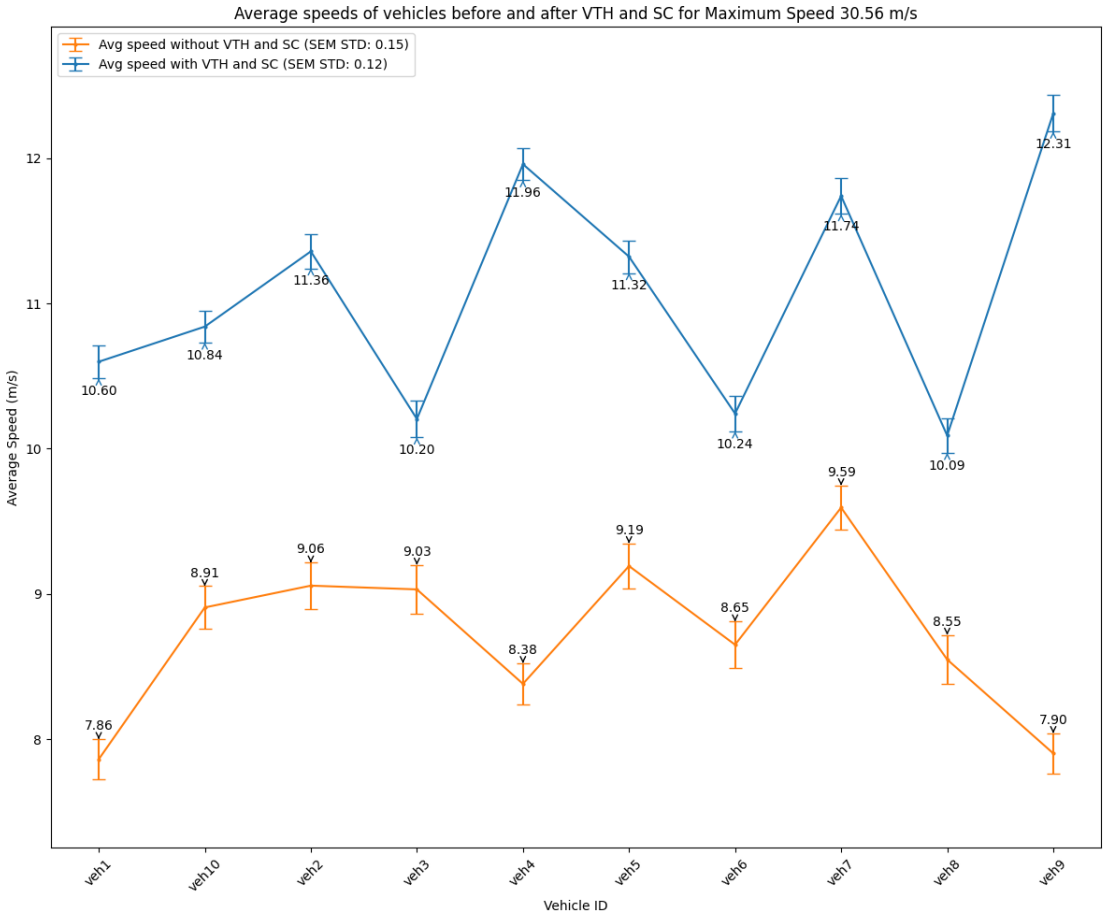

Figure 42 Average speed before variable time headway and average speed after variable time headway and spacing control strategy at the maximum drivable speed of 30.56 m/s for 10 vehicles in the simulation with spatial density 4 vehicles/km

Figure 42 illustrates the comparison between the average speeds of individual vehicles without the VTH and SC and after its implementation. The results demonstrates that the over all performance of all of vehicles is consistently high, which shows that the trend of the vehicle's speed through the simulations is consistent with the macroscopic analysis at higher speeds.

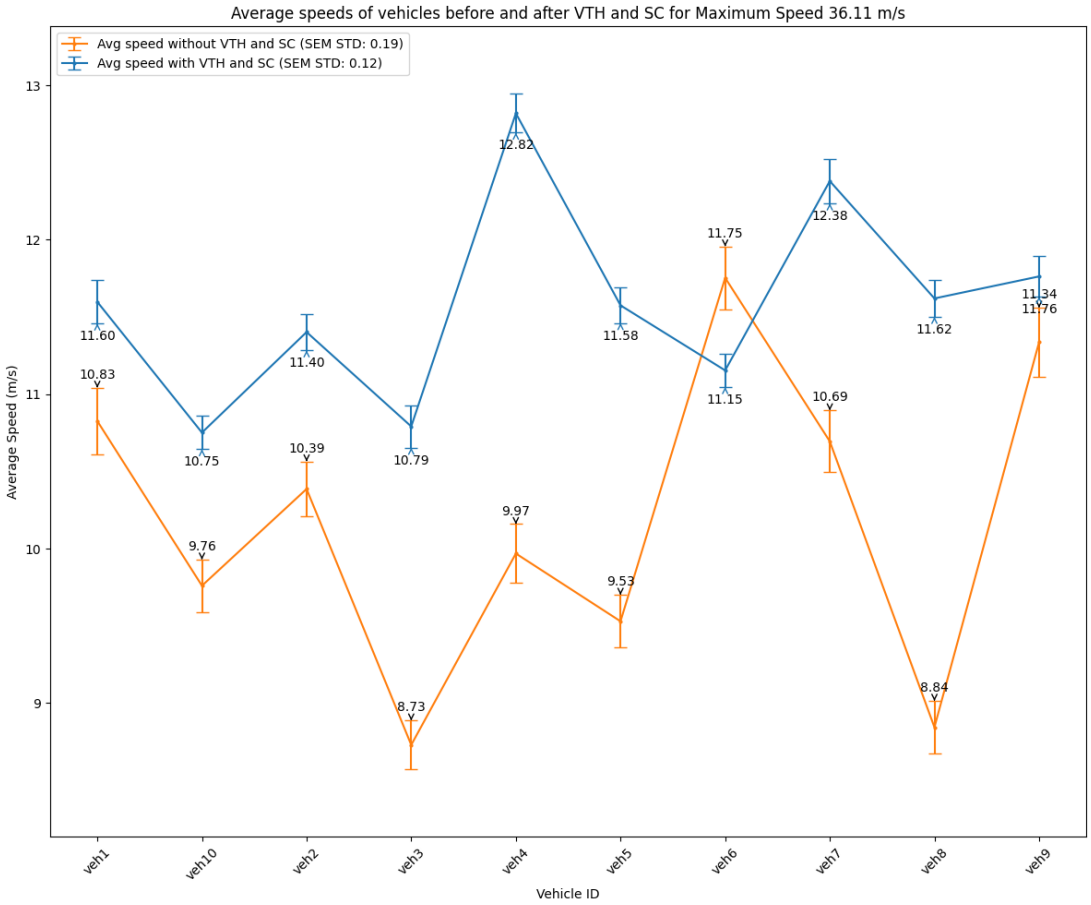

Figure 43 Average speed before variable time headway and average speed after variable time headway and spacing control strategy at the maximum drivable speed of 36.11 m/s for 10 vehicles in the simulation with spatial density 4 vehicles/km

Figure 43 illustrates the comparison of individual vehicles' speeds before the VTH and SC implementation and after it is implemented within the collision avoidance system. Even though the majority of the vehicle's results show a positive trend of increasing their average speeds post the VTH and SC implementation, one of the vehicles shows a slight decline in its speed. Vehicle with ID veh6 shows a decrease in the speed post the VTH and SC implementation. This allows us to further investigate and improve the collision avoidance's client application to regularly monitor the speeds of individual vehicles and to keep them close to optimal so as to get a resultant desirable optimal traffic flow.

As in case of vehicle with ID veh8 and with maximum allowable speed of 8.33m/s, the veh6 in this case shows the lowest time loss, due to slow downs at intersections and higher waiting times at intersections. Due to frequent stoppages, there is a loss in the average speed with which the vehicles traverse with. This is contributed by the vehicle stoppages due to warnings issued by the infrastructure's server application that houses the collision avoidance algorithm which detects potential collisions between the veh8 and other vehicles at intersections by scheduling appropriate traffic flow amongst vehicles.

This also is an example of how the present algorithm prioritizes the safety of the vehicle over enhancing its speed in case of congestions or collisions. To improve this, multiple event strategies can be incorporated which can allow algorithms such as bottleneck resolvers or schedulers to prioritize and sort the vehicle maneuvering including lane changes, steering wheel control, etc. in case of a stoppage that is required.

The consistent and substantial improvement in individual vehicle speeds at this maximum drivable speed aligns with the macroscopic trend observed for the 4 vehicles/km spatial density scenario. We have carried out this only for lower range of maximum attainable speed limits since the Collision avoidance system works closest to its ideal expected behavior in these conditions when the spatial density is also lower.
With the tuning and improvements brought into the application to handle higher spatial densities and at higher speeds, more detailed and extensive microscopic analysis would be beneficial to understand and enhance individual vehicle behavior.

5.4.2 Traffic Management

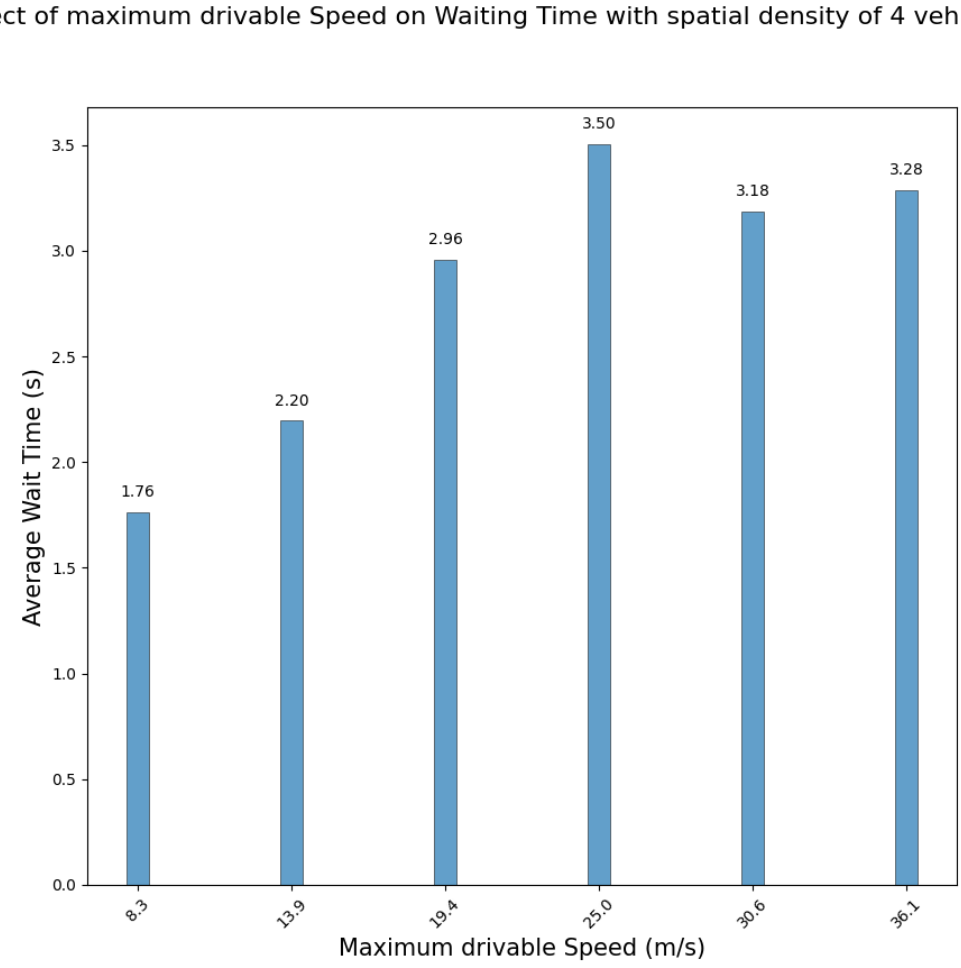

Figure 44 Maximum drivable speed and vehicle’s wait times for spatial density of 4 vehicles/km

This plot in Figure 44 demonstrates the relationship between the maximum driving speed allowed for the lane and the average wait time of each vehicle on the individual lanes. For this data evaluation, we have the waiting times on a 700m horizontal lane considered for a spatial density of 4 vehicles per kilometer. As the maximum drivable speed increases from 8.3 m/s to 27.8 m/s, the wait time also increases from 1.40 s to 2.48 s. This trend indicates that at higher maximum set speeds on the lane, the vehicles try to attain a higher speed, which is leading to bottle neck due to the intersections that are present along this lane. There are two intersections on this lane over which the vehicles experience longer wait times due to more frequent collision chances recorded by the server application that sends DENMs to the vehicles to either reduce their speeds or to stop the vehicles, and hence the maneuvers undertaken at intersections are high when the spatial density is high. The increased wait time is a result of vehicles decelerating and stopping to avoid collisions before regaining speed.

The longer wait times at higher average fleet speeds and high spatial density can contribute to increased congestion at intersections. However, this is due to the Collision Avoidance service's ability to detect and prevent collisions ensuring safety ,which is a priority in traffic management. The trade-off between safety and traffic flow efficiency is evident in this scenario, and the Collision Avoidance system adapts its time headway and spacing control strategy to strike a balance between the two factors.

Figure 45 Maximum drivable speed and vehicle wait times for spatial density of 2 vehicles/km

This plot in figure 45, shows the effect of maximum drivable speeds on wait times for a lower spatial density of 2 vehicles per kilometer. Like the Figure 43, the wait time increases from 1.14 s to 2.44s as the maximum drivable speed increases from 8.3 m/s to 25.0 m/s. This trend suggests that with even at lower spatial density, the wait time increases initially with increasing maximum drivable speeds set for the lane and average speeds of the vehicles (average speed of vehicles increases with increasing maximum drivable speeds). At higher maximum drivable speeds though, the waiting time slightly decreases owing to a lower probability of congestion due to fewer detected collisions. After deceleration, vehicles can more quickly reach their maximum speed due to fewer obstacles.

The decreasing wait times with increasing average fleet speed at low spatial density indicate that the Collision Avoidance system can maintain a more efficient traffic flow in

these conditions. With fewer vehicles on the road, the Collision Avoidance service can adapt its time headway and spacing control strategy to allow for shorter gaps between vehicles, reducing wait times at intersections and improving overall traffic management.

On comparison between the wait times for two different spatial densities translating to 4 vehicles and 2 vehicles/km respectively across various average fleet speeds. For both spatial densities, the wait time increases with increasing maximum speeds initially, however in case of lower spatial density the wait times drop at higher maximum drivable speeds. This comparison highlights the impact of spatial density on wait times, as higher spatial density leads to more frequent collision avoidance maneuvers and, consequently, longer wait times. The comparison between the two spatial densities also emphasizes the importance of considering spatial density in traffic management. The Collision Avoidance service must adapt its time headway and spacing control strategy based on the spatial density to ensure safety and optimize traffic flow. In higher spatial density scenarios, the Collision Avoidance service prioritizes safety by allowing for longer wait times, potentially leading to increased congestion. In lower spatial density scenarios, the Collision Avoidance service can prioritize traffic flow efficiency by allowing for shorter wait times without compromising safety.

By examining the relationships between average fleet speed, spatial density, and wait times, we can understand how the Collision Avoidance system manages collisions and congestion at intersections, ultimately contributing to improved overall traffic management. The Collision Avoidance system’s ability to balance safety and traffic flow efficiency based on the prevailing traffic conditions highlights its potential to revolutionize traffic management in the era of connected and autonomous vehicles.

The waiting times can be reduced by using multiple events being reported and addressed at the same time. In single event strategy, the vehicles that are prioritized to be warned by the infrastructure to stop or change their course of travel including their speeds, acceleration are the ones which are close by distance to a possibility of collisions. This leads to more waiting time for the other vehicles in the traffic fleet. If multiple event strategy is applied at infrastructure’s server application and simultaneously multiple events are reported, there can be multiple actions can be incorporated within the client application such as lane changing, steering wheel controls and other maneuvering actions other than the present evasive maneuvering.

In addition to analyzing the wait times, it is equally important to examine the travel times to gain a comprehensive understanding of the Collision Avoidance system’s impact on traffic management. The following results presented in this analysis illustrate the relationship between maximum drivable speed, spatial density, and travel times. By studying these results, we can assess how the enhanced Collision Avoidance service's variable time headway and spacing control strategies affect the overall travel times recorded from individual vehicles in different traffic conditions. The insights gained from this analysis will help evaluate the effectiveness of the enhanced Collision Avoidance service in optimizing traffic flow while ensuring safety.

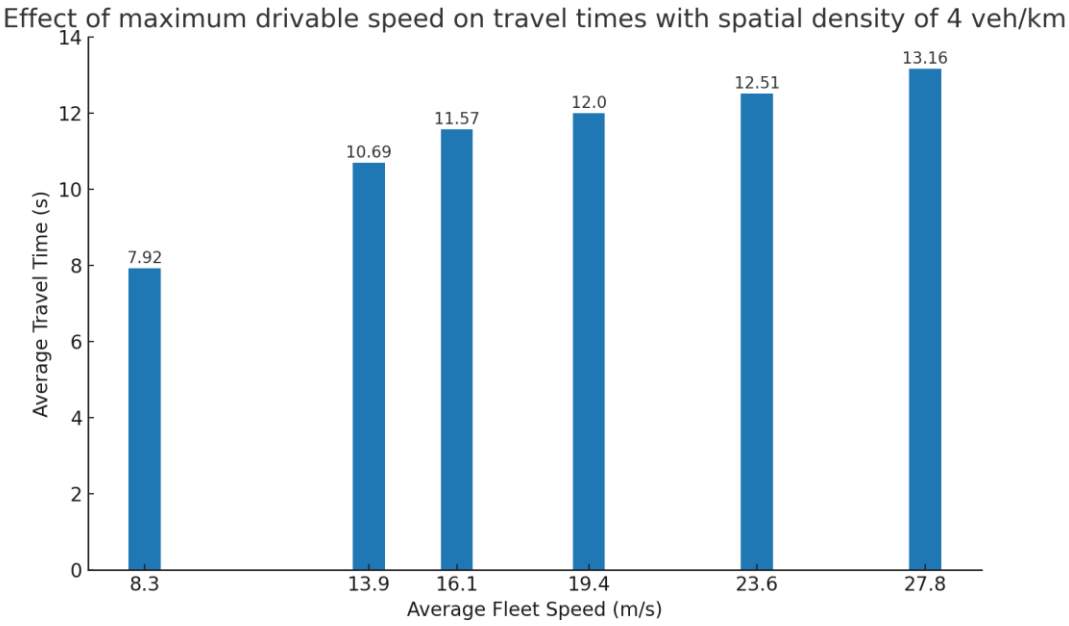

Figure 46 Effect of maximum drivable speed on the travel times with spatial density of 4 vehicles/km

This Figure 46 shows the relationship between the average fleet speed and the average travel time for a spatial density of 4 vehicles per kilometer. As the average fleet speed increases from 8.3 m/s to 27.8 m/s, the travel time also increases from 7.92 s to 13.16 s. This trend indicates that at higher average speeds, vehicles experience longer travel times, despite the expectation that higher speeds would lead to shorter travel times. The increase in travel time can be attributed to the more frequent collision avoidance maneuvers at intersections when the spatial density is high, resulting in longer wait times that contribute to the overall travel time.

The longer travel times at higher average fleet speeds and high spatial density suggest that the enhanced Collision Avoidance service prioritizes safety over traffic flow efficiency in these conditions. The increase in travel time is a consequence of the enhanced Collision Avoidance service adapting its time headway and spacing control strategy to allow for longer gaps between vehicles, reducing the risk of collisions but potentially leading to increased congestion and slower overall traffic flow.

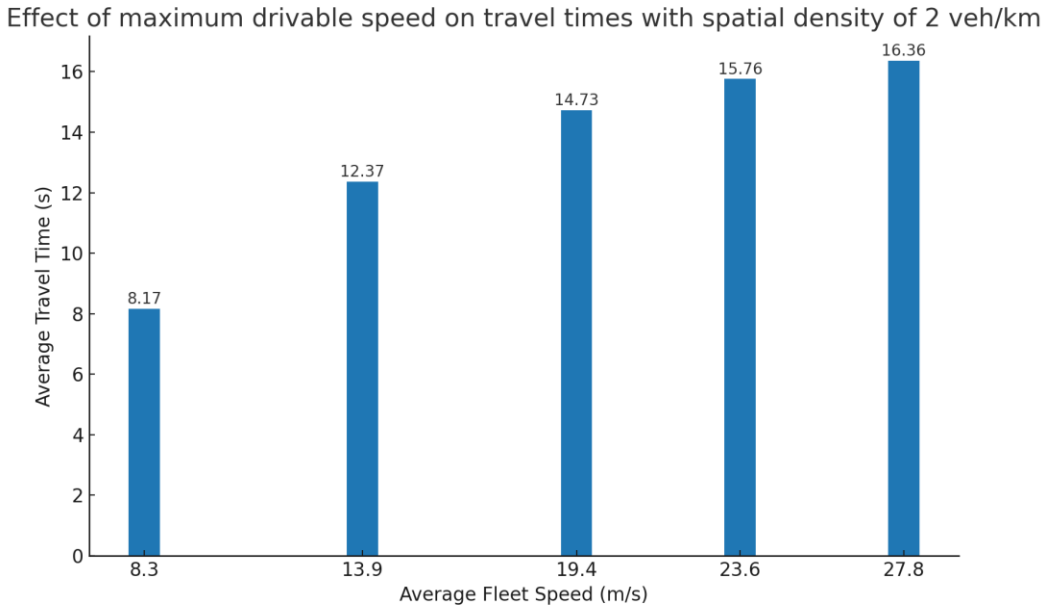

Figure 47 Effect of average fleet speed on the travel times with spatial density of 2 vehicles/km

The increase in overall travel time is because at lower spatial densities, the time duration for which the simulation was run was the same – 300s for a variety of spatial densities simulated. Therefore a lesser number of vehicles have to travel a longer time in the data simulated with 2 vehicles/km as compared to that with 4 vehicle/km.

The figure 47 illustrates the effect of average fleet speed on travel times for a lower spatial density of 2 vehicles per kilometer. In contrast to figure 46, the travel time increases more gradually from 8.17 s to 16.36 s as the average fleet speed increases from 8.3 m/s to 27.8 m/s. The increase in travel time is less pronounced compared to the higher spatial density scenario in the Figure 47. This trend suggests that with lower spatial density, the Collision Avoidance service can maintain a more efficient traffic flow, as there are fewer vehicles and, consequently, a lower probability of congestion due to lower detected collisions at this spatial density as compared to the higher spatial density.

The more gradual increase in travel times with increasing average fleet speed at low spatial density indicates that the Collision Avoidance service can better optimize traffic flow in these conditions. With fewer vehicles on the road, the Collision Avoidance service can adapt its time headway and spacing control strategy to allow for shorter gaps between vehicles while still maintaining safety, resulting in a more efficient traffic flow and relatively shorter travel times compared to higher spatial density scenarios.

On comparing the travel times for two different traffic fleet capacity (10 vehicles and 5 vehicles in simulation) translating to spatial density of 4 vehicles/km and 2 vehicles/km across various average fleet speeds, we can summarize that the travel time generally

increases with increasing average fleet speed. However, the travel times for the higher spatial density (10 vehicles) are consistently longer than those for the lower spatial density (5 vehicles) at each given average fleet speed. This comparison highlights the impact of spatial density on travel times, as higher spatial density leads to more frequent collision avoidance maneuvers and, consequently, longer travel times.

The comparison between the two spatial densities emphasizes the importance of considering spatial density in traffic management. In higher spatial density scenarios, the Collision Avoidance service prioritizes safety, resulting in longer travel times and potentially slower overall traffic flow. In lower spatial density scenarios, the Collision Avoidance service can better optimize traffic flow, leading to relatively shorter travel times while still maintaining safety.

The figures 46 and 47, which are analyzed in this study demonstrate the impact of the Collision Avoidance service's variable time headway and spacing control strategies on travel times under different traffic conditions. By examining the relationships between average fleet speed, spatial density, and travel times, we can understand how the enhanced Collision Avoidance service manages collisions and congestion at intersections, and its effect on overall traffic flow.

5.4.3 Network Parameters

The network traffic generated by the Collision Avoidance application is independent of the vehicle's individual speeds being assessed and monitored the Collision Avoidance Strategy (CAS). The uplink traffic, which consists of CAM messages, is determined by the fixed CAM dissemination rate of 10 Hz and depends on the spatial density of the vehicles in the simulated environment.

The measurements of the traffic load are performed at the application layer, allowing for the analysis of the system's latency in receiving the CAM packets. At lower vehicle speeds and higher spatial densities, the downlink traffic load, which comprises DENM messages being sent by the server application at RSUs to client application on Vehicle's OBUs, increases. However, at higher vehicle speeds, regardless of the spatial density, the downlink traffic load decreases. This phenomenon is attributed to a limitation within the Collision avoidance application which detects and avoids fewer collisions at higher speeds. Consequently, this directly affects the downlink traffic load, resulting in a lower number of DENMs being sent, even in potentially dangerous situations.

Table 6 Traffic load generated in downlink.

Maximum Speed(m/s)	**Number of DENMs sent**	**Number of DENMs received**
18.33	9041	3937
19.44	11284	4458
25.00	12107	5001
27.78	12421	6242
30.56	14028	6436
36.11	13272	6201

Analyzing the table, we observe that as the maximum speed increases from 18.33 m/s (30 km/h) to 36.11 m/s (130 km/h), the number of DENMs sent initially increases but then decrease at higher speeds. This trend supports the discussion pertaining to the limitation of the collision application in detecting collisions at higher vehicle speeds.

The number of DENMs received also exhibits a similar trend, with an initial increase at lower speeds followed by a decrease at higher speeds. However, the ratio of DENMs received to DENMs sent decreases as the speed increases, indicating that a smaller proportion of the sent DENMs are successfully received at higher vehicle speeds. This observation suggests that the collision avoidance application's effectiveness in preventing collisions may be reduced at higher speeds due to the lower number of DENMs being sent and received.

In conclusion, the analysis of the downlink traffic load and the number of DENMs sent and received at different vehicle speeds reveals a limitation in the collision avoidance application's ability to detect and prevent collisions at higher speeds. This limitation results in a decrease in the downlink traffic load and a lower number of DENMs being sent and received, potentially compromising the effectiveness of the collision avoidance system at higher vehicle speeds. Further investigation and optimization of the eCA application may be necessary to address this issue and ensure reliable collision detection and prevention across a wide range of vehicle speeds.

The average latency performance is evaluated for both uplink and downlink data communication of CAMs and DENMs, respectively. The timestamp that is included within the DENM that is sent from the server to vehicles is recorded at the Client application. This helps us to calculate the Downlink latency that is the time gap between the time at which DENM was sent from the server to the time at which it is received from the Client. The data below is evaluated for multiple simulation campaigns of 50 simulations, for 3600 seconds with varying spatial density of 4 vehicles and 2 vehicles per Kilometer. These 50 simulations are repeated over a varying maximum speed range from 8.33 m/s to 36.11 m/s.

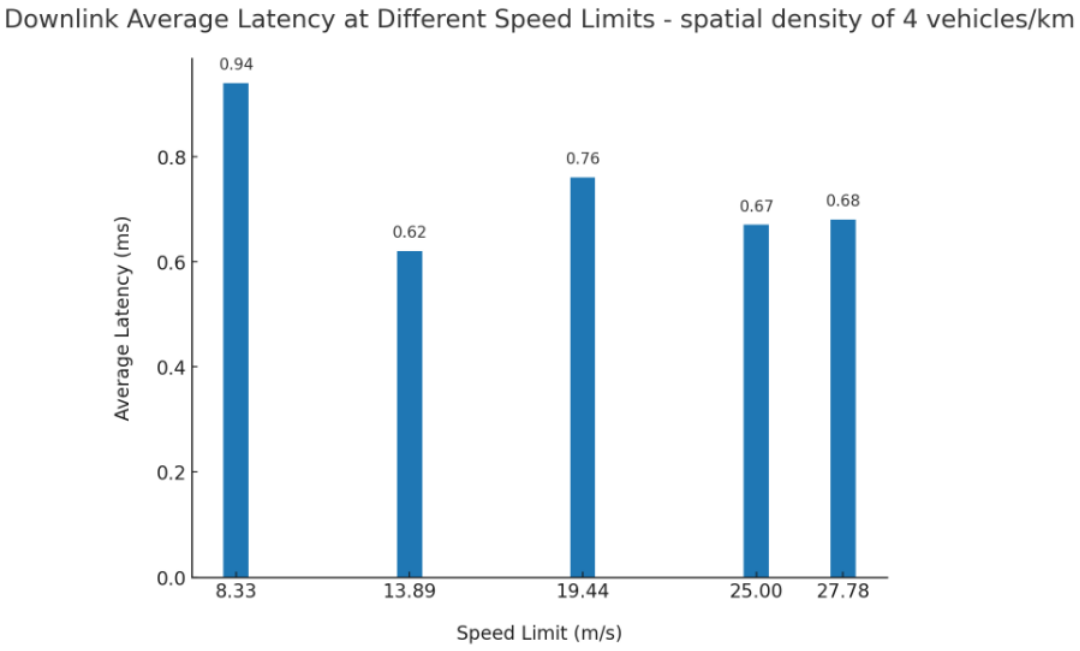

Figure 48 Downlink Average Latency- with spatial density of 4 vehicles/km.

In Figure 48 we can observe that the average latency decreases as the vehicle speed limit increases. The highest latency of 0.94ms is observed at the lowest speed of 8.33 m/s. As the speed limit increases to 27.78 m/s, the latency decreases. This observed trend is due to the tendency of the collision avoidance system to prioritize the processing from the vehicles at higher speeds to avoid the risk of collision and thus, they require a faster

response time. This also proves that the system is optimized to handle the increased message frequency and reduced response time requirements at higher vehicle speeds.

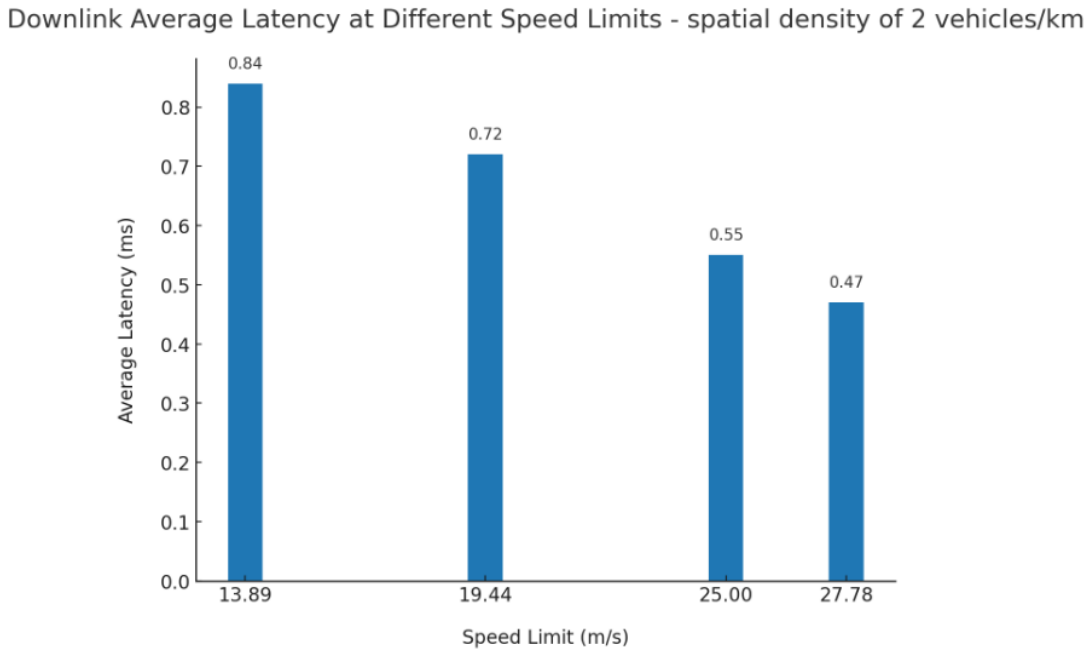

Figure 49 Average Downlink Latency Spatial density 2 vehicles/km.

The Figure 49 shows the Average downlink latency data plot recorded for the spatial density of 2 vehicles/km. We can observe by comparing Figure 48 and Figure 49 the overall latency at each speed is lower with a decreased spatial density than the former (4 vehicles/km). This is due to the faster DENM response by the server toward the vehicles at lower spatial density as the number of nodes that receive the DENMs are also lower. Thus, the resultant data follows the trend of the decreased latency over increasing maximum speed, which is due to the optimization of the collision avoidance system to efficiently and timely avoid collisions at higher speeds.

The uplink latency is evaluated by server application for the CAM messages received at the server application. The CAMs sent by the client application has the timestamp recorded of its generation. This is compared with the timestamp at which it is received at the server application. The uplink latency is independent of the spatial density since it is only dependent of the OBUs CAM encoding and generation process latency and when it is received at server and followed by its decoding. Table 7 illustrates the uplink latency which is evaluated at the server application utilizing the data collected with spatial density of 4vehicles/km in the simulation. There is no significant change in the latency with varying maximum speeds set and for varying spatial density for the simulation.

Table 7 Uplink Latency for spatial density of 4 vehicles/km.

Maximum Speed	**Uplink Latency(ms)**
8.33 m/s	4.45
13.89 m/s	4.43
25.00 m/s	4.44
27.78 m/s	4.50

To analyze the data for Packet Delivery ratio, we again consider two sets of data with varying spatial densities of 4vehicle/km and 2 vehicles/km. Both of sets of data have been measured for increasing order of the maximum speed to be attained by the traffic fleet.

Table 8 Downlink Packet Delivery Ratio.

Maximum Speed	**Spatial Density 2 Veh/km**	**Spatial Density 4Veh/km**
8.33 m/s	0.929	0.992
13.89 m/s	0.983	0.989
19.44 m/s	0.956	0.989
25.00 m/s	0.985	0.988
27.78 m/s	0.980	0.988

The data in Table 8 summarizes the downlink PDR evaluated in a traffic spatial density of 2 vehicle/km. The maximum speed was increased from 8.33 m/s to 27.78 m/s. The results show a slightly lower value at higher speeds than expected which is because of the increasing number of collisions detected at higher speeds that contribute towards the net DENMs being sent from the server to the clients. However, since the spatial density is less, the possibility that the vehicle for which the DENM was sent by the server would have left the safety radius or region of interest that is defined within the server's operability.

Table 8 also details the downlink PDR evaluated for data collected with a spatial density of 4 vehicles/km. The results follow the similar trend however, the values are always above 0.98. This shows that the network's performance is superior at a higher spatial density due to the ability of the traffic to remain within the operation radius of the server for a higher duration due to congestion. The performance can be improved for lower spatial density by increasing the operation radius of servers or setting up more RSUs with server applications. In conclusion of this discussion, both the results demonstrate a considerably high Downlink PDR resultant data which is always above 0.92 indicating a 92% reception rate of DENMs by the traffic participants(vehicles) using collision avoidance system.

Table 9 Uplink Packet Delivery Ratio.

Maximum Speed	Spatial Density 2 Veh/km	Spatial Density 4 Veh/km
8.33 m/s	0.99960	0.99996
13.89 m/s	0.99970	0.99997
19.44 m/s	0.99970	0.99997
25.00 m/s	0.99970	0.99998
27.78 m/s	0.99970	0.99998

The data for Uplink traffic was collected to evaluate uplink PDR for varying spatial densities as before with the maximum speed ranging from 8.33 m/s to 27.78m/s. Uplink PDR is evaluated both at server and client application which record the CAMs received and CAMs sent, respectively. Table 9 shows that the Uplink PDR remains consistently high, around 0.99970 for PDR evaluated when spatial density if 2 vehicles/km, across all vehicle speeds tested. This indicates that a vast majority of CAM sent by the vehicles are successfully received by the server, regardless of the vehicle speed.

The Uplink PDR is evaluated for data collected with spatial density of 4 Vehicles/km with an increasing maximum speed from 8.33 m/s to 25.00 m/s which is also contained within Table 9. These uplink PDR values are consistent with that discussed earlier for lower spatial densities. Thus, we can conclude that spatial density is not a factor that affects the uplink PDR. The high uplink PDR in both the cases of spatial densities suggests that the network is capable of handling the traffic load generated by the vehicle's periodic CAM transmissions, even at higher speeds.

The consistently high PDR values in both uplink and downlink directions demonstrate the robustness of the communication network in supporting the collision avoidance system. The reliable delivery of CAM and DENM messages is crucial for the timely detection of potential collisions and the dissemination of warning information to the vehicles.

However, it is important to note that the PDR results presented here are based on the specific simulation setup and parameters used in the study which are detailed in the earlier chapters under the context of simulation parameters. In real-world scenarios, factors such as network congestion, interference, and obstacles may impact the PDR performance.

5.4.4 Result Discussion

The results presented in this study provide valuable insights into the performance and effectiveness of the enhanced Collision Avoidance (enhanced Collision Avoidance) system, which employs variable time headway and spacing control strategies to improve road safety and traffic efficiency. The analysis covers various aspects, including collision

avoidance, traffic management, and network parameters, offering a holistic view of the system's performance under different traffic conditions and spatial densities.

Collision Avoidance: The collision avoidance analysis demonstrates the Collision Avoidance system's ability to reduce the number of collisions compared to scenarios without the system in place. The results show a clear relationship between the maximum drivable speed and the number of collisions, with higher speeds leading to more collisions. This trend is more pronounced in higher spatial density scenarios (4 vehicles/km) compared to lower spatial densities (2 vehicles/km), indicating that the enhanced Collision Avoidance system's performance is influenced by traffic density.

The comparative analysis of average speed and collisions at different spatial densities highlights the need for further optimization of the collision avoidance algorithm, particularly in high-density, high-speed scenarios. Tuning the algorithm to account for spatial density could help reduce collisions in the 4 vehicles/km scenario to levels similar to those observed in the 2 vehicles/km scenario.

Microscopic analysis of individual vehicle speeds before and after the implementation of the Variable Time Headway (VTH) and Spacing Control (SC) strategies reveals the effectiveness of these strategies in improving traffic flow and safety. The results show that most vehicles maintain higher average speeds after the implementation of the VTH and SC strategies, with the improvement becoming more pronounced at higher maximum drivable speeds. This finding validates the macroscopic trends observed and demonstrates the positive impact of the enhanced Collision Avoidance system on individual vehicle performance.

Traffic Management: The analysis of wait times and travel times provides valuable insights into the enhanced Collision Avoidance system's impact on traffic management. The results show that higher average fleet speeds and spatial densities lead to increased wait times, as the enhanced Collision Avoidance system prioritizes safety over traffic flow efficiency in these conditions. However, in lower spatial density scenarios, the enhanced Collision Avoidance system can maintain more efficient traffic flow, resulting in decreased wait times with increasing average fleet speeds.

The comparative analysis of wait times and travel times at different spatial densities emphasizes the importance of considering spatial density in traffic management. The enhanced Collision Avoidance system must adapt its time headway and spacing control strategies based on the prevailing traffic conditions to balance safety and efficiency. In high-density scenarios, the system prioritizes safety, potentially leading to increased travel times and slower overall traffic flow. In low-density scenarios, the system can optimize traffic flow, resulting in relatively shorter travel times while maintaining safety.

Network Parameters: The analysis of network parameters, specifically the traffic load generated by the enhanced Collision Avoidance application, reveals some limitations and areas for improvement. The uplink traffic, consisting of CAM messages, is determined by

the fixed dissemination rate and spatial density, while the downlink traffic, comprising DENM messages, is influenced by vehicle speeds and collision detection.

The results show that at higher vehicle speeds, the downlink traffic load decreases, indicating a limitation in the enhanced Collision Avoidance application's ability to detect collisions at high speeds. This limitation results in a lower number of DENMs being sent and received, potentially compromising the system's effectiveness in preventing collisions at higher speeds.

To address this issue, further investigation and optimization of the enhanced Collision Avoidance application are necessary. Improving the collision detection algorithm to accurately identify potential collisions at high speeds could help maintain the system's effectiveness across a wider range of vehicle speeds. Additionally, optimizing the DENM dissemination strategy to ensure reliable transmission and reception of critical safety messages could enhance the overall performance of the enhanced Collision Avoidance system.

Thus, the comprehensive analysis of the enhanced Collision Avoidance system's performance presented in this study highlights the system's potential to improve road safety and traffic efficiency. The results demonstrate the effectiveness of the variable time headway and spacing control strategies in reducing collisions and optimizing traffic flow under various conditions. However, the study also identifies areas for improvement, particularly in high-density, high-speed scenarios and in the detection and dissemination of safety messages at higher vehicle speeds.

In Chapter 6 we will summarize the key findings of this study and discuss their implications for the development and deployment of enhanced collision avoidance systems. It will also outline potential future research directions, including the potential impact of the enhanced Collision Avoidance system on the broader context of intelligent transportation systems and its role in enabling safer, more efficient, and sustainable mobility in the era of connected and autonomous vehicles.

6 Conclusion

In this thesis we presented a comprehensive study on the development and evaluation of an Collision Avoidance system that integrates Variable Time Headway and Spacing Control strategies to improve road safety and traffic efficiency. The research builds upon the foundational work presented in "An Edge-Based Framework for Enhanced Road Safety of Connected Cars" [9] and "Edge-based Collision Avoidance for Vehicles and Vulnerable Users" [10], extending their capabilities by incorporating advanced spacing strategies inspired by the work of Chen et al. [11].

The primary objective of this thesis research was to address the limitations of the existing Collision Avoidance system in optimizing and monitoring traffic flow, particularly in adapting to varying traffic conditions and maintaining performance with changing traffic density. By integrating variable time headway (VTH) and spacing control strategies within the Collision Avoidance system and leveraging the multi-stack simulation framework for data collection, this research aimed to enhance the system's performance and adaptability in both collision avoidance and traffic flow optimization.

The methodology employed in this study involved the implementation of the enhanced Collision Avoidance system with VTH and spacing control strategies within the NS-3 and SUMO simulation environments. The system's architecture, including the server and client applications, as well as the collision avoidance algorithm, were designed and implemented to facilitate the integration of these advanced strategies. The simulation setup was carefully designed to model realistic traffic scenarios, with varying speed limits and traffic densities, to evaluate the system's performance under diverse conditions.

In the development of the collision avoidance system, a rules-based logic approach was chosen over other decision-making techniques for several reasons. Firstly, rules-based logic approaches are straightforward and easy to understand. The decision-making process is based on a set of predefined rules and conditions, which makes the system's behavior more transparent and interpretable. This transparency is crucial in safety-critical applications like collision avoidance, where the decision-making process should be easily comprehensible to developers, stakeholders, and end-users.

Secondly, the rules-based approach allows for the definition of specific rules and thresholds to handle different traffic scenarios. By carefully designing the rules based on domain knowledge and expert insights, the collision avoidance system can effectively adapt its behavior to varying traffic conditions, such as congestion levels and vehicle speeds. This adaptability enables the system to optimize safety and traffic flow efficiency in diverse situations.

Thirdly, compared to more complex decision-making techniques, such as machine learning algorithms or optimization methods, rules-based logic approaches have lower computational overhead. The decision-making process involves evaluating a set of predefined rules against the input data, which can be done efficiently in real-time. This

computational efficiency is crucial for collision avoidance systems, where quick and timely decisions are essential to prevent accidents and maintain smooth traffic flow.

Lastly, implementing a rules-based logic approach is relatively straightforward, as it involves translating the domain knowledge and expert insights into a set of rules and conditions. The modular nature of rules-based systems also makes them easier to maintain and update. As new insights or requirements emerge, the rules can be modified or extended without requiring significant changes to the overall system architecture.

While fuzzy logic and other decision-making techniques have their merits, the rules-based logic approach was deemed the most suitable for the collision avoidance system in this thesis due to its simplicity, interpretability, adaptability, computational efficiency, and ease of implementation and maintenance. The rules-based approach strikes a balance between effectively handling different traffic scenarios and maintaining a transparent and efficient decision-making process.

The results obtained from the simulations demonstrate the effectiveness of the Collision Avoidance system in improving road safety and traffic efficiency. The collision avoidance analysis revealed a significant reduction in the number of collisions compared to scenarios without the system, with the performance being influenced by factors such as maximum drivable speed and traffic density. The microscopic analysis of individual vehicle speeds before and after the implementation of the VTH and spacing control strategies validated the macroscopic trends, confirming the positive impact of the Collision Avoidance system on individual vehicle performance.

The traffic management analysis provided valuable insights into the system's ability to adapt to varying traffic conditions. The results showed that the Collision Avoidance system could maintain efficient traffic flow in low-density scenarios while prioritizing safety in high-density situations. The comparative analysis of wait times and travel times at different spatial densities emphasized the importance of considering traffic density in the system's decision-making process.

The network parameter analysis revealed some limitations in the Collision Avoidance system's ability to detect collisions at high vehicle speeds, resulting in a decreased number of DENMs being sent and received. This finding highlights the need for further optimization of the collision detection algorithm and the DENM dissemination strategy to ensure reliable performance across a wide range of vehicle speeds.

To further these efforts and work, the project is currently being investigated to be extended into the reinforcement learning (RL) domain. The exploration of RL techniques in with the research demonstrated in this thesis opens up new avenues for developing more adaptive and robust collision avoidance systems. By incorporating RL and deep reinforcement learning (DRL) algorithms, the enhanced Collision Avoidance system can potentially learn and adapt its strategies in real-time, enhancing its performance in complex and dynamic traffic environments. This approach can lead to the development of

a more intelligent and autonomous collision avoidance system that can effectively handle the challenges posed by real-world traffic scenarios.

This work contributed to the advancement of intelligent transportation systems by presenting an Collision Avoidance system that integrates variable time headway and spacing control strategies. The novelty of the proposed algorithm and the promising results obtained from the simulations and analyses demonstrate the potential of the Collision Avoidance system in improving road safety and traffic efficiency.

6.1 Future work

The findings and limitations identified in this thesis provide several opportunities for future research and development in the field of intelligent transportation systems and V2I based collision avoidance algorithms. Some potential areas for future work include:

1. Optimization of the collision detection algorithm: Further research could focus on improving the accuracy and reliability of the collision detection algorithm, particularly at high vehicle speeds. This could involve the incorporation of more advanced sensor fusion techniques, machine learning algorithms, or the integration of additional data sources to enhance the system's situational awareness.

2. Enhancement of the DENM dissemination strategy: Future work could investigate more efficient and reliable methods for disseminating DENMs to ensure timely delivery of critical safety information, even in high-density traffic scenarios or at high vehicle speeds. This may involve the exploration of advanced networking protocols, multi-hop communication strategies, or the integration of 5G and beyond wireless technologies.

3. Integration of reinforcement learning techniques: Building upon the preliminary discussions in this thesis, future research could focus on the practical implementation and evaluation of reinforcement learning and deep reinforcement learning algorithms within the Collision Avoidance system. This would involve the development of suitable state representations, reward functions, and learning architectures that can effectively capture the complexity of real-world traffic scenarios and enable the system to learn and adapt its strategies in real-time.

4. Expansion to multi-agent systems: Future work could explore the extension of the enhanced Collision Avoidance system to multi-agent scenarios, where multiple vehicles equipped with the system collaborate and coordinate their actions to optimize traffic flow and ensure safety at a network level. This would require the development of advanced coordination mechanisms, communication protocols, and distributed learning algorithms that can handle the challenges of scalability, heterogeneity, and dynamic nature of vehicular networks.

5. Real-world deployment and validation: To fully assess the effectiveness and robustness of the Collision Avoidance system, future research should focus on the deployment and validation of the system in real-world traffic environments. This would involve the integration of the system with actual vehicles and infrastructure, the conduction of extensive field trials, and the evaluation of the system's performance under various weather conditions, road geometries, and traffic patterns. The insights gained from such real-world deployments would be invaluable for the further refinement and optimization of the system.

6. Integration with other advanced driver assistance systems (ADAS): Future work could investigate the integration of the enhanced Collision Avoidance system with other ADAS technologies, such as lane-keeping assist, adaptive cruise control, or automated emergency braking. The synergistic integration of these systems could lead to the development of more comprehensive and effective safety solutions for intelligent vehicles.

By addressing these future research directions, the Collision Avoidance system with variable time headway and spacing control strategy presented in this thesis can be further developed and refined, contributing to the realization of safer, more efficient, and sustainable transportation systems. The integration of advanced control strategies, reinforcement learning techniques, and real-world validation will pave the way for the widespread adoption of intelligent collision avoidance technologies, ultimately improving road safety and traffic management in the era of connected and autonomous vehicles.

7 Reference List

[1] "SAE Levels of Driving Automation™ Refined for Clarity and International Audience." Accessed: Apr. 19, 2024. [Online]. Available: https://www.sae.org/site/blog/sae-j3016-update

[2] "Level-of-Automation-nhtsa.pdf."

[3] "mdtpo-connected-autonomous-vehicle-strategic-plan-final-report-2023-04.pdf." Accessed: Apr. 19, 2024. [Online]. Available: https://miamidadetpo.org/library/studies/mdtpo-connected-autonomous-vehicle-strategic-plan-final-report-2023-04.pdf

[4] "USDOT Connected Vehicle Research Program: Vehicle-to-Vehicle Safety Application Research Plan".

[5] S. Chen, J. Hu, Y. Shi, and L. Zhao, "LTE-V: A TD-LTE-Based V2X Solution for Future Vehicular Network," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 997–1005, Dec. 2016, doi: 10.1109/JIOT.2016.2611605.

[6] S. Davis, "Improving System Reliability in Automotive and Industrial Cameras With Accurate Temperature Sensing," 2022.

[7] Y. Jo, J. Jang, J. Ko, and C. Oh, "An In-Vehicle Warning Information Provision Strategy for V2V-Based Proactive Traffic Safety Management," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19387–19398, Oct. 2022, doi: 10.1109/TITS.2022.3156923.

[8] "ACEA_Road_Safety.pdf."

[9] M. Malinverno, J. Mangues-Bafalluy, C. E. Casetti, C. F. Chiasserini, M. Requena-Esteso, and J. Baranda, "An Edge-Based Framework for Enhanced Road Safety of Connected Cars," *IEEE Access*, vol. 8, pp. 58018–58031, 2020, doi: 10.1109/ACCESS.2020.2980902.

[10] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, "Edge-based Collision Avoidance for Vehicles and Vulnerable Users".

[11] J. Chen, Y. Zhou, and H. Liang, "Effects of ACC and CACC vehicles on traffic flow based on an improved variable time headway spacing strategy," *IET Intell. Transp. Syst.*, vol. 13, no. 9, pp. 1365–1373, Sep. 2019, doi: 10.1049/iet-its.2018.5296.

[12] "SUMO_Simulation_of_Urban_MObility_An_open-source_t.pdf."

[13] "Enabling LTE Emulation by Integrating CORE Emulator and LTE-EPC Network (LENA) Simulator." Accessed: Apr. 01, 2024. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8292642

[14] J. S. Weber, M. Neves, and T. Ferreto, "VANET simulators: an updated review," *J. Braz. Comput. Soc.*, vol. 27, no. 1, p. 8, Dec. 2021, doi: 10.1186/s13173-021-00113-x.

[15] J. Santa, F. Pereñíguez, A. Moragón, and A. F. Skarmeta, "Experimental evaluation of CAM and DENM messaging services in vehicular communications," *Transp. Res. Part C Emerg. Technol.*, vol. 46, pp. 98–120, Sep. 2014, doi: 10.1016/j.trc.2014.05.006.

[16] S. Kato, M. Hiltunen, K. Joshi, and R. Schlichting, "Enabling vehicular safety applications over LTE networks," in *2013 International Conference on Connected*

Vehicles and Expo (ICCVE), Las Vegas, NV, USA: IEEE, Dec. 2013, pp. 747–752. doi: 10.1109/ICCVE.2013.6799889.
[17] S. FENG, C. LIU, C. SHEN, H.-A. CHOI, and R. A. ROUIL, "An Effective and Efficient Dynamic eMBMS Multicast Grouping Scheduling Algorithm in MBSFNs for Public Safety Scenarios," *IEEE Access Pract. Innov. Open Solut.*, vol. 8, p. 10.1109/access.2020.3000251, 2020, doi: 10.1109/access.2020.3000251.
[18] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, May 2013, doi: 10.1109/MCOM.2013.6515060.
[19] D. Fischer, "TIME-TO-COLLISION AND COLLISION AVOIDANCE SYSTEMS".
[20] G. Avino *et al.*, "Support of Safety Services through Vehicular Communications: The Intersection Collision Avoidance Use Case," in *2018 International Conference of Electrical and Electronic Technologies for Automotive*, Milan: IEEE, Jul. 2018, pp. 1–6. doi: 10.23919/EETA.2018.8493191.
[21] M. Malinverno, F. Raviglione, C. Casetti, C.-F. Chiasserini, J. Mangues-Bafalluy, and M. Requena-Esteso, "A Multi-stack Simulation Framework for Vehicular Applications Testing," in *Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, Alicante Spain: ACM, Nov. 2020, pp. 17–24. doi: 10.1145/3416014.3424603.
[22] J. Santa, F. Pereniguez, A. Moragon, and A. F. Skarmeta, "Vehicle-to-infrastructure messaging proposal based on CAM/DENM specifications," in *2013 IFIP Wireless Days (WD)*, Valencia, Spain: IEEE, Nov. 2013, pp. 1–7. doi: 10.1109/WD.2013.6686514.
[23] "Performance Analysis of Existing 1609.2 Encodings v ASN.1".
[24] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, "Edge-Based Collision Avoidance for Vehicles and Vulnerable Users: An Architecture Based on MEC." arXiv, Jan. 06, 2021. doi: 10.48550/arXiv.1911.05299.
[25] J. Zhou and H. Peng, "Range Policy of Adaptive Cruise Control Vehicles for Improved Flow Stability and String Stability," *Intell. Transp. Syst. IEEE Trans. On*, vol. 6, pp. 229–237, Jul. 2005, doi: 10.1109/TITS.2005.848359.
[26] "Constant Spacing Strategies for Platooning in Automated Highway Systems".
[27] Civil Engineering Department, Amirkabir University of Technology, Tehran, Iran, E. Ramezani Khansari, M. Tabibi, Civil Engineering Department, Amirkabir University of Technology, Tehran, Iran, F. M. Nejad, and Civil Engineering Department, Amirkabir University of Technology, Tehran, Iran, "A Study on Following Behavior Based on the Time Headway," *J. Kejuruter.*, vol. 32, no. 2, pp. 187–195, May 2020, doi: 10.17576/jkukm-2020-32(2)-02.
[28] "COOPERATIVE DRIVING : BASIC CONCEPTS AND A FIRST ASSESSMENT OF 'INTELLIGENT CRUISE CONTROL' STRATEGIES," 1991. Accessed: Mar. 13, 2024. [Online]. Available: https://www.semanticscholar.org/paper/COOPERATIVE-DRIVING-%3A-BASIC-CONCEPTS-AND-A-FIRST-OF-Broqua/09437940cacf0c256277c5f67047eda24fcda335

[29] D. Yanakiev and I. Kanellakopoulos, “Nonlinear spacing policies for automated heavy-duty vehicles,” *IEEE Trans. Veh. Technol.*, vol. 47, no. 4, pp. 1365–1377, Nov. 1998, doi: 10.1109/25.728529.
[30] Z. Jiang, H. Zhang, and B. Yang, “An Improved Variable Time Headway Strategy For ACC,” in *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*, Shanghai China: ACM, Sep. 2019, pp. 293–299. doi: 10.1145/3366194.3366246.
[31] “Design and evaluation of an Integrated Full-Range Speed Assistant”.
[32] “Platoon management with cooperative adaptive cruise control enabled.pdf.”
[33] Z. (Sean) Qian, J. Li, X. Li, M. Zhang, and H. Wang, “Modeling heterogeneous traffic flow: A pragmatic approach,” *Transp. Res. Part B Methodol.*, vol. 99, pp. 183–204, May 2017, doi: 10.1016/j.trb.2017.01.011.
[34] S.-C. Lin *et al.*, “The Architectural Implications of Autonomous Driving: Constraints and Acceleration,” in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, Williamsburg VA USA: ACM, Mar. 2018, pp. 751–766. doi: 10.1145/3173162.3173191.
[35] M. Green, “‘How Long Does It Take to Stop?’ Methodological Analysis of Driver Perception-Brake Times,” *Transp. Hum. Factors*, vol. 2, no. 3, pp. 195–216, Sep. 2000, doi: 10.1207/STHF0203_1.
[36] G. Johansson and K. Rumar, “Drivers’ Brake Reaction Times,” *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 13, no. 1, pp. 23–27, Feb. 1971, doi: 10.1177/001872087101300104.
[37] “Performance_Analysis_of_C-V2I-Based_Automotive_Collision_Avoidance.pdf.”
[38] C. C. Lee, “Fuzzy logic in control systems: fuzzy logic controller. I,” *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404–418, Apr. 1990, doi: 10.1109/21.52551.
[39] “A Review of Car-Following Models and Modeling Tools for Human and Autonomous-Ready Driving Behaviors in Micro-Simulation.”
[40] Z. He, X. Qin, P. Liu, and M. A. Sayed, “Assessing Surrogate Safety Measures using a Safety Pilot Model Deployment Dataset,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2672, no. 38, pp. 1–11, Dec. 2018, doi: 10.1177/0361198118790861.
[41] Y. Matsui and S. Oikawa, “Characteristics of Dangerous Scenarios between Vehicles Turning Right and Pedestrians under Left-Hand Traffic,” *Appl. Sci.*, vol. 13, no. 7, Art. no. 7, Jan. 2023, doi: 10.3390/app13074189.
[42] L. Alberro, F. Velázquez, S. Azpiroz, E. Grampin, and M. Richart, “Experimenting with Routing Protocols in the Data Center: An ns-3 Simulation Approach,” *Future Internet*, vol. 14, no. 10, Art. no. 10, Oct. 2022, doi: 10.3390/fi14100292.
[43] B. Bojović, “5G-LENA NR MODULE OVERVIEW: MODELS, IMPLEMENTATION, NR-U EXTENSION AND EXAMPLES”.
[44] T. Zugno, M. Drago, S. Lagén, Z. Ali, and M. Zorzi, “Extending the ns-3 spatial channel model for vehicular scenarios,” in *Proceedings of the 2021 Workshop on ns-3*, in WNS3 ’21. New York, NY, USA: Association for Computing Machinery, Jun. 2021, pp. 25–32. doi: 10.1145/3460797.3460801.

[45] “SUMO Vehicle Trip Information.” Accessed: Apr. 29, 2024. [Online]. Available: https://sumo.dlr.de/docs/Simulation/Output/TripInfo.html

A Simulation Setup

This appendix provides a comprehensive explanation of the simulation framework used for the collision avoidance algorithm and its implementation within the ms-van3t framework. The ms-van3t framework is a powerful tool that offers a wide range of features, making it an ideal choice for the development and evaluation of our enhanced collision avoidance algorithm.

We will begin by revisiting the key features of ms-van3t and identifying the most essential ones that enable the implementation of our collision avoidance algorithm within its architecture. Furthermore, this appendix will provide a detailed guide on building the project using the author's GitHub repository. The repository contains not only the source code for the enhanced collision avoidance algorithm but also data analysis scripts that have been instrumental in generating the results presented in this thesis. The GitHub introduction and repository explanation will offer a step-by-step approach to setting up the simulation environment and running the experiments, ensuring that the results can be easily reproduced and validated.

By the end of this appendix, readers will have a thorough understanding of the ms-van3t framework, its key features, and how they have been leveraged to implement and evaluate our enhanced collision avoidance algorithm. The appendix will also serve as a valuable resource for those interested in building upon this work or conducting further research in the field of collision avoidance for connected vehicles.

A.1 Key Features of ms-van3t

ms-van3t is a simulation framework designed to help researchers develop V2X communication applications with a full-stack development platform to design and implement their own applications. The most important novel features of ms-van3t are:

a. multi-stack support for various access technologies (IEEE 802.11p, 3GPP LTE, 3GPP Release 14 LTE-V2X, 3GPP Release 16 5G NR-V2X): ms-van3t supports multiple access technologies, allowing users to evaluate V2X applications using different communication protocols. This feature is crucial for the implementation of our enhanced collision avoidance algorithm, as it enables testing and evaluation using various communication protocols, such as IEEE 802.11p, LTE-V2X, and 5G NR-V2X. By supporting multiple access technologies, ms-van3t provides a comprehensive assessment of the algorithm's performance under different network conditions and communication standards.

b. Large-scale simulation capabilities: The framework enables the simulation of a large number of vehicles, limited only by the hardware capabilities of the device running the simulation. It also provides a MetricSupervisor module for automatic data collection. Large-scale simulations in ms-van3t allow for the generation of extensive results, supporting the evaluation of the collision avoidance algorithm developed in this thesis work in realistic, dense traffic scenarios. This feature provides valuable insights into the

algorithm's performance and scalability. The MetricSupervisor module further facilitates the automatic collection of data, such as Packet Reception Ratio (PRR) and one-way latency, making it easier to analyze the algorithm's performance in large-scale scenarios.

c. Native integration with the SUMO traffic simulator: ms-van3t leverages the SUMO traffic simulator for vehicle mobility models and realistic urban and highway scenarios. The integration is enabled through the TraCI (Traffic Control Interface) module, which implements a bidirectional coupling between ns-3 and SUMO. The TraCI module allows ms-van3t to retrieve information from SUMO, such as vehicle position, velocity, and acceleration, and to directly control vehicle dynamics from ns-3. This integration enables the realization of complex scenarios, where connected vehicles can be actively controlled to simulate the presence of partially or fully autonomous vehicles. Additionally, the TraCI implementation in ms-van3t fully supports the simulation of Vulnerable Road Users (VRUs), such as pedestrians, which can be included in complex scenarios alongside connected vehicles and non-connected VRUs.

d. Support for pre-recorded GNSS traces: In addition to SUMO, ms-van3t supports the use of pre-recorded GNSS traces for vehicle mobility, allowing for the testing of V2X applications with realistic positioning data.

e. **Emulation mode for Hardware-in-the-Loop (HIL) testing:** ms-van3t enables emulation and communication with external entities, such as real vehicles, for HIL testing and the creation of hybrid scenarios.

f. **Full-fledged ETSI C-ITS stack implementation:** The framework includes a full implementation of the ETSI C-ITS stack, supporting standard-compliant simulations with vehicles exchanging messages (CAMs and DENMs) according to ETSI standards.

g. **Easy switching between different ETSI standard versions:** ms-van3t allows users to seamlessly switch between different versions of ETSI standards for CAMs and DENMs.

h**. Web-based visualizer for demonstration purposes:** A flexible GUI is provided, displaying vehicles on a map, which is particularly useful for demonstration purposes when SUMO is not used.

i. **Open-source availability:** ms-van3t is released under the GNU GPL version 2 license, allowing users to download, modify, and redistribute the framework for both private and commercial use.

A.2 Building the Project and Running Experiments

The GitHub repository for this thesis contains the source code for the collision avoidance algorithm with variable time headway and spacing control strategy and the necessary data analysis scripts. This section provides a step-by-step guide on building the project and running the experiments.

The GitHub repository for this project can be found at: https://github.com/VaishnaviMichiganTech/CAS-VTH-msvan3t. The setup starts from either cloning this repository or downloading the zip file. A prerequisite is that we must use Ubuntu 22.04 or above since the ns3 development directory needs the latest Ubuntu after 22.04 or above.

If this is the first time this repository is being run, the suggested method to run the repo by downloading dependencies is by first downloading or cloning the msvan3t repo from https://github.com/ms-van3t-devs/ms-van3t.

Then, install SUMO libraries by using these commands on the terminal:

sudo add-apt-repository ppa:sumo/stable
sudo apt update
sudo apt install sumo sumo-tools sumo-doc

Test SUMO by opening a terminal and running "sumo-gui".

Step 1: Clone the CAS-VTH-msvan3t repository.

1. Open a terminal and run the following command to clone the CAS-VTH-msvan3t repository:

   git clone https://github.com/VaishnaviMichiganTech/CAS-VTH-msvan3t.git

Alternatively, you can download the zip file of the repository and extract it.

Step 2: Install ms-van3t dependencies

1. Clone the ms-van3t repository:

git clone https://github.com/ms-van3t-devs/ms-van3t.git

2. Navigate to the ms-van3t repository:
   cd ms-van3t

3. Run the sandbox builder script:

   → If this is the first time installing ns-3 or ms-van3t on your system, run:
   ./sandbox_builder.sh install-dependencies

→ If this is not the first time installing ns-3, run:
./sandbox_builder.sh
Step 3: Configure and build ns-3 in the CAS-VTH-msvan3t repository

1. Navigate to the CAS-VTH-msvan3t repository:
   cd CAS-VTH-msvan3t/ns-3-dev
2. Clean the ns-3 repository of previous builds:
   ./ns3 clean
3. Configure ns-3 with the following command:
   ./ns3 configure --build-profile=optimized --enable-examples --enable-tests --disable-python --disable-werror

   The --disable-werror flag is required for Ubuntu 22.04 LTS or later.
4. Build ns-3*:*
   ./ns3 build

Step 4: Run simulations.

1. Navigate to the src directory:
   cd src
2. Run the Start Simulation script:
   ./StartSimualtions

   This shell script runs a set of simulations with varying maximum speed and time values. If needed, you can modify this script for further experimentation with different speeds and command-line features.

Folder structure and important files:

- src/automotive/: Contains application-related files and source code implementing the ETSI ITS-G5 stack for vehicular communications.
- src/automotive/examples/: Contains the application classes (explained in Appendix B).
- src/automotive/model/applications/: Contains the collision avoidance algorithm scripts (server, client, and collision avoidance system application).
  - v2i-lte.cc: Collision avoidance application
  - appServer.cc: Server application
  - appClient.cc: Client application
- src/automotive/examples/sumo-files/: Contains SUMO trace files for running simulations.
- src/automotive/Facilities/: Contains modules for CAM and DENM dissemination logics.
- switch_ETSI_version.sh: Script to switch between CAM and DENM message versions (v1 or v2). Do not modify the ns-3-dev/src/automotive/model/ASN1/currmode.txt file manually.

By following these steps, you should be able to build the project and run the experiments from the CAS-VTH-msvan3t GitHub repository. The provided folder structure and

scripts will help you navigate the codebase and perform simulations with the enhanced collision avoidance algorithm.

A.3 Summary

This appendix has provided a comprehensive overview of the ms-van3t simulation framework and its key features that have enabled the development and evaluation of our enhanced collision avoidance algorithm. The multi-stack support, large-scale simulation capabilities, and native integration with SUMO have been particularly instrumental in assessing the algorithm's performance under various network conditions and traffic scenarios.

The GitHub repository accompanying this thesis serves as a valuable resource for readers interested in building upon this work or conducting further research in the field of collision avoidance for connected vehicles. By following the provided instructions, readers can set up the simulation environment, run the experiments, and reproduce the results presented in this thesis.

As the field of connected vehicles continues to evolve, frameworks like ms-van3t will play a crucial role in the development and evaluation of novel V2X applications, including advanced collision avoidance algorithms. The open-source nature of ms-van3t encourages collaboration and further advancements in this domain, ultimately contributing to safer and more efficient transportation systems.

B Collision Avoidance System: Software Architecture

As discussed in the previous chapters on this thesis work, Collision Avoidance system is a client-server-based application where the edge, which in real-time will be an infrastructural entity that we refer to as Roadside Unit, on which the server application runs the collision avoidance(eCA) application. Collision avoidance system application receives its inputs in the form of the vehicle's real time data packaged within CAM-Cooperative Awareness Messaging protocol, which is generated by nodes in the network, which is the client application that is run within the Vehicles' On Board Units.

There are three major classes in this software architecture: *appClient, appServer, and CollisionAvoidanceSystem.*
The *appServer c*lass is implemented to represent the server side of an application running on a remote host in the simulation.
The *appClient* represents the client side of the application running on the remote nodes that are representation of the vehicles in the simulation and triggers the data communication from vehicles to the host- edge node. *CollisionAvoidanceSystem* algorithm is designed to access the scenario which accommodates the vehicles in the simulation and detecting possible collisions at the intersections of the road infrastructure, and triggering the event based- DENMs to the vehicles which can take further actions.

An application algorithm that forms this network topology, and runs these applications periodically is included in the Automotive module of NS-3. In the further sub-sections, these classes are introduced along with their function and a detailed flowchart too.

B.1 V2I application

The application is compatible with LTE and 802.11p wireless communication stack provided within the msvanet simulation framework. The v2i-lte and v2i-802.11p applications are the two variations. This application sets up the vehicle network simulation using LTE and 802.11p, and SUMO traffic simulator. For the analysis and implementation of collision avoidance system within this thesis work, we work around with v2i-lte to showcase the real-time behavior of collision avoidance algorithm when implemented in road infrastructure using LTE- 5G services.

The v2i-lte application contains the main code for setting up and running the vehicular network simulation using LTE and SUMO. This application is responsible for configuring the simulation environment, creating the necessary objects, and orchestrating the interactions between the *appClient*, *appServer*, and *collisionAvoidanceSystem* applications. The main functionalities of this code include parsing command line arguments, setting up the LTE network topology, creating the remote host (server) and containers for UEs (vehicles) and eNB (base station). The functions within this algorithm help in installing mobility models and LTE devices, setting up TraCI and starting SUMO, creating, and configuring the appServer and appClient applications, defining callback

functions for node creation and shutdown, and running the simulation. The code also includes utility functions for reading the number of vehicles from the mobility trace file and setting up the simulation time. Next, we will study its pseudo code and stepwise explain the algorithm to set a base for further discussion in this section.

Pseudo code:

Figure 50 V2I-Lte application Psuedo Code- Part 1 (source : author)

Parsing command line arguments:

The code uses the *CommandLine* class from ns-3 to define and parse command line arguments.

- It allows the user to specify various simulation parameters, such as the path to SUMO files, mobility trace file, SUMO configuration file, LTE parameters, simulation time, and more.
- The parsed values are stored in corresponding variables for later use in the simulation.

Setting up the LTE network topology:

- The application creates an instance of the *LteHelper* class, which is used to set up the LTE network.
- It also creates an instance of the *PointToPointEpcHelper* class to configure the Evolved Packet Core (EPC) network.

- The *LteHelper* is configured with the *PointToPointEpcHelper* to enable the integration of LTE with the EPC network.

Creating the remote host (server):

- The code creates a remote host node using the *NodeContainer* class and installs the *InternetStackHelper* on it.
- It sets up a point-to-point link between the remote host and the Packet Gateway (PGW) node of the EPC network.
- IP addresses are assigned to the remote host and the PGW using the *Ipv4AddressHelper*.

Creating containers for UEs (vehicles) and eNB (base station):

- The code creates node containers for UEs (vehicles) and eNB (base station) using the NodeContainer class.
- The number of UE nodes is determined by reading the mobility trace file (sumo-files/cars.rou.xml) and counting the number of vehicle elements.

Installing mobility models:

- The code installs mobility models on the eNB and UE nodes using the *MobilityHelper* class.
- The eNB is set to a fixed position using the *SetPosition* method of the *MobilityModel* class.

Installing LTE devices:

- The code installs LTE devices on the eNB and UE nodes using the *InstallEnbDevice* and *InstallUeDevice* methods of the *LteHelper* class.
- It also installs the *InternetStackHelper* on the UE nodes and assigns IP addresses to them using the *AssignUeIpv4Address* method of the *PointToPointEpcHelper*.

```
SETUP TRACI and START SUMO
   CREATE TraciClient
   SET attributes for TraciClient

CREATE and SETUP application for server
   CREATE AppServerHelper
   SET attributes for AppServerHelper

SETUP interface and application for dynamic nodes
   CREATE AppClientHelper
   SET attributes for AppClientHelper

DEFINE callback functions for node creation and shutdown
   FUNCTION setupNewWifiNode(vehicleID)
      // Node creation logic
      RETURN node

   FUNCTION shutdownWifiNode(exNode, vehicleID)
      // Node shutdown logic

START TRACI client with defined callback functions

START simulation
   SET stop time
   RUN simulation
   DESTROY simulation

RETURN 0
```

Figure 51 V2I-Lte application Psuedo Code- Part 2 (source : author)

Setting up *TraCI* and starting *SUMO*:

- The code creates an instance of the *TraciClient* class, which is used to interface with the SUMO traffic simulator.
- It sets various attributes of the *TraciClient*, such as the SUMO configuration file, time step, and more.
- The *TraciClient* is configured with callback functions for setting up and shutting down nodes during the simulation.

Creating and configuring the *appServer* and *appClient* applications:

- The code creates instances of the *AppServerHelper* and *AppClientHelper* classes to configure and install the server and client applications.
- The *AppServerHelper* is used to install the appServer application on the remote host node.
- The *AppClientHelper* is used to install the appClient application on each UE node/vehicular node.
- Various attributes of the appServer and appClient applications are set, such as the server address, *TraCI* client, and more.

Defining callback functions for node creation and shutdown:

- The code defines two callback functions: *setupNewWifiNode* and *shutdownWifiNode*.
- *setupNewWifiNode* is called by the *TraciClient* when a new vehicle enters the simulation. It retrieves a UE node from the node pool, installs the *appClient* application on it, and returns the node.
- *shutdownWifiNode* is called by the *TraciClient* when a vehicle exits the simulation. It stops the *appClient* application on the node and moves the node outside the communication range.

Running the simulation:

- The code sets the simulation stop time using *Simulator::Stop* based on the specified simulation duration.
- It starts the simulation by calling *Simulator::Run*.
- After the simulation ends, it calls *Simulator::Destroy* to clean up the simulation environment.

Utility functions:

- The code includes utility functions for reading the number of vehicles from the mobility trace file (sumo-files/cars.rou.xml) using the *xmlparser* library.
- It also sets the simulation time based on the specified duration.

B.2 Server Application

Implementation of the *appServer* class represents the server- end of application running on a remote host in the simulation. It receives Cooperative Awareness Messages (CAMs) from vehicles, processes them using the Collision Avoidance System (CAS) (*collisionAvoidanceSystem* class*)* , and triggers the transmission of Decentralized Environmental Notification Messages (DENMs) when a potential collision is detected. The main functionalities include setting up communication sockets, configuring DEN and CA services, logging, handling CAMs, triggering DENMs, and printing aggregate output.

Pseudo Code:

```
DEFINE appServer class: INHERITS Application
  DEFINE member variables
    - m_client: Pointer to TraciClient object
    - m_aggregate_output: Boolean flag for aggregate output
    - m_real_time: Boolean flag for real-time mode
    - m_csv_name: String for CSV file name
    - m_csv_ofstream_cam: Output file stream for CSV file
    - m_t2c_threshold: Double value for t2c threshold
    - m_speed: Double value for vehicle speed
    - m_socket: Pointer to Socket object
    - m_denService: DENBasicService object
    - m_caService: CABasicService object
    - m_aggegateOutputEvent: EventId for aggregate output event
    - m_cam_received: Integer count of received CAMs
    - m_denm_sent: Integer count of sent DENMs
    - m_cumulate_delay: Double value for cumulated delay
    - m_collisionAvoidanceSystem: CollisionAvoidanceSystem object

  DEFINE GetTypeId method: STATIC
    RETURN TypeId with attributes and constructor for appServer

  DEFINE constructor
    INITIALIZE member variables

  DEFINE destructor

  DEFINE DoDispose method
    CALL Application::DoDispose()

  DEFINE StartApplication method
    CREATE UDP socket for TX and RX
    BIND socket to receive packets from any IP
    SET receive callback for CAMs using CABasicService
    SET socket, callback, and station properties in DENBasicService
    SET station properties in CABasicService
    IF CSV name is provided THEN
      OPEN CSV file for output
    IF aggregate output is enabled THEN
      SCHEDULE aggregate output event
    SET DENM callback for CollisionAvoidanceSystem
    SET t2c threshold, real-time mode, and free speed for CollisionAvoidanceSystem
```

Figure 52 Server Application Psuedo code- Part 1 (source : author)

```
DEFINE StopApplication method
  CANCEL aggregate output event
  COMPUTE average delay from cumulated delay and CAM received count
  CLEAN up DENBasicService
  IF CSV name is provided THEN
    WRITE data to CSV file and close it
  IF aggregate output is enabled THEN
    PRINT summary

DEFINE StopApplicationNow method
  CALL StopApplication()

DEFINE TriggerDenm method
  CONVERT XY coordinates to longitude and latitude
  BUILD DENM data with collision position and situation data
  CONNECT socket to the sender's address
  TRIGGER DENM using DENBasicService
  INCREMENT DENM sent count
  FREE DENM data

DEFINE receiveCAM method
  INCREMENT CAM received count
  CONVERT CAM data to cam_information_t structure
  UPDATE collision avoidance system maps with CAM information
  COMPUTE message delay
  UPDATE cumulated delay
  FREE CAM data

DEFINE aggregateOutput method
  PRINT current time, CAM received count, and DENM sent count
  SCHEDULE next aggregate output event
```

Figure 53 Server Application Psuedo code- Part 2 (source : author)

Here's a step-wise explanation of the code flow in *appServer* class:

a. The necessary header files are included, such as "appServer.h", "ns3/CAM.h", "ns3/DENM.h", and "ns3/socket.h".

b. The *appServer* class is defined within the ns3 namespace.

c. The *GetTypeId* function is defined, which returns the *TypeId* of the appServer class and sets various attributes such as *AggregateOutput*, *RealTime*, *CSV*, *T2C*, and *Client*.

d. The constructor initializes member variables and sets default values.

e. The *DoDispose* function is defined to perform cleanup tasks when the application is being disposed of.

f. The *StartApplication* function is called when the application starts. It sets up the socket for communication, configures the DEN and CA services, opens a CSV file for logging (if specified), and schedules the *aggregateOutput* function (if enabled).

g. The *StopApplication* function is called when the application stops. It calculates the *aggregateOutput* event, computes the average delay, performs cleanup tasks, and prints summary information.

h. The *TriggerDenm* function is called by the CAS when a potential collision is detected. It builds the DENM data and triggers the transmission of the DENM to the involved vehicles.

i. The *receiveCAM* function is called when a CAM is received from a vehicle. It extracts relevant information from the CAM, updates the collision detection maps in the CAS, and computes the message delay.

j. The *aggregateOutput* function is periodically called (if enabled) to print aggregate output, such as the number of CAMs received and DENMs sent.

k. The *PrintCAMinfo* function is a utility function to print the information contained in a CAM.

l. The *compute_timestamp* function is a utility function to compute the current timestamp.

Thus, the main functionalities of this code include setting up the socket for communication, configuring the DEN and CA services, opening a CSV file for logging, handling received CAMs, triggering DENM transmission when a collision is detected, and periodically printing aggregate output. The code also includes utility functions for processing received CAMs and updating collision detection information.

B.3 Collision Avoidance Application

The *collisionAvoidanceSystem* class plays a crucial role in the enhanced Collision Avoidance application by implementing the Collision Avoidance System (CAS). This system is designed to detect potential collisions between vehicles using information received from Cooperative Awareness Messages (CAMs). This algorithm encompasses a range of functionalities, including updating vehicle information maps for every simulation step, checking for potential collisions, determining collision types, verifying collision detection ranges, calculating collision metrics, and triggering the transmission of warning messages to vehicles at risk of collision. The code also includes utility functions to support these main functionalities.

To better understand the inner workings of the Collision Avoidance System, let's delve into the pseudo code that explains the key functions and their roles in the collision avoidance process.

Pseudo code:

```
DEFINE CollisionAvoidanceSystem class
  DEFINE member variables
    - m_veh_positions: Map to store vehicle positions
    - m_veh_speed: Map to store vehicle speeds
    - m_veh_acceleration: Map to store vehicle accelerations
    - m_veh_angle: Map to store vehicle angles
    - m_veh_last_update: Map to store last update times for vehicles
    - m_veh_cam_generation_time: Map to store CAM generation times for vehicles
    - m_veh_width: Map to store vehicle widths
    - m_veh_length: Map to store vehicle lengths
    - m_veh_address: Map to store vehicle addresses
    - m_veh_type: Map to store vehicle types
    - m_t2c_threshold: Threshold for time to collision
    - m_max_pos_error: Maximum position error
    - m_real_time: Flag for real-time mode
    - m_DENMsendCallback: Callback function for sending DENMs

  DEFINE constructor
    INITIALIZE member variables

  DEFINE UpdateMaps method
    UPDATE vehicle position, speed, acceleration, angle, last update time, CAM generation time, width, length, address, and type in
respective maps

  DEFINE IterateAndCheck method
    FOR each vehicle in m_veh_positions
      GET vehicle position, speed, acceleration, and type
      IF vehicle type is passenger car THEN
        FOR each other vehicle in m_veh_positions
          IF other vehicle type is passenger car AND last update time is recent THEN
            GET other vehicle position, speed, and acceleration
            IF CollisionType indicates collision AND IsinRange returns true THEN
              COMPUTE time to collision (t2c) and space to collision (s2c) using CheckCollision
              IF t2c and s2c are below thresholds THEN
                CALL m_DENMsendCallback with vehicle information
```

Figure 54 Collision avoidance application Psuedo code- Part 1 (source : author)

Let's dive into more detail about the collisionAvoidanceSystem.cpp file and its functionalities, step by step.

1. The collisionAvoidanceSystem class starts by including the necessary header files, such as "collisionAvoidanceSystem.h", "ns3/CAM.h", "ns3/DENM.h", and other standard C++ libraries.

2. The *CollisionAvoidanceSystem::UpdateMaps* function is responsible for updating the vehicle information maps with the received CAM data. It takes a *cam_information_t* structure as input, which contains the CAM information received from a vehicle. The function performs the following steps:

    - Extracts the position, speed, angle, and acceleration information from the CAM.

    - Converts the angle from degrees to radians and calculates the velocity and acceleration components (vx, vy, ax, ay).

    - Updates the vehicle information maps (*m_veh_positions*, *m_veh_speed*, *m_veh_acceleration*, etc.) with the received CAM data, using the vehicle ID as the key.

    - Calls the *CollisionAvoidanceSystem*::*IterateAndCheck* function to check for potential collisions with the updated vehicle information.

3. The *CollisionAvoidanceSystem*::*IterateAndCheck* function iterates through all the vehicle pairs to check for potential collisions. It takes the ID of the first vehicle (vehicle_id1) as input and performs the following steps:

    - Retrieves the position, speed, acceleration, and type information of vehicle_id1 from the vehicle information maps.

    - Iterates through all the other vehicles in the *m_veh_positions* map.

    - Skips checking the vehicle with itself and vulnerable users (pedestrians, cyclists).

    - Checks if the information of the other vehicle is outdated based on the last update time.

    - Retrieves the position, speed, and acceleration information of the other vehicle (*vehicle_id2*) from the vehicle information maps.

    - Calls the *CollisionAvoidanceSystem::CollisionType* function to determine the type of collision (rear-end or crossing) between the two vehicles.

    - If the collision type is crossing, calls the *CollisionAvoidanceSystem::IsInRange* function to check if the vehicles are within a certain range for collision detection.

- If the vehicles are in range, computes the space to collision (STC) threshold based on the vehicle dimensions and calls the *CollisionAvoidanceSystem::CheckCollision* function to compute the time to collision (TTC) and STC.

- If the TTC is greater than zero (indicating a potential collision), triggers the transmission of DENMs to both vehicles involved in the potential collision.

```
DEFINE CollisionType method
  COMPUTE vehicle speeds and angles
  IF angle difference is below threshold THEN
    RETURN false (rear-end collision)
  ELSE
    RETURN true (crossroad collision)

DEFINE IsInRange method
  COMPUTE future positions of vehicles after t2c_threshold
  COMPUTE distances traveled by vehicles and maximum distance
  IF current distance between vehicles is less than maximum distance THEN
    RETURN true
  ELSE
    RETURN false

DEFINE ComputeS2C method
  RETURN square root of sum of squares of vehicle width/2 and vehicle length + width/2 + max position error

DEFINE CheckCollision method
  COMPUTE current distance between vehicles
  IF current distance is less than s2c_threshold THEN
    SET t2c to a small value and s2c to current distance
    RETURN t2c and s2c
  ELSE
    COMPUTE time to collision (t2c) using TimeToCollision
    IF t2c is NO_COLLISION or greater than t2c_threshold THEN
      SET t2c and s2c to NO_COLLISION
      RETURN t2c and s2c
    ELSE
      COMPUTE space to collision (s2c) using SpaceToCollision
      IF s2c is greater than s2c_threshold THEN
        SET t2c and s2c to NO_COLLISION
        RETURN t2c and s2c
      ELSE
        RETURN t2c and s2c
```

Figure 55 Collision avoidance application Psuedo code - Part 2 (source : author)

4. The *CollisionAvoidanceSystem::CollisionType* function determines the type of collision (rear-end or crossing) between two vehicles based on their velocity vectors. It takes the velocity components (vx1, vy1, vx2, vy2) of the two vehicles as input and performs the following steps:

    - Computes the speed modules (vel1, vel2) and angles (angle1, angle2) of the velocity vectors.

    - Compares the absolute difference between the angles with a predefined angular threshold.

    - Returns false if the angle difference is less than the threshold (indicating a rear-end collision) or true otherwise (indicating a crossing collision).

5. The *CollisionAvoidanceSystem::IsInRange* function checks if two vehicles are within a certain range for collision detection. It takes the positions (p1x, p1y, p2x, p2y) and speeds (speed1x, speed1y, speed2x, speed2y) of the two vehicles as input and performs the following steps:

- Computes the future positions of the vehicles after a specified time threshold (*m_t2c_threshold*).
- Calculates the distances traveled by each vehicle during that time.
- Determines the maximum distance traveled between the two vehicles.
- Computes the current distance between the vehicles.
- Returns true if the current distance is less than the maximum distance (indicating that the vehicles are in range for collision detection) or false otherwise.

6. The *CollisionAvoidanceSystem::ComputeS2C* function computes the space to collision (STC) threshold based on the vehicle dimensions. It takes the length and width of the vehicle as input and returns the computed STC threshold.

7. The *CollisionAvoidanceSystem::CheckCollision* function computes the time to collision (TTC) and space to collision (STC) between two vehicles. It takes the positions (x1, y1, x2, y2), velocities (dx1, dy1, dx2, dy2), accelerations (ax1, ay1, ax2, ay2), and the STC threshold as input and performs the following steps:
    - Calls the *CollisionAvoidanceSystem::TimeToCollision* function to compute the TTC between the vehicles.
    - If the TTC is NO_COLLISION or greater than the specified TTC threshold (m_t2c_threshold), returns NO_COLLISION.
    - Otherwise, calls the *CollisionAvoidanceSystem::SpaceToCollision* function to compute the STC between the vehicles at the time of collision.
    - If the STC is greater than the specified STC threshold, returns NO_COLLISION.
    - Otherwise, returns the computed TTC, STC, and the coordinates of the collision point.

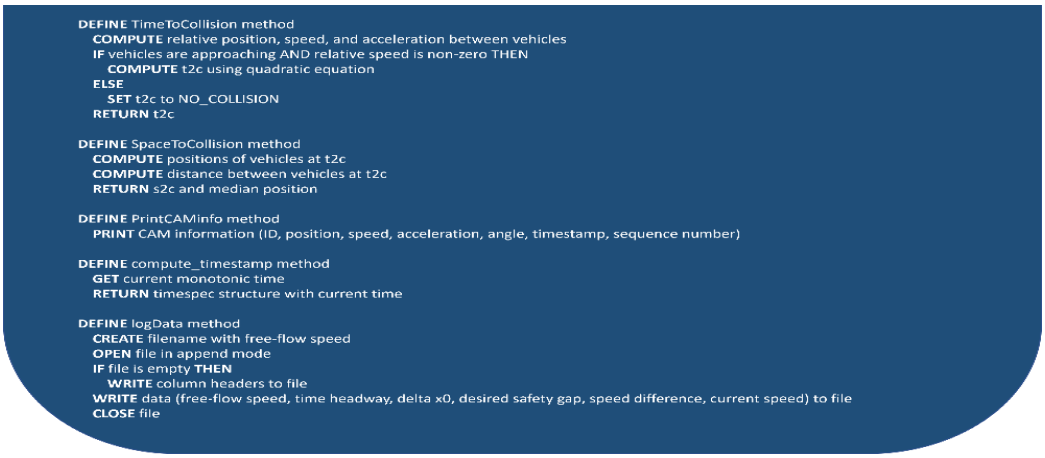

Figure 56 Collision avoidance application Psuedo code- Part 3 (source : author)

8. The *CollisionAvoidanceSystem::TimeToCollision* function computes the time to collision (TTC) between two vehicles. It takes the positions (x1, y1, x2, y2), velocities (vx1, vy1, vx2, vy2), and accelerations (ax1, ay1, ax2, ay2) of the vehicles as input and performs the following steps:

    - Computes the distance and velocity difference components between the vehicles.
    - Calculates the TTC components (t2c_x, t2c_y, t2c_v) based on the distance and velocity differences.
    - Checks if the vehicles are approaching each other and if their relative velocity is non-zero.
    - If the conditions are met, computes the TTC using the TTC components.
    - If the acceleration difference is small, returns the computed TTC.
    - Otherwise, solves a third-degree equation to find the minimum positive TTC considering accelerations.

9. The *CollisionAvoidanceSystem::SpaceToCollision* function computes the space to collision (STC) between two vehicles at the time of collision. It takes the positions (x1, y1, x2, y2), velocities (vx1, vy1, vx2, vy2), accelerations (ax1, ay1, ax2, ay2), and the TTC as input and performs the following steps:

    - Computes the future positions of the vehicles at the time of collision (TTC) considering their velocities and accelerations.

- Calculates the distance between the future positions of the vehicles (STC).
- Returns the computed STC and the coordinates of the collision point.

10. The *CollisionAvoidanceSystem::PrintCAMinfo* function is a utility function that prints the information contained in a CAM message, such as the vehicle ID, position, speed, acceleration, angle, timestamp, and sequence number.

11. The *CollisionAvoidanceSystem::compute_timestamp* function is another utility function that computes the current timestamp using the clock_gettime function and returns it as a struct timespec.

These are the main functionalities and steps involved in the collisionAvoidanceSystem class. This code implements the core logic of the Collision Avoidance System, which receives CAM data, updates vehicle information, checks for potential collisions, computes TTC and STC, and triggers the transmission of DENMs when necessary.

B.4 Client Application

The appClient.cpp file implements the appClient class, which represents the client-side application running on each vehicle in the simulation. This class is responsible for sending Cooperative Awareness Messages (CAMs) and receiving Decentralized Environmental Notification Messages (DENMs). It sets up the necessary sockets and services for communication with the server and other vehicles.

Pseudo Code:

Figure 57 Client Application psuedo code - Part 1 (source : author)

Here's a detailed step-wise flow of the code in the appClient class:

1. The StartApplication() method is called when the application starts. It performs the following tasks:
    - Initializes the client by setting up the socket for communication using the Socket::CreateSocket() function with the appropriate socket factory (e.g., UdpSocketFactory).

- Binds the socket to receive packets using the Socket::Bind() function with the local IP address and port number (e.g., Ipv4Address::GetAny() and port 9).
- Sets up the DEN and CA services by configuring their properties, such as the station ID (e.g., std::stol(m_id.substr (3))), station type (e.g., StationType_passengerCar), and socket for communication.
- Opens a CSV file for logging if the *m_csv_name* parameter is provided.
- Schedules the periodic transmission of CAMs by calling the *startCamDissemination()* function of the CA service with a random desynchronization time

```
DEFINE StartApplication method
  GET vehicle ID from TraCI client
  SET initial speed using TraCI API
  DETERMINE station type based on vehicle class
  CREATE socket for TX and RX
  BIND socket to receive packets from any IP
  CONNECT socket to the server
  SET receive callback for DENMs using DENBasicService::receiveDENM
  CONFIGURE DENBasicService with station properties, callback, and real-time flag
  CONFIGURE CABasicService with socket, station properties, VDP, and real time flag
  OPEN CSV file for logging if CSV name is provided
  SCHEDULE CAM dissemination with random desynchronization

DEFINE StopApplication method
  REMOVE CAM dissemination event
  SAVE waiting time to CSV using SaveWaitingTimeToCSV()
  TERMINATE CAM dissemination using CABasicService::terminateDissemination()
  CLEANUP DENBasicService
  CANCEL check event
  PRINT summary if requested and not already printed

DEFINE StopApplicationNow method
  CALL StopApplication()

DEFINE receiveDENM method
  INCREMENT DENM received count
  GET current position using TraCI API and convert to LonLat
  COMPUTE distance from the collision using distanceEarth()
  UPDATE cumulated delay
  DETERMINE other vehicle type based on DENM cause code
  LOG data to CSV file if CSV name is provided
  IF evasive maneuver is enabled THEN
    CALL evasiveManeuver()

DEFINE evasiveManeuver method
  COMPUTE safety radius based on speed and time value
  GET leader information using TraCI API
  IF leader is present THEN
    CALCULATE time headway, desired safety gap, and deltaX0 using CollisionAvoidanceSystem
    CALCULATE acceleration based on leader's acceleration, speed, and distance
    SET new speed based on acceleration
  ELSE
    CALCULATE new speed based on distance from collision and safety radius
  SET vehicle speed to new speed using TraCI API
  UPDATE status to CONTROLLED
  CANCEL previous check event
  SCHEDULE new check event to monitor collision distance

DEFINE RecordWaitingTime method
  GET accumulated waiting time from TraCI API
  UPDATE total waiting time
  GET current speed using TraCI API
  UPDATE travel time
  CREATE CSV file name with speed value and vehicle ID
  OPEN CSV file for appending
  WRITE waiting time, travel time, and new speed data to CSV
  CLOSE CSV file
  UPDATE previous new speed values
  SCHEDULE next waiting time recording after 1 second

DEFINE SaveWaitingTimeToCSV method
  CREATE CSV file name with vehicle ID
  OPEN CSV file for appending
  WRITE waiting time, travel time, and new speed data to CSV
  CLOSE CSV file

DEFINE checkCollisionDist method
  GET current position using TraCI API and convert to LonLat
  COMPUTE distance from the collision using distanceEarth()
  IF distance is increasing compared to previous distance THEN
    SET vehicle speed back to normal using TraCI API
    UPDATE status to FREE
    CANCEL check event
  ELSE
    SCHEDULE next check event to monitor collision distance after 1 second
```

Figure 58 Client application Psuedo code- Part 2 (source : author)

2. The *StopApplication()* method is called when the application stops. It performs the following tasks:
    - Cancels the periodic transmission of CAMs by calling the *terminateDissemination()* function of the CA service.
    - Cleans up the resources and closes the socket connections.
    - Writes the summary statistics to the CSV file if logging is enabled.
    - Prints the summary information, such as the number of CAMs sent and DENMs received, if the *m_print_summary* flag is set to true.
3. The *receiveDENM()* method is called when a DENM is received from the server. It performs the following tasks:
    - Updates the internal statistics, such as the number of DENMs received (*m_denm_received*) and the cumulative delay *(m_cumulated_delay*).
    - Extracts relevant information from the received DENM, such as the position of the vehicle and the distance to the potential collision point.
    - Logs the received DENM information to the CSV file if logging is enabled.
    - If the vehicle is configured to perform evasive maneuvers (i.e., *m_evasive_maneuver flag* is set to true), it calls the *evasiveManeuver*() function to calculate the safety radius and adjust the vehicle's speed based on the distance from the potential collision point.
4. The *evasiveManeuver*() method is called when an evasive maneuver needs to be performed. It performs the following tasks:
    - Calculates the safety radius based on the vehicle's current speed, acceleration, and a predefined time value (m_time_val).
    - Computes the distance from the potential collision point using the *distanceEarth*() function.
    - Calculates a coefficient (coeff_speed) based on the ratio of the distance from the collision point to the safety radius.
    - Adjusts the vehicle's speed using the *TraCIAPI::vehicle.setSpeed()* function based on the calculated coefficient.
    - Schedules the *checkCollisionDist()* function to periodically check the distance from the collision point.
5. The *checkCollisionDist*() method is called periodically to check the distance from the collision point. It performs the following tasks:
    - Computes the current distance from the collision point using the *distanceEarth()* function.
    - If the distance is increasing compared to the previous distance, it resets the vehicle's speed to the original value using the *TraCIAPI::vehicle.setSpeed()* function and cancels further checks.
    - If the distance is not increasing, it schedules the *checkCollisionDist*() function to be called again after a specified time interval (e.g., 1 second).

The appClient class also includes utility functions, such as *distanceEarth*() for calculating the distance between two points on Earth using their latitude and longitude coordinates.

Overall, the appClient class plays a vital role in the simulation by handling the client-side functionality of sending CAMs, receiving DENMs, and performing evasive maneuvers when necessary. It interacts with the server (appServer) and the TraCI API to communicate with the SUMO traffic simulator and control the vehicle's behavior based on the received notifications.