B.Sc Dissertation in Electrical and Electronics Engineering

Development of communication and networking protocols for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication systems

Adnan Ahmetović

190302079

Faculty of Engineering and Natural Sciences

International University of Sarajevo

June 2023

Acknowledgements:

I would like to express my deepest gratitude and appreciation to Dr. Prof. Tarik Namas for his invaluable guidance, unwavering support, and immense contribution throughout the process of writing my Bachelor's thesis.

Abstract

This thesis is produced under BSc Electrical Engineer Dissertation module covering the works for V2V and V2I communication protocols and proposes a novel V2I and V2V cooperative communication mechanism. It is separated into III chapters, where first two chapters performs a thorough analysis of the existing networking and communication protocols for V2V and V2I communication systems to pinpoint their advantages and disadvantages, along with analysis and insight of proposed novel design protocols in recent years. Focused on the most promising one, V2X that utilizes 5G. 5G can provide a low-time, high-bandwidth and high-reliability operating environment for V2X a5pplication. On the other hand, V2X, which enables vehicle-to-car, vehicle-to-road communications, can support L1~L3 level automatic driving, mainly carrying basic traffic safety operations, such as traffic accident reminders, emergency brake warnings, and intersection collision warnings. Third chapter focuses on the development of the novel design communication protocol, proposing a data forwarding mechanism that enables the vehicle to cooperate with the RSU and the vehicle in the opposite direction. Proposed solution focuses on developing a transmission scheme that maximizes the aggregate throughput of the target vehicle while minimizing download interruptions when is out of coverage. Thesis will be finalized with simulation it and results will be compared to already proposed solutions.

Keywords- Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), V2X, 5G, Autonomous driving, Data sharing, Road safety, Communication protocols, Roadside infrastructure (RSU)

Table of Contents

List of Equations:

List of Figures:

List of Tables:

List of Acronyms (sort by A-Z)

BSM	Basic Safety Message
CAN	Controller Area Network
CAV	Connected and Autonomous Vehicles
CPE	Customer Premises Equipment
CPU	Central Processing Unit
C-V2X	Cellular Vehicle-to-Everything
DSRC	Dedicated Short-Range Communications
GPS	Global Positioning System
HV	Host Vehicle
ICW	Intra-vehicle Communication Wireless
LTE-V2X	
MEC	Mobile Edge Computing
RSU	Roadside Unit
RV	Remote vehicle
TTC	Time to Collision
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular Ad-Hoc Network

List of Nomenclatures (sort by first appearance):

$\boldsymbol{p}$**-** Vehicle density

$\boldsymbol{\lambda}$**-** Arrival rate

$\boldsymbol{E[X_W]}$**-** Willingness of the vehicle to act as a relay

μ- Fading parameter

$\boldsymbol{U_n}$**-** Coverage

$\boldsymbol{O_F}$**-** Probability of packet loss caused by buffer overflow

$\boldsymbol{p_{min}}$**-** Service threshold

$\boldsymbol{\sigma^2}$**-** Service time variance

$\boldsymbol{\gamma}$**-** Probability of a packet smaller than the service threshold in the buffer

$\boldsymbol{E[X_F]}$**-** Average number of interruptions

E[C]- Average transmission rate

$\boldsymbol{\xi}$**-** Traffic load

$\boldsymbol{F_h}$**-** Transfers in multiple paths within coverage

Chapter I

Introduction

[1-3] Data from accelerometers, gyroscopes, and wheel speed sensors provides insights into the vehicle's dynamics and movement. Proximity sensors, cameras, lidar, and radar sensors detect obstacles, pedestrians, and other vehicles, ensuring safe navigation. Engine and performance data, including information about fuel efficiency and emissions, help optimize engine operation and identify faults. GPS and navigation data, combined with mapping information, enable cars to determine their location accurately and provide real-time directions and traffic updates. Internet connectivity allows cars to exchange data with external sources, such as cloud platforms, other vehicles, and service centers, facilitating software updates, traffic information, and remote diagnostics. Human-machine interfaces, like touchscreens and voice commands, enable users to interact with the car and access various features. Environmental data, such as temperature and humidity, is considered for climate control and efficient system operation. Maintenance data assists in scheduling service tasks and optimizing vehicle lifespan. Safety systems data from airbag sensors and stability control systems ensures prompt responses to potential risks. Driver behavior data, including steering input and braking patterns, can be used to analyze driving habits and provide feedback. By leveraging this diverse range of information, cars can operate safely, optimize performance, navigate efficiently, and offer a comfortable and connected driving experience.[4] This could make a bit impact on future of insurance as well, whereas insurance companies would be able to take the cars sensory data and analyze drivers’ behavior, driving style, parking habit, etc. Based on this, they would be able to make policies proportional to their risk assessment, i.e. If the cars telemetry data portray increased fuel consumption, increased G force value, aggressive accelerations, we can suspect that the driver has a habit of aggressive driving, which, when combined with others cars data, such as from steering position sensor, stability control, engine data, etc. concludes that driver is indeed aggressive, therefore should risk of car insurance for such driver would be higher. This also has a big impact on Autonomous vehicles, which rely on sensor fusion techniques, integrating inputs from cameras, lidar, radar, and ultrasonic sensors, to gain a comprehensive perception of the environment. Real-time mapping and precise localization based on high-precision maps and GPS data enable accurate positioning and navigation. Through object detection and tracking, autonomous vehicles identify and monitor vehicles, pedestrians, and obstacles in real-time, supporting safe decision-making. Sophisticated algorithms process the collected data, incorporating machine learning and artificial intelligence to interpret traffic regulations, predict behavior, and generate appropriate driving actions. Connectivity plays a vital role, facilitating data exchange with other vehicles, infrastructure, and cloud platforms, providing access to real-time traffic updates and enabling cooperative driving for improved traffic flow and safety. By leveraging these capabilities, autonomous driving systems strive to reduce human error, enhance road safety, and revolutionize transportation by offering efficient, reliable, and intelligent mobility solutions. All of the mentioned requires high-throughput communication. [5] Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication technologies are employed to meet these requirements. However, V2V communication can be unreliable due to factors like vehicle mobility, time-

varying traffic flow, and disparate communication environments across road sections. Conversely, V2I heavily relies on Roadside Units (RSUs), but the high costs associated with RSU deployment and limited coverage result in frequent disconnects and impact user experience.

This thesis investigates existing protocols employed in V2V and V2I communication systems, focusing on protocols such as DSRC, C-V2X, and IEEE 802.11p. Through a comprehensive analysis, the study evaluates the features, benefits, and limitations of these protocols, including their technical specifications, communication mechanisms, and use cases. The research aims to provide insights into the suitability of these protocols for different application scenarios and environments, facilitating the advancement of connected transportation.

To enhance V2X communication, particularly in scenarios with sparse RSU deployment, a cooperative communication scheme is proposed. The scheme utilizes V2V multi-hop forwarding to extend the V2I connection beyond RSU coverage, improving the average achievable throughput of the target vehicle and maintaining service continuity. The scheme involves vehicle A searching for the farthest vehicle B as the first relay, and as A moves out of RSU coverage, it maintains multiple forwarding routes by continuously seeking relays. An analytical model is developed to study data packet transmission under V2I and V2V cooperation, considering parameters such as RSU distance, willingness of vehicles to assist, cache size of the target vehicle, and data packet transmission. The model provides closed expressions for average throughput and average number of service interruptions, aligning well with simulation results and demonstrating the scheme's advantages in achieving high average throughput and business continuity.

Overall, this thesis contributes to the understanding of V2V and V2I communication protocols, proposes a cooperative communication scheme for V2V-assisted V2I communication, and presents an analytical model to analyze the scheme's performance. The research findings have implications for improving the efficiency and reliability of communication in connected transportation systems.

Analysis of Existing Protocols

The objective of this research is to review existing communication and networking protocols for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication systems. The purpose is to conduct a thorough analysis of these protocols, their technical specifications, and their suitability for different applications in V2V and V2I systems. The scope of the research includes a review and comparison of protocols such as DSRC, C-V2X, 802.11p, LTE-V2X, Zigbee, and Bluetooth. The aim is to identify the advantages and limitations of each protocol in terms of reliability, security, and scalability. Additionally, the research will analyze current trends and future prospects in V2V and V2I communication protocols.

V2V communication protocols

1. Dedicated Short-Range Communications (DSRC): DSRC is a communication protocol designed specifically for V2V and V2I applications. It operates in the 5.9 GHz band and can support communication ranges of up to 1 km. DSRC provides low-latency

communication and supports data rates of up to 27 Mbps, making it ideal for safety-critical applications such as collision avoidance.

2. Cellular-V2X (C-V2X): C-V2X is a cellular-based communication protocol that uses the LTE and 5G networks to enable V2V and V2I communication. C-V2X supports both direct communication between vehicles and communication with infrastructure elements. It offers high data rates and low latency, making it suitable for both safety-critical and non-safety-critical applications.
3. IEEE 802.11p: 802.11p is a Wi-Fi-based communication protocol designed specifically for V2V and V2I applications. It operates in the 5.9 GHz band and offers communication ranges of up to 1 km. 802.11p supports data rates of up to 27 Mbps and provides low-latency communication, making it suitable for safety-critical applications.
4. LTE-V2X: LTE-V2X is a cellular-based communication protocol that uses the LTE network to enable V2V and V2I communication. It supports both direct communication between vehicles and communication with infrastructure elements. LTE-V2X provides high data rates and low latency, making it suitable for both safety-critical and non-safety-critical applications.

V2I communication protocols

1. Dedicated Short-Range Communications (DSRC)
2. Cellular-V2X (C-V2X)
3. IEEE 802.11p
4. Cellular/Wi-Fi Hybrid Communication: This protocol combines cellular and Wi-Fi communication to enable V2I communication. It uses cellular networks to establish communication between the vehicle and the infrastructure element and Wi-Fi for data exchange. This approach offers the benefits of both cellular and Wi-Fi communication, including high data rates, low latency, and large coverage areas.

5. ZigBee: ZigBee is a low-power communication protocol that can be used for V2I communication. It operates in the 2.4 GHz and 868/915 MHz bands and can support communication ranges of up to 100 meters. ZigBee provides low-latency communication and can support data rates of up to 250 kbps.

Dedicated Short-Range Communications (DSRC)

[6-7] As mentioned previously DSRC (Dedicated Short-Range Communication) works by utilizing wireless communication technology to facilitate the exchange of information between vehicles and infrastructure in a short-range environment. It operates in the 5.9 GHz frequency band and employs the IEEE 802.11p protocol, which is an extension of the Wi-Fi standard. This is one of the most important protocols, since it is widely used

Communication Modes: DSRC supports both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication modes. In V2V communication, vehicles directly exchange safety messages with each other, enabling cooperative collision avoidance and awareness. In V2I communication, vehicles communicate with roadside infrastructure units, such as traffic lights, road signs, and toll booths, to receive relevant information and support advanced transportation services.

MAC Layer: The Medium Access Control (MAC) layer of DSRC manages the access to the shared communication channel. It uses a combination of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and Time Division Multiple Access (TDMA) to ensure fair access and reduce collisions between multiple DSRC devices attempting to transmit simultaneously. The MAC layer also provides mechanisms for priority-based access, allowing safety-critical messages to be given higher priority over non-safety-critical ones.

Safety Applications: DSRC is primarily designed to enhance road safety. It enables the implementation of safety applications such as Forward Collision Warning (FCW), Intersection Collision Warning (ICW), Emergency Electronic Brake Lights (EEBL), and Cooperative Adaptive Cruise Control (CACC). These applications leverage the exchanged safety messages to detect potential hazards, provide warnings to drivers, and enable cooperative maneuvers to avoid collisions.

Geolocation and Positioning: DSRC incorporates accurate geolocation and positioning capabilities to ensure the precise exchange of information. It uses Global Navigation Satellite Systems (GNSS) like GPS (Global Positioning System) to determine the vehicle's location and integrate it with the transmitted safety messages. This information is vital for various applications, including lane-level positioning, intersection management, and accurate trajectory estimation.

Data Transmission and Reliability: DSRC supports reliable data transmission by utilizing error correction codes and retransmission mechanisms. It employs Dedicated Short-Range Communications Message (DSRCM) for efficient and structured data exchange. DSRC messages are designed to be compact, enabling fast transmission and minimizing communication overhead. Regulatory Standards: DSRC operates under regulatory standards established by various organizations, including the Federal Communications Commission (FCC) in the United States and the European Telecommunications Standards Institute (ETSI) in Europe. These standards ensure interoperability, compatibility, and compliance with the allocated frequency spectrum and communication protocols. Coexistence with Other Technologies: DSRC technology is designed to coexist and complement other communication technologies used in Intelligent Transportation Systems (ITS). It can work alongside cellular-based technologies like LTE-V2X (Long-Term Evolution Vehicle-to-Everything) and C-V2X (Cellular Vehicle-to-Everything), allowing for hybrid communication systems that leverage the strengths of different technologies to enhance safety and efficiency.

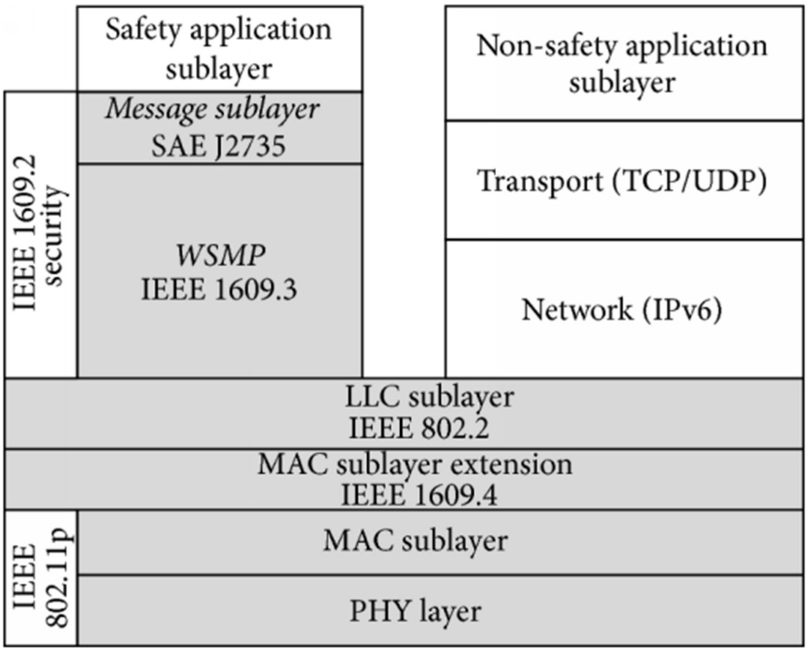

Figure 1. DSRC communication layered architecture

Cellular-V2X (C-V2X)

[8-9] Cellular Connectivity: Cellular-V2X (C-V2X) utilizes cellular networks, specifically the Long-Term Evolution (LTE) and 5G networks, to enable vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), and vehicle-to-network (V2N) communication. It leverages the existing cellular infrastructure, including base stations and core networks, to establish a reliable and extensive communication framework.

Direct and Network Communication: C-V2X supports two communication modes: direct communication and network communication. In direct communication, vehicles exchange information directly with each other without relying on cellular network infrastructure, enabling low-latency and high-reliability communication. In network communication, vehicles communicate through the cellular network, allowing for broader coverage and the exchange of information with infrastructure and other network-connected entities.

High Data Rates and Low Latency: C-V2X benefits from the high data rates and low latency offered by cellular networks. This enables the transmission of large amounts of data, such as high-definition maps, sensor information, and multimedia content, at faster speeds. The low latency ensures near real-time communication, making it suitable for safety-critical applications that require rapid response and decision-making.

Safety and Traffic Efficiency: C-V2X plays a crucial role in improving road safety and traffic efficiency. It enables the implementation of safety applications such as Forward Collision

Warning (FCW), Emergency Electronic Brake Lights (EEBL), Intersection Collision Warning (ICW), and Traffic Signal Preemption (TSP). These applications utilize the exchanged information to detect potential hazards, provide warnings to drivers, optimize traffic flow, and support cooperative driving maneuvers.

Cellular Network Capabilities: C-V2X benefits from the advanced features and capabilities of cellular networks. These include robust network coverage, seamless handover between base stations, network-wide synchronization, and quality of service (QoS) management. These capabilities ensure reliable and uninterrupted communication, even when vehicles are moving across different coverage areas or experiencing network congestion.

Spectrum Efficiency and Scalability: C-V2X optimizes spectrum utilization by operating in the licensed cellular bands, ensuring efficient use of available frequency resources. It can coexist with other cellular services and technologies, benefiting from the continuous advancements and spectrum management strategies employed by cellular network operators. This allows for scalability and future-proofing as cellular networks evolve to higher generations, such as 5G and beyond.

Standardization and Interoperability: C-V2X is standardized by various organizations, including the 3rd Generation Partnership Project (3GPP) and the European Telecommunications Standards Institute (ETSI). Standardization ensures interoperability between different vendors and devices, enabling seamless communication and compatibility across different networks and regions.

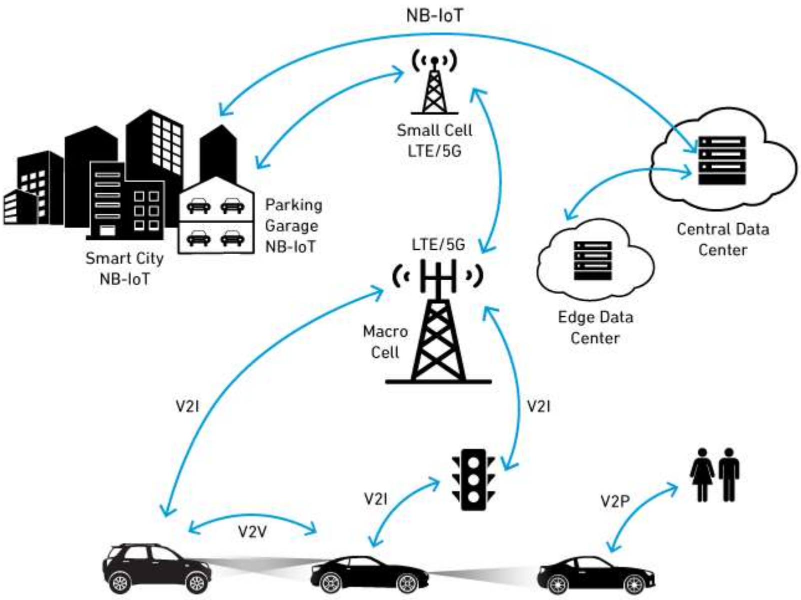

Figure 2. C-V2X communication

Benefits and drawbacks

Dedicated Short-Range Communications (DSRC) Benefits:

- DSRC offers low-latency communication, making it suitable for safety-critical applications.
- It provides a high level of reliability and security, which is critical for V2V and V2I communication.
- DSRC has been widely adopted in the automotive industry, and there is an existing infrastructure to support it.

Dedicated Short-Range Communications (DSRC) Drawbacks:

- DSRC operates in the 5.9 GHz band, which is a limited spectrum and can lead to congestion in urban areas.
- There is uncertainty about the future of DSRC, as some countries are exploring other V2V communication protocols such as C-V2X.

Cellular-V2X (C-V2X) Benefits:

- C-V2X provides high data rates and low latency, making it suitable for both safety-critical and non-safety-critical applications.
- It can operate in both direct communication and network communication modes, making it versatile for different use cases.
- C-V2X leverages the existing cellular infrastructure, which can reduce deployment costs.

Cellular-V2X (C-V2X) Drawbacks:

- The deployment of C-V2X requires the support of cellular network operators, which can be challenging in some regions.
- C-V2X operates in licensed cellular bands, which can lead to additional costs for automotive manufacturers.

IEEE 802.11p Benefits:

- 802.11p offers low-latency communication and high data rates, making it suitable for safety-critical applications.
- It has been widely adopted in the automotive industry, and there is an existing infrastructure to support it.
- 802.11p is a well-established standard, and there is a large ecosystem of vendors and suppliers.

IEEE 802.11p Drawbacks:

- 802.11p operates in the 5.9 GHz band, which is a limited spectrum and can lead to congestion in urban areas.

- The range of 802.11p is limited, and it may not be suitable for long-range communication.

LTE-V2X Benefits:

- LTE-V2X provides high data rates and low latency, making it suitable for both safety-critical and non-safety-critical applications.
- It leverages the existing LTE infrastructure, which can reduce deployment costs.
- LTE-V2X can operate in both direct communication and network communication modes.

LTE-V2X Drawbacks:

- The deployment of LTE-V2X requires the support of cellular network operators, which can be challenging in some regions.
- LTE-V2X operates in licensed cellular bands, which can lead to additional costs for automotive manufacturers.

Because of so many drawbacks and outdated design, we will not list other protocols than mentioned.

Chapter II

Analysis and Insight of Proposed Novel Design Protocols in Recent Years

The emergence of 5G technology offers opportunities to support smart transportation and enable seamless multi-dimensional high-speed information transmission between vehicles, roads, and the cloud. The combination of V2X and 5G can create a fusion network that ensures the continuity of smart transportation services and opens up new possibilities for applications like autonomous driving and remote driving. Conduction of a comprehensive review of existing protocol proposals, optimizations and researches will be needed again to confirm the challenges that are being faces and identify gaps that need to be addressed in further research of this thesis. Based on the insights gained from the literature review, trial to replicate a design of a novel networking and communication protocol will be done, along with proposal for improvements. Consider factors such as latency, throughput, reliability, scalability, security, and support for high-speed mobility. This will be done with utilization of simulation tools or modeling techniques to evaluate the performance of the designed protocol. Lastly, performance evaluation will be conducted by measurement and analysis of key performance metrics such as latency, throughput, packet loss, and scalability.

Exploring the State of Existing Protocols: Uncovering the Limitations and Opportunities in V2X Communication System

[10] Most of the used protocols are pretty old so far, in 1999, the United States Federal Communications Commission (FCC) allocated the 5,725 MHz to 5,875 MHz band of radio frequency for DSRC communication. The ITS Joint Program Office of the US Department of Transportation conducts research on DSRC and other wireless communication technologies and their uses in vehicle safety. When it comes to C-V2X, currently 3GPP has published 27 (3GPP TR22.885) basic application scenes defined by LTE-V2X and 25 (3GPP TR 22.886) enhanced application scenes. Among them, the basic application scenes mainly realize the auxiliary driving function, including active safety (collision warning, emergency braking, etc.) , traffic efficiency (speed guidance), information services, etc. The enhanced application scenes mainly realize the automatic driving function, including the four major functions of vehicle formation, advanced driving, extended sensors, and remote driving.

Although the basic application needs of car networking can support road safety, traffic efficiency, and information service applications, with the continuous evolution of automobile technology and communication technology, [11] V2X basic applications can no longer sense the surrounding environment and share status information. The needs of the network enhance the application. On the basis of the sharing status of transportation participants such as vehicles, it is possible to supplement the transmission of richer and more accurate information. Enhanced applications for vehicle networking have raised stricter communication requirements, such as extremely low communication delays, extremely high reliability, greater transmission rates, farther communication scopes, and support for higher mobile speeds.

[12] 5G has the characteristics of uprising bandwidth and downward low-time extension, which will play a supporting role in the construction of smart transportation and promote more abundant applications of the smart transportation industry, such as automatic driving and remote driving. The combination of V2X and 5G technology can achieve seamless coverage of the network, thereby achieving multi-dimensional high-speed information transmission between "cars-roads-cloud". At the same time, the V2X and 5G joint network can build a fusion network that covers the synergy with direct communication to ensure the continuity of the smart transportation business.

Key technologies of the vehicle networking system
Based on 5G with V2X Networking system architecture

In response to the different requirements for time delay, bandwidth and computing power for the two types of applications for the basic application of car networking and enhanced applications, this part of thesis analyzes proposed car networking system architectures based on the integration of 5G and V2X networks, as shown in Figure 3. [13] On the one hand, 5G can provide a low-time, high-bandwidth and high-reliability operating environment for V2X applications, while using the mobile edge computing technology (MEC) technology to localize, close, and distribute applications, services, and content deployment. On the other hand, V2X, which enables vehicle-to-car, vehicle-to-road communications, can support L1~L3 level automatic driving, mainly carrying basic traffic safety operations, such as traffic accident reminders, emergency brake warnings, and intersection collision warnings.

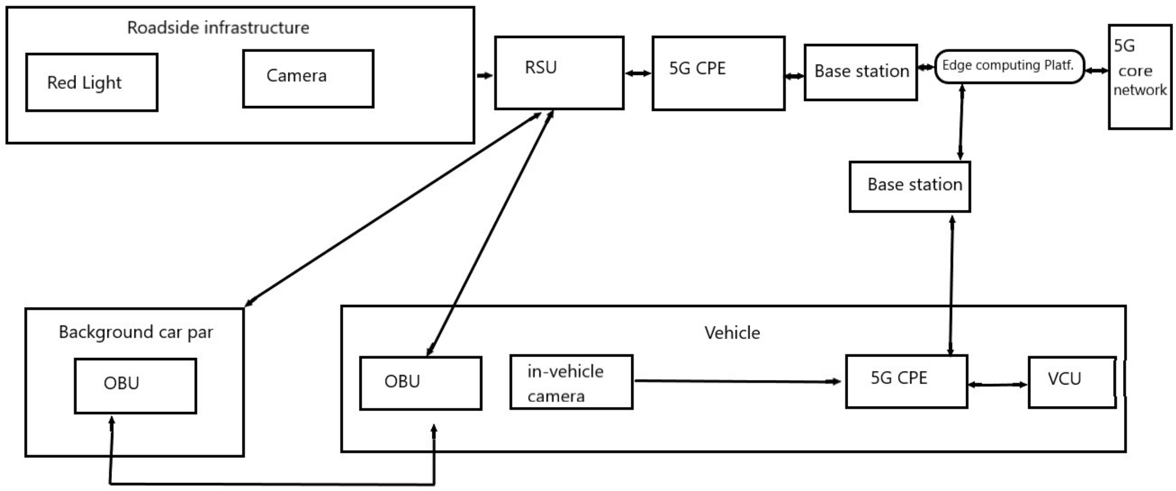

Figure 3. Car networking system architecture based on the integration of 5G and V2X networks

The vehicle networking application is based on the vehicle's information communication, and the communication connection between V2V, V2I and V2P is formed through the V2X network, including the measured vehicle, background vehicle, roadside unit (RSU) and roadside infrastructure, such as traffic green lights, camera head, etc., and through the 5G network, including MEC technology to achieve low delay treatment of high reliability applications.

[14,15] Overall car networking system architecture and main functional modules:

1. Target vehicles: equipped with sensors such as V2X communication terminals, vehicle cameras, laser radars, etc., for applications such as V2X communication and bicycle functions;
2. Background vehicles: V2X communication terminal installed for V2X communication with the measured vehicle;
3. Roadside infrastructure: including traffic lights, cameras, etc., used to provide road information sources for the measured vehicles, to achieve red light warning, vulnerable traffic participants and other scenes;
4. Roadside unit (RSU): used to receive information sent by roadside infrastructure or MEC, and perform V2X communication with surrounding vehicles ;
5. Edge computing platform: provide low-time extension business based on Uu mouth, and localize vehicle data or roadside data;
6. 5G core network: Using a service structure to provide V2X terminals with communication strategies and specific parameter configuration, contract information, and authentication information management services.

Typical application scene algorithm design

This section is based on the typical scenes of the 5G and V2X networks for algorithm development — intersection collision warning and remote driving research.

Intersection collision warning

Intersection collision warning: When the main car, HV (Host Vehicle), approaches an intersection with a risk of colliding with RV (Remote Vehicle), the ICW application will warn HV drivers.

[16] Figure 4 shows the flow chart of the collision warning algorithm at the intersection. First, HV scans the surroundings after receiving the BSM information of RV, identifying vehicles within a distance of less than 150m. During this time, the driver has sufficient time to take avoidance measures.

Afterwards, the algorithm determines if there is an intersection in the direction of the two cars, judging the collision point. If there is no intersection, there is no risk of collision. In that case, no further action is taken.

Next, the algorithm calculates the time and collision time difference (TTC) for both vehicles to reach the collision point based on their current driving state. Finally, if both TTC and TH (threshold) are less than the minimum safety time, Tsafe, a safety report is issued. Otherwise, a collision warning is issued.

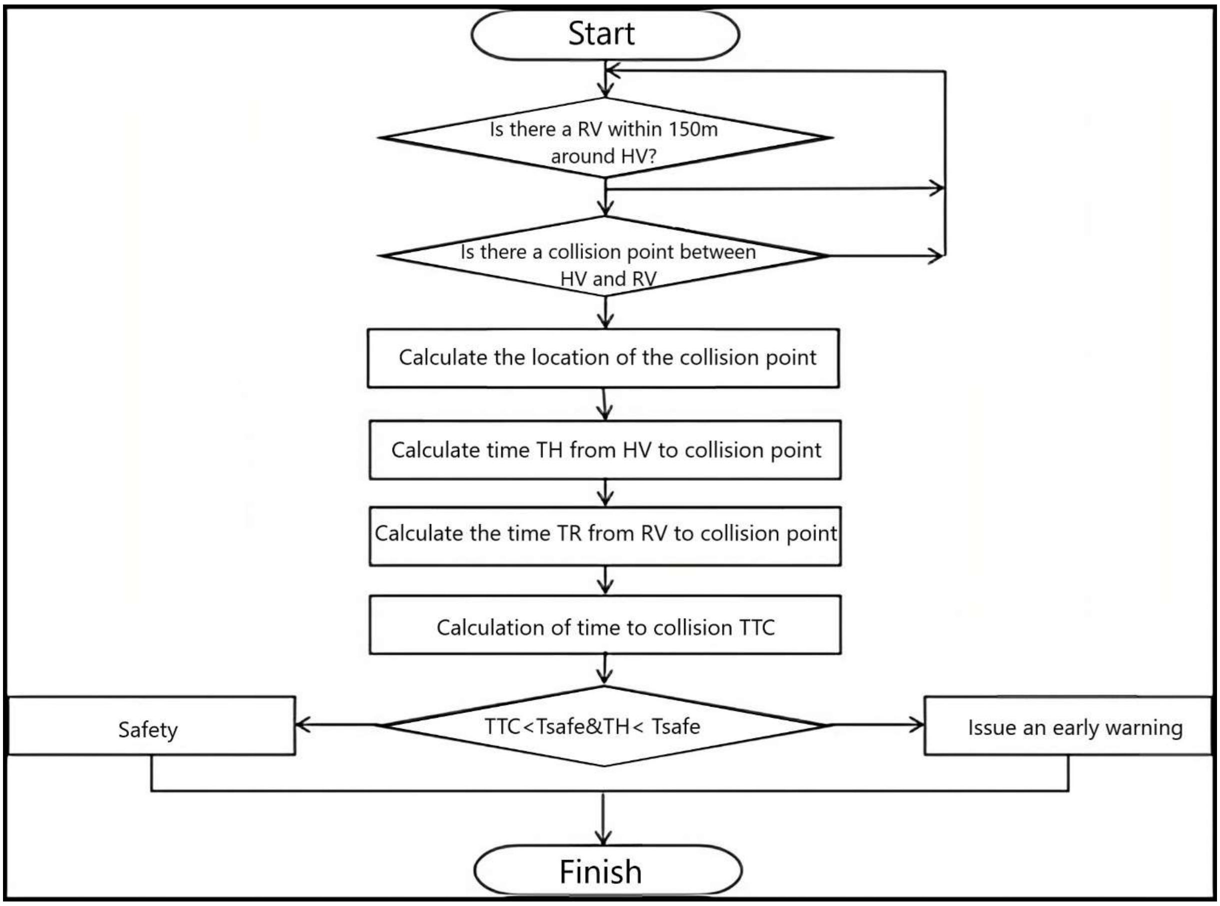

Figure 4. Flow chart of collision warning algorithm at intersections

At a certain moment, the HV and RV states are shown in Figure 5, at which the time location of HV is $H(X_H, Y_H)$, speed is V_H. Acceleration is A_H. The heading corner is θ_H. At the same time, the BSM message received by HV for RV broadcast obtained RV's location point is $R(X_R, Y_R)$, speed is V_R. Acceleration is A_R. The heading corner is θ_R and the distance between the two cars is Dis, and the location of the collision point is expected to be C (X_C, Y_C).

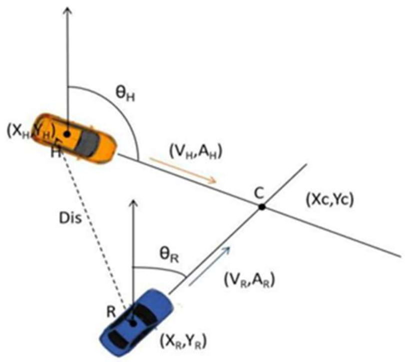

Figure 5. Schematic diagram of the relative position relationship between HV and RV

Eq. 1. Determination of the location of the collision point

[16] Available from Figure 5, the location of the collision point C (X_c, Y_c) is：

$$\begin{cases} X_C = \dfrac{\tan\theta_H \tan\ {}_R(Y_H - Y_R) + (\tan\theta_H X_R + \tan\theta_R X_H)}{\tan\theta_H + \tan\ {}_R} \\ Y_C = \dfrac{(X_H - X_R) + (\tan\theta_H Y_H + \tan\theta_R Y_R)}{\tan\theta_H + \tan\theta_R} \end{cases}$$

Equation II-1. Determination of the location of the collision point

Eq. 2. Calculate the time of the vehicle to the collision point

The distance from HV to the collision point can be obtained from the collision point position H_c and the distance from RV to the collision point Dis_{RC}, rule：

$$\begin{cases} T_H = \dfrac{\sqrt{V_H^2 + 2A_H \mathrm{Dis}_{HC}} - V_H}{A_H} & A_H > 0 \text{ 或 } D_{HC} < \dfrac{-V_H^2}{2A_H} \\ T_R = \dfrac{\sqrt{V_R^2 + 2A_R \mathrm{Dis}_{RC}} - V_R}{A_R} & A_R > 0 \text{ 或 } D_{RC} < \dfrac{-V_R^2}{2A_R} \end{cases}$$

Equation II-2. Calculate the time of the vehicle to the collision point

Eq. 3. Calculation of TTC

$$TTC = |T_H - T_R|$$

Equation II-3. Calculation TTC

TTC and T_HTo trigger the reference value of early warning, when TTC and T_H are also less than T_{safe} at the time, trigger the early warning signal.

Long-range driving

[17] Relying on 5G large bandwidth, low time delay, high reliable network characteristics can achieve full-directional monitoring and intelligent remote control of remote vehicles through remote smart driving platforms for users. 5G remote driving systems include three major parts: vehicle-end equipment, remote control platforms and MEC. The vehicle end equipment uploads driving status information and environmental videos to the MEC in real time, and receives remote control signal from the remote-control platform from the MEC; the remote-control platform real-time obtains vehicle information from the MEC, and transmits the remote-control driving signal to the vehicle through the MEC, thereby realize functions such as remote monitoring and remote driving. [18] The architecture chart is shown in Figure 6.

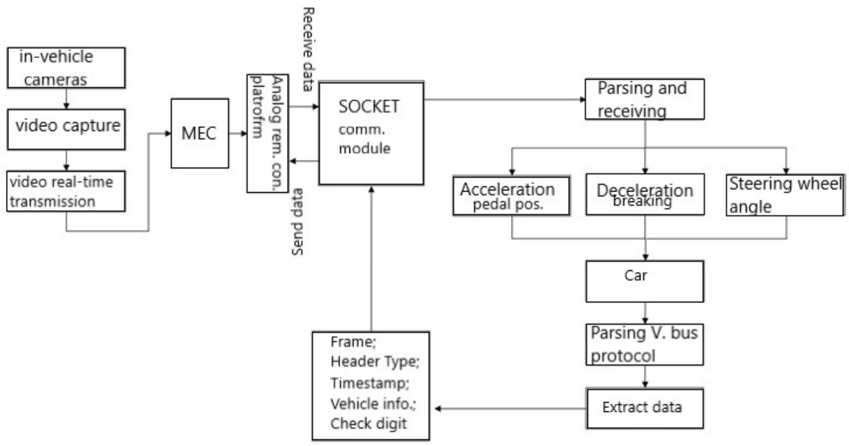

Figure 6. 5G remote driving system frame diagram

Introduction of main functions of various parts of the system ：

- Remote control platform

[19] Remote driving platforms mainly consist of: 5G CPE, off-site grid equipment, central server, driving simulator, and screen. The 5G CPE module and off-site grid equipment perform the same functions as the corresponding vehicle-end equipment and are responsible for sending and receiving information. The central server is primarily responsible for real-time analysis and processing of vehicle return status information and remote monitoring of vehicles. The screen is primarily responsible for displaying real-time video of the vehicle, the vehicle's driving

trajectory (map display), and the management interface of the displayed vehicle. It provides driving simulator drivers with the ability to control vehicle driving based on real-time environmental videos of the vehicle displayed on the screen and operate driving simulators to generate dynamic parameters such as vehicle speed and acceleration. The remote control driving center mainly includes equipment such as steering wheels, brakes, throttle, and other related connection and power supply equipment.

- Car end

[20] The vehicle configuration primarily consists of communication modules, video processing modules, and information extraction modules. The communication module is an embedded development board connected to the vehicle CAN bus, responsible for controlling the vehicle by converting control information into CAN messages. The video processing module connects the data captured by the wide-angle camera in front of the vehicle and the left and right post-view cameras by splicing them together, and then sends the video through the communication module.

The information extraction module understands the protocol information of the vehicle itself, extracts it, and sends it through the communication module. In addition, there is a wide-angle camera in front of the car to capture the scene ahead, and cameras in the left and right rearview mirrors to capture the scenes visible to the driver while driving.

- Network transmission

[21] The vehicle-side network provides a 5G network for the vehicle through 5G equipment, which is responsible for functions such as video streaming. On the simulator side, the 5G device enables a communication network for the simulator device. Within the communication system, the IP obtained by the computer represents a sub-network of a larger network.

Application Verification in Typical Scenarios

Let’s take China as an example here, because of the great investments in recent years in mobility field, with led to lots of experimentation and tests conducted on this field. [22] In the Xiqing Pilot Area, the feasibility of the architecture and functions of the Internet of Vehicles was verified in the two scenarios of intersection collision warning and remote driving. At present, Tianjin (Xiqing) National Vehicle Networking Pilot Zone is the first vehicle network pilot zone in Asia. Semi-enclosed venues and pilot demonstration areas were implemented and promoted. Among them, closed and semi-enclosed venues will focus on Internet of Vehicles technology verification and program testing, and first-come demonstrations focused on pilot applications and large-scale popularization tests of Internet of Vehicles.

Intersection collision warning

In the next part, we will take the experimental results from “C-V2X Vision in the Chinese Roadmap: Standardization, Field Tests, and Industrialization” book, written by Tao Cui, Lantao Li, Zhaoyu Zhang and Chen Sun. It was published on November 2022 and it underlines the concept of V2X 5G communication, portraying excellent data about current experimental results from China, which is world’s leader when it comes to the implementation of autonomous driving.

- Experimental equipment:

Two vehicles equipped with V2X communication equipment, RT-Range test system, AVAD3 alarm data acquisition system;

- Experimental environment:

As shown in Figure 5, there is an unsignalized intersection in the intelligent network demonstration park of China Automotive Industry Corporation.

Figure 7. Experimental environment for collision warning at intersections

- Experimental scene:

As shown in Table 1, HV and RV drive towards the intersection from different directions of the intersection, respectively, with different initial positions and driving speeds.

Table 1. Experimental scene

serial number	HV driving direction	RV driving direction	HV speed (km/h)	RV speed (km/h)	HV distance from intersection (m)	RV to intersection distance (m)
ICW-1	East	North	30	30	100	100
ICW-2	East	South	30	30	100	100

ICW-3	West	North	30	30	150	100
ICW-4	West	South	30	20	100	100
ICW-5	North	East	20	30	100	100
ICW-6	North	West	10	40	50	150
ICW-7	South	East	20	20	100	100
ICW-8	South	West	40	20	150	50

- Experimental data:

After verification, the test data is shown in Table 2. From the table, it can be found that the intersection collision warning can be triggered, and the trigger timing is within the threshold set by the algorithm.[23] At the same time, the realization of the intersection collision warning function depends on the vehicle networking system architecture based on the integration of 5G and V2X networks designed in this paper. According to the application communication requirements of the first phase of C-SAE, the maximum delay of the intersection collision warning application is 100ms. After testing, the communication delays are all lower than 100ms, meeting the standard international requirements.

Table 2. Test data

serial number	Whether to warn	Early warning time data (s)		Average communication delay (ms)	Test Results
		TTC	T_H		
ICW-1	Y	0.4	3.5	87.2	pass
ICW-2	Y	0.2	3.0	93.5	pass
ICW-3	N	-	-	82.5	pass

ICW-4	Y	3.6	1.3	72.1	pass
ICW-5	Y	2.9	3.8	86.9	pass
ICW-6	N	-	-	81.3	pass
ICW-7	Y	0.5	3.7	87.6	pass
ICW-8	Y	3.45	3.28	84.2	pass

Remote driving

Now let's take another article [24], along with article [25], where Huawei partnered with SAIC motor to field test remote driving. Here is the environment and results:

- Experimental equipment:

Driving simulator, two 5G CPEs, one passenger car of a certain brand, two industrial computers;

- Experimental environment:

As shown in Figure 6, the circular road in the intelligent network demonstration park of China Automobile Center is a two-way two-lane road, each lane is 3.5m wide, and the circular road is 550m long [25] states;

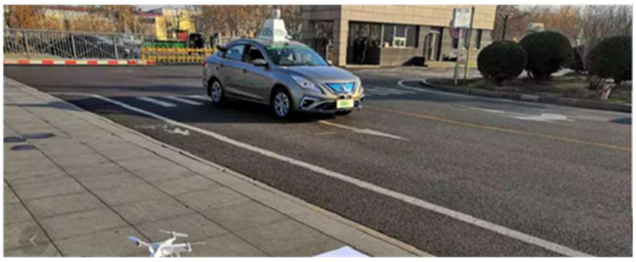

Figure 8. Remote driving test environment

- Experimental functions:

They selected several common driving conditions to test the remote-control system. Choose a flat and dry concrete road for remote control testing. The test efficiency and accuracy are greatly improved, the work intensity of the test is reduced, and the consistency of the test is ensured. The test content is shown in Table 3.

Table 3. Test items

Test items	Preconditions	expected outcome	Test communication delay
remote start	Analog kit connects normally normal network connection	The interface shows online remote driving Vehicle is in remote driving state	15ms
steering wheel	The interface shows online remote driving	The steering wheel hits to the left first and then backs up; the steering wheel hits to the right first, then backs up	21 ms
Drive forward		After driving forward for a while, parking	28ms
Drive back		After driving back for a while, parking	18ms
Turn left		The vehicle drove forward to the intersection, then turned left, and stopped after a while	10ms
Turn right		The vehicle drove forward to the intersection, then turned right, and stopped after a while	23ms
Exit remote driving	The vehicle is stationary and in remote driving	Vehicle display exits from remote driving	29ms
Take over remote driving		After the vehicle enters the takeover state, the vehicle is displayed to exit from the remote driving state, and the cockpit end is displayed online remote driving After evacuating from the takeover state, the cockpit end shows that there is no remote driving online, and the vehicle enters the remote driving state	14ms

Chapter II Conclusion

5G-V2X will become the mainstream form of car networking, highlighting the communication requirements and technological pathways for future car networking applications. Currently, the vehicle networking industry is in a crucial development stage, transitioning from forward-looking and scale-based demonstration applications to technical test certification. It is crucial to assess whether vehicle networking technology can effectively enhance road driving safety, improve traffic efficiency, and enhance the passenger travel experience, as it holds significant importance.

Chapter III

Cooperative V2V-Assisted V2I Communication for Enhanced Throughput and Service Continuity in Sparse RSU Scenarios

We have previously discussed that the cooperative communication between V2I and V2V, known as V2X, has gained significant attention for its ability to achieve continuous service. A crucial aspect in this context is the data forwarding mechanism. In this regard, [26] introduces a data forwarding mechanism that allows vehicles to collaborate with both the RSU and vehicles traveling in the opposite direction, resulting in an improved reachable throughput. However, due to the short contact duration between cars in opposing lanes, it becomes impossible to maintain long-term links between vehicles, thereby limiting the throughput of target vehicles. Furthermore, the mentioned method requires downloading data to the relay before sending it to the intended vehicle, which poses a disadvantage. Disconnection beyond the RSU coverage is particularly problematic for delay-limited services like online video streaming and file downloads. To mitigate vehicle disconnection time, [27] proposes a cooperative store-carry-forward (CSCF) mechanism. For example, in the case of car A, a car B is selected as the first relay in the same direction, enabling data transmission from the RSU to A through B. Through the cooperation between RSUs, another car C traveling in the opposite direction serves as the second relay for A. It is important to note that this method is most effective in scenarios with high RSU density. Additionally, on expressways, the VANET utilizes cars traveling in the same direction to form content dissemination clusters. For instance, in [28], cars moving along the same straight line are considered a chain cluster, with each member responsible for downloading different parts of a requested file from the RSU. After leaving the RSU coverage, the chain cluster members forward their downloaded portions to the target vehicle. The effectiveness of this method depends on the number of chain cluster members, making it less suitable for low vehicle density situations. Moreover, the competition and collisions that occur during data transmission from chain cluster members to the target vehicle can further reduce the target vehicle's throughput.

To complement the existing work, we have designed a cooperation scheme for V2V-assisted V2I communication in scenarios where RSU deployment is limited on highways. This scheme incorporates multi-hop forwarding in V2V communication to extend the V2I connection beyond the RSU coverage, ensuring improved average throughput and service continuity for the target vehicle. The main contributions of this research are summarized as follows:

Proposal of a cooperative communication mechanism between V2I and V2V that enhances the average achievable throughput of the target vehicle while maintaining service continuity. The scheme involves establishing a V2V multi-hop link connecting RSUs located outside the coverage area. To achieve this, the target vehicle A searches for the farthest vehicle B as the first relay and moves in the direction of A when it goes beyond the RSU coverage. The selected relay continues the strategy of finding subsequent relays, enabling continuous communication between the target vehicle and the RSU.

Introduction of an analytical model to study data packet transmission under V2I and V2V cooperation. Previous work, such as one analyzed before, focused on analyzing the relationship between throughput and the distance between adjacent RSUs without considering vehicle assistance. In this research part of thesis, we present a model that considers the interaction among three parameters: RSU distance, willingness of vehicles to assist, and the cache size of the target vehicle. The model provides closed expressions for average throughput and the average number of service interruptions, meeting the requirements for high throughput and service continuity in infotainment applications.

Validation of the analysis results through simulations, demonstrating the scheme's advantages in terms of high average throughput and reliable service continuity. Additionally, when compared to V2I networks, this scheme exhibits significant improvements.

In conclusion, this research explores methods to enhance V2X communication by focusing on data forwarding mechanisms. By addressing limitations in existing approaches and proposing a novel cooperation scheme, we aim to improve throughput and service continuity in V2X networks, thereby enhancing the overall performance and efficiency of vehicle communication systems.

System model as problem statement

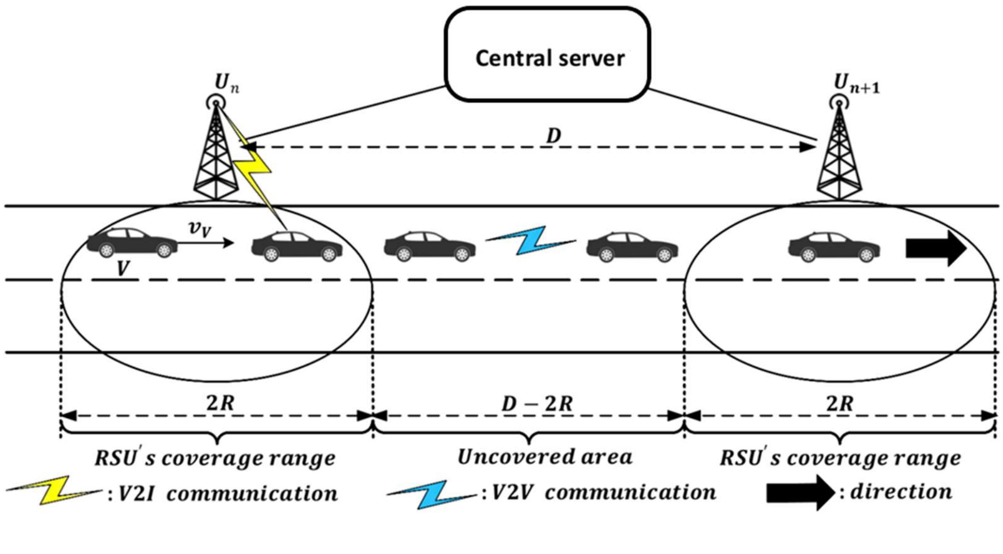

Figure 9. Highway communications scenario with uncovered areas.

As shown in the figure above (Fig. 7), we consider a one-dimensional network topology where RSUs are sparsely deployed along a two-way road. The distance between two consecutive RSUs is denoted as D, and the average coverage diameter of each RSU is 2R. We assume that the width of the road can be considered negligible compared to R. A central server is connected to all RSUs through limited backhaul links. The target vehicle V travels at a constant speed of $v_V(m/s)$. Additionally, other vehicles in the same direction as V are deployed with a density of $\rho(vehicles/m)$. These vehicles travel at different speeds and may change speeds during their journey. Considering only the mobility of each vehicle, we use kinematic equations that represent driving constraints to simulate the movement of the vehicle crowd. For example, the speed of vehicles traveling in the same direction is limited to $[v_min, v_max]$, and the inter-vehicle distance is restricted between the safe distance $s_Safe(m)$ and the following distance $s_Follow(s_Safe < s_Follow)$.

The research focuses on a specific vehicle V, which is within the coverage of the base station U_n. This vehicle requests and downloads files stored on the central server via U_n. When V moves from the coverage of U_n to the coverage of U_{n+1}, our goal is to minimize the disruption during the handover process. RSUs and vehicles employ transceivers through DSRC, enabling data propagation via V2I and V2V service channels. Each vehicle is equipped with a GPS device to share location information [22]. In this research, we utilize the wireless channel fading model [5] to calculate the physical layer transmission rate for V2I and V2V communication.

Problem statement

There are two main problems:

1. Because the RSU coverage is limited and the vehicle stays within the coverage of a single RSU for a short time, it is unlikely that the vehicle can completely download files with a large amount of data within the coverage of a single RSU;
2. When the RSU deployment is sparse, the car often moves out of the coverage area of the RSU, thus experiencing download interruption. The sparser the RSU deployment, the smaller the coverage area and the longer the interruption time.

This research aims to address these issues by developing a transmission scheme that maximizes the aggregate throughput of the target vehicle V while minimizing download interruptions when V is out of coverage. V2V assistance is utilized to extend the communication link back to the RSU from which V has just disconnected. The design challenges include:

- Which neighboring vehicle to choose for transfer?
- Developing a mechanism for data downloading and forwarding that aligns with the stated objectives.

Relay selection and data transfer mechanism

This research proposes a dynamic relay selection and data transmission mechanism to assist V's transmission outside the RSU coverage. We only select one of the vehicles in the same direction as V as a relay, which will be explained in detail below

Data downlink transmission via single-hop V2I communication

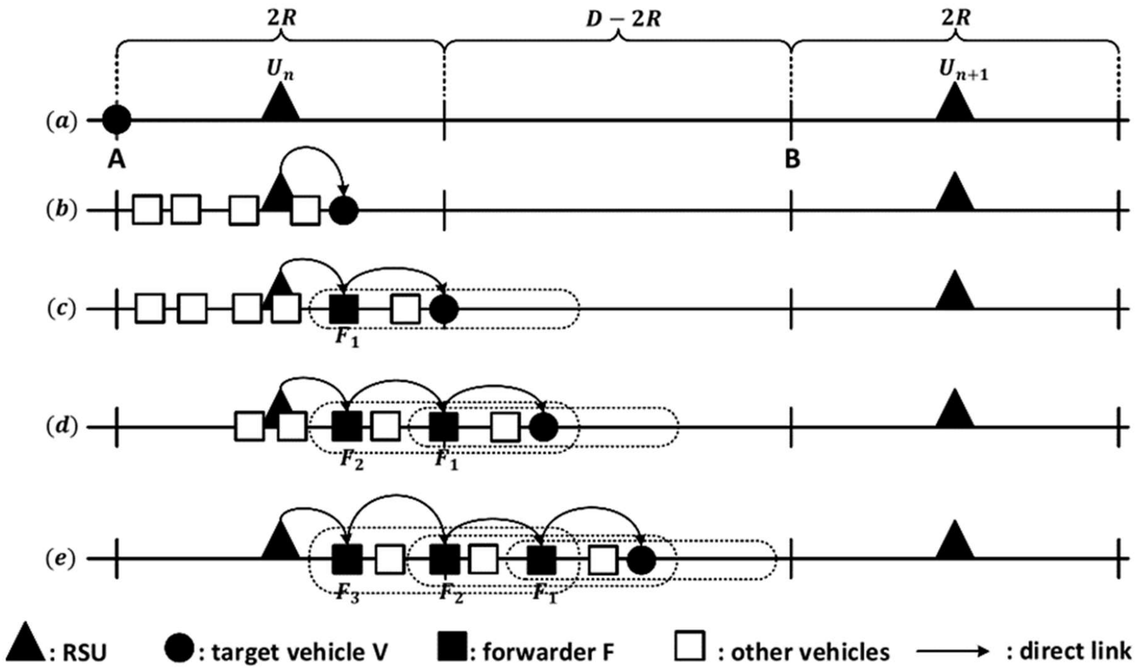

Figure 10. An illustration of the proposed scheme.

As shown in Figure 8, V is located in the coverage area of U_n, and sends a data download request to U_n, $U_n$forwards the request to the central processor, and the processor will divide the file requested by V into U_n before sending it to U_n. multiple small packets. Assuming that the transmission delay between the CPU and U_n is negligible, V fetches data from U_n until V reaches the edge of U_n's coverage.

Dynamic prequel relay selection

In order to continue downloading data outside the coverage, V first selects a relay among its neighbors through which U_n is connected to continue downloading. Vehicle V will broadcast the "request forwarding" information. If there is a vehicle willing to act as a relay within V's transmission range, it will reply its movement information (speed and position, etc.) to V through the "confirmation" message. Its neighbor list is pruned. V will select the farthest car in the list as a transponder, and mark this car as F1 in Figure 10.

C1: The relay is within the communication coverage of the requester, within the V2V direct connection range of the target vehicle.

C2: The relay is within the coverage of U_n, within the V2I direct connection range of U_n.

C3: The relay is in the same direction as the requester, maximizing the connection time with the target vehicle.

C4: The relay is the farthest distance from the requester, minimizing the number of relays.

In the data downlink transmission and relay via multiple V2I-V2V communications, relay F1 downloads the second data packet from U_n via V2I and forwards it to vehicle V via V2V, as shown in Figure 10(c). This process continues until F1 reaches the coverage boundary of U_n. At that point, F1 becomes a requester and selects the next relay F2 in the same process, as shown in Figure (10d). Each relay uses the established V2V link to forward received packets, even if it is beyond the coverage of U_n. Additionally, since relays can change their speed in reality, in order to overcome changes in the real-world topology and broken paths, if the current multiple paths are broken, V will use the same strategy to find new relays again. Figure 10(e) illustrates how this scheme ensures a continuous communication link between V and U_n, consisting of a single-hop V2I link from one RSU to the last transit and a multi-hop V2V link from the device to V.

The process of final relay selection and data transfer continues until V reaches the coverage of U_{n+1} or the file is completely downloaded by V. In the former case, the central server notifies RSU $U_{n+1}$to send the next data packet, and V connects to U_{n+1} through the V2I link to continue downloading the file.

Performance analysis

Data packet loss

There are two reasons for data packet loss. The first reason is the limited size of the buffer area in the vehicle. In this case, data packet loss occurs on the target vehicle V rather than the relay, as the relay forwards the data directly to V without storing it, although the buffer area on the relay is also limited. It is assumed that the packet loss rate caused by the wireless channel is negligible, as stated in [29].

The second reason is related to the cache area on the vehicle. If the data downloaded in the cache area does not meet the threshold requirement, the service will be interrupted. It is also assumed in this research that the service rate is greater than the arrival rate.

Since the intervals for data packets to reach V through single-hop/multi-hop connections are not static and the data in the buffer is processed at a dynamic rate, the G/G/1/b model is used to analyze the number of packets in the buffer and the packet loss at time t. Let x denote the number of packets of size b in the cache. Using the diffusion approximation and the steady-state solution, the steady-state pdf $p(x, t|p_{min})$ is obtained as:

$$p(x,\infty \mid p_{min}) = \begin{cases} -\dfrac{\lambda \cdot U_F}{\beta_L \cdot p_{min}}(1 - e^{\gamma \cdot x}), & 0 \le x \le p_{min} \\ -\dfrac{\lambda \cdot U_F}{\beta_L \cdot p_{min}}(e^{-\gamma \cdot p_{min}} - 1)e^{\gamma \cdot x}, & p_{min} \le x \le b-1 \\ -\dfrac{\mu_s \cdot O_F}{\beta_L}\left(e^{\gamma \cdot (x-b)} - 1\right), & b-1 \le x \le b, \end{cases}$$

Equation III-1 Steady-state PDF

Among them $\alpha_L = \sigma_a^2\lambda^3 + \sigma_s^2\mu_s^3$ and $\beta_L = \lambda - \mu_s(\lambda < \mu_s)$ for the drift and diffusion coefficient, $\gamma = \frac{2\beta_L}{\alpha_L}$ U_F is the probability of a packet smaller than the service threshold in the buffer, and O_F is the probability of packet loss caused by buffer overflow. λ(packets/s) is the average arrival interval, μ_s(packets/s) is the average service rate, p_{min} is the service threshold, σ_a^2 and σ_s^2 are the variance of the arrival interval and service duration, respectively. The mean $1/\mu_s$ and variance σ_s^2 are fixed and preset. Assuming that the arrival interval of vehicle V obeys an exponential distribution, its mean and variance are:

$$E[T_a] = \frac{1}{\lambda} Var[T_a] = {\sigma_a}^2 = \frac{1}{\lambda^2}$$

Equation III-2 Relationship between the arrival rate (λ) and the expected value, variance

Due to the influence of transmission speed and communication load, the average arrival interval can be obtained by [8]

$$\lambda = \frac{\xi_1 \cdot E[C]}{S_{pkt}}$$

Equation III-3 Packet arrival interval

Where E[C] is the average transmission rate of vehicle V;

S is the data packet size, and $\xi(0 << \xi 1)$ is the traffic load. Therefore, two probabilities O_F and U_F of the steady state can be obtained by [7]:

$$O_F = \left[-\frac{\mu_s^2 \cdot (1-e^{-\gamma}) \cdot p_{min}}{\lambda \cdot \beta_L \cdot (1-e^{-\gamma \cdot p_{min}}) \cdot e^{\gamma \cdot (b-1)}} + \frac{\lambda}{\beta_L}\right]^{-1} \triangleq f_1(\lambda),$$

$$U_F = \left[\frac{-\mu}{\beta_L} + \frac{\lambda^2 \cdot e^{\gamma(b-1)} \cdot (1-e^{-\gamma p_{min}})}{\beta_L \cdot p_{min} \cdot \mu_s \cdot (1-e^{-\gamma})}\right]^{-1} \triangleq f_2(\lambda).$$

Equation III-4 O_F and U_F probabilities

Average achievable throughput

This part involves a lot of formulas. The goal is to calculate the throughput of the vehicle in the process of going from one RSU to another RSU. The formula (4) of the previous session also participated in the calculation. The data flow diagram is as follows.

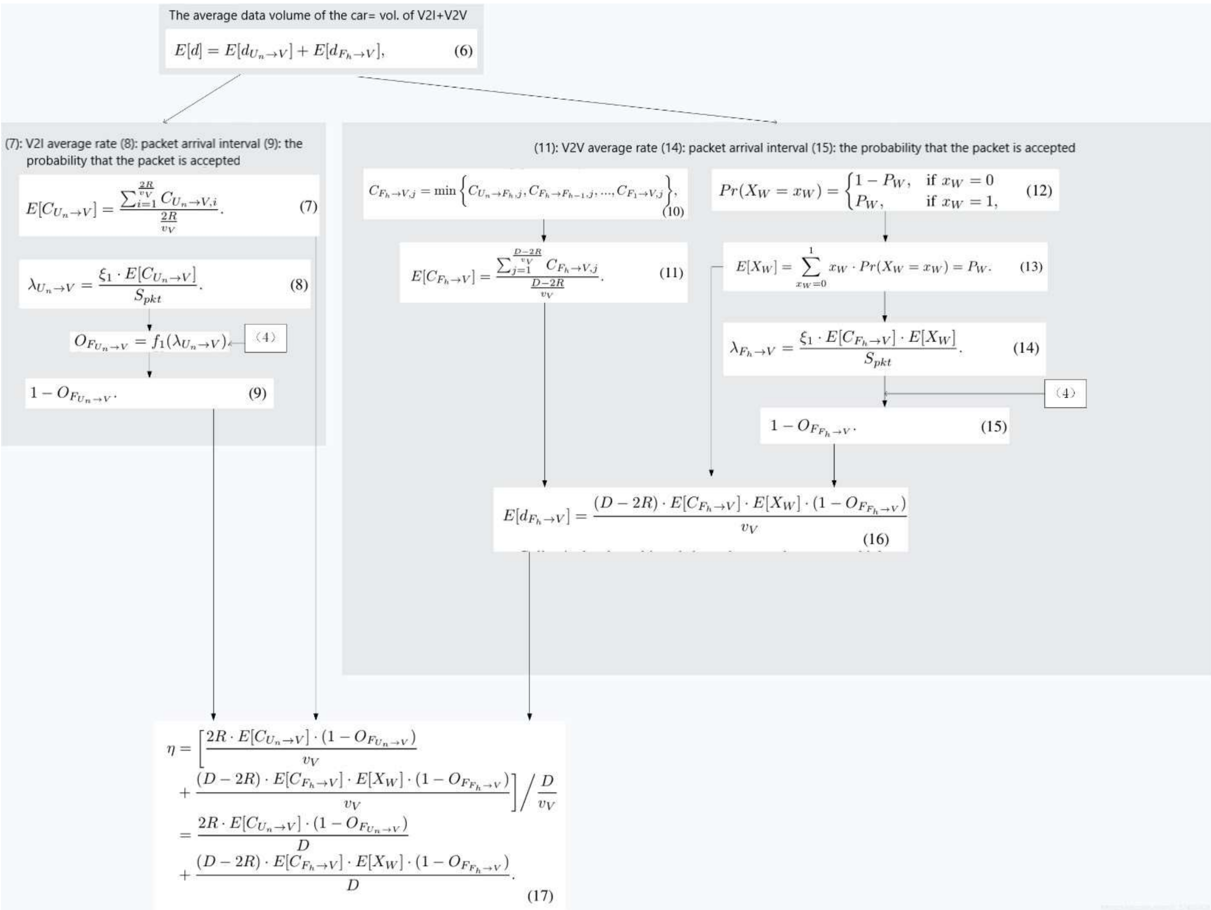

Figure 11. Data Flow Diagram

Considering an update-return process, the achievable throughput of a vehicle V between two adjacent RSUs is known by [1]:

$$\eta = \lim_{t \to \infty} \frac{\dot{d}(t)}{t} = \frac{E[d]}{E[T]}$$

Equation III-5 Efficiency

Where E[d] is the average expectation of the data packets received by V within E[T]. To illustrate the superiority of the method described in this research, D[d] consists of two parts: data transmitted via V2I and data transmitted via multi-hop V2V. Therefore, the total time for V to receive data is D/v_y. The average data volume of vehicle V in time E[T] is:

$$E[d] = E\left[d_{U_n \to V}\right] + E\left[d_{F_h \to V}\right]$$

Equation III-6 Average data volume of vehicle V

In (6), the first item is the average amount of data downloaded by V through V2I, and the second item is the average amount of data received by V from the relay F_h outside the coverage of RSU.

Within the coverage of U_n, the average transmission rate is $E\left[C_{U_n \to V}\right]$, where $C_{U_n \to V}$ represents the amount of data from RSU to V within one second. The time V is within U_n coverage is $2R/v_V$, so the average V2I transmission rate is:

$$E\left[C_{U_n \to V}\right] = \frac{\sum_{i=1}^{\frac{2R}{v}} C_{U_n \to V,i}}{\frac{2R}{v_V}}$$

Equation III-7 Average V2I transmission rate

According to (3), the average inter-arrival rate of packets of V in the coverage area is:

$$\lambda_{U_n \to V} = \frac{\xi_1 \cdot E\left[C_{U_n \to V}\right]}{S_{pkt}}$$

Equation III-8 Average inter-arrival rate of packets

When V receives data from U_n, packet loss occurs with probability $O_{F_{U_n \to V}} = f_1\left(\lambda_{U_n \to V}\right)$ at V. So, a packet sent by RSU to V arrives at V with the following probability:

$$1 - O_{F_{U_n \to V}}.$$

Equation III-9 Probability of packet arrival

When the car reaches outside the coverage area, the average transmission rate from the relay F_n to the car V is $E\left[C_{F_h \to V}\right]$, and $C_{F_h \to V}$ is the relay within one second. The instantaneous amount of data arriving at V, so:

$$C_{F_h \to V,j} = min\left\{C_{U_n \to F_h,j}, C_{F_h \to F_{h-1},j}, \ldots, C_{F_1 \to V,j}\right\}$$

Equation III-10 Size of the j-th packet transmitted from relay F_h to vehicle V.

Among them, F_h, ..., F_1 are U_n the transfers in multiple paths established from V, U_n to V, and $C_{U_n \to F_{h,j}}$ represents the transmission rate from U_n to transfer F_h in one second. At the same time $C_{F_h \to F_{h-1}}$represents the V2V transmission rate between two consecutive multi-hops.

When the car is outside the coverage of the RSU, the time for transferring data from U_n to V is: $(D - 2R)/v_V$, so the average forwarding rate of transferring $F_n$to vehicle V is:

$$E\left[C_{F_h \to V}\right] = \frac{\sum_{j=1}^{\frac{D-2R}{V}} C_{F_h \to V,j}}{\frac{D - 2R}{v_V}}$$

Equation III-11 Average forwarding rate

In the process of multi-hop forwarding, each vehicle participates in forwarding with a certain willingness, which is expressed as a random variable X_W, which is a Bernoulli distribution, expressed as:

$$P\,\mathrm{r}(X_W = x_W) = \begin{cases} 1 - P_W, & \text{if } x_W = 0 \\ P_W, & \text{if } x_W = 1 \end{cases}$$

Equation III-12 Vehicle willingness for multi hop forwarding

Among them, P_W (<=1) is the probability that the vehicle is willing to be a relay, and x_W=1 means that it is willing. So, the expectation of X_W is:

$$E[X_W] = \sum_{x_W=0}^{1} x_W \cdot P\,\mathrm{r}(X_W = x_W) = P_W.$$

Equation III-13 Expectation of X_W random variable

Different from the inter-arrival rate in the coverage area, the inter-arrival rate in the uncovered area is affected by the willingness of forwarding vehicles, so the average inter-arrival rate in the uncovered area is $\lambda_{F_h \to V}$:

$$\lambda_{F_h \to V} = \frac{\xi_1 \cdot E\left[C_{F_h \to V}\right] \cdot E[X_W]}{S_{pkt}}.$$

Equation III-14 Average inter-arrival rate in the uncovered area

Because the vehicle V buffer area is limited, the packet loss rate of receiving data from the relay is defined as $O_{F_{F_h} \to V} = f_1\left(\lambda_{F_h \to V}\right)$, so the probability that the data received from relay F_h reaches V successfully is:

$$1 - O_{F_{F_h} \to V}.$$

Equation III-15 Probability of the successful data arrival

Therefore, the average amount of data forwarded by relay F_n to V is:

$$E\left[d_{F_h \to V}\right] = \frac{(D - 2R) \cdot E\left[C_{F_h \to V}\right] \cdot E[X_W] \cdot \left(1 - O_{F_{F_h} \to V}\right)}{v_V}$$

Equation III-16 Average amount of data forwarded

Overall, the throughput of the target vehicle as it moves between two consecutive RSUs is:

$$\begin{aligned}
\eta &= \left[\frac{2R \cdot E\left[C_{U_n \to V}\right] \cdot \left(1 - O_{F_{U_n \to V}}\right)}{v_V} \right. \\
&\quad \left. + \frac{(D - 2R) \cdot E\left[C_{F_h \to V}\right] \cdot E[X_W] \cdot \left(1 - O_{F_{F_h} \to V}\right)}{v_V}\right] \Big/ \frac{D}{v_V} \\
&= \frac{2R \cdot E\left[C_{U_n \to V}\right] \cdot \left(1 - O_{F_{U_n \to V}}\right)}{D} \\
&\quad + \frac{(D - 2R) \cdot E\left[C_{F_h \to V}\right] \cdot E[X_W] \cdot \left(1 - O_{F_{F_h} \to V}\right)}{D}
\end{aligned}$$

Equation III-17 Overall throughput of the target vehicle

Average number of service interruptions

The number of service interruptions is an important indicator for evaluating service continuity. The lower the average number of interruptions, the longer the service interruption time. In order to ensure service continuity, it is expected that the service interruption duration should be as short as possible, so the average number of interruptions needs to be high. In this research, an interruption occurs when the target vehicle does not download enough data packets (i.e., does not reach the threshold to start data processing). Therefore, V needs to download enough packages to maintain the continuity of the service. Use $E[X_F]$ to represent the average number of interruptions within $E[T]$, and use $E[X_T]$ to represent the expectation of a single interruption duration.

The average number of interruptions within E[T] is formulated as: (probability of package not being large enough*time/duration of an interruption)

$$E[X_F] = \begin{cases} \dfrac{U_F \cdot E[T]}{E[X_T]}, & \text{if } 0 < \lambda < \mu_s \\ 1, & \text{if } \lambda = 0 \cdot \end{cases}$$

Equation III-18 Average number of interruptions

The average duration of a single interruption is given by [7] as:

$$E[X_T] = \begin{cases} \dfrac{p_{min}}{\lambda}, & \text{if } 0 < \lambda < \dfrac{p_{min}}{1s} < \mu_s \\ -\dfrac{p_{min}}{\lambda - \mu_s}, & \text{if } \dfrac{p_{min}}{1s} \leq \lambda < \mu_s. \end{cases}$$
$$\triangleq f_3(\lambda).$$

Equation III-19 average duration of a single interruption

(18) is a model, which will be refined as follows: the average number of interruptions is composed of the average number of interruptions inside and outside the RSU coverage, which are solved separately below. First, the number of interruptions the vehicle has in the coverage area is:

$$E\left[X_{F_{U_n \to V}}\right] = \frac{U_{F_{U_n \to V}} \cdot \dfrac{2R}{v_V}}{E\left[X_{T_{U_n \to V}}\right]}$$

Equation III-20 Interruptions the vehicle has in the coverage area

Where $U_{F_{U_n \to V}} = f_2(\lambda_{U_n \to V})$ is the probability that a packet arriving at V does not reach the service threshold and $E\left[X_{T_{U_n \to V}}\right] = f_3(\lambda_{U_n \to V})$ is the expected duration of a single outage within the coverage of U_n.

The average number of outages for a vehicle outside RSU coverage is:

$$\dot{H}[XF_{F_h} \to V] = \frac{UF_{F_h} \to V_V \cdot \dfrac{D-2R}{v_V}}{\widetilde{H}[XT_{F_h} \to V]}$$

Equation III-21 Average number of outages, out of RSU coverage

where $U_{F_{F_h \to V}} = f_2(\lambda_{F_h \to V})$ is the probability of not reaching the threshold and $E\left[X_{T_{F_h \to V}}\right] = f_3(\lambda_{F_h \to V})$ is the expected duration of a single outage out of coverage.

In summary, the total average number of interruptions can be obtained as:

$$E[X_F] = E\left[X_{F_{U_n \to V}}\right] + E\left[X_{F_{F_h \to V}}\right] \cdot$$

Equation III-22 Total average number of interruptions

Simulation Results

In the simulation, a section of a two-lane road is taken, and its relevant parameters are organized as follows

Table 4. Two-lane road simulation parameters

PARAMETER	VALUE
RSU spacing	2km~6km
The willingness of the vehicle to act as a relay $E[X_W]$	0.8
The speed of the target vehicle V is	17m/s (60km/h)
The speed of other vehicles traveling in the same direction	24m/s (86.4km/h)
Vehicle densityρ	0.003~0.005
IP packet size S_{pkt}	1400bytes
Buffer size	8000 (packets)
Threshold to start data processing	1000 (packets)

The average service time of the cache, denoted as $1/\mu_s$, and its variance σ^2 are both 0.667ms. For each scenario, the simulation was repeated 30 times, and the average results were plotted. The simulation results showed a difference of less than 4% compared to the analysis results. During the simulation, the total number of vehicles was (D+2R). The service rate followed a Poisson distribution, with one sample taken per simulation. The table below summarizes the channel parameters and vehicle movement model.

Table 5. Vehicle movement model and channel parameters

PARAMETER	VALUE
The transmjssion power *(Pt)*	23 dBm
The gain of a transmitter antenna (Gt)	1
The gain of a receiver antenna *(Gr)*	1
The heigh L of a tran smiLLer anlenna (ht)	l m
The height of a receiver antenna *(hr)*	l m
The lost factor *(L)*	1
The fading parameter (μ)	1
The thermal noise power *(Nr)*	-96 dBm
The reference distance *(d0)*	150 m
The wavelength of a wave at 5.9 GHz (Av)	3x10-0. 0508 m/5 . 9xl0g
The path loss exponents (a1 , a 2)	2.1, 3.8
RSU 's average transmjssion range *(R)*	600 m
Vehicle's average transmjssion range *(r)*	300 m
Target vehicle's velocity (*vv)*	17 m/s (= 60 km/h)
[vmin ; Vmax]	(17; 33] m/s ((60; 120] km/h)
[ssaf e ; SF ol l ow]	(60; 200] m
Traffic load (1)	1

Result analysis

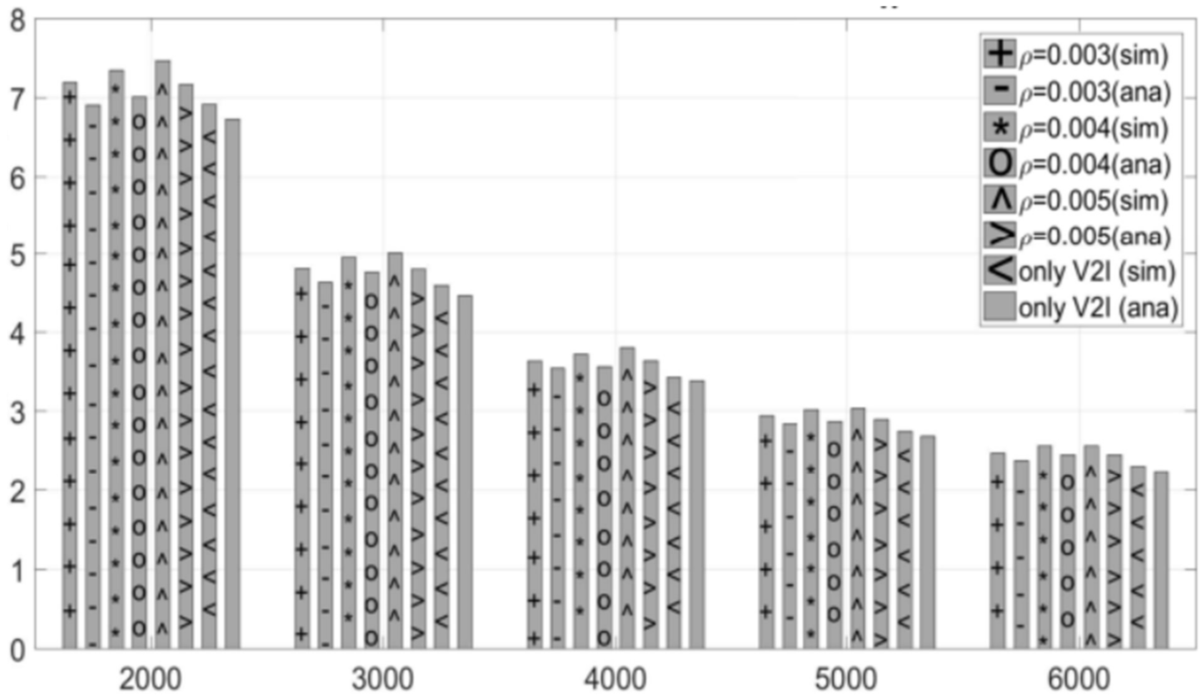

Figure 12. Relationship between the average throughput and the RSU spacing

(Fig. 12.) describes the relationship between the average throughput and the RSU spacing, and the average throughput decreases when the RSU spacing increases. When the RSU density increases, the average throughput reaches a peak value, but because of the mobility of vehicles, the impact of RSU density on the average throughput is not obvious. Compared with the pure V2I network, the average throughput is improved by 7.5%~9.4%.

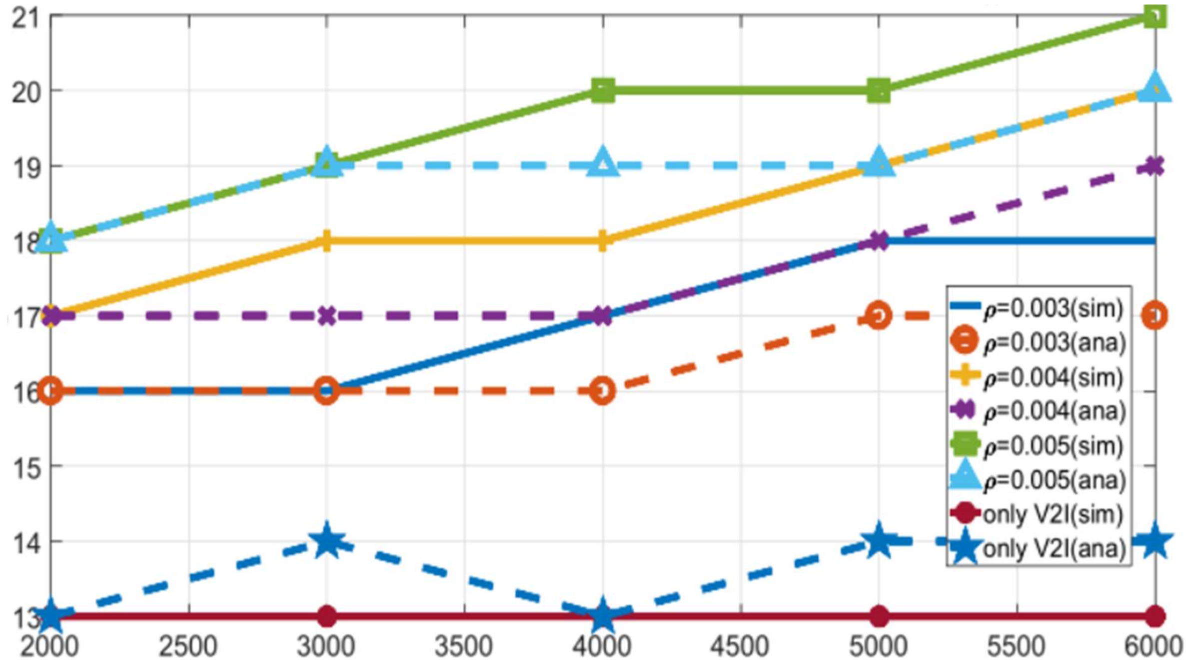

Figure 13. Relationship between the average number of outages and the RSU spacing

(Fig. 13.) describes the relationship between the average number of outages and the RSU spacing, from which it can be seen that the scheme described in this paper is in terms of service continuity due to the pure V2I network, and the increase in the number of service outages indicates that the buffer is still being processed intermittently in some way data. But the pure V2I architecture keeps the same number of service interruptions because it cannot transmit the data to the target vehicle.

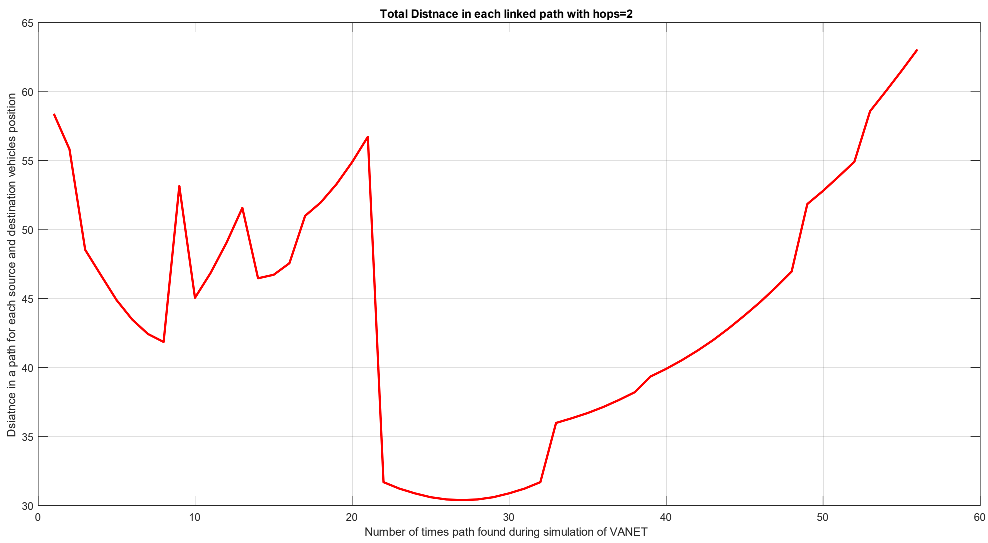

Figure 14. Total distance in each linked path with hops

By executing the simulation (Fig. 14), we can obtain valuable insights into the behavior of the urban city vehicle network. The simulation takes into account the specific characteristics of the chosen source and destination nodes, as well as the overall network structure. It provides a comprehensive analysis of the distance between nodes, which can help determine the efficiency of communication within the network.

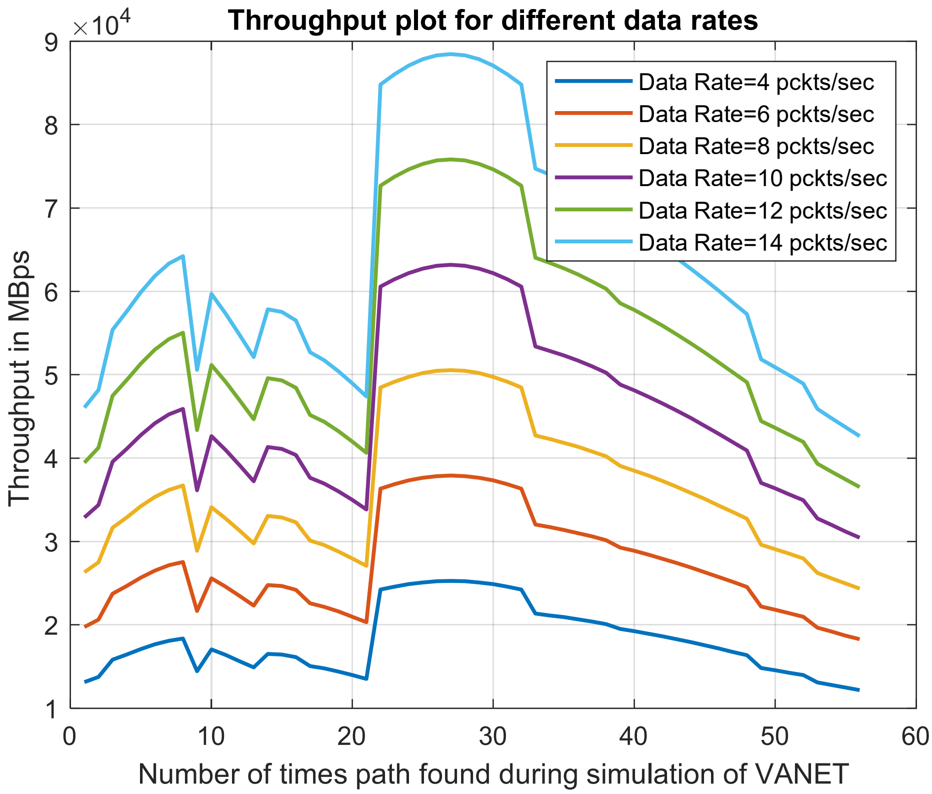

Figure 15. Throughput plot for different data rates

In V2V and V2I communication, the throughput exhibits fluctuations as the number of paths and data rates vary. When considering data rates of 4, 6, 8, 10, 12, and 14 packets per second, the throughput reaches its peak at approximately 20 to 30 paths. This means that within this range of paths, the system achieves the highest data transmission rate. However, it is important to note that the actual throughput may still be influenced by other factors such as network congestion, channel conditions, and resource allocation. By understanding these fluctuations and identifying the optimal number of paths, network designers and researchers can make informed decisions to maximize throughput and improve the efficiency of V2V and V2I communication administrators can optimize data rates and path selection to achieve the desired throughput and ensure efficient data transmission in their network infrastructure.

Conclusion

Through this thesis, a comprehensive understanding of V2V and V2I communication protocols has been developed, and the potential of the proposed V2I and V2V cooperative communication mechanism has been explored. The aim was to identify their strengths and weaknesses, while also examining the recent advancements in novel design protocols. The findings emphasize the importance of leveraging 5G technology and V2X communication to enhance traffic safety and enable advanced autonomous driving features. Among these protocols, the focus was primarily on V2X, which leverages the capabilities of 5G technology. The use of 5G offers significant advantages in terms of low latency, high bandwidth, and high reliability, creating an optimal operating environment for V2X applications. Furthermore, V2X communication enables effective V2V and V2I interactions, facilitating Level 1 to Level 3 autonomous driving functionalities. It plays a crucial role in supporting essential traffic safety operations, including traffic accident reminders, emergency brake warnings, and intersection collision warnings.

Additionally, main focus of this study, was to investigated the performance of a transmission scheme aimed at maximizing the aggregate throughput of a target vehicle while minimizing download interruptions when the vehicle is outside the coverage of a roadside unit (RSU). The scheme utilizes V2V assistance, extending the communication link back to the just-disconnected RSU. Through simulations and analysis, we have examined various factors such as relay selection, data downloading, and forwarding mechanisms.

Our results demonstrate that the selection criteria for relays, based on factors such as coverage, direction, and distance, play a crucial role in achieving the desired goals of maximizing throughput and minimizing interruptions. Additionally, the proposed scheme effectively handles the dynamic nature of real-world topologies by adapting and finding new relays when multiple paths are broken.

Furthermore, our analysis reveals that the average transmission rate from the relay to the target vehicle outside the coverage area is an important factor in determining the amount of data arriving at the vehicle. This highlights the significance of relay performance in maintaining continuous communication and minimizing interruptions during the download process.

Comparing our scheme with a pure V2I network, we have observed substantial improvements in average throughput, ranging from 7.5% to 9.4%. Moreover, the proposed scheme outperforms the pure V2I architecture in terms of service continuity, as evidenced by a lower number of service outages. This indicates the successful utilization of V2V assistance and relay-based communication to mitigate interruptions and provide a more reliable and efficient transmission experience for the target vehicle.

In conclusion, our research contributes to the advancement of vehicular communication systems by addressing the challenges of limited RSU coverage and intermittent connectivity. The proposed transmission scheme offers promising results in terms of enhanced throughput and improved service continuity, thereby paving the way for more efficient and reliable communication in vehicular networks.

References

1. Automotive News. (2023). Automotive News. Retrieved from https://www.autonews.com/
2. MotorTrend. (2023). Automobile. Retrieved from https://www.motortrend.com/automobilemag/
3. ACEA. (2021). Auto industry actively sharing vehicle data, putting consumer choice, safety and security first. Retrieved from https://www.acea.auto/message-dg/auto-industry-actively-sharing-vehicle-data-puttingconsumer-choice-safety-and-security-first/
4. Brown, D., Flickenschild, M., Mazzi, C., Gasparotti, A., Panagiotidou, Z., Dingemanse, J., & Bratzel, S. (2021). The Future of the EU Automotive Sector. Retrieved from http://www.europarl.europa.eu/supporting-analyses
5. Beylot, A., & Labiod, H. (2013). Vehicular Networks: Models and Algorithms. John Wiley & Sons.
6. Lorenz, P. (2003). Networking - ICN 2001: First International Conference on Networking Colmar, France, July 9-13, 2001 Proceedings, Part I. Springer.
7. Emmelmann, M., Bochow, B., & Kellum, C. (2010). Vehicular Networking: Automotive Applications and Beyond. John Wiley & Sons.
8. Chen, S., Hu, J., Zhao, L., Zhao, R., Fang, J., Shi, Y., & Xu, H. (2023). Cellular Vehicle-to-Everything (C-V2X). Springer Nature.
9. Committee, C. T. (2022). LTE Vehicle-to-Everything (LTE-V2X) Deployment Profiles and Radio Parameters for Single Radio Channel Multi-Service Coexistence.
10. Federal Communications Commission. (n.d.). FCC Allocates Spectrum 5.9 GHz Range for Intelligent Transportation Systems Uses. Retrieved from https://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html
11. Pathan, A. K. (2022). Towards a Wireless Connected World: Achievements and New Technologies. Springer Nature.
12. Rommer, S., Hedman, P., Olsson, M., Frid, L., Sultana, S., & Mulligan, C. (2019). 5G Core Networks: Powering Digitalization. Academic Press.
13. Fallgren, M., Dillinger, M., Mahmoodi, T., & Svensson, T. (2021). Cellular V2X for Connected Automated Driving. John Wiley & Sons.
14. (2011). Design of In-vehicle Networked Control System Architectures Through the Use of New Design to Cost and Weight Processes: Innovation Report.
15. Sommer, C., & Dressler, F. (2015). Vehicular Networking. Cambridge University Press.
16. Santa, J., Toledo-Moreo, R., Zamora-Izquierdo, M. A., Úbeda, B., & Gómez-Skarmeta, A. F. (2010). An analysis of communication and navigation issues in collision avoidance support systems. Transportation Research Part C-emerging Technologies, 18(3), 351–366. https://doi.org/10.1016/j.trc.2009.05.007
17. Nowak, T., Sepczuk, M., Kotulski, Z., Niewolski, W., Artych, R., Bocianiak, K., ... Wary, J. (2021). Verticals in 5G MEC-Use Cases and Security Challenges. IEEE Access, 9, 87251–87298. https://doi.org/10.1109/access.2021.3088374
18. ETSI. (2022). MEC PoC#13 on ‘MEC infotainment for smart roads and city hot spots’. Retrieved from

https://mecwiki.etsi.org/index.php?title=PoC_13_MEC_infotainment_for_smart_roads_and_city_hot_spots
19. Shi, W., & Liu, L. (2021). Computing Systems for Autonomous Driving. Springer Nature.
20. Guo, S. (2011). The application of CAN-bus technology in the vehicle. In 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC) (pp. 755-758). doi: 10.1109/MEC.2011.6025574
21. Liu, S., & Ma, X. (2022). Advanced Hybrid Information Processing: 5th EAI International Conference, ADHIP 2021, Virtual Event, October 22-24, 2021, Proceedings, Part II. Springer Nature.
22. 5GAA. (2022). C-V2X Pilot and Demonstration Areas in China. 5GAA Technical Report. Retrieved from https://5gaa.org/
23. Chen, S., Hu, J., Shi, Y., Peng, Y., Fang, J., Zhao, R., & Li, H. (2017). Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G. IEEE Communications Standards Magazine, 1(2), 70–76. https://doi.org/10.1109/mcomstd.2017.1700015
24. Kutila, M., Kauvo, K., Aalto, P., Martinez, V. D., Niemi, M., & Zheng, Y. (2020). 5G Network Performance Experiments for Automated Car Functions. 2020 IEEE 3rd 5G World Forum (5GWF). https://doi.org/10.1109/5gwf49715.2020.9221295
25. Huawei. (n.d.). Huawei Demonstrates 5G-based Remote Driving with China Mobile and SAIC Motor. Retrieved from https://www.huawei.com/en/news/2017/6/5g-based-remote-driving
26. (2015). A Social-based Data Forwarding Mechanism for V2V communication in VANETs. IEEE Conference Publication | IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/abstract/document/7498007
27. Wang, Y., Liu, Y., Zhang, J., Ye, H., & Tan, Z. (2016). Cooperative Store-Carry-Forward Scheme for Intermittently Connected Vehicular Networks. IEEE Transactions on Vehicular Technology, 65(1), 43-56. https://doi.org/10.1109/tvt.2016.2536059
28. Yue, S., & Zhu, Q. (2021). A Mobility Prediction-Based Relay Cluster Strategy for Content Delivery in Urban Vehicular Networks. Applied Sciences, 11(5), 2157. https://doi.org/10.3390/app11052157
29. Jameel, F., Javed, M. S., & Ngo, D. T. (2020). Performance Analysis of Cooperative V2V and V2I Communications Under Correlated Fading. IEEE Transactions on Intelligent Transportation Systems, 21(8), 3476–3484. https://doi.org/10.1109/tits.2019.2929825

Appendix 1: MATLAB calculation codes for Average Number of Interruptions in proposed cooperative communication mechanism between V2I and V2V

```
function E_X_F = calculateE_X_F(U_F_U_n_V, R, v_V, E_X_T_U_n_V, U_F_F_h_V, D,
H_T_X_F_h_V)
    % Calculate average number of interruptions for vehicle V
    E_X_F_U_n_V = (U_F_U_n_V * 2 * R) / v_V / E_X_T_U_n_V;
    E_X_F_F_h_V = (U_F_F_h_V * (D - 2 * R)) / v_V / H_T_X_F_h_V;
    E_X_F = E_X_F_U_n_V + E_X_F_F_h_V;
end

function U_F_U_n_V = calculateU_F_U_n_V(lambda_U_n_V, p_min, beta_L, gamma,
b)
    % Calculate probability that a packet arriving at V does not reach the
service threshold in the coverage area
    U_F_U_n_V = (-(lambda_U_n_V * U_F) / (beta_L * p_min) * (1 - exp(gamma))
* exp(gamma * (b - 1))) ^ (-1);
end

function U_F_F_h_V = calculateU_F_F_h_V(lambda_F_h_V, p_min, beta_L, gamma)
    % Calculate probability that a packet arriving at V does not reach the
service threshold outside the coverage area
    U_F_F_h_V = (-mu / beta_L + (lambda_F_h_V ^ 2 * exp(gamma * (b - 1)) * (1
- exp(-gamma * p_min))) / (beta_L * p_min * mu_s * (1 - exp(-gamma)))) ^ (-
1);
end

function E_X_T_U_n_V = calculateE_X_T_U_n_V(lambda_U_n_V, p_min, mu_s)
    % Calculate expected duration of a single outage within the coverage area
    if 0 < lambda_U_n_V && lambda_U_n_V < p_min / mu_s
        E_X_T_U_n_V = p_min / lambda_U_n_V;
    elseif p_min / mu_s <= lambda_U_n_V && lambda_U_n_V < mu_s
        E_X_T_U_n_V = -p_min / (lambda_U_n_V - mu_s);
    end
end

function H_T_X_F_h_V = calculateH_T_X_F_h_V(lambda_F_h_V, p_min, mu_s)
    % Calculate expected duration of a single outage outside the coverage
area
    if 0 < lambda_F_h_V && lambda_F_h_V < p_min / mu_s
        H_T_X_F_h_V = p_min / lambda_F_h_V;
    elseif p_min / mu_s <= lambda_F_h_V && lambda_F_h_V < mu_s
        H_T_X_F_h_V = -p_min / (lambda_F_h_V - mu_s);
    end
end
```

Appendix 2: MATLAB codes for Wireless Communication System Simulation

```
% Parameters
Pt = 23; % Transmission power (dBm)
Gt = 1; % Transmitter antenna gain
Gr = 1; % Receiver antenna gain
ht = 1; % Height of transmitter antenna (m)
hr = 1; % Height of receiver antenna (m)
L = 1; % Loss factor
```

```
mu = 1; % Fading parameter
Nr = -96; % Thermal noise power (dBm)
d0 = 150; % Reference distance (m)
lambda = 3e10^-0.0508/(5.9e9); % Wavelength of a wave at 5.9 GHz (m)
a1 = 2.1; % Path loss exponent for RSUs
a2 = 3.3; % Path loss exponent for vehicles
vmin = 17; % Minimum velocity (m/s)
vmax = 33; % Maximum velocity (m/s)
safedist = 60; % Safe distance (m)
trafficLoad = 1; % Traffic load

% Distance between consecutive RSUs (m)
dRSU = 2000:1000:6000;

% Initialize arrays for throughput
throughputRSU = zeros(size(dRSU));
throughputVehicle = zeros(size(dRSU));

% Calculate Throughput for different distances
for i = 1:length(dRSU)
    R = dRSU(i); % RSU's average transmission range (m)
    r = R/2; % Vehicle's average transmission range (m)

    % Calculate Path Loss
    d = linspace(1, R, 1000); % Distance from RSU to target vehicle
    d = d(:); % Ensure d is a column vector
    pathLossRSU = L * (lambda ./ (4 * pi * d)).^a1; % Path loss for RSU
    pathLossVehicle = L * (lambda ./ (4 * pi * d)).^a2; % Path loss for
vehicle

    % Calculate Signal-to-Noise Ratio (SNR)
    SNRRSU = Pt + Gt - pathLossRSU - 10*log10(Nr); % SNR for RSU
    SNRVehicle = Pt + Gt - pathLossVehicle - 10*log10(Nr); % SNR for vehicle

    % Calculate Throughput
    throughputRSU(i) = mean(trafficLoad * log2(1 + SNRRSU)); % Throughput for
RSU (bps)
    throughputVehicle(i) = mean(trafficLoad * log2(1 + SNRVehicle)); %
Throughput for vehicle (bps)
end

% Convert Throughput to Mbps
throughputRSU = throughputRSU / 1e6; % Throughput for RSU (Mbps)
throughputVehicle = throughputVehicle / 1e6; % Throughput for vehicle (Mbps)

% Plot Results
figure;
bar(dRSU, [throughputRSU' throughputVehicle'], 'grouped');
xlabel('Distance between Consecutive RSUs (m)');
ylabel('Throughput (Mbps)');
legend('RSU', 'Vehicle');
title('Throughput vs Distance between Consecutive RSUs');
```

Appendix 3. MATALB codes for standard V2V communication

```
axis([0 1000 0 1000]);            % To set the window size to 1000
```

```
DELAY=0.5;
DELAY1=0.45;
[x1,y1] = ginput(2);                % To take input from user for line 1
line(x1,y1,'color','black');        % draw the line (Road 1)
hold on ;
[x2,y2] = ginput(2);                % To take input from user for line 2
line(x2,y2,'color','black');        % draw the line (Road 2)
rnd_speed1=randi([10,20],1,1);
dist_first_line=sqrt((x1(2)-x1(1))^2+(y1(2)-y1(1))^2);      % Distance between
2 points is calculated
speed_new1=round(dist_first_line/rnd_speed1);              % This is the new
speed for 1st vehicle
rnd_speed2=randi([10,20],1,1);
dist_second_line=sqrt((x2(2)-x2(1))^2+(y2(2)-y2(1))^2);     % Distance
between 2 points is calculated
speed_new2=round(dist_first_line/rnd_speed1);              % This is the new
speed for 2nd vehicle
point1=linspace(x1(1),x1(2),speed_new1);  % Random speed is given by taking
new speed number of points in a linspace
point2=linspace(y1(1),y1(2),speed_new1);
point3=linspace(x2(1),x2(2),speed_new2);
point4=linspace(y2(1),y2(2),speed_new2);
first_vehicle=plot(point1,point2,'s','MarkerFaceColor','red');   % Plot first
vehicle on road 1
second_vehicle=plot(point3,point4,'o','MarkerFaceColor','blue'); % Plot
second vehicle on road 2
title('V2V connectivity simulation');                          % Title is
given to the figure
for i=1:1000

    for k = 1:speed_new1                   % for all the values in linspace
         first_vehicle.XData = point1(k);  %first vehicle's x co-ordinate
         first_vehicle.YData = point2(k);  %first vehicle's y co-ordinate
         second_vehicle.XData = point3(k); %second vehicle's x co-ordinate
         second_vehicle.YData = point4(k); %second vehicle's y co-ordinate
         plot(point1(k),point2(k),point3(k),point4(k));
         vehicle_dist=[point1(k),point2(k);point3(k),point4(k)]; % Calculate
the Euclidian distance between two vehicle's positions
         distance1= pdist(vehicle_dist,'euclidean');
          if distance1<=100                % If the distance is within 200 m
            line1=plot([point1(k),point3(k)],[point2(k),point4(k)],'--
','color','green');  % Show connectivity between two vehicles
            pause(0.3);                  % Delay of 0.45
            set(line1,'Visible','off');      % Visibility property of line is
set to 'off'
          end
          pause(DELAY1);                     % Delay of 0.5

    end
        set(first_vehicle,'Visible','off'); % Visibility property of first
vehicle's position is set to 'off'
        set(second_vehicle,'Visible','off'); %Visibility property of second
vehicle's position is set to 'off'

end

hold off;
```

Appendix 4. MATLAB code for standard V2I communication

```
axis([0 1000 0 1000]);                          % To set the window size to
1000
DELAY=0.5;
DELAY1=0.45;
[a,b]=ginput(5);                                % Take input from user
line(a,b,'color','black'); hold on;             % Draw a line connecting
the points taken as input from user
rnd_speed=randi([10,20],1,1);                   % Random speed between 10-
20 m/s is given to the vehicle
dist_whole=sqrt((a(2)-a(1))^2+(b(2)-b(1))^2);   % Distance between 2 points
is calculated
speed_new=round(dist_whole/rnd_speed);
x=linspace(a(1),a(2),speed_new);                % Random speed is given by
taking speed_new points in linspace
y=linspace(b(1),b(2),speed_new);
u=linspace(a(2),a(3),speed_new);
v=linspace(b(2),b(3),speed_new);
w=linspace(a(3),a(4),speed_new);
z=linspace(b(3),b(4),speed_new);
q=linspace(a(4),a(5),speed_new);
r=linspace(b(4),b(5),speed_new);
arr1=[x u w q];                                 % Array of all 'x' co-
ordinates of 4 lines
arr2=[y v z r];                                 % Array of all 'y' co-
ordinates of 4 lines
[xcord1,ycord1]=ginput(1);                      % Select a point for RSU 1
text(xcord1,ycord1,'RSU 1 ','HorizontalAlignment','right');
[xcord2,ycord2]=ginput(1);                      % Select a point for RSU 2
text(xcord2,ycord2,'RSU 2 ','HorizontalAlignment','right');
plot(xcord1,ycord1,'o','MarkerFaceColor','blue'); % RSU 1 plotted on the
figure window
plot(xcord2,ycord2,'o','MarkerFaceColor','blue'); % RSU 2 plotted on the
figure window
p=plot(x,y,'square','MarkerFaceColor','green','MarkerSize',5);   % Vehicle
moving along the road
title('V2I connectivity');                      % Title is given to the
figure
for i=1:1000
    for j=1:length(arr1)-1

           p.XData = arr1(j);              % X co-ordinate for that
particular road segment
           p.YData = arr2(j);              % Y co-ordinate for that
particular road segment
           first_dist=[xcord1,ycord1;arr1(j),arr2(j)]; % Take euclidian
distance between vehicle's position on the road and RSU 1
           distance1=pdist(first_dist,'euclidean');
           second_dist=[xcord2,ycord2;arr1(j),arr2(j)]; % Take euclidian
distance between vehicle's position on the road and RSU 2
           distance2=pdist(second_dist,'euclidean');
```

```
            pause(DELAY1);
            if distance1<=100                            % if distance between
RSU 1 and vehicle's position < 100 m
                    line1=plot([xcord1,arr1(j)],[ycord1,arr2(j)],'--
','color','green'); % Show connectivity to RSU 1

range1=plot([arr1(j),arr1(j+1)],[arr2(j),arr2(j+1)],'color','green');
% plot the points for given line space. Hence moving vehicle effect
                    pause(0.3);
                    set(line1,'Visible','off');                 % Visibility
property is set to ='off'
                    set(range1,'Visible','on');                 % Visibility
property is set to ='off'

            elseif distance2<=100
                   line2=plot([xcord2,arr1(j)],[ycord2,arr2(j)],'--
','color','green');

range1=plot([arr1(j),arr1(j+1)],[arr2(j),arr2(j+1)],'color','green');
% plot the points for given line space. Hence moving vehicle effect
                   pause(0.3);
                   set(line2,'Visible','off');                  % Visibility
property is set to ='off'
                   set(range1,'Visible','on');                  % Visibility
property is set to ='off'

            else

first=plot([arr1(j),arr1(j+1)],[arr2(j),arr2(j+1)],'color','red');
                       set(first,'Visible','on');
            end


    end
end

hold off;
```

Appendix 5. MATLAB code for Vehicular mobility (Open Source)

```
entry=[1500 2500 4000]; %ycoordinates of entry ramps
exit=[1550 2550 4050];  %ycoordinates of exit ramps
trafficdensity=[100 150 200 250 300 350 400 450 500];
v2v=[0 0 0 0 0 0 0 0 0]; %list having the connectivity entries for different
traffice densities
target=41; %car numbered 41 is taken as target
neighbours=[];
duration=[0 0 0 0 0 0 0 0 0];
avg_cont_neighbours=[0 0 0 0 0 0 0 0 0];
for cars=100:50:500
    connectivity=[0 0 0 0 0];
    times=[];
    same_neighbours=[];
```

```
    ycoord1=transpose(randperm(5000,cars)); %the array has ycoordinate
positions of cars on lane 1
    ycoord2=transpose(randperm(5000,cars)); %the array has ycoordinate
positions of cars on lane 2
    ycoord3=transpose(randperm(5000,cars)); %the array has ycoordinate
positions of cars on lane 3
    ycoord4=transpose(randperm(5000,cars)); %the array has ycoordinate
positions of cars on lane 4
    ycoord1=sort(ycoord1,1); %sorting to get cars in increasing order of
positions
    ycoord2=sort(ycoord2,1);
    ycoord3=sort(ycoord3,1);
    ycoord4=sort(ycoord4,1);
    positions=zeros(5000,4);
    speed=zeros(5000,4);
    oldcoord=0;
    %placing cars numbered uniquely in increasing in a 5000x4 matrix named
    %positions and their respective speeds in a mtrix called speed (wherever
no car is present, positions and speed value is zero)
    for j=1:cars
        index=ycoord1(j);
        positions(index,1)=j;   %place car at the position pointed by the
ycoord1 array
        speed(index,1)=(31-22).*rand(1,1) + 22; %speed values (50 & 70 miles
/hour) converted to m/s
    end
    %for cars in lane 2:
    maximum=max(max(positions));    %for car labels to be in continuity
    for j=1:cars
        index=ycoord2(j);
        positions(index,2)=maximum+j;   %for car labels to be in continuity
        speed(index,2)=(31-22).*rand(1,1) + 22; %speed values (50 & 70 miles
/hour) converted to m/s
    end
    %for cars in lane 3:
    maximum=max(max(positions));
    for j=1:cars
        index=ycoord3(j);
        positions(index,3)=maximum+j;
        speed(index,3)=(31-22).*rand(1,1) + 22; %speed values (50 & 70 miles
/hour) converted to m/s
    end
    maximum=max(max(positions));
    %for cars in lane 4:
    for j=1:cars
        index=ycoord4(j);
        positions(index,4)=maximum+j;
        speed(index,4)=(31-22).*rand(1,1) + 22; %speed values (50 & 70 miles
/hour) converted to m/s
    end
    for i=1:5 %five iterations for computing average 5
        timeflag=0;
        neighbours=[];
        for j=1:22 %10 minutes in all=600 secs, value of j is the seconds
value
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % For Lane 1:
```

```
            %safety application messages will be exchanged here (every 1 sec)
            rand1=rand(1,1);
            rand2=rand(1,1);
            if rand1>0.833
                entryramp=randi(3,[1,1]);   %to chose an entry ramp randomly
                if ~(ismember(entry(entryramp),ycoord1)) %checking if no
vehicle is already present there
                    %add a vehicle here
                    ycoord1(length(ycoord1)+1)=entry(entryramp);
                    positions(entry(entryramp),1)=max(max(positions))+1;
%label the new car
                    speed(entry(entryramp),1)=(31-22).*rand(1,1) + 22;
%assign it a random speed
                end
            end
            if rand2>0.833
                exitramp=randi(3,[1,1]);    %to chose an exit ramp randomly
                if ismember(exit(exitramp),ycoord1) %remove if there is car
at that exit ramp that can exit
                    if positions(exit(exitramp),1)~=target %target car should
not be removed
                        ycoord1=ycoord1(find(ycoord1~=exit(exitramp))); %car
location removed from ycoord array
                        positions(exit(exitramp),1)=0;  %car removed
                        speed(exit(exitramp),1)=0;  %corresponding speed
entry cleared
                    end
                end
            end
            ycoord1=sort(ycoord1,1); %sort the vehicles in increasing order
of positions
            for k=1:1    %1 loop assumed to be of 100ms for required
granularity
                for l=1:length(ycoord1) % for each car in the lane
                    if(l<=length(ycoord1))  %to avoid indexing error when no.
of cars reduce during lane changing, in short recalculating the length
                        acc=-5+10*rand(1,1);    %random acceleration value
within -5 and +5 m/s^2
                        r=speed(ycoord1(l),1);
                        newspeed=abs(r+acc);
                        if(l~=length(ycoord1))
                            if(ycoord1(l+1)-ycoord1(l)<=10)
                                flag=1;
                                %look if lane change is poosible :-
                                m=ycoord1(l);
                                for g=m-10:m+10 %checking if any car is
present in the parallel lane near that position
                                    if  ismember(g,ycoord2)
                                        flag=0; %if so,lane change is not
possible in lane2
                                    end
                                end
                                if flag==1  %if there is space for changing
lanes
                                    %change lane here:
                                    ycoord2(length(ycoord2)+1)=m;
```

```
                                    positions(m,2)=positions(m,1);  %move the
car to next lane
                                    positions(m,1)=0;   %clear the car label
from current lane
                                    speed(m,2)=speed(m,1);  %move the
corresponding speed entry to adjacent lane
                                    speed(m,1)=0;   %clear the speed entry
from current lane
                                    ycoord1(l)=0;   %remove the position
entry from ycoord1 array
                                    ycoord2=sort(ycoord2,1);    %sort the
array pointing to next lane to get them back in increasing order after adding
car
                                elseif flag==0 %if lane change was not
possible
                                    dist=ycoord1(l+1)-ycoord1(l);
                                    factor=(-
0.75+sqrt(0.5625+0.02804*dist))/0.01408;

newspeed=min(factor,speed(ycoord1(l+1),1)); %reduce the speed as given in the
project desciption
                                end
                            end
                        end
                        if(ycoord1(l)~=0)
                            oldcoord=ycoord1(l);
                            ycoord1(l)=round(ycoord1(l)+newspeed*0.01);
%position update per 100ms
                            ycoord1(l)=mod(ycoord1(l),5000)+1;  %rollback
position
                            if(positions(ycoord1(l),1)==0)

positions(ycoord1(l),1)=positions(oldcoord,1);  %move the car forward
                                positions(oldcoord,1)=0;    %clear previous
position
                                speed(ycoord1(l),1)=newspeed;   %similarly
with speed matrix
                                speed(oldcoord,1)=0;
                            end
                        end
                    end
                end
                ycoord1=ycoord1(find(ycoord1~=0));  %remove the cleared
entries from ycoord1 array, car is no more present there as it changed lanes
            end

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
            %similar computation for each lanes(refer to detailed comments
above). Lane 2:
            %entry and exit here
            rand1=rand(1,1);
            rand2=rand(1,1);
            if rand1>0.833
                entryramp=randi(3,[1,1]);
                if ~(ismember(entry(entryramp),ycoord2)) %checking if no
vehicle is already present there
                    %add a vehicle here
```

```
                ycoord2(length(ycoord2)+1)=entry(entryramp);
                positions(entry(entryramp),2)=max(max(positions))+1;
                speed(entry(entryramp),2)=(31-22).*rand(1,1) + 22;
            end
        end
        if rand2>0.833
            exitramp=randi(3,[1,1]);
            if ismember(exit(exitramp),ycoord2)
                if positions(exit(exitramp),2)~=target
                    ycoord2=ycoord2(find(ycoord2~=exit(exitramp)));
                    positions(exit(exitramp),2)=0;
                    speed(exit(exitramp),2)=0;
                end
            end
        end
        ycoord2=sort(ycoord2,1);
        for k=1:1
            for l=1:length(ycoord2) % for each car in a lane
                if(l<=length(ycoord2))
                    acc=-5+10*rand(1,1);
                    newspeed=abs(speed(ycoord2(l),2)+acc);
                    if(l~=length(ycoord2))
                        if(ycoord2(l+1)-ycoord2(l)<=10)
                            flag=1;
                            %look if lane change is poosible in lane 1:-
                            m=ycoord2(l);
                            for g=m-10:m+10
                                if  ismember(g,ycoord1)
                                    flag=0; %lane change not possible in
lane 1

                                end
                            end
                            if flag==1
                                %change lane here to lane 1:
                                ycoord1(length(ycoord1)+1)=m;
                                positions(m,1)=positions(m,2);
                                positions(m,2)=0;
                                speed(m,1)=speed(m,2);
                                speed(m,2)=0;
                                ycoord2(l)=0;
                                ycoord1=sort(ycoord1,1);
                            elseif flag==0
                                %look if lane change is poosible in lane
3:-

                                for g=m-10:m+10
                                    if  ismember(g,ycoord3)
                                        flag=0; %lane change not possible
in lane 3

                                    end
                                end
                                if flag==1
                                    %change lane here to lane 3:
                                    ycoord3(length(ycoord3)+1)=m;
                                    positions(m,3)=positions(m,2);
                                    positions(m,2)=0;
                                    speed(m,3)=speed(m,2);
                                    speed(m,2)=0;
```

```
                                        ycoord2(l)=0;
                                        ycoord3=sort(ycoord3,1);
                                    elseif flag==0 %if lane change was not
possible at all
                                        dist=ycoord2(l+1)-ycoord2(l);
                                        factor=(-
0.75+sqrt(0.5625+0.02804*dist))/0.01408;

newspeed=min(factor,speed(ycoord2(l+1),2));
                                    end
                                end
                            end
                        end
                        if(ycoord2(l)~=0)
                            oldcoord=ycoord2(l);
                            ycoord2(l)=round(ycoord2(l)+newspeed*0.01);
%position update per 100ms
                            ycoord2(l)=mod(ycoord2(l),5000)+1;%rollback
position
                            if(positions(ycoord2(l),2)==0)

positions(ycoord2(l),2)=positions(oldcoord,2);
                                positions(oldcoord,2)=0;
                                speed(ycoord2(l),2)=newspeed;
                                speed(oldcoord,2)=0;
                            end
                        end
                    end
                end
                ycoord2=ycoord2(find(ycoord2~=0));
            end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %Lane 3:
            %entry and exit here
            rand1=rand(1,1);
            rand2=rand(1,1);
            if rand1>0.833
                entryramp=randi(3,[1,1]);
                if ~(ismember(entry(entryramp),ycoord3)) %checking if no
vehicle is already present there
                    %add a vehicle here
                    ycoord3(length(ycoord3)+1)=entry(entryramp);
                    positions(entry(entryramp),3)=max(max(positions));
                    speed(entry(entryramp),3)=(31-22).*rand(1,1) + 22;
                end
            end
            if rand2>0.833
                exitramp=randi(3,[1,1]);
                if ismember(exit(exitramp),ycoord3)
                    if positions(exit(exitramp),3)~=target
                        ycoord3=ycoord3(find(ycoord3~=exit(exitramp)));
                        positions(exit(exitramp),3)=0;
                        speed(exit(exitramp),3)=0;
                    end
                end
            end
            ycoord3=sort(ycoord3,1);
```

```
            for k=1:1
                for l=1:length(ycoord3) % for each car in the lane
                    if(l<=length(ycoord3))
                        acc=-5+10*rand(1,1);
                        newspeed=abs(speed(ycoord3(l),3)+acc);
                        if(l~=length(ycoord3))
                            if(ycoord3(l+1)-ycoord3(l)<=10)
                                flag=1;
                                %look if lane change is poosible in lane 2:-
                                m=ycoord3(l);
                                for g=m-10:m+10
                                    if  ismember(g,ycoord2)
                                        flag=0; %lane change not possible in
lane 2
                                    end
                                end
                                if flag==1
                                    %change lane here to lane 2:
                                    ycoord2(length(ycoord2)+1)=m;
                                    positions(m,2)=positions(m,3);
                                    positions(m,3)=0;
                                    speed(m,2)=speed(m,3);
                                    speed(m,3)=0;
                                    ycoord3(l)=0;
                                    ycoord2=sort(ycoord2,1);
                                elseif flag==0
                                    %look if lane change is poosible in lane
4:-
                                    for g=m-10:m+10
                                        if  ismember(g,ycoord4)
                                            flag=0; %lane change not possible
in lane 4
                                        end
                                    end
                                    if flag==1
                                        %change lane here to lane 4:
                                        ycoord4(length(ycoord4)+1)=m;
                                        positions(m,4)=positions(m,3);
                                        positions(m,3)=0;
                                        speed(m,4)=speed(m,3);
                                        speed(m,3)=0;
                                        ycoord3(l)=0;
                                        ycoord4=sort(ycoord4,1);
                                    elseif flag==0 %if lane change was not
possible at all
                                        dist=ycoord3(l+1)-ycoord3(l);
                                        factor=(-
0.75+sqrt(0.5625+0.02804*dist))/0.01408;

newspeed=min(factor,speed(ycoord3(l+1),3));
                                    end
                                end
                            end
                        end
                        if(ycoord3(l)~=0)
                            oldcoord=ycoord3(l);
```

```
                                ycoord3(l)=round(ycoord3(l)+newspeed*0.01);
%position update per 100ms
                                ycoord3(l)=mod(ycoord3(l),5000)+1;%rollback
position
                                if(positions(ycoord3(l),3)==0)

positions(ycoord3(l),3)=positions(oldcoord,3);
                                    positions(oldcoord,3)=0;
                                    speed(oldcoord,3)=0;
                                    speed(ycoord3(l),3)=newspeed;
                                end
                            end
                        end
                    end
                    ycoord3=ycoord3(find(ycoord3~=0));
                end

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %for lane 4:
                %entry and exit here
                rand1=rand(1,1);
                rand2=rand(1,1);
                if rand1>0.833
                    entryramp=randi(3,[1,1]);%3 or 4???
                    if ~(ismember(entry(entryramp),ycoord4)) %checking if no
vehicle is already present there
                        %add a vehicle here
                        ycoord4(length(ycoord4)+1)=entry(entryramp);
                        positions(entry(entryramp),4)=max(max(positions));
                        speed(entry(entryramp),4)=(31-22).*rand(1,1) + 22;
                    end
                end
                if rand2>0.833
                    exitramp=randi(3,[1,1]);%3 or 4???
                    if ismember(exit(exitramp),ycoord4)
                        if positions(exit(exitramp),4)~=target
                            ycoord4=ycoord4(find(ycoord4~=exit(exitramp)));
                            positions(exit(exitramp),4)=0;
                            speed(exit(exitramp),4)=0;
                        end
                    end
                end
                ycoord4=sort(ycoord4,1);
                for k=1:1
                    for l=1:length(ycoord4) % for each car in a lane
                        if(l<=length(ycoord4))
                            acc=-5+10*rand(1,1);
                            newspeed=abs(speed(ycoord4(l),4)+acc);
                            if(l~=length(ycoord4))
                                if(ycoord4(l+1)-ycoord4(l)<=10)
                                    flag=1;
                                    %look if lane change is poosible in lane 3:-
                                    m=ycoord4(l);
                                    for g=m-10:m+10
                                        if  ismember(g,ycoord3)
                                            flag=0; %lane change not possible
                                        end
```

```
                                end
                                if flag==1
                                    %change to lane 3 here:
                                    ycoord3(length(ycoord3)+1)=m;
                                    positions(m,3)=positions(m,4);
                                    positions(m,4)=0;
                                    speed(m,3)=speed(m,4);
                                    speed(m,4)=0;
                                    ycoord4(l)=0;
                                    ycoord3=sort(ycoord3,1);
                                elseif flag==0 %if lane change was not
possible
                                    dist=ycoord4(l+1)-ycoord4(l);
                                    factor=(-
0.75+sqrt(0.5625+0.02804*dist))/0.01408;

newspeed=min(factor,speed(ycoord4(l+1),4));
                                end
                            end
                        end
                        if(ycoord4(l)~=0)
                            oldcoord=ycoord4(l);
                            ycoord4(l)=round(ycoord4(l)+(newspeed));
%position update per 100ms
                            ycoord4(l)=mod(ycoord4(l),5000)+1;%rollback
position
                            if(positions(ycoord4(l),4)==0)

positions(ycoord4(l),4)=positions(oldcoord,4);
                                positions(oldcoord,4)=0;
                                speed(oldcoord,4)=0;
                                speed(ycoord4(l),4)=newspeed;
                            end
                        end
                    end
                end
                ycoord4=ycoord4(find(ycoord4~=0));
            end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %answering the questions:
            new=[];
            [row,col]=find(positions==target);  %store the coordinates of
target car by finding it using its label in the positions matrix
            for o=1:4
                for p=1:5000
                    X=[p,500+(o-1)*3;row,500+(col-1)*3];    %factor of 3
multiplied to take into account the 3m lane separation
                    if positions(p,o)~=0
                        if pdist(X,'euclidean')<50  %communication range=50,
has to be change to run for a different range
                            if j==1

neighbours(length(neighbours)+1)=positions(p,o); %add all communication
neighbors to an array for the 1st second
                            end
```

```
                            new(length(new)+1)=positions(p,o);  %similar
array for every other second which will be overwritten every seconds
iteration
                        end
                    end
                end
            end
            if j==1
                common=intersect(neighbours,new);
                common_cont=intersect(neighbours,new);
            else
                common_cont=intersect(common,new);  %this array will store
the continuos neighbours every second
                common=intersect(neighbours,new);   %recalculate common
neighbours for consecutive seconds iteration
            end
            if timeflag==0
                if length(common)>=3
                    times(i)=j;
                elseif length(common)<3
                    timeflag=1; %set if less than 3 cars are common to avoid
further iteration
                end
            end
            if j==20 %continuous duration of time for question 3
                same_neighbours(i)=length(common_cont); %number of continuos
neighbors after 10 seconds
            end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        end
        [row,col]=find(positions==target);
        for o=1:4
            for p=1:5000
                X=[p,500+(o-1)*3;row,500+(col-1)*3]; %factor of 3 multiplied
to take into account the 3m lane separation
                if positions(p,o)~=0
                    if pdist(X,'euclidean')<50  %communication range has to
be changed here to get outputs for a different range
                        connectivity(i)=connectivity(i)+1; %increment for
every neighbor in communication range
                    end
                end
            end
        end
    end
    avg_cont_neighbours(cars/50-1)=mean(same_neighbours);
    duration(cars/50-1)=mean(times);
    v2v(cars/50-1)=mean(connectivity);
end
%Question 1:
figure(1);
plot(trafficdensity,v2v);
title('Connectivity Plot for communication range = 50 meters');
xlabel('Traffic Density');
ylabel('Average number of nodes');
%Question 2:
figure(2);
```

```
plot(trafficdensity,duration);
title('Connectivity Duration Plot (3 neighbours) for communication range = 50
meters');
xlabel('Traffic Density');
ylabel('Average duration');
%Question 3
figure(3);
plot(trafficdensity,avg_cont_neighbours);
title('Average number of same neighboursfor a continuous period of 10s, range
= 50 meters');
xlabel('Traffic Density');
ylabel('Average no. of same neighbours');
%Outputs for Question 4 taken by running this code for communication
%range=100m
```