UMinho | 2023

Vadym S. Hapanchak

Integration of different radio access technologies for vehicular connectivity

---

Universidade do Minho
Escola de Engenharia

Vadym Serhiyovych Hapanchak

Integration of different radio access technologies for vehicular connectivity

October, 2023

Universidade do Minho
Escola de Engenharia

Vadym Serhiyovych Hapanchak

Integration of different radio access technologies for vehicular connectivity

Doctorate Thesis
Doctorate in Advanced Engineering Systems for Industry

Work developed under the supervision of:

Prof. Dr. António Luís Duarte Costa

October, 2023

COPYRIGHT AND TERMS OF USE OF THIS WORK BY A THIRD PARTY

Acknowledgements

At the end of this important but also a long journey of my life, I would like to thank all the people who in some way contributed to the completion of this doctoral thesis, to whom I convey my sincere thanks.

Foremost, a special thanks to my supervisor Dr. António Luís Duarte Costa for your motivation, dedication, guidance and all the support given during these years. Collaborating with you as a Ph.D. student has been both an honor and a pleasure. I am also deeply grateful to the faculty members and my colleagues at the Laboratory of Computer Communications and Networks of the ALGORITMI center. Your knowledge and friendship have made my research experience better than I could have imagined.

On a more personal note, I wish to convey my profound gratitude to my family, particularly my parents, for their unwavering trust and unconditional support. To my wife, whose patience and understanding sustained me throughout this journey. To our daughter, Ana, who arrived at the beginning of my Ph.D. journey, has been a source of boundless joy and inspiration for us as parents.

I must also express my gratitude to the circle of friends and colleagues who have been by my side throughout this remarkable journey. I must give special thanks to my long-time friends, João Pereira and Paulo Araújo, for their unwavering friendship, shared experiences and motivating.

I also what to acknowledge the Portuguese funding institution ”FCT - Fundação para a Ciência e a Tecnologia” for supporting this research. Program in Advanced Engineering Systems for Industry (AESI) within doctoral scholarship grant: PD/BDE/150506/2019.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

________________________, ______________________________________

(Place) (Date)

____

(Vadym Serhiyovych Hapanchak)

> “*Coming together is a beginning; keeping together is progress; working together is success.*”
>
> — **Henry Ford**, American industrialist and business magnate (en)

Resumo

Integração de diferentes tecnologias de acesso rádio para conectividade veicular.

Nos últimos anos, a comunicação veicular surgiu como um componente crítico dos Sistemas de Transporte Inteligente (ITS) para melhorar a segurança rodoviária, a eficiência do tráfego e a experiência geral de condução. A evolução das redes veiculares ad hoc (VANETs) para as redes veiculares heterogéneas globais chama a atenção significativa da academia e da indústria em todo o mundo. Essa transformação, conhecida como *Internet of Vehicles (IoV),* pode revolucionar a forma como os veículos se comunicam entre si e com a infraestrutura. A coexistência de várias tecnologias no ambiente de comunicação veicular apresenta oportunidades promissoras para melhorar a conectividade. Um sistema heterogéneo incorpora as características únicas de várias tecnologias de acesso por rádio, melhorando o desempenho geral da rede e permitindo que os veículos beneficiem de altas taxas de transferência, baixa latência e alcance estendido de comunicação.

No entanto, apesar do vasto potencial da abordagem multitechnologia das redes veiculares heterogéneas, vários desafios persistem. A necessidade de um mecanismo de transferência de conexão sem interrupção e a seleção em tempo real da rede mais adequada permanecem como problemas de pesquisa em aberto. O foco central desta investigação é o desenvolvimento de algoritmos eficazes para a interoperabilidade funcional de diferentes tecnologias de comunicação sem fios. Para alcançar isso, adotamos o novo paradigma *multipath*, que permite o uso simultâneo das interfaces disponíveis, visando aprimorar a eficiência e a confiabilidade de sistemas de comunicação veicular.

Para reduzir o impacto das transições frequentes entre redes heterogéneas, desenvolveu-se um gerenciador de caminho adaptado para redes veiculares. Esse esquema inovador pode controlar o uso das interfaces disponíveis para garantir a comunicação sem interrupção e fornecer uma ótima experiência do utilizador num ambiente veicular multi-tecnológico. A solução proposta concentra-se na seleção dinâmica de rede para cada serviço, considerando vários fatores, como mobilidade do veículo, contexto de comunicação e requisitos dos serviços em execução. A eficácia da solução proposta é demonstrada por meio de avaliações abrangentes usando uma *framework* de emulação especificamente desenvolvida para apoiar esta investigação. Esta tese contribui para o investigação em curso sobre a comunicação veicular heterogénea e prepara o terreno para um futuro onde as redes veiculares integram perfeitamente várias tecnologias numa nova era de conectividade nas nossas estradas.

Palavras-chave: MPTCP, Multitecnologia, Multipath, Redes heterogéneas, V2X, Veículos Conectados

Abstract

Integration of different radio access technologies for vehicular connectivity

In recent years, vehicular communication has emerged as a critical component of Intelligent Transportation Systems (ITS) to enhance road safety, traffic efficiency, and overall driving experience. The evolution of Vehicular Ad Hoc Networks (VANETs) to global heterogeneous vehicular networks has captured significant attention from academia and industry worldwide. This transformation, known as the Internet of Vehicles (IoV), can revolutionize how vehicles communicate with each other and the surrounding infrastructure. The coexistence of various wireless technologies within the vehicular communication environment shows promising opportunities for improved connectivity. A heterogeneous system incorporates the unique characteristics of multiple Radio Access Technologies (RATs), enhancing the overall system performance.

However, despite the vast potential of the multi-RAT approach, several challenges persist, such as seamless handover mechanism and the real-time network selection. This thesis addresses these challenges by defining innovative techniques that facilitate vehicular communication within heterogeneous networks. A central focus of the research is the development of effective algorithms for the functional interoperability of different RATs. To achieve this, we adopt the novel multipath parading that enables the simultaneous use of available interfaces, aiming to enhance the efficiency and reliability of vehicular communication systems. Unfortunately, the unstable wireless connection and rapid topology changes caused by vehicle mobility can significantly impair the performance of multipath connections.

In order to reduce the impact of frequent handovers across heterogeneous networks, we designed an adaptive cross-layer assisted path manager tailored for vehicular networks. This innovative scheme can control RAT usage to provide an optimal and seamless user experience in a multi-technology vehicular environment. The proposed solution focuses on dynamic network selection for each service by considering several factors, such as vehicle mobility, the communication context, and the requirements of the ongoing services. The efficacy of the proposed solution is demonstrated through comprehensive evaluations using a realistic emulation framework specifically designed to support our investigation. This thesis contributes to the ongoing investigation on multi-RAT vehicular communication. It sets the stage for a future where vehicular networks seamlessly integrate various technologies to create a new era of connectivity on our roads.

Keywords: Connected Vehicles, Heterogeneous networks, MPTCP, Multi-RAT, Multipath, V2X

Contents

Appendices

List of Figures

List of Tables

List of Algorithms

Acronyms

3GPP	3rd Generation Partnership Project
5G	Fifth-generation wireless
5G-NR	5G New Radio
5GAA	5G Automotive Association
ADAS	Advanced Driver Assistance System
AHP	Analytical Hierarchy Process
AI	Artificial intelligence
ANDSF	Access Network Discovery and Selection Function
API	Application Programming Interface
ATSSS	Access Traffic Steering, Switching, and Splitting
BER	Bit Error Rate
C-V2X	Cellular V2X
CAD	Connected and Autonomous Driving
CAM	Cooperative Awareness Message
CAN	Controller Area Network
CAV	Connected and Autonomous Vehicle
CCU	Communications Control Unit
CORE	Common Open Research Emulator
CWND	Congestion Window
D2D	Device-to-Device
DSN	Data Sequence Number
DSRC	Dedicated Short-Range Communications
ECU	Electronic Control Unit
eNodeB	Evolved Node B

ETSI	European Telecommunications Standards Institute
FAHP	Fuzzy Analytic Hierarchy Process
GPS	Global Positioning System
GSM	Global System for Mobile communication
HetNet	Heterogeneous Networks
HMI	Human-Machine Interface
HoL	Head-of-line
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IoV	Internet of Vehicles
ITS	Intelligent Transportation Systems
Li-Fi	Light Fidelity
LTE	Long-Term Evolution
LXC	Linux Containers
MAPDU	Multi-Access Protocol Data Unit
MCDM	Multiple-criteria decision-making
MDP	Markov decision process
MIB	Management Information Base
MIH	Media Independent Handover
MIMO	Multiple-input multiple-output
ML	Machine learning
mmWave	Millimeter wave communication
MPTCP	Multipath TCP
NFC	Near Field communication
NTN	Non-terrestrial network
OBU	On-Board Unit
P2P	Peer-to-Peer

PLR	Packet Loss Ratio
PM	Path Manager

QoS	Quality of Experience
QoS	Quality of Service
QUIC	Quick UDP Internet Connections

RAN	Radio access networks
RAT	Radio Access Technology
RSS	Receiving Signal Strength
RSSI	Receiving Signal Strength Indication
RSU	Roadside Unit
RTO	Retransmission Timeout
RTT	Round-trip time

SAW	Simple Additive Weighting
SDN	Software-Defined Networking
SNR	Signal-to-Noise Ratio
SP	Service Profile
SUMO	Simulation of Urban MObility

TCP	Transmission Control Protocol
TNF	Triangular Fuzzy Number
TOPSIS	Technique of Order Preference Similarity to the Ideal Solution
TraCI	Traffic Control Interface

UDP	User Datagram Protoco
UE	User Equipment
URLLC	Ultra-reliable low-latency communication

V2D	Vehicle to Device
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VANET	Vehicular Ad Hoc Network

VHO	Vertical Handover
VLC	Visible light communication
VoIP	Voice over IP
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Chapter 1

Introduction

Connected vehicles will necessitate the use of heterogeneous or hybrid communication technologies to fully support a diverse range of Intelligent Transportation Systems (ITS) services in various scenarios. This chapter discusses the transformation brought by connected vehicles to human life with the growing integration of ITS to enhance road safety and connectivity services. It highlights the challenges of data transfer and service continuity in vehicular networks, and points out that addressing these challenges is crucial to improve communication efficiency. Most important, this chapter introduces the research questions that need to be answered and mentions the contributions of the thesis to the field of heterogeneous vehicular communication.

1.1 Perspectives and motivation

Vehicles have significantly transformed human lives worldwide, providing numerous benefits. With advancements in information technology and communication, connected vehicles have garnered considerable attention from academia and industry. Connected vehicles communicate with each other and the surrounding infrastructure through appropriate wireless technologies, referred to as Vehicle to Everything (V2X) communications, focusing on interconnecting cars with other surrounding objects [1, 2]. By exchanging information, connected vehicles can effectively prevent accidents and optimize traffic flow [3]. ITS have developed rapidly to enhance road safety and offer new infotainment and in-car connectivity services [4]. They are already in the deep integration phase worldwide. To address the growing demands of vehicular applications, including advanced driving and autonomous vehicles, vehicle manufacturers are now equipping V2X communication devices with multiple wireless interfaces. This approach enhances the connectivity capabilities of vehicles, allowing them to efficiently adapt to evolving communication needs and seamlessly integrate with various wireless networks for optimal performance and functionality. Indeed, combining multiple network technologies provides reliable broadband access for vehicle users and helps to improve communication efficiency and throughput.

However, these technologies' heterogeneity raises significant concerns regarding data transfer and service continuity. The vehicle may experience frequent handovers while traveling from one location to another, leading to potential disruptions in data transmission [5]. These challenges must be addressed to ensure vehicle users' seamless and uninterrupted communication experience. The Radio Access Technologies (RATs) used for vehicle communication possess distinct characteristics, including coverage area, data rate, central frequency, and modulation scheme. Consequently, integrating multiple RATs becomes essential to provide comprehensive connectivity across various networks, ensuring seamless mobility and service continuity for vehicle users. By combining these technologies, vehicular communication systems can effectively adapt to changing network conditions and user requirements, optimizing the overall communication experience.

In a highly dynamic communication environment characterized by frequent topology changes and disruptions in connectivity, meeting Quality of Service (QoS) requirements can be challenging. Ensuring consistent and reliable service delivery while accounting for varying reliability and latency requirements of different applications within a moving vehicle presents another set of challenges. The fluctuating nature of the communication environment demands robust and adaptive solutions to maintain optimal performance and satisfy the diverse needs of vehicular applications. Addressing these challenges needs innovative approaches and careful consideration of the specific requirements of each application, ultimately contributing to a seamless and efficient vehicular communication experience. Providing the 'always best' connectivity in a vehicular network is challenging due to rapid topological changes and random vehicle speed. Since the coverage area of the serving wireless network is limited, frequent disconnections are expected during vehicle movement. For effective communication, vehicles on the move must perform fast handovers between different heterogeneous networks to ensure seamless mobility. However, integrating diverse wireless technologies within a single communication system offers numerous advantages. These include:

- High Data Rates: Using multiple wireless technologies enables higher data rates, allowing for faster and more efficient transmission of information.
- Low Latency: Integrating of diverse wireless technologies can help reduce latency, enabling near-real-time communication and faster response times.
- Extended Communication Range: The combination of different wireless technologies expands the communication range, ensuring connectivity over larger distances.
- Improved Reliability: With multiple wireless technologies, communication devices can switch between networks seamlessly, enhancing reliability by maintaining connectivity even in areas with varying network availability.
- Enhanced Network Resilience: Integrating of diverse wireless technologies enhances network resilience by mitigating the impact of network failures or congestions in one technology by leveraging alternative communication channels.

1.2 Research questions

The interoperability of access technologies in heterogeneous vehicular networks remains a crucial area of research. Making different RATs work together effectively in a highly dynamic vehicular environment requires addressing several challenges. Based on our analysis of the existing literature and considering the open research problems, we can establish primary research questions that need to be addressed:

Is the integration of multi-RAT capabilities capable of improving vehicle connectivity? The initial focus of the research should be on investigating how the integration of diverse wireless technologies can enhance vehicular communication in terms of various aspects such as reliability, throughput, delay, and more. Understanding the potential benefits and limitations of incorporating multiple RATs is crucial for establishing the study's foundation and identifying each technology's strengths in the context of V2X. The research will investigate how communication performance changes when multi-RAT connectivity is used. Key challenges and trade-offs of a heterogeneous approach will be thoroughly examined. These challenges will be addressed through innovative solutions and careful considerations to achieve the desired level of efficiency and reliability in vehicular communication.

How can the interoperability of diverse access technologies be effectively achieved within heterogeneous vehicular networks? The research will define methods and techniques that can be employed to facilitate vehicular communication with multi-RAT capabilities. An effective handover algorithm will be implemented to ensure seamless connectivity, minimize service disruptions, and reduce signaling overhead in heterogeneous vehicular networks. This algorithm will maintain uninterrupted communication for vehicular users, even in dynamic network conditions and mobility. Furthermore, by employing an intelligent network selection procedure, the handover algorithm should guarantee that a particular application can be dynamically assigned to the most suitable network. The research aims to tackle the challenges associated with vehicular mobility and diverse network technologies by implementing the handover algorithm that can anticipate potential mobile handovers and significantly reduces the associated latency. Seamless connectivity between different RATs is vital for maintaining reliable communication during dynamic vehicular movements. The proposed method must ensure the continuity of sessions, even during network transitions. Hence, in this work, we have directed our efforts towards mitigating the disconnection time and its associated adverse effects. To address this challenge, a method that enables vehicles to seamlessly and concurrently establish a connection with any available network should be developed. Also, enabling the reduction of handover latency and minimizing packet loss will help to perform the system better overall. This research aims to enhance vehicular communication performance by optimizing the handover process, providing a smooth and uninterrupted experience for vehicle users.

How to select the most suitable network for a particular service? The network selection algorithm should consider user requirements and real-time network conditions. The algorithm must possess adaptability to the ever-changing vehicular environment and make intelligent decisions that guarantee better connection performance for each application. This adaptability is essential to provide an enhanced vehicular communication experience, accommodating the diverse needs of applications and users in dynamic and heterogeneous environments. By carefully considering user requirements and real-time network conditions, the algorithm can dynamically adjust its decisions to ensure seamless connectivity and optimized application performance. As vehicles traverse different communication environments, the algorithm should select the most suitable network, ensuring each application receives the necessary resources and QoS levels.

How the proposed solution can be evaluated? The validation and evaluation of the proposed solution require comprehensive testing and analysis to ensure its effectiveness and suitability for real-world vehicular communication scenarios. The solution can be tested using simulation environments to create realistic scenarios and replicate various vehicular communication conditions. A realistic testbed environment should be designed to simulate multiple vehicular communication scenarios. The testbed should include different types of vehicles, road layouts, and heterogeneous network infrastructures to mimic real-world conditions.

Addressing these research topics requires sophisticated techniques and algorithms that can adapt to the rapidly changing vehicular environment, ultimately leading to an uninterrupted communication experience for vehicle users. Coordinating multiple RATs to select the network with the best connectivity while considering the specific requirements of each application is indeed a complex task. Finding efficient solutions to optimize network selection and Vertical Handover (VHO) procedure within dynamic vehicular environments constituting a central focus of investigation in this thesis. Through a rigorous investigation of these research questions, this work seeks to contribute to the advancement of heterogeneous vehicular communication substantially.

1.3 Contribution and Outline of the thesis

This thesis contributes to the field of heterogeneous vehicular communication by addressing critical challenges and advancing the state of the art in V2X connectivity. The primary contributions of this research are as follows:

- **Multi-RAT integration**: The investigation into how the integration of diverse wireless technologies can enhance V2X communication in terms of reliability, throughput, delay, and other key metrics. The research demonstrate the potential benefits of combining different RATs to meet the varied applications requirements.

- **Seamless vertical handover**: Developing and implementing an effective handover algorithm to ensure seamless connectivity, minimize service disruptions, and reduce signaling overhead during handovers in heterogeneous vehicular networks. This algorithm is crucial in optimizing resource allocation for enhanced V2X communication.

- **Intelligent network selection**: Exploring methods to design a network selection algorithm that guarantees better connection performance based on real-time network conditions and user requirements. The research adopt proper metrics and adaptive methods to ensure the best connectivity and efficient network utilization.

- **Evaluation testbed**: Establishing a comprehensive validation and evaluation framework, utilizing simulation/emulation and field trials, to assess the proposed algorithms' performance under various scenarios. The research provide quantitative and qualitative insights into the efficiency and effectiveness of the proposed solutions.

The thesis is organized into the following chapters:

1. **Introduction**: Chapter 1 provides an introduction to the field of heterogeneous vehicular communication, outlining the research objectives, contributions, and the motivation behind the study. The chapter also presents the research questions that guide the investigation.

2. **State of the art**: Chapter 2 comprehensively introduces vehicular communication, heterogeneous RATs, V2X hardware, and network selection algorithms. It discusses the benefits and challenges of multi-RAT usage in vehicular networks. It also provides insights into existing RAT interoperability approaches and highlights the gaps this research aims to address.

3. **Emulation framework**: Chapter 3 presents an assessment study using the emulation platform to conduct real-time experiments in heterogeneous vehicular networks. First, emulation setup, which integrates simulated networks with virtualized devices, discussed in detail. The modeled vehicles employ realistic mobility patterns provided by a dedicated traffic simulator. Furthermore, the performance of the multipath approach is analyzed in the context of vehicular communication.

4. **Design of cross-layer path manager**: Chapter 4 focuses on the development and implementation of the seamless handover algorithm across heterogeneous vehicular networks. It presents the design principles, decision-making mechanisms, and adaptive strategies to ensure uninterrupted vehicular connectivity. This chapter deeply explores the applicability of Multipath TCP (MPTCP) in supporting robust and effective Vehicle to Network (V2N) communication. The chapter starts by identifying and discussing several challenges inherent to heterogeneous vehicular networks. Next, it introduce a cross-layer path management strategy, exploiting interactions between upper and lower protocol stack layers to enhance the control of multi-RAT environments.

5. **Performance evaluation**: Chapter 5 analyzes the performance of the proposed Cross-layer Path Manager. First, we describe the methodology and techniques used to evaluate the performance, including details on the experimental setup and data collection. Using our emulation framework, we will show that the proposed strategy facilitates seamless mobility across heterogeneous networks and significantly reduces handover latency and out-of-order packet delivery. The chapter provide a detailed analysis of the results obtained and explores their implications. Moreover, it entails a comparative study of MPTCP's performance, both with and without the activation of our path manager.

6. **Intelligent path selection**: Chapter 6 explores the design of an efficient network selection algorithm, considering factors such as real-time network conditions and user requirements. It discusses the metrics and methods for achieving uninterrupted connectivity and optimized network utilization. This section introduces an intelligent path management algorithm that dynamically allocates the most suitable underlying network for each active application. This selection process considers various factors, including path quality, vehicle mobility, and service attributes. Our novel approach adopts a dynamic and comprehensive strategy that customizes network selection to the specific demands of each service, guaranteeing an optimal pairing with the appropriate access technology at all times.

7. **Discussion and Conclusion**: Chapter 7 provides a comprehensive discussion of the research outcomes and contributions. It draws conclusions based on the research findings and proposes future research directions in the field of heterogeneous vehicular communication.

1.3.1 List of publications

- V. S. Hapanchak and A. D. Costa *”Emulation of Multipath Solutions in Heterogeneous Wireless Networks Over Ns-3 Platform”*. Simulation Tools and Techniques. SIMUtools 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 424. Springer, Cham. `https://doi.org/10.1007/978-3-030-97124-3_1`

- V. S. Hapanchak and A. D. Costa, *”Emulation of Multi-Connectivity in Hybrid Vehicular Networks”*, 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 2022, pp. 1-6, `10.1109/ICECET55527.2022.9873013`.

- V. S. Hapanchak and A. D. Costa, *”Design and Evaluation of a Cross-Layer MPTCP Path Manager for Vehicular Networks”*, 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2022, `10.23919/SoftCOM55329.2022.9911227`.

- V. S. Hapanchak and A. Costa, *”A Cross-layer Approach for MPTCP Path Management in Heterogeneous Vehicular Networks”*, in Journal of Communications Software and Systems, vol. 19, no. 1, pp. 103-113, 2023, `10.24138/jcomss-2022-0177`.

- V. S. Hapanchak and A. Costa and J. Pereira and M. Mesquita Rodrigues, *“An Intelligent Path Management in Heterogeneous Vehicular Networks”*, Journal of Vehicular Communications, 2023, `DOI: 10.2139/ssrn.4502249`.

1.3.2 Miscellaneous

We have made all the source code for the solutions and tools developed during this thesis publicly accessible to the community. This decision reflects our commitment to open collaboration and knowledge sharing, fostering an environment where researchers and practitioners can access, utilize, and build upon our work to advance the field further. By providing this access, we aim to facilitate innovation, encourage peer review, and contribute to the collective effort of enhancing network technologies and solutions. We believe that sharing our code will benefit the academic and research communities and promote the development of practical applications in the broader technological landscape.

- `https://github.com/vandit86/mptcpd-plugin` : The userspace path manager with the cross-layer assistance.

- `https://github.com/vandit86/mptcpd` : The userspace path manager with the improved network selection algorithm.

- `https://github.com/vandit86/ns3-`scratch : The emulation framework for multapath communication experiments.

- `https://github.com/vandit86/ns-3` : The extended version of the framework with the support of vehicle mobility simulation.
- `https://github.com/vandit86/multipath-tcp-tools` : A collection tools for analysis and configuration of MPTCP. Contains applications to help visualize MPTCP packet traces.
- `https://github.com/vandit86/mptcp-traffic-generator` : Client/Server MPTCP traffic generator with the ability to measure one way delay.

Chapter 2

State of the art

This chapter provides an overview of the current knowledge, research, and advancements in the field of heterogeneous vehicular communications. It demonstrates the understanding of the existing literature and establishes the context for the study by highlighting key findings and trends. The chapter summarizes recent research and developments and discusses key concepts fundamental to understanding the field. It also discusses the various methodologies and approaches employed by researchers in the area, identifying the strengths and limitations of these methods and explaining their contributions to advancing knowledge. It's important to emphasize that this chapter provides a concise overview of the current state of the art, which is crucial for coordinating the material presented in the rest of the document. The pre-thesis document gives a more detailed review of existing solutions and approaches to RAT interoperability [6].

2.1 Introduction to vehicular communications

Over the past few years, there has been a surge in the development of vehicular network applications and services to meet the growing demand for ubiquitous access to information. This has led to the emergence of a diverse range of innovative solutions that enable vehicular users to access information from anywhere and at any time [2, 7, 8]. Intelligent Transportation Systems (ITS) have experienced rapid development, aiming to enhance road safety and provide in-car connectivity. Wireless communication is a crucial element in ITS, facilitating cooperation among vehicles, roadside infrastructure, and remote entities to optimize traffic efficiency and enhance the overall comfort of road users. The growth of the intelligent transportation industry has lead to the creation of a secure and efficient traffic management system that benefits drivers and passengers. The vehicle interconnection network facilitates communication among vehicles, between vehicles and road infrastructure, and with service centers through wireless communication technology. This integration seamlessly combines the human-vehicle-road environment, enhancing overall traffic safety and efficiency.

Wireless communication is critical for strengthening in-vehicle driving systems, significantly reducing reaction times, extending the vision range of ego vehicles beyond line of sight, and enabling cooperative maneuver planning. A widely investigated use case is cooperative awareness, where each car equipped with a radio communication device periodically broadcasts its state information, including position and speed, which allows for detecting potential collision threats and facilitates driver assistance systems to prevent accidents effectively [9].

2.1.1 Vehicular Ad Hoc Networks

Vehicular Ad Hoc Networks (VANETs), formed by vehicles equipped with sensors and communication devices, have garnered considerable attention from network operators and service providers. They see the potential of VANETs not only for providing safety and emergency services, but also for offering a wide range of informational and entertainment applications. VANETs have traditionally focused on spontaneous networking through short-range communication and have shown less emphasis on integrating cellular networks. While VANETs can be built upon various wireless technologies, communication standards such as Dedicated Short-Range Communications (DSRC) have been specifically designed for automotive applications [7, 10]. Early deployment of VANETs has been motivated primarily by enabling road safety applications, such as collision warning and emergency event informing, through direct messaging between vehicles and roadside infrastructure. Moreover, VANETs support non-safety infotainment applications, which enable passengers to access various services like interactive communication, internet access, online games, and information updates when vehicles are on the move. However, the DSRC alone falls short of meeting the requirements of future ITS applications, particularly with the rise of autonomous vehicles [11]. Despite significant research in VANETs, challenges related to scalability, limited coverage area, and unbounded delay persist. For instance, the DSRC standard has been designed explicitly for vehicular communications and provides seamless handover by design. Nevertheless, it faces challenges such as low market penetration, dependency on specific infrastructure installations, and limitations in terms of data rate.

In VANET communication, the exchange of data between devices needs to be swift and efficient due to the high-speed mobility of the participating vehicles. This stands in contrast to typical Wireless Fidelity (Wi-Fi) networks, where the hotspot remains stationary, resulting in a relatively stable network. However, VANET communication systems face unique challenges, including compliance with traffic regulations such as obeying traffic lights and stopping at designated points. Nevertheless, applications relying on data forwarding within the VANET infrastructure face limitations concerning connectivity modes, switching capabilities, and available bandwidth, primarily governed by the IEEE 802.11p standard.

Vehicular networks represent a rapidly evolving domain where vehicles establish connections among themselves and with the surrounding road infrastructure. However, the intrinsic high mobility and dynamic topology changes within this environment pose significant challenges when attempting to predict the duration of a vehicle's network connectivity. This prediction relies on a multitude of parameters, including but not limited to velocity, direction, traffic conditions, network signal strength, inter-vehicle and

Roadside Unit (RSU) distances, signal interference, obstacles, multipath fading, transmission range, transmission power, power gain, user preferences, and Quality of Service (QoS) requirements. Additionally, unanticipated events like accidents on the road can disrupt communication flow. Consequently, there is a growing need for the advancement of intelligent VANET technology to address these dynamic and demanding scenarios effectively.

The vehicular network experiences frequent and fast topology changes due to vehicles' high speeds and radio propagation characteristics. Vehicles may move in various directions, resulting in the highly dynamic nature of a vehicular network. The availability of links in a vehicular network is typically low, influenced by factors such as the radio communication range, the direction of vehicles, and their speed. These variables directly impact the lifetime of the link, which may result in higher packet loss with reduced throughput. The density of the vehicular network fluctuates based on the traffic density, which is influenced by both time and geographical area. Network scale can be exceptionally high during traffic congestion in dense urban areas and considerably low in suburban or rural traffic conditions. Vehicles adhere to predictable mobility patterns dictated by the road infrastructure, including roads, streets, highways, traffic lights, speed limits, and prevailing traffic conditions. Unlike other networks, the accessibility of vehicles in heterogeneous networks often depends on their precise geographic location. The presence of buildings, trees, and other objects that act as obstacles to signal propagation, resulting in interference, shadowing, multipath propagation, and fading effects, further complicating the communication process.

In the modern world, vehicular users require seamless and affordable communication while moving in urban, suburban, and even rural areas. For vehicles to engage in VANET communication, each is equipped with specific Wi-Fi devices, including On-Board Units (OBUs) responsible for core communication functions, Application Units (APUs) that display OBU data to drivers, and RSUs which serve as routers, guiding and regulating the flow of information between vehicles. Vehicles are expected to integrate a wide range of ITS applications, such as road safety, traffic control, and various entertainment services.

2.1.2 Internet of Vehicles

The Internet of Vehicles (IoV) paradigm not only leverages ubiquitous vehicular connectivity for a wide range of services but also integrates vehicular intelligence. IoV has the capacity to analyze extensive data gathered from vehicles, road infrastructure, and the surrounding environment, enabling effective driver monitoring and decision-making based on this integrated information. The development of IoV is driven by the need to overcome the commercialization challenges faced by VANETs. The IoV presents significant market opportunities, not only for the automobile industry but also for various other sectors, including IT equipment manufacturing, the software industry, and Internet service providers.

Many applications, including condition monitoring/warning systems, analytic systems, cooperative perception, location-based services, and real-time applications, are expected to be integrated into vehicles within the IoV environment. These advancements aim to provide users with enhanced connectivity and diverse services to enrich their driving experience and improve overall transportation efficiency. The VANET framework's limitations in ensuring continuous and global ITS services result from its pure ad-hoc network architecture, which causes vehicles to lose connectivity and services when disconnected from the ad-hoc network and prevents collaboration with alternative reachable networks while on the road.

Research and development efforts in vehicular communication continue to explore innovative solutions and technologies that can effectively address the evolving demands [12]. The aim is to develop comprehensive and robust vehicular communication systems that can support a wide range of applications and provide seamless integration of vehicles within the ITS framework. The deployment of ITS is anticipated to be extensive within the IoV paradigm, catering to a diverse range of applications, from low-data-rate traffic control services to high-data-rate, latency-sensitive multimedia services. ITS relies on the coordination of sensors, OBUs, and trusted platform modules to facilitate the sharing of critical vehicle information. The communication infrastructure of IoV could enhance response capabilities within ITS, primarily through applications such as safety-critical services. Furthermore, IoV is anticipated to deliver uninterrupted network connectivity and adaptability, even in the face of network disruptions and prolonged delays during emergency situations. The IoV architecture must facilitate the interconnection of vehicles with diverse networks and devices. Consequently, an open and adaptable layered architecture that accommodates various technologies is better suited for this purpose.

Current wireless access technologies have different characteristics regarding coverage area, data rate, central frequency, and modulation scheme. Therefore, each network offers distinct and unique benefits that collectively contribute to the overall effectiveness of the hybrid system. Consequently, the integration of different wireless networks, such as ad-hoc and cellular, is expected to address several requirements of ITS services. Indeed, in future IoV environments, it is envisioned that a variety of wireless interfaces with heterogeneous capacities will coexist [13]. This mix of communication technologies will enable vehicles and infrastructure to efficiently exchange information and support a wide array of vehicular applications, ultimately contributing to safer, more efficient, and connected transportation systems [14].

Figure 1 provides an overview of ITS application categories along with the different available heterogeneous communication options. Each application category has distinct requirements concerning delay, reliability, and throughput. Similarly, each wireless technology option possesses specific capabilities and performance characteristics, making it more or less suitable for serving different applications in various communication and driving scenarios. The challenge here is to efficiently distribute the network bandwidth among various applications using the available multiple wireless networks in a way that maximizes the overall application utility while staying within the constraints of the total available bandwidth.

As depicted in Figure 1, we assume a heterogeneous vehicular network context where multiple networks are typically present to assure a vehicle communication. For a vehicle, each of the available Radio Access Technology (RAT) acts as a communication pipe connecting the vehicle to the network. In situations where there are N wireless networks available, there are essentially N communication pipes, each with its own specific capacity. In scenarios with only one available network, handovers are unnecessary, as a vehicle can either connect directly to the sole available network or remain disconnected. A critical

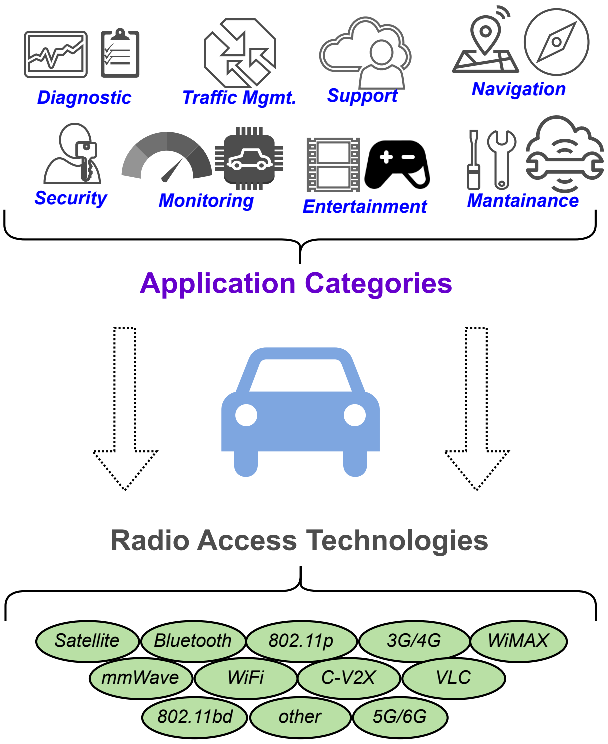


Figure 1: Connected vehicle application categories and communication options.

challenge in vehicular heterogeneous communications lies in effectively mapping the different applications to one or a suitable combination of multiple RATs under fluctuating radio channel and network load conditions. In other words, we want to make mapping from the application layer to the link layer (L5-to-L2 mapping) which allows to route the upper layer packet to a corresponding RAT network type.

In addition to serving as a network interface for the application layer, the developed solution must possess specific attributes crucial for seamless handover between multiple access networks. These properties include:

- The ability to establish relationships with a set of physical interfaces, effectively abstracting or concealing their existence.
- Dynamic connection and attachment capabilities with the physical interfaces.
- Compatibility with multiple access technologies.
- The ability to dynamically map transmission/reception functions of the logical interface with the transmit/receive functions of physical interfaces.
- Maintenance of session flow information for each attached physical interface.

These properties collectively contribute to concealing the presence of multiple physical interfaces on the vehicle, effectively masking any changes occurring at lower layers among the physical interfaces.

2.2 Vehicular heterogeneous networks

An IEEE 802.11p-based roadside access point serving moving vehicles has demonstrated the feasibility of providing internet access to vehicles within a single network. However, relying solely on RSUs for ubiquitous vehicle internet access presents challenges. Establishing high-rate, widespread RSU connections is impractical and costly due to the extensive deployment and maintenance expenses. To ensure that vehicles can access networks at will, even in areas lacking RSU coverage, radio access networks like cellular networks and Wi-Fi can be leveraged to enhance vehicular communications.

The rapid and expansive growth of telecommunication systems, coupled with the widespread deployment of the IoV, has paved the way for the development and standardization of numerous novel communication processes and protocols. Within this landscape, the domain of VANETs has kept pace with this evolution and transitioned into the realm of IoV. In the realm of the IoV, intelligent interfaces play a pivotal role in seamlessly integrating heterogeneous networks. One of the primary objectives within IoV is achieving interoperability among diverse devices and systems. Subsequently, the paradigm of heterogeneous vehicular networks emerged, bringing with it a suite of associated protocols, all aimed at enhancing the communication capabilities of vehicles. For instance, we can establish vehicular heterogeneous networks, as a combination of VANETs and cellular networks, designed to enhance vehicular communications. Similar to VANETs, cellular networks can offer services for vehicles. In cases where internet connections need to transition between these heterogeneous networks, primarily for performance and enhanced availability, ensuring seamless handovers becomes a crucial initial requirement.

In recent years, there has been considerable attention on the evolution of traditional VANETs into global vehicular Heterogeneous Networks (HetNet). The development of vehicular HetNet aims to deliver multiple and heterogeneous services, enabling end-to-end communication and seamless data exchange at reasonable QoS levels for vehicular users, regardless of their location or movement. In the heterogeneous networks, each vehicle can possess multiple radio interfaces, enabling them to concurrently connect to various domains and RATs.

The key idea is to tightly integrate multiple RATs available in the OBUs, allowing data messages to be sent over the network(s) that best suit the performance requirements and network conditions at any given moment. This approach enables the coordination of multiple RATs and ensures that vehicles maintain reliable and efficient connectivity by dynamically adapting to the changing vehicular environment and driving situation. Efficient utilization of heterogeneous vehicular wireless networks has the potential to yield several benefits, including cost savings through the utilization of free Wi-Fi, the facilitation of high-throughput applications by aggregating bandwidth, and the enhancement of network performance in the face of wireless communication's inherent dynamics and uncertainties.

However, current vehicular network applications are primarily designed for use with a single wireless network technology. Consequently, effectively utilizing the capabilities of multiple RATs concurrently presents substantial challenges. Vehicular applications exhibit varying demands for network services, making it a non-trivial task to efficiently allocate resources from the range of heterogeneous wireless networks to these diverse applications. For example, the video streaming to vehicles, demands high-throughput and real-time communication capabilities. On the other hand, location-aware weather forecasts can accommodate communication with delays and lower bandwidth. Furthermore, different wireless networks often exhibit distinct characteristics, such as variations in throughput, delay, and reliability. Integrating these networks into a unified system with a standardized application interface can be a complex task. For instance, cellular links typically offer lower bandwidth but greater service continuity. Free Wi-Fi links, in contrast, tend to provide higher bandwidth but may suffer from service discontinuity. VANET links and paths can offer high bandwidth, but are subject to the mobility of vehicles and encompass intricate dynamics and uncertainties.

The vehicular communication system encompasses not only vehicles but also other communication entities like roadside units, cloud and fog networks, the Internet and pedestrians. Vehicle to Everything (V2X) communication, enabling seamless interaction between vehicles and their surrounding entities, using heterogeneous technologies like Wi-Fi, DSRC and Long-Term Evolution (LTE) networks. This facilitates the dissemination of valuable information to drivers, passengers, the cloud, and even entities in proximity, such as pedestrians and cyclists. The purpose is to enhance safety and cater to both safety-critical and non-safety-related services and applications. The communication architecture of IoV encompasses not just vehicles and RSUs but also a diverse array of communication devices. While this inclusion adds complexity, it aligns the architecture with market demands, distinguishing it from VANETs. As shown in Figure 2, the heterogeneous network architecture of IoV includes different types of vehicular communications, which include Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Vehicle to Network (V2N), Vehicle to Pedestrian (V2P), Vehicle to Device (V2D) etc.

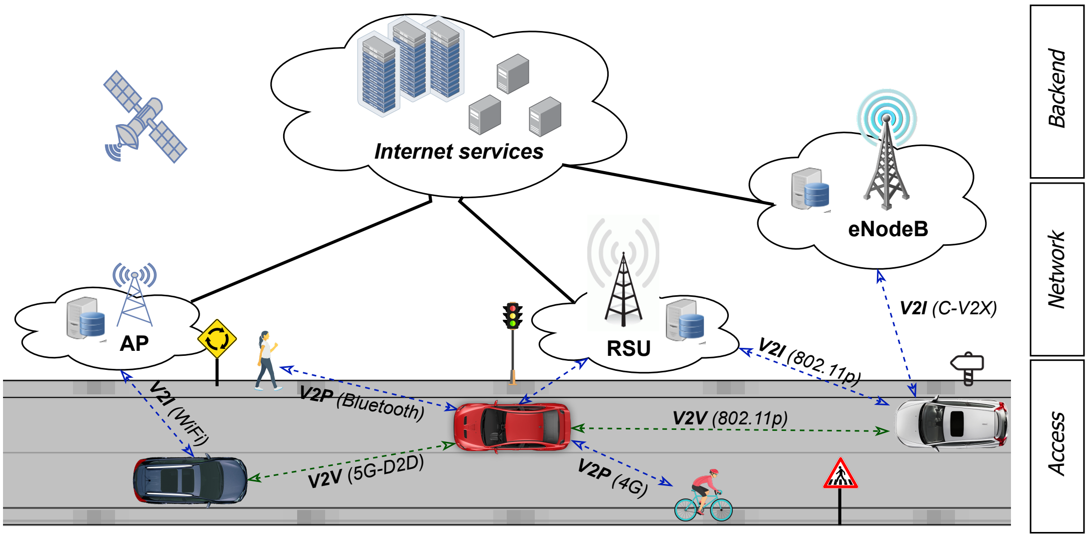


Figure 2: Vehicular heterogeneous communication system.

V2V communication, also known as inter-vehicle communication, facilitates direct communication between vehicles without relying on fixed infrastructure support [14]. V2V communication enables short and medium-range interactions, providing cost-effective deployment options and facilitating the rapid transmission of concise messages with minimal latency. Its primary applications include safety, security, and dissemination purposes. Communication between vehicles primarily serves the purpose of exchanging information, including vehicle location, speed, and driving direction. In V2V communication, vehicles exchange data that describes their internal states and the surrounding environment, enhancing the perceptual horizon of communication partners. The valuable information from a vehicle's sensors can be shared with nearby road users. The wireless network is created spontaneously; thus, the V2V communication follows an entirely ad hoc nature, often referred to as VANETs. When two vehicles are within each other's communication range, they establish a direct connection; otherwise, they employ multi-hop communication to relay data.

V2I communication enables vehicles to establish communication with the roadside/network nodes with stationary positions, primarily for information and data gathering purposes. This communication is facilitated through points of attachment, including RSUs, hotspots, and cellular base stations, that could be strategically placed at dedicated locations along roads and highways, utilizing available infrastructure such as road signs, traffic lights, and toll gantries. V2I communication enables vehicles to interact with and receive information from the infrastructure to enhance road safety and traffic management and provide ITS services to drivers and passengers. For instance, vehicles can interact with the infrastructure to influence factors like traffic light durations and report their own internal state and traffic situations. Moreover, non-safety critical applications are primarily associated with V2N communication, allowing passengers to access a variety of services while vehicles are moving, including internet connectivity, entertainment services, online multimedia content, and interactive communication. The network architecture extends to cloud, fog, and grid networks, as well as the underlying backbone wired network of the service providers and server farms. The communication range of the wireless technology and vehicle speed affect the connection time between the vehicle and the roadside infrastructure. The link availability is low, especially in highway scenarios, where the connection is often lost during data transmission. The rapid topology changes result in packet losses and reduced throughput, which negatively affect communication performance.

V2P/V2D represents an important facet of vehicular communication, facilitating interactions between vehicles and mobile devices carried by pedestrians, passengers, and cyclists [15, 16]. This mode of communication acknowledges the distinct mobility patterns and behaviors of these non-vehicle entities within the transportation ecosystem. Through V2P communication, vehicles can establish connections with pedestrians, passengers, and cyclists, enabling the exchange of critical information, alerts, and data that contribute to improved road safety, enhanced situational awareness, and novel services.

The potential of integrating diverse wireless technologies is evident in its capacity to effectively address the myriad requirements of ITS services, thereby fostering advanced capabilities and heightened performance within vehicular communication systems. To achieve this, the OBU must be equipped to seamlessly support an array of wireless networks, including Wi-Fi, DSRC, Global System for Mobile communication

(GSM), Worldwide Interoperability for Microwave Access (WiMAX), 4G/5G, and beyond. This holistic approach to heterogeneous vehicular communication ensures a comprehensive solution that caters to the dynamic needs of modern vehicular environments. The concept of heterogeneity is also included within the Fifth-generation wireless (5G) framework, as put forth by the 5G-PPP Group [9]. The future 5G networks are envisioned as an amalgamation of diverse wireless networks, encompassing existing and forthcoming RATs. Nonetheless, managing heterogeneous networks presents a complex challenge, potentially leading to network fragmentation and suboptimal utilization of network resources. Moreover, the transition from one RAT to another, coupled with the intricate multi-hop routing of network traffic, can increase end-to-end delays. This intricacy is particularly noteworthy in densely populated vehicular environments characterized by high resource demands and many potential network routing paths. Consequently, the quest for optimal routes within minimal time becomes imperative in such scenarios.

2.3 V2X Radio Access Technologies

Numerous heterogeneous RAT have been assessed for their applicability to V2X communication, such as 802.11p, 802.11bd, 802.11ax, Cellular V2X (C-V2X), WiMAX, cellular networks 4G/5G, Bluetooth, Visible light communication (VLC), Satellite, Near Field communication (NFC) and Millimeter wave communication (mmWave). However, the complex performance requirements inherent in vehicular networking applications present a formidable challenge for any communication technology [17]. In light of this, adopting a heterogeneous approach that harnesses the capabilities of multiple RAT has become indispensable in guaranteeing optimal vehicular connectivity [18]. Each of these RAT possesses distinctive characteristics concerning signal propagation, data throughput, and latency, leading to varying advantages and disadvantages when employed for V2X communications [19]. This section reflects the most recent standards for cellular communications from the 3rd Generation Partnership Project (3GPP) and Wireless Local Area Network (WLAN) communications from the Institute of Electrical and Electronics Engineers (IEEE).

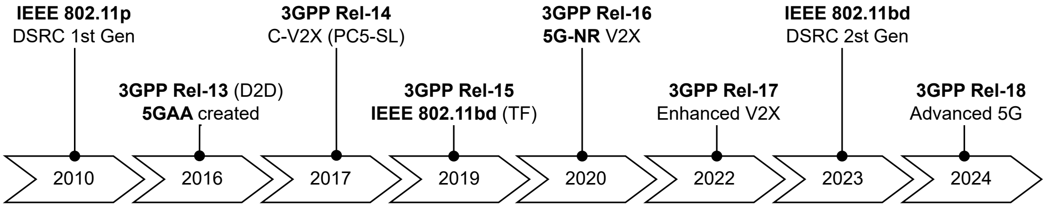


Figure 3: Evolution of V2X standards.

Illustrated in Figure 3 is a comprehensive depiction of the evolutionary trajectory of key V2X communication standards. Of particular note, the ascent of 5G New Radio (5G-NR) showcases encouraging strides in terms of relative enhancements over its predecessor technologies. Notably, 5G-NR demonstrates remarkable potential in facilitating V2X communications and powering applications for autonomous vehicles, surpassing the capabilities of current RATs [20, 21]. The 5G is not a singular standard, but an amalgamation of innovative technologies that collectively represent the imminent progress of mobile cellular networks.

2.3.1 Dedicated Short-Range Communications

WLAN stands out as the predominant and widely adopted option available in the market. The IEEE 802.11 wireless network family dominates this category. Various working groups have contributed to different iterations of the 802.11 family, including 802.11b, 802.11g and 802.11n. Each of these variants possesses distinct characteristics and confronts unique challenges, rendering them suitable for diverse environments. In general, this standard facilitates short-range radio coverage while offering relatively high data rates. For instance, 802.11n claims to support data rates of up to 600 Mbps. However, it became evident that these high data rates were not practically attainable in a mobile environment. The inherent characteristics of VANETs, marked by extreme mobility and frequently changing network topologies, made these variations unsuitable. Consequently, a new version of the WLAN standard, known as IEEE 802.11p, was introduced to address the specific needs needs of vehicular networks. It is designed to accommodate a wide array of applications and services within the realm of ITS, with a particular focus on safety-critical applications.

DSRC, is the pioneering standard and primary wireless technology for VANET [22]. It facilitates communication among vehicles on the road, both with and without the aid of any pre-established infrastructure alongside roads. In 2008, the European Telecommunications Standards Institute (ETSI) designated 70 MHz of spectrum within the 5.8 GHz band for the exclusive use of DSRC to enable “day one” road safety applications, such as frontal collision warnings, blind spot warnings, and intersection motion assistance. At the MAC/PHY layers, DSRC is based on the IEEE 802.11p protocol, which operates fully distributed and facilitates the rapid exchange of concise broadcast messages among vehicles. DSRC systems support vehicle speeds of up to 200 km/h within a communication range of 300 meters to 1 kilometer. However, its operational scope usually spans a limited number of 10 MHz channels, which presents challenges in effectively managing substantial data volumes.

The protocol additionally offers the flexibility to postpone tasks like authentication, encryption, and comprehensive identification to higher-level protocols. In Europe, the network architecture and security protocols are defined in the ETSI ITS-G5 upper layer amendment [2], which provides a multi-hop routing system, the basic transport protocol (BTP) over the GeoNetworking service, and also supports standard Transmission Control Protocol (TCP) and User Datagram Protoco (UDP) over IPv6. The ad hoc nature of IEEE 802.11p brings advantages such as independent operation from cellular infrastructure, ubiquitous connectivity, low control overhead, and potentially low delays. However, due to the random medium access schemes, the performance of V2X communication degrades under channel congestion, resulting in packet loss and increased update times. Furthermore, despite significant research done on 802.11p, it suffers from scalability issues, limited coverage area, and unbounded delay [4].

ITS-G5 is originally based on the IEEE 802.11p standard, which operates in the 5.9 GHz DSRC band.

This frequency band is reserved for transportation-related communication to minimize interference and ensure reliable and low-latency communication. One of the primary use cases for ITS-G5 is in safety applications. Vehicles equipped with ITS-G5 technology can exchange safety-critical information, such as warnings about accidents, road hazards, traffic signals, and pedestrian crossings. This allows vehicles to react proactively to potential dangers. It supports efficient broadcasting of messages. This is crucial for disseminating safety-related information to all nearby vehicles, even when the sender doesn't know their specific locations. The ITS-G5 stack incorporates security measures to protect the authenticity and integrity of messages. One of the key features of ITS-G5 is its ability to coexist with other wireless communication technologies and systems. This is important because the road environment is diverse, and different regions and countries may have different preferences for V2X communication technologies. Recent researches has proposed the implementation of heterogeneous networks to effectively cater to the diverse demands of vehicular applications, encompassing the specific needs of IoV. ITS-G5 has been designed to work alongside other communication technologies, including cellular (e.g., 4G and 5G), Wi-Fi, and other V2X technologies.

Despite the substantial potential of VANETs to mitigate traffic safety and enhance efficiency while reducing operational costs, they have encountered challenges in garnering commercial interest. The commercialization hurdles faced by VANETs encompass several issues, such as their reliance on pure ad hoc network architecture, the unreliability of Internet services, compatibility issues with personal devices, the absence of cloud computing integration, among others.

With the aim to enhance the existing 802.11p technology with advanced PHY/MAC techniques introduced in the newest Wi-Fi standards, a new IEEE 802.11bd standard [23] for next-generation V2X has been recently published in early 2023. The primary design objectives of 802.11bd include supporting high mobility (up to 500 km/h), high data rate and increased communication range (up to 2 km). Emphasis was posed to modifications at the PHY layer, in particular, to Multiple-input multiple-output (MIMO) techniques, improved channel coding, and better pilots placing. Additionally, 802.11bd should guarantee coexistence and backward compatibility with 802.11p.

2.3.2 Cellular networks

As cellular infrastructures have evolved, 4G/LTE has emerged as a popular choice for vehicular environments. It offers a reasonable data rate and a smoother handover management mechanism compared to WLAN. Numerous studies have shown that 4G/LTE deployment in highly mobile environments is challenging and requires careful planning. While the fourth generation of cellular technology (4G LTE) offers some support for V2X communications, its design, which prioritizes multimedia and telephony applications, falls short in meeting the demands of safety-critical scenarios.

C-V2X communication, standardized by the 3GPP, has long been considered a prospective contender in the realm of V2X communications [24]. In 2016 the 5G Automotive Association (5GAA) was created, shaping the path to 5G by supporting C-V2X, as well as working to deliver innovations in on-device intelligence and integration in connected vehicles [25]. The momentum behind C-V2X gained traction following the emergence of vehicular standards within cellular communication, epitomized by 3GPP Release 14 in 2017, denoted as LTE-V2X. Capitalizing on the extensive infrastructure inherent to cellular communication and the introduction of direct Device-to-Device (D2D) communication, C-V2X seamlessly encompasses the entire spectrum of V2X communication paradigms.

However, the intrigue surrounding C-V2X experienced a notable upswing with the advent of the fifth-generation cellular technology, 5G-NR, introduced in 3GPP Release 15 in 2019 [26]. This transformative evolution of cellular networks, orienting toward domains like smart cities, Internet of Things (IoT), and autonomous driving, significantly enhanced cellular technology's allure as a V2X communication contender. Based on research conducted on the current LTE network implementations, it has been shown that C-V2X holds promise as a viable solution for fundamental V2X applications. Nevertheless, while C-V2X might suffice for rudimentary V2X use cases, its capacity to support more sophisticated applications remains constrained by challenges in maintaining consistently low latencies. Both 4G LTE and early 5G-NR deployments encounter challenges in providing consistent support for advanced V2X applications. The reliability concerns are particularly critical, as they directly affect a vehicle's ability to place trust in the received information. The most recent advancements and novel attributes within cellular networks are showcased in the latest iterations of the 3GPP framework, specifically in Releases 17 and 18. Release 17 in 2022 encompasses a spectrum of enhancements to existing capabilities, including MIMO, Ultra-reliable low-latency communication (URLLC), and power savings. Additionally, it introduces innovative features like Non-terrestrial network (NTN) and higher-frequency bands. In the context of cellular advancements, Release 18 in 2024 emerges as the inaugural iteration of 5G Advanced, targeting novel functionalities such as Artificial intelligence (AI) and Machine learning (ML) integration, extended reality applications, network energy conservation, and further improvements to existing features [27].

The escalating need for high-speed internet access, sought after by both consumer and enterprise segments, is propelling a robust surge in global 5G deployment [9]. Prominent trends encompass the increasing traction of open Radio access networks (RAN), ongoing advancements in 5G mmWave technology, and the steady expansion of consumer-centric mobile services. In this context, the potential of 5G looms large, promising to usher in a new era of advanced V2X applications. Yet, the trajectory for realizing these ambitious applications with 5G remains less clear, necessitating concerted efforts from pertinent industries and research communities to unlock the full potential of this technology for the benefit of the V2X ecosystem. These findings underscore the pressing necessity for further development and implementation of intelligent vehicle systems and the forthcoming 5G networks [28]. By refining the design and enhancing the robustness of both intelligent vehicles and the evolving 5G infrastructure, the aim is to establish a dependable foundation for V2X communications, ultimately fostering the safe and effective functioning of connected vehicles.

As the landscape of RATs continues to evolve within the realm of V2X communication, it becomes increasingly clear that pursuing integration at the access layer may not be the most efficient strategy for ensuring the seamless interoperability of heterogeneous wireless technologies [22, 29, 30]. This is primarily attributed to the requirement for continuous updates whenever novel V2X technologies are embraced, or new iterations of standards are introduced.

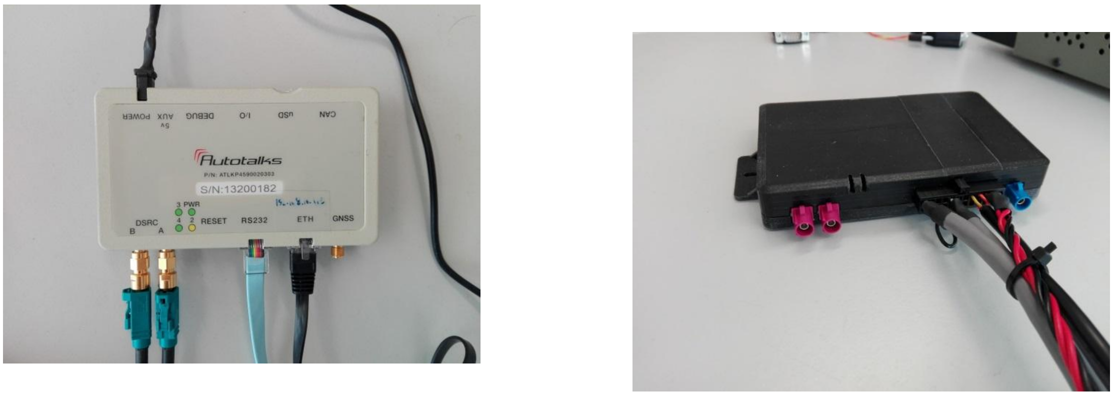
2.4 On-Board Unit

The core concept of cooperative ITS involves equipping individual vehicles with on-board wireless communication devices to enable vehicles to exchange information about themselves and their surrounding environment.

Therefore, within an IoV environment, each vehicle should be able to assume the role of the sender, receiver, and router, distributing information throughout the network. To enable this, vehicles must be equipped with an OBU, designed to facilitate the reception and transmission of messages between vehicles and infrastructure through dedicated RATs. The OBU is equipped with a robust hardware configuration encompassing processing, memory, storage, and communication capabilities, accompanied by a suite of software applications. The OBU's key functions encompass a range of tasks, including wireless radio access, network congestion management, traffic routing, message transmission, data security measures, and adept handling of IP mobility. Furthermore, the OBU computing system offers seamless integration with the Human-Machine Interface (HMI) and establishes a platform for driver interaction.

The rise of V2X applications, including Advanced Driver Assistance System (ADAS) and Connected and Autonomous Driving (CAD), demands the substantial processing and real-time exchange of vast amounts of data from vehicular sensors. Integrated within a vehicle, the OBU functions as a wireless communication device, augmenting the sensor capabilities of vehicles through direct interactions with infrastructure nodes and other OBUs. The OBU gathers environmental data to detect events, driving patterns, and situations. It plays a crucial role in broadcasting critical vehicle information such as location, direction, and speed to designated devices, simultaneously assimilating data from other vehicles. This incoming data serves as valuable input for the OBU's internal algorithms, contributing to accident avoidance strategies. The Communications Control Unit (CCU) embedded within the OBU guarantees connectivity across all supported networks, holding transceiver modules equipped with radio frequency antennas for each communication interface. Modern commercial OBUs now commonly feature a multiple network interfaces (such as DSRC, Wi-Fi, cellular, etc.). Additionally, there is a growing anticipation that radio modules for millimeter wave and visible light communications will likely become accessible within a reasonable timeframe. Therefore, it is important to study the coexistence and interoperability between different architectures in order to ensure a certain level of QoS to a data flow offered by the underlying networks [10, 31, 32]. Moreover, the OBU facilitates seamless integration with various in-vehicle sensors (such as Global Positioning System (GPS), radar, and camera), thus synthesizing data from internal and external communication sources.

The initial iteration of the OBU is presently integrated into vehicles currently in operation, featuring sole DSRC support (known as ITS-G5 in Europe). Consumers have the option to purchase vehicles



(a) The Autotalks PANGAEA V2X Communication Module

(b) The OBU prototype developed by Bosch

Figure 4: Example of 1st generation V2X equipment

pre-equipped with communication hardware to support 'day one' V2X applications. IEEE802.11p-based products are now commercially accessible, with multiple silicon vendors offering them within the market. Illustrated in Figure 4(a) is an example of an integrated OBU produced by *Autotalks*, one of the leading suppliers of automotive-grade V2X communication processors. The Autotalks PANGAEA platform includes an RF transceiver module supporting GNSS and DSRC connectivity, a modem, a security module, and a software stack. Figure 4(b) shows the first V2X board prototype provided by Bosch Car Multimedia with a similar functionality. The OBU device can connect to other vehicles and roadside infrastructure networks using only short-range wireless communication based on IEEE 802.11p radio technology. To gather pertinent sensor data (e.g., speed, acceleration), the V2X board establishes communication with the vehicle's Electronic Control Unit (ECU) via a dedicated connection, such as Controller Area Network (CAN), FlexRay, or BroadR-Reach physical interface.

The absence of regulatory specifications for communicating safety-critical messages led to the divergence of V2X into two primary standards: DSRC and C-V2X, each characterized by distinct architectures. The variation in V2X technology standards presents the task of cost-effectively meeting safety communication requisites. While DSRC chipsets have traditionally been tailored as specialized components for safety-centric direct V2X communication, C-V2X aims to integrate direct V2X connectivity and network connectivity within a unified framework. This divergence complicates the decision-making process for OEMs, who must choose between the two standards, and further challenges the realization of a unified global solution. Automotive OEMs now have the flexibility to select their preferred underlying V2X technology. However, a notable advancement is anticipated with the emergence of new-generation OBUs that can support multiple wireless interfaces, thereby enhancing communication performance. As depicted in Figure 5, Bosch has developed a second-generation OBU prototype, exemplifying an effort to resolve the competition between the two standards. This prototype integrates a mobility-optimized IEEE802.11p, Wi-Fi, Bluetooth, LTE, and C-V2X modems. It offers superior performance, containing a processor capable of running full V2X middleware and applications and optional secure in-vehicle connectivity.

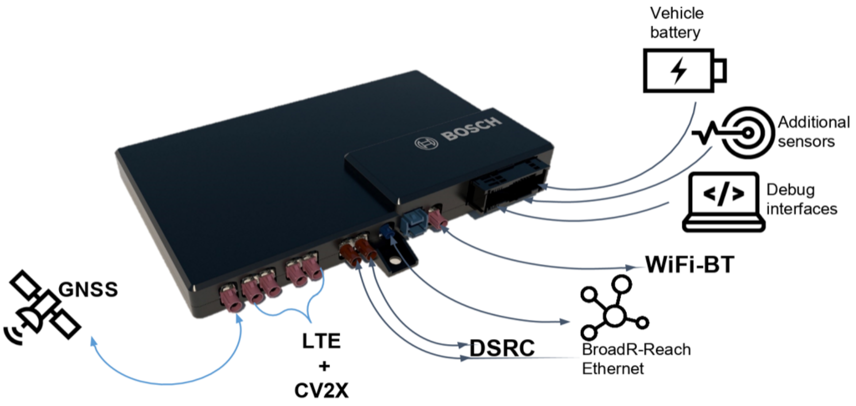


Figure 5: 2nd generation OBU supports multiple RATs.

2.5 Seamless mobility

In the V2X domain, it's hard to keep connectivity uninterrupted because the network environment changes constantly. Therefore, the challenge of Vertical Handover (VHO) is to seamlessly maintain a vehicle user's connection, preferably without any perceivable interruption [33]. Seamless handover serves as the foundational stride when orchestrating the migration of an internet connection, which allows multi-homed OBU to seamlessly migrate active sessions across heterogeneous networks. To ensure seamless mobility within a multi-technology environment, two distinct mechanisms were introduced by IEEE and 3GPP: Media Independent Handover (MIH) [34] and Access Network Discovery and Selection Function (ANDSF)[35, 36] respectively. These mechanisms facilitate smooth vertical handover between various technology types, seamlessly managing IP flow mobility and data offload to optimize network resources utilization and improve the overall network capacity.

The process of seamless handover decision-making plays a pivotal role in upholding uninterrupted services and ensuring the quality of communication for vehicles within regions covered by multiple access networks. Determining the most suitable handover strategy presents a significant challenge, necessitating a delicate balance between meeting the demands of vehicles and optimizing the overall network performance. Multiple approaches and strategies are being explored to tackle this intricate issue.

The VHO procedure can be divided into three phases: information gathering, decision-making, and execution. During the information gathering phase, essential data for the VHO decision is collected. This includes user preferences (like cost and security considerations), network parameters (such as latency and coverage), and terminal details (like battery level and velocity). In the decision-making stage, the optimal RAT is chosen based on the accumulated information, and this decision is then communicated to the handover execution process. The handover execution phase ensures the uninterrupted transition of the active session over the new RAT. Once the transition is completed, the resources of the previous RAT are eventually released.

In this scenario, it's imperative that each vehicle maintains a continuous connection to at least one network. Through the use of soft handover, vehicles also have the capability to be concurrently connected to multiple networks, achieved by employing multiple RATs devices installed on each vehicle. As a result, communication with the remote node can persist uninterrupted, even in the presence of potential changes in the physical interface states.

2.6 RAT interoperability

The traditional V2X data communication paradigm cannot fully exploit the potential of these multiple networks simultaneously. In this context, the data transmission is typically restricted to a single path, even when multiple available paths[1] exist between the source and destination. As a result, the traditional approach limits the overall communication efficiency, especially in dynamic vehicular environments where connectivity may frequently change due to vehicle mobility and the availability of different network technologies. The inability to simultaneously utilize multiple networks leads to suboptimal data transmission rates, higher latency, and potentially increased packet losses, especially during handover events. To address these limitations, a novel approaches and protocols were proposed to take full advantage of the multiple available heterogeneous networks.

Most end-user mobile devices now come equipped with multiple network interfaces. The next generation of V2X communication devices will support *multi-RAT* connectivity, a fundamental feature of next-generation vehicular networks [37], enabling the simultaneous utilization of multiple available heterogeneous interfaces, such as Wi-Fi and LTE. Several interworking mechanisms for combining heterogeneous networks into integrated wireless environments have been proposed. This multi-technological environment shows great potential for enhancing vehicular communication, as it exploits the advantages of diverse RATs and may significantly improve overall communication efficiency. By seamlessly integrating different wireless technologies, multi-connectivity empowers vehicles to maintain continuous and reliable connectivity, even in highly dynamic vehicular environments. However, despite the promising potential of multipath connectivity, there are still uncertainties surrounding its practical implementation [38]. For example, multi-connectivity in 5G introduces complex challenges that must be addressed to realize its benefits for vehicular communication and other applications fully.

One of the key challenges is efficiently managing and coordinating multiple available interfaces in real time. The seamless integration of different RATs requires sophisticated mechanisms to handle handovers, path selection, and load balancing effectively. Additionally, ensuring smooth and uninterrupted communication between the various interfaces demands robust synchronization. Furthermore, the coexistence of diverse wireless technologies within the same network introduces interoperability issues. Different RATs may have varying protocols, data formats, and signaling mechanisms, making it essential to develop

---

[1] In this context, the available RATs are often referred to as paths

standardized communication protocols and interfaces to ensure seamless compatibility. Moreover, implementing multipath connectivity in vehicular networks requires careful consideration of security and privacy concerns. Using multi-RAT parading opens up new attack vectors and potential vulnerabilities, necessitating robust security measures to safeguard communication and protect user data. Ongoing research and collaboration between academia, industry, and standardization bodies are crucial to address these challenges. Innovative solutions and protocols must be developed as heterogeneous vehicular networks evolve to enable efficient multi-RAT connectivity.

2.6.1 Classification of handover mechanisms

After an in-depth analysis of existing handover types in the literature, we have identified a multitude of approaches. This diversity can be attributed to the various steps involved in the handover process, including Initiation, Decision, and Execution. While certain standards cover the Execution step, the Decision step, particularly, lacks standardization. Hence, multiple approaches are found in the literature. We propose classifying handover mechanisms based on several criteria, including:

1. Technologies Used: This criterion considers the types of technologies involved in the handover.
2. Execution Method: It involves how the handover is carried out, whether through hard or soft handovers.
3. Decision Actor: Identifies whether the handover is initiated by the mobile node, the network, or a combination of both.
4. Number of Layers Involved: Reflects the involvement of OSI model layers, including mono-layer and cross-layer handovers.
5. Deployment Architecture: Considers the architecture type, including centralized, fully distributed, partial distributed mobility management and cloud-based solutions.

This classification framework helps organize and understand the diversity of handover mechanisms in vehicular networks. Therefore, these existing handover mechanisms can be described as follows:

- Horizontal Handover: Occurs within the same network technology type (intra-system).
- Vertical Handover: Involves different network technologies (inter-system).
- Hard Handover: Requires the mobile to disconnect from the current network before connecting to another.
- Soft Handover: Permits the mobile node to be connected to multiple networks during the handover process.

- Mobile Node-based Handover: Initiated by the mobile node.
- Network-based Handover: Core network decides on the handover, excluding the mobile node from the process.
- Hybrid Handover: A combination of both MN-based and network-based handovers.
- Mono-layer Handover: Involves a single OSI model layer, for example the MAC-layer.
- Cross-layer Handover: Encompasses interactions between different OSI model layers, such as the link layer (L2), network layer (L3), and transport layer (L4).
- Architectural Types:
  - Centralized Solutions: Control and data plans are centralized, with all packets routed through one entity.
  - Fully Distributed: Both control and data plans are distributed.
  - Partial Distributed: Combines centralized control plan with distributed data plan, as seen in Software-Defined Networking (SDN). Also including, emerging solutions based on cloud computing.

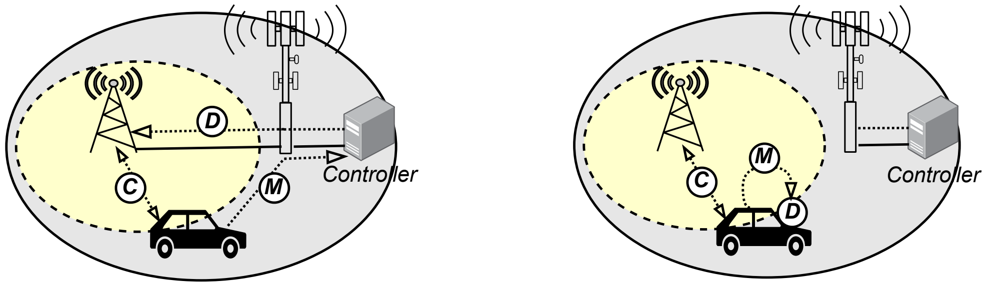
The central and recurring challenge continues to revolve around determining the optimal moment for initiating a handover in a manner that minimizes both the handover latency and its associated effects, particularly with regard to packet loss and the ping-pong effect. This predicament arises from the inevitable disconnection time that precedes connection to the subsequent network. This disconnection time predominantly consists of three components: the disconnection detection time, the time required for configuring a new address, and the duration spent on exchanging handover processing messages.

2.6.2 Centralized, decentralized and hybrid approaches

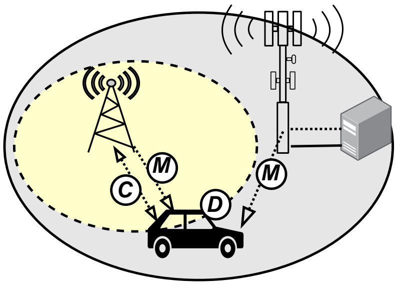
The standardization activities have been exploring different strategies for enabling multi-RAT integration. Many handover mechanisms, also known as mobility management, were proposed in the literature. The current solutions for multi-connectivity can be broadly categorized into three alternatives: centralized, distributed, and hybrid approaches [39]. Figure 6 illustrates different RAT interoperability schemes with one vehicle and two heterogeneous networks.

2.6.2.1 Centralized approach

In the centralized approach, a particular vehicular user is assigned to the most suitable access technology based on the global view of the network status and application requirements. The network selection and vertical handover procedure is assured by a central network entity, which controls the assignment of radio



(a) Centralized (b) Distributed



(c) Hybrid

Figure 6: Representation of centralized (a), distributed (b), and hybrid (c) RAT selection schemes.

Small-dashed lines represent the transmission of information during VHO process. Label D stand for network selection decision. Label C indicate the establishment of data link and label M represents the transmission of measurements reports.

resources for vehicular communication. In a highly dynamic vehicular environment, vehicles frequently switch between networks due to rapid movement, ensuring continuous connectivity to the most suitable network becomes essential for reliable and efficient vehicular communication.

However, traditional centralized handover mechanisms designed for cellular networks are not well-suited for the hybrid vehicular architecture [40]. Despite the inherent challenges in making accurate predictions, many existing centralized solutions continue to rely solely on Receiving Signal Strength Indication (RSSI) measurements before initiating a handover. This approach inevitably leads to a prolonged disconnection period, often referred to as handover latency. Such latency can have detrimental effects on application performance, particularly in terms of packet loss. This is especially concerning for applications with stringent requirements, such as those related to safety and real-time functionality.

In the centralized approach, the primary mobility manager, often referred to as the Home Agent (HA), is situated within the core network infrastructure. Mobility is administered in a centralized manner from this location. Some well-known examples of global mobility management protocols include Mobile IPv4 (MIPv4), Mobile IPv6 (MIPv6), Network Mobility (NEMO), and others. In these centralized solutions, a single misbehaving host has the potential to disrupt internet service for numerous other hosts. Global mobility approaches continues to rely on a centralized architecture, which is susceptible to the drawbacks associated with centralized systems, including increased overhead, bottlenecks, a single point of failure, and limited scalability. Consequently, it may result in considerable handover latency due to the exchange of control messages among the entities involved.

Additionally, the handover frequency for vehicular users is higher than that of cellular users, resulting in excessive signaling overhead. Furthermore, solutions deployed at the radio layer could pose challenges for the interoperability of heterogeneous networks due to the heterogeneity of resource allocation and modulation schemes across RATs. To address this issue, the Access Traffic Steering, Switching, and Splitting (ATSSS) [41] architecture was introduced in 2020 by 3GPP(Rel-16) to exploit transport layer multi-connectivity between heterogeneous networks. ATSSS leverages the capability of the 5G Core to explicitly handle non-3GPP traffic through the Non-3GPP network. It uses a Multi-Access Protocol Data Unit (MAPDU) that is exchanged over multiple RATs, providing functionalities for selection (steering and switching) and aggregation (splitting) at both the lower and upper layers of the IP stack.

The centralized control characteristic of SDN is also widely explored in the literature to offer seamless handover solutions for mobile users [42–45]. SDN allows achieving network programmability and flexibility, making it an essential tool for managing mobile devices and resources [46–48]. By separating the control and data planes, SDN allows centralized management and configuration of network devices, simplifying network operations and improving resource allocation. However, a distributed RAT selection algorithm could be undesirable in vehicular communication for two primary reasons. Firstly, it creates high signaling and message-passing overhead in the operator's network infrastructure. Secondly, it deteriorates the user experience by causing constant fluctuations in client throughput [49]. Centralized schemes based on SDN rely on dedicated network controllers that require periodic reporting from users regarding their coverage status and experienced performance to make tailored decisions. The ongoing network densification challenges this approach, calling for more scalable solutions.

2.6.2.2 Hybrid or cloud-assisted approach

In the hybrid approach, multi-RAT coordination involves establishing a functional relationship between different entities in the system. For instance, infrastructure nodes can share general status indications, such as network congestion levels, with vehicles. Based on this information and their own measurements, vehicles decide which RATs to connect, while the infrastructure node controls the vertical handover process. Alternatively, vehicles can send observed metrics to a cloud-based central server, which makes network selection decisions based on a global view. The server then instructs the vehicle to perform the handover, although the vehicle controls the process. This approach allows vehicles to opportunistically report their positions and channel load measurements to the central server when they have access to network infrastructure (e.g., roadside units). By analyzing wide-scale statistics on road and network conditions, the central server derives recommended network selection strategies distributed to individual connected vehicles to allow them for globally optimized resource utilization.

While the hybrid and centralized approaches provide tailored decisions based on global information, which could often in higher system performance, it also comes with several challenges. One of the main problems with a hybrid solution is scalability. The exponential growth of heterogeneous access nodes and user devices presents a significant challenge in solving the network selection problem through centralized controllers. This is primarily due to the high computational complexity required to find a system-level optimal solution and the substantial increase in signaling messages exchanged between the controllers and the connected devices. As the network becomes more complex and dynamic, the limitations of centralized solutions in handling the large-scale and rapidly changing environment become evident. As the number of vehicles and the complexity of the network increase, the centralized controller can become overwhelmed with the amount of data it needs to process and the number of decisions it needs to make. This can lead to delays in decision-making and may result in suboptimal network selections for some vehicles. Moreover, centralized solutions rely on continuous vehicle reports about their coverage status and experienced performance. This frequent reporting can create significant signaling overhead in the network, consuming valuable resources and potentially causing congestion issues. Additionally, the reliance on a single controller introduces a single point of failure. If the cloud-based central server experiences problems or malfunctions, it can lead to widespread network disruptions and service outages for all connected vehicles. Another concern is the complexity and cost of deploying and maintaining a centralized controller. As the network grows in size and complexity, the resources required to operate and manage the controller can become substantial.

2.6.2.3 Decentralized approach

The increased computational capabilities of the vehicle's OBU devices enable them to autonomously make context-aware decisions by observing and adapting to the surrounding environment. This aligns with the recent trend of decentralizing network functionalities and decisions, as evident in edge and fog networking [50]. The shift towards decentralization allows for more efficient and agile decision-making at the user end, contributing to improved overall network performance and responsiveness. Vehicle users independently select the optimal RATs based on their measurements, representing a decentralized approach. This empowers the vehicle users to make network selection decisions autonomously, utilizing local perceptions and controlling the handover procedure. As a result, RAT interoperability is achieved in a fully independent (distributed) manner, with vehicle users collecting necessary information and making decisions independently. The more advanced algorithms could also leverage shared data from other vehicles or road infrastructure, such as network congestion and traffic load, to make more informed decisions.

Local mobility management works in a small geographic area called a mobility domain, and it follows a distributed management approach. Instead of centralizing mobility management, it is handled within these smaller domains. Notable examples of local mobility management protocols encompass Proxy Mobile IPv6 (PMIPv6), Distributed Mobility Management (DMM), and Fully DMM (FDMM).

The distributed approach is highly scalable compared to centralized solutions, where user-specific information is continuously reported to the network. It reduces the need to constantly register user-specific data to the network side, minimizing signaling overhead and computational complexity. The distributed approach gives each vehicle user autonomy in making network selection decisions. They can choose the most suitable network based on their observations and measurements, allowing for personalized connectivity decisions. Utilizing the distributed approach, vehicle users can rapidly adapt to changing network conditions and real-time context, as they are directly involved in the decision-making process. This enables adaptability to varying traffic and network load conditions, demanding a fast and responsive multi-RAT solution.

While offering autonomy to individual vehicles, the distributed multi-RAT solution may face challenges in evaluating the global network status, potentially leading to non-optimal decisions. Indeed, the distributed scheme can also use additional information shared by other vehicles or road infrastructure. However, ensuring the accuracy and trustworthiness of this information is crucial. Incorrect or misleading data can lead to suboptimal network selection decisions and negatively impact overall network performance. Also, the excessive data exchange can result in increased communication overhead, leading to higher data traffic and potential delays in decision-making. Nevertheless, incorporating diverse data sources empowers vehicles to consider factors beyond their immediate surroundings, enabling them to gain access to a more comprehensive view of the network environment. As a result, vehicles can make more context-aware network selection decisions, providing a satisfactory Quality of Experience (QoS) even under high mobility constraints.

2.6.3 Multipath connectivity

Several studies have proposed a tight integration of different wireless technologies to increase communication efficiency in heterogeneous vehicular networks [51–54], where each system offers unique benefits. However, various problems must be addressed to make different technologies work together under a highly dynamic vehicular environment [55]. Coordinating multiple RATs to select the network with the best connectivity while considering application requirements is challenging. The concept of multipath connectivity has emerged as a promising approach to enhance network service for high-speed vehicles, leveraging multiple interfaces to exploit available wireless technologies and maintain continuous connections [56–60]. Traditional VHO schemes are limited in that they permit the connection to only one access technology at any given moment. Diverging from the conventional sequential utilization of multiple RATs observed in handover scenarios, which has been extensively explored in heterogeneous wireless networks, the heterogeneous multipath V2X hinges on the concurrent utilization of distinct technologies. This approach entails the adoption of aggregation schemes that empower users to tackle the capabilities of multiple available networks simultaneously and in parallel. The advantage of being simultaneously connected to two or more networks can be leveraged to enhance packet reception reliability in the event of one interface going offline. This can be accomplished by ensure packet redundancy following the detection of a disconnection in one of the physical interfaces. The redundancy mechanism involves duplicating received messages across the available physical interfaces. When implemented, this feature has the potential to significantly improve both packet delivery ratios and overall throughput.

In recent years, there has been a growing interest in the use of multipath communication for vehicular networks [55, 61, 62]. This approach involves leveraging multiple RATs to enable reliable and uninterrupted vehicular communication, even in challenging environments [52]. For example, the redundant use of multiple access technologies was considered in [63] to improve the communication reliability of vehicular applications. The paper demonstrates the potential benefits of using various RATs in a highway platoon scenario. In [64], a hybrid vehicular network architecture and protocol stack was proposed, which combines DSRC and C-V2X technologies. The paper addresses the problem of radio resource management, adaptive RAT selection, and VHO algorithm in a highway environment. The effectiveness of the proposed architecture is shown using several simulations with different parameter settings. A multi-radio access technologies scheme for V2X communication have been introduced in [51], where two other radio access technologies (i.e., LTE-Uu and PC5) are combined to improve a system performance and meet high-reliability requirement [65].

Furthermore, authors in [5] have highlighted the potential benefits of employing a logical interface in the context of multihoming in VANETS, particularly when connected to two or more distinct networks. A key advantage of a logical interface lies in its capacity to be associated with multiple physical interfaces, thereby enhancing reliability and throughput between directly connected devices by establishing alternative physical pathways. Thus, the logical interface acts as a gateway/proxy of the physical interfaces. This can be accomplished through a process referred to as “link-layer multiplexing”, which hides the presence of multiple physical interfaces from the protocols operating above the link layer. Notably, it is imperative to note that the responsibility for multiplexing and routing packets across these physical interfaces lies with the link-layer device driver, although the specific mechanisms employed for this purpose are not explicitly detailed.

However, the performance of multipath communication in highly dynamic environments remains a challenge [66]. The frequent handover events caused by vehicle mobility lead to significant service interruptions, limiting the effectiveness of the current multipath solutions in vehicular scenarios. An alternative strategy involves the implementation of interoperability[2] mechanisms at the transport layer, leveraging common protocols among different RATs to address the issues associated with radio layer-based solutions. Integrating multipath functionality at the transport layer augments the potential of vehicular communication devices to effectively utilize multiple available paths, thereby enhancing data transmission efficiency. Using standardized multipath extensions, such as Multipath TCP (MPTCP) [67], vehicles can seamlessly switch between RATs, ensuring uninterrupted connectivity within dynamically changing environments. A discussion on the use of MPTCP in vehicular environments has become an urgent issue [68–71]. So far, research has focused primarily on a complementary combination of WiFi-based IEEE 802.11p and cellular communications based on 3G/4G technology. The 3GPP has recently proposed multipath transport protocols for 5G networks to improve network Interoperability [41, 72]. As a result, the ATSSS architecture was introduced to exploit transport layer multi-connectivity between 3GPP and non-3GPP networks. The recently introduced Path Aware Networking Research Group [73] aims to support research in bringing

---

[2] In the following, we refer to selection and aggregation schemes as RAT interoperability procedures.

path awareness to transport and application layer protocols. It exploits interoperability mechanisms in order to combine different networks, discover information about the properties of a path, and design new algorithms for path selection and scheduling decisions based on this information.

Authors in [57] conducted a survey on the technical challenges and reviewed the existing studies regarding MPTCP, and the implementation of MPTCP in vehicular networks. In [71], the MPTCP was used to seamlessly switch between 802.11 and satellite connections for non-safety-related vehicular infotainment applications. A performance evaluation demonstrates the good performance of the proposed method in a vehicular scenario in which both interfaces transmit video data. The combination of MPTCP and SDN to provide a seamless V2I connectivity was considered in [2]. The authors conducted experiments to evaluate the default performance of MPTCP in heterogeneous networks that employed both Wi-Fi and DSRC. They also presented an approach for installing rules to establish V2I connectivity in software-defined vehicular networks with centralized control. The problem of ultra-reliable low-latency communication for Connected and Autonomous Vehicle (CAV) has been addressed in [56]. A novel framework based on MPTCP was introduced, which employs coding techniques and efficiently distributes packets across multiple wireless network links that may dynamically change. Based on mathematical modeling and performance evaluations, it was demonstrated that the proposed approach effectively optimizes resource utilization and enhances throughput. Authors in [68] proposed a mobility-aware multimedia data transfer mechanism using MPTCP in vehicular networks. MPTCP was adopted for aggregating bandwidth and improving the transmission rate for video streaming services. To measure the quality of subflows and dynamically allocate data to different paths, a quality-aware data distribution technique is utilized. Furthermore, the VHO mechanism was tested for continuous data transmission. The simulation results indicate that the mobility-aware data transmission mechanism effectively enhances throughput while reducing transmission delays. A novel MPTCP scheduling scheme that dynamically enables packet duplication across parallel paths has been recently introduced in [59]. The proposed method can adaptively turn to redundant scheduling in the presence of Head-of-line (HoL) blocking to achieve the best bandwidth performance for high-speed vehicles. Experimental results show that the proposed solution can improve bandwidth aggregation under fast mobility scenarios.

The newly developed MPQUIC [74, 75] stands as a multipath extension to Quick UDP Internet Connections (QUIC), akin to MPTCP, enabling the utilization of multiple links within a single connection. This protocol is based on QUIC's stream-multiplexing capability, adeptly managing concurrent request/response pairs on a single connection using numerous streams [76, 77]. Running on top of UDP, MPQUIC lends itself to user space implementations, facilitating compatibility with unaltered operating systems. As a result, the integration of MPQUIC with existing QUIC implementations is seamless. While it has not yet been specifically evaluated within vehicular scenarios and currently rests outside active development, MPQUIC offers a solid foundation for a potential protocol catering to the demands of V2N scenarios [78].

2.7 Multipath TCP protocol overview

MPTCP is a well-known solution that establishes an effective end-to-end connection through an array of available paths. This technology encompasses a series of TCP extensions, which have undergone standardization within the Internet Engineering Task Force (IETF). MPTCP empowers applications to transmit data across multiple IP addresses and interfaces while operating transparently. This seamless integration of MPTCP into applications offers a standard socket Application Programming Interface (API) in userspace, ensuring that applications remain oblivious to the underlying implementation of multipath communication. MPTCP does not require additional entities in the network to be deployed (e.g., network proxies); only the endpoints need to support MPTCP to guarantee multipath communication between them. Hence, the implementation of the vehicular multipath communication system has become relatively straightforward, without the necessity to establish additional network nodes.

MPTCP can establish multiple TCP connections, referred to as *subflows*, and seamlessly transmit a unified data stream. Thus, it allows the transparent handover execution and the session continuity maintenance. It is considered a decent solution for future 5G mobile networks, aiming to improve user experience with higher throughput and seamless connections [72, 79, 80]. Furthermore, MPTCP is compatible with deployed networks, as each subflow is treated by middleboxes as a standard TCP session, thus avoiding any filtering issues on traditional network infrastructures. Hence, MPTCP requires support only from the end devices, namely the host and server, in order to function. The protocol utilizes the Data Sequence Number (DSN) to indicate the sequence of all packets within an MPTCP stream, ensuring dependable and sequential data delivery across multiple dynamic paths. The capability of MPTCP to concurrently utilize multiple interfaces enables applications to remain active during VHO, whereas traditional TCP connections would be disrupted. Furthermore, MPTCP provides the flexibility to initiate, terminate, or adjust the priority of individual subflows at any juncture within an ongoing session. MPTCP has been incorporated into the Linux Kernel and adopts a modular framework comprising *path manager*, *congestion control*, and *packet scheduler* strategies (Figure 7).

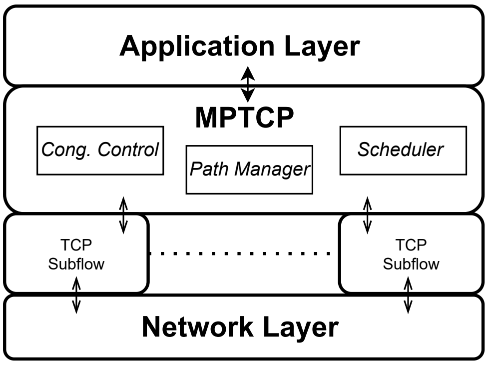


Figure 7: Multipath TCP in network stack.

2.7.1 Packet scheduler

In modern computer networks, effective data transmission management is essential for ensuring efficient, low-latency communication. Packet scheduling is a fundamental mechanism that plays a crucial role in managing the transmission of data packets within a network. In MPTCP a packet scheduler is a component responsible for prioritizing and regulating the transmission of data packets over a network. It plays a critical role in various network scenarios for optimizing data transmission. The primary purpose of a packet scheduler is to determine the order in which packets are sent, ensuring that network resources are used efficiently.

Packet schedulers perform several key functions to ensure that data is transmitted effectively and that network resources are utilized optimally. It assign priority levels to data packets based on link status. For instance, the default scheduler of the Linux MPTCP implementation schedules packets to the subflow with the lowest Round-trip time (RTT) and space in its Congestion Window (CWND). Unfortunately, it don't consider other important criteria such as the type of application (e.g., voice, video, or data), service-level agreements, and QoS parameters. More advanced packet scheduler methods presented in the literature could allocate network resources in a way that guarantees a specific level of performance for high-priority applications [81, 82]. For example, in real-time applications like Voice over IP (VoIP) and video conferencing, minimizing latency is essential. Packet schedulers could ensure that voice and video packets are prioritized, resulting in low latency and smooth communication.

MPTCP packet scheduler play a role in traffic shaping by regulating the flow of packets to match available bandwidth. They prevent network congestion by managing the rate at which packets are transmitted. Fair access to network resources is a core principle of packet scheduling. Packet schedulers prevent any single user or application from monopolizing network resources, ensuring that all users receive equitable access. In environments with limited bandwidth, packet schedulers perform traffic management by ensuring that resources are allocated fairly and that traffic is shaped to prevent congestion. MPTCP packet scheduler is capable of adapting to changing network conditions. For instance, in scenarios where link speeds vary or there is an increased packet loss rate, an adaptive packet scheduler can adjust its scheduling strategy in real-time.

The path scheduling policy is formulated to intelligently distribute data packets across multiple paths, ultimately enhancing MPTCP's performance. In scenarios where multiple network paths or links are available, the packet scheduler becomes crucial in determining the allocation of data segments across various subflows. It can balance the load across these paths, optimizing resource utilization. Various scheduling strategies have been developed to date, including the Lowest-RTT-First (minRTT) scheduler, Constraint-based proactive scheduling (CP), Highest Sending Rate (HSR), Largest Window Space (LWS), Lowest Time/Space (LTS) and others [83, 84]. These algorithms play a critical role in optimizing path selection and data transmission in the MPTCP framework. The standard minRTT scheduler, for instance, directs data through the subflow with the lowest RTT and a non-full congestion window. In cases where a subflow experiences failure during the connection, the scheduler retransmits lost data across accessible subflows.

2.7.2 Congestion Control

While the concept of MPTCP offers several advantages in terms of reliability and performance, it also introduces unique challenges related to congestion control. Traditional TCP, as specified in RFC 5681, is based on a single path assumption, meaning that it expects all traffic to traverse a single network path. In contrast, MPTCP leverages multiple paths, which introduces complexities in managing congestion. The primary challenge is to ensure fair utilization of network resources, prevent congestion collapse, and maintain QoS standards.

MPTCP extends congestion control mechanisms to operate at the subflow level. i.e., an individual path in an MPTCP connection. This enables more fine-grained control over each subflow, ensuring that no single path monopolizes network resources. Furthermore, the coupled congestion control in MPTCP allows subflows to share congestion information. If one subflow experiences congestion, it can inform other subflows about the congestion state, enabling them to react accordingly. This coupling is vital in maintaining overall fairness and avoiding congestion collapse. Each subflow estimates its sending rate based on congestion signals and network conditions. The sender adjusts its rate to avoid congestion and ensure efficient resource utilization. This dynamic adaptation ensures continuous data transmission even when individual subflows are disrupted. Additionally, MPTCP congestion control is designed to be compatible with traditional TCP congestion control mechanisms. When MPTCP communicates with single-path TCP flows, it responds to congestion signals and operates effectively within the traditional TCP ecosystem.

The congestion control algorithms implemented in MPTCP primarily aim to regulate the transmission rate of each subflow, redistributing traffic from more congested paths to less congested ones. This approach enhances throughput and optimizes link utilization. Several congestion control algorithms have been developed, including the Linked Increases Algorithm (LIA), Opportunistic Linked-Increases Algorithm (OLIA), Balanced Linked Adaptation (Balia) and Weighted Vegas (wVegas). These algorithms employ arithmetic models to control the increase of each subflow's congestion window (cwnd), ensuring a balanced distribution of congestion levels among them. Well designed congestion control mechanisms aim to achieve fairness across subflows, ensuring that each subflow receives an equitable share of network resources. This is vital for preventing congestion collapse and maintaining a quality user experience. Numerous congestion control algorithms have been proposed to guarantee fair and equitable treatment among various network users [85–87].

2.7.3 Path Manager

The MPTCP path manager plays a crucial role in managing the communication between a device and multiple networks. The path manager handles the setup and termination of network connections. It initiates connections with available networks, allocates resources, and manages the teardown of connections when they are no longer needed. This dynamic path management ensures efficient resource utilization. Its functionality is essential for optimizing data transmission, improving reliability, and enhancing the overall quality of service.

While the MPTCP framework allows for a modular architecture, there are two primary path managers: the full-mesh and the ndiffports path managers. Each serves specific use cases and can be configured to optimize MPTCP connections. The full-mesh path manager is the default option and is recommended for most deployments. Its design promotes adaptability and efficiency in establishing MPTCP connections. On a client, the full-mesh path manager advertises all of the client's IP addresses to the server. Simultaneously, it listens to all IP addresses announced by the server. On a server, the full-mesh path manager enables the server to automatically discover all available addresses and announce them to the client. One important note is that if the client has N addresses and the server has M addresses, the full-mesh path manager will establish N x M subflows. While this configuration is suitable for many scenarios, it might not be optimal in all situations.

MPTCP regulates subflow utilization via the Path Manager (PM) module, allowing users to configure distinct operational modes for each MPTCP endpoint [88, 89]. The default full-mesh PM attempts to establish subflows between all combinations of IP addresses of the client and remote host. Consequently, subflows are promptly established upon connection initiation and dismantled upon closure of the MPTCP session. This configuration might fail to address the demands of exceptionally dynamic vehicular network environments.

One of the primary functions of the MPTCP path manager is to select the most suitable network path for data transmission. Hence, we believe that further enhancements to the path manager are necessary to cater to the specific requirements of vehicular communications. It should considers various factors such as network quality, available bandwidth, latency, and reliability to make an informed decision.

The path manager could evenly be used for distributing data traffic across multiple network paths. By intelligently balancing the load, the path manager could maximizes the utilization of available resources. By selecting the best path, it ensures that data is transmitted through the most efficient route, optimizing the user experience.

In the event of network failures or deteriorating network conditions, the path manager is responsible for initiating handover procedures. It should detect when a network path becomes unreliable or unusable and seamlessly switches to an alternative path to maintain continuous data transmission. If a subflow becomes congested or degrades in quality, the path manager can initiate the creation of new subflows over alternative paths, effectively load balancing the traffic. The path manager may prioritize certain types of data packets based on predefined rules or QoS requirements. This ensures that critical data, such as real-time communication or emergency services, receives preferential treatment over less time-sensitive traffic. To make informed decisions, the path manager should continuously monitors the performance of each network path, and assesses metrics like packet loss, jitter, and available bandwidth to monitor the status of the connections. This real-time monitoring enables the path manager to respond to changing network conditions. By making intelligent decisions based on network performance, load, and reliability, it seeks to provide a seamless and high-quality data transmission experience in dynamic vehicular network environments.

2.7.4 The *upstream* and *out-of-tree* MPTCP implementation

MPTCP has gone through different versions, with MPTCPv0 and MPTCPv1 representing two significant revisions. MPTCPv0, defined in RFC 6824 [90], was the first experimental protocol version. It introduced the concept of subflows, that make up the multipath communication, allowed devices to use links simultaneously for data transmission. However it was developed "out-of-tree" with different requirements in mind. Out-of-tree implementations are not part of the official (mainline) Linux codebase. They are maintained separately from the primary development efforts of the Linux kernel. Therfore, the kernel should be patched to make MPTCPv0 functionality available. Thus, it was not widely adopted in practice due to limitations in its design and implementation. However, out-of-tree MPTCP implementations also includes experimental MPTCP futures that are not part of the official Linux kernel source code. Therefore, it is widely used for research, specific projects, or scenarios where the mainline MPTCP implementation does not suffice.

The MPTCPv1 protocol is defined in RFC 8684 [91]. It introduced significant improvements over MPTCPv0, including enhanced congestion control algorithms and better handling of subflows. MPTCPv1 includes a standardized congestion control framework that allows subflows to share network resources more fairly. It provides better support for path management and handover, making it more suitable for real-world scenarios. It is designed to be backward compatible with single-path TCP for improved adoption. MPTCPv1 addresses some of the limitations of MPTCPv0 and is nowadays recommended for most practical use cases. The Linux community is responsible for the development and upkeep of the MPTCPv1 stack integrated into the Linux kernel (version 5.6 and later), along with the associated userspace tools and libraries. The new protocol version encompasses distinct design decisions and fewer alterations to the original TCP stack compared to its predecessor. It is seamlessly integrated into the upstream kernel, and recent Linux distributions have already incorporated MPTCP support, obviating the need for system modifications to install and utilize alternative kernels. To enable an application to use MPTCP, it needs to create sockets with the $IPPROTO\_MPTCP$ protocol. This can be achieved by calling the socket() system call like this:

```
socket(AF_INET, SOCK_STREAM, IPPROTO_MPTCP);
```

A program opts for MPTCP when creating a socket, utilizing a system call similar to what's used for TCP. The program subsequently uses the returned file descriptor in conjunction with standard socket-oriented system calls to handle connections (e.g., connect(), bind(), listen(), accept()) as well as send and receive functions. The MPTCP socket also support many options that are commonly used with TCP sockets. Legacy applications can be directed to create and utilize MPTCP sockets instead of traditional TCP ones by using several workarounds accessible for this purpose. The code below is a modified socket creation function that appears to be changing the protocol to accommodate MPTCP. This code intercepts calls to the socket function and modifies the protocol to force the MPTCP usage instead of TCP, which enables multipath capabilities on existing legacy services.

```
int socket(int family, int type, int protocol) {
    if ((family == AF_INET || family == AF_INET6) &&
                        (type & 0xff) == SOCK_STREAM) {
        if (protocol == 0 || protocol == IPPROTO_TCP) {
            protocol = IPPROTO_TCP + 0x100;
        }
    }
    return syscall(__NR_socket, family, type, protocol);
}
```

This work relied on the ”upstream” implementation of MPTCPv1. MPTCP has been included to the upstream Linux kernel v5.6 during the first half of 2020. This community development effort has involved contributors from Red Hat, Tessares, Apple, and Intel [92]. Upstream implementations are considered stable and well-supported by the developer community, and often receive regular updates and improvements. It is subject to the project's release cycles and quality control procedures. However, it's important to note that this implementation is not compatible with MPTCPv0. The out-of-tree implementation's design, which was originally intended to enhance multipath functionality for every TCP connection, doesn't align well with the more generic upstream Linux kernel. This design significantly modify core TCP code and data structures. In contrast, an upstream design seamlessly integrates with the existing TCP stack, ensuring there are no performance drawbacks. It's also designed to be easy to maintain, configure, and use in a variety of deployment scenarios.

By default, each MPTCP connection functions with a single subflow, similar to plain TCP. Proper configuration of both the client and server is needed, potentially involving tools like *iproute2* to enable the utilization of multiple application interfaces. An option exists to set a higher threshold for the maximum allowable number of subflows per connection. Furthermore, there's the capability to designate MPTCP endpoints (IP addresses) for announcement or use in additional subflows. The ongoing development work on the *upstream* MPTCP implementation is still in progress, incorporating new features [93]. Currently, the upstream implementation only has one packet scheduler model, which employs a simplified version of the BLEST algorithm [94].

From a security perspective, MPTCP is well-suited for environments where TCP is already an established protocol, providing enhanced resilience and security measures. An MPTCP connection comprises two or more individual TCP connections, with each TCP connection maintaining the same level of security. While employing multiple TCP connections increases the potential attack vectors from one to two, it substantially reduces the likelihood of an attacker gaining access to the entire data exchange. MPTCP's design explicitly prevents attackers from infiltrating an ongoing MPTCP connection. It is advisable to enforce the use of TLS (Transport Layer Security) with TCP, a recommendation that extends to MPTCP.

However, the MPTCP protocol's current implementation overlooks user mobility and the dynamic attributes of wireless links within vehicular networks. Instead, it relies on transport layer information to approximate channel capacity and latency, which proves inadequate in capturing the fluctuating nature of vehicular communication. As a result, the current MPTCP protocol may exhibit slow link failure detection, especially in cases where the communication path is suddenly disrupted during handover events, resulting in suboptimal communication performance. To address this constraint, our research is dedicated to formulating mechanisms that tailor MPTCP to the dynamic nature of vehicular networks, thereby ensuring faster and more precise link failure detection and proficient path selection during mobility. We aim to deliver a seamless and continuous connectivity experience, enhancing reliability and minimizing service interruptions, elevating the overall user experience within vehicular communication contexts. Our goal is to contribute to developing multipath solutions tailored to address the distinct requirements of high-speed vehicles.

2.8 Introduction to network selection

The advanced computational capabilities of modern vehicle OBUs facilitate the implementation of a distributed VHO procedure. This empowers each connected vehicle to autonomously select the most suitable underlying network based on its individual requirements. This decision-making process operates without coordination between various systems or involvement of the network core in policy decisions [95]. The VHO mechanism must ensure seamless and reliable connectivity at all times and locations. The network selection algorithm plays a crucial role in this process, as it needs to consider multiple factors such as signal strength, application requirements, available bandwidth, vehicle position, and speed. Balancing these factors and making intelligent decisions regarding network handover can significantly impact the overall performance and user experience.

Consequently, selecting the most suitable network for multi-RAT terminals presents a complex optimization problem. Numerous works have explored the network selection process, leveraging various techniques like fuzzy logic [96–98], multiple-attribute decision-making [50, 99–104], Markov chain [105], ML, and game theory [27, 40, 49, 106, 107]. These approaches incorporate inputs from diverse information sources, including networks, mobile devices, and user preferences. ML, a prominent subfield of AI, has emerged as a promising research direction to tackle the network selection problem in vehicular networks by using a data-driven approach [108, 109]. ML offers a flexible toolkit to leverage non-linear parameters and support accurate decision-making in a variety of applications. Typically, ML involves two stages: training and testing. During the training stage, a model is developed using training data, while during the testing stage, the trained model is used to generate predictions. However, selecting the appropriate ML method is crucial to achieve optimal speed, accuracy, and memory utilization, especially in a dynamic network environment. For a comprehensive classification of the related AI approaches for traffic management in MPTCP, please refer to the survey in [108].

Despite significant progress in the convergence of various radio access technologies, research in the literature has mostly focused on low mobility scenarios in wireless environments [40, 49]. However, there have been studies to solve access selection challenges in heterogeneous vehicular environments. For example, the works proposed in [97] and [98], relies on fuzzy logic to estimate the suitability of each RAT to support the QoS requirements of the various applications. The experimental results demonstrate that the combination of fuzzy logic and Multiple-criteria decision-making (MCDM) components is effective in tracking changes in the operating conditions of different RATs and implementing an adjustable context-aware strategy for selecting the optimal underlying network. Two different models were considered in [110] to model the network selection process. The first model is based on Technique of Order Preference Similarity to the Ideal Solution (TOPSIS) , whereas the second model is based on the Markov decision process (MDP). Authors have proposed a framework consists of a complete protocol that includes various modules to be deployed at the network control plane and at the user equipment. The framework enables the selection of the most suitable radio access technology for connected vehicles based on various criteria, including service type, the mobility feature, and traffic dynamics. The simulation models were used to evaluate the performance of the proposed framework, considering two heterogeneous wireless technologies, namely, Wi-Fi and cellular networks. The multi-criteria based handover algorithm for V2I communications have been recently introduced in [111] for use in heterogeneous network environments. An MCDM algorithm was utilized to determine the optimal network for handover by considering the QoS needs of the ongoing services, such as guaranteed bandwidth, latency, Packet Loss Ratio (PLR) and the usage cost. The proposed algorithm utilizes the Simple Additive Weighting (SAW) method [112], with the weighting vector of the decision element determined through the eigenvalue method of Analytical Hierarchy Process (AHP). Furthermore, a weight vector is developed for each application profile to enable the selection of the most appropriate network for handover, based on the QoS requirements of the ongoing services. The problem of seamless handover between Light Fidelity (Li-Fi) and Wi-Fi networks has been addressed in [113], in which an adaptive cross-layer handover algorithm based on MPTCP was demonstrated. According to this algorithm, the MPTCP can switch dynamically from *default* to *full-mesh* operational mode based on the mobility level of users, which is determined using FAHP-TOPSIS method. In an indoor wireless network scenario, a central controller was utilized to constantly monitor the entire network and gather channel status information periodically. The simulation results demonstrate that the proposed algorithm can reduce the handover times and improve throughput and service continuity of users. Authors in [33] investigates a new MCDM weighting technique to improve VHO decision in V2I network for multimedia services transmission. The proposed approach is based on the combination of two main weighting techniques, the AHP technique and the Entropy technique, which takes into account the required quality of service metrics as well as the vehicle velocity. The effectiveness of the proposed VHO method was evaluated for both conversational and streaming services. The results demonstrate that it outperforms commonly used RSS-based techniques and is well-suited for future V2X applications.

Nevertheless, existing solutions in the context of vehicle communications rarely consider a multipath approach. Most research in vehicular heterogeneous networks has concentrated on achieving seamless handover for data offloading, employing a single RAT for communication at any given instance. Furthermore, existing decision-making techniques do not consider the unique QoS requirements of each ongoing service, which can result in suboptimal utilization of network resources. For example, in the case of real-time services like voice and video streaming, which are more susceptible to delay, the decision algorithm may prioritize networks with lower latency. However, the algorithm may choose an interface with the highest available bandwidth for web and data traffic. Some works have suggested defining a service priority value to reflect the significance of the service to the user [43, 99]. Consequently, lower-priority services could be directed to use a non-optimal network to achieve consensus among multiple services.

Chapter 3

Emulation Framework

Multihoming is recognized as a crucial feature of forthcoming vehicular networks, creating a multi-technological environment capable of facilitating high data rates and low-latency communication [79]. To comprehensively assess the performance of emerging multihoming solutions within a controlled environment, deploying a real test bed is necessary. However, modifying and scaling such test beds can be challenging. As a result, computer simulation platforms like ns-3 [114] have become crucial tools for the research community. These simulation tools offer significant power, featuring a wide number of modules that facilitate the analysis of intricate network topologies on a large scale [115–120]. However, many of these modules do not incorporate the intricacies of real-world services, leading to simulation behaviors that are not entirely realistic. For instance, Multipath TCP (MPTCP) is yet to be integrated into any of the established versions of ns-3 [121], and the existing third-party MPTCP modules currently available do not fully adhere to the actual protocol specifications [122–124].

Network emulation offers the opportunity to assess real applications within a controlled and replicable context [125–128]. Common Open Research Emulator (CORE) [129], for instance, is a well-known lightweight network emulator that employs container virtualization methods to ensure separation among emulated hosts.Nevertheless, CORE does exhibit several notable deficiencies, including the absence of streamlined simulation for the link and physical layers, the incapacity to handle intricate scenarios and genuine mobility patterns, and the limited capability to emulate protocol stacks and applications beyond those directly supported by the kernel. In this study, we employ the emulation capabilities of ns-3 to evaluate the performance of MPTCP in heterogeneous vehicular networks. The versatility of ns-3 allows us to construct an emulation environment that closely replicates real-world behavior without the need for an actual network setup. Our emulation setup facilitates real-time experiments utilizing various wireless technologies, including LTE and IEEE 802.11p, which can be simulated using the ns-3 network simulator. Additionally, we enhance the authenticity of our experiments by incorporating genuine mobility models provided by the Simulation of Urban MObility (SUMO) [130] traffic simulator.

3.1 Building blocks of the emulation framework

3.1.0.1 Network simulator

A range of recognized network simulation and emulation tools are accessible to the scientific and research community, offering distinct features and advantages. Among various simulation tools, NS-3 is notable for its high adaptability and reasonable accuracy in evaluating network performance. NS-3, or Network Simulator 3, is an open-source discrete-event network simulation framework. It is designed to model and simulate computer networks and networking protocols. It is highly modular, allowing users to customize and extend its functionality. One can create their own network components and protocols, making it a flexible platform for various research and experimentation needs. NS-3 provides realistic models for various networking components, including wired and wireless communication links, network topologies, routing protocols, and transport protocols. Furthermore, NS-3 supports a wide range of network technologies, including Internet Protocol (IP) networks, wireless networks (Wi-Fi, LTE, 5G, etc.), ad hoc networks, vehicular networks, and more. This versatility makes it suitable for studying various networking scenarios. Moreover, NS-3 has an active and dedicated user community. It also provides extensive documentation, tutorials, and examples to help users get started with simulation development.

For this investigation, we utilize the real-time scheduling mode of ns-3, which differs from pure simulation mode, as it permits the simulation to engage with actual systems and execute real applications. ns-3 possesses various capabilities that enable it to interface with real entities across multiple levels. Specifically, our framework uses the file descriptor network device (*FdNetDevice*) to establish a link between the ns-3 device and a physical device on the host machine, facilitating the connection between simulated and actual systems.

3.1.0.2 Node virtualization

Linux offers an extensive range of virtual networking capabilities that can serve as the foundation for creating a standard emulation setup encompassing virtual nodes, bridges, tap devices, and interfaces. Linux Containers (LXC), often simply referred to as containers, are a lightweight form of virtualization that allow to run applications and their dependencies in isolated environments. Containers rely on features provided by the Linux kernel, such as `cgroups` (control groups) and `namespaces`. These features enable resource isolation and process separation, which are fundamental to containerization.

The virtual node in our test bed can be realized through the LXC, that operates with an isolated perspective of system resources, a distinct process hierarchy, and segregated networking interfaces. Each container runs as an isolated process on a shared operating system kernel, ensuring that it cannot interfere with other containers on the same host. Furthermore, containers are highly efficient because they share the host OS kernel and use fewer system resources compared to traditional virtual machines (VMs). This results in faster startup times and lower overhead. Ordinarily, the virtual bridge of the host machine is used to interconnect real entities, such as LXC, with the nodes located within the ns-3 simulation. This bridge can be viewed as a direct wired link between two virtual interfaces, facilitating data forwarding from the container to the simulated ns-3 node.

3.1.0.3 Traffic simulator

In order to provide realistic vehicle mobility patterns in our test bed, we opted to integrate a well-known and widely used platform for traffic simulation named SUMO. SUMO is an open-source traffic simulation tool that is commonly used for modeling and simulating road traffic in urban areas. It allows users to simulate various aspects of road traffic, including vehicle movements, traffic lights, pedestrian behavior, and public transportation systems. It is particularly useful for studying traffic flow, congestion, and the impact of different traffic management strategies. SUMO is open-source software, which means it is freely available for anyone to download, use, and modify. It includes various traffic models, such as car-following models, lane-changing models, and public transportation models. These models can be customized to replicate specific real-world traffic scenarios. Furthermore, SUMO supports the simulation of various traffic control strategies, including traffic signal timing, priority rules, and right-of-way rules.

Traffic Control Interface (TraCI) is an important component of SUMO that allows users to interact with and control traffic simulations. It enables users to start, pause, and stop simulations and retrieve data about simulated objects, such as vehicles' positions, speed, and direction. Additionally, TraCI allows users to impose actions on the simulated traffic scenario from external sources, such as assigning new routes to vehicles.Using TraCI API, SUMO can be integrated with other simulators, making it a valuable tool for our emulation framework.

3.2 Framework overview

To study multi-connectivity within heterogeneous wireless networks, we start by employing the standardized out-of-tree implementation of MPTCPv0 [90], which has been integrated into the Linux kernel using an available patch [67]. Consequently, the initial phase involves setting up a Linux system operating with the MPTCPv0 kernel. Subsequently, we establish two LXC, each equipped with two virtual interfaces, and segregate them from the remaining operating system components. Depending on the specific demands of the experiment, LXC can be substituted with network namespaces (*netns*), a lighter virtualization technique that solely virtualizes and isolates the network environment. Comprehensive configuration examples for both scenarios have also been made accessible for public reference.

The virtual interfaces (*veth*) serve as conduits for transmitting traffic between the LXC containers and the ns-3 network, and vice versa, facilitated by the FdNetDevice. They establish connections between each container and the corresponding Node within the simulation. The main reason for employing the FdNetDevice over TapBridge is that LTE devices are incompatible with direct mapping to Linux Tap devices, meaning they cannot directly exchange Layer-2 network traffic with external devices. To overcome this constraint, we employ ghost nodes, as initially proposed in [131], to facilitate the transfer of traffic between containers and the simulated LTE network and vice versa, as shown in Figure 8. Ghost nodes in emulation refer to virtual nodes that are used to replicate or simulate the behavior of real network nodes. These virtual nodes simulate a physical presence of the real nodes and are essential in network emulation to mimic the characteristics and functions of actual devices.

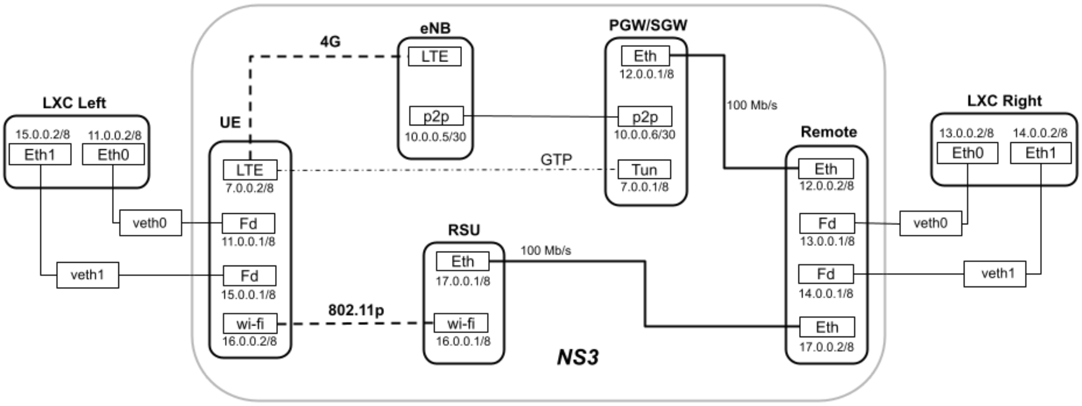


Figure 8: A detailed representation of a network topology.

In this setup, ns-3 operates as a sealed system (black box) featuring simulated elements that send and receive data through virtual links to actual nodes. It is possible since the real-time scheduler aligns the ns-3 simulation with the real-world clock.

A real-time scheduler is responsible for managing and controlling the timing and execution of events within a ns-3 simulated environment. It ensures that actions or processes in the simulation occur according to predefined schedules, deadlines, or timing constraints, mimicking real-world time behavior. ns-3 uses real-time scheduling to control the timing of packet transmissions, routing decisions, and other network-related events. The main goal of a real-time scheduler in simulation is to maintain synchronization with the simulation clock and ensure that events occur at the specified times or according to specific timing constraints. This is crucial for generating accurate and reliable simulation results, especially when studying time-sensitive systems or scenarios.

This emulation method introduces an additional hop in the communication route; however, the impact of this extra hop is negligible and can be disregarded. On the ns-3 side, the User Equipment (UE) node has two wireless interfaces (LTE and 802.11p) to ensure multipath connectivity across the simulated networks. One interface is linked to the LTE network via the Evolved Node B (eNodeB), while the other can connect to a Roadside Unit (RSU) through the 802.11p ad-hoc network. The LENA open-source module [132] for the ns-3 framework was used for the Long-Term Evolution (LTE) technology simulations. It controls the connection between the UE and the eNB and contains the radio protocol stack and LTE access mechanism established by the 3rd Generation Partnership Project (3GPP) standard.

The LENA module was specifically designed as a product-oriented simulation tool, catering to LTE equipment manufacturers by enabling the testing of management algorithms within real-world scenarios.

This module is open-source and follows a collaborative development approach within the community. It comprises two primary components: LTE modeling and EPC modeling. The LTE model encompasses the radio protocol stack, situated within the radio access network air interface connecting User Equipment (UEs) and eNodeBs (eNBs). This radio protocol stack includes various layers, such as the Physical (PHY), Multiple Access Control (MAC), Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), and Radio Resource Control (RRC) layers. The EPC model focuses on the interfaces within the core network. These interfaces facilitate communication between eNodeBs, PGW (Packet Data Network Gateway), SGW (Serving Gateway), MME (Mobility Management Entity), and other core network entities. The EPC model ensures end-to-end IP connectivity through the PGW.

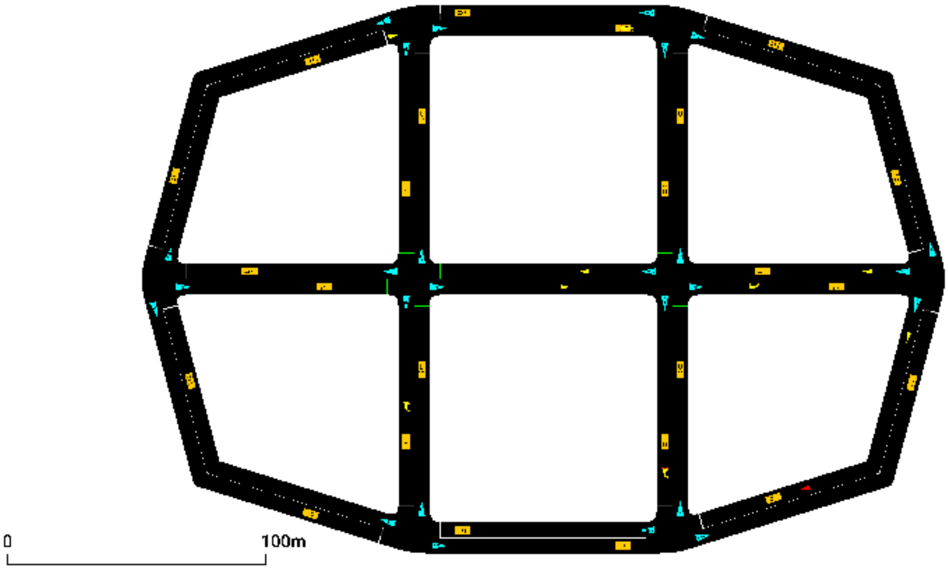


Figure 9: Visualization of simulation scenario with SUMO.

At this point, the emulation session can be visualized using the SUMO-GUI on the traffic simulator side and NetAnim [133] on the NS-3 side, as depicted in Figures 9 and Figure 10. NetAnim, an offline animator powered by the Qt toolkit, facilitates animation of the simulation by utilizing an XML trace file collected during the simulation process. Table 1 shows the network configuration parameters that were employed for initial evaluation. The MPTCP parameters, including the packet scheduler and path manager, have been set to their default values. Additionally, the majority of network parameters have been left at their default settings.

To facilitate the management of vehicle mobility, we utilize the open-source ns-3-based framework called ms-van3t [134]. This framework seamlessly integrates with the SUMO simulator to emulate authentic mobility patterns of vehicles. It employs the *TraCI API* to concurrently operate both simulators, ensuring continuous tracking of each vehicle's position. This synchronization aligns the movements of the ns-3 nodes with those of the vehicles in SUMO, as depicted in Figure 11. Real-time schedulers can regulate the movement of vehicles, traffic signals, and other elements in traffic simulations to mimic real-world

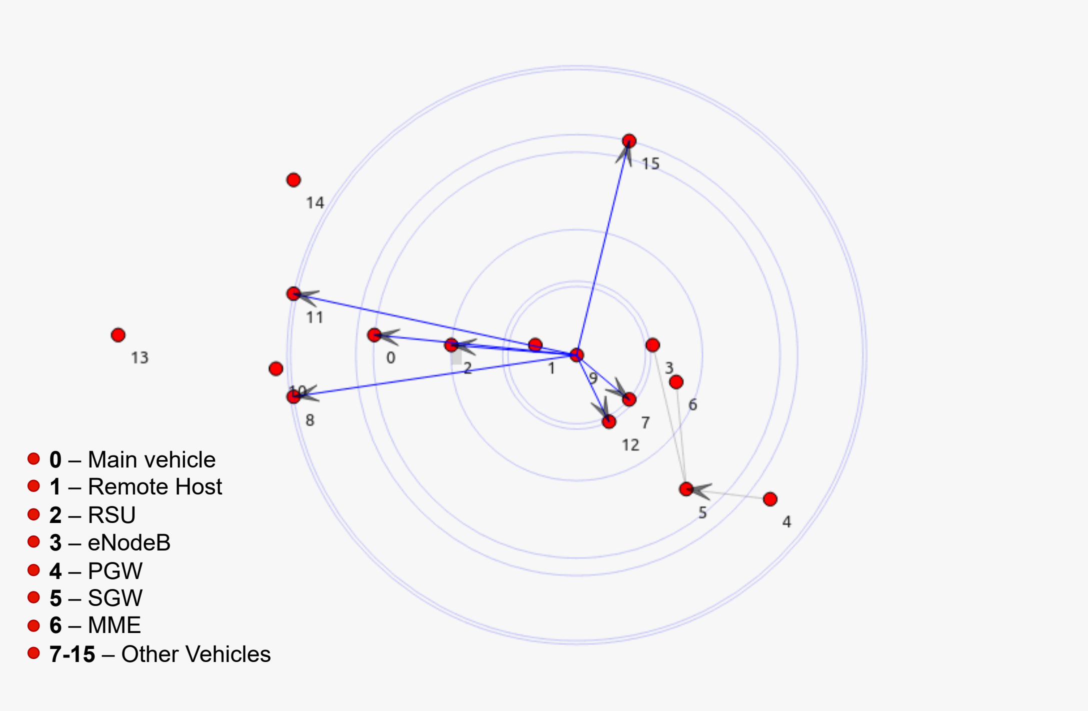


Figure 10: NetAnim screenshot of example scenario.

Table 1: Network parameters

WLAN Network	**Value**
Phys. channel model	YansWifiChannel
Standard WLAN	802.11p
Channel number	172
Frequency	5860 MHz
Channel Width	10 MHz
TX-Power level	23 dBm
Rx-Sensitivity	-101 dBm
Data Mode	OfdmRate6MbpsBW10MHz
Cellular Network	
Standard	LTE
UE Tx Power/Noise Figure	10/9 dBm
eNb Tx Power/Noise Figure	30/5 dBm
Uplink/Downlink freq.	1930/2120 MHz
Acm Model	PiroEW2010
Number of resource blocks	100 (20MHz)
Transmission mode	SISO
PropagationLossModel	Friis

traffic patterns accurately.

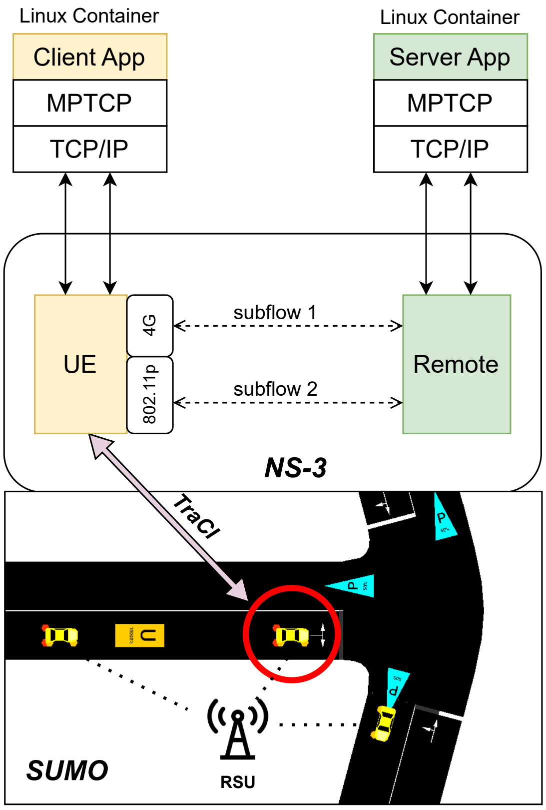


Figure 11: Overview of the emulation framework.

3.3 First experimental validation

In order to analyze the applicability of the vehicular multi-RAT emulation technique, we executed various test scenarios and examined their encountered limitations. All experiments were conducted on an AMD EPYC 7302 machine operating with a VM Linux Mint 20 (64 bits) system, equipped with eight cores and 20 GB of RAM. Once the experiment setup is in place, we can establish connections with both containers and assess performance using standard tools such as *iperf* and *tcpdump*. Additionally, we examined the scalability of our proposed emulation approach by progressively enhancing the intricacy of the network topology.

3.3.1 V2N Scenario

The primary objective of this test is to rigorously validate the effectiveness and robustness of the applied emulation technique, particularly in the context of dynamic scenarios characteristic of vehicular networks. Our analysis is directed towards assessing the connectivity performance within a Vehicle to Network (V2N) heterogeneous setting, wherein the vehicle is required to transmit data through multiple networks, each possessing distinct and varying characteristics. To evaluate our proposed approach, we selected a one possible use case: remote monitoring and data collection for vehicles, as depicted in Figure 12. To achieve this, we've established a remote host responsible for gathering real-time information from vehicles. Once this data reaches the remote server, which is situated within the traffic center's office, a traffic operator can conduct real-time analysis. In the event of any issues with the car, such as sensor data anomalies, potential loss of connection, adverse weather conditions, the vehicle's passengers could be alerted. It's important to highlight that, given our specific application, the primary focus was on minimizing packet loss and latency. Additionally, for the purposes of our testing deployment, we selected two widely accessible technologies: ITS-G5 (IEEE 802.11p) and Cellular (LTE) networks, which is tailored for Vehicular Ad Hoc Networks (VANETs). The selection of these two access technologies from among numerous options available in ns-3 is based on several compelling reasons. Firstly, they have garnered widespread acceptance both in academia and industry. Secondly, these standards have reached a mature stage of development. Thirdly, they have undergone individual deployments and rigorous testing, aligning well with the specific characteristics of vehicular environments. It's worth noting that as 5G technology matures and becomes more widely available, our approach can be seamlessly extended to incorporate it as an additional available technology.

We used the *iperf3* traffic generator tool to establish an MPTCP connection between the vehicle and a remote server behind an RSU and an LTE network infrastructure. Through this experiment, we aim to comprehensively evaluate the emulation's ability to replicate real-world behavior and emulate the intricacies of multipath vehicular communication. In this specific scenario, the vehicle is moving at a maximum velocity of 60 km/h while establishing a multipath connection with a remote host through two distinct network interfaces, namely 802.11p and 4G. It is assumed that the cellular network is always accessible to the vehicle, while ad hoc connectivity with RSUs deployed at different locations is only available intermittently along the road.

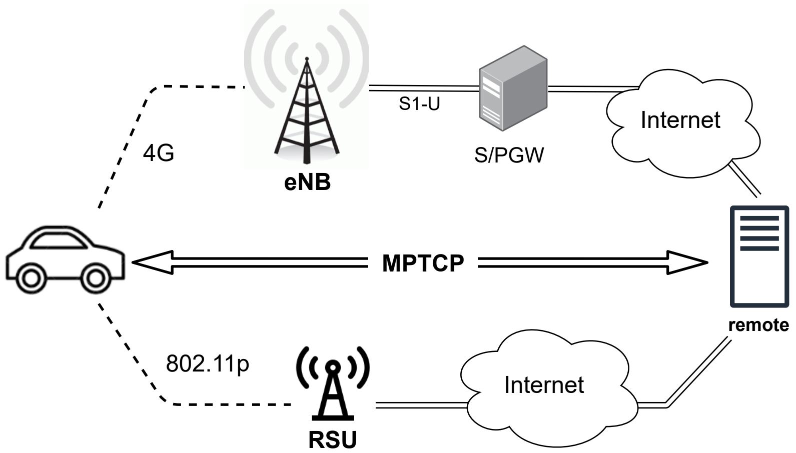
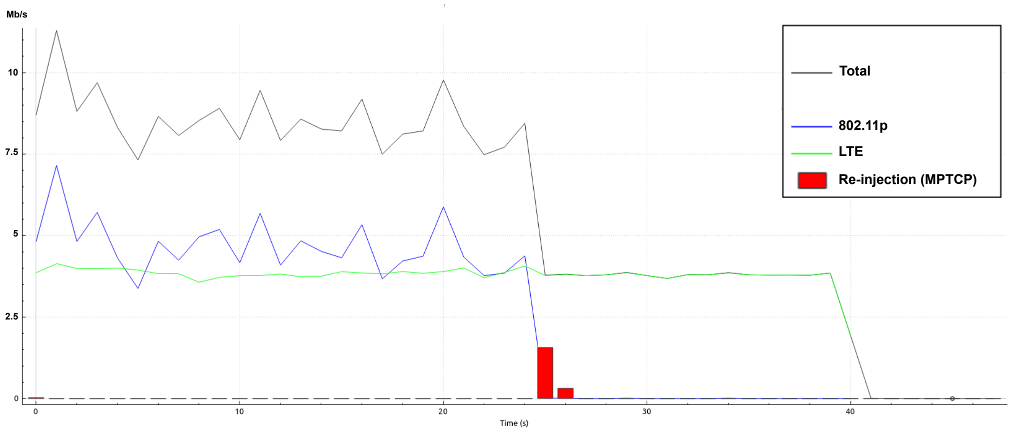
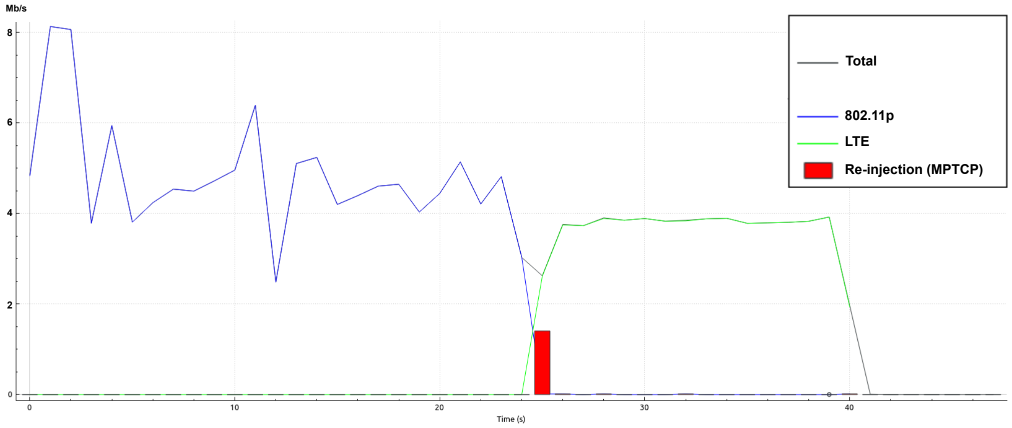


Figure 12: The V2N multipath scenario under consideration.

When employing MPTCP across multiple networks, the established logical connection between two endpoints is fragmented into distinct subflows. When transmitting data, it's partitioned into packets, which are distributed across the various subflows via a process known as scheduling. Upon reception, the receiver reconstructs the original logical connection by collectively processing the packets received from the different subflows. Figure 13 illustrates the average throughput per interface and the overall attained throughput throughout the MPTCP session. Furthermore, the figure illustrates the amount of lost data, which is subsequently re-transmitted by MPTCP upon detecting the connection loss and identifying the corresponding lost packets for re-injection. Initially, the vehicle is moving within the RSU and the eNodeB communication range. Subsequently, the vehicle exits the coverage area of the RSU while maintaining connectivity with the LTE network. Notably, the bandwidth limitation on the ns-3 side replicate the behavior of the simulated wireless technologies, accurately representing the practical limitations experienced on each network.



(a) Aggregation mode



(b) Backup mode

Figure 13: Throughput achieved by MPTCP in (a) aggregation mode and (b) backup mode

3.3.2 Aggregation mode

When a vehicle establishes an internet connection through multiple wireless networks, an application can concurrently utilize these multiple wireless networks to harness the combined bandwidth they offer. This concurrent utilization of multiple networks for enhanced bandwidth is commonly referred to as ”bandwidth aggregation” for simplicity and clarity. The vehicle and remote server collaborate to implement bandwidth aggregation, enabling the application to access the bandwidth of all available wireless networks. To achieve this, they offer a unified and standardized transport-layer MPTCP socket interface to in-vehicle applications and content servers. This interface ensures that the process of bandwidth aggregation remains transparent to these applications, effectively presenting the multiple wireless networks as a singular communication pipe or socket connecting the in-vehicle applications and content servers. This way, developers can make in-vehicle applications and content servers without needing to consider the underlying complexities of wireless communication heterogeneity. Additionally, legacy applications and content servers can be seamlessly supported without requiring any modifications.

As anticipated, MPTCP achieves higher throughput when the vehicle is located within coverage of both networks, enabling a connection to both an LTE base station and an RSU concurrently (Figure 13(a)). As the vehicle moves beyond the range of the RSU, the 802.11p link is disconnected. However, the MPTCP session persists, albeit with a diminished throughput. As the reliability of the 802.11p link diminishes, resulting in packet loss, the MPTCP packet scheduler stops using the unreliable interface, and as the vehicle advances beyond the coverage range of the RSU, the total throughput undergoes a reduction. The cellular network ensures the continuity of the MPTCP session and data transmission, thereby guaranteeing uninterrupted connectivity and facilitating a seamless Vertical Handover (VHO) process for a moving vehicle. The transition during the VHO is not instantaneous, given that MPTCP requires a certain period to detect the degradation of the link and adjust the connection accordingly. MPTCP relies on Retransmission Timeout (RTO) events to identify path loss; therefore, the RTO timer constitutes the primary factor influencing packet re-injection. The default MPTCP packet scheduler adeptly allocates the application data stream between two subflows, effectively utilizing both the cellular and ad-hoc networks to achieve optimal data rates. Subsequently, when the vehicle moves beyond the RSU’s coverage zone, the application remains unaware of the disconnection of the 802.11p link, thereby ensuring continuous operation. Capacity aggregation is, however, poses significant challenges [135], particularly when dealing with heterogeneous paths characterized by variations in delay and loss, as will be demonstrated in the upcoming chapter.

3.3.3 Backup mode

Upon establishing a new connection, MPTCP has the capability to configure backup paths, only becoming operational in the event of a primary subflow failure. To further investigate this, our subsequent experiment was designed to configure the endpoints to define the primary route through the IEEE 802.11p interface, resorting to the cellular network solely when the primary link becomes unable. Therefore, without bandwidth aggregation, an application on a vehicle can utilize only one of the available wireless networks at the same time. As depicted in Figure 13(b), the MPTCP activates the backup path, triggered by the failure of the primary subflow, when the vehicle moving out of the RSU coverage area. As the vehicle loose connection with the RSU, MPTCP have to retransmit packets in flight to achieve a reliable and in-order transmission of data. In such instances, connection loss is identified indirectly through timeout occurrences and absence of acknowledgments. MPTCP re-inject packets once the RTO for a subflow elapses. Packets sent on that subflow, that are not acknowledged, are re-injected on the backup subflow. Detecting the problem, the protocol smoothly shifts the data flow to the predefined backup subflow. The duration between the connection loss and MPTCP's detection typically spans several Round-trip time s (RTTs).

The Linux implementation of Multipath TCP exclusively triggers data re-injection to alternate subflows after an RTO event transpires on the present subflow. Similar to TCP re-transmissions, data re-injections are a sender's corrective response to tackle performance or functional problems. However, this reactive approach might lead to untimely resolutions and abrupt transmission interruptions. A more agile strategy could involve data re-injection upon detecting a sudden rise in RTT and loss rate on a specific subflow. It's noteworthy that data re-injections are only applicable to connections with a minimum of two subflows, and the same data can be sent over several subflows.

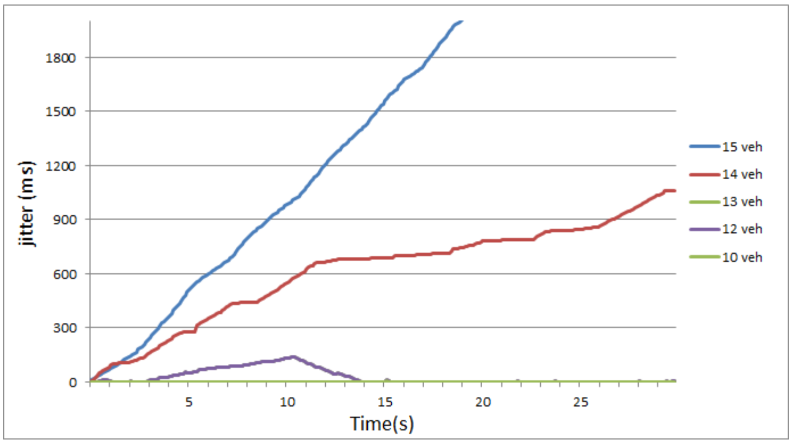
3.4 Framework Limitations

It's important to highlight that the maximum throughput attainable using FdNetDevice for a single TCP connection hovers around 100 Mbps. As suggested by the ns-3 documentation, this constraint is likely attributable to the processing capabilities of the hardware employed in experiments. While the additional hop introduced from LXC to ns-3 marginally contributes to transmission delay, its impact is negligible. Furthermore, our testing framework faces limitations in replicating the accurate behavior of real networks during extensive experiments due to the constrained processing capacity of hardware resources. When ns-3 is operated in emulation mode, real-time event scheduling is used to synchronize the simulation clock with the hardware clock. However, within vast network topology, there can be instances where numerous events appear in rapid succession, leading to a delay in executing enqueued events.

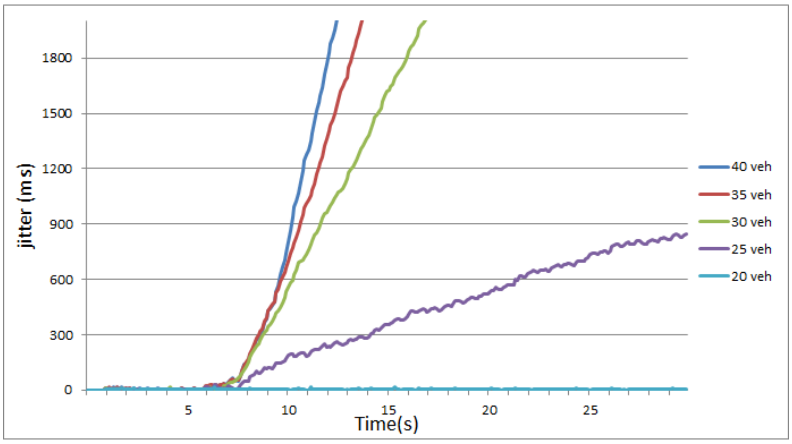
This temporal deviation, named *jitter*, manifests as the variance between the expected event processing time and actual execution time. Simulation jitter refers to the variation in the timing or delay of events in a simulated environment. In simulations, especially real-time simulations, the timing of events is expected to be as precise and consistent as possible to accurately represent the behavior of a real-world system. The jitter occurs when there are unexpected or irregular variations in the timing of events within the simulation. For example, if data packets are expected to be transmitted at regular intervals but there are occasional delays or variations in the timing of packet transmission, that would be considered jitter. Thus, jitter can affect the accuracy of simulation results, particularly in scenarios where precise timing is crucial, such as in real-time systems or simulations of communication networks.

Using Figure 14, our objective is to examine how the simulation jitter is influenced when we increase the number of communication devices in the simulation. The ns-3 real-time scheduler offers a configuration option called *Hard Limit* mode. Once enabled, the experiment halts abruptly if the jitter surpasses a predefined threshold (default 100 ms). The emulation results shows a noteworthy correlation between the level of jitter and the number of network devices integrated into the simulation. The experimental setup with several additional LTE devices connected to a shared base station, all participating in downloading a 512 Kbps UDP traffic stream from a remote host. The quantity of LTE devices integrated within the simulation exerted a substantial impact on the real-time operational efficiency of ns-3, as illustrated in Figure 14(a). Another experiment confirms this result, as comprised of vehicles, each equipped with an 802.11p device running the Cooperative Awareness service, significantly hampering ns-3's proficiency in managing real-time tasks, as shown in Figure 14(b).

Mitigating jitter may require a combination of different strategies, such as optimize code, high-resolution timers, synchronization and parallel processing, depending on the nature of simulation and its specific requirements. Also, ensure that the hardware and system resources on which running the simulation are not heavily loaded. Resource contention can lead to timing irregularities. It's essential to carefully analyze your simulation's behavior and performance to identify and address sources of jitter effectively. Hence, it is essential to consistently monitor the jitter across all emulation scenarios to guarantee a respectable degree of precision in the generated outcomes. Enhancing the system's scalability can be achieved by augmenting the computational resources (CPU, RAM) or, alternatively, implementing the time dilation technique [136], necessitating certain modifications to the Linux kernel.



(a) LTE equipped vehicles



(b) IEEE802.11p equipped vehicles

Figure 14: The dependence of *jitter* on the number of network devices in simulation

Chapter 4

Design of cross-layer path manager

Within vehicular environments, Multipath TCP (MPTCP) holds promising potential for providing a seamless roaming solution and enhanced throughput, as it empowers the system to maintain a stable connection by intelligently switching between available network interfaces. Unfortunately, the existing protocol implementation overlooks user mobility and the volatile attributes of wireless links [137]. Instead, MPTCP relies on transport layer information to estimate channel capacity and latency, which inadequately captures the dynamic characteristics of vehicular networks. This lack leads to slow detection of link failures, particularly in cases of abrupt path disconnections, thereby causing suboptimal communication performance during handover events.

This chapter delves into the suitability of MPTCP in supporting resilient and efficient Vehicle to Network (V2N) communication across heterogeneous networks. Initially, we pinpoint and discuss several challenges prevalent in heterogeneous vehicular networks, encompassing issues such as Head-of-line (HoL) blocking and service disruptions during handover events. Subsequently, we propose a cross-layer path management scheme for MPTCP that harnesses real-time network information to enhance the reliability and efficiency of multipath vehicular communication. Our emphasis lies in adopting a cross-layer scheme, where interactions between upper and lower protocol stack layers are leveraged to enhance the accuracy of path selection.

In our approach, we assume that the vehicle doesn't wait to be disconnected from one network before attempting to connect to another. This is facilitated by the use of multiple network devices on each vehicle, allowing for seamless switching between networks. For architectural purposes, we leverage a distributed handover scheme, which plays a vital role in the effectiveness of our approach. In practice, our objective is for the logical MPTCP interface to primarily handle data transmission and reception, while the multiple physical interfaces serve as the means for radio access. The signaling and control plane components related to path management are implemented at the application layer. Specifically, the signaling aspect of the handover process, which we aimed to anticipate, is accomplished through the use of the distributed path management approach.

Through our emulation results, we demonstrate that the proposed scheme attains seamless mobility across heterogeneous networks and notably reduces handover latency, packet loss, and out-of-order packet delivery. These enhancements directly result in an improved quality of experience Quality of Experience (QoS) for vehicular users, achieved by reducing delays at the application layer.

4.1 Problems analysis

MPTCP offers the potential for seamless mobility during handovers in vehicular communications, achieved by transparently aggregating multiple wireless connections for applications. For example, a vehicle's On-Board Unit (OBU) equipped with two network interfaces could simultaneously utilize them to transmit data while moving on the road. This capability ensures the maintenance of Peer-to-Peer (P2P) connections even in the face of link failures, such as when the vehicle moves out of the coverage range of a network infrastructure node. While MPTCP shows promise in enhancing Vehicle to Everything (V2X) connections, challenges persist concerning service continuity during handoff scenarios. Given the highly dynamic nature of vehicular networks, the protocol must thoroughly assess link characteristics to prevent deteriorated communication performance.

4.1.1 Smooth handover

The process of transferring a connection from one access network to another is referred to as a Vertical Handover (VHO). Among the primary challenges in VHO management is the seamless and continuous connectivity for mobile users as their connections transition across heterogeneous networks [138]. Given the prevalent on-demand data streaming through the Internet, the imperative for uninterrupted connectivity across heterogeneous interfaces is heightened. Typically, the decision to execute VHO hinges on Physical Layer (PHY) indicators such as Receiving Signal Strength Indication (RSSI) and packet transmission timeouts. However, suppose a vehicular user is engaged in an activity that involves streaming data, like watching a video and transitioning from one network to another. In that case, the established connection may ultimately cease to be viable, ending the streaming process abruptly.

MPTCP can provide smooth handovers by shielding the transport layer from network address dynamics. However, distinct network properties such as bandwidth, delay, jitter, and loss rate characterize each path. For reliable transport protocols like Transmission Control Protocol (TCP), the congestion control, loss detection, and recovery mechanisms are often unaware of these variations, potentially leading to inadequate responses to abrupt changes in the network layer when transitioning between different paths. While MPTCP can dynamically transition between diverse wireless technologies, the protocol's detection of connection loss tends to occur after the actual disconnection, potentially necessitating the application to reinitiate the connection. Hence, there arises a need for an approach capable of predicting connection loss preemptively, enabling seamless VHO across heterogeneous networks. This anticipatory strategy becomes crucial in ensuring uninterrupted, high-quality user experiences, especially in dynamic vehicular environments where network conditions fluctuate frequently.

MPTCP assesses channel capacity and latency to arrange packet transmission. Nevertheless, in scenarios characterized by fluctuating channels, MPTCP struggles to efficiently exploit the entirety of the accessible capacity. Achieving seamless VHO is particularly intricate within vehicular networks, mainly due to the highly dynamic nature of the communication environment [66]. The MPTCP has not been designed to operate in unstable wireless links and high mobility contexts. Consequently, the protocol performance might not be optimal in vehicular networks, resulting in interruptions in data delivery to the applications. The method for detecting path failures requires time to determine the state of path deterioration, which can lead to significant data transfer issues during VHO events.

4.1.2 Non-optimal scheduler strategies

In multipath communication, packet scheduling is essential in distributing data packets across various paths, aiming to enhance transmission efficiency. Different scheduling algorithms have been proposed in the literature to cater to diverse objectives [139–142]. The choice of scheduler substantially impacts the performance of MPTCP, particularly in scenarios characterized by connectivity challenges. Whenever an MPTCP sender is about to transmit data, it has to address three key decisions. Firstly, it must identify which subflow(s) are viable for data transmission. This determination is managed by the MPTCP's congestion control mechanism, which tracks a congestion window (cwnd) for each subflow. Subflows with an available cwnd are candidates for data transmission. Secondly, if multiple subflows boast available cwnd, a scheduler is tasked with selecting the subflow through which data will be sent. Finally, upon selecting a subflow, the scheduler must ascertain the amount of data to be dispatched through it. This third decision centers around the granularity of data allocation.

To comprehensively evaluate the performance of MPTCP schedulers in a vehicular scenario, a series of experiments were conducted within our emulation framework, which incorporates two disjoint paths between the vehicle and a remote host in the topology. The default scheduler utilized in the Linux MPTCP implementation determines that the 'fastest' subflow, characterized by the largest bandwidth or lowest packet delivery delay, is selected to transmit data at any given moment. The default MinRTT scheduler [79] uses a subflow until its congestion window is fully utilized, after which the subsequent subflow with the minimum Round-trip time (RTT) is employed. The performance of MPTCP is sensitive to variations in delay among its subflows. As illustrated in Figure 15, the effects of the scheduler become apparent when disparities in delay among paths are present. The red dots represent data segments distributed across LTE and 802.11p interfaces. As expected, the scheduling of traffic on the subflows is unequal when the subflows experience dissimilar network characteristics, i.e., delay and bandwidth. The outcomes reveal that differences in delay lead to scheduling algorithms exhibiting more pronounced time-varying idle intervals. Ensuring load balancing among subflows emerges as an important criterion requiring careful consideration. In path heterogeneity, we observe that the default MPTCP scheduler struggles to optimize certain paths' utilization. These findings emphasize that the viability of multipath communication hinges upon adequate bandwidth capacity across accessible links and minimal disparities in RTT among pathways.

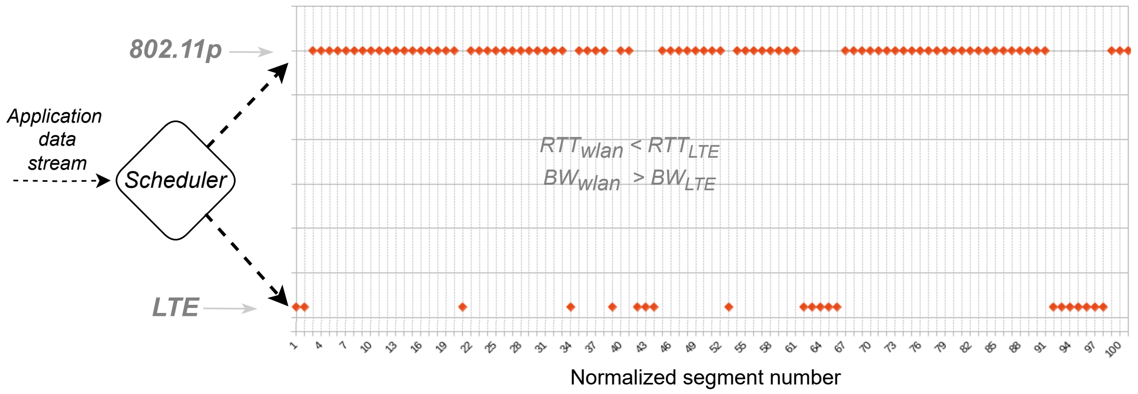


Figure 15: An example of an MPTCP scheduler managing 100 segments across heterogeneous links.

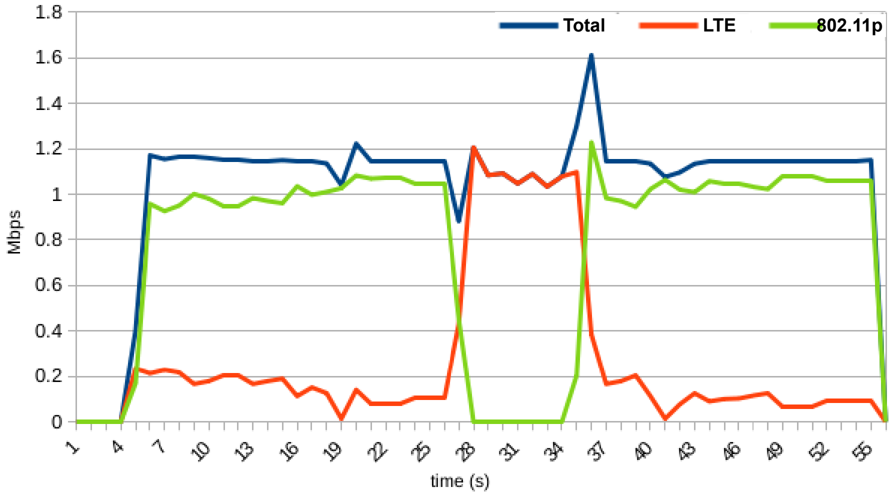


Figure 16: Throughput achieved using asymmetrical paths.

Our experiments show that heterogeneous paths can trigger undesirable application behaviors, such as diminished video streaming quality compared to what could be achieved with the available aggregate bandwidth. Several publications have pointed out that in heterogeneous networks, the throughput of MPTCP using multiple paths can even be worse than that of one TCP connection using only the best path [62, 143]. Figure 16 reveals that the path with higher delay experiences notably diminished effectiveness in terms of utilization. This raises questions regarding the necessity of including such a path within the communication session prior to its actual necessity, particularly when a handover becomes indispensable. Additionally, we note that the performance enhancements delivered by MPTCP are more pronounced with larger data volumes. However, it's noteworthy that shorter data flows may experience performance degradation when managed through MPTCP. A key contributing factor to these observations lies in the asymmetry among the subflows within an MPTCP connection, which can lead to varying performance outcomes. In light of our findings, it becomes evident that the default scheduler not only deviates from the objectives of the congestion control mechanism but also exhibits inefficiencies in utilizing network resources. These insights accentuate the importance of carefully balancing subflows and optimizing path utilization to align with the specific demands of vehicular communication scenarios.

4.1.3 Slow path-loss detection time

Transport layer protocols often encounter challenges when dealing with fluctuating path conditions. Similarly, Multipath TCP uses a similar approach to managing changes in network status as the actual implementation of TCP. It utilizes the standard TCP error recovery mechanism and congestion control to address sudden changes in path availability. MPTCP's reinjection strategy hinges on the Retransmission Timeout (RTO) of TCP, which prompts retransmissions when previously transmitted data is not acknowledged. This timer is set with caution, as it aligns with a substantial reduction in transmission rate due to congestion control. Consequently, TCP employs quicker retransmission methods, like duplicate acknowledgments and fast retransmit [144]. Therefore, the RTO serves as a final option to initiate retransmissions, warranting a conservative selection in TCP, which, in turn, impacts reinjection efficiency in MPTCP.

Moreover, due to the close relationship between MPTCP and TCP, TCP's time-dependent loss recovery mechanisms can also present challenges for networks characterized by high delays and losses. TCP's fast recovery (FR) and RTO mechanisms are directly linked to RTT, which means that the required RTT for recovery remains constant regardless of network capacity growth. While the scheduler is often used to enhance scenarios involving network heterogeneity in MPTCP, little progress can be made with legacy TCP loss recovery mechanisms when subflows not only have varying delays but also different loss characteristics. In this context, a subflow could fail after a single RTO, prompting the packet scheduler to cease using that subflow for data transmission. In vehicular scenarios, MPTCP's handover mechanism is typically reactive, resulting in temporary connection losses due to the varying conditions of wireless networks. Consequently, the QoS suffers as vehicles traverse multiple networks quickly.

4.1.4 HoL blocking

The multipath scenario introduces additional challenges: since packets must be delivered to the application in a sequential order, a loss or delayed delivery of a packet on one subflow can impact the entire multipath transmission. Any subsequent packets from other subflows must be buffered until the missing packet is received, leading to potential disruptions in the overall transmission. Multipath protocols often confront a HoL blocking issue within heterogeneous networks, where significant differences exist in the available bandwidths and latencies of the used paths. The multipath communication can lead to the receiver experiencing the disorderly arrival of packets, as data received through a faster subflow might need to wait for packets from a slower subflow to ensure in-order delivery. If the receiver buffer lacks the capacity, there is a risk that previously received packets might be discarded, resulting in a complete interruption of data transfer.

Considering the varying characteristics of different paths, including differences in path delay, bandwidth, and loss probability, packets sent over distinct paths may encounter disparate end-to-end delays. Consequently, a packet bearing a higher sequence number could potentially reach the receiver before a packet with a lower sequence number. A prevalent issue with Multipath TCP in vehicular networks is the significant discrepancy in RTT across various subflows. This variation in RTT is the root cause of HoL blocking, where data experiences a stall while still in transit through the slower path. This results in a significant data burst when the said content is eventually acknowledged at the connection level. Furthermore, receiver buffers have a finite capacity, which means that the advertised window size on all other subflows could decrease to zero while awaiting a single retransmission on the congested subflow. The high transmission error rate in wireless networks is another contributing factor to out-of-order (OFO) packet delivery. When a packet with a lower sequence number is lost and doesn't reach the receiver in a timely manner, the receiver might need to buffer a significant quantity of out-of-order packets as it awaits the arrival of the missing packet.

Therefore, network access based on MPTCP continues to encounter challenges due to the inherent instability of wireless connections caused by vehicular mobility. Specifically, delays or packet loss on one path, can disrupt the sequence of packets transferred over other paths, leading to an out-of-order arrangement in the receiver buffer. Once the receiver buffer becomes saturated, data transfers across all paths, including those with reliable connections, can be halted. It indicates that path heterogeneity and packet losses largely impair the performance of MPTCP. This HoL-Blocking problem can be addressed through an adaptive Path Manager (PM) approach or an intelligent packet scheduler algorithm. Current solutions focus on employing asynchronous packet scheduling at the sender to mitigate the discrepancy in delays on different paths, ensuring in-order packet arrival at the receiver. This approach, denoted as out-of-order scheduling, effectively addresses moderate path heterogeneity regarding delays. It try to estimate whether a path will cause HoL-Blocking and dynamically adapt scheduling to prevent blocking. However, it needs to improve in handling scenarios involving packet losses and substantial delays during handoffs. Notably, the ofo scheduling method may accumulate lost packets and eventually lead to HoL blocking, which cannot be prevented solely by ofo scheduling [59]. The following section demonstrates the HoL blocking under handoff events within a vehicular scenario.

4.1.5 Path under-utilization

As a vehicle moves beyond the coverage area of an Infrastructure node, the associated subflow becomes nonfunctional due to a weak signal, even though the corresponding interface retains a valid IP address. When a packet is lost and there are not enough duplicate acknowledgments (dupACKs) available, the MPTCP sender must resort to the Retransmission Timeout (RTO) mechanism to retransmit all unacknowledged packets. In this scenario, it requires the time of the initial RTO plus half of the RTT (RTT/2) to successfully retransmit all packets. MPTCP remains unaware of the path's unavailability and initiates a retransmission of lost packets. The RTO duration escalates exponentially when the expected acknowledgment fails to arrive. This sequence of events is depicted in Figure 17, where the second subflow encountered an interruption and recovery within a single MPTCP session. After the path's recovery, the sender must await the successful retransmission to confirm path availability. Thus, MPTCP cannot utilize the recovered path despite the subflow being active. This unnecessary delay results in the vehicle losing a substantial portion of its available connection time, which leads to under-utilization of the recovered path for long-lived flows. Our analysis highlights that MPTCP, in its default configuration, fails to effectively utilize all available paths within vehicular networks featuring loss-prone wireless links. Moreover, conventional RTT-based scheduling policies prove inadequate in mitigating this problem within mobile scenarios. The absence of information from lower network layers regarding their status led to suboptimal utilization of available resources.

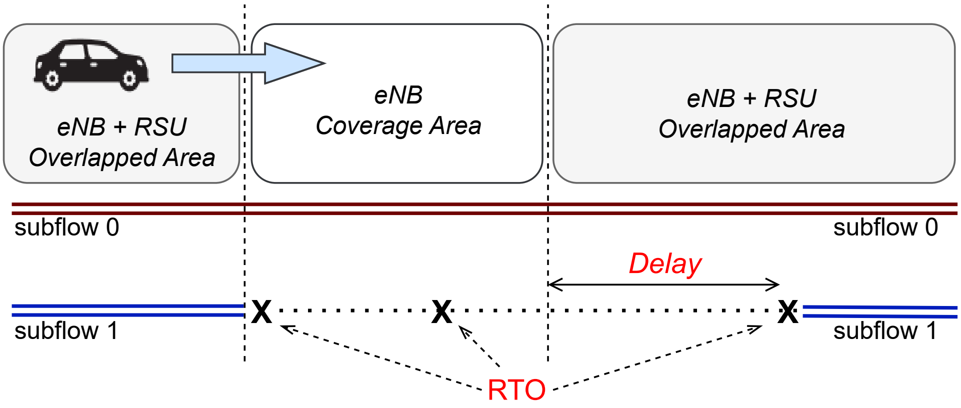


Figure 17: Under-utilization of recovered subflow.

4.2 The cross-layer approach

Numerous studies are currently being conducted to address mobility-related issues in vehicular multi-RAT environments [145–147]. These investigations underscore the importance of integrating cross-layer assistance into the MPTCP solution, leveraging MAC-layer information to estimate path quality [148, 149]. Cross-layer attributes could enhance path selection, as they exhibit a significant correlation with MPTCP's performance. However, only a few of these works have provided the design and implementation details of such a mechanism. A potential solution to enhance connectivity in heterogeneous vehicular networks might involve a proactive VHO mechanism. This mechanism could anticipate connection loss and smoothly transition between access technologies. However, it's important to note that the PM module, responsible for managing subflow utilization, resides entirely within the kernel. Moreover, the standard socket Application Programming Interface (API) provided by MPTCP doesn't offer the ability to control subflow utilization dynamically. In [150, 151], the authors suggest a separation of the MPTCP data plane from the control plane by relocating all PM functions to the *userspace*. They employed the *netlink* inter-process communication mechanism to engage with the MPTCP kernel to retrieve information about MPTCP events and manage established connections. An adaptable API was introduced, which provides access to events and state information from the kernel, enabling the PM to respond appropriately when specific events are triggered.

The MPTCP daemon (mptcpd) was recently introduced to allow the development of customized PM strategies in userspace [92]. It leverages the netlink mechanism to interact with the Linux kernel and control subflow usage from the application layer through the set of instructions. The upstream implementation of MPTCPv1 provides a netlink interface for PM-related operation, automatic endpoint configuration, and tracking per-connection information. As shown in Figure 18, the MPTCP daemon handles connections through the generic API exposed by the Linux kernel. This daemon adopts a modular architecture designed to seamlessly integrate new PM strategies in the form of extensions, or plugins. A plugin is responsible for managing subflows using the available commands. Concurrently, the interaction between kernel and userspace is overseen by the mptcpd, which also forwards in-kernel events to the respective plugin.

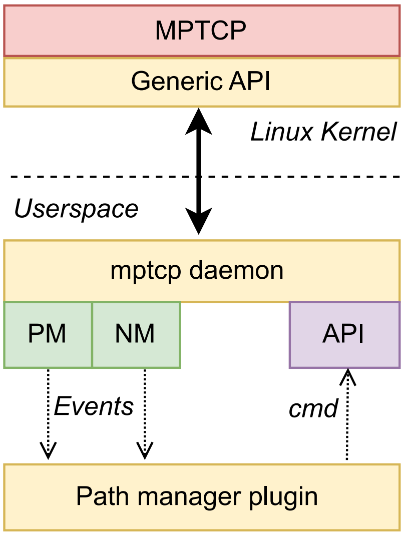


Figure 18: The *userspace* path manager.

A plugin will obtain notifications for events it registered to and should handle them accordingly. Corresponding callback functions are triggered by reacting to various kernel events that can occur during an MPTCP connection. The PM module is responsible for delivering all PM-related events. Typical PM events include connections established or closed, subflow created, addresses advertised, etc. Table 2 lists available PM events currently supported by the generic API. The 'new connection' event occurs when an MPTCP connection is initiated but not completed. Following the successful three-way handshake procedure, the "connection established" event is triggered, carrying information essential for connection identification, such as unique tokens and local and remote addresses. Likewise, the "new subflow" and "subflow closed" events occur upon the establishment or termination of a new subflow, respectively.

Table 2: Path Manager Events

new_connection	A new MPTCP connection has been created
connection_established	New MPTCP connection has been established
connection_closed	MPTCP connection as a whole was closed
new_address	New address has been advertised by a peer
address_removed	Address is no longer advertised by a peer
new_subflow	A peer has joined the MPTCP connection
subflow_closed	A single MPTCP subflow was closed
subflow_priority	MPTCP subflow priority changed

Table 3: Network Monitor Events

new_interface	A new network interface is available
update_interface	Network interface flags were updated
delete_interface	A network interface was removed
new_local_address	A new local network address is available
delete_local_address	A local network address was removed

Table 4: Path Management Commands

add_addr	Advertise new network address to peers
remove_addr	Stop advertising network address to peers
add_subflow	Create a new subflow
set_backup	Set priority of a subflow
remove_subflow	Remove a subflow
get_addr	Get network address
dump_addrs	Get list of MPTCP addresses
flush_addrs	Flush MPTCP addresses
set_limits	Set MPTCP resource limits
get_limits	Get MPTCP resource limits
set_flags	Set MPTCP flags for a local IP address

Lastly, altering a backup priority flag triggers the ”subflow priority” event. For instance, the ”new address” event provides the IP addresses and ID of the endpoint announced by the remote host. Our PM algorithm uses this type of event to store the remote host’s addresses and create new subflows only when the vehicle is within the coverage area of the serving network. This approach offers more flexibility than the existing in-kernel PM, which attempts to establish additional subflows immediately after connection establishment.

Furthermore, the Network Monitor (NM) module assumes responsibility for relaying all network interfaces events. These events encompass scenarios like the availability of a new network interface, updates to interface information, as well as the availability or removal of network addresses. A comprehensive list of such events is provided in Table 3. Similarly, the kernel side can receive commands from the userspace to execute PM tasks. The general API encompasses various command types that have the potential to alter the state of the MPTCP connection. As shown in Table 4, the control level provided is generally sufficient for most vehicular scenarios. For instance, it's feasible to instruct MPTCP to generate a new subflow using the "add subflow" command based on unique connection tokens, local and remote IP addresses, and endpoint IDs, all of which must be predetermined. Similar commands facilitate subflow removal or changes in subflow priority. Appendix A provides a visual representation of the primary building blocks, their relationships within the userspace path management service, and additional supplementary information, such as a list of commands, events, data flows, data structures, and related functionality.

4.3 Path management algorithm

Detection of link loss at the transport layer may encounter delays, resulting in a suboptimal user experience. The envisioned cross-layer PM strategy seeks to preemptively anticipate connection problems by observing MAC-layer information across all accessible interfaces and estimating path quality using Receiving Signal Strength (RSS) measurements. The fundamental concept is identifying an impending connection loss before it happens, preventing data losses. Implemented as a mptcpd plugin, our PM strategy adjusts subflow utilization based on the perceived link quality, and a signal strength threshold determines the decision for handover. This approach aims to enhance MPTCP's responsiveness by proactively controlling link failures, facilitating seamless VHO, and optimizing resource utilization and system resilience.

Based on captured RSS information, our PM can disable subflows linked to weak signal quality and subsequently reactivate them once the signal strength surpasses a predetermined threshold. The involvement of cross-layer assistance becomes indispensable, as real-time network metrics like packet losses and RTT alone may not consistently forecast an imminent handover in highly dynamic vehicular networks. At the same time, the RSS metric can reflect overall channel performance and its widespread adoption as an indicator of wireless environment quality. Naturally, signal strength diminishes as a vehicle moves from the coverage range of an infrastructure node, and a handover is triggered upon crossing the defined threshold. A vehicular user may incur multiple VHO within a single MPTCP session, frequently transitioning between heterogeneous wireless networks along its route. The proposed PM algorithm allows for dynamic creation, removal, or modification of subflow priority, providing control over handover execution in high-mobility scenarios. Unlike the default MPTCP scheme, our PM does not create any additional subflows immediately after the creation of the connection. Instead, the new subflow is created only if the path exhibits good signal quality. We consider only one subflow over each active interface towards the remote host. The provided pseudocode in Algorithm 1 outlines this path management process.

For example, as the vehicle moves away from the Roadside Unit (RSU), a drop in RSS is observed, and once a particular threshold is reached, our PM initiates the transition of the subflow into a `backup` mode. Once a subflow has been set to backup mode, the MPTCP packet scheduler will only use it if the other subflow fails. Likewise, when the vehicle re-enters the RSU's coverage area, the PM will restore the subflow by removing its backup status, enabling the packet scheduler to resume using it for data

Algorithm 1: The path manager algorithm with cross-layer assistance

```
for each interface as $I$ do
    if $I$ is available then
        $\alpha \leftarrow$ smoothing factor $(0 \leq \alpha \leq 1)$
        $RSS_I(t) = (1 - \alpha) * RSS_I(t-1) + \alpha * RSS_I$
    end
end
for each subflow as $S$ do
    $I_s \leftarrow$ interface of $S$
    $T_i \leftarrow$ threshold for $I_s$
    if $I_s$ is not available then
        remove S and continue
    end
    if $RSS_I(t) \leq T_i$ then
        if $S$ is active and $S$ is not backup then
            set $S$ as backup
        else
            if $S$ is not active then
                remove $S$
            end
        end
    end
    if $RSS_I(t) > T_i$ then
        if $S$ is not active then
            establish subflow $S$
            set $S$ as active
        else
            if $S$ is backup then
                set $S$ as non-backup
            end
        end
    end
end
```

transmission. In both scenarios, the protocol will send the `MP PRIO` message to inform the remote host about the subflow status change. Given the high variance of vehicular networks, the RSS value can undergo significant fluctuations, potentially triggering frequent shifts in subflow operational modes. We implement a simple low-pass filter mechanism to mitigate the undesirable `ping pong` effect that could arise when the subflow toggles between modes too frequently. This mechanism aims to moderate the probability of abrupt handovers that may not be necessary. The responsiveness of this filter is determined by the smoothing factor *a*, which indicates the sensitivity of the filter to changes in the input signal. A higher smooth factor means that the PM will respond more slowly to changes in the RSS, resulting in a smoother output. However, an excessively high value of a smooth factor can cause the filter to become too sluggish, delaying the output response. The choice of the smooth factor depends on the specific vehicular application and the desired balance between unnecessary handovers and maintaining a fast response time. The MAC-layer information and related attributes are expected to be retrieved from a vehicle's OBU.

Chapter 5

Performance evaluation

In our experiments, we intend that a vehicle can access a remote host via two heterogeneous wireless technologies (i.e., 802.11p and LTE), as shown in Figure 12. The experimental platform's configuration aligns with the frameworks outlined in the Chapter 3, ensuring the generation of reliable results. The cellular network is considered to be always accessible by the vehicle, whereas a vehicle has casual ad hoc connectivity with Roadside Units (RSUs) that are placed at fixed locations along the road. The cellular 4G and ad hoc IEEE 802.11p wireless technologies are simulated by the ns-3 network simulator, whereas the SUMO traffic simulator provides realistic vehicle mobility. All experiments were performed in real-time with upstream Linux kernel v.15.7. The latest version of `iptools2` was used to configure a single subflow per interface for each MPTCP connection, while the `iperf3` was used to generate traffic between the vehicle and the remote host.

The proposed PM scheme continuously monitors the IEEE 802.11p interface status with the Receiving Signal Strength (RSS) value recorded for each received packet originating from the RSU. Subsequently, the PM makes informed decisions regarding suspending or releasing a subflow. Both vehicles and RSUs execute a fundamental road safety service as defined by the European Telecommunications Standards Institute (ETSI). In this service, Cooperative Awareness Message (CAM) are periodically broadcasted by vehicles and RSUs to nearby vehicles. These CAM messages facilitate the exchange of information about their positions and statuses within a single hop distance. The PM relies on receiving CAM messages from RSUs to collect RSS samples. The threshold for the IEEE 802.11p interface is established at -82 dBm. This threshold represents a critical point at which the reliability of data transmission and reception approaches a critical level, indicating the potential unreliability of the link.

5.1 Interrupted subflow

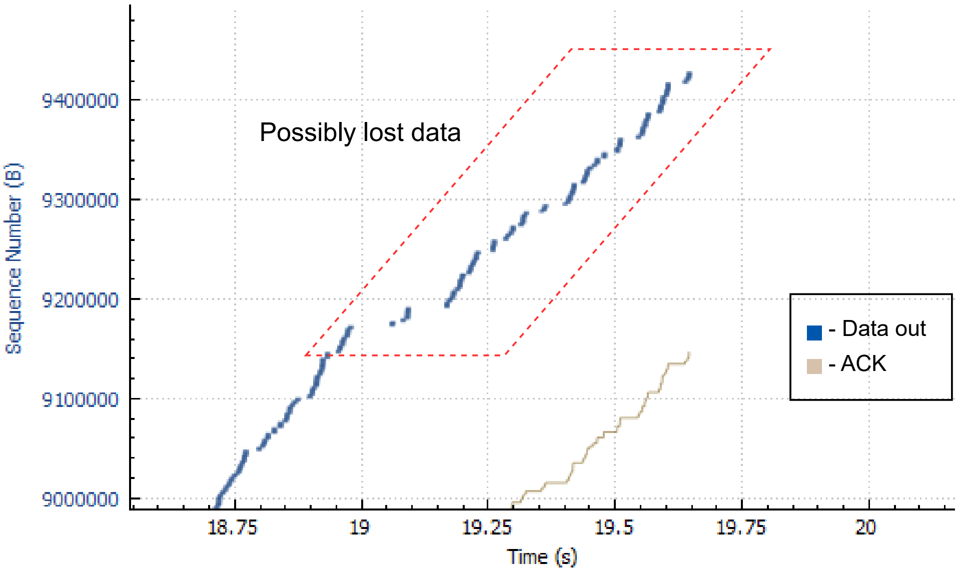
In Multipath TCP (MPTCP), a *subflow* refers to an individual TCP connection between two endpoints (typically a client and a server) within an MPTCP session. MPTCP extends the capabilities of traditional TCP by enabling the simultaneous use of multiple network paths or interfaces for a single connection. Each of these simultaneous connections over different paths is known as a subflow.

In MPTCP, a single high-level connection is established between the client and server. This high-level connection coordinates multiple subflows. Subflows provide robustness in case one path experiences congestion or packet loss. MPTCP can shift the traffic to other subflows, which may lead to better load balancing and improved user experience. MPTCP divides data into packets and spreads them across the active subflows. This enables parallel data transmission, improving the overall throughput and reducing latency. If a subflow experiences packet loss or out-of-order delivery, MPTCP can efficiently recover by using the techniques like reinjection and selective recovery, without the need to retransmit the entire data stream.

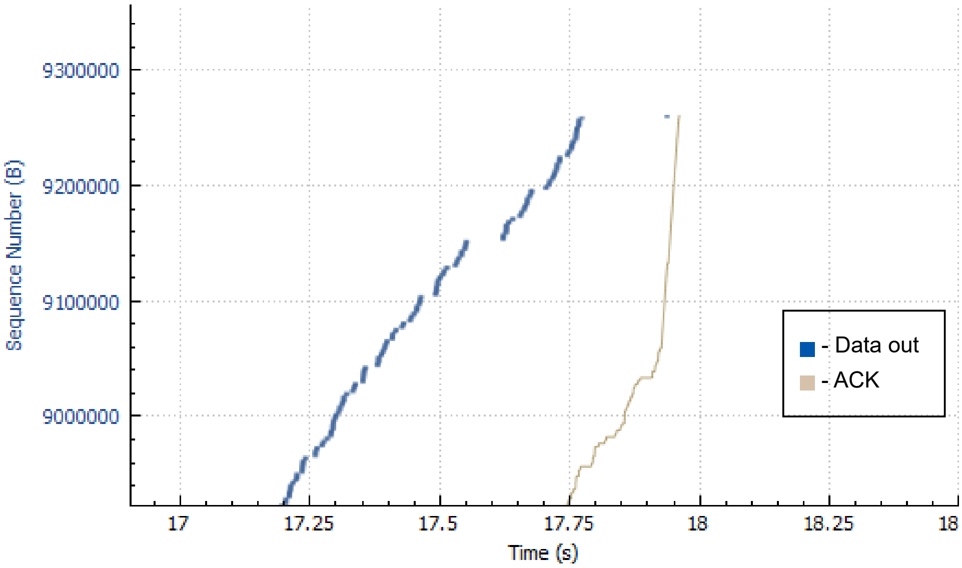
Each subflow behaves like a regular TCP connection, adhering to the TCP congestion control algorithms. They have their own sequence numbers, window sizes, and acknowledgments. In a TCP connection, data is transmitted in segments. When data is sent, the sender expects to receive acknowledgments from the receiver to confirm that the data was received correctly. These acknowledgments, often referred to as ACKs, indicate that a segment has been successfully received. The Data ACK have the similar functionality to the standard TCP cumulative ACK, signifying the amount of data that has been received without any gaps or missing portions. If the sender doesn't receive an acknowledgment for a particular segment within a certain timeframe, known as the retransmission timeout (RTO), it assumes that the segment was lost in transit. To ensure reliable data delivery, the sender retransmits the lost segment. This is a fundamental mechanism in TCP to recover from packet loss.

Now, when the vehicle moves out of the coverage area of RSU, the corresponding subflow is abruptly interrupted, there might be unacknowledged data that was in transit. This means data sent by the sender but not yet acknowledged by the receiver. If the connection is terminated without allowing the sender to retransmit unacknowledged data, this data is considered to be lost. The amount of data lost in this scenario depends on how much data was in transit when the connection was interrupted. It could range from a small amount (e.g., a few bytes) to a larger amount (several kilobytes or more). In typical scenarios, it's desirable to gracefully terminate a TCP connection to ensure that all data is correctly delivered and acknowledged. Abrupt terminations, such as in our vehicular scenario, can result in data loss, as show in Figure 19(a).

The amount of data lost in a TCP connection when it's interrupted abruptly depends on several factors, including the network conditions, the state of the connection, and when the interruption occurs. In a typical TCP connection, when data is sent, it's acknowledged by the receiving end. The sender will retransmit any data that wasn't acknowledged within a certain timeout period (based on the TCP retransmission timer). This means that if the connection is interrupted but later restored, the sender will retransmit the unacknowledged data, resulting in minimal data loss. However, if the connection is abruptly terminated without a chance to retransmit unacknowledged data, the data sent but not yet acknowledged will be lost. The actual amount of data lost can vary significantly from case to case. This could range from a few bytes to several kilobytes, depending on the size of the sender's congestion window and how much data was in transit at the time of the interruption.



(a) Cross-layer PM disabled



(b) Cross-layer PM enabled

Figure 19: Data transmitted over IEEE 802.11p link

MPTCP is designed to handle these situations through its retransmission mechanisms, but it's essential to ensure a graceful connection termination for optimal data delivery. As shown in Figure 19(b), our Cross-Layer PM ensure graceful connection termination before the link became unavailable, to minimize data loss and potential retransmission of unacknowledged data.

5.2 Seamless VHO

To ensure reliable, in-order delivery of data over subflows that may appear and disappear at any time, MPTCP uses a 64-bit Data Sequence Number (DSN) to number all data sent over the MPTCP connection. Each subflow has its own 32-bit sequence number space, utilizing the regular TCP sequence number header, and an MPTCP option maps the subflow sequence space to the data sequence space. This strategic mapping allows for the retransmission of data across different subflows, all linked to the same DSN, should any subflow encounter a failure or disruption. All subflows utilize a common receive buffer and present an identical receive window. MPTCP employs two tiers of acknowledgments. Firstly, it utilizes standard TCP acknowledgments on each subflow to confirm the receipt of segments sent over that subflow, irrespective of their DSN. Secondly, it employs connection-level acknowledgments that operate within the data sequence space. These acknowledgments monitor the progression of the byte stream and manage the sliding of the receive window.

The Data Sequence Mapping components are instrumental in reconstructing the entire data stream. They define how the subflow sequence numbers are mapped to the data sequence numbers. It is described in terms of the starting sequence numbers for both the subflow and the data level, along with the length of bytes for which this mapping remains valid. This mapping is crucial for ensuring that the data is delivered to the application layer in the correct order. In contrast, the sequence numbers at the subflow level (i.e., the standard sequence numbers in the TCP header) is relevant only to the specific subflow.

Figure 20 displays the average throughput per interface, and the total achieved throughput during the MPTCP session. MPTCP can utilize two parallel connections to spread a single data stream and increase available bandwidth. Consequently, the achieved total throughput is higher when the 802.11p and LTE interfaces are used simultaneously. The total throughput drops when the path through the ad hoc network is lost as a vehicle moves out of the RSU coverage area. However, a data stream continues to flow through the cellular network even when the 802.11p link becomes unavailable for packet transmission. Therefore, a moving vehicle can perform a VHO over cellular and roadside infrastructure without service interruption. Upon careful inspection of the packet traces, it is evident why the default MPTCP path manager is not optimal for vehicular networks. Figure 21 and Figure 22 illustrate the evolution of DSN over time when the vehicle moves out of the coverage area of the RSU and the 802.11p link (`subflow 1`) is lost.

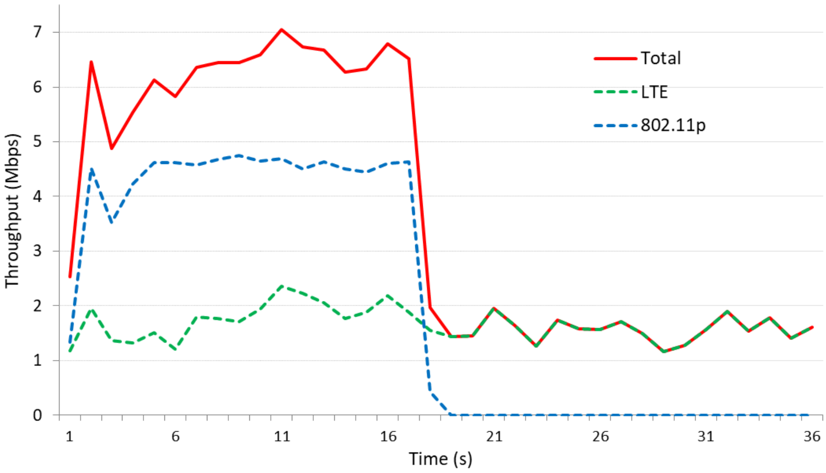
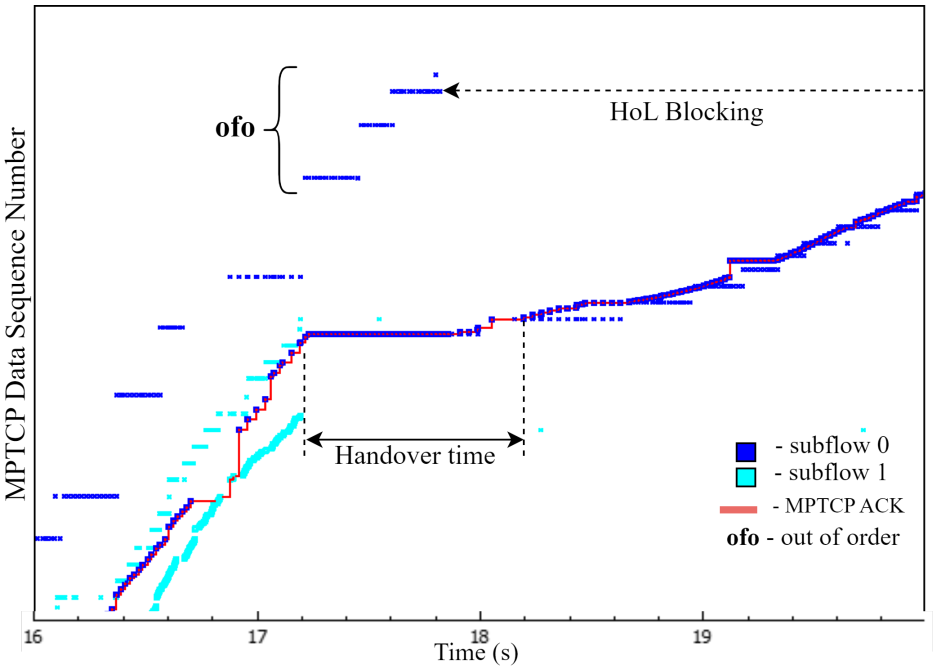


Figure 20: MPTCP aggregate network bandwidth to improve the throughput.

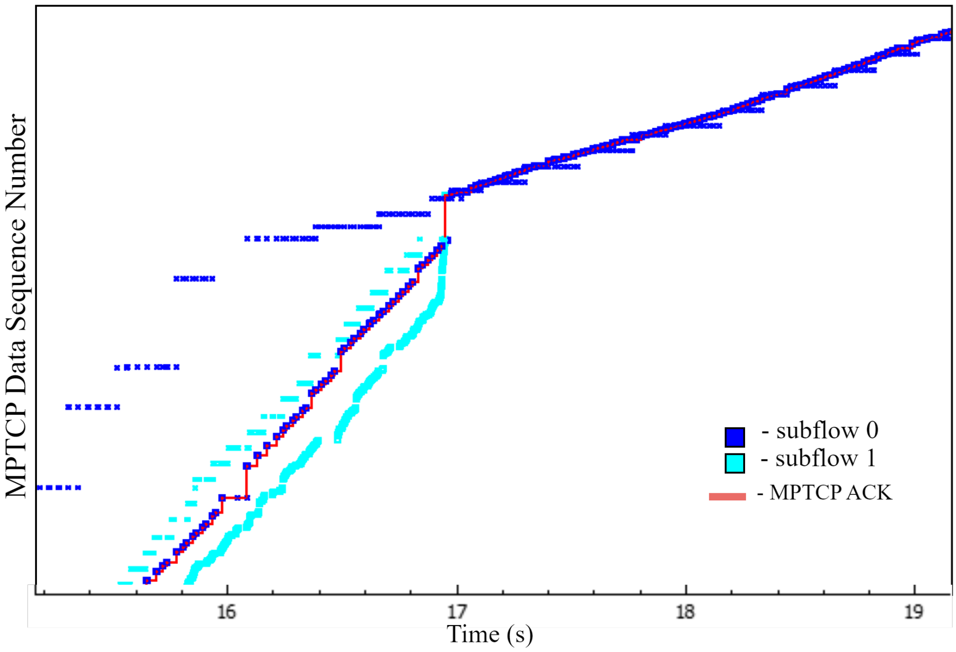
5.2.1 Aggregation mode

Initially, the connection's data is distributed among all available interfaces, with most of the data sent over the ad hoc network with the lowest Round-trip time (RTT). When the 802.11p link becomes unreliable and starts losing packets, the VHO process redirects the data flow to a more reliable path (i.e., `subflow 0`). The handover process is not instantaneous, as MPTCP requires time to detect a link failure and adjust the connection accordingly. This means that there can be a delay before the handover is complete, as shown in Figure 21(a). The packets through the LTE network (`subflow 0`) arrived out of order (ofo), which could lead to Head-of-line (HoL) blocking with more in-flight packets accumulated at the receiving end. Since MPTCP guarantees in-order data delivery, packets received on `subflow 0` must wait for previous segments with lower sequence numbers to arrive, resulting in glitches in the application layer. This can be particularly problematic for real-time applications such as video conferencing or online gaming, where even small delays can be noticeable and disruptive to the user experience. In addition, MPTCP's scheduling mechanism continues to allocate packets to `subflow 1` since it is not immediately aware of the path's status, and it may take several Retransmission Timeout (RTO) periods before the MPTCP recognizes that the path is unavailable. As a result, a significant amount of data can be lost when an interface fails, and the underlying path becomes unusable, requiring retransmission through an available subflow.

The proposed path manager aims to anticipate packet loss and suspend the unreliable subflow before the link eventually breaks. As the vehicle moves away from the RSU, a gradual drop in signal strength is observed in the transition region. The faster a vehicle moves away from the RSU, the more abrupt a drop in RSS. Thus, according to our algorithm, the subflow must be suspended as the RSS drops below the pre-defined threshold. The proposed PM with cross-layer assistance can detect and mitigate lousy path conditions by automatically removing the unreliable `subflow 1` from the ongoing connection before the interface becomes unavailable, as shown in Figure 21(b). Furthermore, our PM can dynamically reactivate the subflow when the path quality improves, continuously optimizing network performance by adapting to

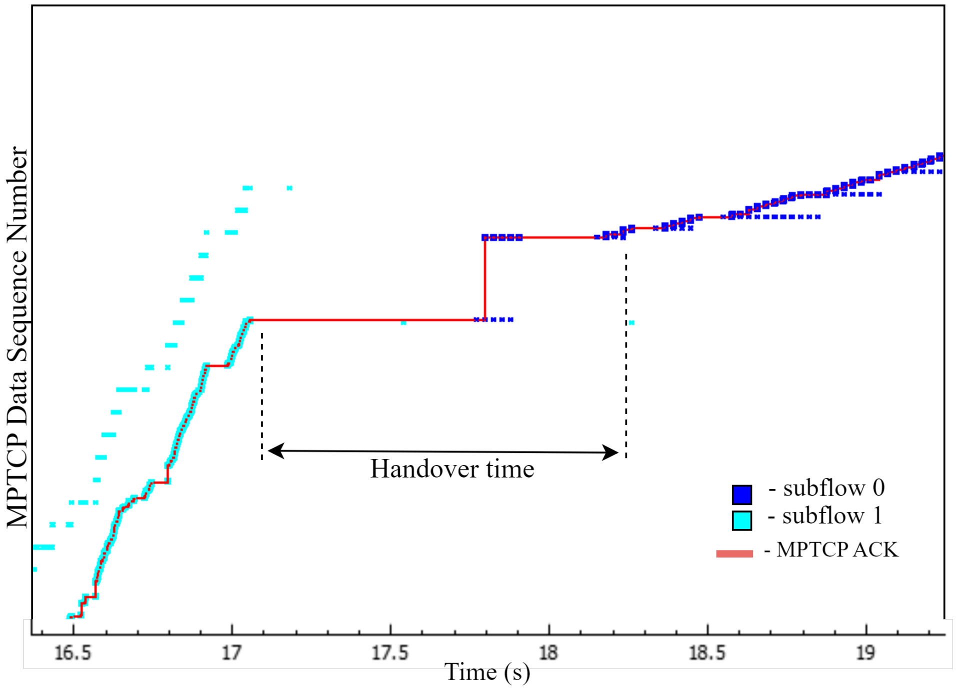


(a) Cross-layer PM disabled

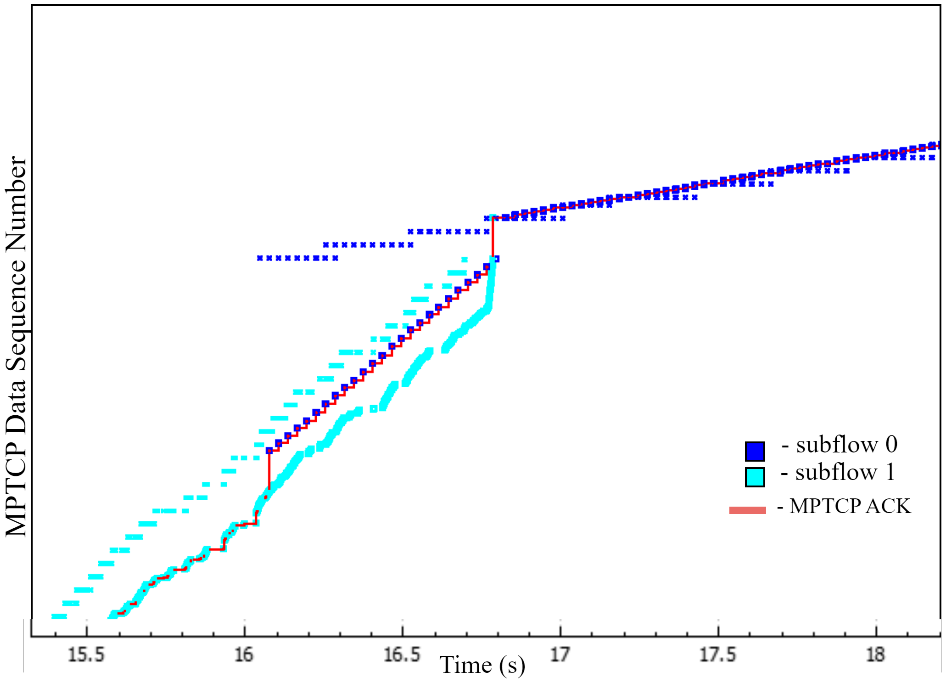


(b) Cross-layer PM enabled

Figure 21: Handover event in detail (aggregation mode).



(a) Cross-layer PM disabled



(b) Cross-layer PM enabled

Figure 22: Handover event in detail (backup mode).

changing conditions.

5.2.2 Backup mode

When MPTCP establishes a new connection, it can generate subflows that serve as backup paths, activated only in the occurrence of a failure in the primary subflow. For our next experiment, we set up the endpoints to use the WLAN network as the primary path and switch to the cellular network only when the WLAN link fails. Figure 22(a) illustrates how MPTCP activates the backup path without cross-layer assistance when the primary `subflow 1` fails due to the vehicle moving out of the RSU coverage. At the start of the MPTCP connection, the 802.11p interface is used as the primary path, while the LTE interface is designated as a backup. In case of poor radio conditions causing the primary path to fail, MPTCP detects the failure and switches the traffic flow to the backup path. However, this approach may not be optimal for vehicular scenario as it results in a delay during the handover process. Furthermore, there is a period where no subflow is active, which can negatively affect service continuity. The issue is that the protocol will continue to retransmit lost data over the broken interface because the subflow was not closed properly, thus worsening the situation.

The proposed cross-layer PM scheme employs a dynamic approach to subflow management. When the RSS falls below the configured threshold, it determines that the primary subflow is underperforming and takes corrective action. Initially, the PM removes the backup status of the cellular network, and for a short period, MPTCP uses both interfaces simultaneously. If the backup subflow was not established initially, an additional subflow is created over the cellular network to ensure redundancy. Subsequently, the PM handles the lossy path by switching it to backup mode, preventing it from being used for data transmission. The underperforming subflow can be removed from the MPTCP session if necessary. The proposed scheme significantly reduces handover delay by completing the VHO procedure while the ad hoc network is still available, as demonstrated in Figure 22(b).

5.3 Application layer delay

The applications with real-time requirements, such as media streaming, are most affected by frequent handovers and path variation, resulting in high data delivery latency and perceivable jitter. We generate traffic and include a timestamp in each data segment to monitor the variation in application delay. The end-to-end delay is measured from when the packet leaves the source application to when it arrives at the destination application. Upon reception of each data segment, we record its arrival time. We can determine the end-to-end application delay by tracking the difference between these timestamps. Our emulation setup uses the same hardware clock on each communication endpoint, ensuring accurate measurements.

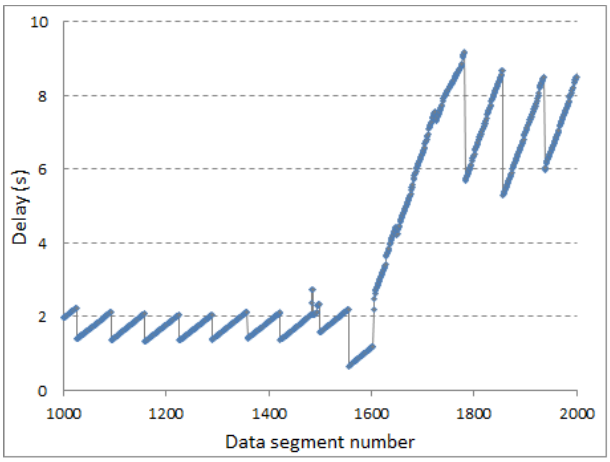
For assessing handover performance, we conducted tests involving unicast real-time applications utilizing MPTCP for communication between the vehicle and the remote server. The application involves the vehicle as the sender, transmitting, for instance, information collected from the vehicle's sensors to the server. These scenarios enable us to evaluate real-time application metrics, such as end-to-end delay, that represent the time taken for a packet to travel from the sender node to the receiver node. It's calculated as the difference between the reception timestamp at the receiver and the transmission timestamp at the sender.

Figure 23 compares MPTCP performance regarding application delay during the handover in a vehicular scenario for kernel v15.5, kernel v15.7, and our proposed cross-layer PM scheme. The results demonstrate that the application layer delay significantly increases when the handover occurs with a standard MPTCP scheme. However, the delay is significantly reduced when using our proposed PM scheme, indicating improved network performance and reduced service disruption during handovers. Following the failure of the IEEE 802.11p link, the end-to-end delay can increase significantly, reaching up to 9 seconds on kernel v15.5 (Figure 23(a)). This high application delay is primarily attributed to the very basic packet scheduler algorithm that fails to consider heterogeneous network characteristics. As a result, we can observe an inferior performance in data delivery speed to the application, as out-of-order packets suffer from prolonged queuing times before being delivered to the application. The MPTCP packet scheduler was improved with better HoL blocking estimation in Linux kernel v15.7. A scheduler algorithm was modified to handle heterogeneous paths better, as represented in Figure 23(b). Still, connection glitches occur when the vehicle moves out of coverage of RSU and a handover is experienced.

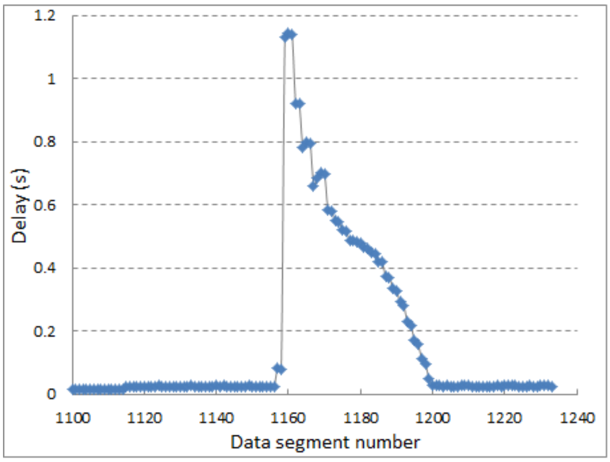
Figure 23(c) illustrates the end-to-end delay when our PM is enabled. The figure demonstrates that both subflows are utilized concurrently to transmit data, leading to a negligible variation in application delay until data segment number 1600 is sent. The slight increase in delay can be attributed to the heterogeneity of the network paths.During a VHO event, our proposed cross-layer PM scheme demonstrates its ability to mitigate end-to-end delay significantly. This is achieved by proactively suspending the underperforming subflow before link failure occurs. Such a strategy guarantees an uninterrupted user experience and optimizes overall network performance. Specifically, in the evaluated scenario, the average application delay at the moment of handoff decreases by an order of magnitude, from 1–10 seconds to 15-80 milliseconds. These outcomes affirm the effectiveness of our proposed approach in enhancing MPTCP performance and elevating network Quality of Experience (QoS) within vehicular environments.

5.4 Recovered path

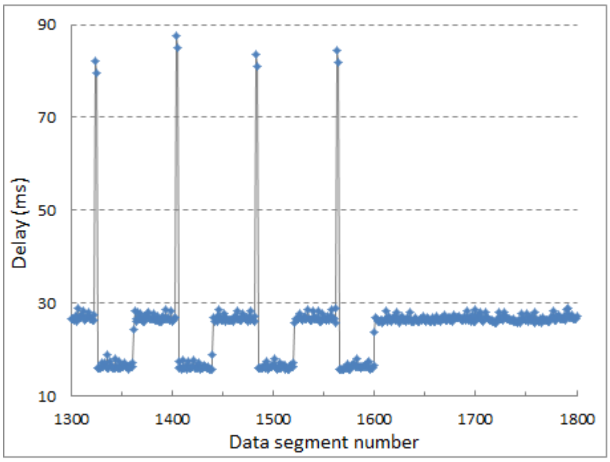
Each byte of data sent in an MPTCP connection has an associated sequence number so that an available subflow can recover lost packets. However, the regular TCP provides the transport service on each subflow, which will start a retransmission timer after sending a packet of data. If a sender does not receive an acknowledgment before the timer expires, it will assume the segment has been lost and retransmit it. When the sender misses too many acknowledgments, the MPTCP considers that packet loss has appeared somewhere between the vehicle and the remote host and decides to stop sending data over that path.



(a) Kernel v5.15



(b) Kernel v5.17



(c) Cross-layer PM enabled

Figure 23: Data delivery latency during vertical handover.

After some time, the protocol starts testing the path availability by sending one packet over the same path again. After the first attempt, the exponential backoff is applied to the RTO, doubling the timeout value between successive retransmissions. Once the 15th retry expires (by default), the subflow is considered broken. With regular MPTCP, there is nothing that the application can do to start using the recovered path other than waiting for successful retransmission. This unnecessary delay after the path recovers causes the vehicle to lose a significant fraction of available connection time.

As the vehicle comes in or out of range of a particular network, additional subflows can be created or removed dynamically. The proposed cross-layer PM can detect path loss or availability much faster due to signal strength monitoring of accessible networks. For instance, when the vehicle moves into the network's coverage area, it can utilize the wireless interface as soon as RSS rises above the predefined threshold. In long-lived data flows, this can result in throughput gains since there is no idle period after the path recovers, as shown in Figure 24. In this example, the additional subflow is created as soon as the 802.11p interface detects an RSU. Similarly, the subflow is removed from the MPTCP connection as the RSS falls below the receiver sensitivity threshold.

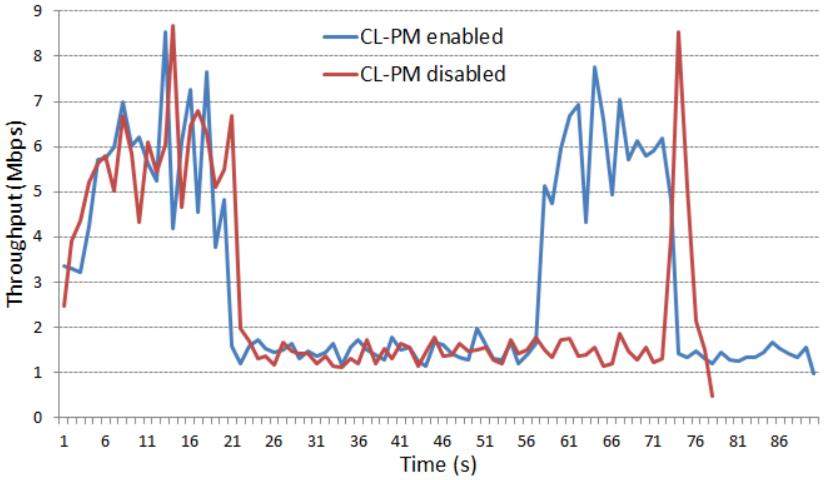


Figure 24: Achieved throughput with recovered path.

5.5 Urban mobility simulation

The cross-layer PM has been evaluated in a Manhattan-like urban scenario with bidirectional streets and one lane per driving direction. This generic urban scenario has been chosen to analyze the effectiveness of the proposed algorithm for V2I communication with different deployments of infrastructure nodes. All vehicles were equipped with LTE and IEEE 802.11p interfaces, running CAM service to exchange real-time information. We have generated the road traffic using the "randomTrips.py" script provided by SUMO. Regarding the mobility model, we have used the Intelligent Driver Model car-following model in order to generate random trips. We used the Dijkstra algorithm as path finding algorithm. The other infrastructure components, including RSUs, the correspondent server node, and SGW and PGW nodes, remained static throughout all simulations. This was achieved by applying a constant mobility model to these components. The scenario includes a single LTE base station covering the simulated area, with a certain number of RSUs uniformly distributed at random intersections. Due to limitations in the number of communication nodes available within the simulation and the resulting increase in simulation jitter, we were unable to assess the influence of the number of vehicles in the urban mobility scenario. However, it's important to note that packet reception in urban mobility can be influenced by the density of packets being transmitted, which in turn depends on the number of packets sent.

A vehicle can lose and regain connection when it crosses through different RSUs while moving on the road. There are regions around infrastructure nodes where the wireless networks exhibit poor link quality due to bad radio conditions. Therefore, the connection time between the vehicle and RSU is changes according to the vehicle's speed and traffic conditions. We conducted two identical simulations to compare the effectiveness of our cross-layer PM with the default MPTCP scheme. Each simulation lasted 30 minutes and involved uploading files of varying sizes from the vehicle to the server. In one test, we disabled our PM, while in the other, we enabled it. After completing the simulations, we compared the total amount of bytes transferred by each interface, as shown in Figure 25. It is evident from the results that our proposed scheme, aided by its efficient handoff mechanism, can utilize the available bandwidth more effectively than regular MPTCP. Furthermore, other experimental outcomes demonstrate that our PM substantially enhances the quality of experience for vehicular users regarding of achievable throughput, packet loss, retransmissions, and duplicate packets.

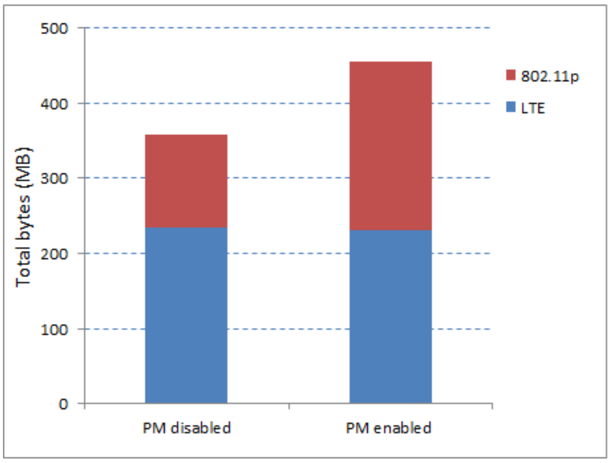


Figure 25: Total volume of transmitted data after 30 min of simulation.

Chapter 6

Intelligent path selection

The Vertical Handover (VHO) algorithm discussed in chapter 4 relies on a single Receiving Signal Strength Indication (RSSI) criterion, which might not always provide a comprehensive solution. Hence, a more sophisticated PM approach is required, incorporating complex policies that consider real-time network attributes and user preferences. Such an approach would demand a more intricate decision-making algorithm capable of dynamically adjusting to evolving network conditions and user needs.

The concept of employing intelligent interface selection empowers vehicular users to seamlessly switch between interfaces associated with different technologies based on the optimal fit for their application requirements. In practice, the choice of an interface for connectivity can depend on various Quality of Service (QoS) parameters, including throughput, delay, or other user preferences like cost-effectiveness. Consequently, having multiple wireless interfaces available ensures that services are delivered through the always best-connected user interface, ensuring a high-quality experience at all times.

An intelligent path management algorithm is proposed in this section to ensure seamless communication in dynamic vehicular environments. The algorithm uses a network selection mechanism based on the Fuzzy Analytic Hierarchy Process (FAHP), which dynamically assigns the most appropriate underlying network for each active application. The selection process considers factors such as path quality, vehicle mobility, and service characteristics. Unlike existing approaches, our proposed method introduces a dynamic and comprehensive approach to network selection that is tailored to the unique requirements of each service to ensure that it is always paired with the optimal access technology. By tailoring the network selection to the specific needs of each application, our path manager can provide optimal connectivity and performance, even in challenging vehicular environments, delivering a better user experience with more reliable connections and smoother data transfers.

6.1 Attributes to consider

The multipath paradigm can support various types of vehicular applications, including infotainment, real-time navigation, and remote diagnostics, which makes it particularly valuable for Vehicle to Network (V2N) communication [55, 57]. While the proposed cross-layer strategy yields significant improvements in data delivery latency and throughput, it employs a Path Manager (PM) algorithm based on a single criterion (i.e., RSS), which may need to be revised to provide a comprehensive solution. To ensure service continuity for vehicular users, an advanced path management strategy is required, one that considers various parameters such as vehicle mobility and real-time network attributes. This strategy should make dynamic and adaptive decisions in response to changing environmental conditions. Furthermore, the decision algorithm should take into account the service pricing and the specific QoS requirements of different applications [99].

Network parameters such as available bandwidth,Packet Loss Ratio (PLR), packet sending delay, and jitter are critical factors that impact the performance of any connection. These factors should be taken into account in the decision-making process for network selection. Additionally, context information about the vehicle, including its speed, location, and movement direction, can play a significant role in this selection process, and this information is readily available through the vehicle's On-Board Unit (OBU).

6.1.1 Latency

Increasing latency poses significant challenges to TCP-based subflows, particularly impacting their data transfer rates and the performance of real-time applications. As latency in a network increases, TCP-based subflows face a notable reduction in their data transfer rates. This slowdown occurs because high latency introduces delays in transmitting and acknowledging data packets. These delays, in turn, hinder the rapid expansion of the TCP congestion window, limiting the achievable throughput. Consequently, the overall data transfer process becomes less efficient and slower. High latency affects the data transfer rate and introduces delays in the data transmission process. These delays can lead to underutilization of available network resources and decreased throughput. Applications that rely on steady, high-throughput connections, such as file downloads and large data transfers, may experience performance degradation. The adverse effects of latency become especially prominent in real-time applications, where responsiveness and low delay are critical. For instance, high latency can result in noticeable lag or delay between player actions in online gaming. In video conferencing applications, latency can cause audio and video synchronization delays, leading to disjointed conversations and difficulties in understanding and responding to others. For live-streaming platforms, latency can result in buffering issues, where viewers experience delays in the live feed or video playback.

Low latency is essential for ensuring a seamless and responsive user experience. High latency can lead to frustration, reduced user engagement, and, in some cases, a complete breakdown in communication or interaction. The quality and usability of real-time applications heavily depend on low latency. Applications designed for real-time communication, collaboration, and entertainment demand minimal delay to deliver a satisfactory user experience. Various techniques, such as traffic prioritization, QoS mechanisms, and edge computing, are employed to reduce latency and enhance network responsiveness.

6.1.2 Jitter

Jitter, in the context of networking, refers to the unpredictable variation in the delay between packets as they traverse the network. In simpler terms, it represents the inconsistency in the time it takes for data packets to reach their intended destination. Jitter can stem from many reasons, including network congestion, fluctuations in the distances between network nodes, differences in the processing times of various devices along the data path, and even variations in the routing of packets. This variability in packet arrival times can have notable implications, especially for applications susceptible to latency and timing precision. Real-time communication services like voice and video calls, online gaming, and live video streaming depend on maintaining a consistent and low-latency network environment. When jitter disrupts the regular arrival of data packets, it can lead to several issues. For instance, in a voice or video call, excessive jitter can result in choppy or distorted audio and video quality, making it challenging for participants to understand each other or follow the content being shared. In online gaming, jitter can lead to lag, causing players to experience delayed responses to their actions, which can be frustrating and disadvantageous. Even for non-real-time applications, jitter can impact data transfer speeds and the overall reliability of network connections. Reducing jitter at the path can ensure smoother and more reliable network experiences, especially when timing and responsiveness are critical.

6.1.3 Packet Loss Rate

PLR is a crucial metric in network performance evaluation. It reflects the proportion of data packets that do not successfully reach their intended destination during transmission across a network. High PLR can have far-reaching implications, affecting data integrity and overall network efficiency. When a network experiences a high PLR, it implies that a significant portion of data packets lost in transit. This loss can result from various factors, including network congestion, hardware failures, or errors in data processing. The consequences of packet loss are multifaceted and depend on the specific application and transmitted data. One immediate consequence of packet loss is the potential for data corruption. When packets are lost, the information they contain is missing from the received data, leading to gaps and inconsistencies. In response, mechanisms such as re-transmission come into play to maintain data integrity.

However, re-transmitting lost packets introduces additional delays and can reduce overall throughput. QoS considerations are intimately tied to packet loss rates. The acceptable level of packet loss varies depending on the nature of the data being transmitted and the application's requirements. For instance, in a Voice over IP (VoIP) conversation, the occasional loss of one or two packets may not significantly impact the conversation's quality, as modern VoIP protocols often compensate for such minor disruptions. However, when packet loss rates rise to levels between 5% and 10%, the quality of the conversation can be noticeably degraded, leading to choppy audio and communication difficulties. In light of these considerations, it becomes apparent that measuring instantaneous packet loss is a valuable endeavor, especially when aiming to implement an efficient PM scheme. Our network selection algorithm can gain critical insights into network performance by closely monitoring and quantifying packet loss rates. This data allows them to make informed decisions and implement strategies to mitigate packet loss, ultimately enhancing the reliability and quality of network services.

6.1.4 Received Signal Strength

The access layer of a network infrastructure plays a pivotal role in providing critical status indicators that guide decision-making processes, particularly in scenarios like seamless handover between networks. These status indicators encompass a range of metrics, including the busy channel rate, Receiving Signal Strength (RSS), and frame transmission statistics. Many existing solutions primarily rely on the RSS measurement to initiate a handover. However, this approach typically results in a significant disconnection time, commonly known as handover latency. Such latency negatively affects the performance of applications, leading to issues like packet loss. This packet loss can be particularly detrimental to certain types of applications. While RSS is a fundamental parameter for assessing link quality, it's essential to recognize that it may not always offer a complete picture of the wireless link's health. RSS, typically measured in dBm (decibels-milliwatts), is a vital metric because it directly reflects the strength of the received signal. A high RSS value generally signifies a robust and reliable signal, making it a commonly utilized factor in handover algorithms.

However, relying solely on RSS can have limitations because other variables can significantly influence the overall quality of a wireless link. One substantial factor to consider is interference. Even if the RSS is strong, interference from various sources, including other devices or electromagnetic signals, can weaken the signal quality. Consequently, interference can introduce packet loss, latency, and jitter, impacting data transmission performance. The distance between transmitting and receiving devices also plays a significant role in link quality. In wireless communications, signal strength diminishes as the distance between devices increases. Therefore, even if the RSS is initially high, it may degrade as devices move farther apart.

The choice of wireless technology is another critical determinant of link quality. Different wireless technologies like Wi-Fi, Bluetooth, or cellular networks have varying characteristics and operate on different frequency bands. Each technology has strengths and weaknesses, affecting parameters like coverage range, data rate, and interference resilience. Physical barriers between devices can also affect link quality. Obstacles like walls, buildings, or natural terrain can attenuate signals, leading to signal loss or degradation. RSS measurement helps gain a more comprehensive understanding of link quality. These include assessing metrics like the Bit Error Rate (BER) and Signal-to-Noise Ratio (SNR). The BER quantifies the accuracy of data transmission by measuring the rate of received bits with errors. A lower BER indicates higher data transmission accuracy. On the other hand, SNR represents the signal power ratio to the noise power in the received signal. A higher SNR typically indicates a cleaner and more reliable signal.

While RSS is a valuable metric for assessing link quality, it should be considered alongside other factors like interference, distance, wireless technology, and physical barriers. A comprehensive evaluation of these variables provides a more accurate basis for making decisions related to network handover and ensuring the seamless and uninterrupted performance of vehicular communications.

6.1.5 Dwell time

Vehicular networks represent an emerging paradigm connecting vehicles and with the road infrastructure. However, their high degree of mobility and dynamic topology changes pose significant challenges in predicting the duration of a vehicle's network connection. This difficulty arises from a multitude of parameters that come into play. Among these parameters, we can consider factors such as vehicle velocity, direction, traffic conditions, network signal strength, distances to access point, signal interferences, obstacles, multipath fading, transmission range, transmission power, power gain, among others.

Dwell time, often referred to as residence time in network management, is a crucial metric that quantifies how long a mobile device, such as a vehicle equipped with communication capabilities, remains connected to a specific network. In the context of vehicular communication, understanding dwell time is essential for optimizing network management, particularly in scenarios involving fast-moving vehicles and dynamic network handovers. Dwell time plays a crucial role in the seamless functioning of mobile communication systems, especially within vehicular networks. When a vehicle establishes a connection with an infrastructure node, such as an Evolved Node B (eNodeB) in cellular networks or an Roadside Unit (RSU) in vehicular communication systems, this connection is maintained for a certain duration. This duration is what we refer to as dwell time.

Dwell time is closely tied to a vehicle's movement within the network's coverage area. As the vehicle moving, it continuously evaluates its connection to the network node. The dwell time represents the average duration the vehicle remains attached to this node. Several factors influence dwell time: The speed at which a vehicle is moving can significantly impact dwell time. In scenarios where vehicles travel at high speeds, such as on highways, dwell times tend to be shorter. This is because vehicles move in and out of network coverage areas rapidly.

The density of network infrastructure in a given area is another crucial factor. In regions with a high concentration of base stations or RSUs, vehicles are more likely to have shorter dwell times as they transition between adjacent nodes. Conversely, dwell times may be longer in areas with sparse infrastructure, as vehicles have fewer options for network handovers. The efficiency of network horizontal handover mechanisms also influences dwell time. Effective handover strategies aim to minimize disruptions during the transition from one network node to another. Therefore, dwell time must be considered for the specific services and applications being used within the vehicle. Measuring dwell time is essential for providing uninterrupted and reliable vehicle connectivity in dynamic vehicular environments.

6.1.6 Cost

Service pricing is an important aspect that should be considered in designing an efficient network selection algorithm that involves the cost of utilizing various services and network technologies. Several factors can influence service pricing, and its consideration in a PM algorithm is critical for optimizing connectivity decisions. Service providers typically offer a range of services with varying data speed, latency, and reliability levels. These services can encompass different generations of cellular networks (e.g., 4G and 5G), Wi-Fi, and specialized vehicular communication services. Each of these services may come with its pricing structure based on factors like data usage, bandwidth, and quality of service. Service pricing can be dynamically influenced by user demand and contextual considerations. For instance, during peak hours or in densely populated areas, the cost of accessing a particular service might increase due to high demand. Conversely, prices may be more competitive in areas with low user activity.

Different applications and services have distinct QoS requirements. For example, safety-critical applications demand low latency and high reliability, while infotainment services prioritize bandwidth for high-quality video and audio streaming. The cost of a service may vary based on the level of QoS it can guarantee. Service providers often offer subscription plans with different pricing tiers. These plans may include data caps, data throttling, or prioritization of specific types of traffic. Users' choice of service plan can impact the cost-effectiveness of utilizing a particular network or service.

Roaming charges can significantly affect service pricing for vehicular networks that involve mobility across geographic regions. Data usage and network access costs may change when a vehicle moves from one service provider's coverage area to another. Some service providers employ dynamic pricing models that adjust based on real-time network conditions and demand. Such models can incentivize network selection algorithms to shift application data consumption to off-peak hours or less congested networks. An effective PM algorithm must perform a cost-benefit analysis when selecting the optimal connection for a vehicle. This involves considering the cost of accessing a particular service or network against its benefits regarding data speed, reliability, and other QoS parameters. This ensures that network selection aligns with the user's preferences, service requirements, and budget constraints, ultimately enhancing the overall connectivity experience in vehicular environments.

6.2 System Model

This section introduces the generic system model for the proposed path management solution, which leverages multiple access technologies to facilitate ubiquitous communication in the context of the Internet of Vehicles (IoV), with a specific focus on V2N communication. The existing Intelligent Transportation Systems (ITS) architecture necessitates the use of a multipath protocol to effectively employ heterogeneous vehicular networking. In this section, we introduce an evolution of the ITS communication architecture aimed at simplifying the implementation of such heterogeneous vehicular networks. This proposed architecture aligns seamlessly with the current ETSI ITS station reference architecture while enabling the dynamic selection and coordination of available Radio Access Technologies (RATs), based on contextual conditions and the specific requirements of the applications in use.

6.2.1 Service Profiles

The requirements and preferences for key network attributes, such as latency, bandwidth, delay variation, error rate, and other factors, will likely differ among various applications. For example, applications like voice and video calls require stringent network attributes. They demand low packet delivery latency, meaning packets must be delivered swiftly with minimal delay to provide a smooth and uninterrupted conversation. Additionally, low jitter is essential to ensure that audio and video data arrive consistently, avoiding disruptions in communication. On the other hand, applications like web browsing, while more tolerant of latency, place a premium on high packet delivery rates. It requires rapid transfer of a substantial amount of data to maintain acceptable throughput. This enables users to browse the internet without experiencing significant lag or delays when loading web pages and content. In transportation services, especially for applications like remote driving, data transmission with minimal delay is paramount. This ensures the vehicle can be controlled in real-time, enabling swift responses to user inputs. Stable signal quality is also necessary to avoid disruptions that could compromise safety during remote driving experiences.

One limitation of traditional network selection schemes is that in the event of a handoff, all established connections on the current network must disconnect and reconnect on the next access network. This can be problematic in multipath communication, as all created subflows on a particular interface must switch to another interface, even if the previous network provided better performance for a specific service. Unlike many other proposed solutions, this work presents a novel PM scheme that considers vehicle mobility, network attributes, and service QoS requirements to optimize network utilization for each application individually. For this purpose, applications are categorized into distinct groups based on their specific requirements. The three service profiles considered in this thesis are *Conversational* (e.g., VoIP or video chat), *Streaming* (e.g., video streaming and entertainment), and *Background* (e.g., web browsing, file transfers). The PM procedure is executed separately for each application group. This enables tailored handoffs specific to the traffic class and ensures consistent performance for data flows across different service types. Tailored network selection guarantees a high-quality user experience for various types of network traffic. For instance, during a phone call, the system can select a low-latency network for uninterrupted communication. At the same time, background applications can use networks with longer communication breaks, reducing packet losses and costs. By optimizing network selection based on the unique needs of each application group, this approach enables efficient use of network resources. Each application gets the most suitable network access, enhancing overall network efficiency.

6.2.2 Multipath in ITS station architecture

The European Telecommunications Standards Institute (ETSI) ITS station reference architecture is a framework designed to provide standardized guidelines for the deployment of ITS in the context of vehicular communication and transportation systems. It defines the architectural components, interfaces, and interactions between different entities within an ITS station, ensuring interoperability and seamless communication among various devices and systems. It describes the communication layers used for vehicular communication, encompassing the physical, data link, network, and application layers. The architecture defines different facilities within an ITS station, such as the communication management facility, security facility, and application support facility. It defines security services and protocols for secure communication. The ETSI ITS station reference architecture promotes interoperability by providing standardized interfaces and protocols, allowing different vendors and stakeholders to develop and deploy compatible ITS systems. The architecture is designed to be scalable to accommodate future technologies and applications, making it suitable for long-term deployment.

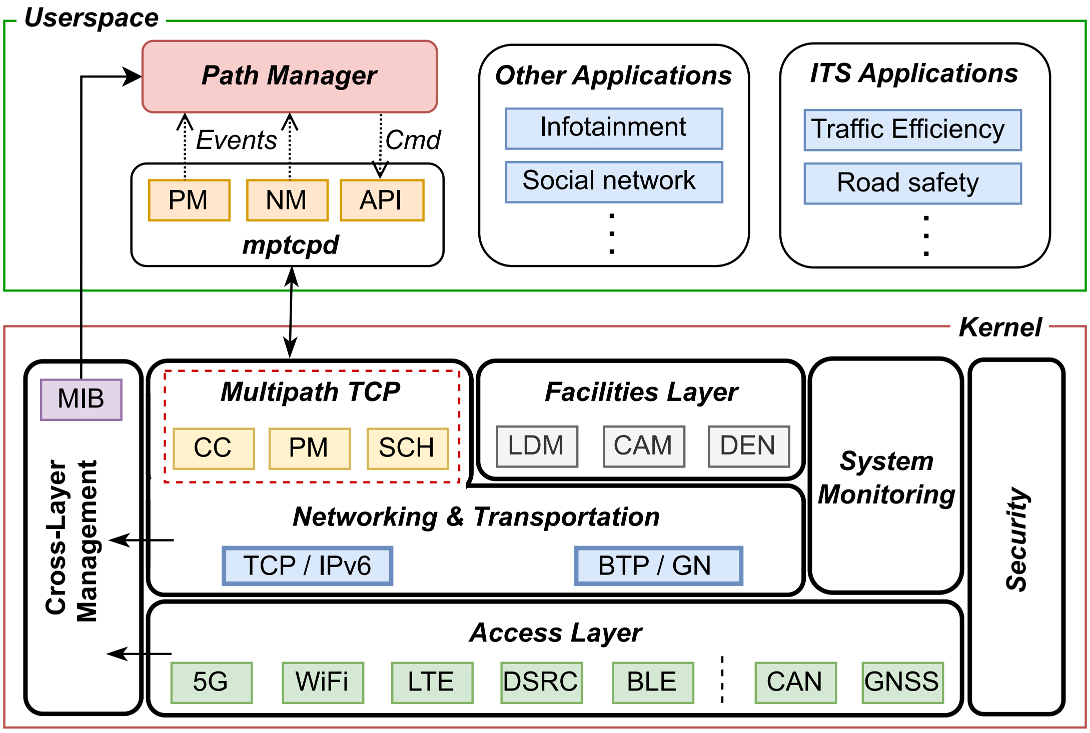


Figure 26: The *userspace* path manager in ITS architecture.

Multipath TCP make a part of Networking & Transportation layer of the ITS station reference architecture. The *userspace* PM interacts with kernel through the *netlink* IPC mechanism.

The ITS reference architecture follows the principles of the OSI model for layered communication protocols which is extended for inclusion of ITS applications [152]. ITS typically rely on the exchange of information between different layers of the communication stack to improve traffic management, enhance safety, and reduce congestion. This architecture takes into account various protocols at the network and transport layers, along with diverse communication technologies. The full ETSI ITS station architecture comprises the following layers: *Access*, *Networking & Transportation*, *Facilities*, *Applications*, *Security* and *Management*, as illustrated in Fig. 26. Each layer is responsible for specific functions, and the data exchanged between layers are often encapsulated in different protocols. It effectively abstracts applications from the intricacies of communication technologies, network configurations, and transport protocols. Within this framework, a Facilities layer encompasses a collection of shared functionalities and data structures that facilitate cooperative vehicular applications and communications. Additionally, cross-layer security mechanism provide robust security and privacy protection. To manage cross-layer information, ITS systems must ensure that each layer can access the relevant information from the layers below it, and that the information is appropriately translated into the relevant protocol format. Therefore, the cross-layer management provide a shared Management Information Base (MIB).

Multipath TCP (MPTCP) is a versatile protocol that has been successfully implemented in the Linux Kernel. It offers the ability for an application to simultaneously send data over multiple IPs and interfaces, without requiring any additional effort from the application itself. This means that MPTCP operates transparently to the application layer, and applications can continue to use the standard socket API as they normally would. The modular architecture of MPTCP supports different path management (PM), congestion control (CC), and packet scheduling (SCH) strategies. When multiple communication paths are available, the packet scheduling method decides which data segments should be transmitted over which subflow. Meanwhile, the PM module controls the use of subflows, with users able to configure specific operational modes for each MPTCP endpoint [153].

Despite this, MPTCP utilizes an unmodified socket interface, and the PM module that regulates the creation and removal of subflows located entirely within the kernel. As a result, MPTCP's standard socket API lacks the ability for dynamic subflow control, i.e., subflows are created immediately following connection creation or when an interface becomes active. In [150], the authors propose to separate MPTCP data plane from the control plane by moving all PM functions into the application layer. The control plane handles functions related to subflow management, without the need for the kernel to manage complex policies, while the data plane is responsible for transmitting and receiving data.

The MPTCP daemon (`mptcpd`) have already been presented in Chapter 4, enabling the development of customized PM strategies in userspace [92]. The mptcpd provides an interface that allows userspace applications to send control messages to the in-kernel path manager, offering a high degree of control over MPTCP connections. To interact with the kernel and obtain information about MPTCP events, the `netlink` inter-process communication mechanism is utilized. This strategic utilization of the netlink interface ensures swift and efficient data transmission between the kernel and userspace. Consequently, real-time monitoring and precise control of MPTCP connections become readily achievable. Furthermore, the MPTCP upstream implementation [91] offers an interface for PM-related operations, automatic endpoint configuration, and tracking per-connection information.

Mptcpd assumes the role of orchestrating all events associated with the path management. This encompasses the seamless delivery of notifications concerning connection initiation, termination, subflow creation, and address advertisements, among other pertinent events. Furthermore, mptcpd is responsible to encompass network interface events by reporting the emergence of new interfaces and the removal of existing ones. This comprehensive event management ensures the efficient orchestration of MPTCP connections and network interfaces. The necessary cross-layer information can be obtained from the MIB and delivered to the userspace in real-time. This information is then used to perform path management for each running application, as we will illustrate in the following section. Once the output from the network selection procedure is generated, the userspace PM can alter the state of MPTCP connection using a set of available commands (Table 4). These commands include requesting to establish a new subflow, modify subflow priority, or remove an existing subflow, as described in Chapter 4.

6.2.3 The proposed PM algorithm

In contrast to the in-kernel implementation, which controls all connections using global settings, the userspace PM enables individual control of each ongoing subflow. This allows our PM system to dynamically adapt to changes in the vehicular environment and ensure targeted QoS performance for each application. In our proposed approach, we operate under the assumption that the vehicle will maintain predefined criteria for each application profile. These criteria are accompanied by threshold values that act as triggers for modifications in the operational mode of subflows. Additionally, it's important to note that, in this study, we focus solely on V2N communication utilizing one subflow per active interface for connection with the remote host. The system is designed to improve MPTCP reactivity by assessing the suitability of paths for specific applications in advance, allowing for optimal resource utilization and minimizing the impact of vehicle mobility. Our main objective is to achieve seamless connectivity and optimize performance, while minimizing latency and data throughput loss related to handovers. The proposed PM scheme is outlined in Algorithm 2.

Our PM scheme can take into account the path quality and adjust subflow usage based on a score attributed to each available network. The score is calculated using an algorithm based on the Multiple-criteria decision-making (MCDM) method, which will be explained in the next section. The resulting score matrix, M, is of size $L \times N$, where L represents the number of service profiles and N represents the number of available networks, is used as an input for PM algorithm. Additionally, threshold values for changing the subflow backup status (T_b) and removing subflows (T_r) must be provided.

A vehicular user may experience multiple vertical handovers during an MPTCP session, frequently connecting and disconnecting from heterogeneous wireless networks along its path. The proposed PM algorithm enables dynamic creation, removal, or modification of the subflow priority, providing control over handover execution in fast mobility scenarios. Unlike the default MPTCP scheme, our PM does not create additional subflows immediately after the connection is established. Rather than using all available IP addresses to create a `full mesh` by default, a new subflow is only created if the path receives a good score from the network selection algorithm. This approach is more flexible than the existing in-kernel PM, which tries to create additional subflows immediately after the connection is established. Once established, the subflow is marked as `active`. If the network score falls below the T_r, the subflow is removed from the connection and marked as `inactive`. However, the PM retains relevant information about the removed subflow so that it can re-establish the subflow when the score increases above the T_r

Algorithm 2: Pseudo-code for intelligent path management

```
Input: $M = [s_{ij}]_{L \times N}$, $T_b$, $T_r$
for each subflow as $SF$ do
    $i \leftarrow$ get service type of SF
    $j \leftarrow$ get network used by SF
    $prio \leftarrow$ get SF priority
    $score = s_{i,j}$
    $bkp = false$
    if $score \leq T_b$ then
        $bkp = true$
    end
    if $score \leq T_r$ and $SF$ is active then
        set $SF$ as inactive
        remove $SF$ from connection
    end
    else if $score > T_r$ and $SF$ is not active then
        set $SF$ as active whith priority = $bkp$
        add $SF$ to connection
    end
    else if $SF$ is active and $prio \neq bkp$ then
        set $SF$ priority = $bkp$
        change $SF$ priority in connection
    end
end
```

again. A similar approach is used to determine the priority of the subflow, meaning that a subflow will be used in backup mode when it is active but has a score value less than the backup threshold (T_b). Note that since subflows from the same connection share a unique token, all of them will be removed after the connection terminates.

6.2.4 Signaling flow chart

To provide a more detailed explanation of the path management process, let us consider a straightforward V2N scenario. In this scenario, the vehicle in motion establishes communication with a remote server via MPTCP utilizing both LTE and 802.11p interfaces. The signaling flow chart for the connection is depicted in Fig. 27 and explained in the following sections:

6.2.4.1 Establish connection

To establish an MPTCP connection, the process closely resembles the standard TCP connection initiation. However, in this case, the SYN, SYN/ACK, and initial ACK packets, along with any accompanying data, are enriched with the presence of the $MP\_CAPABLE$ option. This option indicates that the sender is capable of performing Multipath TCP and wishes to use it for this particular connection. It facilitates the exchange of essential information between hosts to authenticate the establishment of additional subflows.

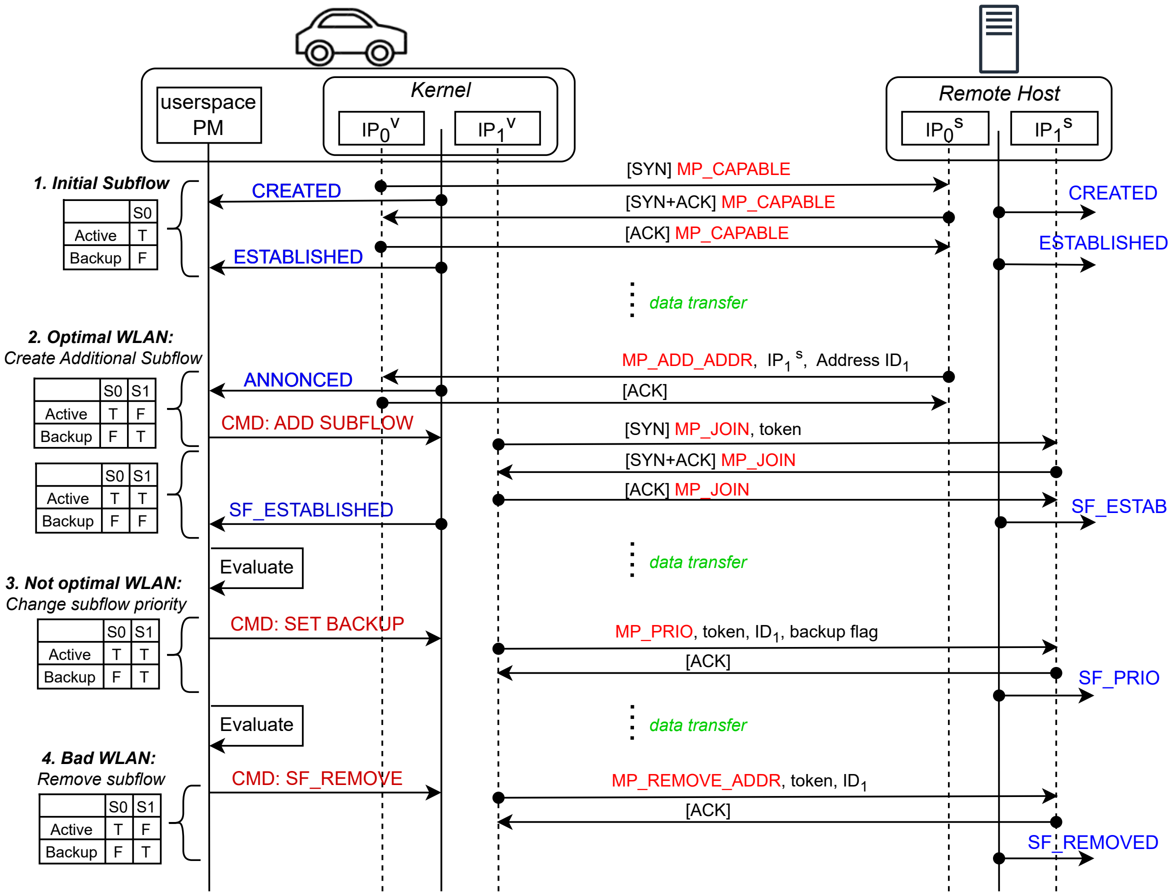


Figure 27: Signals flow diagram of multipath V2N connection.

The initial SYN packet, which contains only the $MP\_CAPABLE$ header, is used to specify the version of MPTCP being used and to exchange flags for negotiating connection features. Furthermore, this option is used to declare the 64-bit keys that the vehicle and remote host have generated for this MPTCP connection. These keys are used to authenticate the addition of future subflows to this connection. This is the only time the key will be sent in the clear on the wire. All future subflows will identify the connection using a 32-bit *token*, which is a cryptographic hash of this key. The algorithm used for this process depends on the authentication algorithm selected. Each host will index connections using the generated token.

Firstly, the vehicle initiates the MPTCP connection through LTE. This involves establishing an initial subflow between IP_0^v (the vehicle's IP address) and IP_0^s (the server's IP address) using a three-way handshake procedure. During this handshake, keys are exchanged in the $MP\_CAPABLE$ message to provide authentication material for future subflow setup. As a result, events are triggered on the PM system, and information about the initial subflow (S_0) is added to the subflows table with `active` and `non-backup` flags. Once the subflow is established, the vehicle can start transmitting data on subflow S_0. When a SYN packet includes an $MP\_CAPABLE$ option but the subsequent SYN/ACK packet omits it, the assumption is made that the sender of the SYN/ACK is not multipath capable. Consequently, the MPTCP session is required to function as a standard single-path TCP session.

6.2.4.2 Create additional subflow

After the initiation of an MPTCP connection through the $MP\_CAPABLE$ exchange, it is possible to incorporate additional subflows into the connection. Vehicle and remote host possess awareness of their own addresses and can acquire knowledge of additional addresses of each other through signaling exchanges. With this information, the vehicle can establish a new subflow over a currently unused pair of addresses. Therefore, a remote host informs the vehicle of an additional address without initiating a new subflow. It communicates its alternative IP address (IP_1^s) and port pair to the vehicle. The vehicle retains the option to later dispatch an MP_JOIN request to this newfound address. Each address is associated with an Address ID, which serves as a unique identifier for that address within a connection. This Address ID not only enables the distinct identification of addresses for removal but also used for subflow creation.

The ADD_ADDR option serves the purpose of declaring additional addresses, through which a host can be connected. This option can be employed at any point during a connection, as the sender wants to announce the availability of new paths. The ADD_ADDR option is integral in this process as it contains an HMAC for authenticating the address's source, ensuring it originates from the original connection. The remote host sends an ADD_ADDR packet to the vehicle, containing its alternative IP address (IP_1^s) and authentication data. The vehicle echoes the packet back to the server to confirm successful receipt. As a result, the `ANNOUNCED` event is triggered on the PM system, providing the IP addresses and ID of the endpoint announced by the remote host. When adding the advertised address to the subflows table, it is initially marked as *non-active* and classified as a *potential* subflow (S_1), without any actual subflow being established.

Periodically, the PM system collects cross-layer attributes and runs the network selection algorithm to assess the quality of available paths. If optimal WLAN conditions are detected, the PM instructs the kernel to establish a new subflow. The key exchange during the $MP\_CAPABLE$ handshake provides the information needed to authenticate the endpoints when setting up new subflows. The process of setting up additional subflows is similar to the typical procedure for establishing a standard TCP connection. However, in this case, the SYN, SYN/ACK, and ACK packets are additionally equipped with the MP_JOIN option.

When a vehicle initiates a new subflow between one of its addresses and one of the remote server's addresses, a unique token, generated from the encryption key, is used to identify the specific MPTCP connection it is joining. To ensure security and authentication, a Hash-based Message Authentication Code (HMAC) is employed. This HMAC leverages the keys established during the $MP\_CAPABLE$ handshake and the random numbers (nonces) exchanged through the MP_JOIN options. Additionally, the MP_JOIN packet encompasses flags and an Address ID, serving as a reference to the source address, all the while accommodating the dynamic nature of network address translation (NAT) changes, without requiring the sender to be aware of such alterations.

The PM then attempts to create another subflow by sending an SYN packet from IP_1^v (the vehicle's WLAN IP address) to IP_1^s (the server's alternative IP address). The new subflow starts as a normal TCP connection, but with the MP_JOIN option and token to identify which MPTCP connection it is joining.

Once the new subflow is established, the `ESTABLISHED` event is triggered, and the subflow status is changed to `active` and `non-backup`. At this point, the vehicle can utilize both interfaces simultaneously for data transmission, since it has access to both LTE and 802.11p.

6.2.4.3 Change subflow priority

At the initiation of a subflow, hosts have the ability to specify their preference for its role as either a primary or backup path. The backup path is only used when the regular path options are not present. Throughout the connection, the vehicle can request a modification in the priority of a subflow, accomplished through the transmission of an MP_PRIO signal. If the computed score for paths undergoes a change, a vehicle might want to notify the remote server about a modification in the subflow priority, for instance, elevating a subflow from its previous backup status to a higher priority over all other remaining subflows. To achieve this, the MP_PRIO option can be used to alter the *backup* designation of the subflow to which it is applied.

If non-optimal WLAN conditions are detected, the PM system instructs the kernel to use S_1 as a backup subflow. One potential approach could involve designating a path with a medium-level score, as determined by the network selection procedure, for use as the backup subflow. The vehicle can request a change in the subflow priority by sending an MP_PRIO signal to the remote host. The `backup` flag is activated on S_1, indicating that this subflow will be utilized only if the regular (S_0) becomes unavailable. Once the WLAN score surpasses the specified threshold, the backup subflow can be reactivated using the same command. It's important to note that the MP_PRIO signal is applicable to a single direction, and the recipient is expected to honor these requests. Nevertheless, the sender of this option retains the flexibility to continue utilizing the subflow for data transmission, even if it has previously signaled the *backup* flag to the other host.

6.2.4.4 Remove subflow

If a vehicle wishes to terminate a specific subflow without closing the entire connection, it can trigger a conventional TCP FIN/ACK exchange. Conversely, when the vehicle wants to notify the remote server of its depletion of data to transmit, it sends a "Data FIN" signal. This signal mirrors the semantics and behavior of a typical TCP FIN but operates at the connection level. Upon successful reception of all data within the MPTCP connection, this message is acknowledged at the connection level with a Data ACK.

In the event of the network selection mechanism detecting poor conditions on the path, such as a decrease in signal strength due to a vehicle moving away from the coverage area of the RSU, the PM system will issue a command to deactivate the corresponding subflow from the data transmission process. During the life of the MPTCP connection, if an address becomes invalid or is no longer preferred, the vehicle should communicate this situation. By doing so, the remote host can promptly eliminate subflows that are associated with the related address. Furthermore, a vehicle has the option to signal that a valid IP address should no longer be utilized, which can be particularly useful for ensuring session continuity during transitions. This is facilitated through the $REMOVE\_ADDR$ signal, used to remove a previously registered address from the connection and terminate any existing subflows that are relying on that specific address. Following this signal, both the vehicle and the remote host should initiate the transmission of RSTs on the impacted subflows. In the context of MPTCP, it's important to note that a standard RST has a limited scope, affecting only the associated subflow and leaving the remaining subflows unaffected. Prior to the removal of an address, the vehicle has the option to utilize MP_PRIO in order to express its desire for a specific path to cease being used.

However, after the subflow removal, the information about S_1 is still retained in the subflows table. This means that when the WLAN network becomes available again, the PM system is ready to establish the additional subflow without having to wait for the ADD_ADDR packet, which may not even arrive, using the MP_JOIN signal and the previously declared token.

This fundamental concept is applied consistently throughout each computational step: following the evaluation of each path by the network selection algorithm, the PM system will initiate the appropriate subflow creation, removal, or backup procedure. If any issues arise during a path management process, MPTCP will seamlessly revert to the conventional single-path TCP connection.

6.3 Network selection framework

The network selection algorithm should identify the optimal network based on the QoS criteria of ongoing applications and a set of cross-layer attributes. Let $R = \{r_1, ..., r_N\}$ represents the pool of networks that are accessible for providing network connectivity, and let the set of network attribute values be $C = \{c_1, ..., c_K\}$, where K denotes the number of observed attributes. These attributes encompass various factors that describe the characteristics of the available networks, such as latency, bandwidth, reliability, etc. We also assume that the vehicle can run a set of L types of services $S = \{s_1, ..., s_L\}$, which may have distinct requirements and preferences regarding network performance. The definition of the network selection problem in multipath environment could be as follows:

> Given a set of available networks (R), a group of services profiles (S) and the set of observed network attributes (C), select the optimal set of candidate networks and their corresponding network usage mode to ensure the best possible performance for a specific service.

The provided definition for the network selection problem describes the key components and objectives of the problem. The main goal is to determine the best combination of networks from R that can deliver optimal performance for a specific service. Network usage node indicates whether a selected network will be used as the primary (regular) or backup connection for a particular service. Following this definition, applications are assigned to network interfaces based on their unique requirements and the capabilities of the networks in question. This section describes the three parts of or network selection procedure, namely: attribute selection and utility values calculation, attribute weight calculation and the network ranking. The

FAHP-based framework of the network selection procedure is illustrated in Fig. 28. The entire process of the score calculation is shown in Algorithm 3.

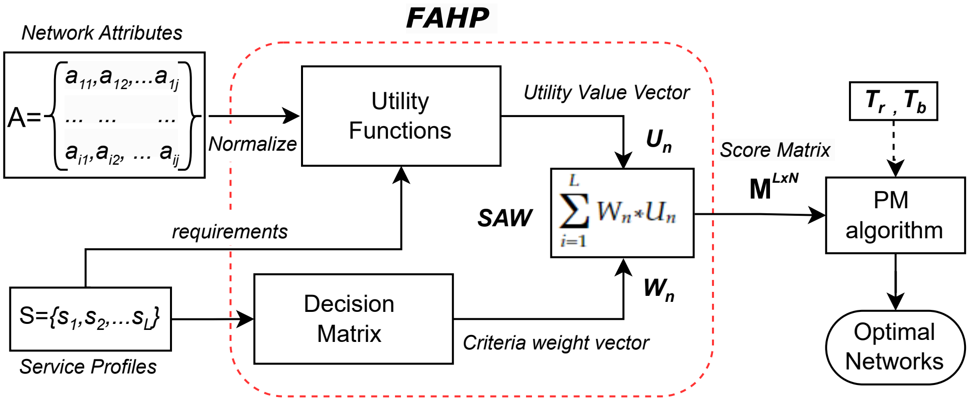


Figure 28: The framework for network selection based on FAHP method.

Algorithm 3: Network score computation

```
Input: $A = [a_{ij}]_{N \times K}$ , $S$ , $R$ , $\alpha$
Data: $M = [m_{ij}]_{L \times N}$                                // score matrix
for each service type as $i$ in $S$ do
    $req \leftarrow$ get requiremets of service $i$
    for each network as $j$ in $R$ do
        if $I$ is available then
            $score$ = FAHP ($req$, $A_{j,:}$)
            $M_{ij} = (1 - \alpha) * M_{ij} + \alpha * score$
        end
    end
end
```

Various network and context information from the vehicular environment are stored in the attribute information matrix $A = [a_{ij}]_{(N \times K)}$, that can be expressed as:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,j} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,j} \end{pmatrix} \tag{6.1}$$

where a_{ij} represents the value of an attribute i observed on the interface j. N indicates the number of available heterogeneous networks, and K represents the number of considered attributes. The algorithm must evaluate each available path, considering the requirements of all defined Service Profiles (SPs). Therefore, the FAHP procedure receives the real-time network attributes along with service requirements, to perform score calculation on each computational stage.

Due to the high variance of vehicular networks, the observed parameters undergoes drastic fluctuations, which may lead to frequent changes in subflow operational mode. A simple low-pass filter mechanism is used to reduce the probability of the ping-pong effect when the subflow switches between two modes frequently, resulting in unnecessary handovers. Here, the smooth factor *a* determines how quickly the filter responds to changes in the input signal. A higher smooth factor means that the PM will respond more slowly to changes in the RSS, resulting in a smoother output. However, too high of a smooth factor can cause the filter to become too sluggish, resulting in a delay in the output response. The choice of the smooth factor depends on the specific vehicular application and the desired balance between unnecessary handovers and maintaining a fast response time.

6.3.1 Utility functions

The utility functions are used to calculate the utility value vector ($\mathbf{U}$) of the network attributes to guarantee that the values are normalized according to the objective requirements of each SP. Utility functions are common in multi-criteria decision analysis. They allow decision-makers to translate personal preferences or criteria into a mathematical form that can guide decision-making processes. Utility functions help in comparing different attributes or criteria on the same scale. Having multiple attributes with different units or scales, normalizing them into a common scale allows for meaningful comparisons. This is particularly important in decision-making processes where is required to weigh the importance of various attributes against each other.

Normalization brings values within a standardized range, typically between 0 and 1. This makes it easier to work with data, especially when combining different attributes. For example, in multi-criteria decision analysis, utility functions can transform diverse criteria into a common metric. Furthermore, utility functions can also be used to reduce the dimensionality of a dataset. By normalizing attributes, we are essentially simplifying the dataset and reducing the risk of attributes with large values dominating the analysis.

Attributes can be classified into two categories: *benefit* attributes and *cost* attributes. For benefit attributes, a higher attribute value corresponds to a higher utility value, as is the case with the RSS metric and dwelling time utilized in this paper. Conversely, for cost attributes such as delay, jitter, PLR, and cost attributes, a higher attribute value results in a lower corresponding utility value. When dealing with attributes that have bilateral constraints, we employ the sigmoid utility function for normalization purposes. Specifically, when both upper and lower thresholds exists, we utilize $f(x)$ and $g(x)$ for benefit attributes and cost attributes, respectively. The expressions for these utility functions are given below:

$$f(x) = \frac{1}{1 + e^{-a(x-b)}} \tag{6.2}$$

$$g(x) = 1 - f(x) \tag{6.3}$$

where a and b are constant coefficients determined according to the service requirements for specific attributes. When dealing with attributes that have unilateral constraints, we employ linear and inverse proportional utility attribute functions. Specifically, for attributes which have only one threshold, we utilize $u(x)$ for benefit attributes and $h(x)$ for cost attributes. The definitions for these functions are presented below:

$$u(x) = 1 - \frac{g}{x} \tag{6.4}$$

$$h(x) = 1 - g \cdot x \tag{6.5}$$

where g is a constant coefficient that may vary with different attributes and service requirements. Table 5 displays the diverse utility functions and corresponding coefficients utilized for different SPs.

Table 5: Utility functions for service profiles.

Attribute	Func.	Conversational	Streaming	Background
RSS	$f(x)$	a=0.15, b=-80	a=0.15, b=-80	a=0.15, b=-80
Delay	$u(x)$	a=0.1, b=70	a=0.07, b=120	a=0.03, b=250
Jitter	$u(x)$	a=0.15, b=40	a=0.12, b=50	a=0.07, b=80
PLR	$h(x)$	g=1/20	g=1/15	g=1/10
Cost	$h(x)$	g=1/50	g=1/50	g=1/50
Dwell T.	$u(x)$	g=30	g=20	g=10

6.3.2 Applying Fuzzy AHP for Network Selection

This study utilized the FAHP method to solve the network selection problem [99, 154]. Compared to the traditional AHP approach [155], FAHP is a better method to handle the uncertainty and ambiguity inherent in decision-making processes. It involves breaking down complex problems into hierarchies and constructing a model with the main objective at the top and alternatives at the lower levels. For our network selection problem, a hierarchical structure was established using six input criteria (i.e., RSS, jitter, delay, PLR, usage cost, and dwell time) and a single output, as shown on Fig. 29. Among many FAHP methods in the literature [154], the Buckley's method [156] was preferred for this work, as due to its superior comprehensibility when compared to other FAHP methodologies we explored.

It's important to note that FAHP is not universally better than traditional AHP. The choice between AHP and FAHP depends on the specific problem, the quality of data available, and the preferences of decision-makers. Traditional AHP may suffice in many cases, especially when data is precise and decision-makers are comfortable providing numerical values for comparisons. However, FAHP is a valuable extension when dealing with more complex and uncertain decision environments. FAHP is based on fuzzy logic, which allows it to handle uncertainty and vagueness in a decision-making process. In real-world scenarios, especially in complex systems or situations with incomplete or imprecise data, FAHP can provide more robust and accurate results compared to traditional AHP.

In some problems, the relationships between criteria and alternatives may not be strictly binary (i.e., yes/no or 0/1). FAHP can model complex and gradual relationships, which is often more representative

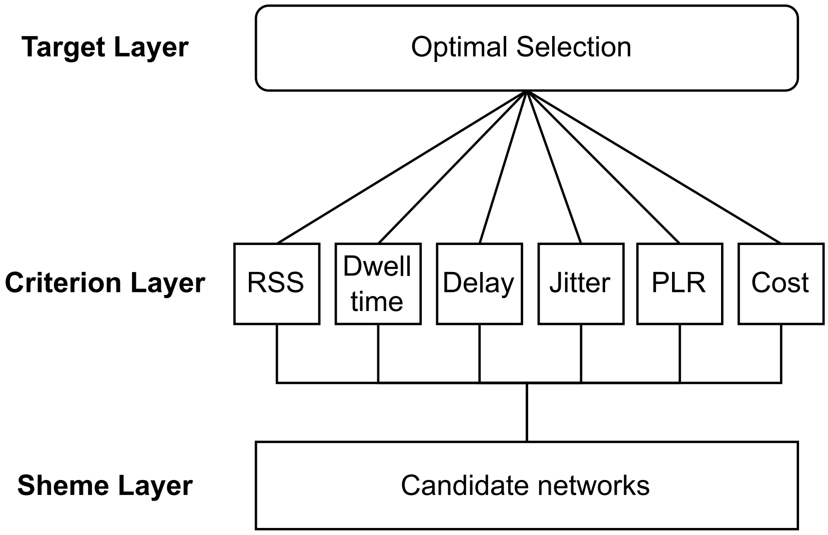


Figure 29: The hierarchy model of FAHP.

of real-world situations. FAHP is well-suited for situations where decision-makers have difficulty providing precise numeric values for comparisons. It allows decision-makers to express their preferences using linguistic variables and fuzzy sets, making it a more practical approach when dealing with human judgments. In situations with vague data and complex, interconnected criteria, FAHP may lead to improved decision quality. The consideration of uncertainty and vagueness in decision models can result in more robust and reliable choices.

The FAHP method is based on the principles of fuzzy logic and involves rating the decision criteria according to their importance using a set of linguistic terms that convey the degree of significance, such as *very important*, *somewhat important*, and *not important*. However, linguistic values present a challenge in mathematical operations. To overcome this limitation, one effective solution is to convert linguistic comparison ratios into fuzzy numbers. In this approach, each linguistic term is mapped to a fuzzy number, representing the underlying meaning of the term. To represent the fuzziness of preferences, we use Triangular Fuzzy Numbers (TNFs), which are defined as $M = (l, m, u)$ where $l \leq m \leq u$. The decision maker expresses the lower limit value, the most favorable value, and the upper limit value through l, m, and u, respectively. Equation 6.6 express a membership function of TFN. The calculation rules of TFN are shown in Equation 6.7 , where $\widetilde{N_1}$ and $\widetilde{N_2}$ are two TFN.

$$\mu_{\widetilde{M}(x)} = \begin{cases} \frac{x-l}{m-l} & l \leq x \leq m \\ \frac{u-x}{u-m} & m \leq x \leq u \\ 0 & otherwise \end{cases} \tag{6.6}$$

$$\begin{aligned}\widetilde{N}_1 + \widetilde{N}_2 &= (l_1 + l_2, m_1 + m_2, u_1 + u_2)\\ \widetilde{N}_1 - \widetilde{N}_2 &= (l_1 - l_2, m_1 - m_2, u_1 - u_2)\\ \widetilde{N}_1 \otimes \widetilde{N}_2 &= (l_1 \otimes l_2, m_1 \otimes m_2, u_1 \otimes u_2)\\ \widetilde{N}_1 \div \widetilde{N}_2 &= (l_1 \div l_2, m_1 \div m_2, u_1 \div u_2)\end{aligned} \tag{6.7}$$

In order to compare the relative importance of attributes, a linguistic approach must be employed, followed by the transformation of linguistic values into TFNs. Table 6.3.2 illustrates the relationship between the importance of attributes and their corresponding TFNs.

Table 6: Importance of TFN value

	Definition	TFN	Reciprocal TFN
1	Equal Importance	(1,1,3)	(1/3,1,1)
2	Intermediate Values	(1,2,4)	(1/4.1/2,1)
3	Moderate Importance	(1,3,5)	(1/5,1/3,1)
4	Intermediate Values	(2,4,6)	(1/6,1/4,1/2)
5	Strong Importance	(3,5,7)	(1/7,1/5,1/3)
6	Intermediate Values	(4,6,8)	(1/8,1/6,1/4)
7	Very Strong Importance	(5,7,9)	(1/9,1/7,1/5)

6.3.2.1 Pairwise comparison

FAHP is a multi-criteria decision-making methodology that involves comparing the relative importance or preference of different criteria or alternatives. Pairwise comparisons provide a systematic way for decision-makers to express their subjective judgments about these preferences. To determine the final scores of each available network, the FAHP method involves assigning weights to the various evaluation criteria by conducting pairwise comparisons between the elements of each hierarchy, thereby establishing the priority of each attribute compared to all other attributes. FAHP calculates weights that represent the relative importance of criteria and alternatives. These weights are crucial for the decision-making process. Pairwise comparisons are the basis for deriving these weights, and they are used in the subsequent mathematical calculations of FAHP. It provides a structured approach to rank or select alternatives based on the criteria. The pairwise comparison results are used to compute overall scores and rankings, and guide the decision-maker in choosing the most suitable alternative.

The relative importance between the attributes is indicated with TFN by considering two criteria at a time. The pairwise comparison matrix $\widetilde{D} = [\widetilde{d}_{ij}]_{(K\times K)}$, can be expressed as:

$$\widetilde{D} = \begin{pmatrix} \widetilde{d}_{1,1} & \cdots & \widetilde{d}_{1,n} \\ \vdots & \ddots & \vdots \\ \widetilde{d}_{n,1} & \cdots & \widetilde{d}_{n,n} \end{pmatrix} \tag{6.8}$$

where K is a number of network attributes and $\widetilde{d_{ij}}$ indicates the decision maker preference of i^{th} criterion over j^{th} criterion expressed by TFN. It must be constructed one comparison matrix for each SP, considering its requirements.

6.3.2.2 Computing the weight vector

By assigning weights to each attribute, we can effectively prioritize and evaluate the different alternatives based on their respective performance in each category. The geometric mean of fuzzy comparison values of each attribute is calculated as in Equation 6.9, where $\widetilde{r}$ and $\widetilde{d_{ij}}$ still represent triangular values.

$$\widetilde{r_i} = \left(\prod_{i=1}^{n} \widetilde{d_{ij}} \right)^{\frac{1}{n}} = (\widetilde{d_{i1}} \otimes \widetilde{d_{i2}} \otimes \cdots \otimes \widetilde{d_{in}})^{\frac{1}{n}} \tag{6.9}$$

Then, to find the fuzzy weight ($\widetilde{w}$) of criterion i, multiply each $\widetilde{r_i}$ with the reverse vector:

$$\widetilde{w_i} = \widetilde{r_i} \otimes (\widetilde{r_1} \oplus \widetilde{r_2} \oplus \cdots \oplus \widetilde{r_n})^{-1} = (lw_i, mw_i, uw_i) \tag{6.10}$$

Since $\widetilde{w_i}$ are still FTNs, they need to be converted into non-fuzzy values. Equation 6.11 shows the defuzzification operation according to the center of area method. Then, weights are normalized by applying Equation 6.12 to find the weight value for each attribute and construct the weight vector $\mathbf{W} = \{w_1, w_2, ..., w_i\}$ for each SP, where $0 \leq w_i \leq 1$.

$$M_i = \frac{lw_i + mw_i + uw_i}{3} \tag{6.11}$$

$$w_i = \frac{M_i}{\sum_i^n M_i} \tag{6.12}$$

The pairwise comparison matrix is constructed according to the requirements of a traffic class and the importance of each network attribute. Table 7 demonstrate the three fuzzy matrices and the corresponding weight vectors for tree service profiles, namely conversational (SP_C), streaming (SP_S) and background (SP_B). Appendix B shows the code in python for calculation of the criteria weigth vector.

6.3.2.3 Score computation

The process of score computation involves collecting raw network attribute values, which are then transformed using utility functions. The coefficients in these functions vary depending on the specific services. The weight vector is initially calculated for each service type from its corresponding fuzzy decision matrix. This vector remains unchanged unless the decision matrix itself is modified, which only occurs when there is a shift in the relative importance of the attributes. Consequently, the equation for score computation is mainly influenced by real-time changes in network attributes.

The scores for each alternative RAT are determined by multiplying the weight vector of each service criterion by its corresponding attribute utility value, based on the Simple Additive Weighting (SAW) method

Table 7: Fuzzy comparison matrix and weights for Background, Conversational and Streaming service profiles

SP_B	RSS	Delay	Jitter	PLR	Cost	Dwell T.	Weight
RSS	1, 1, 3	1, 3, 5	3, 5, 7	1/5, 1/3, 1	1, 1, 3	1, 3 , 5	0.221
Delay	1/5, 1/3, 1	1, 1, 3	3, 5, 7	1/5, 1/3, 1	1/4, 1/2, 1	1/4, 1/2, 1	0.106
Jitter	1/7, 1/5, 1/3	1/7, 1/5, 1/3	1, 1, 3	1/9, 1/7, 1/5	1/7, 1/5, 1/3	1/7, 1/5, 1/3	0.031
PLR	1, 3, 5	1, 3, 5	5, 7, 9	1, 1, 3	1, 2, 4	1, 3, 5	0.328
Cost	1/3, 1, 1	1, 2, 4	3, 5, 7	1/4, 1/2, 1	1, 1, 3	1, 2, 4	0.179
Dweel T.	1/5, 1/3, 1	1, 2, 4	3, 5, 7	1/5, 1/3, 1	1/4, 1/2, 1	1, 1, 3	0.134

SP_C	RSS	Delay	Jitter	PLR	Cost	Dwell T.	Weight
RSS	1, 1, 3	1/4, 1/2, 1	1/3, 1, 1	1, 3, 5	3, 5, 7	1/3, 1, 1	0.160
Delay	1, 2, 4	1, 1, 3	1, 3, 5	1, 3, 5	3, 5, 7	1, 2, 4	0.314
Jitter	1, 1, 3	1/5, 1/3, 1	1, 1, 3	3, 5, 7	1, 3, 5	1, 1, 3	0.210
PLR	1/5, 1/3, 1	1/5, 1/3, 1	1/7, 1/5, 1/3	1, 1, 3	1, 2, 4	1/7, 1/5, 1/3	0.071
Cost	1/7, 1/5, 1/3	1/7, 1/5, 1/3	1/5, 1/3, 1	1/4, 1/2, 1	1, 1, 3	1/7, 1/5, 1/3	0.048
Dwell T.	1, 1, 3	1/4, 1/2, 1	1/3, 1, 1	3, 5, 7	3, 5, 7	1, 1, 3	0.198

SP_S	RSS	Delay	Jitter	PLR	Cost	Dwell T.	Weight
RSS	1, 1, 3	1, 1, 3	1/5, 1/3, 1	1, 1, 3	1, 3, 5	1, 1, 3	0.187
Delay	1/3, 1, 1	1, 1, 3	1/3, 1, 1	1, 1, 3	1, 3, 5	1, 2, 4	0.178
Jitter	1, 3, 5	1, 1, 3	1, 1, 3	1, 3, 5	1, 3, 5	1, 3, 5	0.304
PLR	1/3, 1, 1	1/3, 1, 1	1/5, 1/3, 1	1, 1, 3	1, 2, 4	1, 1, 3	0.133
Cost	1/5, 1/3, 1	1/5, 1/3, 1	1/5, 1/3, 1	1/5, 1/3, 1	1, 1, 3	1/5, 1/3, 1	0.080
Dweel T.	1/3, 1, 1	1/4, 1/2, 1	1/5, 1/3, 1	1/3, 1, 1	1, 3, 5	1, 1, 3	0.117

[103]. We chose the SAW method due to its simplicity and relatively low processing time, which are critical factors for effective decision-making in a highly dynamic environment. SAW method aims to calculate a weighted sum of performance ratings for each RAT across various normalized attributes, as show in Equation 6.13. The resulting *score* value represents the Quality of Experience (QoS) measure that a vehicle user can achieve for a specific service by using one of the alternative networks. It also indicates the probability of the evaluated network being selected for that particular service.

$$\mathbf{W_n} \cdot \mathbf{U_n} = w_1 u_1 + w_2 u_2 + \cdots + w_i u_i \tag{6.13}$$

By comparing the network score of the available candidate networks, our PM mechanism is employed to regulate subflow utilization in the active MPTCP connections. It selects alternatives with the highest scores to establish subflows, while excluding RATs with low scores from the connection based on a pre-defined threshold. For instance, Figure 31 illustrates the probability of the *Background* service to access the WLAN network with the variation of PLR and dwell time attributes. It should be noted that while six parameters are utilized in the decision-making process, we have chosen to focus on the access probability with respect to the two attributes due to limitations in representing the score variation for all parameters together. A comparable outcome can be observed in Figure 30, which displays the probability of the *Conversational* service accessing the WLAN network based on the fluctuation of jitter and dwell time. Based on the scores obtained, the PM will determine whether a subflow should be used as a backup, regular, or removed altogether. Indeed, a higher score corresponds to a greater probability of accessing the 802.11p network for data transmission. For example, a score below 0.5 may indicate that the path has poor conditions, and the associated subflows should be removed from the session. A score between 0.5 and 0.75 indicates acceptable path conditions; however, it is preferred for use in backup mode. A score above 0.75 identifies optimal path conditions, and the network should be used by the PM to establish regular subflow.

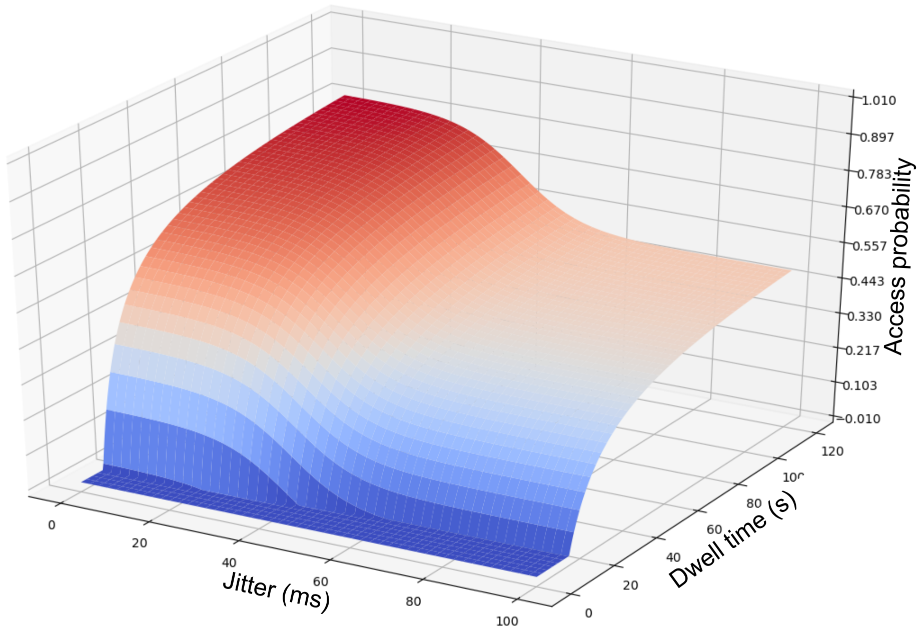


Figure 30: The probability of *Conversational* service to access the WLAN with variation of dwell time and jitter.

6.4 Performance Evaluation

This section details an emulation-based study conducted on the proposed PM scheme. We examine a V2N communication scenario where an OBU equipped vehicle maintains network connectivity, accessing a remote host through two wireless technologies, 802.11p and LTE, as shown in Figure 12. The main focus is to evaluate the impact of proposed PM architecture and FAHP network selection mechanism on the performance of the V2N communication. It is assumed that the cellular network is always accessible to the vehicle, while ad hoc connectivity with RSUs deployed at different locations is only available intermittently along the road. To ensure reliable results, we configured the experimental platform as described in Chapter 3. The proposed PM and with other applications were run in an isolated namespace that is connected to the simulated vehicle. All experiments were conducted in real-time using upstream Linux kernel v.15.9 and `mptcpd` v0.9. The network-related data, encompassing metrics like delay, RSS (Received Signal Strength), PLR (Packet Loss Rate), and jitter, was diligently collected through the ns-3 network simulator. This information was then seamlessly transmitted to the PM for in-depth processing

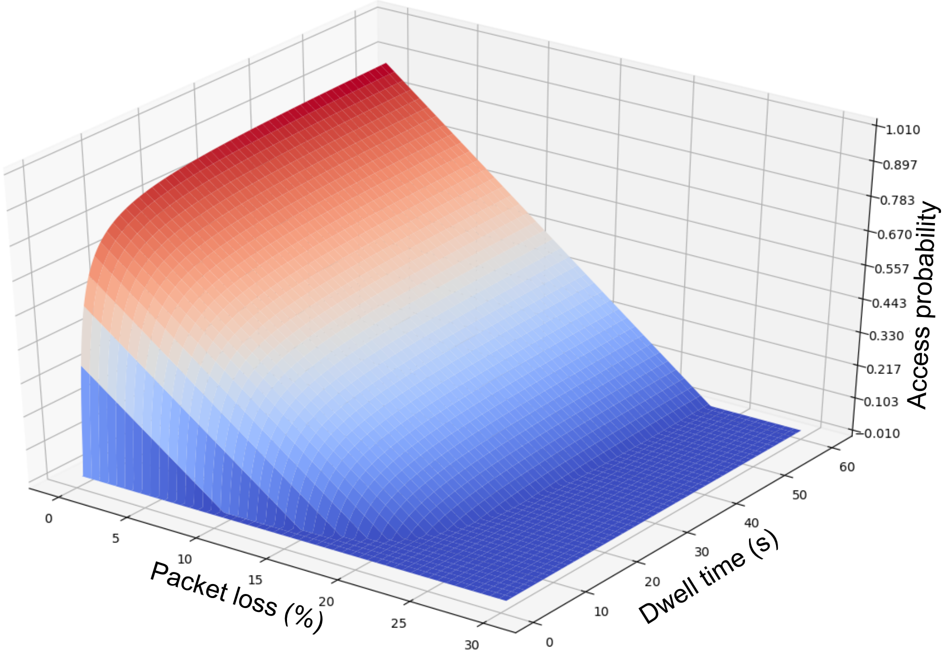


Figure 31: The probability of *Background* service to access the WLAN with variation of dwell time and PLR.

and analysis. Simultaneously, the SUMO traffic simulator was pivotal in furnishing mobility-related insights. It helps track key parameters like vehicle speed, precise positioning, and directional changes. This comprehensive approach allowed for a holistic understanding of the network and mobility dynamics, facilitating more informed decision-making within the system. To configure a single subflow per interface for each MPTCP connection, we used the latest version of `iptools2`. Furthermore, we employed `iperf3` to generate traffic between the vehicle and remote host.

6.4.1 Seamless connectivity

As the vehicle moves on the road within a multi-access environment, frequent handovers are expected. Figure 32 depicts the average throughput per interface and the total achieved throughput during an MPTCP session, where the vehicle uploads data to the server at a rate of 1 Mbps, using the *Background* service profile.

In the initial phase, the PM received favorable scores for both the 802.11p and LTE interfaces via FAHP. This enabled the PM to instruct MPTCP to harness both interfaces concurrently for transmitting a single data stream. As the path conditions, including delay and available bandwidth, closely resembled each other on these two paths, the scheduler efficiently allocated a proportionate share of the data to each interface. However, at $t_1 = 6sec$, the LTE path experienced a notable decline in its score, leading to its demotion to a backup path. Consequently, only the WLAN interface remained active for subsequent data transmission. This transition was instigated by the reduced score on the cellular network, prompting

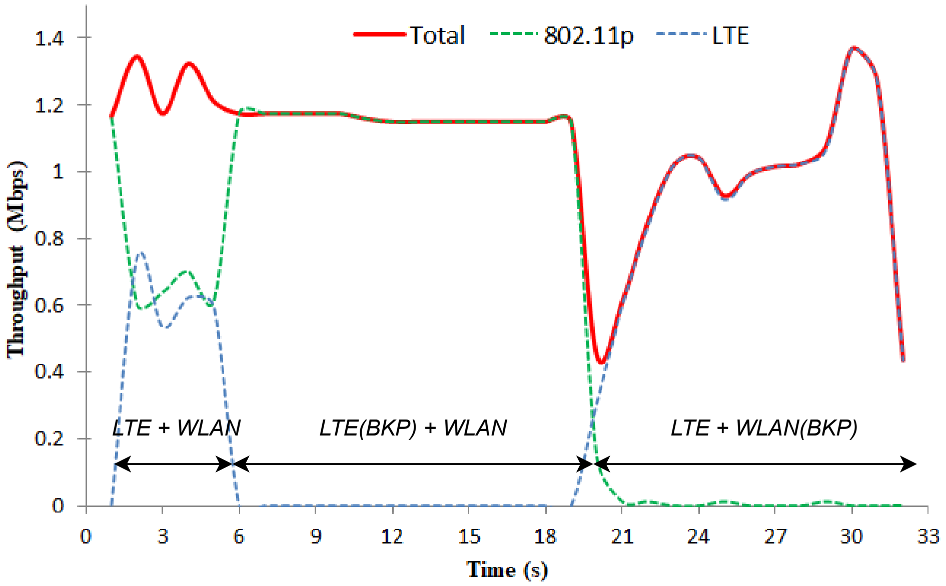
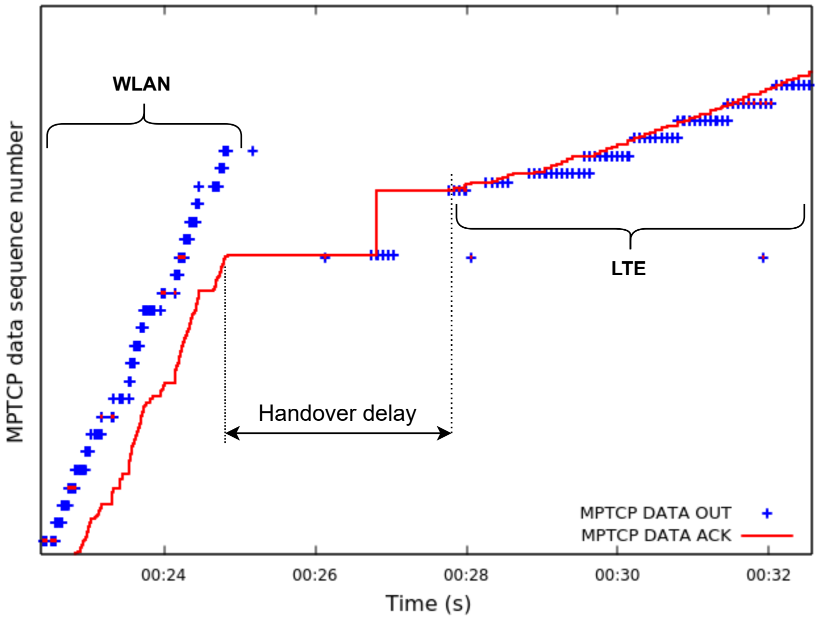


Figure 32: The throughput of the MPTCP session with intelligent path menegment.

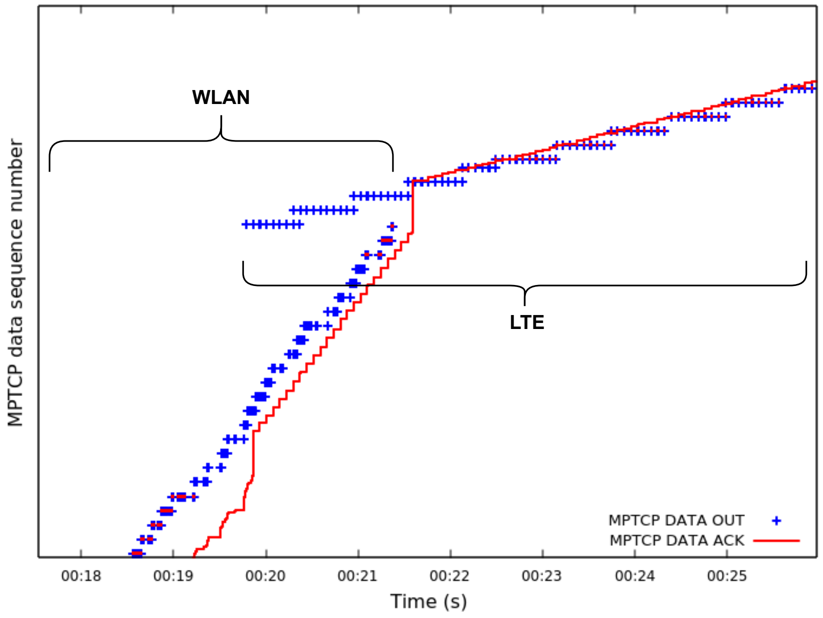
the PM to optimize the data flow by relying solely on the WLAN interface.

Around time $t_2 = 20sec$, the vehicle moves out of the RSU coverage area, resulting in a decrease in the WLAN network path's score. As a result, the PM instructs the kernel to remove the subflow on the that interface, and only the cellular network is used for further data transmissions. The PM removes the backup status of the cellular network, allowing the packet scheduler to resume using the LTE interface for data transmission. Simultaneously, the PM designated the 802.11p interface as the backup, considering its score. Afterward, the PM moves the lossy path to the backup mode in order to prevent it from being used for data transmission. The underperforming subflow could be removed from the MPTCP session if the network score drops below the predefined threshold, as per the proposed PM algorithm. Figure 33 depicts the evolution of Data Sequence Number (DSN) during a handover from a WLAN to an LTE network. This comparison illustrates how MPTCP behaves both before (Figure 33(a)) and after (Figure 33(b)) our path manager is enabled. These figures clearly demonstrate that our path manager can proactively and smoothly switch the connection from one network to another without introducing any delays or interruptions in the data flow. Our path manager can also dynamically reactivate the suspended subflow when the path quality (i.e., score) improves, thereby continuously optimizing the network performance by adapting to changing conditions.

During an MPTCP session, the connection can migrate from one access technology to another multiple times, but any interruption in service will be perceived at the application layer. Our PM is designed to predict deteriorating network conditions and proactively take appropriate corrective actions to ensure service continuity, before the links eventually break. In the case of the vehicle moving away from the RSU, a drop in network score is observed. Therefore, according to our algorithm, the subflow must be suspended as soon as the score drops below the predefined threshold to ensure seamless handover and uninterrupted



(a) Intelligent PM disabled



(b) Intelligent PM enabled

Figure 33: The behavior of the MPTCP during the handover.

service delivery. This demonstrates the effectiveness of the proposed PM mechanism in providing service continuity and dynamic subflow control for moving vehicles over heterogeneous networks.

6.4.2 Application delay

Our path manager was tested in a Manhattan-like urban scenario with bidirectional streets and one lane per direction. During the simulation, the vehicle follows a designated trajectory, moving through various coverage areas while performing data transfer with the remote host. The scenario aimed to assess the algorithm's effectiveness for V2N communication with various infrastructure node deployments. The LTE and IEEE 802.11p interfaces were installed in all vehicles, which ran the Cooperative Awareness Message (CAM) service to exchange real-time information. The scenario featured a single LTE base station and a set of RSUs distributed uniformly at random intersections.

Applications with real-time requirements, such as media streaming, are highly susceptible to frequent handovers and path variation, leading to increased data delivery latency and perceivable jitter. To effectively monitor any fluctuations in application delay during the handover, we generate traffic with a timestamp embedded in each data segment. Our approach involves measuring the end-to-end delay, which is determined by calculating the time taken for a packet to leave the source application and reach the destination application. We record the arrival time of each data segment received, and by comparing the timestamps, we can accurately determine the end-to-end application delay. In order to ensure precision in our measurements, our emulation setup employs a same hardware clock across all communication endpoints.

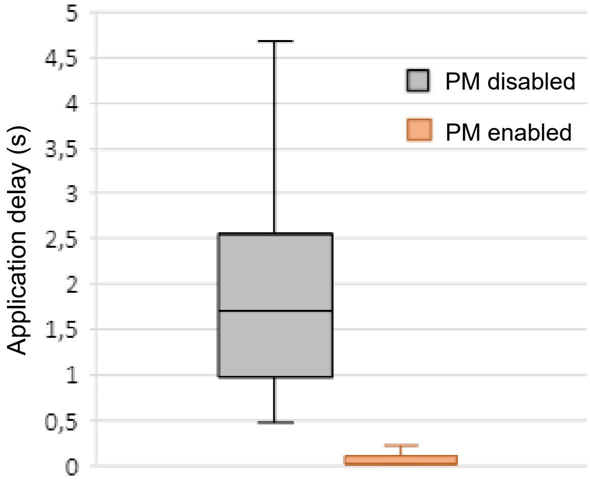


Figure 34: The end-to-end delay during the handover event.

We conducted two identical simulations to compare the effectiveness of our PM with the default MPTCP scheme. In one test, we disabled our PM, while in the other, we enabled it. Each simulation lasted for one hour and involved uploading files of varying sizes from the vehicle to the server. Following the completion

Table 8: Network parameters variation

	RSS(dBm)	Delay(ms)	Jitter(ms)	PLR(%)	Cost	Dwell Time(s)
LTE	-100:30	5:80	2:50	1:10	15:45	30:600
WLAN	-100:30	20:250	5:100	2:20	0:20	5:120

of the simulations, we compared the end-to-end delay, and represented the results in Fig. 34. Our findings reveal that the standard MPTCP scheme led to a notable increase in the application layer delay during handover, ultimately resulting in connection glitches. Unlike, our proposed PM scheme demonstrated a considerable reduction in delay during VHO by effectively controlling subflow usage from the userspace, optimizing network performance and ensuring uninterrupted service for vehicular users.

6.4.3 Network selection probability

This experiment illustrates the response produced by FAHP network selection algorithm under different network conditions for three types of services: background, conversational, streaming. This demonstration provides insight into the algorithm's efficacy and suitability for different network scenarios. We conducted experiments by varying parameters that impact the performance of multipath vehicular communications, including signal strength, delay, jitter, packet loss rate, price and dwelling time. We simulated 10000 network selections activities for each service type by generating random network attributes, as listed in Table 8.

Figure 35 displays the statistical probability of each network type being selected for different service profiles after completing the experiment. The figure illustrates how the FAHP algorithm adapts to diverse service requirements by creating regular and backup (BKP) subflows through available networks. Our algorithm takes into account the service characteristics and network environment to ensure a balanced probability of selecting each type of network. However, it is more inclined to select the cellular network, which is better aligned with the services under consideration in terms of packet losses, latency, and residence time.

WLAN is more likely to be selected for *background* services due to its low cost and acceptable packet loss rate, while LTE has the highest selection probability for delay-sensitive applications. For *conversational* services, low delay and jitter are important, as well as a long-lived and stable connection with the same infrastructure node to avoid frequent handovers and potential interruptions during service. Therefore, cellular networks are preferred for creating subflows, while WLAN networks are mostly considered as backup paths.

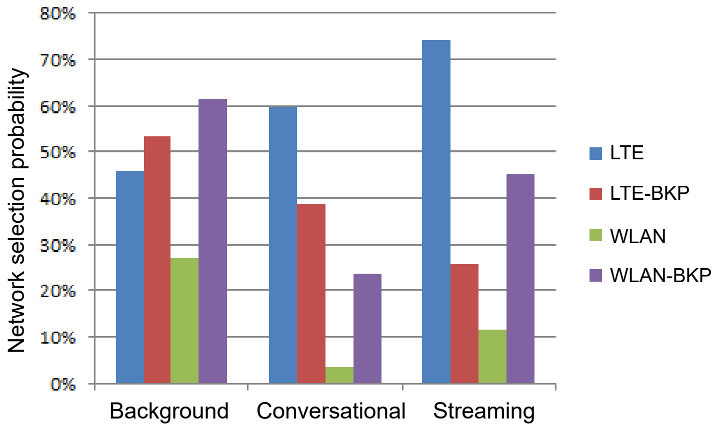


Figure 35: The statistical probability of each type of network to be selected.

Chapter 7

Conclusions and Future work

The integration of different RATs is anticipated to fulfill multiple requirements within ITS. Future Internet of Vehicles (IoV) landscapes envision the coexistence of various wireless interfaces with heterogeneous capabilities. The integration of communication technologies will enable vehicles to facilitate efficient data exchange and support a wide range of vehicular applications, ultimately contributing to developing safer, more efficient, and interconnected transportation systems. Multipath communication offers numerous advantages, including improved connection reliability and increased throughput. The potential of multipath communication to revolutionize highly mobile vehicular networks has been explored in this thesis. The limited literature on this topic suggests that existing multipath designs are unsuitable for highly dynamic networks. Given this challenge, we began our investigation with the goal of adopting the multipath approach to manage RAT interoperability. In particular, we examine the MPTCP as a potential solution to address seamless mobility in vehicular networks.

Hence, this work primarily addresses the challenge of minimizing disconnection time and its adverse effects to vehicular communication. To tackle this issue, we introduce a novel method that enables seamless and simultaneous connections to available networks. As a consequence, our approach proactively anticipates potential mobile handovers, leveraging a soft and efficient utilization of a logical interface. This implementation significantly reduces handover latency and reduces packet loss, which in turn improves the overall system's performance. Furthermore, our proposed method ensures the uninterrupted continuity of sessions for the vehicle whenever possible. Moreover, this framework exhibits the potential for extending its capabilities to optimize traffic load balancing by intelligently routing distinct data flows through diverse networks.

Chapter 3 presents the emulation framework used to analyze multipath connectivity in heterogeneous vehicular environments without setting up a real network. The framework is based on the ns-3 platform, enabling interaction with real systems and access to the functionalities of container-based nodes within the simulation. The proposed emulation method has been rigorously validated for accuracy and reliability across various scenarios. We conducted an illustrative study that examined the behavior of MPTCP when used over ad-hoc and cellular wireless networks within the context of vehicular mobility. Furthermore, a detailed description of the emulation environment and key limitations is given. Moreover, this methodology can be used in a wide range of use cases beyond the scope of MPTCP, thanks to its support for real applications. To help other researchers set up and use similar emulation frameworks, we've made all code examples, scripts, and results available to the public.

In Chapter 4, we investigated MPTCP and its relevance in heterogeneous vehicular networks. Our exploration starts by looking at the advantages and challenges of multipath communication in the scope of ITS. We presented a comprehensive overview of the issues associated with conventional path management and packet scheduler strategies in dynamic environments. To address these challenges, we introduced a cross-layer Path Management (PM), which leveraged interactions between upper and lower protocol stack layers to enhance the accuracy of path management. The proposed PM operates within the userspace and can dynamically adjust subflow usage in response to frequent changes in the link quality.

Chapter 5 provides a comprehensive analysis of the performance of the proposed Cross-layer Path Manager. To begin, we detail the methodology and techniques employed in the evaluation process. This encompasses insights into the experimental setup and the procedures for data collection. Leveraging our emulation framework, we illustrate how the proposed strategy can grantee seamless mobility across heterogeneous networks, resulting in a substantial reduction in handover latency and the occurrence of out-of-order packet delivery. The chapter examines the results derived from these experiments and thoroughly investigates their implications, and compares the performance of MPTCP with and without the activation of our path manager.

In our research, we directed our focus toward a practical V2N (Vehicle-to-Network) scenario, which is the most probable application of the multipath approach in heterogeneous networks. Within this scenario, a vehicle is equipped with ad-hoc and cellular interfaces, enabling it to communicate seamlessly with a remote host while moving on the road. This scenario provided a suitable and realistic context for conducting in-depth investigations of the proposed solutions for non-safety critical applications. As a result, the proposed PM scheme significantly improved communication performance, enabling faster path failure detection and recovery compared to the standard MPTCP. By implementing dynamic and proactive subflow management techniques, our PM algorithm aims to provide reliable and uninterrupted connectivity while optimizing the utilization of accessible network links. The outcome is a more efficient exploitation of network resources, contributing significantly to overall system performance enhancement.

Chapter 6 significantly advances our research, substantially improving the path selection algorithm. Our enhanced algorithm relies on the Fuzzy Analytic Hierarchy Process (Fuzzy AHP), considering multiple network attributes. These attributes include loss rate, dwelling time, delay, jitter, price, and signal strength. By incorporating a comprehensive set of attributes, our network selection algorithm achieves higher accuracy, ensuring optimal network selections in dynamic vehicular environments. The algorithm involves the calculation of attribute utility values and objective weights and assigning scores to evaluate each available access technology dynamically. Furthermore, our PM scheme takes application requirements into deep consideration. It performs network selection for various service types individually, ensuring that the most suitable network is selected for each actively running application. This tailored approach guarantees optimal network choices for diverse application scenarios, enhancing overall network performance and user experience.

The outcome of this approach is a highly efficient utilization of network resources precisely aligned with the unique requirements of each application. This meticulous tailoring significantly elevates the overall system performance and the quality of the user experience. Moreover, our network selection algorithm maintains its operation at the application layer, allowing the flexibility to extend or modify the path selection methodology easily. The emulation experiments demonstrate that the proposed algorithm can manage multipath connections seamlessly. This dynamic control ensures uninterrupted communication, even in complex environments such as vehicular networks.

7.1 Future work

The testing conducted with our path manager in this study primarily relies on emulation. While emulation serves as a valuable initial testing environment, it is essential to acknowledge the necessity for real-world testing to evaluate the performance of the presented solution. Numerous unpredictable factors can significantly influence connection performance in real networks, affecting multi-RAT vehicle equipment's functionality. Real-world networks frequently exhibit variable delays, packet losses, and fluctuating bandwidth conditions. These dynamic network conditions can have a significant impact on the overall performance of the developed PM. While our test bed framework closely simulates real-world vehicular heterogeneous environments, achieving the most accurate and comprehensive insights into these dynamic factors and their potential effects on the MPTCP protocol and our proposed path manager necessitates further real-world testing.

Additionally, it's worth noting that this thesis primarily focuses on the V2N scenario, particularly related to Internet access. V2N communications can be facilitated through various available vehicular communication technologies and typically entail less stringent requirements regarding delay and reliability. However, it is essential to highlight that other communication scenarios encompassing a broader spectrum of transportation elements, such as pedestrians, vehicles, road infrastructure, and cloud environments, could be subject to evaluation in future research. Using our multi-RAT scheme could empower vehicles with enhanced information access and significantly contribute to the development of IoV.

The findings presented in this study primarily center on the utilization of two prominent heterogeneous networks, namely LTE and IEEE 802.11p. These networks were chosen due to their strong potential for delivering effective V2N communication services in the future. However, it's vital to acknowledge that the landscape of wireless technologies is vast and continuously evolving. While our research offers valuable insights into the feasibility and advantages of multipath communication in the context of these two networks, it represents only a fraction of the broader vehicular network ecosystem. To provide a more comprehensive understanding of vehicular communication and multipath strategies, the inclusion of additional RATs is essential. This expanded scope would enable us to perform a more detailed and accurate comparison across various networks. These additional RATs could encompass emerging technologies or existing ones, such as 5G, Wi-Fi 6, or other dedicated vehicular communication protocols. The multipath protocol, operating at the transport layer, stands out as a versatile and technology-agnostic solution. This adaptability and flexibility guarantee its effective management of data transmission, regardless of the type and quantity of RATs available. As a result, the addition of any extra RAT is anticipated to be a seamless and straightforward process.

In addition to real-world testing, conducting a comparative analysis with other multipath protocols is important to gain a comprehensive understanding of their performance. For example, the MPQUIC protocol, which has recently been standardized and is getting attention from different groups, is a valuable candidate to compare with MPTCP in the vehicular situation. MPQUIC is an extension of the QUIC protocol, a reliable transport protocol built upon UDP, offering functionality similar to TCP while introducing features like data multiplexing and prioritization. MPQUIC provides a framework that facilitates the utilization of multi-streaming functionality. Moreover, MPQUIC implementations can be readily integrated as libraries within regularly updated applications. It is expected that our PM scheme can seamlessly integrate into MPQUIC since it is entirely implemented at the application layer, requiring no direct interaction with the kernel. This design simplifies the process of extension and adoption, making it easier to integrate and conduct further testing.

Furthermore, the integration of more advanced techniques, such as artificial intelligence (AI), holds promise for enhancing the network selection algorithm. The dense environment characterized by multiple RATs poses significant challenges in network selection due to the need for more frequent and intricate decision-making processes. Integrating artificial intelligence into ultra-dense heterogeneous vehicular networks can revolutionize network selection, making it more efficient and intelligent. This development aligns with the evolving landscape of 5G, where intelligence is becoming a fundamental feature. Contrary to the current access network architecture, 5G is expected to embrace a user-centric access network structure. This shift will empower vehicles with enhanced capabilities to meet a wide range of communication needs effectively. Vehicles equipped with intelligent OBUs have been at the forefront of advancing artificial intelligence and represent a significant commercial opportunity.

Bibliography

[1] J. Wang et al. *A survey of vehicle to everything (V2X) testing*. 2019. doi: `10.3390/s19020334` (cit. on p. 1).

[2] P. K. Singh, S. S. K. Nandi, and S. S. K. Nandi. “A tutorial survey on vehicular communication state of the art, and future research directions”. In: *Vehicular Communications* 18 (2019), p. 100164. issn: 22142096. doi: `10.1016/j.vehcom.2019.100164` (cit. on pp. 1, 9, 18, 32).

[3] J. M. L. Domínguez and T. J. Mateo Sanguino. “Review on V2X, I2X, and P2X communications and their applications: A comprehensive analysis over time”. In: *Sensors (Switzerland)* 19.12 (2019). issn: 14248220. doi: `10.3390/s19122756` (cit. on p. 1).

[4] K. Kiela et al. *Review of V2X-IoT standards and frameworks for ITS applications*. 2020. doi: `10.3390/app10124314` (cit. on pp. 1, 18).

[5] L. Tuyisenge et al. “A mobile internal vertical handover mechanism for distributed mobility management in VANETs”. In: *Vehicular Communications* 26 (2020), p. 100277. issn: 22142096. doi: `10.1016/j.vehcom.2020.100277` (cit. on pp. 2, 31).

[6] V. S. Hapanchak. *Integration of different radio access technologies for vehicular connectivity*. Research Proposal. 2020. doi: `10.13140/RG.2.2.21826.35522` (cit. on p. 9).

[7] H. Zhou et al. “Evolutionary V2X Technologies Toward the Internet of Vehicles: Challenges and Opportunities”. In: *Proceedings of the IEEE* 108.2 (2020), pp. 308–323. issn: 15582256. doi: `10.1109/JPROC.2019.2961937` (cit. on pp. 9, 10).

[8] F. Arena and G. Pau. “An overview of vehicular communications”. In: *Future Internet* 11.2 (2019). issn: 19995903. doi: `10.3390/fi11020027` (cit. on p. 9).

[9] C. R. Storck and F. Duarte-Figueiredo. “A Survey of 5G Technology Evolution, Standards, and Infrastructure Associated with Vehicle-to-Everything Communications by Internet of Vehicles”. In: *IEEE Access* (2020), pp. 1–1. doi: `10.1109/access.2020.3004779` (cit. on pp. 10, 17, 20).

[10] G. Naik, J. Liu, and J. M. J. Park. “Coexistence of wireless technologies in the 5GHz bands: A survey of existing solutions and a roadmap for future research”. In: *IEEE Communications Surveys and Tutorials* 20.3 (2018), pp. 1777–1798. issn: 1553877X. doi: `10.1109/COMST.2018.2815585` (cit. on pp. 10, 21).

[11] R. Jacob, N. Franchi, and G. Fettweis. “Hybrid V2X Communications: Multi-RAT as Enabler for Connected Autonomous Driving”. In: *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*. Vol. 2018-Septe. 2018, pp. 1370–1376. isbn: 9781538660096. doi: `10.1109/PIMRC.2018.8580953` (cit. on p. 10).

[12] C. Liu et al. “Adaptive Offloading for Time-Critical Tasks in Heterogeneous Internet of Vehicles”. In: *IEEE Internet of Things Journal* 7.9 (2020), pp. 7999–8011. issn: 23274662. doi: `10.1109/JIOT.2020.2997720` (cit. on p. 12).

[13] H. H. R. Sherazi et al. “A heterogeneous IoV architecture for data forwarding in vehicle to infrastructure communication”. In: *Mobile Information Systems* 2019 (2019). issn: 1875905X. doi: `10.1155/2019/3101276` (cit. on p. 12).

[14] K. C. Dey et al. “Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network - Performance evaluation”. In: *Transportation Research Part C: Emerging Technologies* 68 (2016), pp. 168–184. issn: 0968090X. doi: `10.1016/j.trc.2016.03.008` (cit. on pp. 12, 15).

[15] P. Sewalkar and J. Seitz. “Vehicle-to-pedestrian communication for vulnerable road users: Survey, design considerations, and challenges”. In: *Sensors (Switzerland)* 19.2 (2019). issn: 14248220. doi: `10.3390/s19020358` (cit. on p. 16).

[16] A. Hussein et al. “P2V and V2P communication for pedestrian warning on the basis of autonomous vehicles”. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (2016), pp. 2034–2039. doi: `10.1109/ITSC.2016.7795885` (cit. on p. 16).

[17] Z. Machardy et al. “V2X access technologies: Regulation, research, and remaining challenges”. In: *IEEE Communications Surveys and Tutorials* 20.3 (2018), pp. 1858–1877. issn: 1553877X. doi: `10.1109/COMST.2018.2808444` (cit. on p. 17).

[18] M. Sepulcre and J. Gozalvez. “Heterogeneous V2V Communications in Multi-Link and Multi-RAT Vehicular Networks”. In: *IEEE Transactions on Mobile Computing* (2019), pp. 1–1. issn: 1536-1233. doi: `10.1109/tmc.2019.2939803` (cit. on p. 17).

[19] V. Mannoni et al. “A comparison of the V2X communication systems: ITS-G5 and C-V2X”. In: *IEEE Vehicular Technology Conference*. Vol. 2019-April. 2019. isbn: 9781728112176. doi: `10.1109/VTCSpring.2019.8746562` (cit. on p. 17).

[20] W. Anwar et al. “On the Reliability of NR-V2X and IEEE 802.11bd”. In: *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*. Vol. 2019-Septe. 2019. isbn: 9781538681107. doi: `10.1109/PIMRC.2019.8904104` (cit. on p. 17).

[21] G. Naik, B. Choudhury, and J. M. Park. “IEEE 802.11bd 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications”. In: *IEEE Access* 7 (2019), pp. 70169–70184. issn: 21693536. doi: `10.1109/ACCESS.2019.2919489`. arXiv: `1903.08391` (cit. on p. 17).

[22] K. Zheng et al. “Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions”. In: *IEEE Communications Surveys and Tutorials* 17.4 (2015), pp. 2377–2396. issn: 1553877X. doi: `10.1109/COMST.2015.2440103` (cit. on pp. 18, 21).

[23] *IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Next Generation V2X*. https://standards.ieee.org/ieee/802.11bd/7451/. [Online; accessed August-2023]. 2023 (cit. on p. 19).

[24] S. Chen et al. “A Vision of C-V2X: Technologies, Field Testing, and Challenges with Chinese Development”. In: *IEEE Internet of Things Journal* 7.5 (2020), pp. 3872–3881. issn: 23274662. doi: `10.1109/JIOT.2020.2974823` (cit. on p. 19).

[25] S. A. A. Shah et al. “5G for Vehicular Communications”. In: *IEEE Communications Magazine* 56.1 (2018), pp. 111–117. issn: 01636804. doi: `10.1109/MCOM.2018.1700467` (cit. on p. 20).

[26] S. Y. Lien et al. “3GPP NR Sidelink Transmissions Toward 5G V2X”. In: *IEEE Access* 8 (2020), pp. 35368–35382. issn: 21693536. doi: `10.1109/ACCESS.2020.2973706` (cit. on p. 20).

[27] B. Fan et al. “Deep Learning Empowered Traffic Offloading in Intelligent Software Defined Cellular V2X Networks”. In: *IEEE Transactions on Vehicular Technology* 9545.c (2020), pp. 1–1. issn: 0018-9545. doi: `10.1109/tvt.2020.3023194` (cit. on pp. 20, 39).

[28] J. Clancy et al. “Feasibility Study of V2X Communications in Initial 5G NR Deployments”. In: *IEEE Access* July (2023), pp. 75269–75284. issn: 21693536. doi: `10.1109/ACCESS.2023.3296089` (cit. on p. 20).

[29] E. Khorov, I. Levitsky, and I. F. Akyildiz. “Current Status and Directions of IEEE 802.11be, the Future Wi-Fi 7”. In: *IEEE Access* 8 (2020), pp. 88664–88688. issn: 21693536. doi: `10.1109/ACCESS.2020.2993448` (cit. on p. 21).

[30] A. Bazzi et al. “Survey and perspectives of vehicular Wi-Fi versus sidelink cellular-V2X in the 5G era”. In: *Future Internet* 11.6 (2019). issn: 19995903. doi: `10.3390/fi11060122` (cit. on p. 21).

[31] K. Ansari. “Joint use of DSRC and C-V2X for V2X communications in the 5.9 GHz ITS band”. In: *IET Intelligent Transport Systems* October 2020 (2020), pp. 213–224. issn: 1751956X. doi: `10.1049/itr2.12015` (cit. on p. 21).

[32] K. Z. Ghafoor et al. “Enabling Efficient Coexistence of DSRC and C-V2X in Vehicular Networks”. In: *IEEE Wireless Communications* 27.2 (2020), pp. 134–140. issn: 15580687. doi: `10.1109/MWC.001.1900219` (cit. on p. 21).

[33] S. Saafi, S. Hamouda, and S. K. Kallel. “Improving handover decisions for better multimedia services transmission in V2I”. In: *IEEE Vehicular Technology Conference* 2018-June (2018), pp. 1–5. issn: 15502252. doi: `10.1109/VTCSpring.2018.8417883` (cit. on pp. 23, 40).

[34] “IEEE Standard for Local and metropolitan area networks–Part 21: Media Independent Services Framework”. In: *IEEE Std 802.21-2017 (Revision of IEEE Std 802.21-2008 as amended by IEEE Std 802.21a-2012, IEEE Std 802.21b-2012, IEEE Std 802.21c-2014, and IEEE Std 802.21d-2015)* (2017), pp. 1–314. doi: `10.1109/IEEESTD.2017.7919341` (cit. on p. 23).

[35] 3. T. 23.402. *“Architecture Enhancement for Non-3GPP Access,” v13.4.0.* 2015 (cit. on p. 23).

[36] 3. T. 24.302. *““Access to the 3GPP Evolved Packet Core (EPC) via Non- 3GPP Access Networks,” Stage 3, v13.4.0.* 2015 (cit. on p. 23).

[37] T. Sylla et al. “Multi-Connectivity for 5G Networks and Beyond: A Survey”. In: *Sensors* 22.19 (2022). issn: 14248220. doi: `10.3390/s22197591` (cit. on p. 24).

[38] M. Li et al. “Multipath Transmission for the Internet: A Survey”. In: *IEEE Communications Surveys and Tutorials* 18.4 (2016), pp. 2887–2925. issn: 1553877X. doi: `10.1109/COMST.2016.2586112` (cit. on p. 24).

[39] G. Dandachi et al. “Network-centric versus user-centric multihoming strategies in LTE/WiFi networks”. In: *IEEE Transactions on Vehicular Technology* 66.5 (2017), pp. 4188–4199. issn: 00189545. doi: `10.1109/TVT.2016.2597442` (cit. on p. 26).

[40] G. Caso et al. “User-Centric Radio Access Technology Selection: A Survey of Game Theory Models and Multi-Agent Learning Algorithms”. In: *IEEE Access* 9 (2021), pp. 84417–84464. issn: 21693536. doi: `10.1109/ACCESS.2021.3087410` (cit. on pp. 27, 39, 40).

[41] 3GPP. *System Architecture for the 5G System, document TS 23.501, v16.4* (cit. on pp. 28, 31).

[42] H. Tong, X. Liu, and C. Yin. “A FAHP and MPTCP Based Seamless Handover Method in Heterogeneous SDN Wireless Networks”. In: *2019 11th International Conference on Wireless Communications and Signal Processing, WCSP 2019* (2019), pp. 1–6. doi: `10.1109/WCSP.2019.8927998` (cit. on p. 28).

[43] H. Tong et al. “Mobility-Aware Seamless Handover with MPTCP in Software-Defined HetNets”. In: *IEEE Transactions on Network and Service Management* 18.1 (2021), pp. 498–510. issn: 19324537. doi: `10.1109/TNSM.2021.3050627`. arXiv: `2101.00678` (cit. on pp. 28, 41).

[44] R. Duo et al. “SDN-based handover scheme in cellular/IEEE 802.11p hybrid vehicular networks”. In: *Sensors (Switzerland)* 20.4 (2020), pp. 1–17. issn: 14248220. doi: `10.3390/s20041082` (cit. on p. 28).

[45] R. I. Meneguette. "Software Defined Networks: Challenges for SDN as an Infrastructure for Intelligent Transport Systems based on Vehicular Networks". In: *Proceedings - 16th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2020* (2020), pp. 205–212. doi: `10.1109/DCOSS49796.2020.00042` (cit. on p. 28).

[46] A. Mahmood, W. E. Zhang, and Q. Z. Sheng. "Software-defined heterogeneous vehicular networking: The architectural design and open challenges". In: *Future Internet* 11.3 (2019). issn: 19995903. doi: `10.3390/fi11030070` (cit. on p. 28).

[47] J. Bhatia et al. "Software defined vehicular networks: A comprehensive review". In: *International Journal of Communication Systems* 32.12 (2019). issn: 10991131. doi: `10.1002/dac.4005` (cit. on p. 28).

[48] K. Z. Ghafoor et al. "Quality of service aware routing protocol in software-defined internet of vehicles". In: *IEEE Internet of Things Journal* 6.2 (2019), pp. 2817–2828. issn: 23274662. doi: `10.1109/JIOT.2018.2875482` (cit. on p. 28).

[49] A. Keshavarz-Haddad et al. "HetNets Selection by Clients: Convergence, Efficiency, and Practicality". In: *IEEE/ACM Transactions on Networking* 25.1 (2017), pp. 406–419. issn: 10636692. doi: `10.1109/TNET.2016.2587622` (cit. on pp. 28, 39, 40).

[50] A. Zhu et al. "Adaptive Multiservice Heterogeneous Network Selection Scheme in Mobile Edge Computing". In: *IEEE Internet of Things Journal* 6.4 (2019), pp. 6862–6875. issn: 23274662. doi: `10.1109/JIOT.2019.2912155` (cit. on pp. 29, 39).

[51] J. Lianghai et al. "Multi-RATs support to improve V2X communication". In: *IEEE Wireless Communications and Networking Conference, WCNC*. Vol. 2018-April. 2018, pp. 1–6. isbn: 9781538617342. doi: `10.1109/WCNC.2018.8377432`. arXiv: `1802.08463` (cit. on pp. 30, 31).

[52] E. Ndashimye et al. "Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey". In: *Computer Networks* 112 (2017), pp. 144–166. issn: 13891286. doi: `10.1016/j.comnet.2016.11.008`. url: `http://dx.doi.org/10.1016/j.comnet.2016.11.008` (cit. on pp. 30, 31).

[53] M. T. Suer et al. "Multi-Connectivity as an Enabler for Reliable Low Latency Communications - An Overview". In: *IEEE Communications Surveys and Tutorials* 22.1 (2020), pp. 156–169. issn: 1553877X. doi: `10.1109/COMST.2019.2949750` (cit. on p. 30).

[54] M. M. Rosello. "Multi-path Scheduling with Deep Reinforcement Learning". In: *2019 European Conference on Networks and Communications, EuCNC 2019* (2019), pp. 400–405. doi: `10.1109/EuCNC.2019.8802063` (cit. on p. 30).

[55] A. Zekri and W. Jia. *Heterogeneous vehicular communications: A comprehensive study*. 2018. doi: `10.1016/j.adhoc.2018.03.010` (cit. on pp. 30, 31, 80).

[56] S. R. Pokhrel, N. Kumar, and A. Walid. “Towards Ultra Reliable Low Latency Multipath TCP for Connected Autonomous Vehicles”. In: *IEEE Transactions on Vehicular Technology* 70.8 (2021), pp. 8175–8185. issn: 0018-9545. doi: `10.1109/tvt.2021.3093632` (cit. on pp. 30, 32).

[57] L. Chao et al. “A brief review of multipath tcp for vehicular networks”. In: *Sensors* 21.8 (2021), pp. 1–34. issn: 14248220. doi: `10.3390/s21082793` (cit. on pp. 30, 32, 80).

[58] N. Williams et al. “Multipath TCP in Vehicular to Infrastructure Communications”. In: *Grenville Armitage Centre for Advanced Internet Architectures Tech. Reports A. 140828* August (2014), p. 15. doi: `10.1.1.648.8732` (cit. on p. 30).

[59] I. Khan and K. Chen. “EBA: Efficient Bandwidth Aggregation for Connected Vehicles with MPTCP”. In: *IEEE Internet of Things Journal* (2021). issn: 23274662. doi: `10.1109/JIOT.2021.3065911` (cit. on pp. 30, 32, 60).

[60] M. R. Palash and K. Chen. “MPWiFi: Synergizing MPTCP based simultaneous multipath access and WiFi network performance”. In: *IEEE Transactions on Mobile Computing* 19.1 (2020), pp. 142–158. issn: 15580660. doi: `10.1109/TMC.2018.2889059` (cit. on p. 30).

[61] H. Zhang, F. Bai, and X. Ju. “Heterogeneous vehicular wireless networking: A theoretical perspective”. In: *2015 IEEE Wireless Communications and Networking Conference, WCNC 2015* (2015), pp. 1936–1941. doi: `10.1109/WCNC.2015.7127764` (cit. on p. 31).

[62] N. Williams et al. “Multipath TCP in Vehicular to Infrastructure Communications”. In: *Grenville Armitage Centre for Advanced Internet Architectures Tech. Reports A. 140828* August (2014), p. 15. doi: `10.1.1.648.8732` (cit. on pp. 31, 58).

[63] R. Jacob et al. “Exploiting Multi-RAT diversity in vehicular ad-hoc networks to improve reliability of cooperative automated driving applications”. In: *IEEE Vehicular Technology Conference* 2019-Septe (2019), pp. 1–7. issn: 15502252. doi: `10.1109/VTCFall.2019.8891072` (cit. on p. 31).

[64] Z. H. Mir et al. “Enabling DSRC and C-V2X Integrated Hybrid Vehicular Networks: Architecture and Protocol”. In: *IEEE Access* 8 (2020), pp. 180909–180927. issn: 21693536. doi: `10.1109/access.2020.3027074` (cit. on p. 31).

[65] L. Miao, J. J. Virtusio, and K. L. Hua. “Pc5-based cellular-v2x evolution and deployment”. In: *Sensors (Switzerland)* 21.3 (2021), pp. 1–14. issn: 14248220. doi: `10.3390/s21030843` (cit. on p. 31).

[66] Q. De Coninck and O. Bonaventure. “Observing network handovers with multipath TCP”. In: *SIGCOMM 2018 - Proceedings of the 2018 Posters and Demos, Part of SIGCOMM 2018* (2018), pp. 54–56. doi: `10.1145/3234200.3234214` (cit. on pp. 31, 57).

[67] *Linux Kernel MultiPath TCP project (out-of-tree kernels before v5.6)*. https://multipath-tcp.org/. [Online; accessed August-2023]. 2023 (cit. on pp. 31, 44).

[68] D. Zhu et al. “Mobility-aware multimedia data transfer using Multipath TCP in Vehicular Network”. In: *2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017* (2017), pp. 1041–1046. doi: `10.1109/IWCMC.2017.7986429` (cit. on pp. 31, 32).

[69] J. Mena, P. Bankole, and M. Gerla. “Multipath TCP on a VANET: A Performance Study”. In: *SIGMETRICS 2017 Abstracts - Proceedings of the 2017 ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems* (2017), pp. 39–40. doi: `10.1145/3078505.3078555` (cit. on p. 31).

[70] J. Mena, P. Bankole, and M. Gerla. “Multipath TCP on a VANET”. In: *Performance Evaluation Review* 45.1 (2017), pp. 39–40. issn: 01635999. doi: `10.1145/3078505.3078555` (cit. on p. 31).

[71] S. Rene et al. “Multipath TCP architecture for infotainment multimedia applications in vehicular networks”. In: *IEEE Vehicular Technology Conference* 2015 (2015), pp. 1–5. issn: 15502252. doi: `10.1109/VTCSpring.2015.7145942` (cit. on pp. 31, 32).

[72] L. Ding et al. “Understanding commercial 5G and its implications to (Multipath) TCP”. In: *Computer Networks* 198 (2021). issn: 13891286. doi: `10.1016/j.comnet.2021.108401` (cit. on pp. 31, 33).

[73] IETF. *Path Aware Networking Research Group*. https://irtf.org/panrg. [Online; accessed December-2022]. 2022 (cit. on p. 31).

[74] Q. De Coninck et al. “Multiflow QUIC : A Generic Multipath Transport Protocol”. In: *IEEE Communications Magazine* 59.May (2021), pp. 108–113. issn: 15581896. doi: `10.1109/MCOM.001.2000892` (cit. on p. 32).

[75] Q. De Coninck and O. Bonaventure. “Multipath QUIC”. In: (2017), pp. 160–166. doi: `10.1145/3143361.3143370` (cit. on p. 32).

[76] M. Kosek, T. Shreedhar, and V. Bajpai. “Beyond QUIC v1 : A First Look at Recent Transport Layer IETF Standardization Efforts”. In: *IEEE Communications Magazine* 59.April (2021), pp. 24–29. issn: 15581896. doi: `10.1109/MCOM.001.2000877`. arXiv: `2102.07527` (cit. on p. 32).

[77] Z. Krämer et al. “On the Potential of MP-QUIC as Transport Layer Aggregator for Multiple Cellular Networks”. In: *Electronics (Switzerland)* 11.9 (2022), pp. 1–15. issn: 20799292. doi: `10.3390/electronics11091492` (cit. on p. 32).

[78] J. Manzoor et al. “On the Performance of QUIC over Wireless Mesh Networks”. In: *Journal of Network and Systems Management* 28.4 (2020), pp. 1872–1901. issn: 15737705. doi: `10.1007/s10922-020-09563-8`. url: `https://doi.org/10.1007/s10922-020-09563-8` (cit. on p. 32).

[79] H. Wu et al. “Multipath Scheduling for 5G Networks: Evaluation and Outlook”. In: *IEEE Communications Magazine* 59.4 (2021), pp. 44–50. issn: 0163-6804. doi: `10.1109/mcom.001.2000881` (cit. on pp. 33, 42, 57).

[80] M. Karimzadeh et al. “MultiPath TCP to support user’s mobility in future LTE network”. In: *Proceedings - WMNC 2017: 10th Wireless and Mobile Networking Conference* 2018-Janua (2017), pp. 1–8. doi: `10.1109/WMNC.2017.8248845` (cit. on p. 33).

[81] W. Tang et al. “A MPTCP Scheduler Combined with Congestion Control for Short Flow Delivery in Signal Transmission”. In: *IEEE Access* 7 (2019), pp. 116195–116206. issn: 21693536. doi: `10.1109/ACCESS.2019.2933880` (cit. on p. 34).

[82] M. K. Zhang et al. “Multipath TCP Scheduling Optimization Based on Q-Learning in Vehicular Heterogeneous Network”. In: *Yingyong Kexue Xuebao/Journal of Applied Sciences* 37.2 (2019), pp. 261–270. issn: 02558297. doi: `10.3969/j.issn.0255-8297.2019.02.011` (cit. on p. 34).

[83] B. Y. Kimura, D. C. Lima, and A. A. Loureiro. “Packet Scheduling in Multipath TCP: Fundamentals, Lessons, and Opportunities”. In: *IEEE Systems Journal* 15.1 (2021), pp. 1445–1457. issn: 19379234. doi: `10.1109/JSYST.2020.2965471` (cit. on p. 34).

[84] B. Y. L. Kimura and A. A. F. Loureiro. “MPTCP linux kernel congestion controls”. In: *arXiv* (2018), pp. 1–3. issn: 23318422. doi: `arXiv:1812.03210v1`. arXiv: `1812.03210` (cit. on p. 34).

[85] B. He et al. “DeepCC: Multi-Agent Deep Reinforcement Learning Congestion Control for Multi-Path TCP Based on Self-Attention”. In: *IEEE Transactions on Network and Service Management* 18.4 (2021), pp. 4770–4788. issn: 19324537. doi: `10.1109/TNSM.2021.3093302` (cit. on p. 35).

[86] Z. Xu et al. “Experience-Driven Congestion Control: When Multi-Path TCP Meets Deep Reinforcement Learning”. In: *IEEE Journal on Selected Areas in Communications* 37.6 (2019), pp. 1325–1336. issn: 15580008. doi: `10.1109/JSAC.2019.2904358` (cit. on p. 35).

[87] W. Li et al. “SmartCC: A Reinforcement Learning Approach for Multipath TCP Congestion Control in Heterogeneous Networks”. In: *IEEE Journal on Selected Areas in Communications* 37.11 (2019), pp. 2621–2633. issn: 15580008. doi: `10.1109/JSAC.2019.2933761` (cit. on p. 35).

[88] M. Chen et al. “Improvement and Implementation of a Multi-Path Management Algorithm based on MPTCP”. In: *Proceedings - Conference on Local Computer Networks, LCN* 2020-Novem (2020), pp. 134–143. doi: `10.1109/LCN48667.2020.9314778` (cit. on p. 36).

[89] D. Ishida, D. Nobayashi, and T. Ikenaga. “Path-selection method based on the available bandwidth of interfaces”. In: *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018* (2018), pp. 815–818. doi: `10.1109/CSCI46756.2018.00163` (cit. on p. 36).

[90] A. Ford et al. *TCP Extensions for Multipath Operation with Multiple Addresses*. RFC 6824. 2013-01. doi: `10.17487/RFC6824`. url: `https://www.rfc-editor.org/info/rfc6824` (cit. on pp. 37, 44).

[91] A. Ford et al. *TCP Extensions for Multipath Operation with Multiple Addresses*. RFC 8684. 2020-03. doi: `10.17487/RFC8684`. url: `https://www.rfc-editor.org/info/rfc8684` (cit. on pp. 37, 87).

[92] O. Othman. *Using Upstream MPTCP in Linux Systems*. Tech. rep. (cit. on pp. 38, 62, 87).

[93] https://github.com/multipath-tcp/mptcp_net-next. [Online; accessed December-2022]. 2020 (cit. on p. 38).

[94] S. Ferlin et al. "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks". In: *2016 IFIP Networking Conference (IFIP Networking) and Workshops, IFIP Networking 2016* (2016), pp. 431–439. doi: `10.1109/IFIPNetworking.2016.7497206` (cit. on p. 38).

[95] Z. Mahmood. *Chapter 3 Seamless V2I Communication in HetNet: State-of-the-Art Seamless V2I Communication in HetNet: State-of-the-Art and Future Research Directions in HetNet: State-of-the-Art and Future Research Directions (BOOK)*. 2020. isbn: 978-3-030-36166-2. doi: `10.1007/978-3-030-36167-9` (cit. on p. 39).

[96] I. Tal and G.-M. Muntean. "Towards Reasoning Vehicles: A Survey of Fuzzy Logic-Based Solutions in Vehicular Networks". In: *ACM Computing Surveys* 50 (2017-12), pp. 1–37. doi: `10.1145/3125640` (cit. on p. 39).

[97] F. Bouali, K. Moessner, and M. Fitch. "A context-aware user-driven framework for network selection in 5G multi-RAT environments". In: *IEEE Vehicular Technology Conference* 0 (2016). issn: 15502252. doi: `10.1109/VTCFall.2016.7880848` (cit. on pp. 39, 40).

[98] S. Bhadouria and R. Roshan. "Network selection by vertical handoff in heterogeneous vehicular network using fuzzy MADM-TOPSIS". In: *Lecture Notes in Electrical Engineering* 472 (2018), pp. 317–324. issn: 18761119. doi: `10.1007/978-981-10-7395-3_36` (cit. on pp. 39, 40).

[99] H. W. H.-W. Yu and B. Zhang. "A Hybrid MADM Algorithm Based on Attribute Weight and Utility Value for Heterogeneous Network Selection". In: *Journal of Network and Systems Management* 27.3 (2019), pp. 756–783. issn: 10647570. doi: `10.1007/s10922-018-9483-y` (cit. on pp. 39, 41, 80, 96).

[100] H. W. Yu and B. Zhang. "A heterogeneous network selection algorithm based on network attribute and user preference". In: *Ad Hoc Networks* 72 (2018), pp. 68–80. issn: 15708705. doi: `10.1016/j.adhoc.2018.01.011` (cit. on p. 39).

[101] A. Zhu et al. "Adaptive Access Selection Algorithm for Multi-Service in 5G Heterogeneous Internet of Things". In: *IEEE Transactions on Network Science and Engineering* 9.3 (2022), pp. 1630–1644. issn: 23274697. doi: `10.1109/TNSE.2022.3148399` (cit. on p. 39).

[102] X. Guo et al. “Multiattribute Access Selection Algorithm for Heterogeneous Wireless Networks Based on Fuzzy Network Attribute Values”. In: *IEEE Access* 10.July (2022), pp. 74071–74081. issn: 21693536. doi: `10.1109/ACCESS.2022.3186149` (cit. on p. 39).

[103] J. M. Marquez-Barja et al. “Breaking the vehicular wireless communications barriers: Vertical handover techniques for heterogeneous networks”. In: *IEEE Transactions on Vehicular Technology* 64.12 (2015), pp. 5878–5890. issn: 00189545. doi: `10.1109/TVT.2014.2386911` (cit. on pp. 39, 100).

[104] T. Bhattasali et al. “Network Selection Using AHP for Fast Moving Vehicles in Heterogeneous Networks”. In: *Advances in Intelligent Systems and Computing* 395.November (2016), pp. 65–84. issn: 21945357. doi: `10.1007/978-81-322-2650-5` (cit. on p. 39).

[105] A. Aboud, H. Touati, and B. Hnich. “Markov Chain based Predictive Model for Efficient handover Management in Vehicle-to-Infrastructure Communications”. In: *2021 International Wireless Communications and Mobile Computing, IWCMC 2021* (2021), pp. 1117–1122. doi: `10.1109/IWCMC51323.2021.9498927` (cit. on p. 39).

[106] X. Wang et al. “Intelligent User-Centric Network Selection: A Model-Driven Reinforcement Learning Framework”. In: *IEEE Access* 7 (2019), pp. 21645–21661. issn: 21693536. doi: `10.1109/ACCESS.2019.2898205` (cit. on p. 39).

[107] L. Li et al. “A reinforcement sensor embedded vertical handoff controller for vehicular heterogeneous wireless networks”. In: *Sensors (Switzerland)* 13.11 (2013), pp. 15026–15047. issn: 14248220. doi: `10.3390/s131115026` (cit. on p. 39).

[108] S. J. Siddiqi et al. “Towards AI-enabled traffic management in multipath TCP: A survey”. In: *Computer Communications* 181 (2022), pp. 412–427. issn: 1873703X. doi: `10.1016/j.comcom.2021.09.030` (cit. on pp. 39, 40).

[109] J. Sun et al. “ES-DQN-Based Vertical Handoff Algorithm for Heterogeneous Wireless Networks”. In: *IEEE Wireless Communications Letters* 9.8 (2020), pp. 1327–1330. issn: 21622345. doi: `10.1109/LWC.2020.2990713` (cit. on p. 39).

[110] T. Taleb and A. Ksentini. “VECOS: A vehicular connection steering protocol”. In: *IEEE Transactions on Vehicular Technology* 64.3 (2015), pp. 1171–1187. issn: 00189545. doi: `10.1109/TVT.2014.2327241` (cit. on p. 40).

[111] E. Ndashimye, N. I. Sarkar, and S. K. Ray. “A Multi-criteria based handover algorithm for vehicle-to-infrastructure communications”. In: *Computer Networks* 185.November 2020 (2021), p. 107652. issn: 13891286. doi: `10.1016/j.comnet.2020.107652`. url: `https://doi.org/10.1016/j.comnet.2020.107652` (cit. on p. 40).

[112] H. Yu, Y. Ma, and J. Yu. “Network Selection Algorithm for Multiservice Multimode Terminals in Heterogeneous Wireless Networks”. In: *IEEE Access* 7 (2019), pp. 46240–46260. issn: 21693536. doi: `10.1109/ACCESS.2019.2908764` (cit. on p. 40).

[113] Q. Hu et al. “Adaptive cross-layer handover algorithm based on MPTCP for hybrid LiFi-and-WiFi networks”. In: *Ad Hoc Networks* 134 (2022). issn: 15708705. doi: `10.1016/j.adhoc.2022.102923` (cit. on p. 40).

[114] *ns-3 Home Page*. http://www.nsnam.org. [Online; accessed May-2023]. 2020 (cit. on p. 42).

[115] J. S. Weber and M. Neves. “Open Access VANET simulators : an updated review”. In: (2021) (cit. on p. 42).

[116] T. Zugno et al. “Extending the ns-3 spatial channel model for vehicular scenarios”. In: *ACM International Conference Proceeding Series* (2021), pp. 25–32. doi: `10.1145/3460797.3460801` (cit. on p. 42).

[117] G. Nardini et al. “Simu5G–An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks”. In: *IEEE Access* 8 (2020), pp. 181176–181191. issn: 21693536. doi: `10.1109/ACCESS.2020.3028550` (cit. on p. 42).

[118] V. T. Le et al. “A multi-domain network simulator based on NS-3”. In: *SIMULTECH 2020 - Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* September (2020), pp. 217–224. doi: `10.5220/0009831602170224` (cit. on p. 42).

[119] T. K. Lee et al. “Building a V2X Simulation Framework for Future Autonomous Driving”. In: *2019 20th Asia-Pacific Network Operations and Management Symposium: Management in a Cyber-Physical World, APNOMS 2019*. Institute of Electrical and Electronics Engineers Inc., 2019. isbn: 9784885523205. doi: `10.23919/APNOMS.2019.8892860` (cit. on p. 42).

[120] V. Venkataramanan and S. Lakshmi. “A case study of various wireless network simulation tools”. In: *International Journal of Communication Networks and Information Security* 10.2 (2018), pp. 389–396. issn: 2073607X. doi: `10.13140/RG.2.2.32223.30885` (cit. on p. 42).

[121] T. Anh, Q. Pham, and Y. Hadjadj-aoul. *An ns-3 MPTCP Implementation.* Vol. 1. Springer International Publishing, 2019, pp. 14–26. isbn: 9783030144135. doi: `10.1007/978-3-030-14413-5` (cit. on p. 42).

[122] J. Lamps et al. “Conjoining emulation and network simulators on linux multiprocessors”. In: *SIGSIM-PADS 2015 - Proceedings of the 3rd ACM Conference on SIGSIM-Principles of Advanced Discrete Simulation* (2015), pp. 113–124. doi: `10.1145/2769458.2769481` (cit. on p. 42).

[123] M. Prakash and A. Abdrabou. “On the fidelity of ns-3 simulations of wireless multipath tcp connections”. In: *Sensors (Switzerland)* 20.24 (2020), pp. 1–23. issn: 14248220. doi: `10.3390/s20247289` (cit. on p. 42).

[124] M. Coudron and S. Secci. “An implementation of multipath TCP in ns3”. In: *Computer Networks* 116 (2017), pp. 1–11. issn: 13891286. doi: `10.1016/j.comnet.2017.02.002` (cit. on p. 42).

[125] A. R. Portabales and M. L. Nores. “Dockemu: Extension of a scalable network simulation framework based on docker and ns3 to cover iot scenarios”. In: *SIMULTECH 2018 - Proceedings of 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* Simultech (2018), pp. 175–182. doi: `10.5220/0006913601750182` (cit. on p. 42).

[126] E. Petersen, G. Cotto, and M. Antonio To. “Dockemu 2.0: Evolution of a Network Emulation Tool”. In: *2019 IEEE 39th Central America and Panama Convention, CONCAPAN 2019* 2019-Novem (2019). doi: `10.1109/CONCAPANXXXIX47272.2019.8977002` (cit. on p. 42).

[127] G. Shah, Y. P. Fallah, and S. D. Gupta. “RVE-CV2X : A Scalable Emulation Framework for Real-Time Evaluation of CV2X-based Connected Vehicle Applications”. In: (2020), pp. 135–142. arXiv: `arXiv:2101.06548v1` (cit. on p. 42).

[128] M. A. To, M. Cano, and P. Biba. “DOCKEMU - A Network Emulation Tool”. In: *Proceedings - IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2015* (2015), pp. 593–598. doi: `10.1109/WAINA.2015.107` (cit. on p. 42).

[129] *CORE (Common Open Research Emulator).* https://coreemu.github.io/core/index.html. [Online; accessed August-2023]. 2020 (cit. on p. 42).

[130] D. Krajzewicz et al. “SUMO (Simulation of Urban MObility); An open-source traffic simulation”. In: 2002-01, pp. 183–187. isbn: 90-77039-09-0 (cit. on p. 42).

[131] A. Fouda et al. “Real-Time Video Streaming over NS3-based Emulated LTE Networks”. In: *International Journal of Electronics Communication and Computer Technology* 4.3 (2014), pp. 659–663 (cit. on p. 44).

[132] *5G-LENA simulator.* https://5g-lena.cttc.es/. [Online; accessed August-2023]. 2023 (cit. on p. 45).

[133] https://www.nsnam.org/wiki/NetAnim_3.108. [Online; accessed October-2023]. 2020 (cit. on p. 46).

[134] M. Malinverno et al. “A Multi-stack Simulation Framework for Vehicular Applications Testing”. In: *DIVANet 2020 - Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications* March 2021 (2020), pp. 17–24. doi: `10.1145/3416014.3424603` (cit. on p. 46).

[135] S. Ferlin et al. “MPTCP Meets FEC: Supporting latency-sensitive applications over heterogeneous networks”. In: *IEEE/ACM Transactions on Networking* 26.5 (2018), pp. 2005–2018. issn: 10636692. doi: `10.1109/TNET.2018.2864192`. arXiv: `1807.11059` (cit. on p. 51).

[136] J. Yan and J. Dong. “A Virtual Time System for Linux-container-based Emulation of Software-defined Networks”. In: (), pp. 235–246 (cit. on p. 53).

[137] T. Zhu et al. “GCLR: GNN-Based cross layer optimization for multipath TCP by routing”. In: *IEEE Access* 8 (2020), pp. 17060–17070. issn: 21693536. doi: `10.1109/ACCESS.2020.2966045` (cit. on p. 55).

[138] J. Hochst et al. “Learning Wi-Fi Connection Loss Predictions for Seamless Vertical Handovers Using Multipath TCP”. In: *Proceedings - Conference on Local Computer Networks, LCN*. Vol. 2019-Octob. 2019, pp. 18–25. isbn: 9781728110288. doi: `10.1109/LCN44214.2019.8990753`. arXiv: `1907.10493` (cit. on p. 56).

[139] M. S. Hossain and M. Atiquzzaman. “Analysis of Proxy Mobile IPv6: A network-based mobility solution”. In: *Proceeding of the 15th International Conference on Computer and Information Technology, ICCIT 2012* (2012), pp. 338–344. doi: `10.1109/ICCITechn.2012.6509795` (cit. on p. 57).

[140] V. A. Vu and B. Walker. “Redundant multipath-TCP scheduling with desired packet latency”. In: *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM* October (2019), pp. 7–12. doi: `10.1145/3349625.3355440` (cit. on p. 57).

[141] A. Mondal et al. “PPoS: A Novel Sub-flow Scheduler and Socket APIs for Multipath TCP (MPTCP)”. In: *2018 24th National Conference on Communications, NCC 2018* (2019), pp. 1–6. doi: `10.1109/NCC.2018.8600192` (cit. on p. 57).

[142] Y. S. Lim et al. “ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths”. In: *Performance Evaluation Review* 45.1 (2017), pp. 33–34. issn: 01635999. doi: `10.1145/3078505.3078552` (cit. on p. 57).

[143] F. Yang, P. Amer, and N. Ekiz. “A scheduler for multipath TCP”. In: *Proceedings - International Conference on Computer Communications and Networks, ICCCN* (2013). issn: 10952055. doi: `10.1109/ICCCN.2013.6614091` (cit. on p. 58).

[144] F. Chiariotti et al. “Analysis and Design of a Latency Control Protocol for Multi-Path Data Delivery with Pre-Defined QoS Guarantees”. In: *IEEE/ACM Transactions on Networking* 27.3 (2019), pp. 1165–1178. issn: 15582566. doi: `10.1109/TNET.2019.2911122` (cit. on p. 59).

[145] H. Sinky, B. Hamdaoui, and M. Guizani. “Seamless Handoffs in Wireless HetNets: Transport-Layer Challenges and Multi-Path TCP Solutions with Cross-Layer Awareness”. In: *IEEE Network* 33.2 (2019), pp. 195–201. issn: 1558156X. doi: `10.1109/MNET.2018.1800017` (cit. on p. 61).

[146] B. Liao et al. “Cross-layer path selection in multi-path transport protocol for mobile devices”. In: *arXiv Computer Science* (2020). issn: 23318422. doi: `arXiv:2007.01536v1`. arXiv: `2007.01536` (cit. on p. 61).

[147] K. Nguyen et al. “An approach to reinforce multipath TCP with path-aware information”. In: *Sensors (Switzerland)* 19.3 (2019), pp. 1–12. issn: 14248220. doi: `10.3390/s19030476` (cit. on p. 61).

[148] T. Shreedhar et al. “QAware: A Cross-Layer Approach to MPTCP Scheduling”. In: *17th International IFIP TC6 Networking Conference, Networking 2018*. 2018, pp. 190–198. isbn: 9783903176089. doi: `arXiv:1808.04390v1`. arXiv: `1808.04390` (cit. on p. 61).

[149] M. Fukuyama et al. “Throughput Improvement of MPTCP by Selective Bicasting with Cross-Layer Control in Wireless Environment”. In: *Proceedings - International Computer Software and Applications Conference* 2 (2018), pp. 204–209. issn: 07303157. doi: `10.1109/COMPSAC.2018.10230` (cit. on p. 61).

[150] B. Hesmans et al. “SMAPP: Towards smart multipath TCP-enabled applications”. In: *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015* (2015). doi: `10.1145/2716281.2836113` (cit. on pp. 61, 87).

[151] B. Hesmans and O. Bonaventure. “An enhanced socket API for Multipath TCP”. In: *ANRW 2016 - Proceedings of the ACM, IRTF and ISOC Applied Networking Research Workshop* (2016), pp. 1–6. doi: `10.1145/2959424.2959433` (cit. on p. 61).

[152] *ISO 21217:2014, Intelligent transport systems — Communications access for land mobiles (CALM) — Architecture*. 2014 (cit. on p. 86).

[153] S. Singh et al. “NexGen S-MPTCP: Next Generation Smart MultiPath TCP Controller”. In: *IEEE Access* (2022). issn: 21693536. doi: `10.1109/ACCESS.2022.3205418` (cit. on p. 87).

[154] Y. Liu, C. M. Eckert, and C. Earl. “A review of fuzzy AHP methods for decision-making with subjective judgements”. In: *Expert Systems with Applications* 161 (2020), p. 113738. issn: 09574174. doi: `10.1016/j.eswa.2020.113738`. url: `https://doi.org/10.1016/j.eswa.2020.113738` (cit. on p. 96).

[155] T. L. Saaty. “How to make a decision: The analytic hierarchy process”. In: *European Journal of Operational Research* 48.1 (1990), pp. 9–26. issn: 03772217. doi: `10.1016/0377-2217(90)90057-I` (cit. on p. 96).

[156] M. B. Ayhan. “A Fuzzy Ahp Approach For Supplier Selection Problem: A Case Study In A Gearmotor Company”. In: *International Journal of Managing Value and Supply Chains* 4.3 (2013), pp. 11–23. issn: 22307966. doi: `10.5121/ijmvsc.2013.4302`. arXiv: `1311.2886` (cit. on p. 96).

Appendix A

Userspace path manager

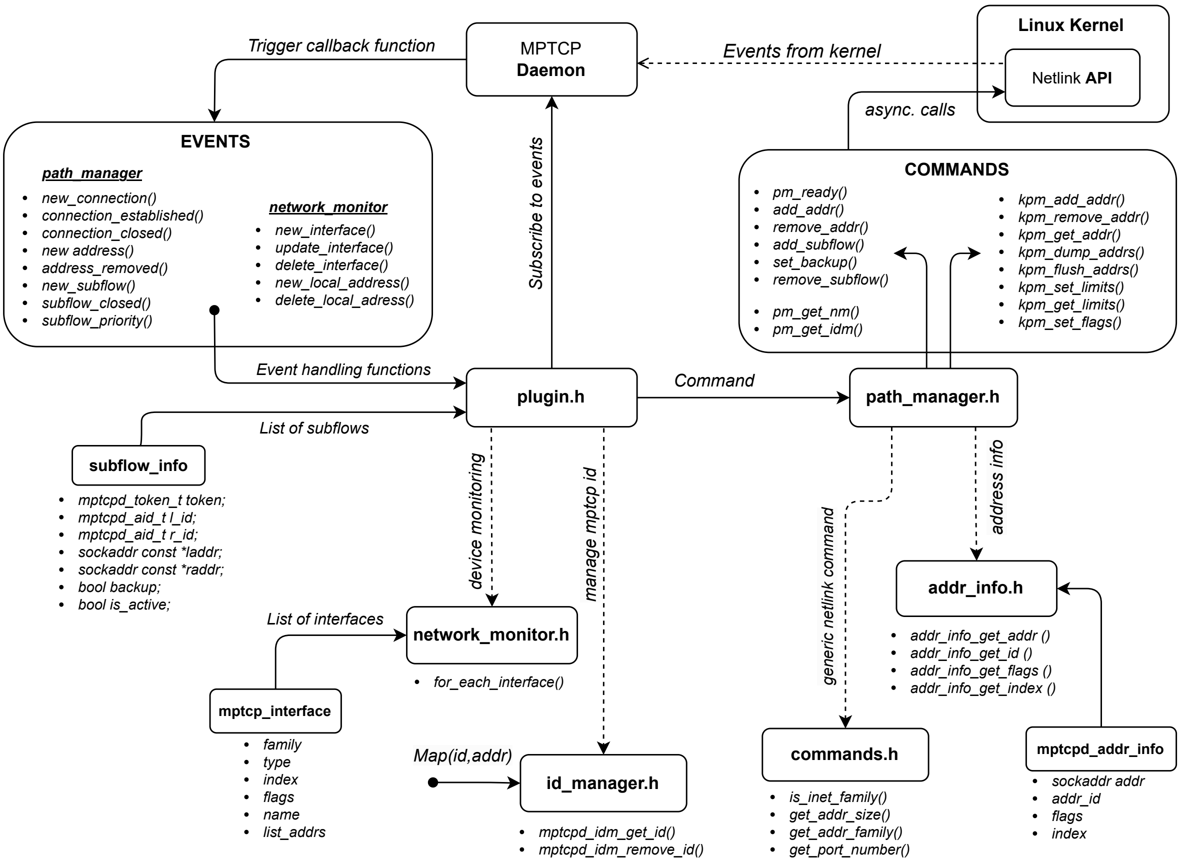


Figure 36: The building blocks of the userspace path manager.

Appendix B

Calculation of the criteria weigth vector

```
###############################################################################
This code implements the Fuzzy AHP method to assign weights to criteria or
alternatives in a decision-making process. It is used to handle uncertain and
imprecise data in multi-criteria decision analysis, providing a structured way
to make informed decisions based on a set of criteria.

The function returns four sets of values: the fuzzy weights (f_w),
crisp weights (d_w), normalized weights (n_w), and the consistency ratio (rc).

###############################################################################

# Required Libraries
import numpy as np

# Function: Fuzzy AHP
def fuzzy_ahp_method(dataset):
    row_sum = []
    s_row   = []
    f_w     = []
    d_w     = []
    inc_rat  = np.array([0, 0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41,
                        1.45, 1.49, 1.51, 1.48, 1.56, 1.57, 1.59])
    X = []
    for i in range(len(dataset)):
        for item in dataset[i]:
            value = (item[0] + 4 * item[1] + item[2]) / 6
```

```
            X.append(value)
    X  = np.asarray(X)
    X  = np.reshape(X, (len(dataset), len(dataset)))
    for i in range(0, len(dataset)):
        a, b, c = 1, 1, 1
        for j in range(0, len(dataset[i])):
            d, e, f = dataset[i][j]
            a, b, c = a*d, b*e, c*f
        row_sum.append( (a, b, c) )
    L, M, U = 0, 0, 0
    for i in range(0, len(row_sum)):
        a, b, c = row_sum[i]
        a = a**(1/len(dataset))
        b = b**(1/len(dataset))
        c = c**(1/len(dataset))
        s_row.append( ( a, b, c ) )
        L = L + a
        M = M + b
        U = U + c
    for i in range(0, len(s_row)):
        a, b, c = s_row[i]
        a, b, c = a*(U**-1), b*(M**-1), c*(L**-1)
        f_w.append( ( a, b, c ) )
        d_w.append( (a + b + c)/3 )
    n_w      = [item/sum(d_w) for item in d_w]
    vector   = np.sum(X*n_w, axis = 1)/n_w
    lamb_max = np.mean(vector)
    cons_ind = (lamb_max - X.shape[1])/(X.shape[1] - 1)
    rc       = cons_ind/inc_rat[X.shape[1]]
    return f_w, d_w, n_w, rc

###############################################################################
```