Efficient AI and Prediction Techniques for Smart 5G-enabled Vehicular Networks

by

Noura Aljeri

A thesis
submitted to the University of Ottawa
in partial fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

With the recent growth and wide availability of heterogeneous wireless access technologies, inter-vehicle communication systems are expected to culminate in integrating various wireless standards for the next generation of connected and autonomous vehicles. The role of 5G-enabled vehicular networks has become increasingly important, as current Internet clients and providers have urged robustness and effectiveness in digital services over wireless networks to cope with the latest advances in wireless mobile communication. However, to enable 5G wireless technologies' dense diversity, seamless and reliable wireless communication protocols need to be thoroughly investigated in vehicular networks. 5G-enabled vehicular networks applications and services such as routing, mobility management, and service discovery protocols can integrate mobility-based prediction techniques to elevate those applications' performance with various vehicles, applications, and network measurements.

In this thesis, we propose a novel suite of 5G-enabled smart mobility prediction and management schemes and design a roadmap guide to mobility-based predictions for intelligent vehicular network applications and protocols. We present a thorough review and classification of vehicular network architectures and components, in addition to mobility management schemes, benchmarks advantages, and drawbacks. Moreover, multiple mobility-based schemes are proposed, in which vehicles' mobility is managed through the utilization of machine learning prediction and probability analysis techniques. We propose a novel predictive mobility management protocol that incorporates a new networks' infrastructure discovery and selection scheme. Next, we design an efficient handover trigger scheme based on time-series prediction and a novel online neural network-based next roadside unit prediction protocol for smart vehicular networks. Then, we propose an original adaptive predictive location management technique that utilizes vehicle movement projections to estimate the link lifetime between vehicles and infrastructure units, followed by an efficient movement-based collision detection scheme and infrastructure units localization strategy.

Last but not least, the proposed techniques have been extensively evaluated and compared to several benchmark schemes with various networks' parameters and environments. Results showed the high potentials of empowering vehicular networks' mobility-based protocols with the vehicles' future projections and the prediction of the network's status.

List of Publications

Referred Journal Papers

- Boukerche, A., Magnano, A., and Aljeri, N., 2017. Mobile IP Handover for Vehicular Networks: Methods, Models, and Classifications. *ACM Computing Surveys*, 49(4), Article 73.

- Aljeri, N. and Boukerche, A., 2018. Movement Prediction Models for Vehicular Networks: An Empirical Analysis. *Wireless Networks*, 25(4), pp.1505-1518.

- Sun, P., AlJeri, N. and Boukerche, A., 2018. An Energy-Efficient Proactive Handover Scheme for Vehicular Networks Based on Passive RSU Detection. *IEEE Transactions on Sustainable Computing*, 5(1), pp. 37-47.

- Aljeri, N. and Boukerche, A., 2019. A Two-Tier Machine Learning-Based Handover Management Scheme for Intelligent Vehicular Networks. *Ad Hoc Networks*, 94, p.101930.

- Aljeri, N. and Boukerche, A., 2019. A Dynamic MAP Discovery and Selection Scheme for Predictive Hierarchical MIPv6 in Vehicular Networks. *IEEE Transactions on Vehicular Technology*, 69(1), pp.793-806.

- Aljeri, N. and Boukerche, A., 2020. Fog-Enabled Vehicular Networks: A New Challenge for Mobility Management. *Internet Technology Letters*, 3(6), e141.

- Aljeri, N. and Boukerche, A., 2020. Smart and Green Mobility Management for 5G-enebaled Vehicular Networks. *Transactions on emerging telecommunications technologies*, e4054.

- Aljeri, N. and Boukerche, A., 2020. ADVICE-LOC: An Adaptive Vehicle-Centric LOCation Management Scheme for Intelligent Connected Cars, *Ad Hoc Networks*, 107, p.102-223.

- Aljeri, N. and Boukerche, A., 2020. Mobility Management in 5G-enebaled Vehicular Networks: Models, Protocols, and Classifications, *ACM Computing Surveys*, 53(5), Article 92.

Referred Conference Papers

- Aljeri, N., and Boukerche, A., 2017. Performance Evaluation of Movement Prediction Techniques for Vehicular Networks, In Proceedings of the *the IEEE International Conference on Communications (ICC)*, pp.1-6.

- Aljeri, N., and Boukerche, A., 2017. A Predictive Collision Detection Protocol Using Vehicular Networks, In Proceedings of *the IEEE 28th International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp.1-5.

- Sun, P., Aljeri, N., and Boukerche, A. 2017. A Novel Passive Road Side Unit Detection Scheme in Vehicular Networks, In Proceedings of *the IEEE Global Communications Conference (GLOBECOM)*, pp.1-5.

- Aljeri, N., and Boukerche, A. 2018. Mobility and Handoff Management in Connected Vehicular Networks. In Proceedings of *the 16th ACM International Symposium on Mobility Management and Wireless Access*, pp.82-88.

- Aljeri, N., and Boukerche, A. 2018. An Efficient Movement-Based Handover Prediction Scheme for Hierarchical Mobile IPv6 in VANETs. In Proceedings of *the 15th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pp.47-54.

- Sun, P., Aljeri, N., and Boukerche, A. 2018. A Novel Proactive Handover Scheme for Achieving Energy-Efficient Vehicular Networks. In Proceedings of *the 14th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, pp.23-28.

- Aljeri, N., and Boukerche, A. 2019. Probabilistic Neural Network-Based Road-Side Unit Prediction Scheme for Autonomous Driving, In Proceedings of *the IEEE International Conference on Communications (ICC)*, pp.1-6.

- **(Best Paper Award)** Aljeri, N., and Boukerche, A., 2019. A Novel Online Machine Learning Based RSU Prediction Scheme for Intelligent Vehicular Networks,In Proceedings of *IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pp.1-8.

- Aljeri, N., and Boukerche, A. 2019. An Optimized Link Duration-Based Mobility Management Scheme for Connected Vehicular Networks. In Proceedings of *the 16th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pp.7–14.

- Aljeri, N., and Boukerche, A. 2019. An Efficient Handover Trigger Scheme for Vehicular Networks Using Recurrent Neural Networks. In Proceedings of *the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, pp.85–91.

- Aljeri, N., and Boukerche, A. 2020. Load Balancing and QoS-Aware Network Selection Scheme in Heterogeneous Vehicular Networks. *ICC'20, IEEE International Conference on Communications (ICC)*, pp.1-6.

Acknowledgements

First, I would like to thank my supervisor, Prof. Azzedine Boukerche, whom I am proud to be his student, for all his much needed financial support, patience, and inspiring guidance to help me reach my highest potential. From the first day, I trusted you with every step up to this moment, the end of my Ph.D. journey, and the beginning of a new one. His continuous motivation and belief in every student's capabilities and limits made me see what I would like to be in the future. Your support has proved invaluable, Professor! Thank you!

An exceptional gratitude to my parents, Faisal, and Maha, for their unconditional love and kindness, for encouraging and pushing me to pursue my graduate studies. Their immense emotional and moral support along the way made me strong enough to reach this chapter of my life. Mom and Dad, you are my rock!

I want to thank my children, Fares, and Saad, for standing by me, supporting me in their own way, and being in my life. This is all for them! I am thankful to all my family, brothers, sisters, and friends who have encouraged me along the way.

Last but not least, I would like to thank my fellow graduate students in PARADISE Research Laboratory for their friendship, support, and feedback. I am thankful to have the opportunity to work alongside them in the lab.

To my loves, my boys Fares and Saad!

Table of Contents

List of Tables

List of Figures

Abbreviations

3GPP 3rd Generation Partnership Project

5G Fifth Generation of Cellular Networks

AP Access Point

API Application Programming Interface

CCA cooperative collision avoidance

CN Corresponding Node

CoA Care of Address

D2D Device to Device

DMM dynamic mobility management

DSRC Dedication short-range communication

EKF Extended Kalman Filter

FC Fog computing

FRAN Fog-based radio access network

GPS Geographic positioning system

HA Home agent

HetNet Heterogeneous Networks

HMIP Hierarchical Mobile IPv4

HMM Hidden Markov Model

IETF Internet engineering task force

IoT Internet of Things

IP Internet Protocol

IRTF Internet Research Task Force

ITS Intelligent transportation system

KF Kalman Filter

LAN Local area networks

LISP Locator/ID separation protocol

LMA Local Mobility Anchor

LSTM Long-short term memory

LTE Long Term Evolution

MADM Multiple attribute decision making

MAG Mobility Access Gateway

MAN Metropolitan area network

MAP Mobile anchor point

MIH Media independent handover

MIMO Multiple input multiple output

MIP Mobile IP

MME Mobility Management Entity

MN Mobile node

ONF Open Networking Foundation

PF Particle Filter

PMIPv6 Proxy Mobile IPv6

PNN Probabilistic Neural Networks

QoS Quality of Service

RNN Recurrent Neural Network

RSRP Reference Signal Receive Power

RSS Received signal strength

RSUs Roadside Units

SDN Software-defined network

SDVN Software-defined vehicular network

SNIR Signle to noise ratio

SVM Support Vector Machine

TTT Time to trigger

UE User equipment

UKF Unscented Kalman Filter

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

V2X Vehicle to Everything

VANET Vehicular Ad Hoc Networks

VFN Vehicular fog-based network

WAVE Wireless Access in Vehicular Environment

WCNs Wireless Communication Networks

WiFi Wireless Fidelity

WLAN Wireless Local Area Network

Chapter 1

Introduction

The continuous demand for on-the-fly Internet connectivity by users and service providers and the fast expansion of wireless networks has led to the growth of vehicular networks as a vital component for the content delivery of various services and applications [211]. That includes road safety, traffic management services, and infotainment applications, which strongly reflect the volume of data transmissions being processed in current wireless networks. To cope with that growth, current vehicular network standards and protocols must be improved to handle the high increase in data flows while considering vehicular network conditions and limitations.

Mobility management schemes, reliable content delivery services, robust topology management protocols, among others, needed to be evolved intelligently and autonomously. In order to allow Internet services and applications to be delivered through vehicular networks environment, efficient and reliable communication between vehicles and RoadSide Units (RSUs) is essential. The development of IEEE802.11p and many routing protocols attempt to handle the rapid mobility of vehicles [1]. However, the vehicles' rapid mobility and topology changes have a significant impact on such protocols' performance.

When we consider mobility management in particular, many challenges arise to its development for vehicular networks, including service interruptions and transition costs [241]. Related studies to improve the mobility management protocols have been proposed by the research community through the integration of different types of wireless networks [223]. However, this integration still suffers from service disruption during the handover process. Several studies have explored the potentials of using predictive techniques to design intelligent decision-making solutions based on the past and current status of the network conditions and vehicle movement measurements to overcome such issues. Moreover, ve-

hicular network applications and protocols may rely on accurate mobility information to derive a decision such as which network to connect to next?; what is the best route to deliver content information from source to destination?; or how to optimize the networks' overload through context information. Nonetheless, obtaining such knowledge requires efficient prediction techniques to estimate future movement patterns and paths. Additionally, the rapid mobility of vehicles and the degree of error in positioning systems create a challenging problem regarding any mobility prediction model's accuracy and efficiency.

1.1 Motivation

Recent developments and the ubiquitous availability of wireless technologies and inter-vehicle communication systems are anticipated to lead to heterogeneous wireless networks' evolution. Based on standards such as the IEEE 802.11, 5G, and WiMAX, coupled with mobile IP for the content delivery of road safety and infotainment services to drivers over the new generation of vehicular networks. However, the performance of traditional mobility management protocols over vehicular networks is affected by network topology changes and vehicles' high mobility. Therefore, an efficient mobility management solution that mitigates the challenges of vehicles' mobility is needed.

This issue can be improved by connecting with the upcoming RSU early enough and smooth the transition process. The concept of early connection requires the prediction of such transition beforehand, which can be easily adjusted for any mobility management protocol. Since streets, intersections, and roads constrain vehicles' movement, vehicles' future projections are more foreseen. That can be achieved by estimating their future trajectories with respect to time. The available mobility prediction models are mostly based on movement projections, probability modeling, or pattern matching. Each with its benefits and drawbacks in terms of accuracy, complexity, and completeness.

1.2 Objectives and Contributions

This thesis's main objectives are to design and evaluate several mobility prediction-based protocols in efforts to enhance vehicles' connectivity and communication in vehicular network applications. In addition, we combine both temporal and historical data with multiple predictors to increase the accuracy of each estimation. In Figure 1.1, we illustrate the overall design of the used framework throughout the thesis, in which the process is divided into four main steps, the network's design, monitoring, prediction, and decision-making. The

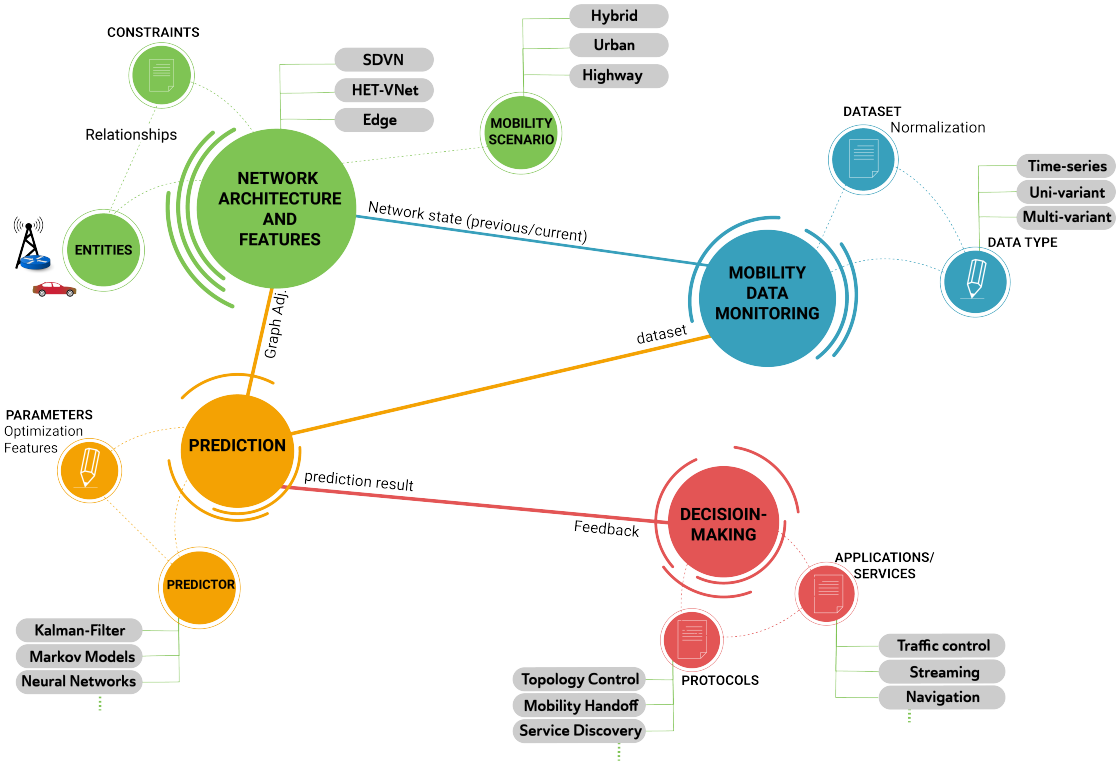


Figure 1.1: Framework overview.

network design is an essential part of the next steps, as it defines the relationships and constraints between network entities (i.e., vehicles, roadside units, base stations), in addition to the type of measurements and information that can be used in the prediction step (i.e., vehicles' trajectory, networks status). In the monitoring step, valuable information is collected in real-time and converted to data sequences as inputs to the prediction step. In the prediction steps and based on the used techniques, each predictor has a set of parameters that needs to be optimized depending on the shape, length, and type of dataset used. Finally, the predictor results are used by vehicular network's protocol to derive a decision that leads to increased networks' performance and quality of service. The contributions made in this thesis are listed below

- A comprehensive classification of vehicular network design and architecture followed by evaluating mobility management protocols based on standards, benchmarks, and deployed network architectures. We discuss each mobility management classification solutions, issues and challenges, possible solutions, and future research directions.

- A performance evaluation of movement prediction techniques for vehicular networks. The comparison between several predictors is evaluated based on different mobility models, including real and synthetic datasets of vehicles' mobility. We provide insights into their capabilities and their limitations.

- An efficient collision detection algorithm that uses the wireless communication standards in VANETs to accurately detect potential accidents. We adopt a movement-based prediction technique to predict the vehicles future trajectories. Thus, reducing the network communication overhead.

- A novel and simple passive RSU location estimation scheme, by which the vehicle can pre-determine the desired RSU to communicate based on its own route information. We utilize the doppler effects and the maximum likelihood estimation method to derive an estimation.

- A novel predictive hierarchical handover protocol for mobile IP in vehicular networks. We combine the stochastic probability analysis of a hidden Markov model, and the vehicles' movement projection to predict the next handoff. The proposed scheme outperforms all other handover protocols in reducing the handover latency and packet loss.

- A novel dynamic Mobile Anchor Point (MAP) discovery and selection scheme for mobility management in vehicular networks. The dynamic network topology control improves the handover latency and reduces the packet drop rate. The proposed scheme grants any access router the ability to self-configure its MAP selection based on the network load and average session duration.

- A novel adaptive location management protocol for vehicular networks. The proposed protocol utilizes vehicles' individual mobility projection and estimates their residual time within each access point. The protocol preemptively adjusts the vehicles' registration and location updates, depending on its movement projections.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 introduces a general overview of vehicular network design and architecture, as well as the key requirements for the next generation of wireless communications. Chapter 3 presents the general definitions and

benchmarks of mobility management protocols for vehicular networks and classify related work solutions based on the adopted network design.

Chapter 4 evaluates several movement prediction models for vehicular networks in terms of accuracy and efficiency in predicting vehicles' projections. Chapter 5 introduces two case studies of vehicular network applications based on prediction techniques showing the advantages of empowering current protocols with accurate predictors. In Chapter 6, we propose a novel predictive handover management protocol for hierarchical vehicular networks, as well as a new dynamic topology control scheme based on a hidden Markov model and an estimation of the session to mobility rate. Chapter 7 presents our predictive handover protocol by integrating multiple prediction models to estimate handoff trigger time efficiently. Chapter 8 presents a novel online machine learning-based roadside unit prediction scheme to improve mobility management schemes. Chapter 9 presents an optimized location management scheme based on adaptive estimation of vehicles' link lifetime within each network coverage. Finally, Chapter 10 concludes the thesis contributions and points out several future research directions.

Chapter 2

Vehicular Networks: Design and Architecture

2.1 Introduction

The new market demand from end-users, developers, and service providers, for high-speed wireless Internet access, have led both industry and academia to work on the development and design of the next generation of wireless networks [6]. Nowadays, people rely on mobile Internet services all the time, while driving vehicles or riding on public transportation. This leads to an enormous increase in data transmissions with the currently available networks. Vehicular networking has been a vital component for mobile technologies to support massive data load through various applications.

In general, vehicular networks application can be classified as *safety-based* applications and *non-safety* applications [226] [235] [187]. The former includes safety messaging, alerts, and warnings, which are disseminated among entities in a vehicular network (i.e., vehicles and roadside units). Safety applications infer high efficiency and very low latency in comparison to non-safety services and applications. As for non-safety services, they include traffic management services and infotainment applications. Even though traffic management services, like those of traffic monitoring, traffic-lights control, toll road service, road diversion alerts, and local area maps updates, do not require low latency demands. Still, they need to be prepared to handle the amount of data flow in the network [182]. Whereas, traffic control applications include smart navigation, finding optimal routes for intelligent vehicles, and smart traffic light scheduling and traffic flow control through data dissemination.

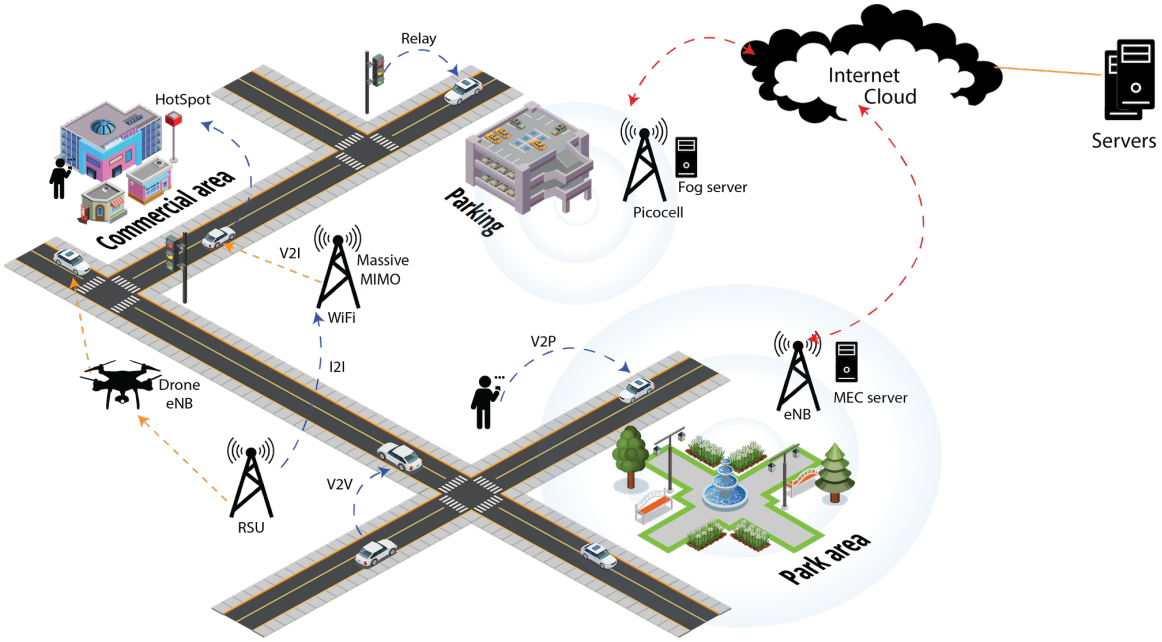


Figure 2.1: Vehicular networks design.

Furthermore, applications such as autonomous driving are anticipated to be highly accurate and effective in terms of improved data rates transmission, reduced latency, and a wider variety of communication. One promising solution is through the advancement of V2X communication in the next generation of heterogeneous vehicular networks. The next generation of wireless networks incorporates several heterogeneous communication techniques to sustain different quality requirements and constraints from service providers, applications, and users. Currently, vehicular networks rely on wireless communications such as WAVE/IEEE802.11p, Cellular Long Term Evolution (LTE), and, most recently, 5G systems to deliver different kinds of traffic management, infotainment applications, and safety services. However, enabling seamless mobility across various access networks is a fundamental issue to provide services without interruptions.

2.2 Vehicular Networks: An Overview

In Vehicular Networks, communications are usually exhibited between vehicles through Vehicle to Vehicle (V2V) communication technology, and also with infrastructure through (V2I) communication protocols, such as access routers/points, roadside units, and cells.

With the proposed 5G features, vehicles will now have more flexibility and robustness in communication and applications. Vehicle to Everything (V2X) communication was introduced to provide all types of communication between vehicles, infrastructure, network, and pedestrian [44]. Therefore, improving traffic efficiency, on road safety citeyounes2015performance, and QoS-aware infotainment services. Several existing paradigms are starting to shift their work toward enabling 5G by further enhancing the Wireless Communication Networks (WCNs) to reach the requirement of 5G specifications [61]. As current WCNs are across spectrum resources congestion with limited bandwidth, which raises a new challenge to utilize available sparse spectrum efficiently.

Over the past decade, the IEEE802.11p/DSRC [1] has been considered the *de facto* protocol for vehicular networking applications and services. Dedicated short-range communication (DSRC)/WAVE is designed for vehicular environments, as a wireless communication medium that provides fast data packet transmissions between vehicles. DSRC combines GPS positioning and wireless network to disseminate safety services between vehicles and cooperative collision avoidance (CCA), with seven channels in the MAC layer on the 5.9 GHz spectrum, one channel for control packets, and the remaining ones are data service channels.

Since the new era of wireless networks has heterogeneous architecture, to include cellular and non-cellular technologies, a standard interface is needed to facilitate communication and transition between different techniques — one of the first solutions introduced by the IETF, Proxy Mobile IPv6 [58]. The PMIPv6 is a mobility management protocol that works in a centralized manner. It includes a Local Mobility Anchor, referred to as LMA, to establish communication tunnels with the Mobility Access Gateways (MAGs), referred to as the core entity. Each MAG is responsible for users' data packets tunneled to the LMA, then to the Internet. Whereas the LMA is responsible for forwarding packets from the Internet to the end-users through their corresponding MAG. To manage the network status between LMA and MAG, Proxy Binding updates and acknowledgement packets are interchanged.

Another solution to support the heterogeneity of wireless access networks is the Media Independent Handover (MIH) framework [206]. The MIH protocol offers a standard platform interface separate from all access technologies. The Media Independent Handover Function (MIH), known as IEEE802.21, aims to provide interworking between IEEE 802 systems and other wireless access technologies (e.g., IEEE 802.11p and cellular networks). Three primary services are defined by the MIH: Event Services (ES), Control Systems (CS) and Information Services (IS). The event service tracks the changes in the link characteristics and reports them, while the control system passes movement and connectivity-related commands. As for the information service, it eases the handover process by offering a

Table 2.1: Current radio access technologies for vehicular communications.

Aspect	**Wi-Fi**	**WAVE (802.11p)**	**LTE-Advanced**	**5G**
Channel spectrum	20 MHz	10 MHz	up to 640 MHz	1-2 GHz
Frequency Band	5/2.4 GHz	5.9 MHz	5.72-5.75 GHz	1.8,2.6 GHz
Coverage	Intermittent	Intermittent	Ubiquitous	Intermittent
Range	100 m	1 km	30 km	50 m
V2V support	Ad hoc	Ad hoc	D2D	C- V2X
flow rate	6-54 Mbps	3-27 Mbps	1.5/3 Gbps	>10-50 Gbps (expected)
Handoff type	Horizontal	Horizontal	Horizontal, vertical	Horizontal, vertical

mechanism for discovering possible neighboring networks within a geographic region.

Despite the fact that IEEE802.11p is deemed the *de facto* vehicular communication norm, several research studies investigated the potentials of the Long Term Evolution (LTE) to assist vehicular networks. [12] discussed such a possibility through the strengths and weaknesses of LTE in vehicular applications. A discussion on the feasibility of LTE-V2V and V2I for future 5G vehicular networks was presented in [214] and [138]. Cellular networks initiated the fifth generation of cellular networks (5G), as a new leap toward high speed, ultra-low latency network. The integration between 3GPP and non-3GPP wireless access technologies through the Evolved Packet Core was introduced in the 3GPP's Release 15 using Fast Proxy Mobile IPv6. Nonetheless, their protocol does not consider the availability of the resources before selecting the target cells, leading to handover failures and QoS degradations. Also, EPC has no disconnection process from the previous network after switching. Hence, it may lead to high overhead and cache issues, whereas the FMIPv6 and MIPv6 still suffer from additional signaling and traffic load. In Table 2.1, we present a comparison of different wireless communication technologies used in vehicular networks.

2.3 Architectures of Vehicular Networks

In this section, we classify vehicular networks architectural design into four different categories: HetNet-based, SDN-based, Fog-based, and Hybrid-based designs. In what follows,

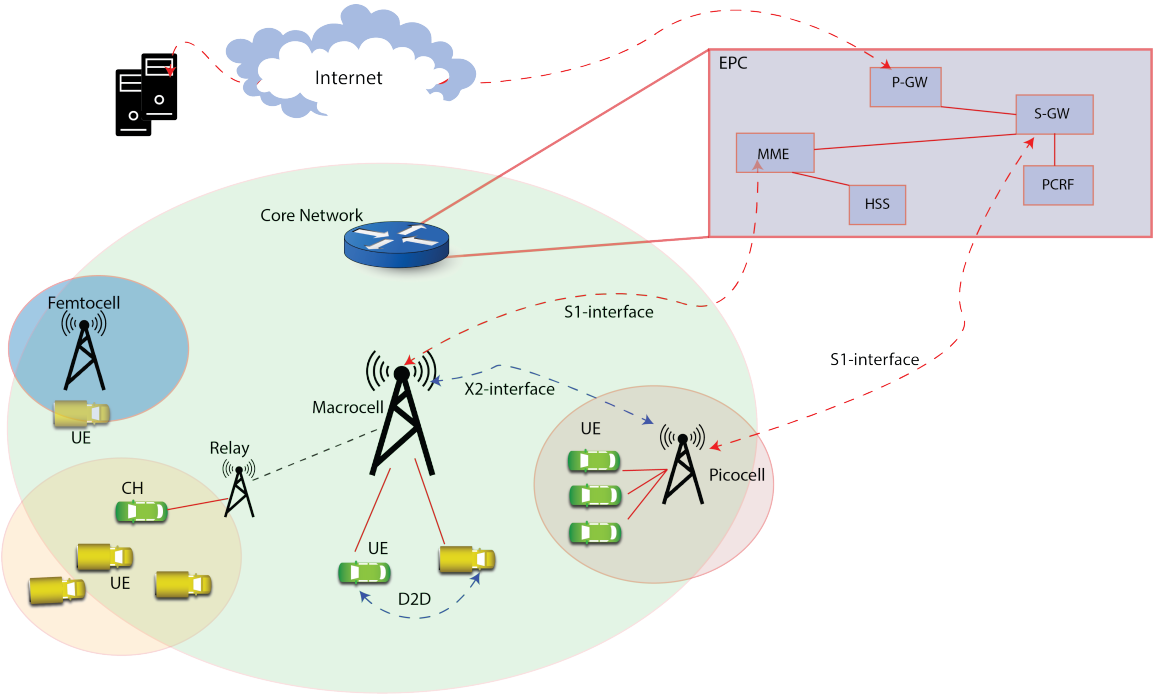


Figure 2.2: HetNets architecture.

we describe each architecture components and discuss the related studies toward the deployment and implementation of each vehicular network architecture.

2.3.1 Heterogeneous-based Architecture

With the emergence of autonomous driving in 5G-enabled vehicular networks and Intelligent Transportation Systems (ITS), new essential requirements are needed, including an ultra-low transmission delay [66]. The integration of cellular networks with the dedicated short-range communication (DSRC) standard and other wireless technologies is considered a potential solution for meeting the communication requirements of the 5G systems. As we mentioned earlier, LTE was introduced to assist vehicular applications [12] [192], because of the drawbacks in the IEEE802.11p standard in terms of low scalability and low capacity. Hence, the heterogeneity of different wireless technologies is a possible direction toward efficient mobility management and reliable communication in vehicular networks.

The 3GPP specified the new generation of wireless networks, LTE / LTE-Advanced, which provides Internet Protocol (IP) data, and signaling transmissions, with a round-trip

time below $10ms$, and transfer latency up to $100ms$ [2] . The LTE network architecture comprised of the Core Network (CN) and the Access Network (AN), in which the latter handles the radio channel resources and handover decisions through the eNBs cells or base stations (evolved NodeB). Whereas the CN comprises three primary components, namely the Serving Gateway (S-GW), the Packet Data Network Gateway (P-GW), and the Mobility Management Entity (MME). The mobility management entity monitors authentication, security, and user's location. The S-GW is used to route, forward, and connect to the Policy and Charging Rules Function (PCRF). Finally, the P-GW enables the communication between the IP and the switch. In order to accommodate the growing load of users and data transmission on the network, the 3GPP introduced the Multi-Tier Heterogeneous Networks (HetNet). Small and simpler base stations define HetNets with different transmission capacity, range, carrier frequencies, backhaul links, and communication protocols, named micro, pico, and femtocells, in decreasing order of powers. Figure 2.2 portrays a general view of HetNets architecture.

Femtocells or Home eNBs (HeNBs) are wireless short-range access stations with a limited cost and power consumption capability. HeNBs allow signal transmission at high-frequency bands in low range, thus increasing spectral efficiency. The integrating mobile femtocells into a vehicular environment can help improve scalability, signal-to-noise, plus interference (SNIR) and throughput [48]. The small cells, in Table 2.2, are designed to support higher data rates for low-speed UE and allow macrocell traffic to be offloaded to smallcells. Macrocells provide low to moderate communication for the global range so that any UE can be reached at any velocity. Nonetheless, the deployment of a massive amount of smallcells brings another challenge to resource allocations, interference mitigation, mobility management, and QoS efficiency [224]. For example, in a high-speed mobility environment, an abundant amount of transitioning will occur between small network cells along the road, which will then increase the network overhead and services disruption. Also, because of the different functionalities between macrocells and smallcells, the handover decision (i.e., when to perform handover and to which cell) should be carefully performed. That is to not only rely on network parameters such as signal strength, but also to support intelligent decision making with respect to current UE's service status, available resources, and surrounding network options.

Concerning the interference mitigation issue, the vast expansion of ubiquitous data and network densification (i.e., abundant cell deployment) have resulted in an increased spectrum interference. To enhance the spectral efficiency, a Multiple-Input-Multiple-Output (MIMO) [188] wireless technology is used, which enables various antennas to transmit more data packets with higher channel's ability simultaneously. Furthermore, the millimeter-wave transmission technique has been proposed to link users inside the vehicles and to

Table 2.2: Types of cells in HetNet-based networks.

Cell	**transmission power**	**coverage limit**	**Location**
Macro-	46 dBm	1-30 km	Outdoor
Micro-	30-37 dBm	<2 km	Outdoor
Pico-	23-30 dBm	<300 m	Outdoor-indoor
Femto-	<23 dBm	< 50 m	Indoor
Relay	30 dBm	300 m	Outdoor

overcome interference issues [180]. Additionally, the mm-wave allows for more significant bandwidth allocation and expand current channels bandwidth limitation, thereby increasing data transfer rates and capacity.

2.3.2 Software Defined-based Architecture

Software-Defined Networks paradigm is based on the concept of dividing the network architecture into control (Network) and data (forwarding functions) planes. The SDN data plane includes forwarding devices connected through wireless radio access networks (WRANs) or wired network, whereas, the control plane includes SDN controllers. Each controller is provided with an open Application Programming Interface (API) to support the programming capability of network infrastructure. A southbound API (SI) between the centralized SDN controller and forwarding nodes, where OpenFlow [141] is the commonly used protocol. A northbound API (NI) is used to communicate between the SDN controller and the application layer, which enables administrators to control the forwarding plane rules and policies remotely.

SDN architecture's primary concept is the forwarding of packets based on several data flow policies. The network configuration can be adapted in real-time, allowing the control management plane to update or add new services and devices. In addition to the network abstraction, there is parallel management of common functions such as bandwidth management, security, and control access. The Internet Engineering Task Force (IETF), Internet Research Task Force (IRTF), Open Networking Foundation (ONF), and IEEE802 Local Area Network (LAN)/Metropolitan Area Network (MAN) standards committees all worked on exploring the concept of software-defined networks and its components.

The ONF [166] works on standardizing the SDN architecture entities, including the

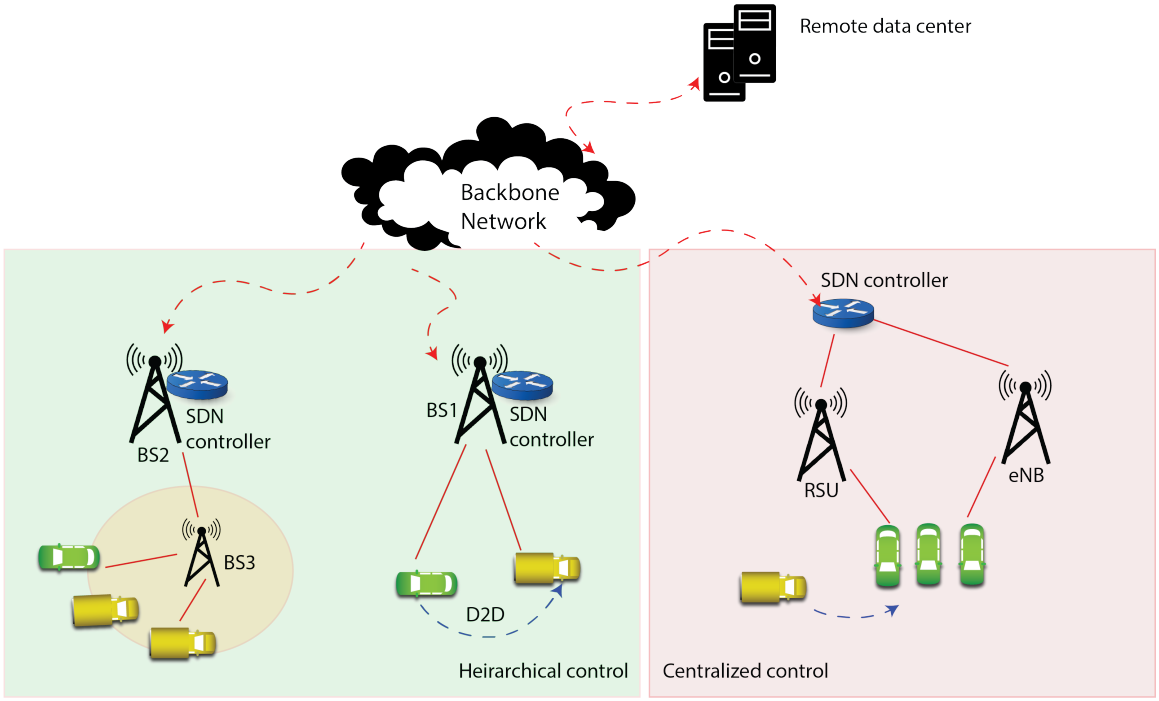


Figure 2.3: SDN-based vehicular networks architecture.

OpenFlow protocol [141]. OpenFlow defines the communication rules or policies that ensure high-performance efficiency between the control and data planes. An OpenFlow switch comprises of several tables (i.e., flow, group, and meter), in addition to a communication link to the controller. For each record, the flow table is used to map and forward packets and comprised of counters and actions. For example, with each data packet arriving at the switch, it will either use a similar record action or forward that packet to the controller. The flow record includes a match field, priority, counters, instructions, time-outs, cookie, and flags [141]. Upon receiving the packet, the controller decides the route in which that packet can follow and maintaining the switch flow-table. Multiple flow-tables may be assigned to an OF-switch while the meter-table triggers the performance operations of the data flow. An overview of OpenFlow can be found in [131] [166].

The deployment and integration of new services and protocols are now possible through the simplified networking design and implementation in software-defined networks while centralizing and controlling protocols without accessing individual network hardware equipment, as shown in Figure 2.3. Therefore, enabling centralized management and control of networking devices with common APIs. Many research studies worked on standardizing the design of wireless SDN with 5G networks such as SoftAir [7], OpenRoads [231], Cloud-

Table 2.3: Vehicular SDN-based approaches.

Related Work	**Features**	**Motivation**
Zheng et *al.* [249]	Multi-layer cloud-RAN	Networks Heterogeneity
He et *al.* [82]	Adds vehicle to cloud communication	Networks Heterogeneity
Atwal et *al.* [14]	Distributed control plane (Cloud-based)	Guarantees QoS
Duan et *al.* [56] [57]	Introduced a dual cluster head selection method	Supports increasing data traffic
Zhang et *al.* [51]	Hierarchical control units	Improved network performance during loss of connection scenario

MAC [213], and many more [209]. In what follows, we discuss some of the architectures proposed in the literature that have worked on the design of SDN with 5G networks; specifically, those related to vehicular network communication and applications.

The implementation of the Software-Defined Networks (SDN) concept within the next generation of vehicular networks can be of great help in elevating several issues, including mobility management and improving current wireless networks (e.g., WiFi, Cellular, and vehicular networks) [57]. SDN was initially tested and designed for wired network environments, such as campus networks, with high-speed switches. Because SDN architecture is broadly general, we can easily integrate it into wireless networks.

An SDN-based heterogeneous vehicular network, named SDVN, proposed in [82], which aims to narrow the barrier between the application's requirement and vehicular network's limitation. This is done through the integration of vehicle-to(-vehicle, -infrastructure, and -cloud) communications [82]. In order to overcome the SDN management overhead, they added a trajectory-based prediction policy to update vehicle status. Several vehicular application scenarios were studied to demonstrate the efficiency of such architecture using traffic based simulation. In their model, the authors [82] divided the data plane into stationery data plane (such as base stations and cells), and mobile data plane. As for the control plane, it monitors the state of every switch on vehicles and RSUs. Whereas, the vehicles' location, speed, and the overall state of the network define the switch status. Moreover, the standard OpenFlow is used as a communication interface between the planes.

Another work in [14] proposed an SDN-based architecture that takes advantage of the cloud to improve vehicular network mobility management issues. They identified several challenges in SDN-based vehicular networks, including the lack of a straight-forward solution of SDN deployment in vehicular networks. Their model introduces a cloud-based global controller for seamless connectivity and mobility through the use of local controllers. The local controller will enable continuous communication sessions even when global controllers fail or unavailable. This is done by using a logically distributed control plane, in which the control functionality of the central cloud controller is outsourced to local controllers on each on-board units. Two QoS applications are implemented to illustrate the mobility management efficiency of the proposed model.

An SDN-based vehicular network proposed in [56] [57], to handle the high data traffic and improve heterogeneous network management, by introducing a dual cluster head structure and adaptive beamforming coverage. They argue that vehicular HetNet is facing several challenges, such as network densification, due to high data rates and complexity of dynamic hardware adaptation. The cluster head vehicle is in charge of forwarding traffic from vehicles and communications to the base station, therefore reducing the network signaling overhead. Different vehicle groups can be identified by the transmission angle and Received signal strength (RSS). The performance of the proposed scheme has been evaluated using MATLAB simulation in terms of bit error rate, SNR, and throughput rate. The authors in [51] proposed a hierarchical software-defined based strategy for vehicular networks, HSDVN, which focused on path request and reply strategy between vehicles and SDN-enabled RSUs. However, they did not address the issue of frequent handover.

An early study on the cloud-based software-defined heterogeneous vehicular network (SERVICE) proposed in [249]. In which, they present three levels of cloud resources: Micro cloud, local cloud, and remote cloud, each has a different characteristic. The SDN design is divided into a network, control, and application layer. The network layer, which corresponds to the SDN data plane, includes communications, computing, and storage resources. The control layer, located in the middle as a service proxy that provides control functions and translates the requirements of users.

Due to the vehicles' high mobility, the centralized structure of the SDN controller will affect the performance of real-time communications [51]. A work by [249] proposed a hierarchical control layer to reduce the number of handovers, by defining a primary and secondary controller. The latter serve one area that includes macrocell and smallcell. They discuss four kinds of handover, Intra-/Inter Service Area and Intra-/Inter- System. The former happens when vehicles move from the same service area or to another service area without leaving the wireless access system, while the latter occurs when vehicles move between different systems in the same service area or different service areas. In Table 2.3,

we show a comparison between different vehicular SDN-based network architecture.

The IP addresses have only local significance in SDN-based networks, and the controller specifies the policies for forwarding each flow based on OpenFlow protocol [122]. Because of the SDN framework design, it would be simple to customize the network management functions, where operators can effectively program the control plane. As mentioned earlier, the OpenFlow protocol was initially created for fixed networks and did not support mobile networks' specific functions, such as configuration and monitoring mobile network cells. Therefore, to implement the SDN concept on wireless base stations, requires the support of OpenFlow protocol, to enable communication with the controller and benefit from SDN properties.

Despite the advantages of SDN, several issues might arise with the implementation of SDN-based Vehicular networks, especially with the 5G heterogeneous type networks [137]. This includes the deployment of SDN controllers, centralized controller [13] (i.e., a single point of failure [160]), scalability in dense areas in real-time applications, and vulnerability to many security attacks. A distributed mobility management based on SDN for 5G networks, named (S-DMM), presented in [160], in which all components related to mobility are independent of various techniques. Their results indicate more scalability compared to well known mobility management solutions. Moreover, their work showed lower mobility costs while maintaining similar handover latency and delay compared to other solutions. Nonetheless, seamless handover between multiple domains is still an open issue and the impact of the enormous computational complexity of mobility management on the centralized control plane.

2.3.3 Fog Computing-based Architecture

The rapid increase of large scale cloud-based applications that provide several services, from infrastructure to software as a service, has introduced a new challenge to cloud computing systems. For example, connected vehicles and content delivery applications require constant communication with cloud servers [196]. Bonomi et *al.* [28] first introduced fog computing in 2012, which grants storage, data, and services to end-users in a virtualized system. The fog cannot operate as a standalone mode and requires the cloud, since its main objective is to offer cloud computing services in a real-time strategy, by providing data migration from cloud centers to the network edge. Therefore, saving bandwidth resources, reduce energy consumption, and shorten latency. Many services can take advantage of such architecture to efficiently deliver information to clients such as location-aware services.

Fog Computing (FC) is a paradigm proposed as an expansion to the cloud paradigm [198].

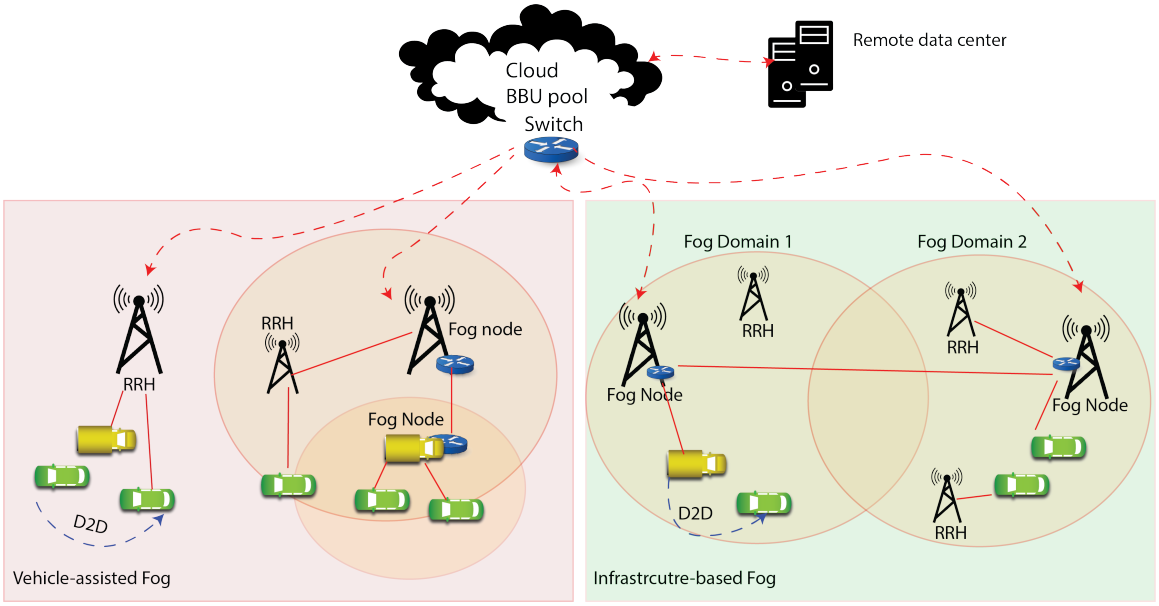


Figure 2.4: Vehicular Fog-computing architecture.

The fog enables functions to be located at the network edge closer to end-users. It addresses the drawbacks in cloud computing for latency-critical applications, such as connected vehicles, and emergency alerts, and data content delivery. Moreover, fog reduces service latency, improve Quality of Service (QoS), and user experience [28]. The fog-based access points (F-APs) are responsible for hosting services at the network edge or on smart devices.

Fog is usually introduced in macrocell base stations and WiFi access points as fog servers. When users or vehicles move between different fog servers more frequently, it will lead to service disruptions and degradation in the user experience. The fog server supports mobility by using routing and addressing protocol, named Locator/ID separation protocol (LISP), that is developed by Cisco Systems. Therefore, allowing Fog servers to communicate with end-devices. Several solutions on Fog-based Radio Access Network (FRAN) are introduced in [178] [121]. Since our focus is merely on vehicular networks enabled solutions for 5G mobility management, we only look into the solutions introduced in fog-computing that involves the design of vehicles networks.

Vehicular Fog-based Networks (VFN)

Recently, the fast rise in the number of vehicles in public transport infrastructure brought heavy road congestion and presented more slow-moving vehicle conditions. In addition to

the increased amount of parked cars, particularity in down-town regions making vehicles a valuable resource to the design of networks. Henceforth, the utilization of vehicles as infrastructures for communication and computational resources comes as an added value solution to reduce the cost and time for vehicular applications [90].

VFN employs end-vehicles clients or edge-vehicles to carry out some communication and computation functions, as seen in Figure 2.4. One of the main advantages of VFN is its proximity to clients, exploiting the best options to enable vehicles to collaborate to deliver services. The multi-tier architecture of the VFN is illustrated in Figure 2.4, compared to the original model of the fog-computing paradigm, VFN uses vehicles as part of the fog server layer. Specifically, VFN may take advantage of slow-moving vehicles and parked vehicles to either communicate services to other vehicles or aggregate computation resources from remote clouds to vehicles. Palattella et *al.* [90] studied the capability of utilizing vehicles as part of the fog in terms of communication link quality to evaluate how reliable communication is between vehicles.

Palattella et *al.* [90] implemented a vehicle fog-computing system using vehicles as infrastructures to expand network assets by evaluating the vehicle's velocity and movement status. The authors added four scenarios in which vehicles can be used as infrastructure. They defined two types of vehicles moving and parked vehicles that can carry service and applications. By understanding the connectivity relationship between vehicles, one can better assess the usability of vehicles as infrastructure. A capability analysis was done on real mobility traces from both Shanghai and Beijing in urban scenarios, where they found that more than 80% of moving vehicles have relatively low-speed. In contrast, parked vehicles tend to be significant in numbers during specific times in a day. However, authors only considered one type of fog nodes (i.e., vehicles), and QoS was not considered as a factor to latency. In the same context, an architecture for vehicular fog computing was presented in [92] along with vehicular applications use cases and security problems. Considering most vehicular applications require on-the-fly knowledge and information for traffic control and safety applications. Their framework includes three primary components, cloud servers as the cloud layer, vehicles in the data layer, and infrastructure in the fog layer. By integrating fog into vehicular networks, cloud servers will have the ability to migrate some of its functions closer to the edge for real-time efficiency, storage, and communication. Several challenges can be present along with the integration of fog-computing and vehicular environments, including the rapid changes in vehicles' mobility, making it harder to predict where and when the fog should be deployed or used. Moreover, efficiently utilizing vehicles power supply will also affect the overall system, since rechargeable embedded batteries in vehicles could help in performing vehicular fog computing tasks even when vehicles are parked.

An intelligent vehicular network based on regional fog architecture is proposed in [244] to accommodate big data in smart cities. The authors introduced two primary levels; the fog level is responsible for the cloud servers, local fog servers, and coordinator server, and the edge level includes vehicles communication through WAVE or cellular networks, and the Internet of Things applications. Several scenarios were considered in their work, including handover between fog servers to support mobility, distributed computation, multipath file downloading, and multi-source data acquisition. Furthermore, the authors included a hierarchical resource management model in which the energy-aware model is introduced among similar fogs networks and QoS-aware management among different fogs, to optimize the efficiency of their architecture.

2.3.4 Hybrid-based Architecture

Current development in 5G-enabled vehicular networks seeking to guarantee users and application requirements has encouraged many researchers to design efficient system architecture for vehicular networks [67]. The integration of fog computing in software-defined networks aims to capture the benefits of both paradigms to enhance the overall systems, networking, and services.

Ge et *al.* [67] proposed a 5G-enabled vehicular network that integrates fog, cloud, and SDN paradigms. Three main planes are introduced, the control plane derives control instructions, and the data plane gathers data, while the application plane generates rules. Adding the fog cell structure to the network model reduces the handover frequency problem in typical SDN networks. The design is composed of SDN controllers, Cloud centers, Roadside Unit Centers, regular roadside units, fog clusters, and vehicles. Vehicles positioning information is gathered through onboard GPS sensors and communication between vehicles, and access routers/points or vehicles are provided, with a fronthaul link between roadside units and roadside center. Each fog cluster is composed of an area of vehicles and roadside units, including one vehicle acting as a gateway link. When the gateway vehicle leaves the transmission range of the RSU, a handover process is established to delegate another vehicle to be the gateway. Issues may arise with such structure, as proper selection of the gateway vehicles must be done and also the optimal size of a fog cluster. The transmission delay was reported to be less than $1ms$ when the transmission distance was $300m$ [67] .

Lee et *al.* [125] define the fog as a collection of RSUs and base stations controlled by an RSU center communicating with the SDN controller. Deployment is an issue in this case, whether it is taking into account the mobility and density of vehicular areas and traffic [45].

Their architecture operates in the hybrid control mode, in which the system control is shared by the controller and Cellular base stations and SDN Roadside unit controllers. The SDN controllers only send abstract policy, and the RSUs specify the rules' behavior according to local information. The benefit of using Fog controllers is to assist in services migration to multiple base stations along the road and updating data forwarding rules and services hosting. However, their work requires overhead communication to send back data information to the SDN controllers.

A hierarchical 5G VANET architecture proposed in [111] by integrating SDN and Cloud-RAN to efficiently allocate resources, including fog computing framework at the edge, to avoid frequent handover. The topology structure of their architecture comprised of Fog-Zone controllers, Fog-Cluster-heads, Fog-vehicles, Fog-BBU-controllers, SDN controller, and optical transmission networks. Their hierarchical architecture reduces the overhead on the centralized controller. In their simulation, the controllers' transmission delay and control overhead is evaluated and have shown minimal results in different vehicles' densities. However, integration issues with various technologies is a problem.

2.4 5G and Beyond

Over the last decade, the rapid evolution of wireless communication technologies has been witnessed through industries and research communities, from 4G and LTE-A system to high-speed Wireless LANs. The next generation of vehicular networks is expected to be integrated with cellular networks and various wireless technologies, to provide a smarter and safer intelligent transportation systems [151]. Furthermore, the demand for real-time applications and services requires more bandwidth and lower latency to meet the desired Quality of Service [6]. All together, lead for a new upgrade to the current communication technologies. Hence, 5G must provide adequate and reliable communication between heterogeneous networks, specifically in ultra-dense networks.

The METIS project [61] has laid down the building foundation of the beyond 2020 5G mobile and wireless system and specified the technical needs to foreseen the 5G requirements. Many other research groups started identifying the 5G vision, including Huawei [94], 5GNOW [89], Nokia, 5G Forum, and more [5]. In what follows, we identify the main requirements to enable 5G in vehicular networks, the currently available technologies, and how they fit in the next generation of vehicular networks.

2.4.1 5G Requirements

The next generation of wireless networks is envisioned to sustain enormous services, devices, and applications, including connected vehicles, the Internet of Things (IoT), and smart city services [6]. According to Intel, the next generation of wireless networks will deliver reactive, smart, and connected devices through efficient and reliable communication, massive Machine-to-Machine (M2M) connectivity, and improved mobile broadband. In what follows, we discuss the requirements of 5G systems.

High data rates: The 5G networks are anticipated to accommodate about 100 times data rates over current 4G networks [61]. Many techniques, such as millimeter-wave communication, heterogeneous networks, device-to-device communication, and massive MIMO, can be used to handle such performance. MIMO utilizes various access points to migrate more information, resulting in a higher flow rate simultaneously. Currently, all 802.11n wireless network standards support MIMO, and the evolved Massive MIMO technology was later introduced to provide superior energy and spectral efficiency.

High Scalability: 5G networks are expected to accommodate up to 100x devices. Hence, an increase in signaling and data transmission, requiring enough frequency spectrum resources. Although current mobile communication systems are dedicated to providing consistent mobile broadband experience, yet many users today do not have a good quality of experience when surrounded by crowds. The objective is to increase the signaling capability and data transmission, which may require enough frequency spectrum resources to accommodate the growth in network size. Hence, providing fast connections and services to satisfy user experience at any time.

Ultra-low latency: The expected latency of 5G systems is to be 2 to 5 milliseconds. The current LTE network round-trip latency is around 15 milliseconds, and the DSRC round-trip latency is approximately 10 milliseconds. Several solutions can help in providing such latency through the use of D2D communication, Software-defined networks, and cloud RAN. Surely, in vehicular networks, enabling ultra-low latency will increase the efficiency of different applications. [113] presented a study of vehicular safety applications' requirements, including speed warning, cooperative forward collision warning, road condition warning, and more, which requires very low latencies.

Reduced energy consumption and increased energy efficiency: To create a fully connected society, millions of devices must be connected, from sensors to actuators, in which, their main concern is low energy consumption and low cost. Some solutions include RF energy harvesting and environmental energy sources [6]. 5G systems are envisioned to reach ten times longer battery life. Although several previous works on energy efficiency

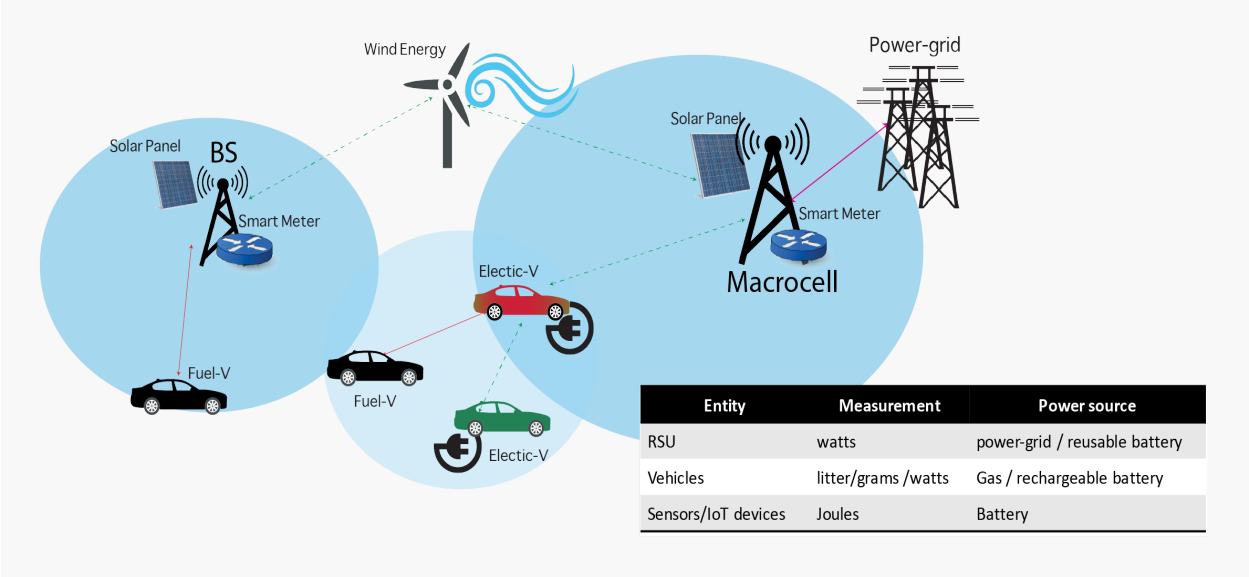


Figure 2.5: Smart energy-aware vehicular networks

have been explored, yet it was mostly on legacy wireless networks [191]. Therefore, new studies on 5G networks can be extended from previous concepts. Energy efficiency-aware cells that distribute the load on antenna's and automatically switching off when not needed. Backhaul systems are moving toward being self-organized and self-configured to reduce manual interference to reach energy efficiency.

2.5 Green Vehicular Networks

The growing demand for energy resources and the impact of energy consumption on the environment and reduction in available fossil fuel are the base of an imminent energy crisis in the next decade [88]. As current energy resources becoming scarce and expensive, the impact of their use on our climate is becoming more evident over the past and present years, which extends beyond just an increase in temperature, rising sea levels, and diseases. In reality, we cannot desert any existing power source with the click of a button. There must be a clear and practical plan to adjust the current consumption and work on the quick production of green, modern, and renewable sources of energy as soon as possible. With this in mind, vehicles and radio towers are the main building blocks of vehicular networks, which are considered some of the highest consuming entities of energy that have a profound impact on the environment. In addition, the current increase in resource-hungry applications and

services in vehicular networks needs to be reconfigured and optimized to meet the new vision of green energy transformation, especially with the existence of resource-poor mobile entities [243]. However, with the increasing number of wireless communication technologies, smart devices, and sensors, the design of such an energy-enabled wireless network is not trivial [80]. Therefore, an efficient and reliable self-powered system is crucial for the next generation of intelligent vehicular networks. A broad overview of a smart energy-aware intelligent vehicular network example is presented in Figure 2.5 along with each network entity power source and measurement unit.

2.5.1 Infrastructure Design

The current standard technique to resolve the challenge of the vehicular environment's rapid mobility is to increase the transmission power of all base stations to reach guaranteed high quality of service and user satisfaction [80]. However, the expansion of the RSUs communication range comes with the cost of smaller data rates, increased packet collision, and high delivery rates. Furthermore, with the increased amount of data traffic load, service providers worked on this issue by using smallcells [242], to cover more areas. This also comes at the cost of more power supply to run those smallcells and heavy burden on the power grid.

To overcome the expensive deployment of grid-powered base stations, industries are now looking at battery-enabled roadside units infrastructure (B-RSUs), which are capable of reducing energy consumptions and elongating their duration. It is noteworthy that some highways and rural areas do not have the convenience of grid system access, making them applicable for battery-oriented roadside infrastructure [228]. Several telecommunication companies have designed green energy-powered base stations in rural areas to compensate for the lack of power grid access. However, with the existence of such energy-limited antennas, a new challenge is facing the design of heterogeneous vehicular networks from the unbalanced battery-life and high sustainable energy demand to transmission power optimization requirements. For instance, during peak hours and in dense areas, data traffic demands are high and renewable-enabled RSUs can not provide enough energy. The major challenge is to minimize the energy consumption while maintaining reasonable data transmission rates. Several solutions could be applied, such as adaptive scheduling of the RSU's traffic and optimizing communication singles with an intelligent model to reduce energy consumption [242].

Table 2.4: Types of electric vehicles.

Type	Features	Limitation
Battery (BEV)	500 km, fully electrical, 30-45 minute for 80% charge	Still requires grid power and cost 700$ - 1000$ to recharge
Plug-in Hybrid (PHEV)	Rechargeable batteries, cheaper than traditional vehicles	Only 20-80km on electric motor, gasoline engine
Hybrid (HEV)	Regenerative braking, cheaper than other EVs, no charging constraint	Gasoline engine and fuel tank
Fuel-cell (FCEV)	Hydrogen and Oxygen electricity , Water-only emission	Expensive, Still in development phase

2.5.2 Hybrid Vehicles

Recently, automotive industries are working toward the design and deployment of environment friendly electric vehicles (EVs) that will be available to everyone with low cost and high battery efficiency. The common assumption of the vehicles' substantial storage, computing, and power resources might not be valid in the next decade since people are shifting more and more toward gas-free vehicles (i.e., Electrical Vehicles). Thus, making them now prone to power constraint and requires efficient and optimized management.

Several types of electric vehicles are now introduced by the automotive industry, including Battery, Fuel-cell, Hybrid, Plug-in Hybrid electric vehicles [227] in effort to reduce emissions and fuel displacements. In Table 2.4, we compare the different types of electric vehicles existing up to date. Clearly, with the existence of such a variety of models, managing their energy consumption and provisioning is not a straight forward solution. Many studies investigated the energy's optimization problem in electrical vehicle [132] and identified the possible solution directions [250]. The related works optimization strategy targets the charge sustaining model, battery management, and more, which utilize deterministic and fuzzy logic models to obtain optimal results. A study on the influence of driver pattern behavior on the electric car's energy consumption level [34] showed that the combination of vehicle's speed and acceleration have small correlation because of the regenerated braking energy. Furthermore, many survey articles related to managing hybrid vehicles' energy consumption are discussed in the literature [103] and can be grouped according to their management strategy [189]. To this end, conventional mobility management solutions need to be adjusted and adapted to cope with the changes in vehicular environment.

2.6 Summary

The next generation of wireless technologies are promising solutions for vehicular networks, in which vehicles will have access to emergency services and high bandwidth-intensive applications in very low latencies such as video streaming and real-time traffic conditions. However, in the 5G-enabled environment, vehicular networks face several significant problems affecting current architecture designs such as mobility management, back-haul networking, air interface, and traffic safety [234]. All the described vehicular networks' architecture and communication standards in the previous section require proper mobility management for vehicles.

Chapter 3

Mobility Management: Benchmarks, Definitions, and Components

3.1 Introduction

Mobility management is comprised of two main parts, *handover* and *location* management protocols. The latter is mainly concerned with the mobile terminal's (i.e., vehicle's) location updates and paging techniques, and the former (i.e., handoff management) deals with sustaining an ongoing vehicle's connection while switching between different points of access. As vehicles travel across cells or local radio access boundaries, the network must be able to locate vehicles and automatically route/forward data packets to the new vehicles' location. A seamless handover process with very low handover latency that provides quality of service from source to target access networks guarantees effective mobility management support. Therefore, mobility management is once again getting significant exposure from both academia and industry in the next generation of wireless networks.

3.2 Mobility Management for Vehicular Networks

Mobility management enables the user to connect and switch between different foreign networks on the go while maintaining its connectivity with its home network or any other network. Since VANETs architecture includes V2V and V2I communications, different mobility management schemes need to be considered to integrate current solutions with the rapidly changing topology of VANET. In vehicular networks, two types of transitions

may occur for vehicles: intradomain, and interdomain transition. Intradomain transition refers to moving between different access routers/points of the same network (i.e., homogeneous networks). Whereas, interdomain transition refers to the vehicle movement between different networks technologies or service providers (i.e., heterogeneous networks). According to the transition state, different mobility management protocols can be deployed. In general, mobility management consists of two main building blocks: location management and handoff management.

Location management: involves two phases, which enables the network to track the vehicles' current AP for data forwarding. The first phase is location registration (or update), where the vehicle periodically notifies the access point of its presence, thus enabling the access router and HA to keep track of the current point of attachment. The second phase is the process of data forwarding and routing packets to the vehicles current point of attachment using pre-existing roaming cache in the network. Due to the high mobility and topology changes in VANET, traditional location management solutions will amplify the bandwidth and cause additional handoffs to vehicles trying to connect to location servers. Many challenges face the design of location management in VANETs, including scalability issues, routing, security, and location update delays, due to the dynamic and high mobility nature of vehicular network entities.

Handoff (or handover) management: is the process of maintaining a mobile node connection while it switches between two points of access. The handover method comprised of three phases, agent discovery, HA registration, and data flow control. Handoff management faces several challenges before it can be considered suitable for vehicular networks. Vehicles are unable to receive any data packets while establishing connections with the new foreign network, and the registration of the new CoA with the vehicles' HA. Therefore, having high latency and regular disruptions in IP services every time the vehicle changes its network attachment. Some of the main issues that contribute to this poor performance are the following; i) HA communication cost, where the vehicles' home could be located far away and requires long-distance communication and resources allocation; ii) access points selections, choosing the best suitable access point, network or technology to initiate the registration process with the vehicles' HA might lead to performance degradation in case of incorrect AP assignment or failure; iii) handover overhead is another issue that contributes to poor network connection during handoff with the increase in the number of vehicles in the network. In the following, we present several solutions toward addressing mobility management issues over vehicular networks.

Further classification of the mobility management solutions can be defined according

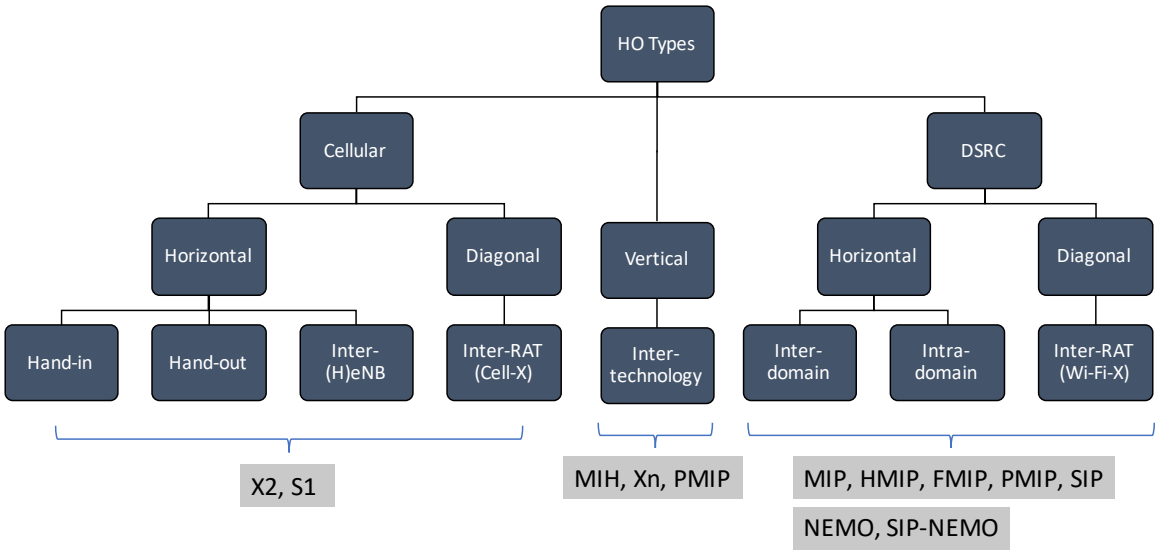


Figure 3.1: Handover management taxonomy.

to the involvement of the vehicles' MN in the process of location and handoff management. When the vehicles' MN takes part of the signaling for the handoff decisions from and to an access router, it is called *host-based* mobility management. Whereas, when the access routers are the sole entity responsible of the location and handoff process, it is called *network-based* mobility management. Another way to view the mobility management in VANETs is by first, identifying whether or not vehicles are being managed individually (Terminal Mobility) or as a network group (Network Mobility). Then, we may further explore the solutions from different layers of the TCP/IP protocol stack reference model. A general taxonomy of different handover management solutions is presented in Figure 3.1.

In what follows, we focused on two kinds of mobility management solutions, a host-based hierarchical mobile IP and various predictive solutions that would enhance the performance of mobility management. First, we describe the general structure of the hierarchical and predictive mobile IP protocols and identify the current related works toward enhancing the mobile IP handover protocol for VANETs. Then, we give a summary and comparison between all discussed solutions and their limitations.

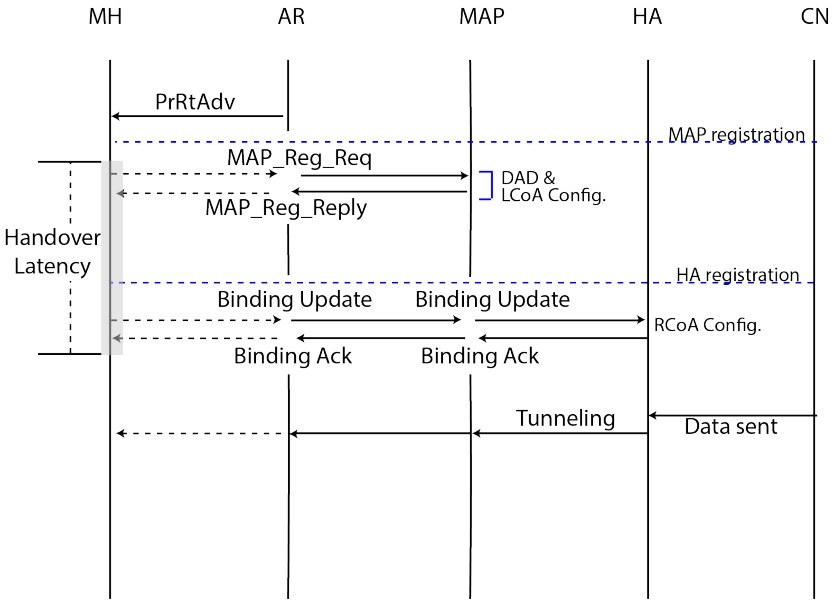


Figure 3.2: HMIP handover protocol.

3.2.1 Hierarchical Mobile IP

Soliman et al. first introduced the hierarchical MIP [21] to reduce the cost of HA registration by using Mobility Anchor Points (MAP) that act as a Regional Care-of-Address (RCoA) for any mobile host in its area. This reduces the registration cost with long distance HA. Each MAP region/domain maintains several Access Points (APs), which act as the Local Care-of-Address (LCoA) for any Mobile Host (MH) in their region. In case a MH moves between two different MAPs, a MAP and HA binding update process is required (i.e.: inter-mobility handoff). Otherwise, MH's movements within the same domain area are only required to initiate a binding update process with the current domain MAP (i.e. intra-mobility handoff). The inter-mobility handoff process is illustrated in Figure 3.2, which is divided into two phases of MAP discovery and registration, and HA registration and update.

In recent studies on the enhancement of hierarchical MIP, researchers mainly addressed the problem of inter-mobility handover issues. This is because the handover latency is increased during the conversion between different MAP domains. In the following, we review the recent studies that have addressed the problem of inter-mobility handover, including MAP selection optimization, MAP overhead reduction, and cost.

Table 3.1: Hierarchical handover approaches.

Articles	**Objective**	**Contribution**	**Advantages**	**Drawbacks**
Soliman et *al.* [21]	Reduce HA communication	Introduces MAPs	Reduces registration costs	Inconsistency due to MAP transitions
Teng et *al.* [207]	Reduce MAP transition costs	Adds inter-MAP communication	Improves handover consistency	MAPs may still be in distant networks
Nath et *al.* [157]	Reduce MAP transition costs	Adds AP tables in MAPs, uses vehicle movement for management	Smoother MAP transitions	Tables have low scalability, movement is unreliable
Kawano et *al.* [109]	Improve MAP selection	Adds a hierarchical tree structure to MAPs	Improves MAP coordination, balancing MAP loads	Increases overhead to coordinate MAPs
You et *al.* [205]	Improve MAP selection	Vehicles connect to a primary and secondary MAP	Improves reliability and consistency of handover performance	Increases overhead and network loads
Lee et *al.* [126]	Reduce network load	Adds IP paging extension	Reduces unnecessary MAP and vehicle communication	Increases latency delay when MAP needs to forward IP packets to vehicle
Mirzamany et *al.* [149]	Reduce MAP loads	Two-layer MAPs for intra-domain handovers	Reduces inter-domain latency and MAP loads	Increases inter-domain transition overhead and complexity

Mobility Anchor Point Transition Costs

The costly procedure of a vehicle switching between MAPs may create inconsistency in handover performance. When a switch occurs, the handover procedure becomes more expensive than the standard mobile IP handover, due to latency and overhead costs in contacting the HA and setting up the new MAP. To address this problem, H. Teng et *al.* [207] add a context message protocol between anchor points. The context messages allow MAPs to communicate information directly to each other during a MAP transition. This reduces the required registration exchange between the vehicle and the MAP, and additionally lowers the registration delay, since one MAP can directly begin forwarding IP packets to the vehicle's new MAP. One problem with this approach is the additional network load produced by the context messages.

Another approach that uses context messages to reduce transition costs is proposed by P. Nath et al [157]. In their approach, the MAPs maintain tables of all nearby APs and their related MAPs. The vehicle's current MAP uses the vehicle's general movement to estimate which AP it will connect to next. The MAP will then check its tables and see whether the vehicle is about to leave for another domain. If this is the case, the MAP will send context messages to the new MAP to reduce the registration costs of MAP transitions. MAP transitions will therefore be smoother when successful; however, the problem is two-fold. First, having MAPs maintain tables on all nearby APs can be difficult to maintain or update, and is not scalable if AP density is high. Second, a vehicle's movement is unreliable, which will cause many incorrect assumptions about which AP is next.

Optimizing Mobility Anchor Point Selection

There is also the issue of a vehicle connecting to a MAP that is not optimal for performance. This could occur when a vehicle moves along the line between two MAP domains. K. Kawano et *al.* [109] introduce a multilevel HMIP approach; this adds a tree structure to MAPs that allows improved MAP coordination and better distribution of network loads. This is done to improve MAP performance, but suffers from the increased overhead required for MAP coordination. Another approach to this issue is by T. You et *al.* [205], who propose a robust HMIP where a vehicle registers with a primary and secondary MAP at the same time. If failure to connect with the primary MAP occurs, or if the vehicle quickly transitions away from that MAP, it can rapidly recover by switching to the secondary MAP. This reduces the recovery time that occurs when the wrong MAP is selected. This can be particularly helpful in vehicular networks, where frequent topological changes could result in quick MAP changes. However, registering with two MAPs incurs additional overhead

and network loads to the MAPs, which becomes largely problematic with an increase in vehicle density.

Reducing Mobility Anchor Point Overheads

When vehicle density increases, the high overhead and network loads for MAPs can put a lot of stress on the network. This issue is amplified by methods adding more overhead and MAP loads to resolve other HMIP issues. J. Lee et *al.* [126] add an IP paging extension to HMIP to reduce the amount of vehicle tracking otherwise required by the MAP. The extension has the MAP page for the vehicle's location when it receives packets that are destined for it. By having a paging system, the MAP only needs to keep close track of active vehicles. This reduces the unnecessary tracking otherwise required for vehicles that idle within a MAP's domain. One concern with this approach is the additional delay that occurs before a vehicle receives the first wave of IP packets upon becoming active. Also, in urban situations, vehicles may make sudden and frequent changes in speed. This could cause major performances issues due to the frequent switch between active and inactive.

Binding updates can then be sent according to this estimation instead of at predefined increments, thus reducing the number of packet exchanges necessary for the MAP to track a vehicle. However, large performance costs can occur if a vehicle changes location in less time than was estimated. This would also incur additional performance issues within the urban environment, where traffic may cause large variations in dwell time.

Mirzamany et *al.* [149] propose using a two-level MAP system to distribute the MAP load without affecting communications with the corresponding node. The two-level system is composed of global MAPs (GMAPs) and local MAPs (LMAPs), where GMAPs are located at the standard MAP location, and LMAPs are set up between the GMAP and APs. This allows each MAP to have a lower load, since LMAPs handle fewer APs than a standard MAP, and the GMAP must only communicate with the vehicle during LMAP transitions. Another benefit of this approach is that it reduces intra-domain handover latency, since LMAPs are even more local, and with lower MAP loads their response will be nearly instantaneous. One issue with this approach is the incurring of additional overhead upon a domain transfer, as the vehicle would then have to establish connection through an LMAP and GMAP.

Summary

An overall performance summary is provided in Table 3.2, where the values of low, medium, and high are based on the analysis found within the section. In summary, the HMIP's re-

Table 3.2: Hierarchical handover performance comparison (**H**: High, **M**: Medium, **L**: Low).

Articles	**Simulation Tools**	**Scalability**	**Handover Latency**	**MAP Transfer Cost**	**Overhead**	**Network Load**	**Consistency**
Soliman et *al.* [21]	None	H	M	L	M	M	H
Teng et *al.* [207]	NS-2	H	M	L	M	H	M
Nath et *al.* [157]	Analytical, parameter estimation	L	M	L	L	H	L
Kawano et *al.* [109]	Analytical	M	M	M	H	L	H
You et *al.* [205]	Columbia IP Micro-mobility Software [39]	M	M	L	H	H	H
Lee et *al.* [126]	NS-2	H	M	M	L	L	L
Mirzamany et *al.* [149]	NS-2	M	L	M	L	M	L

duction of HA communication improves the handover registration costs when a vehicle is within a single MAP's region. However, some HMIP aspects, such as switching between MAP regions and additional overhead, remain problematic and limit its improvement in regard to network performance. In addition, HMIP approaches do not address the performance issues of the handover's discovery process.

3.2.2 Predictive Mobile IP

The predictive handover aims to conduct the handover process in advance, instead of directly modifying the mobile IP architecture. This is achieved by predicting which AP the vehicle will connect to next, before the vehicle begins transitioning between APs. By identifying the next AP so far in advance, the handover process can easily be completed ahead of time to provide a smooth AP transition. The largest issue with the predictive approach is the unreliability of the advanced handover, which is mostly caused by poor AP prediction performance. However, predictive handovers hold great potential in vehicular networks due to the increased predictability of vehicular movement. Vehicles are forced to follow roads and signs, limiting their movement options to specific directions. Methods for handover prediction can be categorized as probability analysis, pattern-matching, and movement projection.

Probability Analysis

Probability analyses for predictive handovers most commonly use a probability modeling technique to determine the next AP. The modeling usually considers either sample statistical information or analytical derivation to determine the probability relationships between the variables and the potential APs. The method then uses these probabilities to calculate a vehicle's next most likely AP. Table 3.3 provides a summary of the literature, and Table 3.4 shows an overall performance comparison between the different approaches. These tables are based on the following analytical comparison.

One approach to probability modeling is proposed by Lassabe et *al.* [124], which uses a K^{th} Markov model for prediction. The K^{th} Markov model is used to model the probabilistic relationship between APs, based on the previous APs to which a vehicle has been connected. First, the Markov model is trained with a set of sample vehicle data to determine the probability values. After training, the final values are used to calculate the most likely next AP for the vehicle based on the previous APs with which it has connected. A vehicle's previous AP will often indicate its future AP; however, this approach suffers in

Table 3.3: Predictive handover approaches (Probability analysis).

Articles	**Variables Used**	**Math Tools Used**	**Advantages**	**Drawbacks**
Lassabe et *al.* [124]	AP series probability	K^{th} Markov Model	Effective where dominant patterns exist	Low accuracy with similar probabilities
Becvar [19]	AP relational probability	Handover frequency calculation	Improves consistency	Lower accuracy due to reduced information
Wijngaert et *al.* [218]	AP relational probability	Handover frequency calculation	Higher accuracy	Very large overhead costs
Al Masri et *al.* [139]	AP transition time + area, vehicle speed	Markov model	Reduces packet drop rate	Added complexity not justified by small benefits
Pack et *al.* [170]	AP handover frequency, dwell time	Dwell time average + handover frequency calculation	Ensures consistent handovers if dwell time is small	Added overhead and complexity, requires accurate data
Akyildiz et *al.* [8]	AP series + relational probability	Probability mass function	Improves accuracy and consistency	Limited by individual method performance
Kim et *al.* [112]	Vehicle's AP history, AP location	X^2-distance-based technique	Differentiates between individual vehicles	Physical history is unreliable
Fazio et *al.* [62]	Location measurements	Distributed Markov Chains	Reduced calculation and less misleading	Cannot differentiate similar AP probabilities
Bohlooli et *al.* [26]	Statistical	Matches vehicle's turn choices at intersections	Reduces processing costs, predicts longer segments of vehicle's path	Incorrect predictions have large costs, lack of temporary data reduces accuracy

Table 3.4: Predictive handover performance comparison (Probability analysis) (**H**: High, **M**: Medium, **L**: Low).

Articles	**Simulation Tools**	**Scalability**	**Processing Cost**	**Memory Cost**	**Overhead**	**Network Load**	**Consistency**
Lassabe et *al.* [124]	LASH [27], context databases	M	M	M	L	L	L
Kim et *al.* [112]	Relative anchor mobility model, mathematical simulation	H	H	M	L	L	M
Becvar [19]	Matlab, Manhattan mobility model, urban microcell path loss model	L	L	M	L	L	L
Wijngaert et *al.* [218]	OPNET, Har-Xia-Bertoni path loss model [81]	L	M	M	H	H	H
Al Masri et *al.* [139]	Matlab, Monte Carlo simulations	H	M	L	M	L	L
Pack et *al.* [170]	C code, micro-mobility model	L	M	H	H	M	M
Akyildiz et *al.* [8]	Analytical mobility model, mathematical simulation	M	H	H	M	L	M
Fazio et *al.* [62]	Mathematical simulation	H	L	L	L	L	L
Pazzi et *al.* [176]	NS-2, random mobility model	H	L	L	L	L	L

performance if the vehicle's AP history is misleading. Since roads restrict the direction in which a vehicle can travel, indirect routes are often necessary, which may then cause prediction errors.

In an attempt to resolve the issue of misleading vehicles' connection history, Becvar [19] reduces the size of the system being predicted. Instead of considering a vehicle's entire AP history, only the vehicle's current AP and its probability relationships to neighboring APs are used. This greatly simplifies prediction requirements by reducing the probability calculations and the AP memory a vehicle would otherwise maintain. The probability calculation for determining which AP the vehicle will connect to next becomes a simple frequency calculation of

$$P_{x,y} = \frac{H_{x,y}}{\sum_{y=1}^{N} H_{x,y}} \tag{3.1}$$

where $H_{x,y}$ is the number of handovers from AP_x to AP_y, and N is the total number of neighbors. However, one problem that arises with this approach is its inability to distinguish between individual vehicles.

Kyriakakos et *al.* [123] also choose to consider only local AP variables as done in [19], but include the previous AP to provide some distinction between vehicles. They also introduce a learning automaton to improve long-term performance by adjusting probability variables. The automaton updates the variables according to a trial-and-error method, which adds weight to neighbor AP probabilities if the prediction is correct, and removes weight from the predicted AP if it is wrong. This is done using equation 3.2 with weight w, where the vehicle transitions from AP_x to AP_y.

$$P_{x,y} = P_{x,y} \pm w(1 - P_{x,y}) \tag{3.2}$$

The prediction results are retrieved by vehicles informing the learning automaton by communicating over the network. Two learning automatons are used: one for global AP probabilities, and one that keeps track of individual vehicle path results. The goal of adding the second automaton is to learn each vehicle's path, since vehicles are more likely to follow the same path as they have previously traveled. The learning automatons show improvement to accuracy with the passage of time, but require additional overhead for communication prediction results. In addition to overhead, the second learning automaton requires a large database to maintain the probability information of every passing vehicle.

Wijngaert et *al.* [218] use a similar approach as [19], but extend the prediction to determine the next three most-likely APs. They show that predicting the next three

APs instead of only one greatly improves accuracy, partly because a vehicle will most commonly have that many realistic options. The problem with this method, however, is the large increase in overhead. To conduct an early handover with three APs requires a default of two APs to waste resources. Once APs begin reaching saturation in high-density traffic scenarios, this additional overhead will create large performance costs, reducing the benefits of handover prediction.

A separate method for using probability analysis to improve the handover is proposed by Al Masri et *al.* [139]. In their approach, they use a Markov model to probabilistically model the session activity, and then use this information to determine the best moment to conduct the handover. The authors model the session as having two potential states, off and on, and at each time step they calculate the probability of the state changing. To ensure the handover is conducted before the vehicle exits its current AP's range, the AP transition time, $t_t r$, is also considered. This is done using equation (3.3), where R_1 represents the inner radius of an AP's transition area, R_2 represents the AP's ending range, and v is the vehicle's speed. The radii R_1 and R_2 are determined by using a throughput threshold range.

$$t_{tr} = \frac{R_2 - R_1}{v} \tag{3.3}$$

If the handover is conducted when network activity is low, overall performance will be less affected. However, a session activity is very unpredictable, making this approach unreliable. In addition, there is also the risk of a vehicle changing speeds, which may result in poor handover timing that will cause more performance degradation than the standard handover.

The idea presented in [218] is expanded upon by Pack et *al.* [170], who propose a handover database used for prediction. The database records the neighbor AP to which the vehicle connects, as well as the dwell time the vehicle spends within that AP. First, the most likely APs are derived based on the handover frequency, as done in [218]. Next, the recorded dwell times, R, are used in addition to probability to determine how many APs the vehicle should predict. The final transition probability between AP_x and AP_y, $T(x, y)$, is determined by

$$T(x,y) = n \times E(R(x,y)) \times P_{x,y} \qquad E(R(x,y)) = \frac{1}{K} \times \sum_{k=1}^{K} t_{out} - t_{in} \tag{3.4}$$

where $E(R(x, y))$ is the expected dwell time, $P_{x,y}$ is the original handover probability,

Table 3.5: Predictive handover approaches (Movement projection).

Articles	**Variables Used**	**Math Tools Used**	**Advantages**	**Drawbacks**
Hernandez et *al.* [84]	vehicle's location and velocity	Kalman filter	Removes the similar probabilities issue	Low accuracy when vehicles turn
Su et *al.* [204]	Vehicle's location, velocity, and dwell time, AP transition range	Two-dimensional kinematic equation	Reduces packet loss and latency from packet forwarding	Difficult to predict sudden movement changes
Gunasekaran et *al.* [76]	Road information, vehicle's location and velocity	Distance threshold comparison	Reduces packet loss and packet overhead	Large performance costs if error occurs
Fang et *al.* [63]	Signal strength	Signal-to-movement matching	Reduces external requirements	Unreliable due to short AP ranges
Bhaskar et *al.* [22]	Signal strengths	Signal strength projection	Reduced requirements, improves timing	Unreliable due to inconsistent, noisy measurements
Sadiq et *al.* [190]	Signal strength, vehicle location and velocity	Distance square calculation	Improves reliability	Increases processing and overhead costs
Almulla et *al.* [9, 10]	Location measurements	Angle threshold calculation	Increases projection accuracy	Reduced consistency due to misleading behavior
Park et *al.* [173]	Vehicle location and movement	Kalman filter	Reduces handover packet loss and latency	Large overhead for multi-casting packets, increased costs if prediction error

and n is a scaling factor for adjusting the timer values. If a dwell time is small for the predicted AP, the algorithm then extends the number of APs for early registration to ensure a smoother connection. By using a more dynamic approach to determining the number of APs to which the vehicle can commit, additional accuracy is attained without producing too much overhead. In addition, with V2V communication and increased speeds, vehicles are much more likely to have sudden subsequent handovers.

The method proposed by Akyildiz et *al.* [8] uses both the handover history method from [19] and the AP history method from [124] in parallel. After calculating the next most likely AP using both methods, the two results are then weighed against each other based on their determined reliability. These reliabilities are calculated based on how confident each method is with the information used in its prediction. Thus, the proposed approach uses the best of the two methods (i.e., history method), depending on the given situation. Since these methods have strengths in different scenarios, an overall improvement in prediction is observed. However, this parallel approach is still limited by the maximum individual performances of the other methods, improving performance by only a small amount.

A halfway point between using trajectories and regions is proposed by Kim et *al.* [112], who predict the next AP by considering a vehicle's AP history and its general direction. At each handover, the vehicle's location and AP are recorded. When more than one measurement set is recorded, the angle θ at time t is calculated using equation (3.5), where x and y are the position measurements.

$$\theta^{-1} = tan^{-1}(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}) \tag{3.5}$$

The angle information is then compared to previous vehicles that have connected to the same set of APs. The vehicle is then assumed to connect to the same APs as the vehicle that has the closest-matching angles, achieved by using a similarity equation and threshold. One example of similarity calculation that is provided by the authors is the X^2-distance-based technique, shown in equation (3.6). This equation calculates the relative value difference between matrices a and b through comparison of all n matrix values.

$$X^2(a,b) = \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{a_i + b_i} \tag{3.6}$$

By using regional history and general direction, some improvement can be seen in prediction performance. However, this approach suffers from a similar issue as [124], where road restrictions can often cause a vehicle to move in indirect paths. A road that causes

the vehicle to move in an indirect path will disrupt the AP to angle pattern matching, resulting in prediction errors.

Fazio et al [62] propose analyzing a series of location measurements while the vehicle is within a single AP, instead of using AP history and direction analysis. They use distributed Markov chains to determine the next most probable AP based on the path the vehicle is traveling. Calculating the AP probabilities is then done using the maximum likelihood estimator using the incoming location measurements. A similar approach is proposed by Pazzi et *al.* [176], who use a hidden Markov model instead of distributed Markov chains. They then predict the AP with the highest probability, the prediction algorithm being triggered when signal strength begins to dwindle.

The methods in [62] and [176] require much less calculation than [112], since they only uses local location information and require processing only one matrix instead of a set. Additionally, there is the benefit of local location information being less misleading than long-term history analysis, because temporal data is more likely to represent a vehicle's next movement. Despite these benefits, it must still be resolved how the vehicle can differentiate between two APs of similar probabilities. For example, a T intersection can result in two vehicles sharing the same path, but with different resulting directions.

Movement Projection

Movement projection approaches use temporal data related to a vehicle's current movement to predict its future location, as opposed to deriving the most probable future movement through considering recorded statistical information. The benefit of this approach is that it does not suffer from the same issues as a statistical approach. Misleading history is not a problem because only temporal data is considered, and two options with similar probabilities can be distinguished using temporal data that differentiates between these options. We provide a summary of these approaches in Table 3.5 and an analytical performance comparison in Table 3.6.

An example of movement projection is proposed by Hernandez et *al.* [84], who use a Kalman filter to model the vehicle's location and velocity, and to determine its future AP. The Kalman filter estimates the vehicle's movement from noisy GPS measurements through a two-step process of estimation and update. This involves using an initially estimated movement x and the GPS measurement z at time increment k, which are defined by equations (3.7) and (3.8), respectively.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{3.7}$$

Table 3.6: Predictive handover performance comparison (Movement projection)(**H**: High, **M**: Medium, **L**: Low).

Articles	**Simulation Tools**	**Scalability**	**Processing Cost**	**Memory Cost**	**Overhead**	**Network Load**	**Consistency**
Hernandez et *al.* [84]	RAMON [85], mathematical simulation	H	L	L	M	L	L
Su et *al.* [204]	GloMoSim [238], free space propagation model	H	L	L	M	L	L
Gunasekaran et *al.* [76]	NS-3	L	L	M	L	L	L
Fang et *al.* [63]	Recorded movement trace, mathematical simulation	M	L	L	M	L	L
Bhaskar et *al.* [22]	OMNET++, MiXiM [116]	H	L	L	L	L	L
Sadiq et *al.* [190]	Mathematical simulation	H	M	L	M	L	M
Almulla et *al.* [9, 10]	NS-2, statistical analysis	H	M	L	M	L	M
Park et *al.* [173]	Numerical analysis	M	L	L	H	M	L

$$z_k = Hx_{k-1} + v_k \tag{3.8}$$

These values are weighed against each other by using the Kalman gain calculated with equation (3.9), where P_k is the estimated error, H is a scaling matrix, and R is the estimated covariance. This Kalman gain represents the estimated error of the initial estimation and the GPS measurement from the vehicle's actual movement. The final value movement is finally calculated using equation (3.10).

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{3.9}$$

$$x_k = x_k + K_k(z_k - Hx_k^-) \tag{3.10}$$

The method also introduces the concept of ghost agents, which are entities that are added to APs to be used after a prediction is made. These agents reserve resources and initiate an early registration procedure before the handover begins to reduce handover latency times.

Su et *al.* [204] extends the prediction introduced by [84], including a vehicle's location and speed to project a vehicle's remaining dwell time for its current connection. Predicting the dwell time D is done with equation (3.11), where $a = v\cos\theta$, $b = v\sin\theta$, r is the transmission range of the AP, and x and y represent the two-dimensional distance between the AP and the vehicle.

$$D = \frac{-(ax + by) + \sqrt{(a^2 + b^2)r^2 - (ay - bx)^2}}{a^2 + b^2} \tag{3.11}$$

Projection of the remaining dwell time is used to determine the handover's timing, thus allowing packet-forwarding to the new AP to begin without requiring a trigger from the vehicle. This reduces packet-loss that would otherwise occur from misdirected packets, and additionally reduces the latency caused by a vehicle waiting for its forwarded packets.

These projection approaches have high prediction accuracy with consistent vehicle movement, but begin to suffer great performance consequences in situations where changes in movement are common. Both [84] and [204] are unable to predict if a vehicle is about to turn or rapidly change in speed. Therefore, if one of these events occurs, a prediction error will occur almost every time. For [204], where handover timing is also predicted, an additional spike in packet-drop rate will also occur. Gunasekaran et *al.* [76] attempt to address

these issues, proposing to cross-reference the vehicle's route according to a road database. This reduces the number of possibilities for being misled; however, a vehicle that behaves differently from the expected route will suffer very large degradation in performance.

Fang et *al.* [63] propose removing the dependence on GPS measurements by deriving the vehicle's movement and the next AP through the use of temporal network measurements instead. The vehicle interprets changes in signal strength in its current AP and surrounding APs to estimate its movement. Instead of projecting the specific movement, it predicts with which AP it will most likely connect next, based on projected signal strengths. This is not to be confused with the fast handover, which also uses signal measurements to recognize an upcoming handover. The predictive method takes current signal strength changes and estimates which AP it will connect to next, whereas the fast handover waits for the AP change to begin. The problem with this approach, however, is that the signal interference of the environment can easily cause noise problems that are far worse than GPS measurements. This noise can then cause prediction errors when the method tries to project the signal changes. In addition, AP coverage ranges are relatively small, often leaving the vehicle without sufficient data to determine its next AP.

The method proposed by Bhaskar et *al.* [22] also utilizes network measurements to project the next AP. However, this is not an attempt to derive the physical movement with the signal; instead, the authors analyze the changing received signal strengths (RSS) to determine the next AP. This is calculated by sampling the AP at a rate of f_s, deriving the RSS at time t with equation (3.12).

$$RSS_t = \frac{t}{f_s} \tag{3.12}$$

Thus, it is no longer necessary to have multiple nearby APs, and reduces the calculation costs for prediction, but maintains enough information for predicting the handover timing. The result of this approach is an amplification of the noise issues also faced by [63]. Since it analyzes fewer signals than [63], disruptions in the signal will cause more regular and more disruptive prediction errors. In addition, depending on fewer signals is less reliable due to the natural inconsistency of a signal reading.

Sadiq et *al.* [190] introduce an approach for using both signal strength and movement projection to predict the next best AP and the dwell time. The movement projection for calculating the dwell time is done very similarly to equation 3.11. However, this dwell time estimate is also used to determine the best next AP by additionally considering the distance to neighboring APs. This distance estimation, d, is then weighed against each AP's faded signal-to-noise ratio f_{SNR} and current channel capacity c to produce a probability value p_v

Table 3.7: Predictive handover approaches (Pattern Matching and Hybrid).

Articles	**Combines**	**Variables**	**Math Model**	**Advantages**	**Drawbacks**
Hu et *al.* [91]	Pattern + Projection	Vehicle movement and speed	Degree of matching calculation	More consistent, unlikely to be misled by probabilities	Large processing and storage costs for comparing trajectories
Yavas *al.* [232]	Pattern + Projection	Road segments, vehicle location	Degree of matching calculation	Reduced calculation and memory costs	Less reliable without projection consideration
Jeney et *al.* [98]	Pattern + Projection	Vehicle location, dwell time, signal-to-noise ratio	Degree of matching calculation	Improves turn detection and accuracy	Large processing costs to compare movements
Bohlooli et *al.* [26]	Pattern + Statistic	Road information, vehicle turn sequences	Turn sequence matching	Reduces processing costs, predicts longer segments of vehicle's path	Incorrect predictions have large costs, lack of temporary data reduces accuracy
Paik et *al.* [171]	Projection + Statistic	Vehicle location and speed, AP location	Confidence interval and probability threshold	Increases consistency	Limited by individual performances, increased complexity
Liu et *al.* [135]	Pattern + Projection	Vehicle location and speed, vehicle AP history	Kinematic equations, AP history matching	Projection improves pattern choosing, consistency improved	Accuracy still limited by individual approaches' accuracies

for each neighboring AP. This is calculated with equation 3.13, where *alpha* is a condition weight based on the maximum values of each variable, calculated by equation 3.14.

$$p_v = \alpha \times d \times f_{SNR} \times c \times p_v \tag{3.13}$$

$$\alpha = \frac{1}{\times d \times f_{SNR} \times c \times p_v} \tag{3.14}$$

These projection approaches do not consider potential sudden changes in the vehicle's behavior. Almulla et *al.* [9, 10] aimed to resolve this issue by adding a turn detection scheme that examines a vehicle's movement angles to determine if a turn is about to occur. The scheme derives these angles through the use of four consecutive GPS measurements. Equations (3.15) and (3.16) are used to calculate the angles values, where m_1, m_2, m_3 are the first three GPS measurements and n is the vehicles current location.

$$\cos\alpha = \frac{(n - m_3) \times (m_2 - m_1)}{|n - m_3| \times |m_2 - m_1|} \tag{3.15}$$

$$\cos\beta = \frac{(n - m_3) \times (m_3 - m_2)}{|n - m_3| \times |m_3 - m_2|} \tag{3.16}$$

These angles are compared to a set threshold, and if either drops below the threshold value, a turn is assumed. This method improves recognition of a turning vehicle and thus improves the AP prediction. However, accuracy still suffers in situations where abrupt movement changes occur.

The method proposed by Park et *al.* [173] uses a similar GPS-based prediction approach as [84], but has a different protocol for handling the prediction. Once the next AP has been predicted, the vehicle uses binding update packets, similar to the fast handover, to notify the next AP and prepare the handover in advance. The binding update is sent to the HA, which then establishes a connection with the next AP and multi-casts the packets to both the old and new APs. After the vehicle connects with the next AP, a notification is sent to the HA to stop sending packets to the old AP. Packets dropped during the handover are greatly reduced using this method, and handover latency is reduced. However, this approach requires large overhead for the multi-casting, which can potentially occur for extended periods of time if the prediction occurs prematurely.

Pattern Matching and Hybrid

Pattern matching and hybrid approaches are discussed together, since pattern matching uses either probability analysis or movement projection for patterns. Summary of the discussed approaches can be found in Table 3.7, with an analytical comparison of their performances displayed in Table 3.8. Pattern matching records individual vehicle movements and matches the current vehicle's movement to the closest related recorded movement. The approach assumes the current vehicle will continue to follow the recorded related pattern. Pattern matching can often provide more insight into a vehicle's future movement, but has higher calculation costs than probability modeling approaches. One example of pattern matching can be found in Hu et *al.* [91], where partial trajectories are matched to determine a vehicle's future location. At each interval, a trajectory is calculated based on the vehicle's current movement, denoted as x. After a series of trajectories are collected into array X, they are compared to a database of previous vehicle trajectories, T; equation (3.17) is used to determine the closest matching trajectory, where D_y represents the closest matching trajectory in sequence T_y to array X, and N is the total number of trajectories in the database.

$$P(T_x|X) = \frac{\frac{1}{D_x}}{\sum_{n=1}^{N}(\frac{1}{D_n})} \tag{3.17}$$

A problem with this approach, however, is the high cost of comparing a vehicle's trajectories to an entire database when there are time restrictions for providing an output. In addition, the large variety of potential movements of a given vehicle can make it difficult to match trajectories to a specific pattern.

Yavas et *al.* [232] propose a similar movement matching method to [91], but use vehicle regional movements instead of trajectories. Their method divides the roads into small, discrete segments, then compares a vehicle's recent segment history to previous vehicle segment histories. The approach then assumes the vehicle will next move to the road segments of the closest matching segment path. By dividing the road into discrete segments instead of directly considering a vehicle's continuous movement, the number of possible observations is greatly reduced. This reduces the calculation cost for comparing path history as there are fewer potential combinations and simpler math is required for comparing discrete values. However, one setback is that the method does not consider vehicle speed or acceleration, which can often be revealing in regard to a vehicle's intentions.

Jeney et *al.* [98] compare a vehicle's GPS measurements to a database of previous vehicle locations to predict the next AP. A series of GPS measurements recorded by the

Table 3.8: Pattern matching and hybrid performance comparison (**H**: High, **M**: Medium, **L**: Low).

Articles	**Simulation Tools**	**Scalability**	**Processing Cost**	**Memory Cost**	**Overhead**	**Network Load**	**Consistency**
Hu et *al.* [91]	Kohonen self-organizing neural network [77]	M	H	H	L	M	M
Yavas et *al.* [232]	Mathematical simulation	L	M	L	L	L	L
Jeney et *al.* [98]	Native IPv6 UMTS-WLAN testbed	L	H	H	M	L	M
Bohlooli et *al.* [26]	Collected vehicle movement data, mathematical simulation	M	L	L	L	L	L
Paik et *al.* [171]	Numerical analysis	L	L	L	M	L	M
Liu et *al.* [135]	Mathematical simulation	M	H	M	L	M	M

vehicle are sent over the network to be compared to a database. The database is then used to determine the closest matching vehicle, and return the next most likely AP. In addition to the GPS prediction, the vehicles also record network information stored within the database. This includes dwell times, signal-to-noise ratios, and handover timing. By also having access to the network information, the handover timing can be predicted, and packets can be forwarded appropriately to further reduce handover costs.

Instead of using GPS measurements, Bohlooli et *al.* [26] take advantage of the road movement restrictions in their pattern matching. A vehicle can only move in a straight line, except at road junctions, where there are turn possibilities. Thus, the method uses vehicle turning directions and road segment lengths to make predictions. A vehicle's turning sequence is recorded and compared to previously recorded turn sequences. The closest matching sequence is then used to predict the vehicle's future turn decisions. With the additional information on road lengths, the vehicle's path between road junctions is predicted with minimal calculation. The benefit of this approach is that it provides details on a vehicle's entire path, while only having to make pattern comparisons at each intersection. However, an incorrect turn prediction will lead to large errors in path prediction. Additionally, the lack of consideration given to vehicle speed and acceleration negatively impacts this approach. This makes two path options of similar probability difficult to distinguish.

Statistical analysis and movement projection prediction approaches have been shown to perform accurately under certain circumstances, and also have problematic scenarios. The strengths and weaknesses of each approach compliments the other, giving rise to hybrid approaches. One example is that of Paik [171], who conducts movement projection and probability individually, then weighs them against each other based on a confidence variable. The author also introduces a threshold value to ensure that a prediction is not made based on faulty network data. The threshold represents the distance the vehicle should travel before making a prediction; otherwise, signal interference could cause the vehicle to prematurely output a prediction. By having the vehicle wait a certain distance away, enough information is acquired to produce a more reliable prediction. However, a potential problem with a distance threshold is that a vehicle may change direction in an unexpected way, resulting in a much shorter distance till the handover; the threshold would prevent any prediction from being made.

Liu et *al.* [135] use movement projection for short-term prediction and a pattern-matching cell prediction for long-term prediction. First, the movement projection is used to determine the next most likely AP, achieved by considering the vehicle's current movement while within the area of an AP. After the vehicle has moved through multiple APs, its overall AP pattern is compared to the previous vehicle's AP history. This is used to determine a more generalized projection of the vehicle's movement. These two operations

are conducted separately, as done in [171], except that the pattern matching approach will sometimes use the movement projection's predicted AP to help differentiate between similar pattern predictions. By including some overlap between the two methods, the prediction is more informed and accuracy is improved.

Neighbor Discovery

Beyond addressing prediction accuracy, other aspects of the predictive handover have also been investigated in the literature. One such aspect is the discovery of neighbor AP information, which is required for the vehicle to begin an early handover registration before actually reaching the AP. A popular approach for discovering neighbors was developed by Mishra et *al.* [150], who introduce neighbor graphs that are populated through the use of context message communication between vehicles and APs. Whenever a vehicle conducts a handover between two APs, it sends two context messages. The first is sent to the new AP with information on the previous AP, and the second is sent to the previous AP providing information on the new AP. The APs are then able to discover neighboring APs to which a vehicle may connect next. A large benefit of this approach is that it relies only on network information. This allows the neighbor graphs to be built without outside influence, and still allows APs to discover neighbors that are potentially far away.

One example of a prediction approach that uses the neighbor discovery method is proposed by Hadjiefthymiades et *al.* [78], who also introduce a datagram relocation coordinator to conduct the prediction and manage resources. The coordinator utilizes the vehicle's past movements and the neighbor graphs to predict the most likely APs for the handover. The coordinator then initiates packet buffering at these APs, minimizing packet loss. By having neighbor information and predicting multiple neighbor APs, the approach provided a reliable improvement to packet loss. However, the approach requires high overhead to buffer packets for multiple neighbors, and begins to experience large degradation in performance as vehicle density increases.

Summary

Literature on predictive handovers has mostly focused on proposing methods to improve the prediction accuracy. Using probability analysis, movement projection, or pattern matching, the various methods attempt to determine which AP the vehicle will connect to next. However, issues remain within each approach. First, probability analyses experience difficulty when two AP options have similar probabilities due to their inability to differentiate

Table 3.9: Mobility Management based on chosen Network.

Classification	Method	Benefits	Challenges
HetNet-based	Adds small cells, interface between different RAT	Improves data rates and reliability	Deployment and transition criteria between different cells, ping-pong, HO failure
SDN-based	Separation of data and control planes	Improves network management	Ping-pong events, Densification and frequent HO issues
Fog-based	Offload control/functions to fog cluster area	Closer to edge devices	When and where to offload
Hybrid-based	Combination of SDN, Fog, Cloud, Virtualization	Improves network management and data rate	Deployment, Coherence and management between different schemes

between vehicles. Next, movement projection fails when there are sudden turns, as it is unable to consider the probability of a vehicle changing direction. Finally, pattern matching requires very high complexity and calculation requirements to maintain and compare many different patterns, while still experiencing, to a lesser extent, the issues of the other approaches.

3.3 Mobility Management Based on Network's Architectural Design

In this section, we focus on mobility management solutions based on deployed the network's architecture, including HetNets- , SDN-, Fog-, and Hybrid-based networks. We compare each design methodology, benefits, and problems when dealing with vehicular networks mobility.

3.3.1 HetNet-based Networks

Mobility management (MM) has been initially defined in the LTE standard to solely involve macrocells, which has been widely studied in the literature [70] and reported reliable and seamless mobility. One of the reasons is the macrocells massive size, which means minimal handover failure (HOF) and ping-pong (PP) events are expected to occur. However, with the increasing number of smallcells and different radio access technologies that are combined in HetNets, new mobility management for HetNets is needed.

The handover process between cells, Radio Access Technologies (RATs), or carriers is performed differently. It requires proper management to enable vehicles to cross from one cell to another while preserving the same quality of service. The handover management offers a seamless transition between similar types of cells, different cell types, and distinct access technologies. Two main types of handover may occur in HetNets, *Horizontal* handover and *Vertical* handover. The horizontal handover (HHO) deals with the transfer of ongoing sessions within the same technology or the same network, while vertical handover (VHO) happens between cells of different technologies. In HHO HetNets, several scenarios may occur, which include *Hand-in*, when the mobile device moves from macrocell coverage to smallcell coverage; *Hand-out*, when the mobile device returns from smallcell coverage to macrocell coverage; and *Inter-(H)eNB*, when a mobile device transition between smallcells. As for VHO, three possibilities of handoff can be present, inter-RAT, inter-LTE, and inter-Technology. The latter is concerned with the handover process between 3GPP and non-3GPP access technologies. The handoff process between different radio access technologies within the cellular networks is referred to as inter-RAT. While, the inter-LTE handoff is done between different versions of LTE (3G,4G,..). The 3GPP has since presented the Access Network Discovery and Selection Function (ANDSF) [70] as a new element to the Evolved Packet Core (EPC). This function supports the UEs to locate and connect to different RATs (e.g., WiFi, WiMAX).

The 3GPP specifies two standard interfaces between the wireless access point and the network core, namely X2 and S1. The X2 interface is designed for the handover between two eNBs served within the same MME pool, and considered faster than S1 and defined only for Intra LTE handover. While S1 is used for Intra LTE and occasionally inter-RAT, only if the two eNBs are not connected to the same MME, X2 interface is not defined between two eNBs, or X2 procedure has failed due to errors. In the case of inter-HeNB, either S1 or X2 interface is used, except when control access is required in the Mobility Management Entity, thus requiring the transition of handover requests through the EPC. The hand-out process can be directly done through the X2 interface unless it does not exist. Therefore, control packets need to be delivered through infrastructure links to the

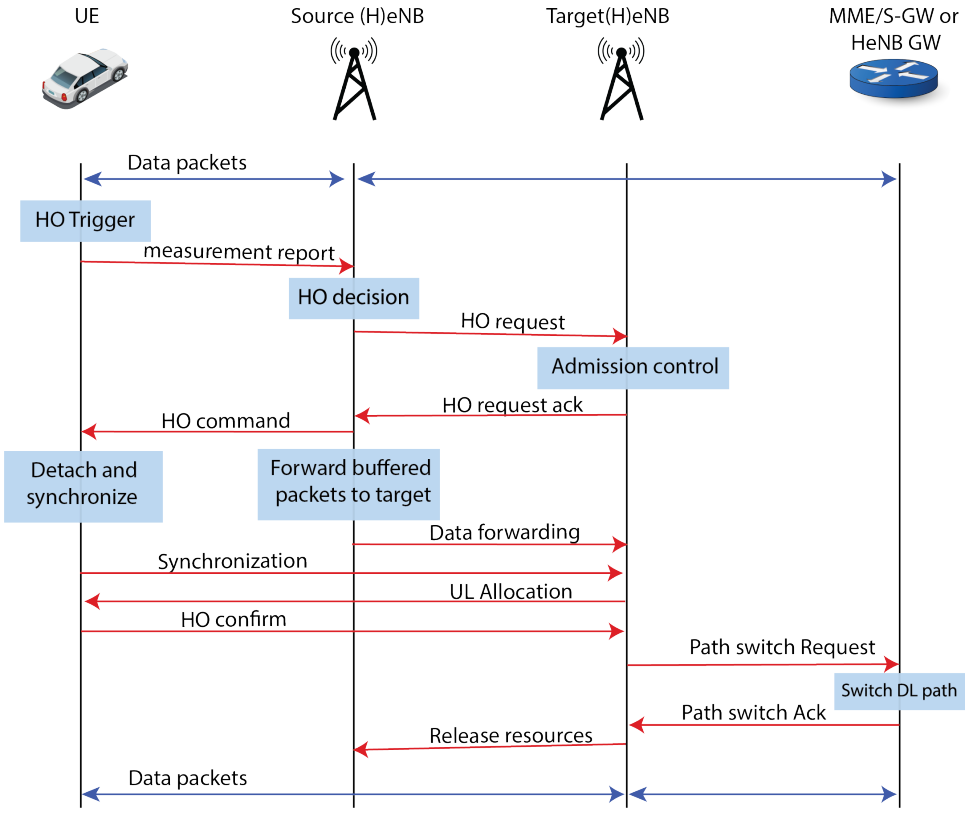


Figure 3.3: Handover process in HetNets.

CN. However, because cells have different backhaul routes, it is the most challenging case.

In general, the handover process in HetNets is known to be a network-based terminal-assisted process, which means that the serving cell is the one responsible for the handover decision, while the user equipment (UE) (i.e., vehicle) gathers the measurement information. The handover process is divided into several phases: measurement/initiation (Trigger), decision making (Network selection), and execution, as seen in Figure 3.3. The handover initiation may be done on the UE or by the network. In either way, multiple measurements of near approximate access networks are gathered by the UE, such as Bet Error Rate, the distance between UE and cells, and received signal strength. Then, when the UE detects a trigger event, a measurement report is sent to the source cell to derive a handover decision. In the decision making (selection) step, a choice is made to which cell should the vehicle connects to next depending on the measurement report, resource availability, and network load. Finally, the execution phase is when a new link is set up with the target cell to authenticate, synchronize, and reconfigure network resources and

settings. However, each step of the handover process may lead to an unpleasant experience for users. In the following, we address each stage of the handover procedure independently in HetNet-based vehicular networks and report on the methods currently suggested in the related works.

Table 3.10: Handover trigger schemes in HetNet-based vehicular networks.

Related work	**Measurements**	**Method/ Model**	**Goal**	**Network**	**Limitation**
Xenakis et *al.* [225]	SINR	dynamic hysteresis margin	reduce UE's power transmission	Two-tier femtocells	ping-pong effect, not applicable for high speed
Wu et *al.* [219]	RSSI, Velocity	Low/high speed triggers	offload macrocells traffic, reduces redundant HO	Femto-macrocells	highway environment only, overhead of scanning
Li et *al.* [128]	RSRP	Time-series prediction	Reduce HO latency	Two-tier femtocells	limited to single measurement
Becvar et *al.* [20]	RSSI	Two-thresholds prediction	Reduce redundant HO	Macrocells	Insufficient use of recourse
Omheni et *al.* [165]	SINR, RSSI	Pre-HO threshold	Reduce HO delay and signalling	Inter-RAT	movement and velocity is not considered
Kita-gawa et *al.* [115]	RSRP, failure cause, failure rate	Adjustable HO margin	reduce HO failure, ping pong	macrocells	Limited to macocells

Handover Initiation (Trigger) Phase:

Typically, the handover trigger is initiated due to several factors, either to offload high data traffic from macrocell, to enhance the received signal strength at the terminal device, or about to leave the current serving cell. In HetNets, the handover trigger is made at the serving cell with the assistant of the UE measurements report. Each vehicle integrated with a User Equipment (UE) will periodically monitor the downlink signal strength of nearby cells and processes the information by eliminating the signals' fading impacts and estimating defects in the handover measurements. If a criterion of the handover event is encountered within Time to Trigger (TTT), then a measurement report will be transferred by the user equipment to the current serving cell.

The standard rules of the handover function are categorized from A1 to A6 and B1 to B2 events, as specified in [2]. Event A3 is the most commonly used event to trigger the handover process, which is when the RSS value of the new cell exceeds the current cell. The Time to Trigger guarantees that a ping-pong event can be eliminated by specifying a time window in which a triggering event occurs to transfer the measurement record to the current cell. However, the static selection of the handover hysteresis and TTT values is no longer effective in HetNets [127]. A large TTT value with a high-speeding vehicle may experience an increased degradation in the reference signal received power (RSRP), referred to as the HO failure problem. Whereas a low TTT value will cause very frequent handover to or from cells, referred to as ping-pong problem, leading to high-performance degradation.

Besides, the reference signal received power (RSRP) estimations are conventionally used for handover measurements. However, in heterogeneous networks, different coverage area sizes may lead to mobility performance degradation, if we continue using the same set of parameters. Several studies worked on adding more measurements such as user preferences, velocity, cost, power consumption, security, handoff latency, RSS, available bandwidth, network connection duration. In the following, we discuss the solutions presented in related works that address the drawbacks of the conventional handover trigger phase.

RSS-based HO trigger schemes: The Received Signal Strength is the computed signal power amount that is received by the end entity. Many schemes continued to use the received signal strength (RSS) as an indication to initiate or trigger the handover process. Each vehicle moving within the communication range of a cell will periodically measure the RSS value from each neighboring cell. However, a fixed threshold value is not anymore applicable in heterogeneous networks. Some research studies proposed the use of a dynamic RSS threshold value depending on mobile node velocity [152]. A vertical handoff protocol based on MIH/PMIPv6 for optimization is proposed in [165], in which they added a pre-handover process of resources checking, authentication, resources reservation, and IP

assignment. The authors defined two thresholds to help predict the degradation of link quality. However, additional context messaging overhead is introduced in their work.

SINR-based HO trigger schemes: The signal-to-interference and noise ratio (SINR) defines the quality of wireless cells by computing the average received power from a signal reference signal, which can represent noise and interference from the network. A two-tier LTE network handover decision is proposed in [225] to reduce the transmission energy of mobile terminals. The proposed scheme is dependent on adjusting the handover Hysteresis Margin to a defined signal-to-interference and noise ratio threshold and using the quality measurements of the target cell. However, having a predefined threshold value is not an optimal solution with the presence of several radio access technologies.

Mobility-based HO trigger schemes: In heterogeneous vehicular networks, a single indicator such as RSS or SINR is inefficient, since different criteria need to be looked at before a decision is reached, in which some of them might be conflicting. The authors in [219] proposed a speed-based handover scan trigger scheme in Femto-Macrocell Networks. When the speed of the mobile terminal is higher than a pre-defined value, then the mobile terminal will only scan macrocells, thereby avoiding unnecessary handover to femtocells. However, pre-defined values usually tend to be unrealistic, which needs to be redefined for different scenarios and environments. It is noteworthy that mobility-based solutions (i.e., vehicle's location/speed) have been intensively used for the handover decision (selection) phase, and very few studies used mobility for the handover trigger phase.

Prediction-based HO trigger schemes: Predicting the handover process ahead of time could save a lot in handover latency by initiating the handover early enough to complete the necessary resource allocation in the target cells. [128] proposed a predictive solution based on time series analysis theory using the reference signal received power (RSRP). The handover trigger works in two-priority event evaluation method, in which the classical handover trigger first initiated, if no target cell meets a specific condition, then the predictive trigger method is commenced. Results showed that the handover latency is reduced with high prediction accuracy. In another work [20], a two threshold-based handover mechanism is proposed using the RSSI parameter with hysteresis. Their scheme uses a sequence of RSSI samples to determine the number of estimated handovers between the current and the target cells. Therefore, assessing the handover probability of the available cells. Additionally, the authors studied the impact of RSSI fluctuations on the handover prediction. Another work by [200], used a grey theory to implement a handover trigger scheme for 5G HetNet in railways. A handover trigger prediction is performed by using the received signal quality (signal-to-interference) from the measurement report, into the grey model, to predict the next measurements; thereby triggering the handover ahead

of time to enhance the overall performance.

Handover Decision (Network Selection) Phase:

According to the measurement report, the serving cell or eNB chooses the best-reported target cell for the upcoming handover (usually A3 event). Then the preparation for the handover begins, where the handover procedure is completed by the current serving cell and, together with the target cell, performs the handover execution. The impact of the network choice is more notable in the presence of larger heterogeneous dense networks and varying cells' coverage. Many handover decision algorithms for HetNets were proposed in the literature, in which most of them choose the target cell based on the type of traffic, signal strength, or UE speed.

However, the handover decision should not be any more limited to the RSS measurement but also the cells' interference levels, the required quality of service, available bandwidth, cell load, current user preferences, and cost. Each set of measurements is categorized as network metrics, user preferences, or application requirements. The network metrics include cell coverage, cost, available capacity, load, communication latency, and more. In terms of user preferences, it consists of the available budget, desired quality, energy conservation needs, and so on. The application requirements combine the quality threshold, delay, throughput needs, jitter, and packet loss.

MADM-based network selection schemes: Multiple Attribute Decision Making (MADM) provides a decision-making tool to consider multiple criteria for decision making. The literature presented several MADM techniques, each with a distinct way to process the measurement's set. The Simple Additive Weighting (SAW) [236] is one of the most common MADM techniques that compute the weighted sum of each network cell's values that can be reached by the vehicle. Whereas the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) chooses the access point closet to the optimal selection that is relatively stable and far from adverse. The Multiplicative Exponential Weighting (MEW) utilizes multiplication to connect each cell's parameter score. In addition, ELECTRE is focused on a pair-wise contrast between the applicant networks ' parameters. A performance evaluation comparison of MADM methods is presented in [208], for cell selection in wireless networks.

A network selection scheme is proposed by [159] for V2I communication over multi-tier HetNets, in which vehicles are designed with two interface cards for both LTE-A and Wi-Fi. Their network measurements include signal strength, trajectory, distance, average link duration time, and load, where the vehicle is responsible for initiating the network selec-

Table 3.11: Handover decision solutions in HetNet-based vehicular networks.

Related work	**Measurements**	**Model**	**Goal**	**Network**
Stevens-Navarro et *al.* [203]	link reward function QoS, signal cost function	Markov decision process	Reduce number of VHO, Maximize QoS	Inter-RAT
Huang et *al.* [74]	SINR, TTT	Markov Chain	Reduce Ping-pong events	Macro-, Femtocells
Zang et *al.* [237]	(connection, location, and velocity) (power, bandwidth, and location of BSs)	Reward Markov Decision Process	Reduce Frequent HO	mmWave small-, Macrocells
Moon et *al.* [155]	SINR, state	state-dependent, probability	reduce HO failure	small- , Macrocells
Guidolin et *al.* [75]	context-aware cell traffic load,	non-homogeneous Markov chain	reduce frequent HO	femto-, macrocells
Lee et *al.* [127]	UE speed, cell configuration	UE speed grouping	reduce HO link failure and ping pong event	pico-, macrocells
Jeong et *al.* [99]	movement pattern	Next location prediction	Reduce unnecessary HO	femto-, macrocells
Wang et *al.* [217]	RSS, data rate, BER, movement trend	decision-tree and feedback	Reduce frequent HO	inter-RAT
Demarchou et *al.* [55]	distance, trajectory, X-threshold	Markov chain Skipping technique	Reduce HO failures	ultra-dense cells
Wu et *al.* [220]	RSRP (trigger), CDR and HOR	dynamic fuzzy Q-Learning	Reduce number of HO	macro- ,smallcells

tion before approaching the target cell coverage. Their model was driven by the Access Network Discovery and Selection function (ANDSF), as discussed in Rel-8. However, they eliminate opposite direction cells from the selection method, which could be problematic in some scenarios where the vehicle movement is unpredictable. They tested their scheme on macrocells and smallcells network with LTE-A and IEEE802.11n communication technologies, showing high throughput and packet delivery ratio in comparison to conventional RSS-based network selection method.

A network selection scheme for HetNets was presented in [236], based on the user-selected features and network attributes, by combining multiple MADM approaches. The mixed-methods calculates the utility function value of each adjacent network and uses a threshold value to avoid the ping-pong event. Their simulation scenario is based on the selection between WLAN, LTE-A, GSM, and UMTS. The UE collects six network parameters, including capacity, bit error rate, latency, delay, packet drop rate, jitter, and service cost. A network selection process is performed given the surrounding set of available base stations, network parameters, and traffic classes. The FAHP method is used to compute the network parameters' weight and use preference values. The TOPSIS method is used to calculate user preference's utility values, while the Entropy method finds the network parameters' weights. Then an integration between Entropy and FAHP results is used to find the ultimate network's parameter values combined with the user's preference values to derive a utility rate for each base station. Their findings indicate a decrease in the number of handovers with relatively high gain values compared to different hybrid-based network selection methods. However, their work is constrained by a static threshold value that needed to be adjusted according to that specific scenario.

Intelligent-based network selection schemes: The selection process of the next network cell was proposed in [158] based on fuzzy logic for V2I communication over heterogeneous wireless networks. In which, vehicles will self-evaluate nearby access points candidates to select the best network based on fuzzy logic inference system. The selected network environment includes WLAN roadside units and LTE macrocell, in aim to decrease the delay in handover by preselecting the target network ahead of time. Vehicles calculate candidates' dwelling time using the cell's location and the vehicle's direction and choose the maximum one. However, their work doesn't take into account any QoS parameters and users' preferences, which are both essential in nowadays vehicular application. To tackle this problem, a handover decision approach has been introduced in [71] based on the fuzzy Analytic Hierarchy Process (AHP). The work in [71], presented a fuzzy matrix containing several network parameters (bandwidth, delay, jitter, BER, and cost) depending on the used application (voice, video, or best-effort application) to derive crisp weights. Another work in [220] proposed a dynamic fuzzy QLearning method for smallcell network mobil-

ity management, in which fuzzy rules are constantly generated through system learning. They simulated a UE movement with 10 km/h average speed and compared to CDR (measures user experience) and HOR (measures signaling load) with different time-to-trigger thresholds.

A vertical handoff protocol proposed in [217], based on the decision trees model for network selection using WiMAX, WAVE, and cellular networks. They first define the transition probability distribution through multiple parameters, which include transmission rate, signal strength, bit error rate, location trend, and blocking probability. The movement trend is described as the relationship between vehicle movement trend and access points [217]. They introduced four types of user preferences, continuous network, network bandwidth, network cost, and service orientation priorities. The handoff decision will then be measured according to the chosen user preferences. A feedback mechanism is added to the handoff decision in [217] based on the vehicle service and motion states. However, their decision tree model calculation time was reported to be over $180ms$, which is relatively high for time-sensitive applications and high-speed vehicles.

Several studies investigated the handover design in mmWave-based heterogeneous networks [68, 145, 237]. A handover decision scheme method was introduced in [237] using a Markov Decision Process (MDP) to enhance users' quality of experience in mmWave-based heterogeneous networks using user movement data. To estimate the quality of experience, the weighted sum of link throughput and the handover costs are used. Additionally, they added a candidate elimination method to reduce the model complexity according to specific options. Their algorithm avoids excessive handovers compared to other benchmark schemes (SINR-based handoff, Simple additive weight, and reinforcement learning [SARSA, Q-learning]). Mezzavilla et *al.* [145] introduced Value Iteration Algorithm (VIA) and a Markov decision process (MDP) framework for optimal handoff decisions in mmWave Cellular networks. Another work in [43] proposed a vertical handover decision algorithm based on a Markov model using several networks and mobility parameters. A reward and cost functions are used to derive a total QoS reward value. Results were compared to SAW and RSS based network selection methods. A dual connectivity protocol with an uplink control signaling system was proposed in [68] to improve handover in 5G mmWave Mobile Networks. However, having an intelligent-based handover scheme could introduce higher complexity rates and decision processing delays, in addition to elevated signaling overhead.

Decision Functions-based network selection schemes: The use of cost functions has been seen as another approach for network selection to support performance efficiency for different types of quality specifications in energy, cost, and service use. An energy-utility function based network selection scheme, proposed in [161], for heterogeneous networks. The network selection is user-centric, where the terminal user chooses the access network

based on several measurements, including cost, link quality, battery consumption, network load, and velocity. Their goal is to provide an efficient energy-based handoff management scheme to reduce power consumption. The used cost function is based on the billing of voice call duration and downlink data volume. Their utility function relies on power usage gain, cost, network burden, and link stability to produce the best network selection decision.

Cost-based decision-making techniques for mobility handover combine several variables, such as energy consumption, cost, and bandwidth. Different measurements are assigned different weights according to the network's environment and user preferences. A utility function optimization method was proposed in [222] for network selection in heterogeneous networks. Taking into account the mobile terminal QoS requirements and preferences, distinct services, user channel state information, and network traffic load. However, a decision function is usually a reactive approach that may yield an increased handover latency and is unsuitable for real-time applications.

D2D-based network selection schemes: Communication technologies such as WiFi Direct, Bluetooth, and Near Field (NF) can be used to assist in the handover process, named Device-to-Device (D2D) communications. [167] proposed a handover scheme that incorporates D2D communication to reduce energy consumption and increase packet delivery ratio, using a stochastic geometry tool. The UEs have the opportunity to connect through the D2D channel when the cell's signal quality value degrades, which generally happens due to the UE moving toward the edge of a cell's communication range. Simulation was conducted using macrocells and compared to conventional LTE handover methods based on several conditions, A2 to A4 events (as presented in 3GPP LTE release 8).

Chen et *al.* proposed, in [42], a D2D joint and a half handover scheme using the devices' signal quality. The handover decision method takes into account several parameters, including RSRP, time to trigger, and a threshold value. A Time-to-Trigger and a threshold value are explicitly assigned to D2D handoff decisions. In another work [233], Yilmaz et *al.* introduced a cluster-based solution for D2D handoff trigger, by postponing the handover decision between two cells and the UEs until the RSRP value is below a predefined D2D control condition threshold. The use of D2D can lead to several challenges [169], including an increase in energy consumption of mobile devices, interference management, resource allocation, and handover management.

Handover Execution/Completion Phase:

In the final step of the handover process, the serving and target cell carry out with the aid of the UE, the execution or completion phase through specific network protocols. During

Table 3.12: Handover execution types.

Type	current RSU	new RSU	HO	interface
1	eNB	eNB	Normal-	X2
2	HeNB	eNB	Hand-out	S1
3	eNB	HeNB	Hand-in	S1
4	HeNB	HeNB	Inter-	S1/X2
5	3GPP	non-3GPP	inter-RAT	Xn, MIH

this phase, the communication link of a UE is migrated to the target cell and finished with the handover process completion. The amount and direction of the HO execution method signaling rely on the type of the serving or target cells, and the need to use the access control, as seen in Table 3.12.

Each type of handover treats the execution phase differently. In the case of the hand-out handover (i.e., from smallcell to macrocell), an SN status transfer is established between the serving smallcell to the target cell through the Mobility Management Entity (MME), and the target cell will start buffering packets from the serving cell. For the handover from macrocell to smallcell, a request for transfer will be sent directly to the MME, including the information of the target HeNB. Then, the MME will validate the UE status on the target HeNB, and transfer the UE information to the target cell (i.e., HeNB). The target cell will then verifies the reported handover and allocate a set of resources for the UE. Finally, an acknowledgment message will be sent by the target cell back to the MME, and the handover confirmation is returned to the UE through the serving cell. Moreover, if the handover is from smallcell to macrocell, then the target cell will receive the handover request directly. After the handover command is completed, a path switch request is received by the MME and the serving cell to redirect the DL path, and a UE context release is made by the source cell.

Discussion

Depending on the network structure, given scenario, and available measurements, choosing the optimal set of criteria to evaluate the handover trigger, decision, and execution is vital. In efforts to reduce traffic load, control overhead in the overall network performance, many research studies presented solutions in each step of the handover process, and with different measurements set. Although several methods were proposed to enhance the handover trigger phase, it yet suffers from very early or late handover initiations. Hence, the development of an intelligent handover trigger scheme is still an open challenge for

5G-enabled vehicular networks. In the handover decision making solutions, MADM-based network selection methods are simple and straightforward, relying on specific formulas to obtain a result. On the other hand, Intelligent-based network selection schemes that usually undergo many iterations by probabilistic rules to achieve optimal results. However, with very few numbers of repetitions, this would yield incorrect results [236]. Besides, even though sophisticated mathematical techniques usually produce intelligent-based models, it suffers from an increased processing delay [159].

Recently, the 3GPP has released new documentation on fully independent 5G systems [Rel. 15/16], which specify a new standard for 5G cellular systems, and include three main components, 5G access network (5G-AN), 5G core network (5GC) and user equipment (UE) [2]. Furthermore, a new separation of control and user planes functions have been introduced, where the MME entity in previous releases is replaced by a session management function (SMF), in control of the UE IP address allocation and PDU session control, Access and mobility management function (AMF), controls idle state mobility handling and NAS security, and user plane function (UPF), controls packet processing and transmission operations. Besides, the 5G introduced some new functionalities, such as network slicing, new QoS framework, and a new approach for service-based architecture concept. The handover procedure uses a new Xn or N2 reference points to transfer a UE from a source NG-RAN node to a target NG-RAN node. The triggering of a handover process is similar to the previous LTE-A protocol, such as new radio conditions, load balancing, or due to specific QoS flow events. Several types of handover process presented, inter NG-RAN handover (Xn based, with three variances, and N2-based), and handover process between 3GPP and non-3GPP access. Nonetheless, due to the lack of recent research studies on such systems for vehicular network environment, one can foresee similar mobility management issues and concerns that appear in cellular 5G networks.

3.3.2 SDN-based Networks

The concept of an SDN-based vehicular network introduces a significant advantage in handling and managing the various technologies for the next generation of vehicular network communications. In [122], a theoretical comparison of the performance of X2 handover and SDN-based handover in terms of signaling cost and the influence of the network size. It is noticed that by increasing the number of access points and mobility rates, the handover based on SDN managed to reduce the signaling cost compared to the X2 standard. A general illustration of the handover process in SDN-based networks is presented in Figure 3.4.

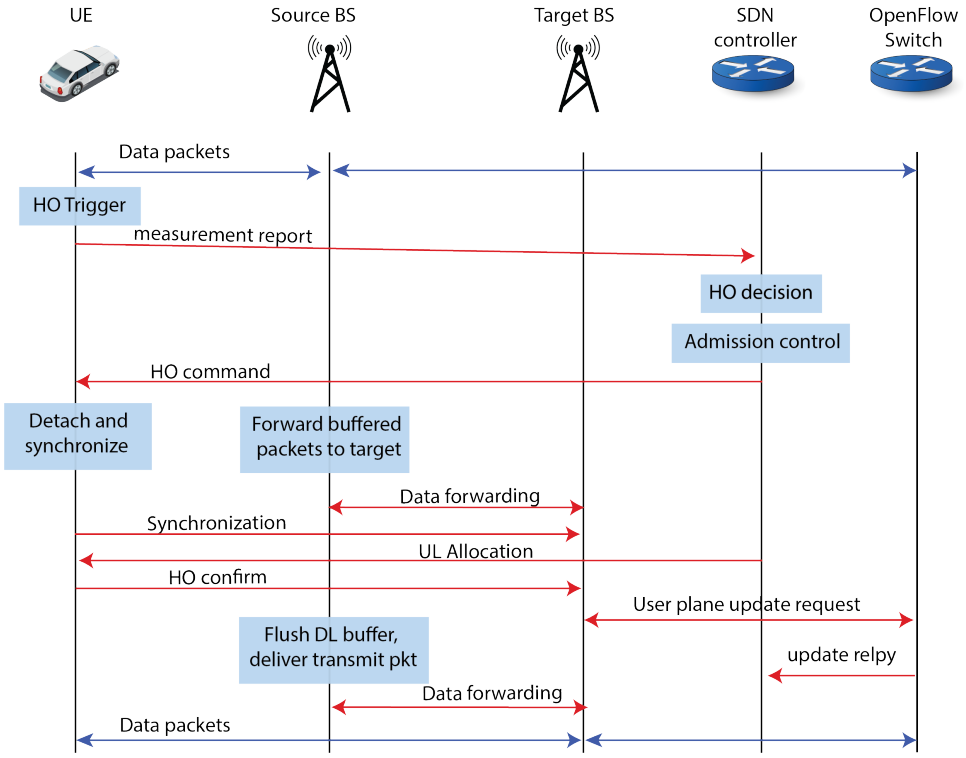


Figure 3.4: SDN-based vehicular network handoff process.

Offloading Mechanism

Even though current smartphones provide an offloading mechanism from cellular to WiFi, it is still not recommended for latency-critical safety applications or in case of autonomous driving. Therefore, efficient handover mechanisms are needed to transfer communication from cellular to WiFi and vice versa without having QoS degradation. When and to which access point should the traffic offload go to, is one of the questions that need to be answered before performing a handover decision. Huang et *al.* [93] introduced a predictive control method named OHD-SDN for offloading V2I communication using SDN-based architecture. They used velocity, direction, location, and neighboring roadside units to derive a handover decision between cellular and IEEE802.11p networks. The SDN structure is used as a mobility management controller to decide on the time of the handover trigger. Their goal was to reduce the traffic load of cellular networks without losing the quality of service.

Aujla et *al.* [15] proposed an SDN-based offloading mechanism for vehicular networks through the use of a priority manager and a load balancer. They defined different types

of offloading techniques, including the WiFi and IP flow offloading. They stated four procedures for implementing their protocol, a network selector, a priority manager, an offload manager, and traffic load balancing method. Starting off with the offloading manager, that calculates the load on a controller using the total number of packets one controller can handle and the number of received packets. Then, messages will be either routed using the flow tables or delivered to the priority manager based on a threshold value. A handover decision is based on one-leader multiple-following Stackelberg games to select networks. A comparison was made over generated traffic movements from the city of Patiala. Results indicate less number of handovers occurred while maintaining reasonable data throughput. However, this work does not consider the movement of the vehicle in different networks.

Network Densification

The centralized concept of SDN networks could lead to massive traffic congestion, especially in ultra dense-networks. The new trend of fully distributed handover decision is considered in recent works. Sharma et *al.* [194] proposed an enhanced SDN structure by utilizing unmanned aerial vehicles (UAVs) as on-demand forwarding switches. The UEs acts as terminals to UAVs, and the latter are terminal to macro Base stations (MBSs). They proposed two approaches, a centralized approach, with central mobility management entity, and a distributed approach. The measurements report from the UE includes the RSSI, RSRP, RSRQ, and channel quality indicator. Performance evaluation was measured in terms of signaling overhead, handover latency, and delay.

Bilen et *al.* [25] proposed a handover scheme based on the Markov chain in a software-defined 5G network with the presence of ultra-densification. They argue that minimizing the handover delay will improve the performance in ultra-dense 5G networks. The Markov chain scheme is based on the estimation of available resources and transition probabilities to assign the most optimal base station to the mobile node's OpenFlow table ahead of time, thus reducing the delay of the handover process. The authors in [25] modeled the transition probability from one cell to neighboring cells through the physical movement of the mobile node. However, no further parameters were considered in the model, such as mobile node speed, direction, or QoS requirements.

Kaul et *al.* [108] proposed a handover and load balancing scheme on top of their previous work on dynamic network control for ITS, named (D2-ITS). Their early work is focused on implementing a hierarchical control, which can be altered to the network conditions and environment. Their objective is to transfer one vehicle's control functionality to another device as the vehicle moves. Handover activation and next controller selection are based on RSSI values and can be triggered either the controller or the vehicle.

Zhao e t *al.* [248] proposed a context-aware handover scheme based on multiple criteria for software-defined 5G heterogeneous networks, using fuzzy logic rules. The parameter gathered from the user equipment is sent to the centralized SDN server, including the network measurements, user preferences, and usage requirements. In particular, they consider both network and mobile user measurements, such as the network RSSI, range, throughput, and the mobile user's speed, direction, and quality of experience. Then the handover process is triggered based on the changed behavior of the gathered measurements by a predefined threshold. The authors proposed three fuzzy-rule layers that consider multiple factors, including mobility, network, and satisfaction. Their protocol was compared to the based RSSI-based handover scheme in terms of throughput, bandwidth cost, and handover frequency. The users' mobility model is simulated on random way-point with an average speed of $10m/s$. Still, it was not clear how their work would react in case of high-speed vehicle mobility, although it could be adjusted for low-speed movement and intersection areas.

Fondo-Ferreiro et *al.* [65] introduced a fast handover decision protocol in SDN-enabled wireless networks using flow-type predictions of past user behavior. Both centralized and distributed algorithm allocation was tested using real data. The flow predictor is based on a simple Markov chain model to estimate the next service flow of a user. The handover decision is based on the current state of the access point, quality of radio links, and user needs' based on past application usage history.

Discussion

The SDN paradigm is recently considered one of the solutions to the heterogeneity between different network technologies. SDN-enabled Internet of Vehicles (IoVs) is reported to provide high resource utilization efficiency and ease mobility management [14]. In the aforementioned handover solutions, two main concerns have been studied, namely offloading mechanism and densification issues in SDN-based vehicular networks. A comparison of current handover solutions in SDN-based vehicular networks is presented in Table 3.13.

The handover decision in such networks is either to reduce the load on macrocells or communication session needs to be transferred to another better quality cell. One of the main challenges in SDN-based vehicular networks is in the presence of ultra-dense systems, because of the centralized nature of SDN structure; this may lead to traffic congestion issues. Several surveys have investigated the distributed principle of SDN networks, but the implementation and governance of such networks are still underway. Whilst most mobility management solutions aim to either reduce the load on cells or reduce the overall

Table 3.13: Handover solutions in SDN-based VeNET.

Articles	**Model**	**Performance metrics**	**Goal**
Huang et *al.* [93]	HO decision - Offloading technique	Throughput, Packet loss	Reduce load on cells
Aujla et *al.* [15]	Network selection based on Game theory	number of HO's	Reduce load on cells
Sharma et *al.* [194]	On-demand UAVs forwarding switches	End-to-End delay, signaling overhead, and handover latency	Distributed approach
Zhao et *al.* [248]	Multi-attribute hierarchical fuzzy inference	throughput, bandwidth cost	Reduce Frequent HO's
Bilen et *al.* [25]	Markov chain (available resource estimation and transition probabilities)	HO delay, HO failure ratio	Reduce handover delay

handoff delay, not many research studies have been done toward secure- and energy-efficient mobility management schemes in vehicular SDN-based networks.

3.3.3 Fog-based Vehicular Networks

With conventional heterogeneous vehicular networks, several problems may occur during the handover process such as ping-pong, radio link failures, small coverage area, severe co-channel interference, and heavy load on the core network. Unnecessary handovers or radio link failures are common, either due to the serving cells small coverage region or the high speed of vehicles. A vehicular fog-computing network shifts some of the BBU pool's control methods from/to the Fog-Access point (F-AP), Macro-Remote Radio Heads (MRRH), or Small-Remote Radio Heads (SRRH). The handover decision making and the process will be different than in HetNets or Cloud-RAN, to guarantee QoS to end-users.

Fog-enabled vehicle nodes (F-VNs) can be categorized into either high-speed vehicles or low-speed vehicles. The high-speed F-VN are preferably connected through macrocells with large coverage areas, whereas low-speed vehicles connect to smallcells or access points with lower coverage areas. Henceforth, handover management in such networks is a relatively challenging problem. Several types of handoffs may occur depending on the involved entities in the handover process. For example, when a vehicle (i.e., F-VN) moves between F-AP and MRRH, a measurement report is collected by the F-AP from the F-VN to perform a handover decision. Then a request for handover is processed by the F-AP to the MRRH using the F-AP gateway and BBU pool. An acknowledgment of request is then sent back to the source F-AP, and the same process happens between SRRH and MRRH. The handover process between similar points of F-APs or SRRHs is made through an S1 interface. Typically, the handover decision between MRRH to an F-AP is usually more challenging, because of the many available F-APs around.

A fast handover scheme based on neighboring vehicles, namely CVFH, for vehicular fog communication is proposed in [24]. They present a proactive handover mechanism to establish an advanced handoff process that relay on neighboring vehicles to select the new access router/point and acquire the IP address in advance. Meanwhile, vehicles can continue its communication link with the serving AP. Each vehicle keeps track of nearby vehicles through beaconing, where the most qualified neighbor is chosen to be the aided-vehicle. Qualification is based on several conditions; i) the chosen aided-vehicle must be served by a different access point, and two; ii) its position must be in front of the current vehicle, and no broadcast reply was sent to the current vehicle. If all conditions are met, that neighbor vehicle will send a response after a defined time interval [24].

The authors specified each packet format, authentication, association, association reply, neighbor request, neighbor reply, and success fail messages. Their scheme was compared with the IEEE802.11 handover standard in NS-2 simulation in terms of handoff delay and throughput. It was only tested on highway small scale scenario with vehicles' speed between 50 to 90 km/h.

Fog-based deployment on vehicles or access points will help in getting resources closer to edge devices. However, this feature creates a new issue, as some vehicles will be closer than others to the fog nodes. In the case of high-velocity vehicles, fast handovers will occur between fogs. One solution is to minimize the disruption time during the handover process, which has been studied in [142]. The authors presented a handover scheme based on a machine learning model for vehicular fog computing, using a feed-forward neural network to estimate an optimal fog-node. In addition, they used a recurrent neural network approach to assess the projected results at a particular location and time to minimize transition disruptions during fog-to-fog handover. The distance between the vehicle and fog node is initially used as a handover criterion, which will be used as heuristic data to identify the transition points between fogs. A comparison between different fog predictors and cost predictors showed their model selection is the highest in terms of accuracy. However, their work was limited to fog-to-fog handoffs and was not tested on heterogeneous networks and built on the assumption of known vehicles' trajectories.

In the same context, several related works focused on the network selection method in fog-based vehicular networks using the fuzzy logic approach. Recently, in [97], a vehicular fog architecture is proposed combing both fog-computing and multi-access edge computing (MEC) paradigms. In which, the MEC is used on the vehicular cloud as a centralized control. A proper selection methodology is needed to allow vehicles, acting as fog nodes, to access the MEC server or cloud. The fog gateways selection approach in [97] is to initially select a group of possible cells based on fuzzy logic and then optimize the number of selection through the Ant Colony Optimization technique. In the initial step, the selection of candidates is based on a specified set of parameters and the fog's type. The authors introduced two types of vehicular fog, static and mobile. In the static vehicular fog, that metric used to evaluate the candidates includes the quality of available networks links (i.e., RSSI), the number of neighbors, and the rate of stay (RoS) the vehicles spent in a parking space (represented by low, medium and high). As for the mobile vehicular fog, when vehicles are not parked and rather moving at low speed, the used parameters include only the amount of adjacent cells and link quality (RSSI). After the candidate set is formed, they used the ACO approach to optimize the selection set. However, vehicles movement are usually unexpected and vary in speed, which may affect the performance of their model.

Follow Me Fog (FMF) framework proposed in [16] to support a seamless handover

scheme between access points. The idea of pre-migrate computation jobs through monitoring received signal strengths. Their objective was to reduce latency when handover is triggered by using a pre-handover approach. The authors presented their method through a state transition model. To initiate the pre-migration state, each IoT device must keep track of the received signal strength of its serving and neighboring fog cells. When the handoff trigger threshold value is reached, the migration is also triggered. Following that, in case the trigger was falsely initiated, the migration stops and the IoT devices will go back to the initial step. However, relying on the received signal strength only may introduce higher communication overhead and unreliable handoff triggers.

Table 3.14: Handover solutions in Fog-based vehicular networks.

Articles	**Model**	**Measurements**	**Goal**	**Limitation**
Memon et *al.* [142]	Neural Networks	distance between fog and vehicle, time	predicting the best fog node	assumes known vehicles trajectory, only fog-to-fog HO
Bao et *al.* [16]	pre-migration concept	RSS	reduce HO latency	Limited to RSS for HO triggers
Jabri et *al.* [97]	Fuzzy logic + ACO	RSSI, RoS	Reduce comm. cost (bandwidth, cellular fees)	uncertainty in vehicles movements
Bi [24]	vehicle-assisted HO	RSSI, packet loss rate	Reduce HO latency	Adds comm. overhead

Discussion

The fog has shown to be an efficient solution to meet 5G requirements in terms of latency and scalability by providing data content and services closer to users rather than on the centralized cloud. Several proposed schemes considered utilizing slow-moving vehicles and halt vehicles as fog nodes to offload resources. However, the accuracy and efficiency of choosing vehicles and clusters have not been thoroughly explored. In addition, many related works did not look into the heterogeneity of wireless technologies in 5G-enabled

vehicular networks. Rather than that, most related works focused on one type of migrating or handover of information from one fog to another fog. A comparison of different fog-based networks is presented in Table 3.14.

3.3.4 Hybrid-based Networks

In the previous section, we discussed that the presence of frequent HO reduces the performance of SDN at roadside units. A possible solution is to derive a distributed or decentralized approach to deal with the vast control overhead that goes to the control plane. Some solutions worked on the integration of the fog and cloud on the network edge either on approximate small-cells or through vehicle clustering techniques. However, the proper deployment of such architecture and communication structure is essential to avoid performance degradation.

Table 3.15: Handover schemes in hybrid-based vehicular networks.

Articles	**Network**	**Measurements**	**Model**	**Goal**
Duan et *al.* [198]	5G-Cloud	User satisfaction, Speed, delay, pkt loss.	Fuzzy Logic	always best connected
Zhang et *al.* [246]	SDN and Fog	SINR	two-threshold conditions	Reduce signaling overhead
Prados-Garzon et *al.* [181]	SDN and partial virtualization	RSRP (A3 event)	MPLS-tunnels, X2-HO	Reduce HO delay
Qiu et *al.* [183]	NFV and Fog	RSRP (A3 event)	direct X2-interface between fogs	reduce signaling cost

A hybrid handover scheme proposed by Zhang et *al.* [246] for a software-defined and fog-based vehicular network, which models a QoS constraint for vehicles. The SDN controller is assumed to be charge of allocating appropriate communication links according to the vehicle speed, communication distance, and QoS requirements. The handover process is preformed on the edge cloud by the controller, therefore removing the load from the

core cloud. Two types of HOs are considered in their work, Intra-edge cloud handover (same cloud) using the X2 interface and inter-edge handover (different clouds) using the S1 interface. To avoid failure, they proposed a hybrid access handover scheme with two progressive conditions to decide the handover trigger time based on the SINR threshold value. Simulation results showed that their method presents better performance without the redundant signaling overhead. Authors failed to specify when to user Intra-/Inter-edge handovers. Also, they only consider the RSS-based trigger event of the X2/S1 handover, which doesn't take into consideration the dynamic nature of vehicles' movements and vehicular topology, leading to frequent unnecessary handoff's. A recent study looked into the integration between fog and mobile edge computing paradigm in [172], which can provide a considerable gain in terms of monetary cost and latency. For latency-sensitive applications at the edge of networks, the authors introduced a fog-assisted architecture for handover management in vehicular networks. Driven by OpenFog Consortium, services are moved close to the end devices and users, including the application logic, data, and networking services.

For 5G vehicular cloud computing technologies, Skondras et *al.* [198] proposed a fuzzy-based network selection method. The vertical handover trigger is performed in the fog, given the velocity of vehicles, and the network SINR, which are reported by vehicles, to assess whether or not a handover trigger is required. The cloud will select the new network cell depending on the velocity of the vehicles; in addition, vehicles will be notified of the decision via the fog infrastructure. The initiation of the transition and selection of networks is based on the Pentagonal Fuzzy interval values. In contrast, the Pentagonal Fuzzy ANP algorithm is used to estimate the weights of the handover trigger and the selection of the network. Their results indicate that the measurement of Always Best Connected is satisfied and outperformed existing vertical handover schemes in terms of user satisfaction grade and handover count.

Additionally, several solutions used Network Function Virtualization (VFN) to aid in the handover decision, which introduces an added-value in terms of scalability, cost, and energy consumption. Prados-Garzon et *al.* [181] proposed a handover scheme based on partial virtualization of 5G SDN-based networks. Their goal is to reduce the handover process execution time (i.e., latency), by operating at the link level. As for the measurement report trigger event, the conventional handoff trigger method is used, when a target cell's received reference signal (RSRP) exceeds a predefined threshold value (A3 event). Furthermore, they used MPLS tunnels for the data plane to replace the GPRS protocol, which has shown to produce high overhead. Two types of handover are conducted in their simulation, the handover between similar switch domain (intra-switch) and the handover between different switch domains (inter-switch). With the assumption of sufficient channel

resource is present to handle the data rate requested by each device. Results conducted on NS-3 simulation have shown that the handover initiation and completion times are nearly stable up to 1 Gbps of data rate per user, which satisfies the control plane delay constraints for 5G networks. In Table 3.15, we present several handover schemes that address mobility management issues in hybrid-based vehicular networks.

3.4 Summary

Mobility management is fundamental for managing and improving the performance of vehicular networks. In the next generation of vehicular networks, several networking paradigms are currently studied to satisfy 5G requirements. However, managing vehicles' mobility throughout different networks is problematic. Without efficient management of vehicles' movement, service disruption and quality of service degradation are inevitable.

Chapter 4

Movement-based Prediction Techniques

In this Chapter, we will study in detail the different vehicles' movement prediction techniques for vehicular networks, which will be later evaluated and tested to fit the mobility prediction schemes. Our focus of study is on stochastic-based predictors: Kalman Filter (KF), Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF) for vehicles' movement. Before that, let us first look into how vehicle's movement is defined for VANETs, and the challenges that could face such prediction.

Various models of location-based prediction for moving objects have been introduced in the literature, such as data mining techniques, which work efficiently in real-time predictions. However, data mining requires the existence of considerable historical data. Another model represented the location by a linear function of time, with measurements of the most recent location and velocity information. However, this type of prediction is limited to the short term, which could lead to the *fork dilemma*, as introduced in [100]. Another problem that could affect the prediction is the uncertainty of the geographical positioning systems, caused either by data or error in measurements.

In vehicular networks, the vehicle state vector X is defined by the set of data (1), which describes the vehicle's movement at time t:

$$[p_x, v_x, a_x; p_y, v_y, a_y] \tag{4.1}$$

where p_x, v_x, a_x correspond to the latitude point, velocity, and acceleration respectively, and p_y, v_y, a_y correspond to the longitude point, velocity, and acceleration of a vehicle. To

measure the change in the state of the vehicle within Δt, we use the kinetic equation of motion:

$$x = x_0 + v\Delta t + \frac{1}{2}\, a^2 \Delta t$$

$$v = v_0 + a\Delta t$$

where x_0 is the initial estimated state of the vehicle, v is the velocity, and a is the acceleration of the vehicle (assuming constant acceleration at time t). In the following subsections, we describe the four prediction models in detail and discuss the advantages and disadvantages of each of them.

4.1 Kalman Filter-based Prediction Model

Kalman filter [107] is a recursive Bayesian estimator that efficiently estimates the state of a linear system, and minimizes the estimated error covariance to reach an optimization.

The Kalman filter works in two steps recursively at each time step: *Time update*, which predicts the next estimation of a current state, and *Measurement update*, which adjusts the current state estimation with actual measurements at time t.

1. Time update (Prediction step):

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k + w_{k-1} \tag{4.2}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{4.3}$$

where w_k represents a normal probability distribution of the process white noise, u is the control input vector (assumed zero as acceleration are considered states), P_k^- is the priori estimate error covariance matrix, and Q is the process noise covariance.

$$x = \begin{bmatrix} p \\ v \\ a \end{bmatrix} A = \begin{bmatrix} 1 & t & 0.5t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} Q = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix}$$

To produce the priori estimated state x_k^- at time interval k, the vector x, which is the previous state of the vehicle, is multiplied by the state transition model A, which is the vehicle motion at time k.

2. Measurement update (Correction step):

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{4.4}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{4.5}$$

$$P_k = (I - K_k H) P_k^- \tag{4.6}$$

where $\hat{x}_k$ is the posteriori state estimation at time step k, P_{k-1} is the posteriori estimate error covariance matrix, H is the measurement equation, R is the measurement noise, which in our work is equal to the GPS measurement noise, and K_n is the Kalman Gain matrix. To update the estimated value of X_k, we use the GPS measurement z_k.

$$R = \begin{bmatrix} R_x & 0 \\ 0 & R_y \end{bmatrix} H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The Kalman filter performance is dependent on the accuracy of the state dynamics and the measurements of the GPS data, which are described as the error in process, measurements Q and R respectively [18]. In our comparison, we treated them in an ad hoc manner by tuning their values.

4.2 Extended Kalman Filter-based prediction model

The EKF approximates a nonlinear equation by linearization of the Taylor series expansion [106]. EKF linearizes the nonlinear system by partial differentiation using the Jacobian matrix to estimate the state of a system. To predict the next state of a system, a priori estimation $\hat{x}_k^-$ and the error covariance P_k^- are calculated as follows:

$$\hat{x}_k^- = f(\hat{x}_{k-1}^-, u_k, w_k) \tag{4.7}$$

$$P_k^- = F_k P_{k-1} F_k^T + W_k Q_t W_k^T$$

where F_k, W_k are the Jacobian matrices of the nonlinear system $f(.)$ and w is the partial differentiation of function $f(.)$ to the process noise (8). The covariance matrix of the state noise is represented by Q_k.

$$F_k = \frac{f(\hat{x}_{k-1}^-, 0, 0)}{x}, \quad W_k = \frac{f(\hat{x}_{k-1}^-, 0, 0)}{w} \tag{4.8}$$

and are initially assumed as

$$F = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} W = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

In the correction step, the Kalman gain K_k is used to calculate the a posteriori estimate of the system dynamics and the error covariance P_k as:

$$K_k = P_k^- H^T (H P_k^- H^T + V_k R V_k^T)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, v_k))$$

$$P_k = (I - K_k H) P_k^-$$

Similarly, the matrix H is the Jacobian matrix of the function h partial differentiation, and V represents the Jacobian matrix of the measurement noise v. The initial error covariance P_0 is set to a large value.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} P_0 = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$$

where $K_k(z_k - h(\hat{x}_k^-, v_k))$ is the innovation equation that calculates the difference between predicted and real measurements.

4.3 Unscented Kalman Filter-based prediction model

The Unscented Kalman filter [105] avoids linearizing around the mean as the EKF by having a set of sigma points that represent the Gaussian random variable of the state vector. The UKF is based on Unscented Transformation (UT) [105]. It uses the nonlinear equation $f(.)$ (8) of the vehicle state vector x with n number of states to compute a set of sigma points matrix X, as follows:

$$\chi_0 = \hat{x}, \quad \lambda = \alpha^2(n + \kappa) - n \tag{4.9}$$

$$\chi^i = \hat{x} - (\sqrt{(n+\lambda)P_x})_i$$

$$\chi^{i+n} = \hat{x} - (\sqrt{(n+\lambda)P_x})_i$$

where $i = 1, .., n$ is the i^{th} column or row of the matrix square root of P and $\hat{x}$. P_x are the mean and covariance of the state vector. The symbol λ is a scaling factor with influencing constants α, κ, indication the distance between the sigma points and the mean, and is usually set to a very small value. They transform each sigma point through the nonlinear function, and compute the Gaussian of their transformation and weight. The weights are calculated as follows:

$$\begin{aligned} w_m^0 &= \frac{\lambda}{n+\lambda} \\ w_c^0 &= w_m^0 + (1 - \alpha^2 + \beta), \qquad \beta = 2 \\ w_m^i &= w_c^i = \frac{1}{2(n+\lambda)}, \qquad i = 1, .., 2n \end{aligned} \tag{4.10}$$

The Unscented Kalman filter model differs from EKF in the prediction by computing both the sigma point mean and the covariance at each time step t. Each sigma point is initiated through the process model, given by

$$\hat{\chi_t} = f(\chi_{t-1}, u_t)$$

The predicted mean $\hat{x_t}^-$ and covariance $\hat{P_t}^-$ are given by

$$\hat{x_t}^- = \sum_{i=0}^{2n} w_m^i \hat{\chi_t^i}$$

$$\hat{P_t}^- = \sum_{i=0}^{2n} w_c^i (\hat{\chi_t^i} - \hat{x_t}^-)(\hat{\chi_t^i} - \hat{x_t}^-)^T + R_t$$

The measurement update equations in the UKF are computed as follows:

$$\hat{Z_t} = h(\hat{X_t}), \quad \hat{z_t} = \sum_{i=0}^{2n} w_m^i \hat{Z_t^i} \tag{4.11}$$

$$S_t = \sum_{i=0}^{2n} w_c^i (\hat{Z_t^i} - \hat{z_t})(\hat{Z_t^i} - \hat{z_t})^T + Q_i$$

$$\hat{P_t^{x,z}} = \sum_{i=0}^{2n} w_c^i (\hat{X_t^i} - \hat{\hat{x_t}})(\hat{Z_t^i} - \hat{z_t})^T$$

where each point $\hat{Z}_t$ is instantiated through the observation model $h(.)$, $\hat{z}_t$ is the predicted observation, S_t is the innovation covariance, and the cross covariance matrix is $\hat{P}_t^{x,z}$. In the final step of updating and correcting the estimated mean and covariance, we use the following equations:

$$K_t = \hat{P}_t^{x,z} S_t^{-1}$$
$$\hat{x}_t = \hat{x}_t{}^{-} + K_t(z_t - \hat{z}_t)$$
$$P_t = \hat{P}_t - K_t S_t K_t^T$$

4.4 Particle Filter-based prediction model

Particle filters are sequential Monte Carlo methods based on the weight representation of probability densities of any given state model. Monte Carlo methods define a general class that converts closed-form statistical quantities to distributed samples, using their average for estimation [202]. The distributed samples are referred to as particles. In the following, we generally describe how the particle filter-based prediction model works.

Initially, we generate a set of random samples $(x_1^i : i = 1, .., N)$ from the probability density function (PDF) $p(x_0)$ and set the weight of each sample to $w_1^i = 1/N$. The equation of the PDF construction is obtained recursively as follows:

$$p(x_k|z_{k-1}) = \int \underbrace{p(x_k|x_{k-1})}_{\text{dynamic model}} \underbrace{p(x_{k-1}|z_{k-1})}_{\text{prior}} dx_{k-1}$$

At each time step k, for each particle, we sample the prior x_{k-1}^i of the PDF using the system model $x_k^i = f_{k-1}(x_{k-1}^i, wk - 1^i)$, and w^i is the PDF of the system noise $p(w)$.

$$p(x_k^i) = p(x_k^i|x_{k-1})$$

Using the measurement y_k, we can form the weight of particles based on a likelihood equation of each prior sample.

$$w_k^{*i} = w_k - 1^i p(z_t|x_t^i)$$

Then, we normalize the weight by

$$w_k^i = \frac{w_k^{*i}}{\sum_{j=1}^N w_k^{*j}}$$

The posterior probability density is then calculated by

$$p(x_{0:k}|z_{1:k}) = \sum_{i=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i)$$

where $\delta(.)$ is the Dirac delta function. After a while, the basic particle filter will suffer from *degeneracy* or *sample impoverishment*. This means, that all but a few particles will have negligible weights. To solve this problem, researchers introduced resampling methods, such as Sampling Importance Resampling (SIR) [83].

The basic idea of the resampling technique is to replace the light-weighted sample with the high-weighted ones. This step can be done when needed, either in predefined time step, or by finding the effective number of particles [133] as

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}$$

In this case, the resampling is performed when the number of effective particles is below the total number of particles. Some of the most commonly used resampling methods include systematic resampling [114], stratified sampling and residual sampling [134].

4.5 Comparison of Movement Prediction Models

In the previous subsections, we have presented four different stochastic-based prediction models, which share the same basic structure of two recursive steps: prediction step and update step. Each prediction model has its own advantages and disadvantages.

The computational cost of each prediction model is one of the main factors for selecting such models. Depending on the importance of speed and available storage in a given system, one can choose the most applicable prediction model. In case of vehicles movements prediction, speed is necessary because of the vehicles' high mobility. The time efficiency of one iteration of Kalman Filter prediction algorithm is in $O(n^{2.376})$, where n is the size of the system state vector $\hat{x}$. As for EKF and UKF, which is of order $O(n^3)$. However, Particles filter is the most computationally expensive algorithm. Where each iteration of the algorithm is in $O(N^n)$, where N is the number of particles and n is the size of the system state vector. Taking in mind, that the number of particles could be very high.

Kalman filter movement-based prediction is the optimal solution for a linear process model, and the lowest in computational cost. However, Kalman filter requires both func-

Table 4.1: Comparison of movement prediction models.

Prediction Model	System Dynamics	Advantages	Disadvantages
KF	Linear	Lower computational rqmts. Optimal if assumptions hold	Does not work on nonlinear systems
EKF	Nonlinear	Linearize the system	Not very stable with bad initial guess
UKF	Nonlinear	Use of Sigma points	Not optimal
PF	Nonlinear	Accurate prediction of N particles	Computationally expensive

tions f and h to be linear with the assumption of Gaussian and white zero mean noise terms. Extended Kalman filter works on nonlinear models, which cover almost all real life processes, by calculating the Jacobian matrix of both f and h functions around the estimated state. However, EKF does not work well in all cases, particularly if the initial guess is incorrect. This was addressed using the Unscented transformation. Hence, rather than using the Jacobian matrix to linearize the functions, a number of sigma points would be used to propagate them through the actual non-linear function. In this case, each sigma point calculates the mean and covariance, yielding more accurate results than EKF.

Finally, Particle filters also work on nonlinear systems with similarity to EKF and UKF consisting of the measurement update and the time update, yet differing through the use of Monte-Carlo methods. Particle filter is considered the most expensive computational technique, and could suffer from *degeneracy* or *sample impoverishment*. This means that all but a few particles will have negligible weights. Table 4.1 shows the differences between the four models.

4.6 Performance Evaluation

The performances of the four different prediction models have been implemented in MATLAB on a Mac computer with 2.4 GHz, Intel Core i7. In our simulation, each vehicle

is equipped with a prediction model and a GPS measurement sensor. For the prediction of the vehicle's movement, we set the state of the vehicle's dynamics to include position (x_k, y_k), velocity v_k and acceleration a_k. In the next subsection, we will first state the simulation model used in our experiments, as well as the performance metrics. We then discuss the results of our implementation, with respect to two different measurement models. The first measurements include position and speed, and the other measurement model integrates the vehicle's heading angle, which describes the vehicle's direction.

4.6.1 Simulation model

We compare the different models using two sets of mobility traces. The first set is a synthetic vehicle mobility trace from Cologne, Germany [212], and the second set is a generated mobility trace using SUMO simulator and OpenStreet Map (OSM) of Ottawa, Canada with the following scenarios: Manhattan-grid, Highway, and Hybrid mobility trace, as shown in Figure 4.1.

In the generated mobility trace of SUMO, a mobility trace of 500 vehicles move randomly with speed ranging between 0-20 m/s, following the road constraints of the Ottawa city maps. Readings were taken every second, and include the vehicle position x and y as well as its speed s at time t. As for the mobility traces from Cologne city, the dataset includes the vehicles' coordinates in meters and speed (m/s). Readings were taken every one second, and the city topology is shown in Figure 4.1(d), which is referred to as a Spider-like model.

4.6.2 Performance metrics

The aforementioned prediction models all estimate and predict the state variables using a set of mathematical equations. We use the following performance metrics to evaluate the accuracy of the prediction models:

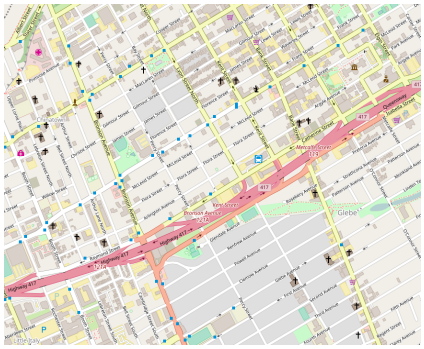
- *Distance Error (DE)* computes the Euclidean distance error between the measured location x_k, y_k and the predicted location $\hat{x}_k, \hat{y}_k$

$$D_{error} = \sqrt{(x_k - \hat{x_k})^2 + (y_k - \hat{y_k})^2} \tag{4.12}$$

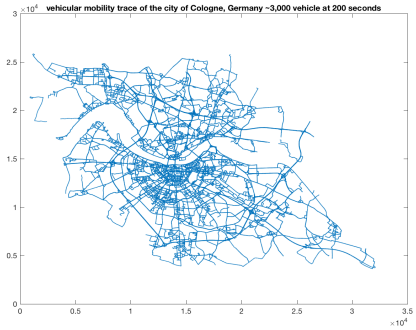
- *Root Mean Square Error (RMSE)* computes the square root of the mean square error between measured and predicted locations.

a: Ottawa City: Highway

b: Ottawa City: Urban



c: Ottawa City: Hybrid



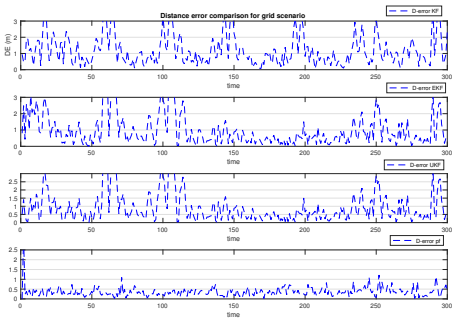
d: Cologne city

Figure 4.1: Mobility traces extracted from OpenStreetMap.

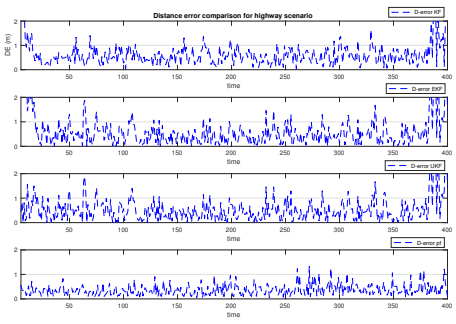
$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_k - \hat{x_k})^2} \tag{4.13}$$

4.6.3 Simulation results

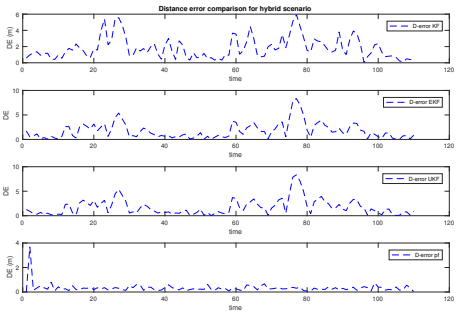
We start by comparing the Distance error between the predicted trajectory and the actual path of a selected set of mobility traces from all scenarios in Figure 4.2(a-c). The results indicate that the distance error in the grid mobility scenario is the worst among the Ottawa mobility scenario because it has many intersections. Hence, in this case the prediction result fails to predict the turns and intersections model. However, the prediction always corrects its estimation by new vehicles' measurement information; thus, the next prediction would be much more accurate.



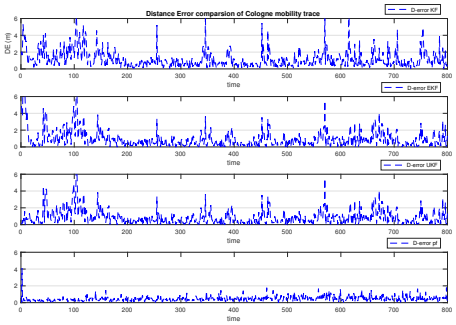
a: Urban



b: Highway



c: Hybrid



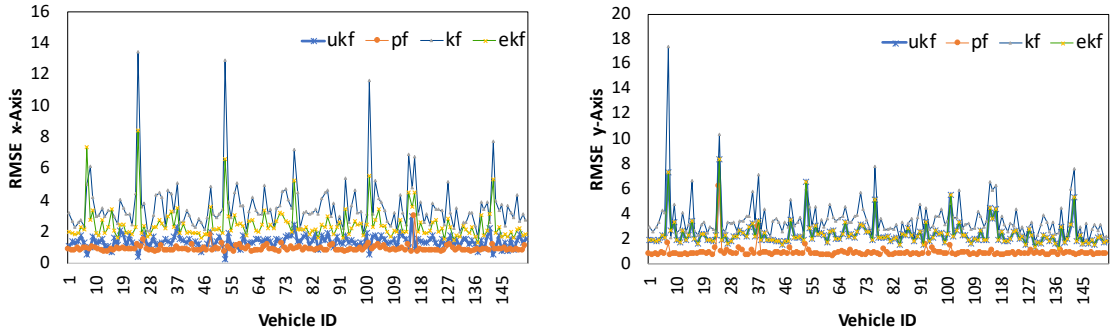
d: Cologne City

Figure 4.2: Distance Error comparison of different Mobility traces .

In the Cologne city mobility trace illustrated by Figure 4.2(d), results indicate similarities to the grid scenario, since the Cologne city map topology is similar to Manhattan-like city. One can also notice that Kalman filter showed the worst results in comparison to the other filters, except in the Highway scenario, which is reasonable considering that the highway mobility moves straight ahead, and the vehicle rarely tends to change its state of motion from its linear state.

Furthermore, the Particle filter showed results with accuracy exceeding 50% compared to the other filters and in all scenarios. This indicates that PF works in almost any situation of nonlinearity. Looking at the hybrid scenario in Figure 4.2(c), the results illustrate ups and downs in the distance error measures, due to the fact that hybrid mobility involves both straight lines of highway cutting the Manhattan-like city.

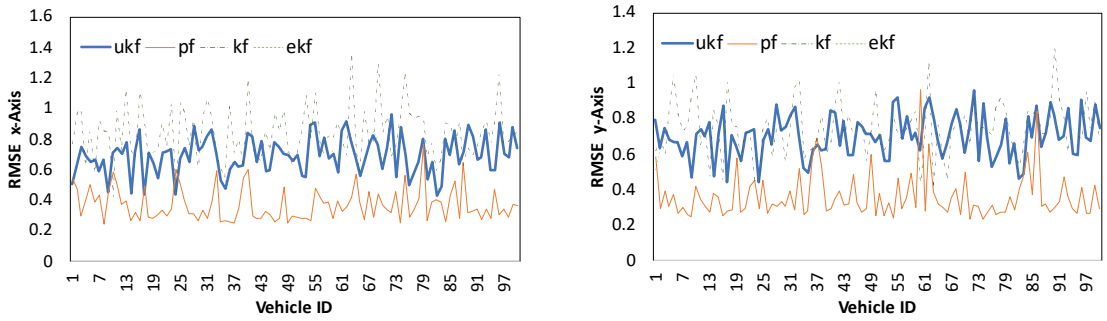
In Figures 4.3–4.6, we show the root mean square error on $RMSE_x$ and $RMSE_y$ for the four prediction-based models, KF, EKF, UKF, and PF. The results confirm that particle filter is superior to the other filters with respect to the RMSE, due to its features in which N number of particles participate in estimating the next location of a vehicle. Each



a: RMSE x

b: RMSE y

Figure 4.3: Cologne mobility trace comparison (sample 200 vehicles).



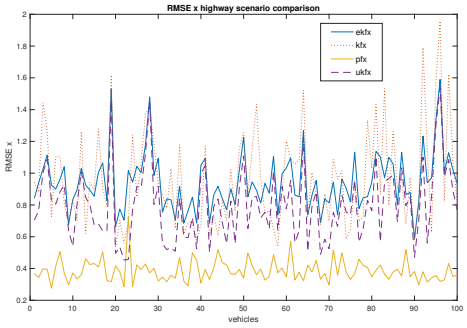
a: $RMSE_x$

b: $RMSE_x$

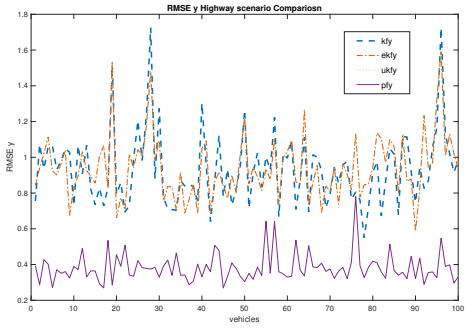
Figure 4.4: Ottawa City urban mobility trace comparison (sample 200 vehicles).

particle will be evaluated to the measurement weight accordingly in process to eliminate the light-weighted ones, and recursively producing new estimations.

We expanded our comparison to evaluate the performance of the prediction models in terms of two different measurements. One measurement used only position information (x, y), while the other used position and the heading angle (θ) of the vehicle, as seen in Figures 4.7–4.8. The results showed that adding the direction of the vehicle into the measurement equation dramatically enhanced the performance of all Kalmans' models. In this case, the prediction would easily correct its estimation with the aid of the new measurements. This leads us to the underlying result that enhancing the performance of any predication model requires the integration of more data. Consequently, the particle filter-based prediction model did not change with the addition of the new measurement data, because PF relied on the weight of predicted particles and neglected the incorrect predictions, rather than the error covariance to correct the prediction.

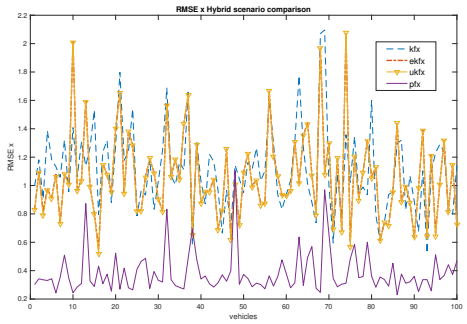


a: $RMSE_x$

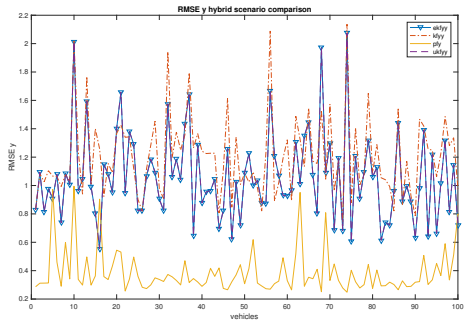


b: $RMSE_x$

Figure 4.5: Ottawa City highway mobility trace comparison.

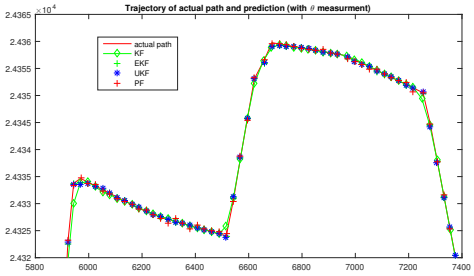


a: $RMSE_x$

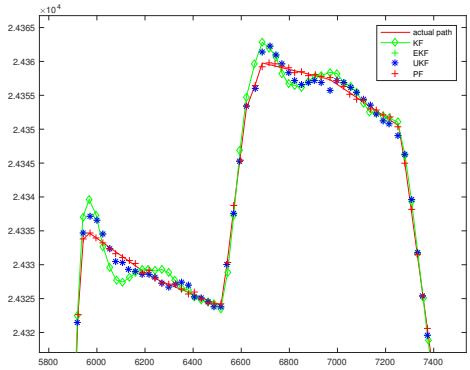


b: $RMSE_x$

Figure 4.6: Ottawa City hybrid mobility trace comparison.



a: including heading angle θ



b: without heading angle θ

Figure 4.7: Trajectory comparison between the predicted and actual path of Cologne mobility model.

measurements / mobility			predictors			
			KF	EKF	UKF	PF
Without direction θ	SUMO	highway	0.9604	0.9405	0.7949	0.3865
		grid	1.1553	1.0434	1.0129	0.3799
		hybrid	1.10925	1.05939	1.05415	0.3852
	Real Mobility	Cologne city	3.6053	2.426	1.3537	0.8646
With direction θ	SUMO	highway	0.42633	0.1814	0.18134	0.38557
		grid	0.46555	0.19114	0.19114	0.3653
		hybrid	0.45598	0.20388	0.20388	0.3871
	Real Mobility	Cologne city	2.4056	1.5239	1.5222	0.8467

Figure 4.8: Comparison of average RMSE with respect to measurements.

4.7 Summary

Vehicular networks have played an increasingly important role in future Intelligent Transportation Systems, providing safety, assistance to drivers, and traffic control management. Because such networks do not need any infrastructure or centralized administration, they can be used for a variety of applications such as traffic control, public safety, and infotainment. However, many problems remain that must be solved before vehicular networks become commonplace. In this section, we focused upon the vehicles' movement-based prediction models for vehicular networks. We have discussed the differences between KF, EKF, UKF, and PF. We have compared these techniques using different sets of mobility models: real mobility traces collected from Cologne city and generated mobility traces of Ottawa city with highway, grid, and hybrid scenarios. Our results indicated that Particle filter outperformed all three techniques with respect to Distance error and Root Mean Square Error.

Chapter 5

Location-based Prediction Techniques: Applications and Use Cases

Vehicular networks have played an increasingly important role in future Intelligent Transportation Systems, as they are useful for providing safety, assisting driving, and managing traffic control. Many vehicular network applications rely on accurate location information. Wireless network management, routing [31] [29], mobility management, service discovery [4] [3], and collision avoidance protocols would improve their performances with more timely information of the vehicle's next location. A number of location prediction-based models has been introduced in tracking and navigation applications for wireless ad hoc networks and wireless sensor networks.

The rapid mobility of vehicles is one of the major characteristics of vehicular networks, affecting the accuracy and performance of location-based predictors. Fortunately, the movement of vehicles is constrained by streets, intersections and roads. Therefore, vehicles' future movements are more predictable. The prediction of vehicles' movements is achieved by estimating their future locations with respect to time. The available prediction models are based either on movement projection, probability modeling, or pattern matching. In movement projection, the current physical movement of the vehicle is used to predict its future location. Probability modeling uses statistical models of previously known movement information to make the prediction. Similar to this idea, pattern matching also uses previously known information, then matches current movement of vehicles to the most relevant gathered data.

Table 5.1: Selective studies on prediction for vehicular networks.

Reference	Application	Technique	Predictors
Krakiwsky et al. [119]	Positioning	Stochastic+history-based	KF, map matching
Feng et al. [64]	Location prediction	Stochastic	KF
Vosselman et al. [215]	Road mapping	Stochastic	KF
Merah et al. [144]	Route prediction	History-based	Sequential pattern
Namboodiri et al. [156]	Routing	Deterministic-based	Link life-time
Jia et al. [101]	Target tracking	History-based	Sequential pattern
Pascale et al. [174]	Traffic flow	Stochastic	Jump markov linear system KF

Many applications designed for vehicular networks would benefit from prediction models to enhance their performance and capabilities in different ways [33] [143]. For example, mobility management, could benefit from prior information of vehicles' locations for seamless registration and handoff to other networks. Also, collision awareness and traffic control systems could use the information to predict traffic jams and traffic management [32], in addition to tracking, routing, and localization issues [30]. To the best of our knowledge, limited work has been completed that aims to understand the insights of movement prediction for vehicular networks. In this Chapter, we are concerned about the vehicles' movement prediction-based models for vehicular networks.

Prediction-based protocols and applications mainly focused on historical-based models and the assumption of linear movement of the vehicles, as seen in Table 9.1. Also, most studies investigated the integration of prediction-based models with routing protocols, mobility, and traffic management applications. Prediction techniques can be divided into three categories: deterministic, history-based, and stochastic models. Deterministic prediction uses the vehicle's kinetics to compute the future position of a vehicle, which is easy to define as a state, but is less accurate and reliable in vehicular movement. History-based prediction learns from repeated movement patterns to predict future positions, which will take time to learn from patterns and requires occasional resets, as movement patterns could change over time. Finally, the main focus of stochastic models is to correct the prediction error using probabilities, which is more appropriate for predicting vehicle's movements.

In this Chapter, we investigate two case studies of movement-based prediction techniques that empower vehicular networks applications including collision detection and Roadside Units (RSUs) localization mechanisms.

5.1 Collision Detection Mechanism

In this Section, we propose an efficient collision detection algorithm that uses the existing wireless communication standards in VANETs to accurately detect potential accidents. We use a movement-based prediction technique to predict the vehicles future trajectories. Therefore, reducing the network communication overhead.

5.1.1 Related work

Existing collision detection and warning systems can be divided into two approaches either using minimum distance to collision or potential risk assessments approach. In the minimum distance to a collision, researchers use the vehicle kinematic or the signal strength to calculate the time of an accident. As for the potential risk assessment, it mostly involves the use of probabilities modeling and analysis.

Wu et al. [221], proposed a method using Support Vector Machine (SVM) for early car accident detection in VANET. Their protocol sends warning messages and suggestions to the drivers to avoid a car crash. In addition, Oh et al. [163], proposed an estimation of rear-end crash using vehicle trajectory data. They estimate the probability of the car is changing lane or going straight using Binary Logistic Regression (BLR) model and trajectory data. Their data are obtained using video sequences of vehicle images. Joerer et al. [104], estimated the intersection collision probability. An accident happens if two given trajectories are in the potential collision area, where they define the size of the potential collision area using the vehicle width. Joerer et al. evaluated the correctness of their model on predefined intersections scenarios. Liebner et al. [130], proposed an additional vehicle's velocity profile at the intersection to determine the driver intent. By comparing the trajectory of the past few seconds with the simulated driver behavior to estimate their posterior probabilities. Liebner et al specified the driver intends to be either go straight, stop at the stop line, turn right or turn right, but stop at the pedestrian crossing.

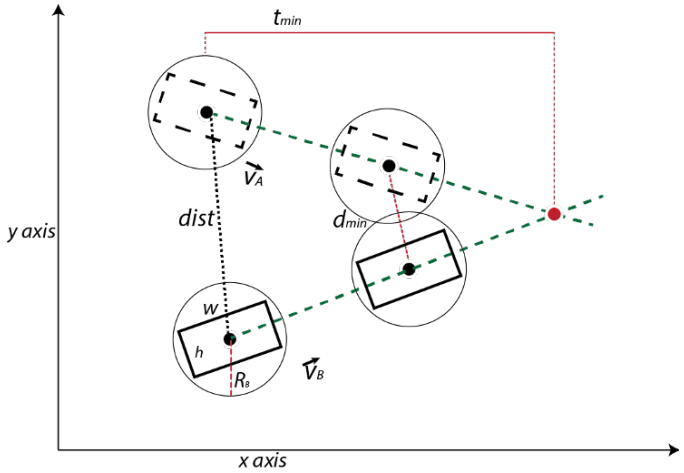


Figure 5.1: Illustrative example of collision detection (Scenario 2).

5.1.2 Collision detection mechanism

Inter-vehicle communication allows vehicles to share and propagate their current position, velocity, and acceleration rate. A collision detection mechanism depends on the vehicle ability to track its own movement and their neighbors over time. In our protocol, we assume that each vehicle is aware of its location using Global Positioning System (GPS) sensor. The accuracy of the location information will affect the performance of the collision detection mechanism, and we will not discuss it here.

Given two vehicles A and B that moves with a defined velocity, which is a physical vector with both magnitude and direction (as seen in Figure 5.1). we can define both vehicles as follows:

$$a = [p_x^A, p_y^A, v_x^A, v_y^A], \; =[p_x^B, p_y^B, v_x^B, v_y^B] \tag{5.1}$$

where p_x^A and p_y^A are the coordinate of vehicle A's initial position. As v_x^A and v_y^A are vehicle A's velocity at time t. Likewise, for Vehicle B coordinates and velocity at time t. We then measure distance between two vehicles with respect to time by:

$$d = \sqrt{d_x^2 + d_y^2} \tag{5.2}$$

$$d_x = |(p_x^A + v_x^A \times t) - (p_x^B + v_x^B \times t)| \tag{5.3}$$

Therefore, the time when two vehicles are the closest can be derived by solving the above distance function for time step t, at the point where the first derivative is equal to zero. Thus, the time of minimum distance is given by:

$$t_{min} = \frac{f_1 + f_2}{{v_x^A}^2 - 2v_x^A v_x^B + {v_x^B}^2 + {v_y^A}^2 - 2v_y^A v_y^B + {v_y^B}^2} \tag{5.4}$$

where

$$f_1 = p_x^A \times v_x^A - v_x^A \times p_x^B - (p_x^A - p_x^B) \times v_x^B$$
$$f_2 = p_y^A \times v_y^A - v_y^A \times p_y^B - (p_y^A - p_y^B) \times v_y^B$$

If t_{min} is positive, then the time of minimum distance between any given two vehicles' positions is in the future. Else, the time to collision have already happened in the past. Therefore, to determine if a collision will occur or not, we check the minimum distance d_{min} at time t_{min} using the following equation, which will be further described in the following section.

$$d_{min} = ((t_{min} \times v_x^A - t_{min} \times v_x^B + p_x^A - p_x^B)^2 + (t_{min} \times v_y^A - t_{min} \times v_y^B + p_y^A - p_y^B)^2)^{1/2} \tag{5.5}$$

Table 5.2: Vehicles' neighbouring list.

id_0	t_0	$(x,y)_0$	$velo_0$	t^0_{min}	d^0_{min}
...	...	...	...	...	...
id_i	t_i	$(x,y)_i$	$velo_i$	t^i_{min}	d^i_{min}

5.1.3 Predictive Collision Detection Method

In this part, we introduce our proposed predictive collision detection method. As vehicles will periodically broadcast their geographic information to their first hop neighbours, each vehicle will maintain a nieghbours list, as seen in Table 5.2. The neighbour list includes neighbour id, time stamp, current position, and speed. In addition to the minimum distance d_{min} and the time of collision t_{min} that will be estimated from the collision detection algorithm.

Algorithm 5.1 Collision detection method.

Input: List of neighbours

$N \leftarrow len(neighbors)$
$i \leftarrow 1$
while $i < N$ **do**
    $t^i_{min} \leftarrow$ min. time to collision
    $d^i_{min} \leftarrow$ min. distance at t_{min}
    if $t_{min} \geq 0$ **then**
        if $d_{min} \leq (R_a + R_b)$ **then**
            collision occurs within t_{min}
            if $t_{min} < \alpha$ **then**
                collision is not avoidable
            end if
            if $\alpha < t_{min} < T$ **then**
                collision is avoidable
                Notify driver
            else
                no collision course,
                no message needed
            end if
        end if
    end if
end while

Basic collision detection mechanism

While vehicles periodically broadcast their geographic information, any vehicle receives a packet from its neighbor will add the vehicle information to its list of neighbors. If the vehicle is already a neighbor, it will update the current information and time stamp. Then initiate the collision detection mechanism (Algorithm 5.1).

In order to determine the time to collision, a threshold α is needed to determine if the accident is avoidable or not. Recent studies [72] have shown that the reaction time of a driver between a collision detection event and the application of brakes varies from $0.7s$ to $1.5s$. Therefore, we define a threshold α as follows

$$\alpha = RT + M_{delay} + R_{Dec} + \gamma \tag{5.6}$$

where , RT is the reactive time of driver and M_{delay} is the communication delay between vehicles, and R_{dec} is the vehicles' rate of declaration.

In Algorithm 5.1, if t_{min} is greater than zero, meaning a collision could occur in the future, we calculate the distance between the two vehicles with respect to both vehicles radius. If the distance is less than or equal to the sum of both vehicles' radius $R_a + R_b$, it will then compare the minimum time with a threshold α, if the minimum time t_{min} greater than or equal to α, which means the two vehicles are in a collision path that could be avoided, a warning message will be sent to both vehicles. Otherwise, the collision is unavoidable.

The aforementioned collision detection mechanism requires continuous communication between the vehicles, which leads to an increase to the communication overhead. Therefore, we introduce a second part to enhance our protocol, by using a prediction model to estimate the vehicles trajectories and reduce the communication packets.

Kalman filter movement-based prediction

Using the described Kalman filter technique in Chapter 4.1, we estimate the next position of the vehicles. Where each vehicle will periodically predict its own state and send their future trajectories along with the neighbouring broadcast message defined in the previous section. To optimize the number of generated packets and reduce the load on the networks, each vehicle will measure the error of its own prediction. If the error is below a threshold β, then it will not send a broadcast for time t. Therefore, the received vehicle will use the predictive measurements to evaluate the minimum time to collision. When the prediction error increases, then new predictive measurements will be sent along with the broadcast message. We defined the error threshold β by using several empirical experiments.

In the following section, we start by evaluating the correctness of the collision detection mechanism through multiple accident scenarios.

5.1.4 Collision scenario's

We simulate our protocol using the network simulator (ns-2) [96]. Where we define the following test scenarios to evaluate the correctness of the collision detection algorithm:

- **Test 1:** Vehicle A and B facing each other. In which, both vehicles have similar or different speed. Figure 5.2 shows the simulated scenario, where our detection

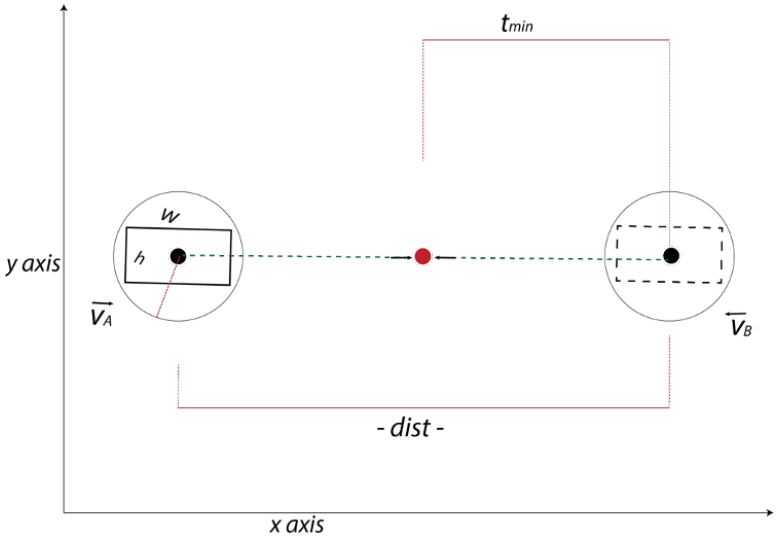
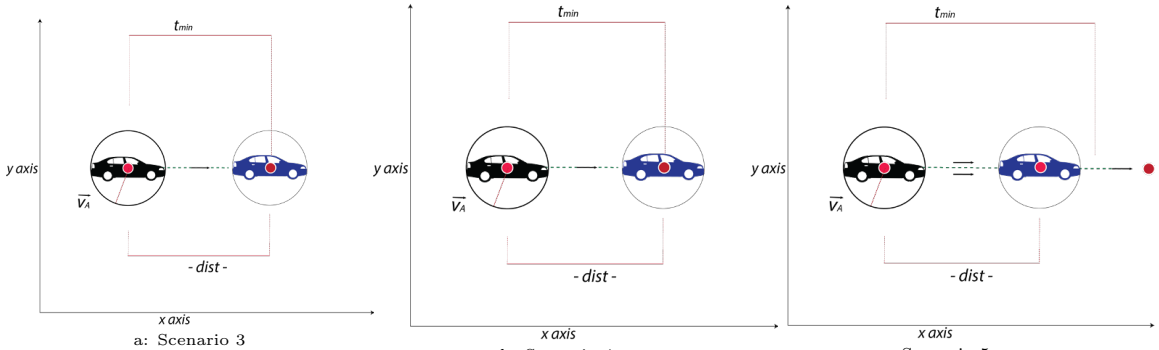


Figure 5.2: Scenario 1



a: Scenario 3 b: Scenario 4 c: Scenario 5

Figure 5.3: Collision scenarios.

algorithm was able to accurately detect the collision point and time to collision in maximum of 20 seconds beforehand when the distance between two vehicles is 200 meters apart. The collision was detected early, due to the defined coverage range of each vehicle, which lead to the discovery of the collision way early.

- **Test 2:** Vehicle A and B moving toward a collision point. In this scenario, both vehicles are moving toward a collision point with similar velocity(as seen in Figure 5.1).

- **Test 3:** One vehicle is moving toward stationary or halt vehicle. In this case, the vehicle is on a collision course and the accident is unavoidable as vehicle direction is

Table 5.3: Simulation parameters.

Parameter	Value
Mobility Simulator	SUMO
simulation duration	$600s$
vehicles	100-400
vehicle speeds	0-30 m/s
road network	Urban
Vehicles' coverage	$100m$
Network simulator	NS-2.35
update interval	$1s$

not changed nor brake is applied. The collision detection algorithm accurately detect the minimum time of collision. Figure 5.3a illustrate the test scenario.

- **Test 4:** Vehicle A and B are moving toward the same target behind each other, and vehicle B is moving at high speed behind. In this scenario, the collision detection method has detected the collision within 20s, when the two vehicles are 200 meters away from each other. Figure 5.3b illustrate the test scenario.
- **Test 5:** Vehicle A is moving toward vehicle B in a near crash path, then vehicle A changes direction. The collision detection algorithm will initially presume both vehicles are colliding in future, if both vehicles stay in same direction and speed as seen in Figure 5.3c. After new measurements are received, the vehicle will correct its detection and overwrite the previous assumption.

5.1.5 Experiments

The previous scenarios have been also simulated with the addition of the KF prediction model. Our results indicate that using predicted states of the vehicles to predict the time of collision is as accurate as using real measurements. In Figure 5.4, we show a comparison between the primary and predictive collision detection protocol in detecting minimum time to collision.

We expand our tests and simulation to evaluate the scalability of the detection algorithm. We setup the vehicular network with 100 to 400 vehicle communicating between each other using IEEE 802.11p standard. Table 5.3 defines the simulation parameters. Also, We choose Ottawa city to map the urban road network, imported in SUMO simulator from OpenStreetMap.

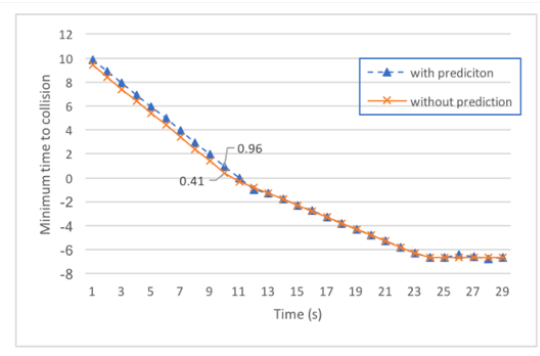


Figure 5.4: Minimum time to collision.

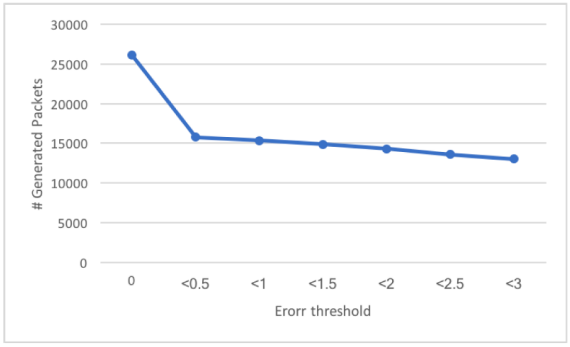


Figure 5.5: Number of generated packets.

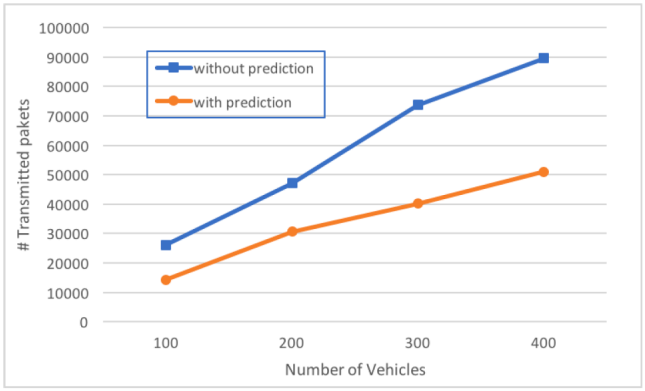


Figure 5.6: Communication overhead comparison.

We start by evaluating the impact of choosing a prediction error threshold on the network overhead in Figure 5.5. The result shows that increasing the threshold will reduce the overhead in transmitted packets. Nevertheless, a very low threshold will lead to a small number of the accurate prediction being used to replace the transmitted packets. Also, a very high threshold will result in the use of inaccurate prediction and thus incorrect collision detections. Thus, in our protocol we choose the value of the prediction error threshold after several empirical experiments. In Figure 5.6, we illustrate the difference in using predictive models to reduce the communication overhead, without losing detection accuracy.

$$acc = \frac{TP}{TP + FP} \tag{5.7}$$

The evaluation of the detection algorithm accuracy (5.1.5) is measured by dividing the simulation results into three categories; true positive (TP), in which the predicted future collision happened. false positive (FP), is when the predicted future collision did not occur. and false negative (FN), is when the real collision was not detected. Based on the simulation parameter in Table II and the addition of different collision scenarios within the vehicular network mobility, we achieve 92.60% accuracy after multiple runs with 10-30-50 actual simulated accidents.

5.1.6 Summary

Road collisions and detection are the focus of many researchers. Accurately detection the time of accident and prevention is a crucial problem. In this Section, we proposed a predictive collision detection protocol that uses a routing protocol to detect potential collision among neighboring vehicles. Our results indicate that with the use of prediction methods to estimate the vehicles future trajectories, has dramatically decreased the overhead of transmitted packets. Nevertheless, our protocol assumes ideal conditions, were drivers are not fatigue or distracted, which is not the case in most today accidents. As a future work direction, we plan to incorporate additional factors such as the driver status as well as the road conditions to detect road accidents.

5.2 Passive Road Side Unit Detection Scheme in Vehicular Networks

In this section, we propose a novel and simple *passive RSU location estimation scheme*, by which the vehicle can pre-determine the desired RSU to communicate based on its own route information. More precisely, by exploring the doppler effects of the received signal of the vehicle, the RSU location estimator is derived by the maximum likelihood estimation method.

5.2.1 Related Work

Passive detection/localization schemes are widely adopted by many applications, such as passive bistatic radar system [245, 247], acoustics sensor network [184], and many others [162, 210]. As well-known, the doppler effect is a common physical phenomenon in the

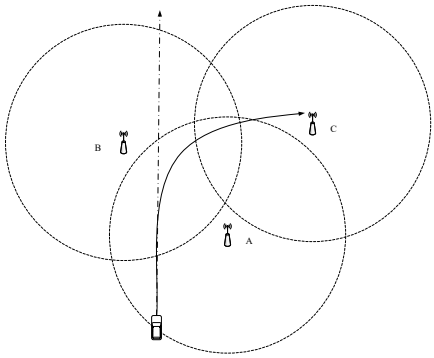


Figure 5.7: Illustration of the uncorrect predicted vehicle moving trajectory.

wireless environment. Hence, it is also an important factor that widely adopted by the passive radar techniques. In [245], by exploring the doppler effect of the wireless signal, the authors designed a maximum likelihood location estimation scheme for the bistatic radar. Similarly, in [49], a cepstrum moving target detection (CEPMTD) algorithm was presented. In this work, the range information and Doppler information of the moving targets were obtained based on taking cepstrum and Fourier analysis of the received signal in the fast and slow time dimensions, respectively. Moreover, some approaches are derived to improve the target detection probability. For example, in [168], the authors proposed a observer moving trajectory design scheme to optimize the target detection probability by maximizing the determinant of the Fisher Information Matrix (FIM).

Moreover, in the wireless sensor networks, the localization methods can be categorized into range-based localization and range-free localization respectively [46]. In our work, we focus on the range based localization. Generally, as summarised in [46, 175], there are four commonly used passive range-based methods, which are: Energy based scheme, Angle of Arrival (AOA) method, Time of Arrival (TOA) method and Time Difference of Arrival (TDOA).

Received Signal Strength (RRS) method is one of the most important energy-based schemes [175]. Based on the power of the received signal and corresponding signal attenuation model, the receivers can estimate the distance to the signal emitter. Then, by taking measurement at multiple locations, the position of the emitter can be estimated. Besides the energy-based methods, many research efforts focus on the localization schemes based on TOA [162, 210] and TDOA [136, 146], respectively. Based on the multiple TOA of the periodic signal that measured by the single receiver , in [210], the authors presented a signal emitter geo-location scheme based on the Maximum Likelihood Estimation (MLE) method. Further, the authors theoretically verified their estimator by deriving the

Cramér-Rao-Lower-Bound (CRLB). While, in [162], the authors proposed the emitter position estimation scheme by exploring the measured TOA of signal that propagated through multiple Non-Line-Of-Sight (NLOS) paths. The authors proved the probability that the fixed single sensor can estimate the emitter's position, with a predetermined waveform and at least five known propagation paths. In [136, 146], the TDOA measurement was adopted to estimate the position of the single- and multiple-acoustic-source, respectively. Moreover, as the complementary of the RSS and TOA methods, the AOA technique, which is most commonly used by the multiple-antenna sensor. The AOA information is calculated based on the difference of the arriving time that a specific signal reaching each antenna of the sensor. Further, by taking multiple measurements in different locations, the position of the signal emitter can be detect. In [179], Peng *et al.* employed the AOA method to address the node localization problem in wireless sensor networks. However, these techniques relay on either the particular beacon/period signals or the complicated equipments (e.g., multi-antenna sensors) which will increase the system complexity and introduce extra cost. Moreover, in these approaches, the doppler effect has to be eliminated by other techniques, which increase the computation complexity.

This section presents a fairly simple solution to detect the position of emitter (i.e., the RSU) by the single vehicle, in which the doppler effect is explored. Moreover, based on the calculated bearing angle, the position of the RSU is estimated based on the Maximum Likelihood Estimator. The details of the problem statement and the formula derivation will be presented in the next two sections.

5.2.2 Problem Statement

In this section, we will present the problem and the corresponding assumptions in details.

As aforementioned, the passive RSU localization problem will be considered. We assume that the vehicle can communicate with the stationary RSU based on a fixed transmission frequency f_t (e.g., 5.9 GHz in 802.11p standard) and known by the vehicle. The vehicle moves with a certain velocity, which is a physical vector with both magnitude and direction. In this case, by exploring the Doppler effects of the frequency shift, the vehicle can measure the bearing angle to the RSU from its trajectory. The relation of the bearing angle θ_i and the its velocity at time t_i is given by,

$$f_r^{(i)} = f_t + f_t \frac{v_r^{(i)} \cos(\theta_i)}{c} \tag{5.8}$$

where $f_r^{(i)}$ denotes the received the frequency by the vehicle at time t_i. The vehicle velocity

at time t_i is represented by $v_r^{(i)}$. c is the signal transmission speed, which is equal to the light speed. Consequently, as shown in Figure 5.8, we can see that the location of the RSU can be obtained by taking the measurements of the bearing angle in different positions of the vehicle's trajectory, which is given by

$$\theta_i = \arccos\left[\frac{(f_r^{(i)} - f_t)c}{f_t v_r^{(i)}}\right] \tag{5.9}$$

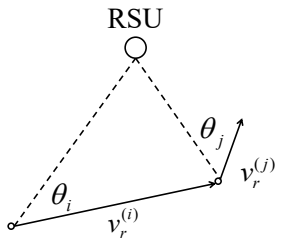


Figure 5.8: Illustrate of the bearing angle to the trajectory.

Moreover, in order to simplify the analysis while without losing generality, we assume that the vehicle and the RSUs are lied on a flat surface.

5.2.3 Mathematical Formulation of the Estimator Design

In this section, we will present the mathematical formulation of the estimator in details.

As aforementioned, we assume that the vehicle and the RSUs are lied on the flat surface. In this case, we can derive the two-dimensional location estimator based on the obtained bearing information that measured by the vehicle. Let $\boldsymbol{T_T}^{(1\times 2)} = [x_T,\ y_T]^T$ represent the position of the base-station in the Cartesian coordinate system, where T denotes transpose. Then, by considering the bearing angle and the measurement error, the $\boldsymbol{T}^{(1\times 2)}$ with these two parameters can be formulated as,

$$\theta_i = F^{(i)}(\boldsymbol{T}) + \epsilon_i \tag{5.10}$$

where ϵ_i is the random measure error at time t_i, and $F^{(i)}(\boldsymbol{T})$ is given by

$$F^{(i)}(\boldsymbol{T}) = \arctan\left(\frac{y_i - y_T}{x_i - x_T}\right) \tag{5.11}$$

Since the unknown position of the base-station/RSU is fixed, we assume that the measurement error $\boldsymbol{\epsilon}$ can be represented by a Gaussian distribution with a zero mean and variance σ^2. Then, we have the conditional probability density function (PDF) as

$$p(\boldsymbol{\theta}|\boldsymbol{T}) = \frac{1}{(2\pi)^{\frac{N}{2}} det^{\frac{1}{2}}(\boldsymbol{C})} \times \exp\left\{-\frac{1}{2}\left[\boldsymbol{\theta} - \boldsymbol{F}(\boldsymbol{T})\right]^T \boldsymbol{C}^{-1}\left[\boldsymbol{\theta} - \boldsymbol{F}(\boldsymbol{T})\right]\right\} \tag{5.12}$$

where $\boldsymbol{C}$ denotes the covariance matrix of the measurement error $\boldsymbol{\epsilon}$ which is a $N \times N$ positive definite matrix. Now, based on the Maximum Likelihood Estimation (MLE) [110], the estimator for the unknown RSU's position $\boldsymbol{T_T}$ can be found by maximizing the log-likelihood function $\ln p(\boldsymbol{\theta}|\boldsymbol{T})$, or equivalent to minimize the following quadratic function

$$J(\boldsymbol{T}) = \left[\boldsymbol{\theta} - \boldsymbol{F}(\boldsymbol{T})\right]^T \boldsymbol{C}^{-1}\left[\boldsymbol{\theta} - \boldsymbol{F}(\boldsymbol{T})\right] \tag{5.13}$$

However, based on Eq. (5.11), we find that the model is nonlinear. In order to simplify the derivation, the linearization is necessary. In order to adopt the Gauss-Markov theorem, a reference position has to be selected. Moreover, the reference position needs to be close to the actual position of the base-station. Hence, we can choose a reference position $\boldsymbol{T_r} = [x_r,\ y_r]^T$ that located within the polygon that bounded by the measured bearing line. Further, we expanse $\boldsymbol{F}(\boldsymbol{T})$ by the first-order Taylor series about the reference position, which is given by

$$\boldsymbol{F}(\boldsymbol{T}) \approx \boldsymbol{F}(\boldsymbol{T_r}) + \boldsymbol{R}(\boldsymbol{T} - \boldsymbol{T_r}) \tag{5.14}$$

where $\boldsymbol{R}$ is the Jacobian matrix of $\boldsymbol{F}(\boldsymbol{T})$ when $\boldsymbol{T} = \boldsymbol{T_r}$ and given by

$$\boldsymbol{R} = \begin{bmatrix} \frac{\partial F^{(1)}}{\partial x_1}\big|_{x=x_r} & \frac{\partial F^{(1)}}{\partial y_1}\big|_{y=y_r} \\ \vdots & \vdots \\ \frac{\partial F^{(N)}}{\partial x_N}\big|_{x=x_r} & \frac{\partial F^{(N)}}{\partial y_N}\big|_{y=y_r} \end{bmatrix} \tag{5.15}$$

Now, by substituting Eq. (5.14) and Eq. (5.15) into Eq. (5.13), we have

$$
\begin{aligned}
J(\boldsymbol{T}) &= [\boldsymbol{\theta} - \boldsymbol{F}(\boldsymbol{T_r}) - \boldsymbol{R}(\boldsymbol{T} - \boldsymbol{T_r})]^T \boldsymbol{C}^{-1} [\boldsymbol{\theta} \\
&\quad - \boldsymbol{F}(\boldsymbol{T_r}) - \boldsymbol{R}(\boldsymbol{T} - \boldsymbol{T_r})] \\
&= (\boldsymbol{H} - \boldsymbol{RT})^T \boldsymbol{C}^{-1} (\boldsymbol{H} - \boldsymbol{RT}) \\
&= \boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}
\end{aligned} \tag{5.16}
$$

where $\boldsymbol{A} = \boldsymbol{H} - \boldsymbol{RT}$ and $\boldsymbol{H} = \boldsymbol{\theta} - \boldsymbol{F}(\boldsymbol{T_r}) + \boldsymbol{RT_r}$ which is not dependent on $\boldsymbol{T}$. Then, we need to derive the estimator $\hat{T_T}$ to minimize $J(\boldsymbol{T})$. Firstly, we need to expanse the above formula as

$$
\begin{aligned}
J(\boldsymbol{T}) &= \boldsymbol{H}^T \boldsymbol{C}^{-1} \boldsymbol{H} - \boldsymbol{T}^T \boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{H} - \boldsymbol{H}^T \boldsymbol{C}^{-1} \boldsymbol{RT} \\
&\quad + \boldsymbol{T}^T \boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{RT}
\end{aligned} \tag{5.17}
$$

Now, by using the method introduced in [110], the gradient of $J(\boldsymbol{T})$ needs to be derived and we have,

$$
\frac{\partial J(\boldsymbol{T})}{\partial \boldsymbol{T}} = -2\boldsymbol{H}^T \boldsymbol{C}^{-1} \boldsymbol{R} + 2\boldsymbol{T}^T \boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{R} \tag{5.18}
$$

Setting the above gradient to be zero, the estimator is given by

$$
\begin{aligned}
\hat{\boldsymbol{T_T}} &= (\boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{R})^{-1} \boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{H} \\
&= \boldsymbol{T_r} + (\boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{R})^{-1} \boldsymbol{R}^T \boldsymbol{C}^{-1} [\boldsymbol{\theta} - \boldsymbol{F}(\boldsymbol{T_r})]
\end{aligned} \tag{5.19}
$$

Then, the covariance matrix of $\hat{\boldsymbol{T_T}}$ can be calculated as

$$
\begin{aligned}
\boldsymbol{C}_{\hat{\boldsymbol{T_T}}} &= E[\hat{\boldsymbol{T_T}} - E(\hat{\boldsymbol{T_T}})][\hat{\boldsymbol{T_T}} - E(\hat{\boldsymbol{T_T}})]^T \\
&= (\boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{R})^{-1} \boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{C} \boldsymbol{C}^{-1} \boldsymbol{R} (\boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{R})^{-1} \\
&= (\boldsymbol{R}^T \boldsymbol{C}^{-1} \boldsymbol{R})^{-1}
\end{aligned} \tag{5.20}
$$

While, if we assume that the vehicle measures the bearing angle N times. For each time, we have

$$\begin{aligned} \sin\theta_r^{(i)} &= \tfrac{y_r - y_i}{d_r^{(i)}} \\ \cos\theta_r^{(i)} &= \tfrac{x_r - x_i}{d_r^{(i)}} \end{aligned} \tag{5.21}$$

where $d_r^{(i)}$ is the distance from the vehicle to the reference position at time t_i. Then, the matrix $\boldsymbol{R}$ is given by

$$\boldsymbol{R} = \begin{bmatrix} -\frac{\sin\theta_r^{(1)}}{d_r^{(1)}} & \frac{\cos\theta_r^{(1)}}{d_r^{(1)}} \\ \vdots & \vdots \\ -\frac{\sin\theta_r^{(N)}}{d_r^{(N)}} & \frac{\cos\theta_r^{(N)}}{d_r^{(N)}} \end{bmatrix} \tag{5.22}$$

Moreover, if we assume that the measurement errors are independent, the covariance matrix can be derived by

$$\boldsymbol{C} = \begin{bmatrix} \sigma_{\theta_1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{\theta_N}^2 \end{bmatrix} \tag{5.23}$$

By substituting Eq. (5.22) and Eq. (5.23) into Eq.(5.19), we can finally derive the position of the base-station as follows,

$$\begin{aligned} \hat{x} = x_r &+ (\sum_{i=1}^{N}\frac{a^2}{c}\sum_{i=1}^{N}\frac{b^2}{c} - \sum_{i=1}^{N}\frac{ab}{c})^{-1} \\ &\times \sum_{i=1}^{N}\frac{(\theta^{(i)} - \theta_r^{(i)})[a(\sum_{i=1}^{N}\frac{ab}{c}) - b\sum_{i=1}^{N}\frac{a^2}{c}]}{d_r^{(i)}\sigma_{\theta_i}^2} \end{aligned} \tag{5.24}$$

$$\begin{aligned} \hat{y} = y_r &+ (\sum_{i=1}^{N}\frac{a^2}{c}\sum_{i=1}^{N}\frac{b^2}{c} - \sum_{i=1}^{N}\frac{ab}{c})^{-1} \\ &\times \sum_{i=1}^{N}\frac{(\theta^{(i)} - \theta_r^{(i)})[a\sum_{i=1}^{N}\frac{b^2}{c} - b(\sum_{i=1}^{N}\frac{ab}{c})]}{d_r^{(i)}\sigma_{\theta_i}^2} \end{aligned} \tag{5.25}$$

where $a = \cos\theta_r^{(i)},\ b = \sin\theta_r^{(i)},\ c = (d_r^{(i)})^2\sigma_{\theta_i}^2$. Now, based on both Eq. 5.24 and Eq. (5.25), we will demonstrate the performance of the presented work.

5.2.4 Verification of Theoretical results

To verify the correctness of the proposed passive RSU detection scheme for the single vehicle, we adopt MATLAB to simulate the proposed work. We set the unit of the distance and the speed (i.e. the magnitude of the velocity) to be *meter* (m) and *kilometer per hour* (KM/H), respectively. Moreover, without losing generality, we assume that the trajectory of the vehicle is a straight line. Hence, we randomly select a position for the RSU. The trajectory of the vehicle is a straight line whose gradient is equal to one. Moreover, we set the test area to be a $400m \times 400m$ rectangle area. The speed of the vehicle varies from $50KM/H$ to $110KM/H$. In the simulation, we test two types of the wireless communication standards, i.e., 802.11p and 802.11 b/g/n, respectively. The 802.11p standard is specifically developed for the vehicular networks with 5.9 GHz frequency. While, the 802.11 b/g/n standards are the mostly used in wireless communication networks with 2.4 GHz frequency. Further, in order to measure the accuracy of the proposed estimator, the notation, Root-Mean-Square-Miss-Distance (RMSMD) [210], is adopted, which can be calculated by,

$$RMSMD \triangleq \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_T)^2 + \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_T)^2} \tag{5.26}$$

where N is the number of the test rounds of the Monte-Carlo method (MCM) [110]. For each specific case, we test 100 rounds. The whole test is repeated 5000 rounds.

Figure 5.9 depicts the RMSMD of the proposed work with 2.4 GHz and 5.9 GHz transmission frequency. We can see that the higher transmission frequency generates better performance. The reason for this phenomenon is that, with the same speed, the doppler effect of the higher transmission frequency is more obvious. With the same relative speed, based on Eq. (5.10), we can see that $\triangle f = f_t \times \frac{v_r\cos(\theta)}{c}$ increases with f_t increasing. Moreover, following the increasing speed of the vehicle, the RMSMD results for both cases decrease, i.e., the accuracy of the estimation increases. Since the doppler effect is more obvious when the relative speed between the RSU and the vehicle is high.

As aforementioned, in order to linearize the estimator, a reference point is essential. The initial measurement is necessary for choosing the reference point. Hence, we test the effect of the number of the initial measurements to the proposed estimator. We set the

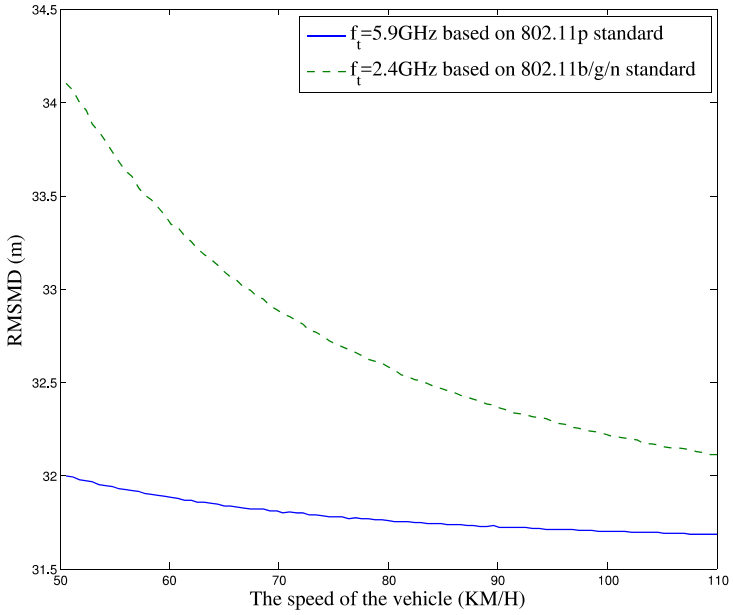


Figure 5.9: The RMSE miss distance with different transmission frequency.

speed of the vehicle to be 110KM/H. The number of initial measurements varies from 2 to 100. We repeats the MCM-based test. As shown in Figure 5.10, we can see that the RMSMD for both 802.11p and 802.11b/g/n achieves relatively stable results, even though the RMSMD in 802.11b/g/n facing higher vibration.

To this end, we can conclude that proposed passive RSU detection scheme is accurate and 802.11p standard is more suitable for the passive detection in Vehicular Networks environment. Meanwhile, we also proved that the number of initial measurements for selecting reference point will not affect the performance of the proposed scheme. Since the proposed estimator is calculated based on the single vehicle, the scheme is fully distributed.

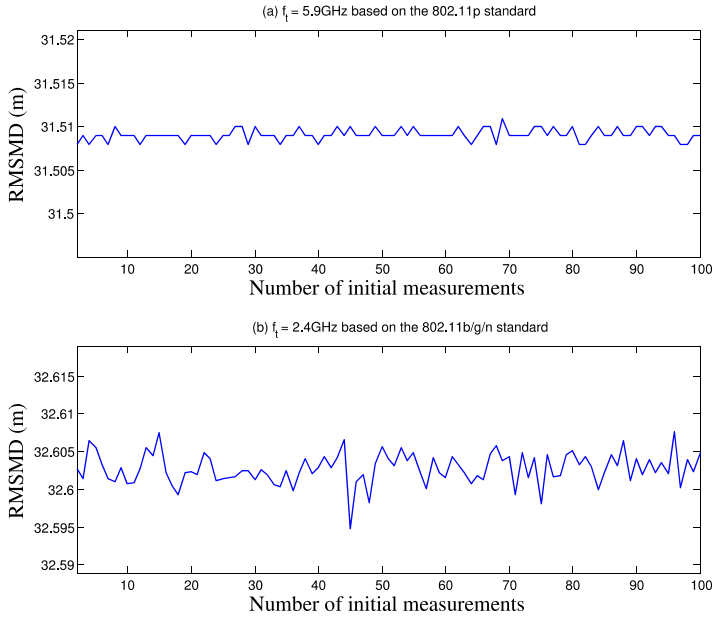


Figure 5.10: The RMSE miss distance with different number of initial measurement.

Chapter 6

Predictive Handover Management Scheme for Intelligent Vehicular Networks

In this Chapter, we present a dynamic MAP discovery and selection protocol for predictive hierarchical mobile IPv6 (DyMAP) that tackles the disadvantages of the Hierarchical Mobile IP protocol (See Section 3.2) in the vehicular environment by decreasing inter and intra-mobility latency and packet drop rate during the handover process, as seen in Figure 6.1. In our approach, by forecasting potential networks access gateways and their respective domains' MAP to initiate a pre-registration, the cost in connections updates and replies can be waived. We use an adaptive Hidden Markov Model (AMulti-HMM) based on the multi-observation model to predict the next feasible access point. The multiple-observation model uses the vehicles mobility measurements through an EKF model. Also, we introduce a dynamic network topology control that uses the vehicles' average session duration time in an AP to adjust the APs' domain selection, thus reducing the average packet drop rate and the handover latency.

The remainder of this Chapter is organized as follows. The used HMM and mobility estimation models are presented in Section 6.1. In Section 6.2, we describe the proposed Dynamic MAP discovery and selection scheme for predictive HMIPv6 handover. The performance evaluation results and discussions are given in Section 6.3, followed by the conclusion in Section 6.4.

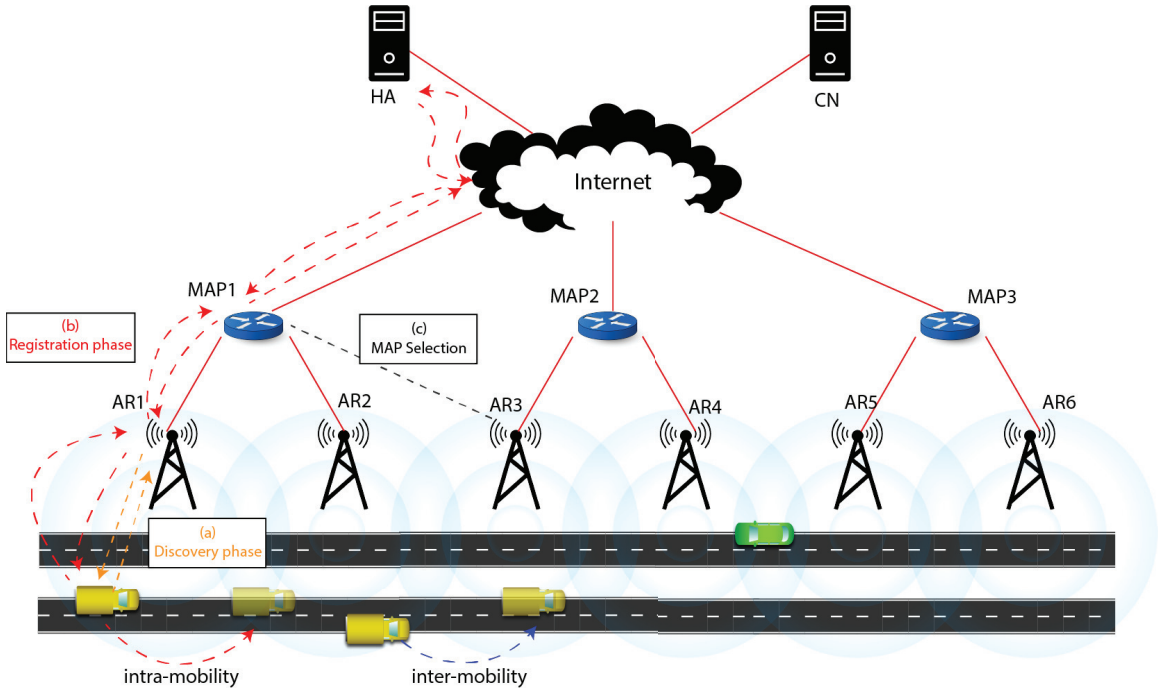


Figure 6.1: An illustrative example of HMIP.

6.1 System Model

We start by describing the mathematical model behind the adaptive multi-observation Hidden Markov Model (AMulti-HMM). Then, we present the characteristics of different observation sequences used within the HMM and present the movement-based Extended Kalman Filter model. Finally, we extend the predictive handover protocol to include a dynamic MAP selection scheme. Assuming a vehicular network with infrastructure-based communication between roadside units and wireless communication between mobile vehicles and access points, vehicles are identified by their location (longitude/latitude) and speed, as they move on the city roads. The deployment of APs is initially predefined to maximize network coverage, where overlapping coverage may occur. This creates a two-layer hierarchical topology of MAPs and APs, which is later defined in the performance evaluation section. Each vehicle is also equipped with a GPS sensor which is used as measurements for EKF to estimate the vehicles' movement projections.

6.1.1 Adaptive Hidden Markov Model

The HMM [129] has been widely utilized in speech recognition applications through the use of a left-right structural model. This model does not permit transitions to states with lower indices than the present state. Another model of HMM, named Ergodic that is fully connected and every state can be in one step reached from any other state. Henceforth, we construct a fully connected HMM model for our protocol, since the mobility of vehicles is unrestricted and can connect to a previously known state.

The HMM model is comprised of three main components [17], namely, the transition probability (transition matrix) A, the observation probability (emission matrix) B, and the initial distribution of states π. An HMM can be described by having N different states: S_1, S_2, ... ,S_N, which are the nearby Access Points (APs). At each time step t=$1, 2, .., T$, a transition occurs to the states, where the actual state at time t is defined as ν_t. That is the vehicle transform from the current AP to a new AP after an observation is measured. The number of distinct observation is represented by $O = \{1, ..M\}$, which define the computed movement attributes of vehicles. In our Predictive Hierarchical MIP, the AP states are hidden, and the vehicles' direction, speed, and location are the observations. Hence, the HMM model can be defined as a 3-Tuple $\lambda = (A, B, \pi)$. So the main elements of the HMM are the following:

- N, which is total number of neighboring APs as HMM states.
- M, which is number of observations variant $O = \{1, ..M\}$ defined by the vehicle movement attributes.
- π, which is the initial states probabilities, and π_i defines one AP's initial probability.

$$\pi_i = P(\nu_1 = S_i), \quad 1 < i < N \tag{6.1}$$

- A, which is state transition matrix, where a_{ij} is the probability of transition between two distinct access points i, j. A transition probability matrix is defined as follows

$$A_{i,j} = \begin{Bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,j} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,j} \end{Bmatrix} \tag{6.2}$$

$$a_{ij} = P(\nu_t = S_j | \nu_{t-1} = S_i), \quad i, j \in N.$$

That is the probability of being connected to access point (ν_t) given the previous connected access point is (ν_{t-1}) and $\sum_{i=1}^{N} a_{ij} = 1$.

- B, which is the observation matrix (emission probability), and b_{jk} is the likelihood of an observation k from a vehicle being connected to AP j. The observation probability matrix is as follows

$$B_{j,k} = \begin{Bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,k} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{j,1} & b_{j,2} & \cdots & b_{j,k} \end{Bmatrix} \tag{6.3}$$

$$b_{jk} = P(z_t = O_k | \nu_t = S_j), \quad j \in N, k \in M.$$

That is the probability of having an observation of the vehicles' state z_t given that the vehicle is connected to access point (ν_t) and $\sum_{j=1}^{N} b_{jk} = 1$.

We can train and estimate the probability values based on the network observations, using the HMM model $\lambda = (A, B, \pi)$ mentioned above. This can be done in two phases: adaptive learning and initial matrix re-estimation. The adaptive learning process, adopted from the method proposed by [153], is used to recursively estimate the AP probabilities, thus relying only on previous time steps to update the observation matrix. The AP probability can be described in two steps; using the initial distribution $\pi = \{\alpha_0(i)\}$ (3), we derive the AP probabilities (4) at $t = 1$, and the AP probability at time t by function (5).

$$\alpha_1(i) = \pi_i . b_i(o_1) . a_i(1) \tag{6.4}$$

$$\alpha_t(i) = \frac{\alpha_{t-1}(i) . a_{t-1}(ap_{t-1,i}) . b_{t-1}(i, z_t)}{\sum_i^N \alpha_{t-1}(i) . a_{t-1}(\nu_{t-1,i}) . b_{t-1}(i, z_t)} \tag{6.5}$$

and the APs' initial value is as follows

$$\pi_i = \frac{e(i) . \delta(i - \nu_a)}{\sum_{j=1}^{N} \pi_j} \tag{6.6}$$

where $\delta(.)$ is the Kronecker delta and determined by the argument value, if the value is zero then result will be 1 and zero otherwise. Since the state of the observation is usually visible after the handover is completed, a weight value $e(i)$ is used instead of the likelihood calculation, where the weight of the ith packet is determined by the following.

$$e(i) = \begin{cases} 1, & \text{if } i = 0. \\ \frac{1}{e(i-1)} & \text{if } 1 \leq i \leq L. \end{cases} \tag{6.7}$$

where L is the maximum sequential number, to ensure that the weight does not diverge to zero. Then, the transition and emission matrices are calculated as follows

$$a_{ij} = \frac{a_{ij} + \delta(i - \nu_p).\delta(j - \nu_a).e(i)}{e(i) + \sum_{n=1}^{N} a_{in}} \tag{6.8}$$

$$b_{jk} = \frac{b_{jk} + \delta(k - z_c).\delta(j - \nu_a).e(i)}{e(i) + \sum_{m=1}^{N} b_{im}} \tag{6.9}$$

where ν_p is the predicted AP, ν_a is the actual connected AP, and z_c is the measured observation at the time of prediction.

6.1.2 HMM Multiple-Observation Scheme

In this section, we define the HMM multiple observation sequences O with regards to several mobility measurements of the vehicle. The measurements will be thoroughly evaluated in our simulation experiments (see Section 6.3) to fully understand the impact of different vehicles' characteristics on the prediction model. Three attributes that can be extracted from the vehicles' sensing systems are used: vehicles' directional angle, speed, and location. In order to calculate the vehicles' directional angle θ, we define the direction of a given vehicle as follows.

$$\begin{aligned} dir_x &= x_p - x_c \\ dir_y &= y_p - y_c \\ \theta &= atan2(dir_y, dir_x) \end{aligned} \tag{6.10}$$

where x_p, y_p are the next estimated trajectory locations of a vehicle. Whereas $atan2(y, x)$ is the four-quadrant inverse tangent function that returns the angle between the positive axis of a plane and the point given by (x, y). After defining the directional angle of the vehicle,

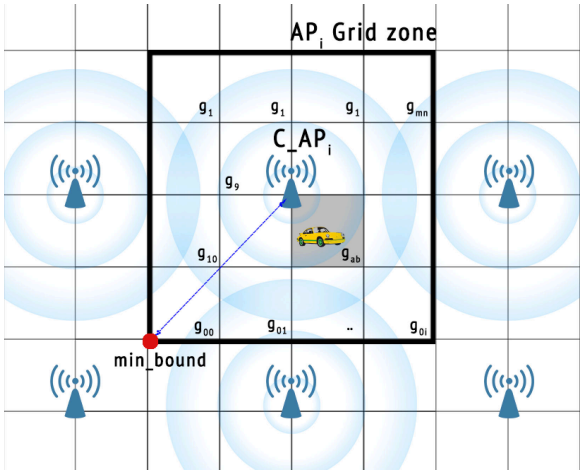


Figure 6.2: Determining vehicles' grid location within AP coverage zone.

we discretize the data into features of equal width, in order to transfer continuous values into discrete counterparts. Thereby, we set θ to M_1 distinct values for each direction. We consider eight different directions: N, NE, E, SE, S, SW, W, NW. Concerning the vehicles' speed s, derived from the EKF procedure, we convert the measured rate into a discrete set of data for the observation model. The vehicles' speed will be categorized into M_2 sets of distinct speed ranges, as described in Table 6.1. The limit of the categories M_2 depends on the average maximum speed exhibited by any vehicle. In our model, we assume the vehicles' speed does not exceed 30 m/s.

Table 6.1: Discretization of vehicles' speed.

S_t	speed range (m/s)	status
1	$s = 0$	halt
2	$0 < s \leqslant 3$	very slow speed
3	$3 < s \leqslant 6$	slow speed
..	..	..
l	$21 < s \leqslant 24$	average speed
..	..	..
M_2	$i < s \leqslant j$	maximum vehicles' speed

Finally, we choose the vehicles' location as the third observation state that will be used in the HMM model. The vehicles' location is represented by longitude and latitude, which is given from the on-board GPS system and estimated by the EKF model. Each vehicle calculates the distance between itself and the lower bound of the current connected AP

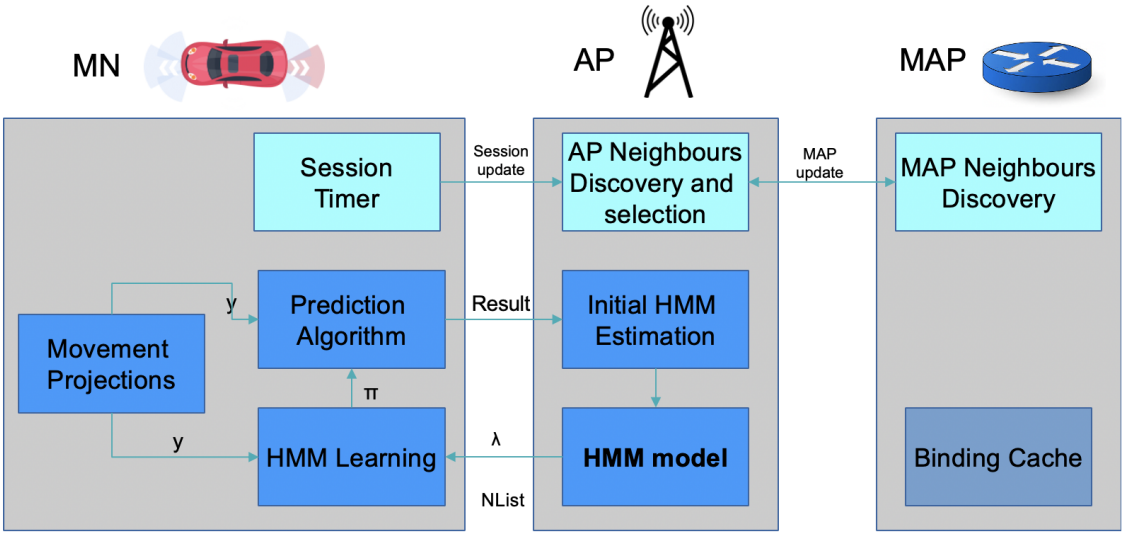


Figure 6.3: Dynamic predictive-based hierarchical mobile IP (DyMAP) protocol.

coverage area. Thereby, the location of vehicles is mapped to the grid number on the map, as seen in Figure 6.2. If each grid is of size $M_3 \times M_3 m$, and the total possible number of grids in one AP is defined by the maximum communication range of the Access point.

In the context of vehicular networks, the vehicles' direction, speed, and location do not necessarily determine the individual probability of connecting to the next AP. The movement of vehicles differs depending on the road structure, topology, and driver intentions. Two vehicles have similar directions, and yet connect to different APs. In another example, two vehicles in an intersection may have similar speeds and yet exhibit different directions, or vice-versa. The uncertainty in the vehicles' movement will affect the prediction of the next possible AP. Henceforth, we consider a multiple observation model, which combines vehicles' direction, speed, and location trajectories, to construct a better observation of the vehicles' current status.

6.2 DyMAP: Dynamic MAP Discovery and Selection Scheme for Predictive HMIPv6

In this section, we start by describing the predictive hierarchical handover protocol by using the previously acquired HMM and mobility measurement model. As seen in Figure 6.3, the P-HMIP handover has three primary components: MAPs, APs, and MHs. The MAP

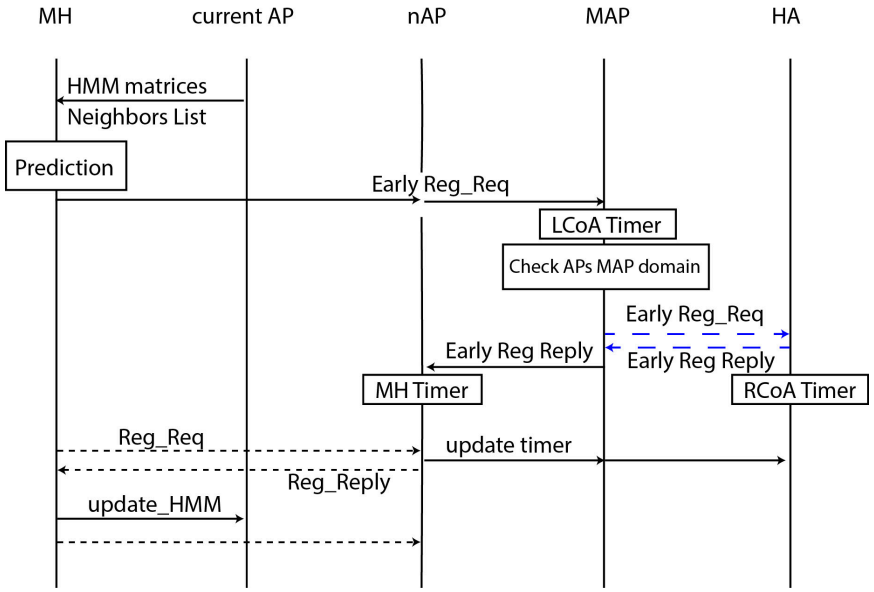


Figure 6.4: Predictive hierarchical mobile IP process.

is the link between the MH and its HA and comprises several APs under its domain; each AP separately handles the initial HMM measurements and retains a table of adjacent APs. The MH will use the EKF periodically to track and predict the movements of the vehicles and will be responsible for deriving a prediction using the HMM elements from the APs. Initially, an equal number of APs are assigned to each MAP in the network, which will then be dynamically updated according to the mobility and handoff process of the vehicles.

In the following, we divide the handover process into three main phases, *prediction and registration* phase, that monitors the vehicles' movement projections and initiates an early handover process when a new AP prediction is made by the Multi-HMM model. The *MAP discovery and update* phase, which allows the Access Points to discover neighboring APs and their corresponding MAPs as well as update the Multi-HMM model initial estimation with previous predictions results. In Figure 6.4, we illustrate the predictive HMIP handover process. Finally, *dynamic MAP selection* phase, which allows APs to adaptively switch between MAPs according to the reported vehicles' mobility sessions; thus, reducing the cost of inter-mobility handover in high mobility areas.

Algorithm 6.1 Prediction algorithm

Input: EKF $\hat{x}_t, \hat{x}_{t-1}$
Input: λ, $list_N$,O
1: **procedure** PREDICTION
2: **while** $timer_p > t$ **do**
3: $O_t \leftarrow MovementProjection(\hat{x}_t, \hat{x}_{t-1})$
4: $p \leftarrow HmmEstimation(O_t, \lambda)$
5: **if** $AP_p \in list_N$ **then** ▷ AP is in list
6: $EarlyRegReq(AP_p, MAP_p)$
7: $timer_p(cancel)$ ▷ prediction timer
8: **else**
9: $timer_p(update)$
10: **end if**
11: **end while**
12: **end procedure**

6.2.1 Prediction and registration phase

A copy of the latest update of the HMM model, including the initial, transition, and emission matrices, is sent to the vehicle when it first transitions to a new access point in a MAP domain. Also, vehicles will also have a list of discovered neighboring access points $List_N$, as presented in Figure 6.5. From that moment, the vehicle will periodically trigger the prediction method to predict the initial APs' probabilities using the measured mobility projections of the vehicle. The prediction method in Algorithm 6.1 works in two steps: First, retrieve the vehicles' observed movement projection from the EKF model. Second, discretize the calculated directional angle, speed, and grid location, for the observation symbol O. The observed state will then be fed to the HMM estimation method to yield an AP estimation vector of the initial states in the HMM model, as presented in Algorithm 6.2.

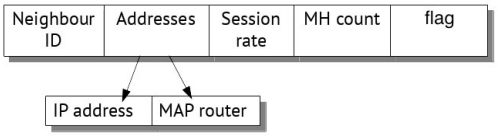


Figure 6.5: Neighbors list.

When an AP probability reaches a certain threshold α that marks enough learning has been made and an estimation is present, an early registration request packet $Early_{Reg_req}$

Algorithm 6.2 HMM estimation method

Input: λ , O, N, M
Output: O is within the range of observation symbols
Output: $N > 0$
procedure HMMEstimation
for all $\alpha_i(t)$ **do**
Update APs' Probability - Equations (4)(5)
end for
if maximum$(\alpha_i(t)) > \theta$ **then** ▷ threshold θ
return i_{max} ▷ Prediction is reached
else
return $N + 1$
end if
end procedure

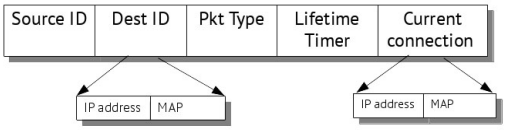


Figure 6.6: Early registration request packet.

is sent to the new AP corresponding MAP through the current MAP infrastructure. The early registration packet in Figure 6.6 contains the vehicles' current AP, current MAP, the predicted AP, and its domain MAP, and a lifetime timer. If the domain of the predicted AP is the same as the current domain, a simple registration request Reg_{req} will be sent to the new AP and a lifetime timer $Timer_k$ is set to overcome false prediction. Otherwise, if the MAP of the predicted access point and current MAP domain are different, then the new MAP adds the vehicle information in its cache, and an early registration request is forwarded to the vehicles' Home Address (HA). Also, the predicted AP with a lifetime timer is set in case of false prediction. When the vehicle enters the new AP communication range, a registration packet is transferred to the target AP. No extra messages are sent to the MAP nor the HA, when the target AP is already predicted, thereby completing the registration phase. An update message is sent to the new MAP and HA to cancel the lifetime timer $Timer_k$ and notify the previous MAP to stop the communication with the MH.

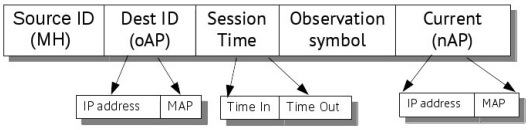


Figure 6.7: Prediction update packet.

6.2.2 MAP discovery and update phase

Following the vehicle's connection to a new AP after a prediction is formed, the lifetime $Timer_t$ of the new connected MAP is updated, and the MAP sends an update packet to the MNs' corresponding HA to update its time as well. Another update packet $Predupdate_{AP}$ is also sent back to the previous AP from the MH to remove the vehicles' information and update the previous AP HMM elements and session rate. In case of failure of prediction, a standard registration process will be used as the HMIP handover protocol [21], and an update packet will be sent as well for learning.

The update packet in Figure 6.7 is sent through the new domain MAP to the previous one, which contains the new registered AP, predicted AP, and the observation symbol. At the reception of an update packet, the AP will first check if the newly registered AP is already in its neighboring list $List_N$. If not, it will add a new AP neighbor to its list and update the HMM model using equations (6,8, and 9). Since the number of neighboring APs are limited in our protocol N, when a new AP is discovered, the neighboring AP with the lowest probability will be dropped and replaced with the new one.

During the process of handoffs and location updates, each MH will also keep track of their session duration in the current AP. When a vehicle first connects to an AP (oAP), it marks the session entry time, and after moving away from the oAP to a new AP (nAP), the MH records the session end time. Along with the prediction update packet in Figure 6.7, the MH includes the session in and out time. Upon receiving the update packet from the MH, the AP will either update or include a new AP session rate. Each AP maintains a session rate with information about its neighbors. An AP_i record in the $list_N$ table in Figure 6.5 holds the average session duration rate $Srate_i$ of the current AP before an MH moves to the new AP_i and the number of handoffs between the current AP and the new AP $MHcount_i$. The algorithm used to handle the incoming update packet is described in Algorithm 6.3.

Algorithm 6.3 Update method

Input: $update_{packet}\{AP_p, AP_a, O, in_s, out_s\}$
Input: $list_N, \lambda$
for all $update_{packet}$ **do**
  $j \leftarrow list.indexOf(AP_p)$
  if $AP_a \in list_N$ **then** ▷ Neighbor's data is available
    $i \leftarrow list.indexOf(AP_a)$
    $\lambda \leftarrow HMMupdate(\lambda, i, j, O)$
    $list_i.SessionRate \leftarrow SessionRate + (out_s - in_s)$
    $list_i.MHCount \leftarrow list_i.MHCount + 1$
    if $list_{indx}.flag = Pending$ **and** $list_{indx}.map = myMap$ **then**
      $list_{indx}.flag \leftarrow complete$ ▷ nAP changed MAP
    end if
  else
    $list \leftarrow list \cup AP_a$
    $\lambda \leftarrow HMMupdate(\lambda, N, j, O)$
    $list_N.SessionRate \leftarrow (out_s - in_s)$
    $list_N.MHCount \leftarrow 1$
    $N \leftarrow N + 1$
  end if
end for

6.2.3 Dynamic MAP selection phase

Initially, the hierarchical structure of the mobile IPv6 is predefined to a specific structure such that each MAP maintains a group of APs, which does not take into account the vehicles mobility and road structures. When the HMIP protocol starts, each AP will advertise the domain MAP attached to, while discovering its neighborhood. Periodically, each AP checks its neighbor table $list_N$ to estimate whether a change of domains is required. When a session rate between current AP (cAP) and a neighbor access point (nAP) from a different domain MAP (nMAP) is less than a threshold κ, a change of domain request is sent to the current APs' domain (cMAP) with the neighbors' access point address nAP and $nMAP$. The current AP neighbor list will flag the nAP as pending for change of domain until a new update message is received from an MH. The cMAP then forward the transition of domain request to the nAP through the network. Upon receiving the change of domain request, the nAP will start using the cMAP domain as the point of attachment and will advertise a MAP option in its area. The dynamic MAP selection method is described in Algorithm 6.4. In case of an ongoing active session in the nAP, the

Algorithm 6.4 Dynamic MAP selection method

Input: $list_N, timer_t, N$
for all $timer_t$ **do**
$min \leftarrow list_0.sRate$
$indx \leftarrow -1$
for i=*1* ***to*** N **do**
if $list_i.sRate < min$ **then**
$indx \leftarrow i$
$min \leftarrow list_i.sRate$
end if
end for
if $min \leq \kappa$ **and** $list_{indx}.map \neq mymap$ **then**
if $list_{indx}.flag \neq pending$ **then**
$list_{indx}.flag \leftarrow pending$
$SendDomainUpdate(list_{indx}, mymap)$
end if
end if
$timer_t(update)$
end for

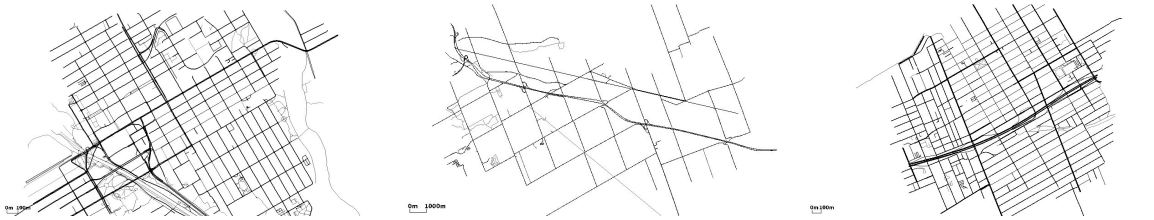
change of domain will be valid only for the new upcoming sessions. When the registration phase starts again by any incoming MH, the nAP and oAP neighbor list will be updated accordingly. To better estimate the need for domain changes, the AP does not initiate the domain's update method unless two or more neighbors are discovered and enough sessions are reached to derive an estimation.

6.3 Performance Evaluation

The performance evaluation of the proposed (DyMAP) handover protocol is done through the use of NS-2 network simulator. First, we illustrate the used simulation parameters to analyse the produced results. Then, we define our experiment's mobility and network setup and state the evaluation matrices. Second, we evaluate the impact of different mobility traces and vehicles densities over the multiple HMM observation symbol, including single observations (speed, location, directional angle) and multiple observations. Finally, we evaluate the handover latency and packet drop rate of our dynamic predictive handover protocol in comparison to different handover management protocols.

Table 6.2: Simulation parameters

Parameter	Value
Mobility Simulator	SUMO
simulation duration	500s
PHY/MAC	IEEE 802.11p
vehicles	100-400
vehicle speeds	0-30 m/s
road network	Urban - highway
AP coverage	200m
Network simulator	NS-2.35



a: Ottawa city: Urban b: Ottawa city: Highway c: Ottawa city: Hybrid

Figure 6.8: Road network topologies

Figure 6.9: Shenzhen City

6.3.1 Simulation environment

We generate the mobility trace using SUMO simulator [118] and OpenStreet Map (OSM) [79] of the City of Ottawa with different environments, as seen in Figure 6.8. We deploy the APs in each map to guarantee full coverage, where 802.11p protocol is used in the MAC layer. In Table 8.1, we define the simulation environment parameters. Different mobility traces are used to evaluate the performance of the proposed protocol and are as follows:

Synthetic Data:

In this set of data, we generated Highway, Urban, and a combination of both road environments. All road data sets represent Ottawa city and imported from OpenStreetMap. While SUMO traffic generated is used for the vehicles' flow and routes, as seen in Figure 6.8. The number of cars ranges between 50-500, as for the vehicles' movement trajectories, we use the Krauss car-following model with vehicle speed ranges between 0-30 m/s.

Real Data:

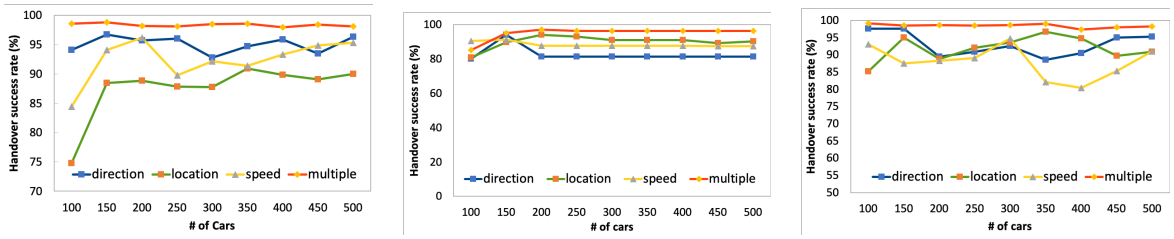
We also choose a real data-trace of Shenzhen city, China [40] as seen in Figure 6.9. The original trace contains about 13,000 taxis mobility traces on average of 30 seconds per day for nine days. For the purpose of our experiments, we reduce the data to 30 minutes between 6:00-6:30 am. The taxis' speeds between 0-70 m/s. We deploy around 800 roadside units on an area of 8x24 k/m with a coverage area of 200 m for each RSU.

The performance of the predictive HMIP is evaluated in terms of scalability and efficiency using the following performance metrics: i) *handover latency*, which computes the average time of handover between APs; ii) *packet drop rate*, which computes the total number of dropped packets over the total number of sent packets; iii) *prediction success rate*, which is the ratio of successful AP prediction over the total number of successes and failures. Before evaluating our protocol against different handover protocols, we study the handover prediction accuracy in term of success and failure using different observational elements.

6.3.2 Prediction accuracy

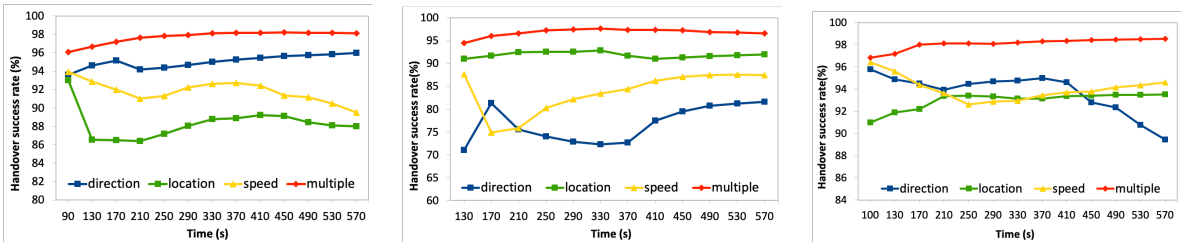
The EKF prediction error has been already evaluated by comparing its performance over different scenarios in Chapter 4, which indicates that the EKF model is resilience to different vehicles movements over the three scenarios. Furthermore, the EKF performed the best in highway scenario, followed by hybrid and then grid scenarios. This is due to the fact that in the grid mobility trace, vehicles experience more frequent turns in movement, which leads to higher prediction error. Such an issue can be addressed if more measurements are included in the EKF model, such as the heading angle of the vehicle (See Chapter 4). Therefore, using EKF to predict and update GPS measurements will provide the HMM model with a more accurate estimation of the vehicles' movement projections rather than raw GPS inputs.

The handover prediction accuracy is evaluated by comparing all different vehicles' observation against different scenarios and densities with 95 % confidence intervals, as illustrated in Figure 6.10. The multiple observation symbol outperforms every other single



a: Hybrid b: Highway c: Urban

Figure 6.10: Different prediction observation models accuracies vs traffic density: (a) Hybrid environment, (b) Highway environment, (c) Urban environment.



a: Hybrid b: Highway c: Urban

Figure 6.11: Different prediction observation models accuracies vs time in Ottawa city environments.

observation in all situations with an average of 98% success in urban and hybrid environments and 94% success in the highway environment. Also, the individual observations showed inconsistency and instability along with the different environments, as seen in Figure 6.10(a-c). This occurred because the direction, speed, and location of a vehicle do not independently determine the probability of connecting to the next AP, as vehicles tend to exhibit similar movement projections and yet choose different access points. When a combination of multiple observations is adapted, an increase in the number of successful estimations is observed.

Additionally, it is noted that the performance of the different observations varies with varying environments of road. In the hybrid environment, the direction observation outperforms speed and location. The characteristic of the hybrid environment combines both urban and highway, and thus the location parameter does not efficiently describe the observed state of a vehicle. For example, two vehicles may be geographically in the same grid and yet present different directions, thereby leading to a varied AP selection. As for the highway scenario, a more stable and relatively closer success rate is observed. This is due to the highway environment omitting sudden turns or changes in the vehicle's movement. In the urban environment, shown in Figure 6.10(c), a more changeable prediction is exhibited in single observations and higher accuracy with the multiple observation model.

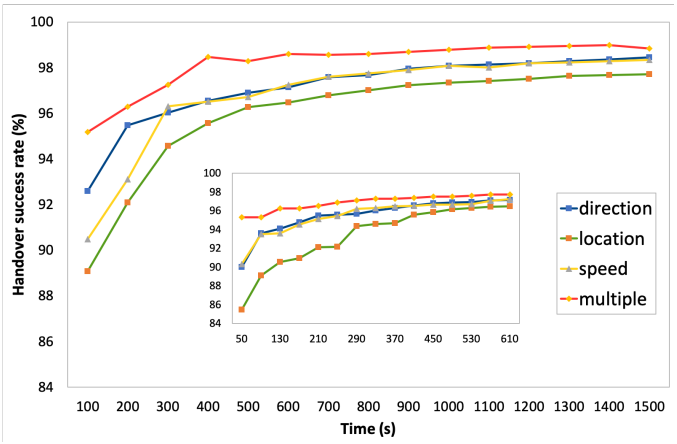


Figure 6.12: Handover prediction accuracies vs time in Shenzhen City mobility trace.

The progress of the prediction model over time shows apparent inconsistency among different environments. In Figure 6.11(a-c), we illustrate the handover success rate versus the simulation time for the various observations. Therein, the single observation demonstrates instability along the timeline, due to vehicles presenting similar behaviors in individual observations and not following the same connections with APs along their path. In Figure 6.12, we illustrate the prediction success rate over time in Shenzhen city mobility trace. The multiple observations model showed the highest handover success rate compared to single observation models, with an average of 98% success rate. Furthermore, the use of real mobility traces for a more extended time showed that all observations eventually converged. Furthermore, it has been noted that the chosen real dataset presented higher similarity measurements in term of the vehicles' direction and speed than the generated mobility traces in SUMO, which explains the stable results of the single observation models.

We further investigate the accuracy of the multiple observation (Multi-HMM) in comparison with other prediction models, i.e., Naive Bayes (NB) [11] and k-Nearest Neighbor (k-NN) [52] prediction methods. NB is a simple probabilistic classifier based on the Bayes theorem. The k-NN is a classification technique that assigns each instance to the most common class out of its k nearest neighbors. To make the comparison of models relatively equal, we derive an online version of the NB and k-NN models. In which, both networks will re-train their models periodically to mimic a real-time prediction scenario.

In Figure 6.13, we illustrate the prediction accuracies of different models in Ottawa's urban trace. As seen, the Multi-HMM achieved the highest average of 97% accuracy rate compared to NB and k-NN. The performance of k-NN gets better over time, and NB accuracy maintains an average of 91%. One of the disadvantages of k-NN is its need to store

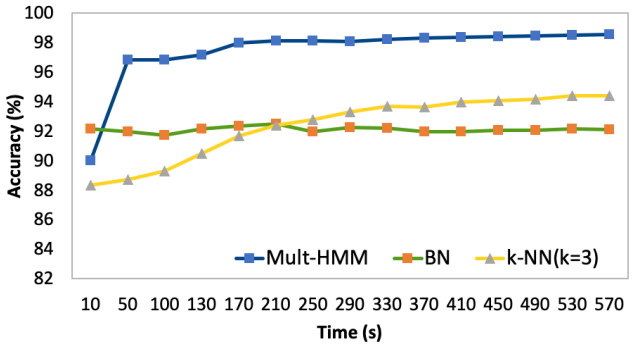


Figure 6.13: Handover prediction accuracies comparison in Ottawa Urban mobility trace.

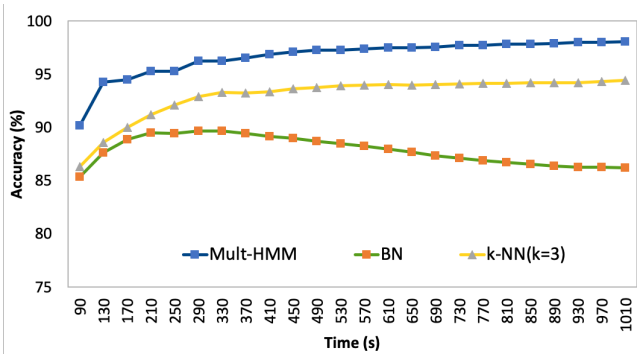


Figure 6.14: Handover prediction accuracies comparison in Shenzhen city mobility trace.

all of the training data. However, a Multi-HMM model maintains only three probability matrices. Besides, BN predicts an AP class for a set of features to an observation, but it does not consider the dependencies between observations. An example of such reliance would be a vehicle slowly increasing its speed or direction, which might yield a pattern leading to different APs. This problem appears more obvious in Shenzhen city mobility trace comparison in Figure 6.14. In Figure 6.14, NB illustrates the worst prediction accuracy with an average of 85%, while Mult-HMM showed the highest percentage of 96% prediction accuracy.

6.3.3 Network performance

In regards to the efficiency of our proposed predictive handover protocol, we evaluate the network performance with respect to latency and packet drop rate. We use different traffic densities and environments, as described earlier in this section. The benchmarks handover

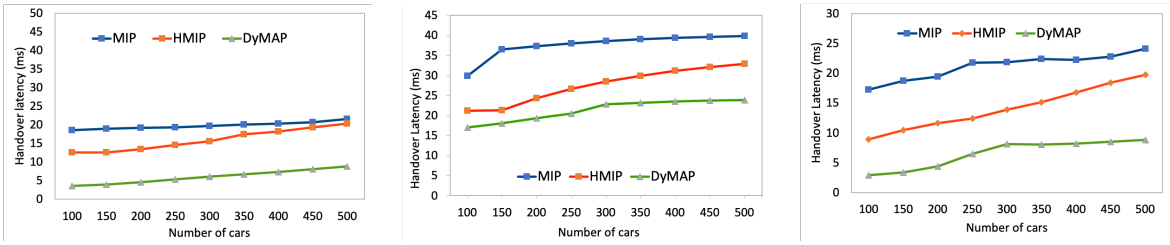


Figure 6.15: Different mobile IP protocols' handover latencies vs the number of vehicles: (a) Hybrid environment, (b) Highway environment, and (c) Urban environment.

methods MIP and HMIP are implemented and evaluated for comparison.

In Figure 6.15, we illustrate the average successful handover latency versus the number of vehicles. The predictive HMIP outperforms all protocols, with approximately 50% reduction in handover delay. Our protocol predicts the new AP early enough to start the registration process before the vehicle enters the new AP range. The other protocols begin the registration process when the vehicle approaches the communication range of the new AP and L2 trigger is initiated, thus increasing the handover process delay. In Figure 6.15, the hybrid, and urban scenarios presented similar handover latency averages for each protocol. This is due to the distributed nature of the generated vehicles mobility in both scenarios, which reduces the load on the APs during the handover registration process. As for the highway scenario, the generated vehicles mobility were condensed in a specific area, which is Ottawa highway road, this has led to an increase in APs' load and thereby an increase in handover latency. Furthermore, It is noted that in the hybrid environment, both MIP and HMIP have similar results with respect to the handover latency. In our understanding of the network performance, this is due to the randomized topology structure of the hierarchical network. There, the placement and the selection of the domains' area did not reflect vehicles' mobility and the road structure. The result of that led us to incorporate a dynamic MAP discovery and selection scheme to the predictive HMIP, which is later explained in the next subsection.

In Figure 6.16, we show the performance of all handover protocols in a real mobility trace from Shenzhen city. The predictive hierarchical mobile IP handover protocol reduces the handover latency by 80% compared to MIP and HMIP. The advantage of predicting the next possible connection early enough contributes significantly to reducing the handover latency, particularly in inter-mobility handoffs. Even though an incorrect prediction could yield a higher latency, it will not exceed the normal rate of the standard HMIP and does not affect the overall performance of the protocol. The predicted access point has a reasonably high accuracy rate with the use of multiple observation HMM. One of the main features of

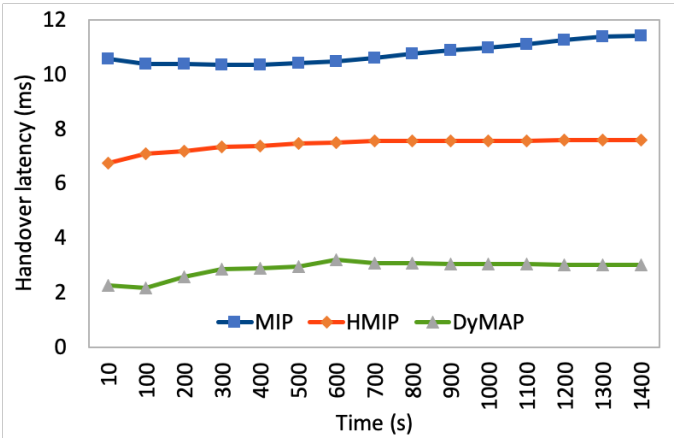


Figure 6.16: Handover latency comparison in Shenzhen city mobility trace.

hierarchical mobile IP is the addition of Mobile Anchor Points (MAPs), which act as the vehicle's RCoA and manage multiple APs in its domain. This reduces the cost of contacting the vehicles' HA for registration and binding updates when a vehicle moves between APs within the same domain. That process guarantees a smooth transition between local APs, thus resulting in lower packet drop rate.

6.3.4 Dynamic MAP selection performance

Having a predictive HMIP handover protocol does not necessarily exclude false predictions, which reduces the performance of the handover protocol when it happens. The prediction failure will lead to the use of the underlying standard HMIP handover protocol to re-connect with the Mobile IP network, and hence, the system will suffer again of the same drawbacks explored in the standard HMIP. Therefore, a solution is needed to overcome the disadvantages of prediction failure.

To fully comprehend the influence of the network topology selection and deployment on the performance of the DyMAP handover protocol, we evaluate two different approaches: i) a randomized *static* domain distribution of N MAPs over K APs, in which we assign b equal number of APs to each MAP; ii) a *dynamic* MAP selection scheme based on the average session duration of inter-mobility handover between two neighboring APs.

Before presenting the results of the dynamic predictive HMIP, we have empirically validated the most acceptable threshold values that result in higher measurement accuracy. As we mentioned in previous sections, the MHs compute their session duration time in the

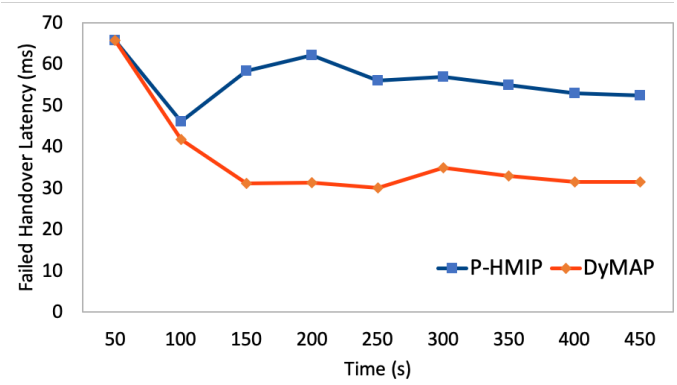


Figure 6.17: Average handover latency in urban environment.

old connected AP after moving to a new AP. The value of the session duration is sent back the old AP to calculate the average session duration time between itself and the newly connected AP. To find the required threshold value for an AP to request a change of domain, a range of values are tested in the simulation environments. In which, the best value of κ is chosen to minimize the overhead, handover latency, and packet drop rate.

The simulation setup illustrated in Figure 1 is used to demonstrate the impact of the dynamic MAP selection scheme when the predictive HMIP fail to estimate the correct new handoff. The communication links of the simulated network are fixed to 50 ms for the communication between the MAPs and infrastructure (i.e., Internet) and 2 ms for the communication links between MAPs, APs and MHs.

In Figure 6.17, we illustrate the average handover latency over time of both the static and dynamic predictive HMIP. The results indicate that after the selection of MAPs is changed in the network, a reduction in the handover latency is observed. The dynamic schema reduces the handover latency by 32% in compared to the static predictive HMIP. This is due to the dynamic change of MAP selection after an AP has collected sessions data from the MHs about its neighboring APs. When the network topology is first selected and deployed, it does not consider the vehicles' mobility factor in the selection of the network topology, hence resulting in poor performance of the hierarchical mobile IP. If the average duration time between two APs of different domains is short, a higher packet drop rate will be exhibited. The reason is that the inter-mobility handover process requires more time to finish the registration with the MHs' home address.

6.4 Final Remark

Internet protocol (IP) compatibility development for vehicular networks has been recently investigated in the literature; however, the vehicles' high mobility and topology changes affect the performance of traditional mobility management protocols over VANETs. In this section, a predictive hierarchical mobile IP handover protocol is proposed for vehicular networks, that overcomes the drawbacks in HMIP protocol. The probability analysis of adaptive HMM and the vehicles' movement projections are combined to predict the subsequent handover process ahead of time accurately. We investigated the impact of the vehicles' movement characteristics in terms of the prediction accuracy under different environments and densities. The results indicate that the early prediction of the next connection enhances the overall mobile network performance in terms of latency and packet loss. Also, the effect of the domains' topology selection on the performance of the hierarchical mobile IP was studied for vehicular networks. The results show that a mobility-aware network topology influences the mobile IP performance in terms of handover latency, cost, and packet drop rate.

Chapter 7

A Predictive Handoff Trigger Scheme for Intelligent Vehicular Networks

in this Chapter we propose a two-tier machine learning-based handover management scheme that detects the need for a handover trigger ahead of time, and initiates an early handover registration process to avoid any communication disruption. With our scheme, the vehicle will take the decision of the handover time, based on a machine learning model, to predict the received signal strength values. When a handover trigger is initiated, a hidden Markov model-based AP selection strategy is implemented in a hierarchical-based mobile IP (as described in Chapter 6). We evaluate the performance of various forecasting models for time-series sequence prediction, which will be then incorporated in the handover trigger scheme and AP selection.

The remainder of this Chapter is organized as follows: Section 7.1 defines the system model and preliminaries. The proposed triggering protocol is described in Section 7.2. In Section 7.3, numerical analysis and simulation results of the proposed scheme are presented. We conclude our discussion and future directions in Section 7.4.

7.1 Problem Statement and Preliminaries

Assuming a region of the plane with l APs $AP_1, AP_2, ...AP_l$ with fixed known locations (x_i, y_i) for $i = 1, .., l$. The AP broadcasts ads about their location and service to vehicles within the transmission range. Based on the received broadcast packet, the vehicle can evaluate the received signal strength of a given packet at time t. To measure the power

present in a received radio signal, a propagation model is needed to predict the average received signal strength at a given distance from the AP transmitter. Furthermore, due to the presence of buildings and an indirect line-of-sight path between the vehicle and an AP, a path loss factor will be added to the signal strength. The literature [186] has shown that the average received signal power decreases logarithmically with distance. Thus, the received signal strength indicator (RSSI) can be estimated using the following equation:

$$RSSI = dBm_0 - 10nlog_{10}\frac{d}{d_0} + \epsilon \tag{7.1}$$

where dbm_0 is the received signal strength at 1 metre, d is the distance and n is the propagation constant or path loss exponent [60]. Being able to obtain the vehicles' RSSI values over time can indicate whether or not a handover is needed, if it is, a handover trigger is initiated. Each vehicle in our model will maintain a list of RSSI values $\{rssi_t, rssi_{t+1}, ..., rssi_{t+L}\}$ over window time L with the current AP connection. Current and previous RSSI values can be used to estimate the next k possible signal strength, which allows the vehicle to anticipate the need for handover early enough to discover and register new CoA. To predict the next RSSI values sequence, a forecasting model is required. Many time-series forecasting techniques are available in the literature [140]. Recurrent Neural Networks (RNNs) is a well-known method in machine learning for time-series sequence forecasting [147], which can retain state from one iteration to the next by using its own output as input for the next step. However, the simple recurrent network suffers from its inability to capture long-term dependencies in a sequence.

The Long-Short-Term Memory (LSTM) network [87] is a recurrent neural network (RNN) architecture that is popularly used for time-series sequence prediction. LSTM was designed to overcome the exponentially decaying error and vanishing problems [87] in recurrent learning, using an efficient gradient-based learning algorithm. In general, LSTM network tries to take advantage of information in the past time, in which the output depends on the current input and previous inputs. A detailed description of the model can be found in [87].

The standard RNN network architecture consists of input layer, hidden layer, and output layer. The input layer contains the input vector $\{rssi_t\}$ at time t, which is propagated to the hidden layer. The hidden layer updates its hidden layers status $h_t^1, ...h_t^k$ at time t and computes the output y_t based on the input $\{rssi_t\}$. To update the hidden layer states

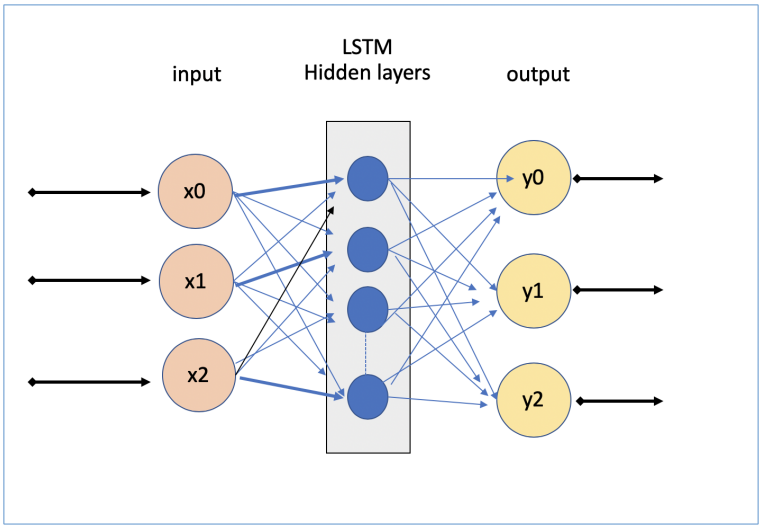


Figure 7.1: LSTM Network architecture.

in typical RNN architecture, we use the following equations:

$$\begin{aligned} h_t^1 &= f(w_{ih_t^1} x_t + w_{h_{t-1}^1} h_{t-1}^1 + \dot{b}_t^1) \\ h_t^k &= f(w_{h_t^{k-1}} h_t^k + w_{h_{t-1}^k} h_{t-1}^k + \dot{b}_t^k) \end{aligned} \tag{7.2}$$

where f is a nonlinear hidden layer function that is set as a sigmoid function, w defines the weight matrix, and $\dot{b}$ is the bais vector. The hidden layer in an LSTM network also includes the memory cells and gate units, which changes the f function in the general RNN model in order to store information. The memory cell would define some linear interactions to decide what to keep in memory and output the hidden state h_t. Figure 7.1 describes the general structure of an LSTM network.

The proposed ML-based handover trigger scheme is divided into three parts: data preprocessing, learning, and prediction. We assume that vehicles will be able to maintain the sequence of received power strength indicator (RSSI) from the current AP update packets. Using the provided RSSI sequences as input vector $rssi_t$, we train our LSTM model in an offline manner and use the saved model parameter to predict a new RSSI sequence. Furthermore, while collecting the RSSI values for training, we consider an input vector $rssi_t$ with window size L, which is the size of one RSSI sequence vector required for prediction. The input vector window size is predefined in our model, and has been tested with different sizes to better fit the handover time and network delay.

7.2 LSTM-based Handoff Trigger Scheme

In this section, we describe the proposed ML-based handover management protocol using the previously derived LSTM model. The proposed handover management scheme uses the Hierarchical MIPv6 protocol as a baseline [199] and consists of three main phases: *handover trigger*, *AP selection*, and *update*. In the first phase, we use the LSTM network to estimate the next k sequence of RSSI values periodically, and use those values to decide whether on not a switch is needed according to a predefined threshold. If a decision is made to initiate the AP selection phase, the HMM presented in our previous Chapter 6 will be used with the vehicles' current measurements to predict the next Access Point. Otherwise, no prediction will be required. In the following, we describe each phase in our handover management scheme.

When a vehicle first connects to a new AP in a MAP domain, it also receives a copy of the current adaptive HMM elements $\lambda = (A, B, \tau)$ and a list of discovered neighbors $List_N$. Then, the vehicle initiates the handover trigger algorithm to estimate when to switch using the observed vehicles' RSSI data. The handover trigger method in Algorithm 7.1 works in two steps. First, the vehicles' observed RSSI value is collected and the handover trigger procedure is started. While the number of observed RSSI does not reach L window size, the vehicle will keep updating the RSSI values with the current CoA. Otherwise, the $LSTMPredict$ method, Algorithm 7.2, will be called and given the sequence of data to forecast the next k values. If the average value of the new sequence is below a threshold $thresh$ and values are left-skewed, then the handover process is triggered to find a new CoA. We assume that the LSTM network will be trained off-line, and each vehicle will receive a copy of the learned LSTM network to forecast RSSI values.

When a new AP connection is requested by a vehicle in the next phase of the AP selection phase, it will calculate its current direction, speed and location with respect to the current AP, and include it with the observation symbol O. The observed state will be used in the HMM estimation method to yield an AP estimation vector of the initial states in the HMM model, as presented in Chapter 6. When an AP probability reaches a certain threshold α which marks that enough learning is constructed, and estimation is present, an early registration request packet $Early_{Reg_req}$ is sent to the new APs' domain through the current MAP infrastructure. The early registration packet contains the vehicle's current AP, current MAP, the predicted AP, and its corresponding MAP, and the observed state. If the predicted AP domain is the same as the currently connected domain (i.e., intra-mobility case), no HA binding update process is required, and the new MAP will forward the early registration packet to the new predicted AP. Otherwise, if the predicted MAP domain is different from the current MAP domain (i.e., inter-mobility case), then the new MAP adds

Algorithm 7.1 Vehicles' Handover Trigger

Input: $Seq[T]$,$thresh$, $\lambda(A, B, \pi)$, $list_N$,O
Input: $updatePacket$ from Coa
procedure RSSI SEQUENCE TRIGGER
while $timer_p > t$ **do**
$RSSI$.add(GetRSSI($updatePacket$))
$Seq.append(RSSI_i)$
if Seq.IsFull() **then** ▷ L window sequence is collected
$\hat{rssi}$= LSTMPredict(Seq) ▷ Return list $T + k$ RSSI
else
$time_p.update()$
return ▷ Not enough RSSI data
end if
if $AvgSkewed(\hat{rssi}) < thresh$ **then**
$p \leftarrow HmmEstimation(O_t, \lambda)$
if $AP_p \in list_N$ **then** ▷ Predicted AP is in list
$EarlyRegReq(AP_p, MAP_p)$
else
$RegularRegistration()$ ▷ Register to the nearest AP
end if
$Seq \leftarrow Empty$ ▷ Reset sequence
$timer_p.cancel()$
end if
$Seq \leftarrow Seq[1:]$ ▷ Shift sequence and remove the first value
$time_p.update()$
return
end while
end procedure

Algorithm 7.2 LSTM prediction method

Input: $Seq\{rssi_t\}$,$LSTMmodel$
procedure LSTMPREDICT
$\hat{y} = LSTMmodel.predict(Seq\{rssi_t\})$
return $\hat{y}$
end procedure

the vehicle information in its cache, and an early registration request is forwarded to the vehicles' HA, with a lifetime timer set in case of false prediction. When the vehicle enters the new AP coverage, a registration message will be sent to the new AP. If the new AP was already predicted, no more messages are sent to the corresponding MAP nor to the HA, and the registration process will be completed. An update message is sent to the MAP and HA to cancel the lifetime timer $Timer_k$. In the event that, the HMM prediction method did not return a predicted AP, the vehicle will initiate an early registration using its own AP ads cache.

Following the vehicle attachment to a new AP and MAP, after a prediction is formed, an update packet $update_{AR}$ will be sent back to the previous AP from the MH to remove the vehicle's information and update the previous AP HMM parameters. The update packet is sent through the new registered MAP to the previous MAP, which contains the new registered AR, predicted AP and the observation state. The previous AP will then update its current neighbor's list, and update the HMM model with the new information. We assume that each AP will not have more than N neighbors close by.

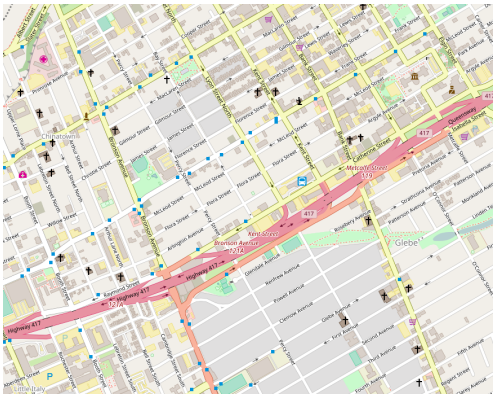
7.3 Performance Evaluation

In this section, we evaluate the performance of our handover management scheme with respect to prediction accuracy, handover latency, throughput, and dropped packets rate of the LSTM model and the proposed handover trigger scheme. We simulated our network using the network simulator NS-2. The road data of the city of Ottawa in Figure 7.2 is imported using OpenStreetMap [79], while, the traffic flow and vehicles' routes are generated by SUMO [118]. Access points are deployed within a 200 m communication range that maximizes the map area coverage. The network parameters are shown in Table 7.1.

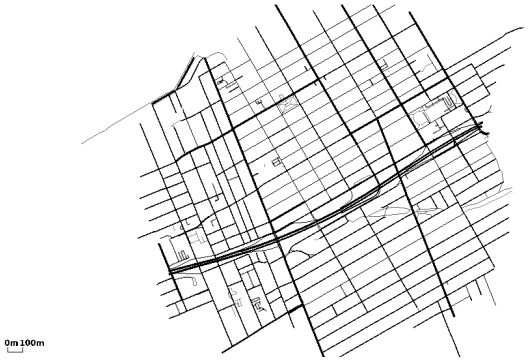
Before using the LSTM network model, we pre-process the RSSI dataset and rescale it to fit the prediction model better. In the following sections, we describe the data preprocessing methodology, the LSTM model accuracy, and network performance.

7.3.1 Data Preprocessing

In our simulations, the RSSI dataset was collected during a simulation with S interval time. We scale and fit the data using MinMax scaler, train the network during the first period of the simulation, and test the network afterwards. A sample of a vehicle's measurements



a: OpenStreetMap



b: SUMO generated map

Figure 7.2: Ottawa city road map

Table 7.1: Simulation parameters

Parameter	Value
Mobility Simulator	SUMO
simulation duration	500s
PHY/MAC	IEEE 802.11p
vehicles	400
vehicle speeds	0-30 m/s
road network	Ottawa Urban
AP coverage	200 m
propagation model	log-normal
Wireless sensitivity	-85 dBm
Network simulator	NS-2.35

is presented in Figure 7.3. The top peaks of the data in Figure 7.3 indicate a handover has been performed, and the vehicle does not have RSSI data at this point.

7.3.2 Prediction Accuracy Evaluation

To evaluate the performance of the LSTM model, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are employed. We define the error measurements as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(\hat{y}_i - y_i)| \tag{7.3}$$

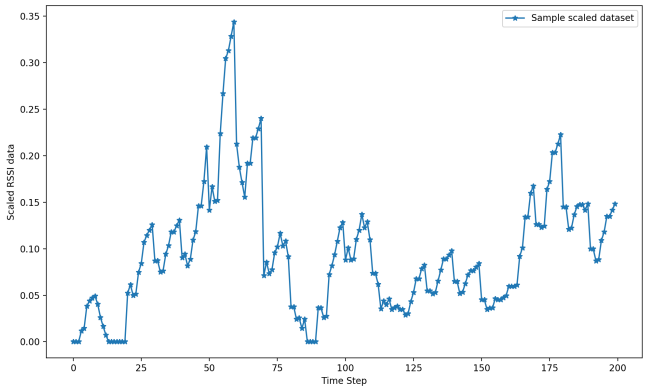


Figure 7.3: Scaled RSSI sequence sample of vehicle V_x.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|(\hat{y}_i - y_i)|^2} \tag{7.4}$$

where N is the size of training or test samples, $\hat{y}$ is the predicted value, and y is the actual output value.

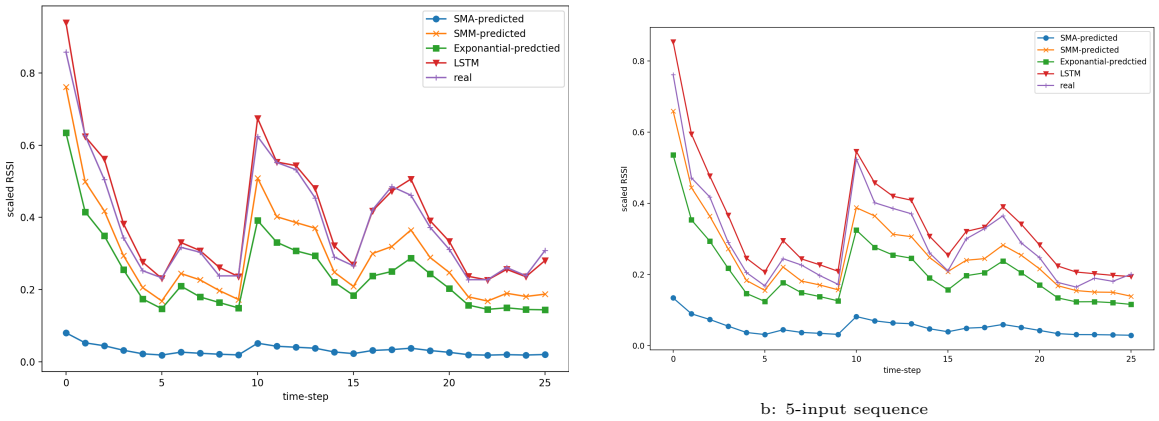
We first evaluate the performance of the proposed scheme in comparison to different prediction models. In the RSSI time-series forecasting scheme, we compare the LSTM with different time-series prediction models, including the simple models of historical mean and median [154], exponential smoothing [95], and Neural networks [120]. We choose the benchmarks, simple moving average (SMA), simple moving median (SMM), Brown's Simple Exponential Smoothing (exponentially weighted moving average)(SES), and the conventional neural networks (CNN) for comparison. The SMA and SMM predict the next possible RSSI values in time series, based on the median and average of N previous values as follows:

$$\hat{s}_t = \frac{1}{T}(s_{i-1} + s_{i-2} + ... + s_{i-T}), \quad i > T \tag{7.5}$$

The Exponential smoothing method assigns weights α to the given input set, in which the smallest weights are given to the oldest observations, as follows:

$$\hat{s}_{t+1|t} = \alpha * s_t + (1 - \alpha) * \hat{s}_{t|t-1} \tag{7.6}$$

The CNN is the closest to the LSTM model compared to the others, and has shown



a: 10-input sequence

b: 5-input sequence

Figure 7.4: Comparison of different time-series model one-step prediction values and real values for 10 and 5 inputs sequence

promising results in terms of image recognition and classification [120]. The CNN consists of convolutional layers, pooling layers, and fully-connected layers. In our experiments, we used a one-dimensional convolutional layer with pool size set to 2. A detailed description of the CNN model is found in [120].

The comparison of their performance in terms of one step prediction is presented in Figure 7.4 for five and ten input windows. The SMM outperformed all other models in the 5-input window with RMSE value of 1.467, followed by the LSTM model with 1.641. The SES reported 4.221 and SMA was the worst among all models with RMSE value of 19.087. As for the 10-input window, SES, LSTM, and SMM all had very close results in terms of RMSE with values of 2.374, 2.742, and 2.504 respectively. The comparison of their performance showed that the LSTM and SES require long time-series data to perform better in terms of one-time step prediction. Unfortunately, simple moving average and median only work with consistent changes, and perform poorly in a dynamic environment such as VANETs [53]. Therefore, since we want to use multiple time step forecasting for our handover trigger scheme, we choose the LSTM model, which better fits our network. However, one can still use the SES or SMM as a forecasting model, when historical trends are less affected by sudden changes.

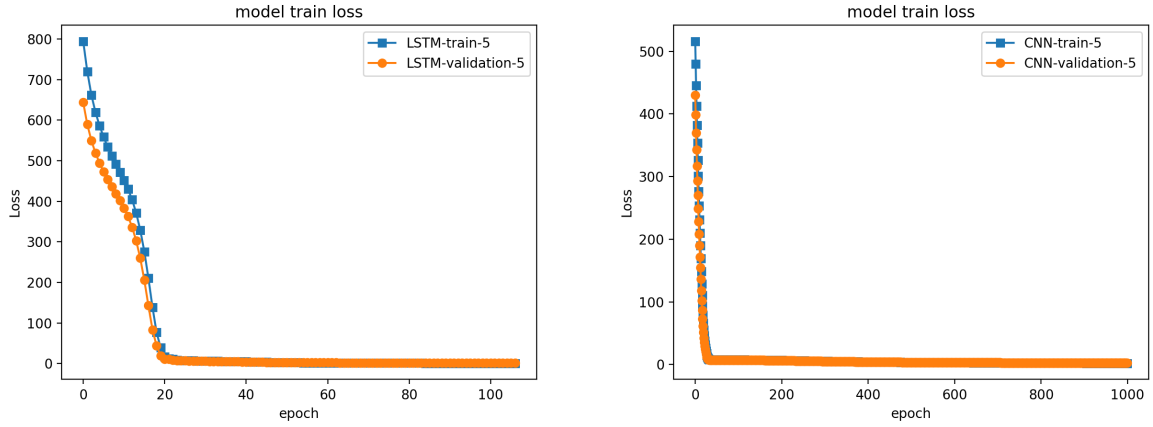
In our next set of experiments, we look further into both the LSTM and CNN models for multiple time-series forecasting in term of training and testing. Our first concern was to choose the most appropriate window size for prediction. In Table 7.2, we demonstrate the Mean Square Error loss of both LSTM and CNN models in training, validation, and

Table 7.2: Comparing input-output sequence window in LSTM and CNN model.

Model	[in-out]	train loss	val loss	test loss
LSTM (20)	[5-5]	42.607	34.819	4.250
	[6-4]	55.2958	43.5444	3.5376
	[7-3]	40.5635	31.9762	3.3245
	[8-2]	74.8508	57.9437	2.8141
LSTM (50)	[5-5]	54.5108	41.5638	4.9823
	[6-4]	68.1267	46.4459	6.0073
	[7-3]	51.7304	35.8894	3.3672
	[8-2]	124.3477	82.1910	6.7889
LSTM (100)	[5-5]	66.6582	45.2914	7.9242
	[6-4]	43.5615	30.0948	5.1045
	[7-3]	38.5577	23.7136	4.1351
	[8-2]	39.5542	22.9963	3.7246
Model		train loss	val loss	test loss
CNN (1D)	[5-5]	11.5689	10.8022	5.3872
	[6-4]	9.9481	8.7508	3.4389
	[7-3]	7.0871	6.7677	4.8571
	[8-2]	12.7638	10.5065	3.3005

testing. The results were taken after averaging several runs over 1000 epochs with different input-output sequence shapes. Both LSTM and CNN showed relatively close loss values when tested on an unseen datasets, but the CNN performed better during the training and validation step. Furthermore, the LSTM model of 20 layers showed the lowest loss when tested on unseen datasets and showed a decrease in loss when the input shape was long and the output was short. Nonetheless, due to the fact that in our system, the handover process latency requires more time to finalize the handover registration with a new AP, we preferred to choose the 5-input and 5-output window size to allow the handover protocol to finish the registration, and also to consider the possibility of wrong predictions.

In Figure 7.5, we illustrate the training and validation loss of both LSTM and CNN for 5-input and 5-output time-series sequence prediction. Here, the LSTM showed a high loss in training and validation in comparison to the CNN model, yet LSTM showed better performance than CNN when tested on an unseen dataset, as the average RSME of testing



a: LSTM b: CNN

Figure 7.5: A 5 input/5 output time-series sequence prediction training and validation loss for LSTM and CNN networks.

LSTM is 0.990 and 1.125 for CNN. The error difference between the LSTM network and CNN does not significantly distinguish a winning ML network selection for our objective. However, if a choice were to be formed with the requirement of a faster learning approach, then CNN would be chosen. Since our model is trained in an offline manner, training is not crucial for our proposed handover prediction. Thereby, we adopt the LSTM network to forecast the RSSI time-series sequence. After choosing LSTM as a prediction model for the mobility management trigger scheme, we incorporate the trained model into the vehicles for testing on a new dataset.

7.3.3 Network Performance

We evaluate the network performance of our protocol in terms of handover latency and average packet drop rate. In our previous work in Chapter 6, we have shown that the predictive HMIP protocol outperforms the standard Mobile IP and HMIP protocols with approximately 60% reduction in handover delay. However, the original predictive HMIP used the HMM exhaustively to predict the next AP, even though the vehicle was still connected to the same AP and did not require early registration at that time. This resulted in a very early registration request that would cost the network resources and bandwidth. To overcome this drawback, we integrated the handover trigger scheme presented in earlier sections, which evaluates the current and predicted RSSI sequence to decide on the handover switch. Thereby, the new protocol will not overuse the HMM model, and reduces the gap between the early registration and the actual handover event.

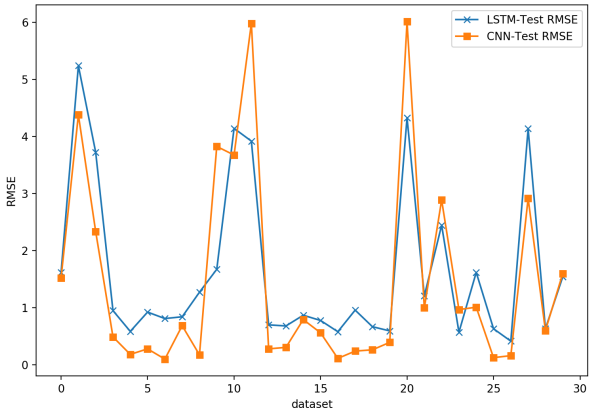


Figure 7.6: RMSE of LSTM and CNN testing on unseen dataset.

In Figure 7.7, we illustrate the dropped packet versus the simulation time, with and without the use of the handover trigger scheme. The performance of the hybrid ML-based approach has reduced the number of dropped packets with approximately 18% reduction in lost packets. The reason for that is that the original approach performs the registration request too early, which forces the vehicles' HA to forward the CN's packets to both the current AP and the predicted AP, until a transition happens and the vehicle disconnects from the previous AP. This could result in too many dropped packets in the event of a wrong prediction. Therefore, the deployed handover trigger scheme makes sure the vehicle will not perform an early registration, unless needed in a reasonable time window to cover the complete handover process.

In Figure 7.8, we illustrate the difference between both Predictive HMIP and ML-based HMIP in terms of the number of received packets. Results indicate that our scheme yields a higher packet delivery ratio compared to the original scheme, because of the appropriate handover trigger time. The proposed handover trigger scheme ensures that no handover will be initiated too early, and will estimate the exact window in which the vehicle needs to switch between access points.

7.4 Final Remark

In this Chapter, we have proposed a machine learning-based handover management scheme in VANETs. By taking into account the accuracy of the LSTM network in forecasting RSSI

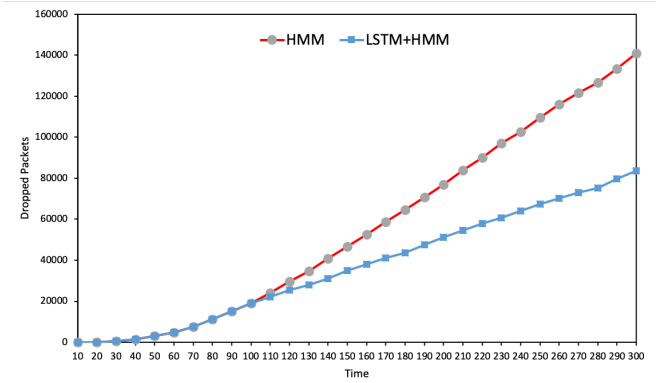


Figure 7.7: Comparison of the hybrid and single handover scheme in term of number of dropped packets over simulation time.

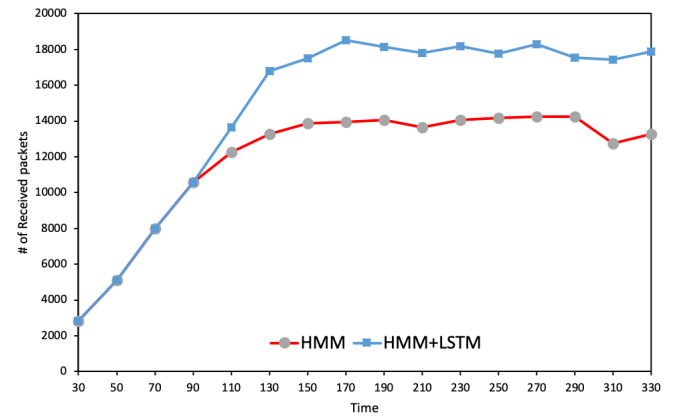


Figure 7.8: Comparison of the hybrid and single handover scheme in term of number of received packets over simulation time.

data sequences to induce an early handover trigger decision, we avoid long handover latency and services disruption in vehicles. We further add to our model a second tier that decide on which access point the vehicle should commit to, using an HMM model. Our results showed a high accuracy of the LSTM model in comparison to various forecasting models. Also, the integration of an ML-based handover trigger scheme resulted in improved network performance.

Chapter 8

An Online ML-Based RSU Prediction Scheme for Intelligent Vehicular Networks

8.1 Introduction

In recent years, Machine Learning (ML) models are playing a huge role in the design of Intelligent Vehicular Networks [59,197]. There are two types of ML that can be adopted for predicting the vehicles' upcoming connection, namely supervised learning, which includes classification methods, and unsupervised learning, such as clustering and rules extraction. In vehicular networks, the vehicles' movement information can be easily extracted using the network communication packets. Henceforth, a semi-supervised learning method can be adopted for this purpose.

In this Chapter, we aim to maximize the accuracy of the Roadside Unit (RSU) prediction, taking in mind communication cost and computational time. Thus, increasing the probability of a smooth transition between networks with minimal disruption time window. We propose an online ML-based RSU prediction scheme for Vehicular Networks, that uses the underlying state-of-the-art mobility management protocol (Mobile IPv6), to integrate the prediction model within the network layer. We develop an online Probabilistic Neural Network (PNN) model to predict the next RSU in a real-time manner via incremental learning strategy. Finally, we compare and evaluate the accuracies of the chosen prediction model in comparison to several well-known machine learning classifiers and discuss their performances.

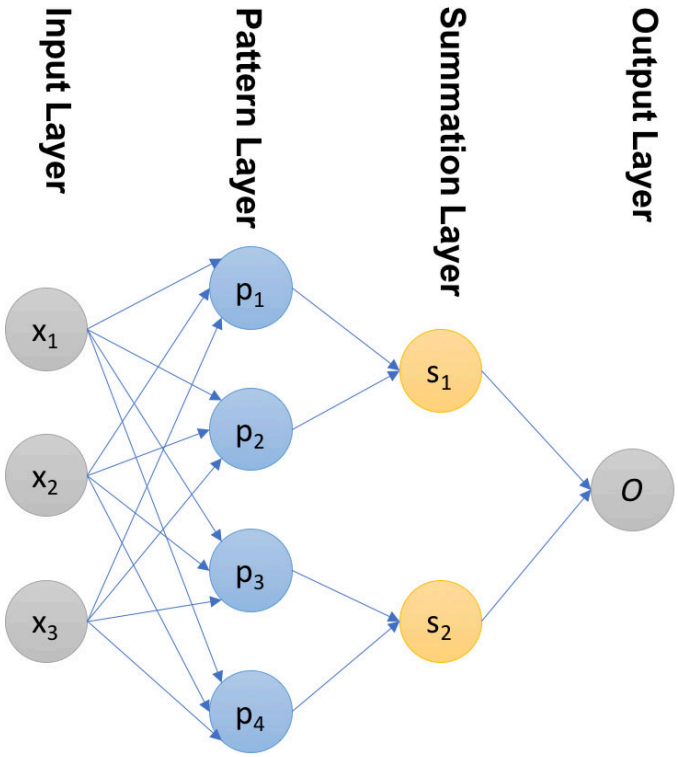


Figure 8.1: PNNs architecture.

The remainder of this Chapter is organized as follows. Section 8.2 presents the proposed online machine learning-based roadside unit prediction scheme. Followed by the simulation and performance evaluation of the protocol in Section 8.3. Finally, the concluded remarks are discussed in Section 8.4.

8.2 OPeN-PreS: Online Probabilistic Neural Network-based RSU Prediction Scheme

In this section, we describe the proposed neural network model for predicting the next RSU using vehicles' mobility information. Then, we describe the online PNN-based prediction scheme implemented within VANETs mobile IPv6 mobility management protocol.

The Probabilistic Neural Network is a machine learning algorithm inspired by Bayesian networks [177] and Kernel Fisher discriminant analysis [148]. PNNs have shown promising

results for classification problems [86, 185, 197], where local minimums problem does not affect the decision of the network. The probabilistic neural network differs from the standard back-propagation method in the way it processes the output layer. The main layers of a PNN model consists of an input layer, pattern layer, summation layer, and output layer. The input layer consists of n neurons, one for each of the n input features of an input vector $x = (x_1, .., x_n)^T \in R^n$. It distributes the input to the pattern layer's neurons. The pattern layer's neurons are partitioned into k sets, one for each class in the training set, through a Gaussian function. The summation layer has c neurons, one for each class. It adds the output from the pattern layer's neurons of a given class. Thereby, generating probability values to estimate the likelihood of output class. The prediction model algorithm works as follows: Read new input data X_new at the input layer, then calculates the Gaussian kernel of each input vector:

$$F_{k,i}(X) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{||X - X_{k,i}||^2}{2\sigma^2}\right) \tag{8.1}$$

where $x_{k,i} \in R^n$ is the kernel's center, and σ is the adaptable *spread* parameter. When the value of the spread parameter is close to 0, it resembles the nearest neighbor classifier of several training samples. Then, calculate the class-conditional probability of each class in the summation layer:

$$G_k(X) = \sum_{i=1}^{M_k} w_{ki} F_{k,i}(X), \quad k \in 1, ..., K \tag{8.2}$$

where M_k is the number of pattern neurons of class k, and w_{ki} is a positive coefficient such that $\sum_{i=1}^{M_k} w_{ki} = 1$. The output layer classifies an input vector, x, based on the maximum probability obtained using Bayes' theorem. Finally, the output layer is responsible of selecting the higher class-conditional probability as follows:

$$C(X) = arg \max_{1 \le k \le K} (G_k) \tag{8.3}$$

thereby, the class with the highest probability will be assigned a value 1. The selection of a spread parameter is further studied in Section 8.3.

In the context of vehicular network's mobility management, each vehicle presents a certain feature upon connecting to the next access point, either by location, direction, speed, or signal strength. We define the input vector X_i of the prediction model to be carrying the vehicles' trajectory measurements. The connected RSU represent the output class G_i, assuming the network has M RSU's and each has k neighboring classes.

The initialization of the prediction model is done locally on each Foreign Network,

Algorithm 8.1 AP initialize and update methods.

```
1: procedure INITIALIZE
2:     Send Broadcast Hello − Msg
3:     Set boardcastTime
4:     Set Threshold
5:     Set TrainTime
6:     Set Flag ← True
7:     while broadcastTime > T_curr do
8:         Recieved neighboring AP Packet
9:         updated Neighboring list List_N
10:    end while
11:    Create PNN
12: end procedure
13: procedure UPDATE METHOD
14:    for Each UpdatePkt do
15:        if TrainTime > T_curr then
16:            update TrainSet with received packet
17:            return
18:        end if
19:        if TrainTime <= T_curr and Flag then
20:            PNN ← TrainSet
21:            Flag ← False
22:        end if
23:        if !Flag then
24:            update X                          ▷ update the dataset vector
25:            acc ← accuracy_score(AP_a, AP_p)
26:            if acc < Threshold then
27:                Retrain PNN
28:            end if
29:        end if
30:    end for
31: end procedure
```

in which every Foreign Network is responsible of updating the prediction model through context packet sent through the network from neighbors. Initially, all acess points will share one broadcast message with its neighbors to allow each of them to identify the neighboring list. Upon the vehicles' connection to a new acess point, it will send an update packet

Algorithm 8.2 MH prediction method.

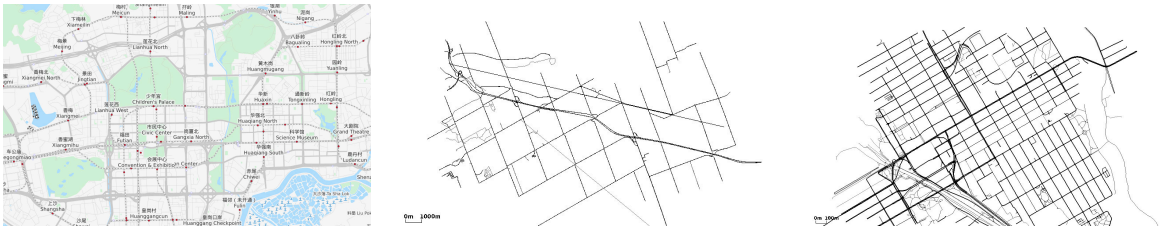
Input: PNN, $list_N$, X_i, $Timer$

```
1: procedure AP Prediction
2:     for Timer do
3:         X_i get current movement information
4:         AP_p ← PNN(X_i)
5:         if AP_p ∈ list_N then
6:             return AP_p
7:         else
8:             reset Timer
9:         end if
10:    end for
11:
12: end procedure
```

to the previous point with the vehicles' measurements information. The packet contain the IP of new acess point, the predicted point, and the vehicles' movement measurements. Each acess point will follow Algorithm 1 to identify neighbors and update the prediction model.

As aforementioned, each AP initializes a prediction model and trains the model after $TrainTime$ with the gathered data from the vehicles' mobility. Vehicles connecting to acess points are responsible of maintaining their mobility information until a transition happened or a prediction is made. When a vehicle is connected to the new acess point, it will send an update message to the previous acess point with its mobility information prior to transition, in addition to the predicted acess point and the real one. When the training time is up, the prediction model will be trained and ready to predict the next data inputs. In case the prediction accuracy drops below a certain threshold, the PNN model will be retrained with the new dataset vector.

A vehicle will start the transition between acess points only when a prediction is made or the distance between the new acess point and the vehicle is less than the distance between the current acess point and the vehicle or the vehicle lost its communication with the current acess point and require a new connection. Upon connecting with the new acess point, the vehicle will also receive a copy of the prediction model weight matrix along with the neighbors list. Each vehicle will periodically collect its movement measurements and initiate the prediction using the received model copy. When a prediction is made, the vehicle will initiate an early registration process with the predicted acess point and start



a b c

Figure 8.2: Road network topologies: a: Shenzhen City b: Ottawa Highway c: Ottawa Urban.

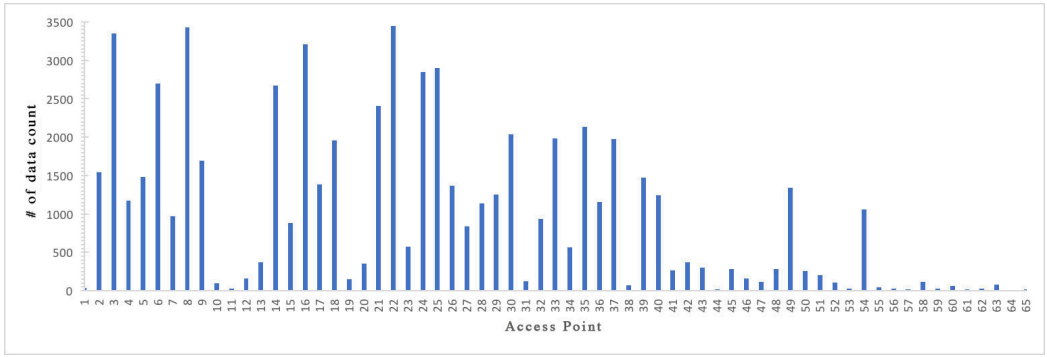
Table 8.1: Simulation parameters

Parameter	Value
Mobility Simulator	SUMO
simulation duration	500 s
PHY/MAC	IEEE 802.11p
vehicles	100-400
vehicle speeds	0-30 m/s
road network	Urban - highway
AP coverage	200 m
Network simulator	NS-2.35

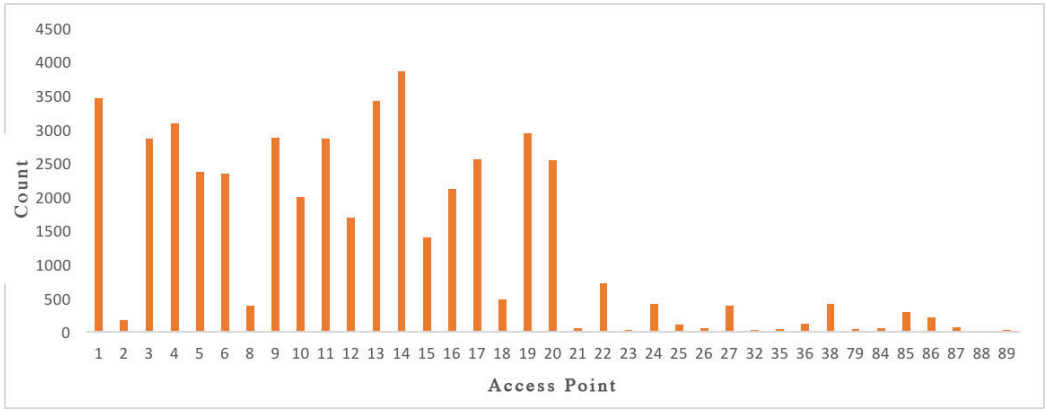
the transitioning phase. In case of a wrong prediction, in which the vehicle communication with its current acess point is weakened and no prediction has been made. Thereby, the vehicle will check in its acess points cache for the most applicable point to connect to. If the vehicle cache is empty, then the original Mobile IP transition process will be initiated.

8.3 Performance Evaluation

In this section, we evaluate the performance of the machine learning-based mobility prediction model for VANETs. We first provide a detailed description of the network and road environments used in evaluating the simulation results. Followed by data preprocessing and optimization of the online prediction model parameters. Then we evaluate the performance of the online prediction model in comparison with various well-known classifiers and discuss our findings. We choose the prediction accuracy and latency as performance measures for analysis. To generate vehicle mobility data, we use SUMO tool [118] for both Urban and Highway environments. In addition, Ns-2 simulator is used to simulate



a: Urban



b: Highway

Figure 8.3: Number of dataset for each AP class.

the wireless mobility management network and protocol. Table 8.1 summarizes the used parameters for the simulation.

8.3.1 Environment and Network Setup

In our experiments, we use two different road structures, Ottawa Urban and Highway road network, and Shenzhen City road network. The first set of road networks are downloaded from OpenStreetMap [79]. The Ottawa Urban road network is $4000mx3000m$ of Ottawa City Centre, in which we aim to test the prediction in case of high turn possibility and various road intersections. The road structure of Urban networks usually present a challenging task for many predictors, because of the vehicles sudden movement around intersection that most predictors fail to anticipate. As for the Ottawa Highway road network, illustrated in Figure 8.2, where vehicles usually present higher speed rate and less prone to

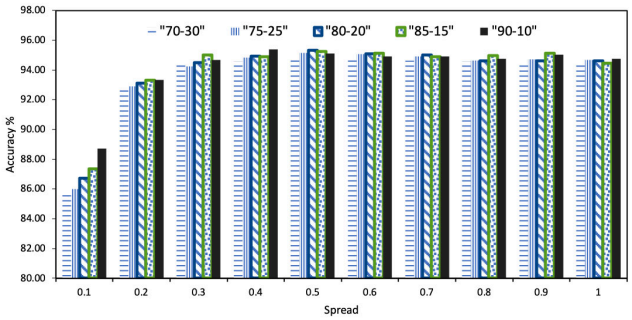


Figure 8.4: Spread optimization with respect to data division ratio.

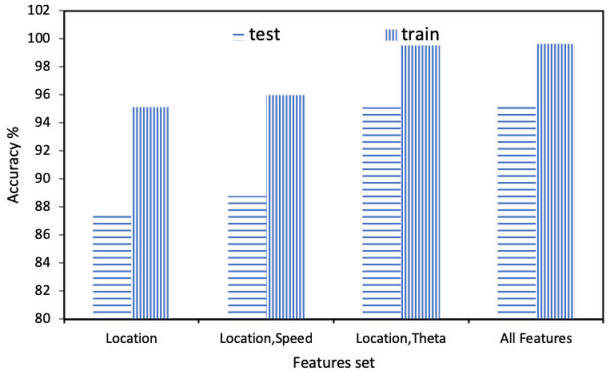


Figure 8.5: Different features accuracies for Urban dataset.

sudden movements. Such environments present the challenge of an increase frequency of FA changes and requires quick predictions as the vehicle speed increases. Finally, Shenzhen city road network taken from [40] represents 8x24 k/m of the city structure, as shown in Figure 8.2(a). The use of such network in our experiment will provide a more realistic measurements and evaluation of the proposed model. The generated mobility of SUMO represents the synthetic data for both Urban and Highway mobility of Ottawa City. The number of vehicles ranges between 100-500 vehicles. As for the vehicles' movement, we use the Krauss car-following model with vehicle speed ranges between 0-30 m/s

The generated mobility of SUMO and Shenzhen Taxi's are imported into NS-2 for simulation. We choose the IEEE 802.11p protocol in the MAC layer, as it has been recently adapted for vehicular network mobility. We focus on the infrastructure-to-vehicle communication, since the deployment of RSUs are set to maximize the network coverage, thus allowing all vehicles to access RSU communication range.

We extend the mobile IPv6 protocol to implement the prediction model OnlinePNN and adjusted according to the proposed scheme in previous sections. In the following, we first optimize the prediction parameters to best fit the vehicular networks environment, then analyse the dataset features to choose the best set of features that describe the I2V connection.

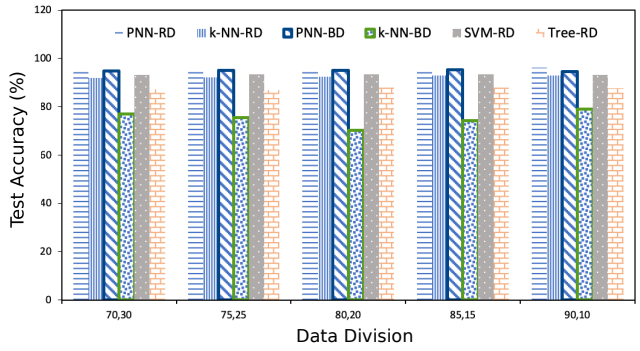
8.3.2 Parameters and Features Optimization

In order to evaluate the accuracy of a classifier, several techniques can be used. One method is to split the dataset into training and prediction. In another method, cross-validation, which divides the dataset into equal-sized subsets and each subset are trained on all the other subsets. The average of multiple rounds of cross-validation is derived from giving an estimate of the predictor accuracy. In our test, we use the dataset splitting technique to calculate the prediction accuracies of each classifier. In Figure 8.3, we show the number of collected datasets in each environment and their distribution on each RSU. As noticed, the Urban network fully utilize the deployed RSUs except for few outliers. However, the number of datasets in the Highway environment shows more concentration on half of the deployed RSU, which could be further improved by rearranging them. Yet, the scope of this work is merely improving the accuracy and the efficiency of the prediction model and not the load distribution on each antenna.

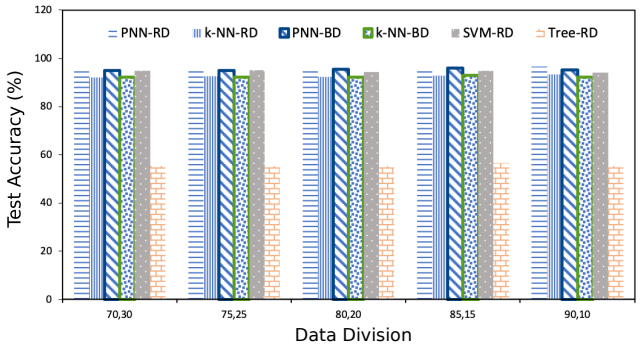
To fully utilize the neural network model, we first need to optimize both the smoothing parameter and data features. In Figure 8.4, we evaluate the influence of spread selection on PNN for RSU prediction when the spreading interval between 0.1 and 1. We found that the best value obtained from each data division is between 0.4 and 0.6 for As for the features set, the optimum set of features that represent the vehicles' characteristic is evaluated as seen in Figure 8.5, where the accuracy of different feature sets is tested. The results indicated that the location and direction features report the highest prediction accuracy in comparison to the rest. This is valid as the vehicles' speed is not an effective feature to predict vehicles' movement, where different vehicles may exhibit different speeds and yet connect to the same access point.

8.3.3 Prediction Accuracy

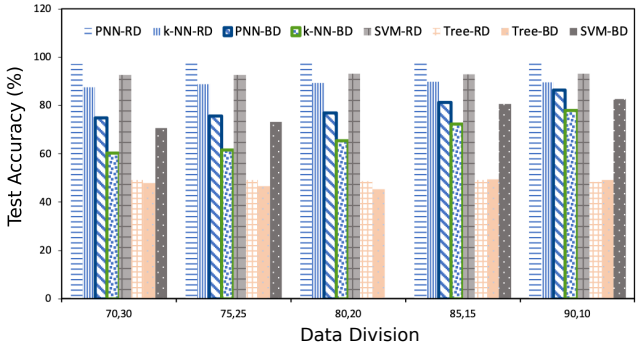
The Probabilistic Neural Network prediction model is compared against several well-known classifiers, namely, k-NN, SVM, and Decision Trees in term of accuracy and time efficiency. The Support vector machine is based on statistical learning theory, in which it adopts the



a: Urban



b: Highway



c: Shenzhen

Figure 8.6: Comparison of different predictors accuracies over different train/test divisions.

structural risk minimization principle. It has been shown that even though SVM shows relativity acceptable accuracy rates, it performs poorly as the dataset size increases. As for the Decision Tree classifier, which uses a tree-like graph for each class and their possible set of features [35]. The k-nearest neighbor (k-NN) is based on the proximity of each instance

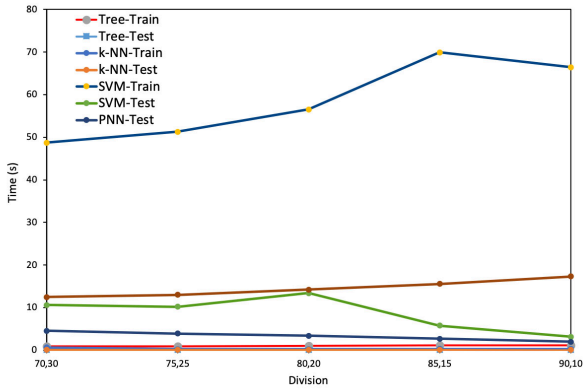
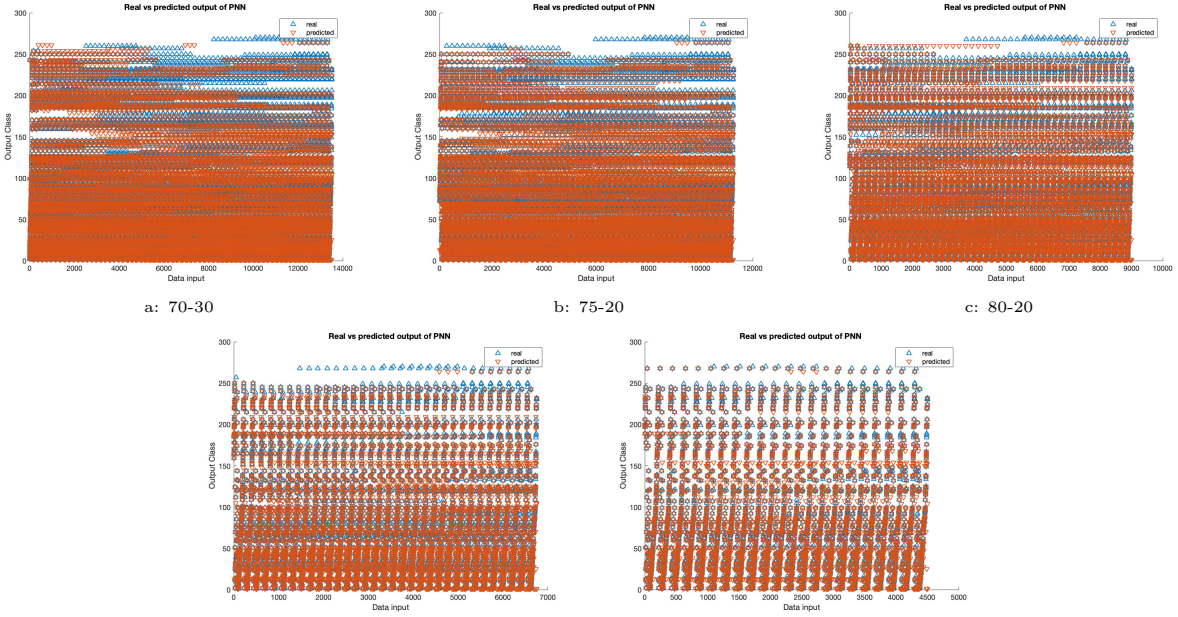


Figure 8.7: Comparison of different predictors training and testing time over the number of dataset.



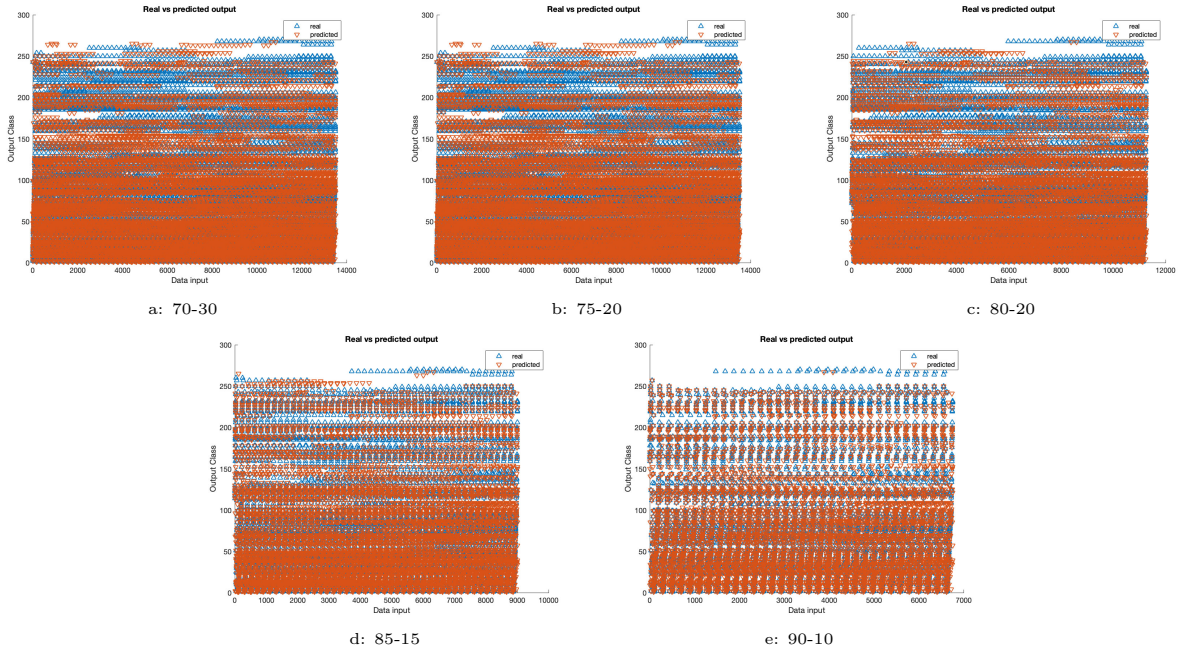
a: 70-30 b: 75-20 c: 80-20

d: 85-15 e: 90-10

Figure 8.8: PNN prediction classes under various data division with prediction accuracy average [94.86, 95.02, 95.02, 95.39, 94.6].

to other instances with similar features [52].

In Figure 8.6, we show the predictors accuracies in different environments and divisions. First we wanted to evaluate the performance of the predictors in an offline manner. We extended our investigation to evaluate the performance of each predictor when the data



a: 70-30 b: 75-20 c: 80-20

d: 85-15 e: 90-10

Figure 8.9: k-NN prediction classes under various data division with average prediction accuracies [77.12, 75.59 ,70.22 ,74.22, 79.01].

is given randomly and in sections. The goal is to study the predictors performance when portions of input data may not be seen by the model after training. As seen in Figure 8.6, the accuracy of PNN model outperform all classifiers in different divisions. In Shenzhen city environment, the difference between PNN and other models is clearly observed. As for the Urban scenario, most predictors preform well except for the decision tree and k-NN when the division of data was in sections rather than random. In the case of the Highway scenario, all models preformed relatively the same, due to the fact that in highway scenarios, vehicles tend to show similar movements projections when connecting to any new acess point as they are moving on the highway. The results in Figure 8.6, indicates that choosing the PNN model is the appropriate choice among other classifiers to predict the vehicles' next acess point connection. Furthermore, we looked into the time efficiency of each predictor in Figure 8.7. The results of both training and testing time of both k-NN and Decision Tree showed the lowest computational time in comparison to PNN and SVM. Yet, the reflected accuracy of the Decision Tree shows the worst performance. In addition, SVM showed the worst computational time with an average of 58 s for training and 8.5 s for testing. The PNN model showed a relatively good computational time with an average of 13 s for training and 3 s for testing. At this point, we eliminated the Decision Tree

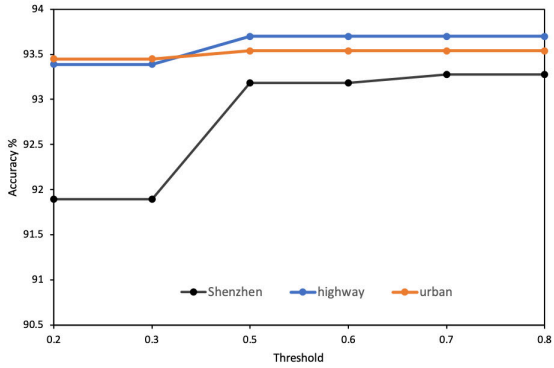


Figure 8.10: Impact of threshold value on the accuracy of OnlinePNN with respect to various mobility traces.

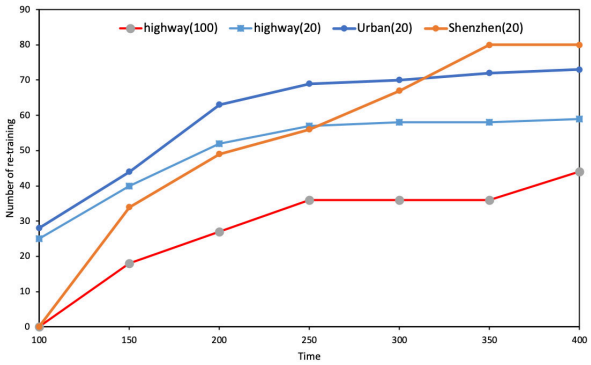
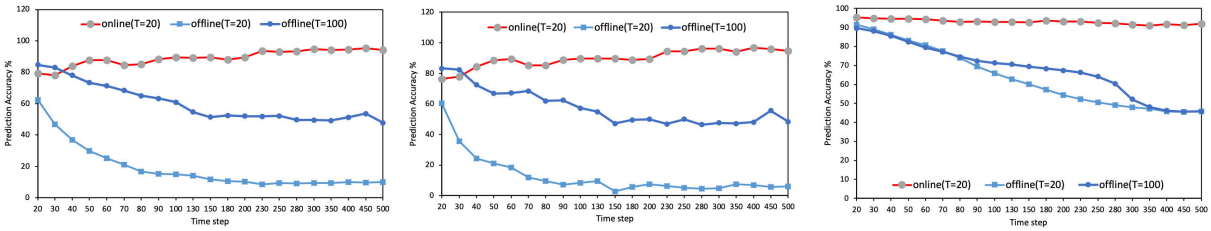


Figure 8.11: Comparison of number of performed training in OnlinePNN model with respect to various mobility traces.

model, as it presented the worst accuracy outcomes. As for the SVM model, it showed even higher training time as the training set increased, which indicates that it will be unsuitable for VANETs high scalability. Therefore, in the following set of experiments, we choose only the k-NN for further evaluation. Figures 8.8 and 8.9 demonstrate the real verse the prediction output classes for both PNN and k-NN models in Shenzhen city mobility scenario. As seen, the PNN model outperformed the k-NN model in all data division with an average prediction accuracy of 95% and k-NN average of 74% prediction accuracy.

In the following set of experiments, we evaluate the performance of the PNN model in an online manner within mobile IP protocol. Starting with investigate the affect of retraining the prediction model after the accuracy rate goes below α threshold value. We set the initial training time to be $20s$ in which each AP will keep collecting the vehicles' movements vector and its corresponding RSU connection. In Figure 8.10, we illustrate the impact of threshold value on the accuracy of the prediction model over the simulation



a: Highway mobility b: Urban mobility c: Shenzhen City mobility

Figure 8.12: Comparison between online and offline prediction accuracies.

time. As the threshold value increases, the accuracy of the onlinePNN predictor increases, but until a certain point in which the accuracy is relatively stable. In case of Shenzhen real data scenario, the initial threshold of 0.2 showed the least accuracy in comparison to Ottawa highway and urban scenarios. This is due to the huge environment area of Shenzhen city map and also the spread of vehicles on the map, thus not allowing the initial training to collect enough vehicles movement informations. Therefore, we choose 0.5 as a threshold value for the onlinePNN prediction model.

In Figure 8.11, we show the number of model retrains over simulation time of different environment with a threshold set to 0.5. As seen, the highway environment showed the least numbers of required retraining due to its simple structure in comparison to the Urban scenario and Shenzhen City. In addition, when we change the initial training period from $20s$ to 100s, the result indicate that the longer time we have to collect dataset for training, the lower required model retraining times.

Finally, in Figure 8.12, we illustrate the difference between the online and offline PNN model in different mobility environments. The onlinePNN outperformed the offline method with an average accuracy of 90% in highway and urban scenarios and 93% in Shenzhen city scenario. Also, we have shown in Figure 8.12, that the more time given for the predictor to collect training set, the more accurate predictions are made. Nonetheless, our results does not take in mind the RSU load nor the vehicles Quality of Service demands, which can further improve the efficiency of the proposed model.

8.4 Final Remark

In this chapter, we proposed an online machine learning-based roadside unit prediction scheme for vehicular networks. We choose the probabilistic neural network classification model to predict the next access point using the vehicles' movement information. We have

implemented our scheme on top of the mobile IP protocol using the network simulator NS-2. A thorough evaluation of the proposed scheme has been presented and compared with various known classifiers. Our results showed the efficiency of the machine learning-based model in term of accuracy and time complexity. During investigation we have shown the deployment of access points may further enhance the performance of the prediction model by looking into the load balancing issue and Quality of Experience (QoE).

Chapter 9

Adaptive Location Management Scheme for Intelligent Vehicular Networks

In this Chapter, we propose an ADaptive VehIcle-CEntric LOCation management scheme (ADVICE-LOC) for intelligent vehicular networks. The redundancy in vehicles' location updates and registration with foreign networks are optimized based on vehicles' movement projections. In general, the proposed protocol consists of two main parts, the estimation of vehicles' residual time within a network infrastructure unit using the vehicles' mobility data, and then utilize that knowledge to adapt the location management scheme in terms of networks' registration and update. Moreover, the proposed scheme preemptively adapts the vehicles' session time depending on the surrounding networks and changes in vehicles' mobility; thus, reducing the communication overhead and eliminating the redundancy in location updates with the current access point. Furthermore, we thoroughly evaluate several time-series predictors in the urban mobility environment and report their advantages and disadvantages. The most appropriate predictor is then used to derive a link estimation time. Finally, the proposed protocol is simulated and evaluated under multiple network and accuracy metrics.

The remainder of this Chapter is organized as follows. The related literature review and discussion are presented in Section 9.1 followed by the problem statement and preliminaries in Section 9.1. The proposed adaptive location management protocol is detailed in Section 9.1. A detailed discussion and evaluation of the proposed protocol are presented in Section 9.1 and compared to benchmark mobility management protocols. Finally, we conclude our findings in Section 9.1.

9.1 Background

In this section, we discuss and survey the related studies addressing mobility management issues within vehicular networks. More precisely, we examine studies that aim to optimize wireless communications during vehicles' mobility within foreign networks. The study of mobility management can be described as the maintenance of ongoing wireless sessions and connections between vehicles and any services that can be either provided by local service gateways, or on the cloud services. Many of these require continuous communications over the Internet to deliver services efficiently. Thus, seamless mobility management is required to avoid communication interruptions.

In Chapter 3, we discussed in details the mobility management schemes for vehicular networks. However, most designs improve handoff latency in some scenarios but increase the network's signaling overhead. Offloading, on the other hand, is typically done due to an over-crowded system, and load or a new quality of service requirements is needed from another network. In both cases, an efficient and seamless transition of communication between the mobile node and the selected offloading station needs to be implemented.

The location management process deals with the vehicles' tracking and location updates. While the vehicle is roaming within a foreign network, periodic updates of the vehicle location and registration are made every predefined period of time. This will help the vehicle or the infrastructure units (i.e., access point/router) to track vehicles' movements and initiate the handover process accordingly. In addition, this will also help the home agent to track the vehicles' location and forward data packets to their current locations in a real-time manner. However, Mobile IP standard does not consider any form of paging support and requires continuous location updates to maintain connectivity with the access point, even if it is not moving or is in idle mode. Therefore, abundant singling overhead is produced and, in the case of battery-powered vehicles, high power consumption is introduced. To address this issue, location management protocols can be intelligently adapted depending on the context information of vehicles' movement, network conditions, and traffic conditions. One possible direction to enable the adaptivity of location management schemes, by monitoring and utilizing the vehicles' residual time within the communication range of an access point.

The study of the residual time estimation and the communication efficiency of various mobility models have been addressed in the research community [47, 117, 240]. This relationship is widely considered in many routing protocols [229], whether the link estimation represents stability, time, or availability. Because of the vehicle's rapid mobility changes, maintaining a routing path has become a vital issue for efficient wireless communication.

Thus, by estimating the availability and stability of a link between a vehicle to another vehicle or a gateway point, network efficiency is improved in terms of packet delivery ratio and communication latencies.

Several studies worked on estimation the link duration time using cost-function evaluation or machine learning approaches. Bi et al. [23] proposed a neural network-based mobility prediction scheme to estimate the link duration between neighboring vehicles, thus allowing a pre-creation of routing paths. Link stability has also been used in vehicular named data networks [54], to reduce the number of forwarded interest and data packets. To the best of our knowledge, few works have been done that investigate adopting link estimation techniques in mobility management schemes.

9.2 Problem Statement and Preliminaries

In this section, we define the set of requirements and assumptions to derive the estimated session time between a vehicle and an access point or router, or as we refer to in what follows, link estimation time. Then, we describe the characteristics in vehicles movement projections as data input for time-series forecasting methods. We assume that vehicles are moving in predefined road structures while the access point's deployment is maximizing the roadmap coverage.

9.2.1 Link duration estimation

The duration time in which a vehicle resides in the communication range of an AP is called, Link Estimation Time (LET). The duration of this link is affected by several factors, including vehicles' speed, the distance to the access point, and the quality of the signal, among others. In Figure 9.1, an example of a vehicle movement projections is displayed. If we only consider the distance between the vehicle and an access point, the derived time will be the shaded area; however, this does not reflect the real remaining residual time of the vehicle within the communication range of the access point. Therefore, resulting in false estimation. In this work, we assume that each vehicle is equipped with on-board GPS sensors and capable of obtaining its current location coordinates (x_i, y_i). From this information, each vehicle can calculate its own movement projections periodically using two consecutive positions. In addition, RSUs or APs have a standard communication range R. Thus, the minimum estimated link duration time is derived as follows.

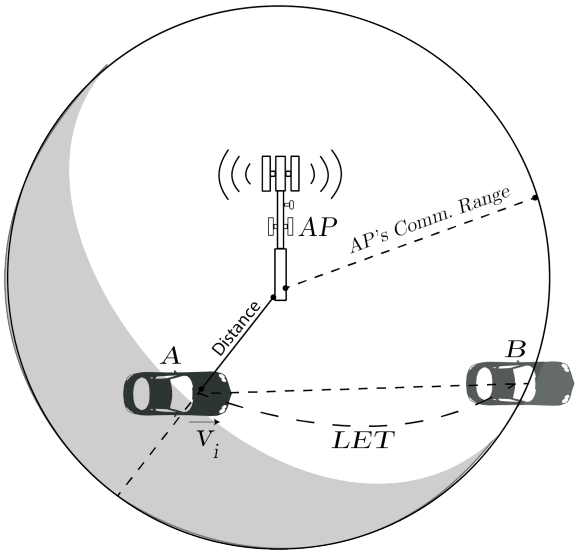


Figure 9.1: Link Estimation Time (LET).

In the line-of-sight situation, a vehicle can communicate with an access point within its communication range. An access point with position (AR_x, AR_y) and R range of communication can be described as the equation of circle

$$(x - AR_x)^2 + (y - AR_y)^2 = R^2 \tag{9.1}$$

A vehicle moving from a point $A(x_{i-1}, y_{i-1})$ to point $B(x_i, y_i)$ describes a line of multiple points (i.e., consecutive series of locations) along the straight movement path and can be calculated as follows

$$\begin{aligned} x(T) &= (x_i - x_{i-1})T + x_{t-1} \\ y(T) &= (y_i - y_{i-1})T + y_{t-1} \end{aligned} \tag{9.2}$$

$$((x_i - x_{i-1})T + x_{t-1})^2 + ((y_i - y_{i-1})T + y_{t-1})^2 = R^2 \tag{9.3}$$

deriving a quadratic equation

$$aT^2 + bT + c = 0 \tag{9.4}$$

where

$$\begin{aligned} a &= (x_t - x_{t-1})^2 + (y_t - yt - 1)^2 \\ b &= 2[(x_t - x_{t-1})(x_{t-1} - AR_x) + (y_t - y_{t-1})(y_{t-1} - AR_y)] \\ c &= (x_{t-1} - AR_x)^2 + (y_{t-1} - AR_y)^2 - R^2 \end{aligned} \tag{9.5}$$

Finally, solving for T, using the alternative quadratic formula

$$T = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \tag{9.6}$$

Now, we can find the next location of the vehicle at time T is $(x(T), y(T))$, assuming it remains in the same direction with velocity v. This point will determine the remaining distance between the vehicle's current position and the edge of the AR communication range.

$$LET = \frac{dist(x(T) - x_t, y(T) - y_t)}{v} \tag{9.7}$$

Furthermore, by taking in mind the packet delivery time and round trip time into consideration, we adjust the estimated LET to deduct the packet round trip by

$$\begin{aligned} Packet_d(t) &= Tran(t) + Prop_d \\ RoundTrip(t) &= 2 \quad Packet_d(t) + P_d \end{aligned} \tag{9.8}$$

where $tran(t)$ is the transmission time, $Prop_d$ is the propagation delay time, and P_d is the processing delay.

9.2.2 Vehicles' movement projections

The aforementioned link duration time estimates the vehicles' session time with the current access point, using only current measurements of vehicles' movements. This does not take into account the upcoming changes in the vehicle trajectories. Thus, predicting the vehicles' upcoming trajectory (i.e., location, speed, direction) will derive a better estimation to the residual time in which a vehicle will stay within the current connected access point. In order to do so, several prediction models can be adopted to predict future trajectories.

Vehicles' movements are usually tightly coupled to measurements of previous movements. Thus, providing a look-back window to the vehicle's previous movement patterns will lead to the estimation of upcoming projections.

Time-series forecasting

The early exploration of the time-series forecasting technique was first initiated by Yule [193], who proposed the Autoregressive model that predicts values using the weighted sum of previous series of observations. This simple technique has shown great results in so many linear

systems. However, in many scenarios this technique becomes very unpredictable.This can be seen in nonlinearity of vehicles' dynamic mobility situation. Depending on the given series of complicated time series, one needs a more general theoretical model to predict and forecast future observations accurately.

Over the past decade, two major groups of models were developed in order to provide accurate predictions to much more complex applications that can capture longer and complex time-series data. The two development techniques are machine learning (e.g., Fuzzy logic [41], neural networks [50]), and state-space models (e.g., Kalman Filters [107]). Machine learning, and neural network models in particular, are capable of capturing longer and larger amounts of time-series datasets, from rule-based machine learning models to data-driven methods. However, most of the machine learning techniques require large datasets to perform efficiently.

Let us assume that a sequence of vectors $x(t), t = 0, 1, ...$ represents a time-series data, where t is the sampling rate. In our context, the sequence of vectors represents the vehicles' movement measurements including location, direction, and speed. The rate to which the sequence is sampled and the size of each sequence differs from one system to another. In vehicular networks, vehicles are usually tracked and traced over very small period of time to accurately localize them. There is typically a time-step gap lasting between 1 - 5 seconds. To predict future observation of the vehicle's movement, we record the sequence of observations up to a time-window. Through our protocol, as presented in Algorithm 9.1, we collect the sequence of time-series observations of each vehicle using a queue embedded within the vehicle to record the last k observations of movement. The selection of the gap rate and the look-back window is studied and evaluated in Section 5.

9.3 ADVICE-LOC: The ADaptive VehIcle-CEntric LOCation Management Scheme

In this section, based on the derived measurements, we describe the proposed mobility and location mobility scheme phases that optimizes the communications overhead while maintaining high delivery rate and low packet loss during the vehicles' movement within an access point. The mobility management process can be described in two main phases: the discovery and registration phase and location/session update phase, as presented in Figure 9.2.

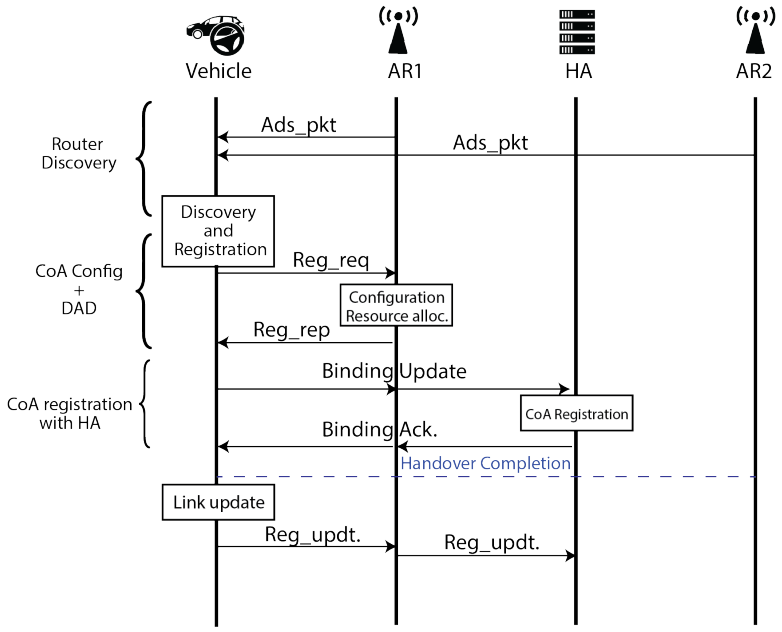


Figure 9.2: AD-MIP protocol

9.3.1 ARs' discovery and registration phase

When the vehicle moves through different networks, it requires a new point of access to be able to connect to and reach multiple services. Particularly in mobile IP, the vehicle starts with a permanent address through its Home Agent network. When a new foreign network is discovered or the vehicle moves far from its home network, a new Care-of-Address is needed to register the new access point and notify the home agent for tunneling ongoing communication toward that access point. The described process is known as Mobile IP. When the vehicle switches between two access points, a handover mechanism is required to register a new CoA and transfer the connection from the old access point to the new access point. The discovery and registration process is presented in Algorithm 9.2.

The vehicle is able to discover and register new access points through two main methods: access point advertisement messages or the vehicles' solicitation requests. When the vehicle discovers that its location has changed, it will either use the cache of neighboring access points or request a new connection through a broadcast message. Following the discovery part, the vehicle will then choose the most appropriate access point and send a registration request that will lead to CoA configuration and resource allocation for that vehicle. In our case, selection of an access point depends on the router distance and the received signal

Algorithm 9.1 Periodic time-series sequence

//Collect vehicles' movement measurement sequence every gap time-step.

Input: $Cache_V, gap, Timer_{seq}, cAR, curTime$

```
    procedure DATA COLLECTION
 2:     while Timer_seq ≤ curTime do
            Get vehicle's measurement vector m
 4:         if Cache_V is full then
                remove Cache_V.head()
 6:         end if
            add Cache_V.push(m)
 8:         Timer_seq.set(gap)
        end while
10: end procedure
```

strength. The access point will reply with the registration reply packet and will initiate the Binding update part. That is, the vehicle will send a binding update to its home agent for CoA registration and tunneling, followed by the home agent binding acknowledgement packet. At this point, the vehicle is now able to receive data from any corresponding service from the new access point through its home agent. In hard handover process, the vehicle will not be able to receive any data packets and communication between the vehicles, and the corresponding service is disrupted. However, a soft handover will perform the transition process before the vehicle leaves its current access point, therefore reducing the disruption.

9.3.2 Vehicles' session lifetime estimation

Following the vehicle registration with a new access point, a periodic update of registration and locations is needed to maintain the vehicle's current communication. Otherwise, after the registration lifetime expires, the home agent will assume the vehicle is no longer in the current access point range and will remove it from the forwarding table. Not to mention, any ongoing communication will be interrupted and not received by the vehicle. In a standard mobility management protocol, the update process, or location paging, is done periodically and in a predefined lifetime regardless of the context of the current session. Since vehicles have varying behavior and mobility, having a constant variable for updates will result in very high communication overhead. This will be significantly observed when presented in a dense environment. For this purpose, we propose an adaptive session lifetime estimation method that considers the context of the vehicle's movement to dynamically set

Algorithm 9.2 AR Discovery and Registration Phase

//Upon the reception of advertisement packet Ads_k from AR_k at t time-step

Input: $Cache_{AR}, Adptime, cAR, curTime$

```
    procedure DISCOVERY METHOD
 2:     while Ads_k.packet do
            if Ads_k.ip ∈ Cache_AR then
 4:             Update Cache_k
            else
 6:             Add new cache entry Cache_k.new
            end if
 8:         if cAR == Cache_k.ip then
                if Adptime < curTime then
10:                 Registration(Update)
                end if
12:         else if cAR == −1 then
                Registration(New)
14:             cAR ← Ads_k.ip
            end if
16:     end while
    end procedure
18: procedure REGISTRATION METHOD
        if New then
20:         Find nearest AR with high RSS
        end if
22:     Predict Movement using Cache_V
        Compute Adptime (9.7) EstimateLink
24:     Send Reg_Req
    end procedure
```

and update the location and registration lifetime between the vehicle and its current access point. Thus, we avoid redundancy in update packets and cost of re-registration when it is not necessary.

When the vehicle initially discovers an access point in its proximity, it will trigger the link estimation method. The method described in previous sections will first predict the upcoming projections of the vehicle's movement using a sequence of time-series previous data. The link estimation time is then calculated through the knowledge of the vehicle's

Algorithm 9.3 Link Estimation method

//Decision of remaining with the current AR or initiating a new registration depending on the updated LET and nearby ARs

Input: $Cache_{AR}, Adptime, cAR, curTime$

```
    procedure LinkUpdate method
2:      newAR ← closest Cache_AR
        Get cAR
4:      if Eq(9.7)(cAR) ≤ Threshold then
            initiate Reg_Update(newAR)
6:      end if
    end procedure
```

future projection and the access point's location. In case the vehicle is stationary, the registration lifetime is set to Ti, which represents the maximum lifetime the vehicle can stay registered with its current access point without update. However, the vehicle may change its movement or direction during its time within an access point communication range.

9.3.3 Preemptive session update

To anticipate the vehicles' movements after setting the session lifetime with the current access point, each vehicle will periodically monitor its changes in movement and the surrounding access points every λ time. If the vehicle detects a difference of threshold α in the current measured session lifetime and the initial one or the vehicle is moving further to the border of its current access point and closer to another router, a new registration process will be triggered. This registration will occur either with a new access point or a re-registration to the current router with an updated session lifetime. The update process and link estimation is presented in Algorithm 9.3.

9.4 Performance Evaluation

In this section, we provide a thorough performance evaluation of the proposed adaptive location management protocol compared to the benchmark protocol, MIPv6 location management protocol in different mobility environments and scenarios.

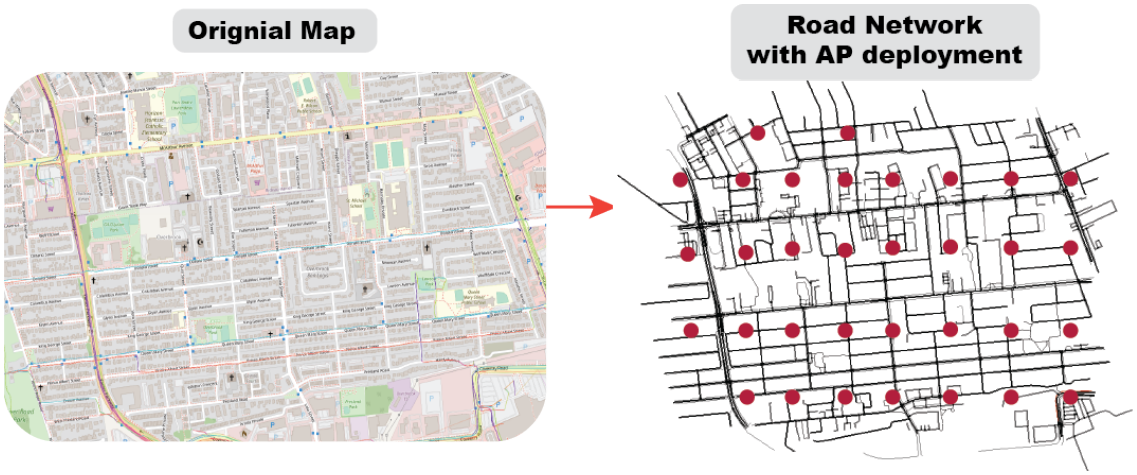


Figure 9.3: Road environment of Ottawa city from OpenStreetMap.

9.4.1 Network setup

Using the network simulator NS-2 and SUMO [118] to generate the mobility traces of Ottawa's Urban scenario extracted from the OpenStreetMap [79], we set the number of vehicles at 100 to 700 vehicles with speeds ranging between 0-30m/s. In Figure 9.3, we show the used mobility environment.

The network is comprised of 34 access points distributed along the network road sections to ensure network coverage over all roads. Initially, each vehicle is attached to its home address and receives a stream of data from a corresponding node with a packet frequency of 250pkt/s and packet size 160$byte/packet$. The used physical protocol is set to be IEEE 802.11p. In Table 9.1, we summarize our simulation parameters, with access point that provide a transmission range of 250 m.

9.4.2 Trajectory prediction accuracy

In this section, we evaluate three time-series models, the Long-Short-Term Memory (LSTM) [87], the General Regression Neural Networks (GRNN) [201], and the Simple Moving Average (SMA) [102, 239]. To evaluate the trajectory predication models, we use the Root Mean

Table 9.1: Simulation parameters.

Parameter name	**Parameter value**
Simulation Time	500 s
PHY/MAC	IEEE 802.11p
Transmission range (m)	250 m
Number of vehicles	700
Simulation Environment	Urban - Ottawa/Downtown
Mobility Model	Krauss Car Following Model
Vehicle speeds	0-30 m/s

Square Error (RMSE) as a means to calculate their accuracies.

$$RMSE = \sqrt{\frac{1}{N}\sum_{0}^{n}(\hat{x} - x)^2} \tag{9.9}$$

We conduct an evaluation of the predictors using Python Keras model and neupy algorithms. The used dataset is extracted from the generated mobility of the previously described environment. We focus the evaluation on an Urban scenario since it is more prone to changes in trajectories than the highway scenario. The generated dataset contains the following information: Time stamp, Vehicle ID, Location (x,y), Speed, and the current access point registered.

Before we evaluate the prediction accuracy, we first study the vehicles' movement over time and the impact of the chosen dataset sampling time, as seen in Figure 9.4, using 255,853 recorded data from 500 vehicles.

Vehicles movement analysis

The changes in vehicles' mobility, displayed in Figure 9.4, shows that speed is the lowest measurement that can be affected by the selected gap size (ie., time-step value). This is due to the fact that vehicle acceleration increases steadily. A vehicle will not be able to increase its speed from 0 to 30 m/s in a span of 5 seconds. This indicates that any gap selection to

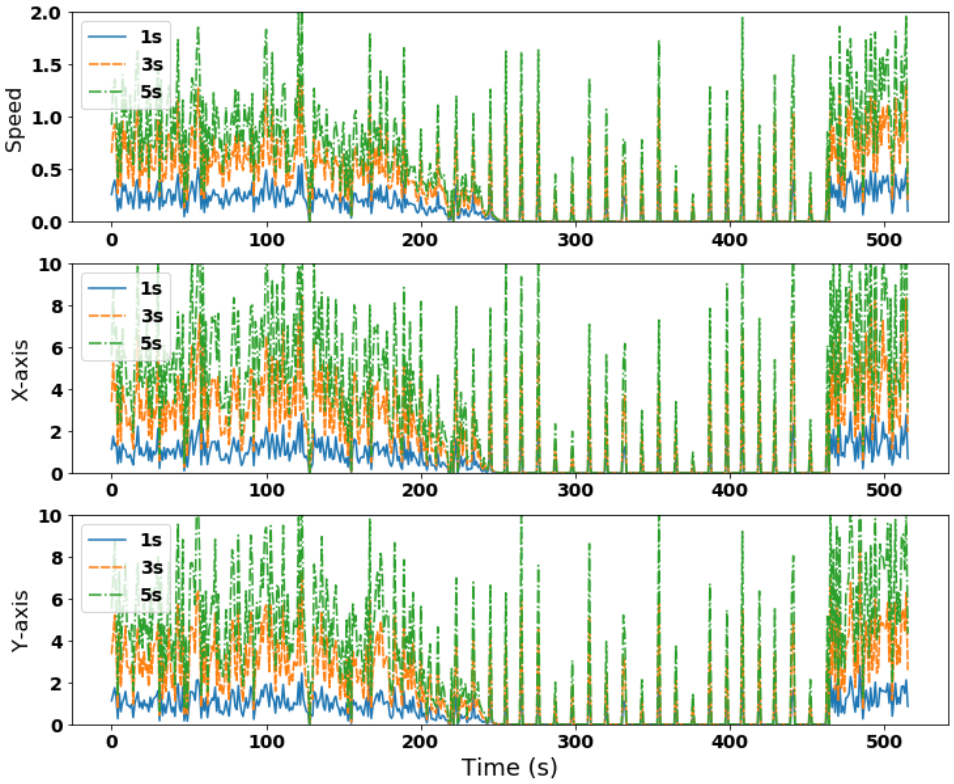


Figure 9.4: Average change in vehicles' mobility with different sampling gaps using Ottawa Urban scenario including 500 vehicles' mobility data.

predict the vehicle speed within a time of less than 5 seconds will not be of much difference. In terms of the vehicle's location, in Figure 9.4, we can see more noticeable changes in the X-axis and Y-axis. The movement projections jump from 0-2 meter difference in 1 second gap up to 0-10 meter difference for 5 seconds gap. Depending on the maximum coverage of access points, having a 2 meter difference and a 10 meter difference in movement will impact the connection between the vehicle and the access point if not considered carefully. Therefore, we need to assess the accuracy of each predictor when given a dataset of different gap values. Henceforth, after thorough evaluation of the average time vehicles reside in each access point communication range and the window size, we select 1, 3, and 5 second gaps as our test cases.

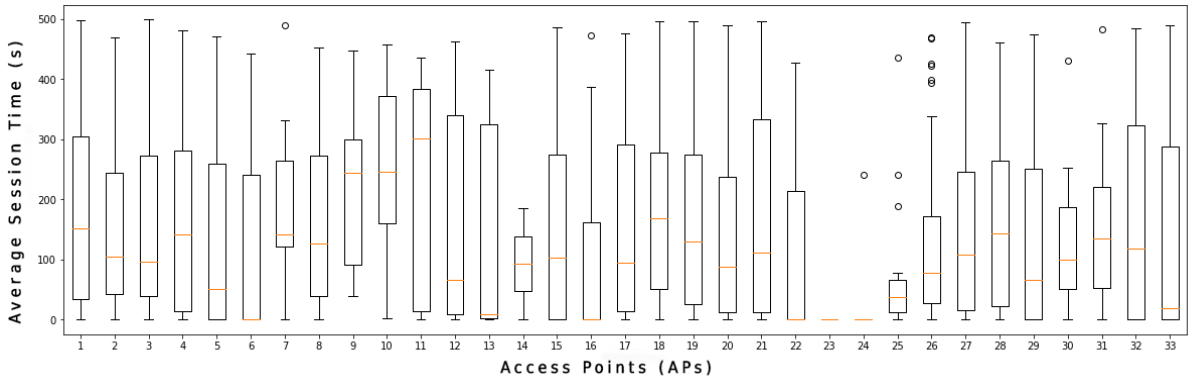


Figure 9.5: Average vehicles' residual time in each access point.

Time window and gap

Following the choice of dataset gaps, it is essential to determine how many batches of data we need in order to predict the next values. In the time-series prediction, a look-back or a window size needs to be chosen wisely to better fit the desired application and scenario. When considering a vehicle's movement within an access point, we can analyse the average session time between a vehicle and an access point and derive an appropriate time-window to the prediction models.

In Figure 9.5, the session time between vehicles and access points is displayed and resulting in an average session time of 130 s, 1.2 s of average minimum, and 426 s of average maximum. If we assume the vehicle will update its location with the current access point every 1 second, this means an average of 130 updates will be made. This number of updates can be significantly reduced if we can estimate the vehicle session duration time within an access point, while taking into consideration the movement projection of the vehicle. To this end, in all cases of 1, 3, and 5 seconds gap window, one can determine the time-window size by considering the cost and accuracy of how far back in time we can look.

Before looking at the accuracy of each predictor, we evaluate the correlation between different vehicles' movement features including time, velocity, directional angle, current access point, x-axis point, y-axis point, and velocity in both directions. Accordingly, the highest relationship is noticed between the vehicles' directional angle and velocity in both directions, while the lowest relationship is presented between the speed and all other features. In addition, the location of the vehicle showed the highest impact on the access point's feature. Since our goal is to predict the vehicles' next projection in terms

Table 9.2: prediction models configuration.

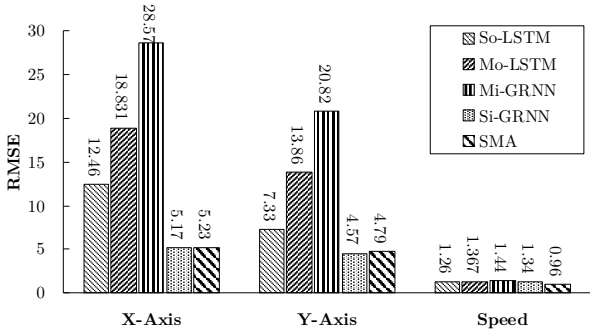
Model type	Parameter values
So-LSTM	Layers[4], Batch[1], Window[5,10], Input[4], Output[1], Train[70% with 33% validation], Test[30% unseen dataset]
Mo-LSTM	Layers[7], Batch[1], Window[5,10], Input[4], Output[3],Train[70% with 33% validation], Test[30% unseen dataset]
Si-GRNN	X-Y-std[0.003], S-sdt[0.05]
Mi-GRNN	std[0.1]

of speed and location, we choose the vehicles' directional angle, speed, and location to be the selected features for the prediction models.

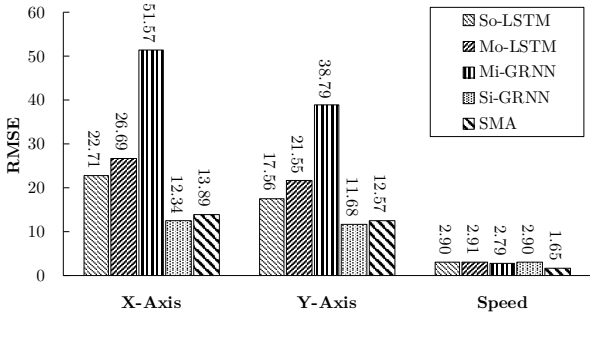
The evaluation of all of the predictors' accuracy is done two fold. We look at the impact of the dataset gap and history time-window, in addition to different variants of each model. The time-window represents the number of look-back steps needed to predict the next features' estimations. We choose the time-window 5 and 10 as a standard for comparisons, since our maximum of 5 gaps and 10 window size will result in 50 time-steps required to reach one prediction. This will result in longer waiting time to derive a result; meanwhile, the vehicle movement will be much more difficult to predict. In the second fold, we evaluate the use of single or multiple models to derive a prediction, as we will discuss in the following section.

Prediction accuracy

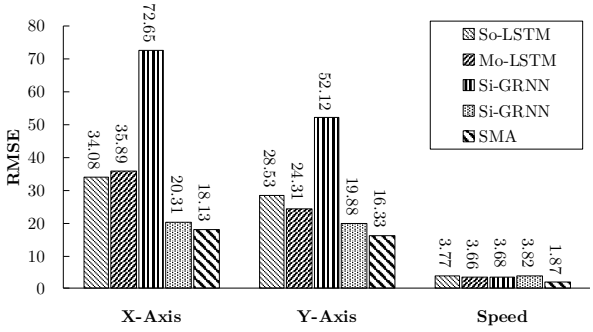
The prediction models' setup parameters of both LSTM and GRNN models is presented in Table 9.2. The parameters were chosen after multiple empirical evaluations of the generated accuracy of each model. In Figures 9.6 and 9.7, we demonstrate the RMSE of all predictors' variants in terms of velocity, x-axis location, and y-axis location. The predictors' variants are described as follows. The LSTM model includes two variants of different parameters, Single feature output-LSTM (So-LSTM) , represents a 4-Layers model, with 1 batch, 4 features input (i.e., velocity, locationX, locationY, and Angle), and 1 Feature output. This means that we created one model for each feature. The Multiple output-LSTM (Mo-LSTM), means one single model with multiple features output and 7



a: Sampling [1s].



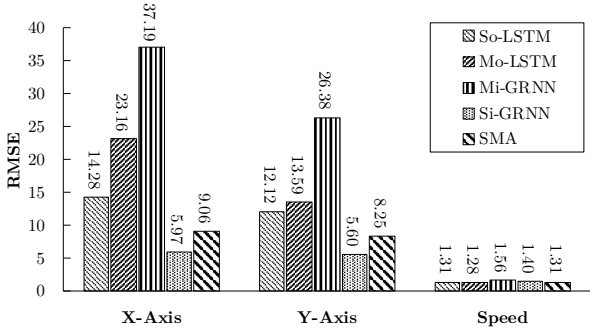
b: Sampling [3s].



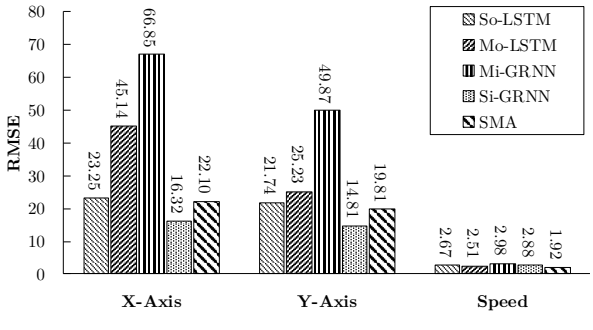
c: Sampling [5s].

Figure 9.6: Prediction accuracy comparison with windows size 5.

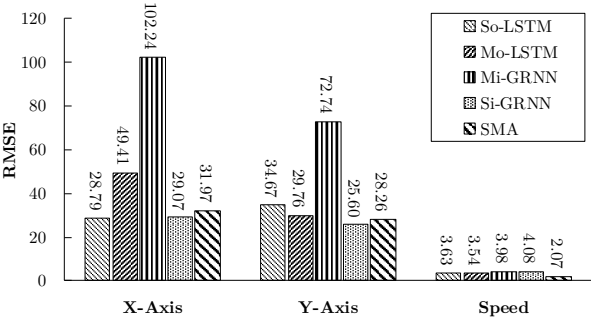
Layers. Furthermore, the GRNN model was divided into two types: single feature input (i.e., Si-GRNN) or multiple (Mi-GRNN) feature input to one feature output; in both cases, this means that each feature will have a separate model. Finally, the SMA predictor was tested on single feature input and output, which means that same as the GRNN model, each feature will require a separate model for prediction.



a: Sampling [1s].



b: Sampling [3s].



c: Sampling [5s].

Figure 9.7: Prediction accuracy comparison with windows size 10.

When we look closely at each gap size, in Figure 9.6, we derive that in all gap cases the Mi-GRNN model with multiple features input showed the worst performance compared to the other models. Meanwhile, Si-GRNN performance indicated better results. Overall, the SMA model reported the lowest RMSE in terms of vehicle's speed in all gap scenarios, and as good as the Si-GRNN in terms of location error, except in the case of gap size 5.

The same analogy presented in window size 10 in Figure 9.7. However, the overall results show that the higher the window size (i.e., from 5 to 10), the higher the prediction error rate. More precisely, the difference in error between 5 and 10 window size and gap size 1 reported an average of 11%, -4%, 9%, 6%, and 22% for all predictor's variants So-LSTM, Mo-LSTM, Mi-GRNN, Si-GRNN, and SMA, respectively. This indicates that the Moving Average predictors is not scalable in terms of window size; the longer the sequence is, the worse the error.

The results obtained on the LSTM model with one and multiple features output in Figure 9.6 and 9.7 shows that having a separate model for each feature results in lower RMSE values over all gaps and window sizes. This indicates that multiple-input-multiple-output models are not suitable with such a dataset. The result of the GRNN model indicates that having multiple features input shows the highest RMSE in comparison to all other models, while the one feature input GRNN model showed the lowest error rate compared to LSTM models and very similar values to the SMA model.

Discussion

In general, one can argue that with both a complicated model such as LSTM and a simpler version of GRNN training set, time to learn and to derive a prediction are required. This also might go through multiple training periods over time, depending on the dataset used and variance of mobility presented in each scenario to avoid high error rates over time. Nonetheless, neural network models have shown their suitability to many applications and scenarios with high accuracy results. When compared to the simple model of SMA, the latter showed to be a simple and effective solution to predicting vehicles' mobility projections with high accuracy. However, SMA is a valid solution only if we consider one-step prediction, while in neural network models the implementation of multiple-step prediction is more valid and efficient.

Alternately, we have considered the idea of evaluating two other possible prediction strategies that are based on the region of interest and on independent vehicles. In the first option, each AP region is equipped with a separate prediction model that will only be designated to vehicles within its proximity. This is because we noticed that vehicles tend to show similar movement behaviors within a given region, and thus the prediction model does not require data trained from further areas/regions. Therefore, we succeeded in reducing the computational and storage cost of single predictor to multiple predictors in each region. Moreover, with this distributed strategy in mind, providing each access point with its own predictor will remove the single point of failure problem that will be present in a centralized prediction entity in a cloud server. This type of prediction strategy is

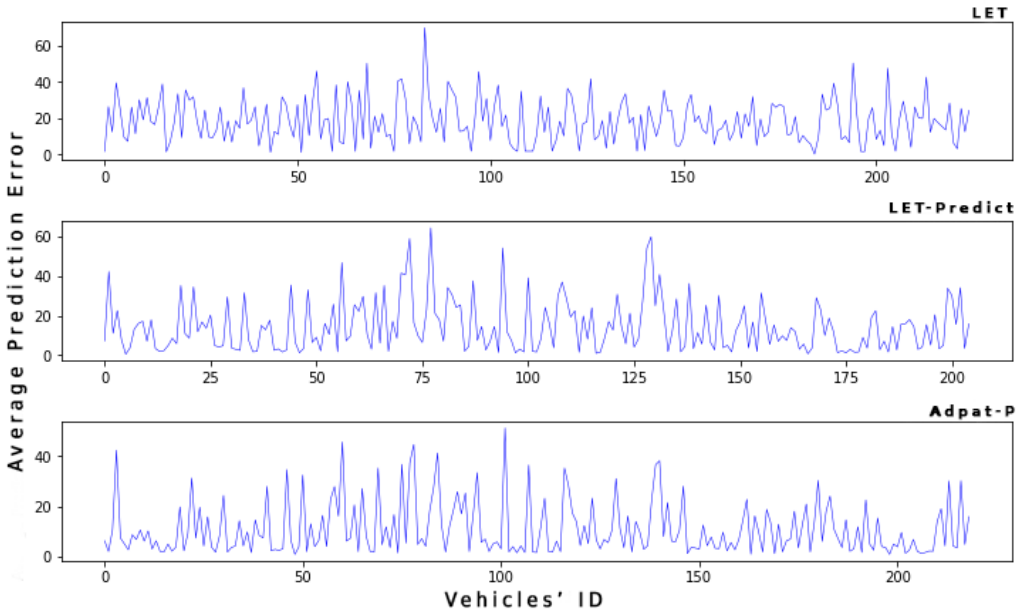


Figure 9.8: Link estimation error in Urban environment.

well suited in mobility management applications, and has been reported to be an effective solution to handover predictions using machine learning methods. However, if we are only looking at the prediction of individual vehicle's movement and simpler models such as SMA or KF, then having each vehicle equipped with such a method is a more effective solution. This type of strategy will provide real-time prediction to each vehicle individually, without the cost of wireless communication to the nearest network units nor the time it takes to receive a reply of prediction.

9.4.3 Link estimation accuracy

While considering the previously evaluated predictors, in the next set of experiments we use a simple moving average predictor to derive a link estimation value, due to its simplicity and efficiency in determining the next values. The estimated error metric, which represents the difference between the estimated and real link lifetime, is chosen to evaluate the accuracy of estimation.

Figure 9.8 demonstrates the estimation error of the original LET, which does not consider future movement projections, Predictive-LET which uses the vehicles' future movement in the evaluation of the link lifetime, and adaptive predictive-LET. The results in

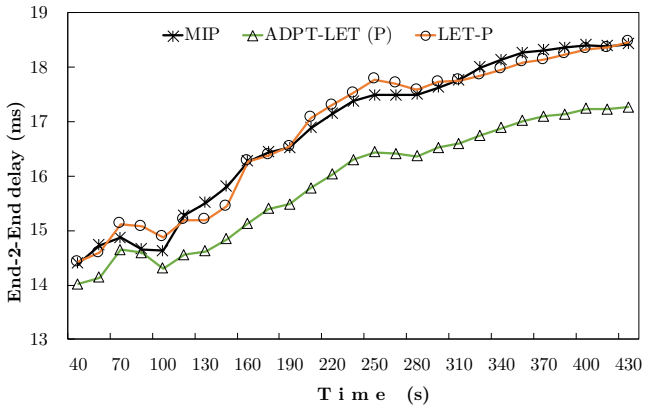
Figure 9.8 reported an average estimated error of $19s$, $15s$, and $11s$ seconds for each of the LET, Predictive-LET, and Adaptive predictive-LET respectively. The reason for this observed performance is that the naive LET model only considers the vehicle's current position and not its future trajectory and speed, thus resulting in inaccurate link estimation time. Meanwhile, the use of adaptive-based predictive LET method showed the lowest error in link estimation.

9.4.4 Network evaluation

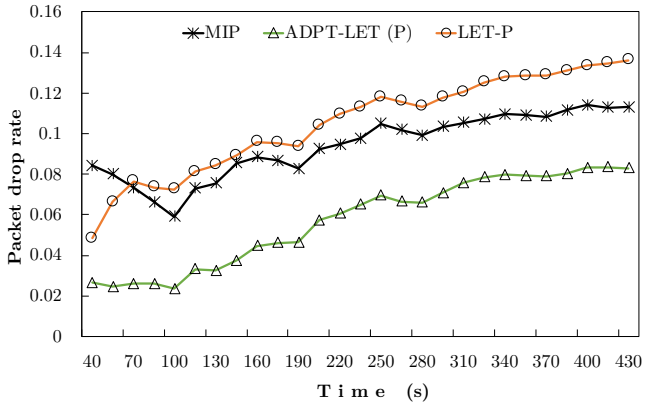
With respect to the network performance, we select the following metrics to evaluate the proposed ADVICE-LOC method.

- *Packet delivery rate*, defines the number of successful packets received by the vehicle from a corresponding node.
- *End-2-End delay*, compute the time required for a packet to be received by the vehicle.
- *Packet drop rate*, defines the number of dropped packets that are not received by the vehicle over the total number of packets sent.
- *Update cost rate*, compute number of actual performed update processes over the total number of update requests.

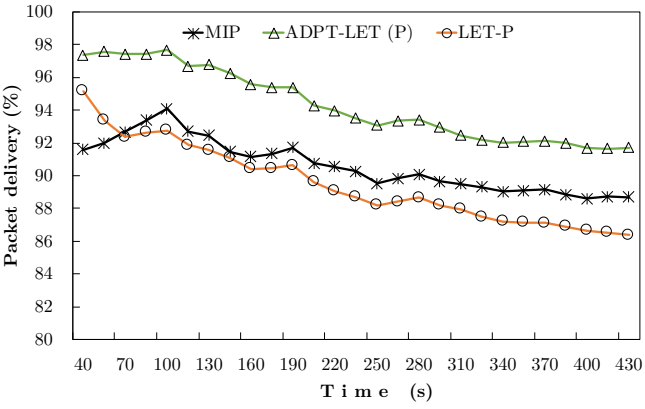
In Figure 9.9, we illustrate the performance of the proposed adaptive vehicle-centric location management protocol compared to the static-based location management protocol of Mobile IP under Ottawa's Urban mobility scenario. We compare the standard MIP location management protocol to the two variants of the link-based location management scheme, LET-P, which estimates the link estimation and sets the session lifetime accordingly, and the adaptive link estimation method, ADPT-LET(P), which preemtively updates the session lifetime depending on the context of vehicles' mobility. The evaluation of the network performance is captured through the simulation setup defined earlier in terms of packet delivery and loss metrics. With respect to the end-to-end delay in Figure 9.9a, the results showed an E2E average of 17 ms for both MIP and LET-P, while the ADPT-LET(P) resulted in 15 ms average delay. Since the LET-P protocol does not consider the vehicles' mobility changes after estimating the initial session lifetime, it has resulted in the vehicle being connected to the access point for longer periods of time than it should be, and a handover will only occur after the vehicle loses connection with the access point. This leads to more data packets being cached in the home agent or foreign network until a



a: End-to-End delay.



b: Packet drop rate.



c: Packet delivery rate.

Figure 9.9: Comparison between the static and the two variance of adaptive mobility management schemes.

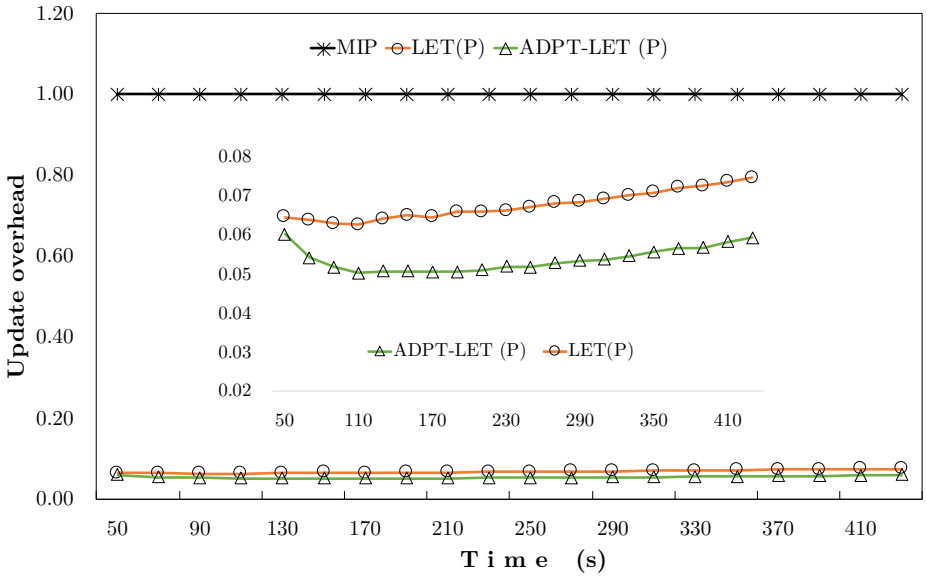


Figure 9.10: Location and registration updates cost.

new route to the vehicle location is established. Meanwhile, the adaptive LET-P was able to preemptively evaluate the link availability with the current access point compared to the signal quality of neighboring access points. This allows the vehicle to perform an early handover process and/or update the session lifetime.

In Figures 9.9b and 9.9c, the average packet drop rate and packet delivery ratio are illustrated. The lowest packet drop rate is found in the adaptive LET-P version of our protocol with an average of 0.05 compared to the standard MIP and the LET-P with 0.09, and 0.10, respectively. The percentage of packet delivery ratio over time resulted in an average of 95% in the adaptive LET-P and 89% for both MIP and LET-P. The results showed that LET-P was not very helpful in elevating the network performance; however, it has only contributed to the reduction of communication overhead during the update process. Figure 9.10 portrays the cost of updating vehicles' location and registration over a time period in which the vehicle is connected to an access point. The results indicate that most unnecessary and redundant communication packets during the update process have reduced dramatically in both LET-P and adaptive LET-P compared to the standard MIP.

9.5 Final Remarks

Wireless connectivity is essential to provide satisfactory services and applications for connected and autonomous vehicles. Various wireless access networks technologies, aim to provide reliable communication solutions for autonomous driving-based applications. However, the increase in number of vehicles and the rapid mobility of vehicles between different access points impact the performance of such wireless networks. In this Chapter, we have presented an efficient adaptive location management protocol for intelligent connected vehicles. Moreover, we have evaluated multiple mobility prediction models in order to estimate the link duration time of each vehicle while adjusting the vehicles' network registration and location update lifetime. The performance of multiple time-series predictors and the estimated link duration time has been studied and evaluated under multiple variants of gaps and window sizes. The results indicated that our proposed adaptive location management scheme significantly reduces the communication overhead while maintaining a low end-to-end delay and high packet delivery rate.

Chapter 10

Conclusion and Future Research Directions

10.1 Contribution Summary

In this thesis, a number of schemes were proposed which utilize mobility-based prediction techniques for intelligent vehicular networks applications and services. A general design framework is proposed to the implementation of mobility-based prediction solutions along with thorough analysis of the existing solutions of different network's deployment architectures. Two prediction-based mobility management schemes were put forth, that aim to reduce the latency of communication and disruption of connections while maintaining prediction accuracy. In the first protocol, HMM and EKF were used to predict the vehicles' upcoming connection beforehand and initiate an early transition process. This process guarantees vehicles' connectivity before losing connections with the current point of access. Then, we improved the protocol in which the topology of the network can adapt dynamically based on the mobility rate between two neighboring access points; thus, reducing the cost of intra-domain handoffs. The second handover triggering scheme reduces the communication and computational cost the previous model by predicting the time-window in which the handover is needed. To further evaluate the prediction of next access point, we developed an online PNN-based prediction method that accurately estimate the next point of access using the vehicles current movement measurements.

The evaluation of the proposed technique has been compared with various known classification techniques and showed high accuracy and time complexity compared to other solutions. Furthermore, the overhead cost of location management protocol has been stud-

ied and a new adaptive solution was proposed, which uses estimated link duration time to dynamically adjust the vehicles' periodic communication with network infrastructure units and registration with their home agent.

10.2 Future Work

The main goal of this thesis was to address mobility-based prediction for intelligent vehicular networks applications and services through various protocols and techniques. This goal has been achieved, as presented in the previous chapters including multiple schemes for mobility management and detection mechanism to support and improve the networks' performance and efficiency. However, Several research directions have not yet been fully explored and studied toward enabling autonomous vehicular networks. In the following, we address a few potential research directions that can be further explored.

Security issues: With the rapid increase of vehicular applications operating in vehicular fog-computing networks (VFC), SDN networks, and HetNet-based vehicular networks, Internet connectivity, and seamless mobility became a vital component of wireless communications. However, an important issue and challenge of security and privacy is always present and not fully considered [90]. The concept of content sharing and data exchange will introduce several security issues, such as authentication, protection, and misuse of protocols [195]. In consequence, vehicles will face more danger from Malware attacks and information sniffing. One possible direction is through the use of group-based authentication and key agreement protocols during the handover process.

Deployment and Configuration Issues: While heterogeneous vehicular networks provide different QoS technologies and access points to choose from depending on the desired services, another issue comes along with such densification of cells. The deployment and configuration methods in 5G-enabled vehicular networks are essential, due to the highly dense deployment and small coverage area of roadside units. Enabling self-optimization and self-configuration systems (SONs) would help in tackling those issues. In SON-based networks, each base station is capable of determining the optimal parameters autonomously [115]. However, effective deployment techniques and networks design, stay an ongoing challenge that the research community has not addressed widely. Furthermore, with the proposed solutions of distributed SDN-based networks, a deployment and selection problem is presented. The question of how to model the distributed system and where to deploy controllers is still an open issue in vehicular networks environment, Especially with the presence of dynamic mobility patterns in vehicle movement and speed. Some works have been done toward the use of smart clustering techniques for distributed SDN-based

networks [216]. Nonetheless, limited research work considered vehicles' mobility patterns and changes in the design of the clustering-based SDN network.

Energy-aware models: Introducing the fog-computing based paradigm to vehicular networks, empowered many applications to deliver services with high scalability and ultra-low latency. Nonetheless, with the rapid expansion in the number of vehicles and the frequent handover, an increase in network load is imminent. Limited battery power constraint in wireless devices is greatly influenced by the number of control and data signaling between different entities in wireless networks. Few works concentrated on energy-efficient solutions in mobility management protocols [195]. However, vehicular networks is a highly dynamic environment, making it a challenge to the design of energy management solutions in such rapidly changing conditions. The common assumption of the vehicles' substantial storage, computing, and power resources might not be valid in the next decade since people are shifting more toward gas-free vehicles (i.e., Electrical vehicles). Thus, making them now inclined to power constraint and requires proper management. In all previous mobility management solutions, the assumption of endless power supply will soon become a new factor in the handover decision, and next network selection process.

Distributed mobility management: The fast-growing requirement for broader internet traffic is driving network carriers to search for options to expand the bandwidth usage without increasing the load on the core network. As mentioned before on SDN-based networks, several solutions were toward a distributed approach [69] or increasing the number of cells and reducing their coverage in HetNet-based networks. A distributed mobility management that allows traffic to be offloaded closer to the edge might be a suitable solution for ultra-dense systems. However, the proposed distributed networks are very primitive and have not been thoroughly investigated to better fit in real-time scenarios.

Integration of Satellite Access in 5G: The integration of satellite access in 5G is another aspect that may be considered in future directions that focuses on enhancing the performance of the vehicular network to meet the specified 5G requirements. In a recent study [230], a seamless handover scheme based on software-defined satellite networking architecture is proposed. The protocol was evaluated on the physical layer and compared to current hard and hybrid handover schemes for satellite networks. In the case of various satellite deployment with different coverage ranges, the network selection is based on the highest cell's quality, precisely, the received signal strength indicator. If the RSSI decreases under a specific limit, the handover method will be initiated. However, the research direction in using satellite access for 5G systems is still immature, and very few works have been done toward using satellite access to improve the performance of the network.

Beyond 5G: The recent advancements toward the deployment of 5G wireless technol-

ogy have witnessed a tremendous increase due to the promoted efficiency and low-latency of such systems. However, many debates have been raised by the research community, which states that the promise of a 5G system will be insufficient due to the abundant growth in network devices and applications. This has initiated a new set of directions toward the vision of beyond 5G systems, which is still unclear [164]. Recent studies considered Terahertz communication to be one of the building blocks to beyond 5G wireless technology [36]. In which, wireless networks can utilize communication bands all the way to 10 THz in comparison to millimeter-wave systems with 10 GHz. With this in mind, mobility management may not anymore have problems in network densification and latency issues — a recent study on millimeter-wave and THz-based smart railway mobility presented in [73]. However, many new challenges are expected to be present before reaching that goal [37] [38].

References

[1] 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. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, Dec 2016.

[2] 3GPP. 3rd generation partnership project. http://www.3gpp.org. Accessed on March 2020.

[3] Kaouther Abrougui, Azzedine Boukerche, and Richard Werner Nelem Pazzi. Design and evaluation of context-aware and location-based service discovery protocols for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):717–735, 2011.

[4] Osama Abumansoor and Azzedine Boukerche. A secure cooperative approach for nonline-of-sight location verification in vanet. *IEEE Transactions on Vehicular Technology*, 61(1):275–285, 2011.

[5] Mamta Agiwal, Abhishek Roy, and Navrati Saxena. Next generation 5g wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 18(3):1617–1655, 2016.

[6] Ian F Akyildiz, Shuai Nie, Shih-Chun Lin, and Manoj Chandrasekaran. 5g roadmap: 10 key enabling technologies. *Computer Networks*, 106:17–48, 2016.

[7] Ian F Akyildiz, Pu Wang, and Shih-Chun Lin. Softair: A software defined networking architecture for 5g wireless systems. *Computer Networks*, 85:1–18, 2015.

[8] Ian. F. Akyildiz and Wenye Wang. The predictive user mobility profile framework for wireless multimedia networks. *IEEE/ACM Transactions on Networking*, 12(6):1021–1035, Dec 2004.

[9] Mohammed Almulla, Yikun Wang, Azzedine Boukerche, and Zhenxia Zhang. A fast location-based handoff scheme for vehicular networks. In *2013 IEEE International Conference on Communications (ICC)*, pages 1464–1468, June 2013.

[10] Mohammed Almulla, Yikun Wang, Azzedine Boukerche, and Zhenxia Zhang. Design of a fast location-based handoff scheme for ieee 802.11 vehicular networks. *IEEE Transactions on Vehicular Technology*, 63(8):3853–3866, Oct 2014.

[11] Theodore Anagnostopoulos, Christos Anagnostopoulos, and Stathes Hadjiefthymiades. Efficient location prediction in mobile cellular networks. *International Journal of Wireless Information Networks*, 19(2):97–111, 2012.

[12] Giuseppe Araniti, Claudia Campolo, Massimo Condoluci, Antonio Iera, and Antonella Molinaro. Lte for vehicular networking: a survey. *IEEE communications magazine*, 51(5):148–157, 2013.

[13] Tewelde Degefa Assefa, Rashedul Hoque, Elias Tragos, and Xenofontas Dimitropoulos. Sdn-based local mobility management with x2-interface in femtocell networks. In *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2017.

[14] Kuldip Singh Atwal, Ajay Guleria, and Mostafa Bassiouni. Sdn-based mobility management and qos support for vehicular ad-hoc networks. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 659–664. IEEE, 2018.

[15] Gagangeet Singh Aujla, Rajat Chaudhary, Neeraj Kumar, Joel JPC Rodrigues, and Alexey Vinel. Data offloading in 5g-enabled software-defined vehicular networks: A stackelberg-game-based approach. *IEEE Communications Magazine*, 55(8):100–108, 2017.

[16] Wei Bao, Dong Yuan, Zhengjie Yang, Shen Wang, Wei Li, Bing Bing Zhou, and Albert Y Zomaya. Follow me fog: Toward seamless handover timing schemes in a fog computing environment. *IEEE Communications Magazine*, 55(11):72–78, 2017.

[17] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[18] Vinay A. Bavdekar, Anjali P. Deshpande, and Sachin C. Patwardhan. Identification of process and measurement noise covariance for state and parameter estimation using extended kalman filter. *Journal of Process Control*, 21(4):585 – 601, 2011.

[19] Zdenek Becvar. Efficiency of handover prediction based on handover history. 4:41–47, 12 2009.

[20] Zdenek Becvar, Pavel Mach, and Boris Simak. Improvement of handover prediction in mobile wimax by using two thresholds. *Computer Networks*, 55(16):3759–3773, 2011.

[21] L. Bellier, K. El-Malki, C. Castelluccia, and H. Soliman. Hierarchical mobile ipv6 (hmipv6) mobility management. RFC 5380, 2008.

[22] S. G. Bhaskar, G. Kumar, and P. Ramana Reddy. Design and analysis of an adaptive handover protocol for 4g networks. *Journal of Shanghai Jiaotong University (Science)*, 20(2):209–217, 2015.

[23] Y. Bi et al. An efficient pmipv6-based handoff scheme for urban vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3613–3628, 2016.

[24] Yuanguo Bi. Neighboring vehicle-assisted fast handoff for vehicular fog communications. *Peer-to-Peer Networking and Applications*, 11(4):738–748, 2018.

[25] Tugce Bilen, Berk Canberk, and Kaushik R Chowdhury. Handover management in software-defined ultra-dense 5g networks. *IEEE Network*, 31(4):49–55, 2017.

[26] Ali Bohlooli and Kamal Jamshidi. A gps-free method for vehicle future movement directions prediction using som for vanet. *Applied Intelligence*, 36(3):685–697, 2012.

[27] Bernard Boigelot. Liege automata-based symbolic handler. www.montefiore.ulg.ac.be/ boigelot/research/lash/, accessed 2020.

[28] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, page 13–16. Association for Computing Machinery, 2012.

[29] A. Boukerche and C. Tropper. A distributed graph algorithm for the detection of local cycles and knots. *IEEE Transactions on Parallel and Distributed Systems*, 9(8):748–757, 1998.

[30] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, 31(12):2838–2849, 2008.

[31] Azzedine Boukerche and Steve Rogers. Performance of gzrp ad hoc routing protocol. *Journal of Interconnection Networks*, 02(01):31–48, 2001.

[32] Azzedine Boukerche, Amber Roy, and Neville Thomas. Dynamic grid-based multicast group assignment in data distribution management. In *Proceedings Fourth IEEE International Workshop on Distributed Simulation and Real-Time Applications (DS-RT 2000)*, pages 47–54, 2000.

[33] Azzedine Boukerche and Jiahao Wang. Machine learning-based traffic prediction models for intelligent transportation systems. *Computer Networks*, 181:107530, 2020.

[34] Andreas Braun and Wolfgang Rid. The influence of driving patterns on energy consumption in electric car driving and the role of regenerative braking. *Transportation research procedia*, 22:174–182, 2017.

[35] Nicola Bui, Matteo Cesana, S Amir Hosseini, Qi Liao, Ilaria Malanchini, and Joerg Widmer. A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques. *IEEE Communications Surveys & Tutorials*, 19(3):1790–1821, 2017.

[36] Sherif Adeshina Busari, Kazi Mohammed Saidul Huq, Shahid Mumtaz, and Jonathan Rodriguez. Terahertz massive mimo for beyond-5g wireless communication. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[37] Sherif Adeshina Busari, Shahid Mumtaz, Saba Al-Rubaye, and Jonathan Rodriguez. 5g millimeter-wave mobile broadband: Performance and challenges. *IEEE Communications Magazine*, 56(6):137–143, 2018.

[38] Angela Sara Cacciapuoti, Kunal Sankhe, Marcello Caleffi, and Kaushik Roy Chowdhury. Beyond 5g: Thz-based medium access protocol for mobile heterogeneous networks. *IEEE Communications Magazine*, 56(6):110–115, 2018.

[39] Andrew T Campbell and Javier Gomez-Castellanos. Ip micro-mobility protocols. *ACM SIGMOBILE Mobile Computing and Communications Review*, 4(4):45–53, 2000.

[40] Clayson Celes, Fabricio A Silva, Azzedine Boukerche, Rossana Maria de Castro Andrade, and Antonio AF Loureiro. Improving vanet simulation with calibrated vehicular mobility traces. *IEEE Transactions on Mobile Computing*, 16(12):3376–3389, 2017.

[41] Pauline ML Chan, Ray E Sheriff, Yim Fun Hu, P Conforto, and C Tocci. Mobility management incorporating fuzzy logic for heterogeneous a ip environment. *IEEE Communications Magazine*, 39(12):42–51, 2001.

[42] Ho-Yuan Chen, Mei-Ju Shih, and Hung-Yu Wei. Handover mechanism for device-to-device communication. In *2015 IEEE conference on standards for communications and networking (CSCN)*, pages 72–77. IEEE, 2015.

[43] Lin Chen and Hui Li. An mdp-based vertical handoff decision algorithm for heterogeneous wireless networks. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6. IEEE, 2016.

[44] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, and Li Zhao. Vehicle-to-everything (v2x) services supported by lte-based systems and 5g. *IEEE Communications Standards Magazine*, 1(2):70–76, 2017.

[45] Huang Cheng, Xin Fei, Azzedine Boukerche, and Mohammed Almulla. Geocover: An efficient sparse coverage protocol for rsu deployment over urban vanets. *Ad Hoc Networks*, 24:85–102, 2015.

[46] Long Cheng, Chengdong Wu, Yunzhou Zhang, Hao Wu, Mengxin Li, and Carsten Maple. A survey of localization in wireless sensor network. *Int. J. Distrib. Sens. N.*, 8(12), Dec. 2012.

[47] Sungsoon Cho and John P Hayes. Impact of mobility on connection in ad hoc networks. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 3, pages 1650–1656. IEEE, 2005.

[48] Mostafa Zaman Chowdhury, Seung Que Lee, Byung Han Ru, Namhoon Park, and Yeong Min Jang. Service quality improvement of mobile users in vehicular environment by mobile femtocell network deployment. In *ICTC 2011*, pages 194–198. IEEE, 2011.

[49] Ji chuan Li, Xiao de Lu, Hui Zhang, Peng cheng Yang, Yu Liu, and Mao sheng Xiang. Moving target detection in the cepstrum domain for passive coherent location (pcl) radar. *Front. Inf. Technol. Electron. Eng.*, 16(9):785 – 795, 2015.

[50] Razvan-Gabriel Cirstea, Darius-Valer Micu, Gabriel-Marcel Muresan, Chenjuan Guo, and Bin Yang. Correlated time series forecasting using multi-task deep neural networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1527–1530, 2018.

[51] Sergio Correia, Azzedine Boukerche, and Rodolfo I. Meneguette. An architecture for hierarchical software-defined vehicular networks. *IEEE Communications Magazine*, 55(7):80–86, 2017.

[52] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[53] Jan G de Gooijer, Bovas Abraham, Ann Gould, and Lecily Robinson. Methods for determining the order of an autoregressive-moving average process: A survey. *International Statistical Review/Revue Internationale de Statistique*, pages 301–329, 1985.

[54] Antonio M de Sousa, Francisco Araújo RC, and Leobino N Sampaio. A link-stability-based interest-forwarding strategy for vehicular named data networks. *IEEE Internet Computing*, 22(3):16–26, 2018.

[55] Eleni Demarchou, Constantinos Psomas, and Ioannis Krikidis. Mobility management in ultra-dense networks: Handover skipping techniques. *IEEE Access*, 6:11921–11930, 2018.

[56] Xiaoyu Duan, Yanan Liu, and Xianbin Wang. Sdn enabled 5g-vanet: Adaptive vehicle clustering and beamformed transmission for aggregated traffic. *IEEE Communications Magazine*, 55(7):120–127, 2017.

[57] Xiaoyu Duan, Xianbin Wang, Yanan Liu, and Kan Zheng. Sdn enabled dual cluster head selection and adaptive clustering in 5g-vanet. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2016.

[58] Mahmoud Hashem Eiza, Qi Shi, Angelos K Marnerides, Thomas Owens, and Qiang Ni. Efficient, secure, and privacy-preserving pmipv6 protocol for v2g networks. *IEEE Transactions on Vehicular Technology*, 68(1):19–33, 2018.

[59] Mica R Endsley. Autonomous driving systems: A preliminary naturalistic study of the tesla model s. *Journal of Cognitive Engineering and Decision Making*, 11(3):225–238, 2017.

[60] Vinko Erceg, Larry J Greenstein, Sony Y Tjandra, Seth R Parkoff, Ajay Gupta, Boris Kulic, Arthur A Julius, and Renee Bianchi. An empirically based path loss model for wireless channels in suburban environments. *IEEE Journal on selected areas in communications*, 17(7):1205–1211, 1999.

[61] Mikael Fallgren, Bogdan Timus, et al. Scenarios, requirements and kpis for 5g mobile and wireless system. *METIS deliverable D*, 1:1, 2013.

[62] Peppino Fazio, Mauro Tropea, and Salvatore Marano. A distributed hand-over management and pattern prediction algorithm for wireless networks with mobile hosts. In *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 294–298. IEEE, 2013.

[63] Fang Feng and D. S. Reeves. Explicit proactive handoff with motion prediction for mobile ip. In *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)*, volume 2, pages 855–860 Vol.2, March 2004.

[64] Huifang Feng, Chunfeng Liu, Yantai Shu, and Oliver WW Yang. Location prediction of vehicles in vanets using a kalman filter. *Wireless personal communications*, 80(2):543–559, 2015.

[65] Pablo Fondo-Ferreiro, Saber Mhiri, Cristina López-Bravo, Francisco J González-Castaño, and Felipe Gil-Castiñeira. Fast decision algorithms for efficient access point assignment in sdn-controlled wireless access networks. *IEEE Transactions on Network and Service Management*, 2019.

[66] Xiaohu Ge, Hui Cheng, Guoqiang Mao, Yang Yang, and Song Tu. Vehicular communications for 5g cooperative small-cell networks. *IEEE Transactions on Vehicular Technology*, 65(10):7882–7894, 2016.

[67] Xiaohu Ge, Zipeng Li, and Shikuan Li. 5g software defined vehicular networks. *IEEE Communications Magazine*, 55(7):87–93, 2017.

[68] Marco Giordani, Marco Mezzavilla, Sundeep Rangan, and Michele Zorzi. An efficient uplink multi-connectivity scheme for 5g millimeter-wave control plane applications. *IEEE Transactions on Wireless Communications*, 17(10):6806–6821, 2018.

[69] Fabio Giust, Luca Cominardi, and Carlos J Bernardos. Distributed mobility management for future 5g networks: overview and analysis of existing approaches. *IEEE Communications Magazine*, 53(1):142–149, 2015.

[70] Győző Gódor, Zoltán Jakó, Ádám Knapp, and Sándor Imre. A survey of handover management in lte-based multi-tier femtocell networks: Requirements, challenges and solutions. *Computer Networks*, 76:17 – 41, 2015.

[71] Raman Kumar Goyal, Sakshi Kaushal, and Arun Kumar Sangaiah. The utility based non-linear fuzzy ahp optimization model for network selection in heterogeneous wireless networks. *Applied Soft Computing*, 67:800–811, 2018.

[72] Marc Green. " how long does it take to stop?" methodological analysis of driver perception-brake times. *Transportation human factors*, 2(3):195–216, 2000.

[73] Ke Guan, Guangkai Li, Thomas Kürner, Andreas F Molisch, Bile Peng, Ruisi He, Bing Hui, Junhyeong Kim, and Zhangdui Zhong. On millimeter wave and thz mobile radio channel for smart rail mobility. *IEEE Transactions on Vehicular Technology*, 66(7):5658–5674, 2016.

[74] Francesco Guidolin, Irene Pappalardo, Andrea Zanella, and Michele Zorzi. A markov-based framework for handover optimization in hetnets. In *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pages 134–139. IEEE, 2014.

[75] Francesco Guidolin, Irene Pappalardo, Andrea Zanella, and Michele Zorzi. Context-aware handover policies in hetnets. *IEEE Transactions on Wireless Communications*, 15(3):1895–1906, 2015.

[76] Raja Gunasekaran, Ananthakrishnan Rajakumar, K. Nirmal Raja, Jeykumar Ramkumar, Sarath K. S. Kumar, Shylesh Umapathy, and Anna. An efficient handover prediction & initiation algorithm for vehicular communication in 4 g wireless networks. 2014.

[77] Shyam M Guthikonda. Kohonen self-organizing maps. *Wittenberg University*, 98, 2005.

[78] Stathes Hadjiefthymiades, Stamatis Papayiannis, and Lazaros Merakos. Using path prediction to improve tcp performance in wireless/mobile communications. *IEEE Communications Magazine*, 40(8):54–61, 2002.

[79] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *Ieee Pervas Comput*, 7(4):12–18, 2008.

[80] Tao Han and Nirwan Ansari. Powering mobile networks with green energy. *IEEE Wireless Communications*, 21(1):90–96, 2014.

[81] Dongsoo Har, Howard H Xia, and Henry L Bertoni. Path-loss prediction model for microcells. *IEEE Transactions on Vehicular Technology*, 48(5):1453–1462, 1999.

[82] Zongjian He, Jiannong Cao, and Xuefeng Liu. Sdvn: Enabling rapid network innovation for heterogeneous vehicular communication. *IEEE network*, 30(4):10–15, 2016.

[83] K. Heine. Unified framework for sampling/importance resampling algorithms. In *2005 7th International Conference on Information Fusion*, volume 2, pages 6 pp.–, July 2005.

[84] E. Hernandez and S. Helal. Predictive mobile ip for rapid mobility. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 684–689, Nov 2004.

[85] Edwin Hernandez and Abdelsalam Helal. Ramon: Rapid-mobility network emulator. In *lcn*, page 0809. IEEE, 2002.

[86] Thomas J Hirschauer, Hojjat Adeli, and John A Buford. Computer-aided diagnosis of parkinson's disease using enhanced probabilistic neural network. *Journal of medical systems*, 39(11):179, 2015.

[87] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[88] Mikael Höök and Xu Tang. Depletion of fossil fuels and anthropogenic climate change—a review. *Energy policy*, 52:797–809, 2013.

[89] E. Hossain, M. Rasti, H. Tabassum, and A. Abdelnasser. Evolution toward 5g multi-tier cellular wireless networks: An interference management perspective. *IEEE Wireless Communications*, 21(3):118–127, June 2014.

[90] Xueshi Hou, Yong Li, Min Chen, Di Wu, Depeng Jin, and Sheng Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, 2016.

[91] Weiming Hu, Xuejuan Xiao, D. Xie, Tieniu Tan, and S. Maybank. Traffic accident prediction using 3-d model-based vehicle tracking. *IEEE Transactions on Vehicular Technology*, 53(3):677–694, May 2004.

[92] Cheng Huang, Rongxing Lu, and Kim-Kwang Raymond Choo. Vehicular fog computing: architecture, use case, and security and forensic challenges. *IEEE Communications Magazine*, 55(11):105–111, 2017.

[93] Chung-Ming Huang, Meng-Shu Chiang, Duy-Tuan Dao, Hsiu-Ming Pai, Shouzhi Xu, and Huan Zhou. Vehicle-to-infrastructure (v2i) offloading from cellular network to 802.11 p wi-fi network based on the software-defined network (sdn) architecture. *Vehicular Communications*, 9:288–300, 2017.

[94] Huawei. 5g a technology vision. http://www.huawei.comlSgwhitepaper/, 2013. Last accesed June 2020.

[95] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[96] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[97] Issam Jabri, Tesnim Mekki, Abderrezak Rachedi, and Maher Ben Jemaa. Vehicular fog gateways selection on the internet of vehicles: A fuzzy logic with ant colony optimization based approach. *Ad Hoc Networks*, 91:101879, 2019.

[98] G. Jeney, L. Bokor, and Z. Mihály. Gps aided predictive handover management for multihomed nemo configurations. In *2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)*, pages 69–73, Oct 2009.

[99] Byungjin Jeong, Seungjae Shin, Ingook Jang, Nak Woon Sung, and Hyunsoo Yoon. A smart handover decision algorithm using location prediction for hierarchical macro/femto-cell networks. In *2011 IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–5. IEEE, 2011.

[100] Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, and Christian S. Jensen. Path prediction and predictive range querying in road network databases. *The VLDB Journal*, 19(4):585–602, Aug 2010.

[101] Xin Jia, Zuo Long Wu, and Hsin Guan. The target vehicle movement state estimation method with radar based on kalman filtering algorithm. In *Applied mechanics and materials*, volume 347, pages 638–642. Trans Tech Publ, 2013.

[102] Bingnan Jiang and Yunsi Fei. Vehicle speed prediction by two-level data driven models in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1793–1801, 2016.

[103] Qi Jiang, Florence Ossart, and Claude Marchand. Comparative study of real-time hev energy management strategies. *IEEE Transactions on Vehicular Technology*, 66(12):10875–10888, 2017.

[104] Stefan Joerer, Michele Segata, Bastian Bloessl, Renato Lo Cigno, Christoph Sommer, and Falko Dressler. A vehicular networking perspective on estimating vehicle collision probability at intersections. *IEEE Transactions on Vehicular Technology*, 63(4):1802–1812, 2013.

[105] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, March 2004.

[106] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the kalman filter to nonlinear systems. volume 3068, pages 3068 – 3068 – 12, 1997.

[107] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.

[108] Anuj Kaul, Li Xue, Katia Obraczka, Mateus AS Santos, and Thierry Turletti. Handover and load balancing for distributed network control: applications in its message dissemination. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE, 2018.

[109] K. Kawano, K. Kinoshita, and K. Murakami. A multilevel hierarchical distributed ip mobility management scheme for wide area networks. In *Proceedings. Eleventh International Conference on Computer Communications and Networks*, pages 480–484, Oct 2002.

[110] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, volume 1 of *Prentice-Hall signal processing series*. Prentice-Hall Inc., A Pearson Education Company, 1993.

[111] Ammara Anjum Khan, Mehran Abolhasan, and Wei Ni. 5g next generation vanets using sdn and fog computing framework. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2018.

[112] Hahnsang Kim and K. G. Shin. Predictive routing of contexts in an overlay network. In *2009 IFIP/IEEE International Symposium on Integrated Network Management*, pages 57–64, June 2009.

[113] Taehyoung Kim, Antoine G Hobeika, and Heejin Jung. Evaluation of the performance of vehicle-to-vehicle applications in an urban network. *Journal of Intelligent Transportation Systems*, 22(3):218–228, 2018.

[114] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[115] Koichiro Kitagawa, Toshihiko Komine, Toshiaki Yamamoto, and Satoshi Konishi. A handover optimization algorithm with mobility robustness for lte systems. In *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1647–1651. IEEE, 2011.

[116] Andreas Köpke, Michael Swigulski, Karl Wessel, Daniel Willkomm, PT Haneveld, Tom EV Parker, Otto W Visser, Hermann S Lichte, and Stefan Valentin. Simulating wireless and mobile networks in omnet++ the mixim vision. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, page 71. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[117] Vasileios Kotsiou, Georgios Z. Papadopoulos, Periklis Chatzimisios, and Fabrice Theoleyre. Label: Link-based adaptive blacklisting technique for 6tisch wireless industrial networks. In *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '17, pages 25–33, 2017.

[118] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187, 2002.

[119] Edward J Krakiwsky, Clyde B Harris, and Richard VC Wong. A kalman filter for integrating dead reckoning, map matching and gps positioning. In *IEEE PLANS'88., Position Location and Navigation Symposium, Record.'Navigation into the 21st Century'.*, pages 39–46. IEEE, 1988.

[120] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[121] Yu-Jen Ku, Dian-Yu Lin, Chia-Fu Lee, Ping-Jung Hsieh, Hung-Yu Wei, Chun-Ting Chou, and Ai-Chun Pang. 5g radio access network design with the fog paradigm: Confluence of communications and computing. *IEEE Communications Magazine*, 55(4):46–52, 2017.

[122] Sławomir Kukliński, Yuhong Li, and Khoa Truong Dinh. Handover management in sdn-based mobile networks. In *2014 IEEE Globecom Workshops (GC Wkshps)*, pages 194–200. IEEE, 2014.

[123] M. Kyriakakos, N. Frangiadakis, L. Merakos, and S. Hadjiefthymiades. Enhanced path prediction for network resource management in wireless lans. *IEEE Wireless Communications*, 10(6):62–69, Dec 2003.

[124] Lassabe, Canalda, Chatonnay, and Spies. Predictivemobility models based on kth markov models. In *2006 ACS/IEEE International Conference on Pervasive Services*, pages 303–306, June 2006.

[125] GM Lee. Software defined networking-based vehicular adhoc network with fog computing. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1202–1207. IEEE, 2015.

[126] Jun-Seob Lee, Jae-Hong Min, Ki-Shik Park, and Sang-Ha Kim. Paging extension for hierarchical mobile ipv6: P-hmipv6. In *The 11th IEEE International Conference on Networks, 2003. ICON2003.*, pages 245–248, Oct 2003.

[127] Yejee Lee, Bongjhin Shin, Jaechan Lim, and Daehyoung Hong. Effects of time-to-trigger parameter on handover performance in son-based lte systems. In *2010 16th Asia-Pacific Conference on Communications (APCC)*, pages 492–496. IEEE, 2010.

[128] Hongjia Li, Song Ci, and Zejue Wang. Prediction handover trigger scheme for reducing handover latency in two-tier femtocell networks. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 5130–5135. IEEE, 2012.

[129] X. Li, M. Parizeau, and R. Plamondon. Training hidden markov models with multiple observations-a combinatorial method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):371–377, 2000.

[130] Martin Liebner, Michael Baumann, Felix Klanner, and Christoph Stiller. Driver intent inference at urban intersections using the intelligent driver model. In *2012 IEEE Intelligent Vehicles Symposium*, pages 1162–1167. IEEE, 2012.

[131] Jianqi Liu, Jiafu Wan, Bi Zeng, Qinruo Wang, Houbing Song, and Meikang Qiu. A scalable and quick-response software defined vehicular network assisted by mobile edge computing. *IEEE Communications Magazine*, 55(7):94–100, 2017.

[132] Jichao Liu, Yangzhou Chen, Jingyuan Zhan, and Fei Shang. Heuristic dynamic programming based online energy management strategy for plug-in hybrid electric vehicles. *IEEE Transactions on Vehicular Technology*, 68(5):4479–4493, 2019.

[133] Jun S. Liu and Rong Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.

[134] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

[135] Tong Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *IEEE Journal on Selected Areas in Communications*, 16(6):922–936, Aug 1998.

[136] A. Lombard, Y. Zheng, H. Buchner, and W. Kellermann. Tdoa estimation for multiple sound sources in noisy and reverberant environments using broadband independent component analysis. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 19(6):1490–1503, Aug. 2011.

[137] Adnan Mahmood, Wei Emma Zhang, and Quan Z Sheng. Software-defined heterogeneous vehicular networking: The architectural design and open challenges. *Future Internet*, 11(3):70, 2019.

[138] Barbara M Masini, Alessandro Bazzi, and Enrico Natalizio. Radio access for future 5g vehicular networks. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pages 1–7. IEEE, 2017.

[139] M. A. A. Masri, A. B. Sesay, and A. O. Fapojuwo. Session state aware handover procedure for voip sessions in heterogeneous wireless networks. In *IEEE Wireless Communications and Networking Conf. (WCNC)*, pages 3011–3016, 2014.

[140] Timothy Masters. *Neural, novel and hybrid algorithms for time series prediction.* John Wiley & Sons, Inc., 1995.

[141] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[142] Salman Memon and Muthucumaru Maheswaran. Using machine learning for handover optimization in vehicular fog computing. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 182–190. ACM, 2019.

[143] A. F. Merah, S. Samarah, and A. Boukerche. Vehicular movement patterns: A prediction-based route discovery technique for vanets. In *2012 IEEE International Conference on Communications (ICC)*, pages 5291–5295, 2012.

[144] Amar Farouk Merah, Samer Samarah, and Azzedine Boukerche. Vehicular movement patterns: a prediction-based route discovery technique for vanets. In *2012 IEEE International Conference on Communications (ICC)*, pages 5291–5295. IEEE, 2012.

[145] Marco Mezzavilla, Sanjay Goyal, Shivendra Panwar, Sundeep Rangan, and Michele Zorzi. An mdp model for optimal handover decisions in mmwave cellular networks. In *2016 European conference on networks and communications (EuCNC)*, pages 100–105. IEEE, 2016.

[146] Feng Miao, Diange Yang, Rujia Wang, Junjie Wen, Ziteng Wang, and Xiaomin Lian. A moving sound source localization method based on tdoa. In *Proc. INTER-NOISE'14*, Melbourne, Australia, Nov. 2014.

[147] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. Evolving deep neural networks. pages 293–312. Elsevier, 2019.

[148] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pages 41–48. Ieee, 1999.

[149] E. Mirzamany, A. Lasebae, and O. Gemikonakli. Using aggregated rsvp in nested hmipv6. In *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 716–721, Aug 2012.

[150] Arunesh Mishra, Minho Shin, and WA Arbaush. Context caching using neighbor graphs for fast handoffs in a wireless network. In *INFOCOM 2004. Twenty-third*

AnnualJoint Conference of the IEEE Computer and Communications Societies, volume 1. IEEE, 2004.

[151] Rupendra Nath Mitra and Dharma P Agrawal. 5g mobile technology: A survey. *ICT Express*, 1(3):132–137, 2015.

[152] Shantidev Mohanty and Ian F Akyildiz. A cross-layer (layer 2+ 3) handoff management protocol for next-generation wireless systems. *IEEE transactions on mobile computing*, 5(10):1347–1360, 2006.

[153] G. Mongillo and S. Deneve. Online learning with hidden markov models. *Neural Comput.*, 20(7):1706–1716, 2008.

[154] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[155] Jung-Min Moon, Jungsoo Jung, Sungjin Lee, Anshuman Nigam, and Sunheui Ryoo. On the trade-off between handover failure and small cell utilization in heterogeneous networks. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 2282–2287. IEEE, 2015.

[156] Vinod Namboodiri and Lixin Gao. Prediction-based routing for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 56(4):2332–2345, 2007.

[157] P. Nath and C. Kumar. User movement-based hierarchical mobility management scheme (um-hmip) for ip networks. In *3rd Intl. Conf. on Computer and Communication Technology*, pages 210–215, 2012.

[158] Emmanuel Ndashimye, Nurul I Sarkar, and Sayan Kumar Ray. A novel network selection mechanism for vehicle-to-infrastructure communication. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 483–488. IEEE, 2016.

[159] Emmanuel Ndashimye, Nurul I Sarkar, and Sayan Kumar Ray. A network selection method for handover in vehicle-to-infrastructure communications in multi-tier networks. *Wireless Networks*, pages 1–15, 2018.

[160] Tien-Thinh Nguyen, Christian Bonnet, and Jérôme Harri. Sdn-based distributed mobility management for 5g networks. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–7. IEEE, 2016.

[161] Quoc-Thinh Nguyen-Vuong, Nazim Agoulmine, and Yacine Ghamri-Doudane. A user-centric and context-aware solution to interface management and access network selection in heterogeneous wireless environments. *Computer Networks*, 52(18):3358–3372, 2008.

[162] Alan O'Connor, Pawan Setlur, and Natasha Devroye. Single-sensor rf emitter localization based on multipath exploitation. *IEEE Trans. Aerosp. Electron. Syst.*, 51(3):1635 – 1651, JULY 2015.

[163] Cheol Oh and Taejin Kim. Estimation of rear-end crash potential using vehicle trajectory data. *Accident Analysis & Prevention*, 42(6):1888–1893, 2010.

[164] John F O'Hara, Sabit Ekin, Wooyeol Choi, and Ickhyun Song. A perspective on terahertz next-generation wireless communications. *Technologies*, 7(2):43, 2019.

[165] Nouri Omheni, Imen Bouabidi, Amina Gharsallah, Faouzi Zarai, and Mohammad S Obaidat. Smart mobility management in 5g heterogeneous networks. *IET Networks*, 7(3):119–128, 2017.

[166] ONF. Software-defined networking: The new norm for networks. Technical report, Open Networking Foundation, April 2012.

[167] Antonino Orsino, Margarita Gapeyenko, Leonardo Militano, Dmitri Moltchanov, Sergey Andreev, Yevgeni Koucheryavy, and Giuseppe Araniti. Assisted handover based on device-to-device communications in 3gpp lte systems. In *2015 IEEE globecom workshops (GC Wkshps)*, pages 1–6. IEEE, 2015.

[168] Yaakov Oshman and Pavel Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Trans. Aerosp. Electron. Syst.*, 35(3):892 – 902, July 1999.

[169] Kaouthar Ouali, Meriem Kassar, and Kaouthar Sethom. Handover performance analysis for managing d2d mobility in 5g cellular networks. *IET Communications*, 12(15):1925–1936, 2018.

[170] S. Pack and Y. Choi. Fast handoff scheme based on mobility prediction in public wireless lan systems. *IEE Proceedings - Communications*, 151(5):489–495, Oct 2004.

[171] Eun Kyoung Paik and Yanghee Choi. Prediction-based fast handoff for mobile wlans. In *10th International Conference on Telecommunications, 2003. ICT 2003.*, volume 1, pages 748–753 vol.1, Feb 2003.

[172] Maria Rita Palattella, Ridha Soua, Abdelmajid Khelil, and Thomas Engel. Fog computing as the key for seamless connectivity handover in future vehicular networks. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1996–2000. ACM, 2019.

[173] Hee-Dong Park, Yong-Ha Kwon, Kang-Won Lee, Young-Soo Choi, Sung-Hyup Lee, and You-Ze Cho. Network mobility management using predictive binding update. In Ajit Pal, Ajay D. Kshemkalyani, Rajeev Kumar, and Arobinda Gupta, editors, *Distributed Computing – IWDC 2005*, pages 560–565, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[174] Alessandra Pascale, Monica Nicoli, and Umberto Spagnolini. Cooperative bayesian estimation of vehicular traffic in large-scale networks. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2074–2088, 2014.

[175] Neal Patwari, Joshua N. Ash, Spyros Kyperountas, Alfred O. Hero III, Randolph L. Moses, and Neiyer S. Correal. Locating the nodes cooperative localization in wireless sensor networks. *IEEE Signal Process. Mag.*, 22(4):54 – 69, Oct. 2005.

[176] R. Pazzi, Z. Zhang, and A. Boukerche. Design of a fast handoff scheme for vehicular mesh networks with directional antennas. In *Proceedings of the First ACM Intl. Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pages 97–100, 2011.

[177] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[178] Mugen Peng, Shi Yan, Kecheng Zhang, and Chonggang Wang. Fog computing based radio access networks: Issues and challenges. *arXiv preprint arXiv:1506.04233*, 2015.

[179] Rong Peng and Mihail L. Sichitiu. Angle of arrival localization for wireless sensor networks. In *Proc. SECON'06*, volume 1, pages 374–382, Sept 2006.

[180] Z. Pi and F. Khan. An introduction to millimeter-wave mobile broadband systems. *IEEE Communications Magazine*, 49(6):101–107, June 2011.

[181] Jonathan Prados-Garzon, Oscar Adamuz-Hinojosa, Pablo Ameigeiras, Juan J Ramos-Munoz, Pilar Andres-Maldonado, and Juan M Lopez-Soler. Handover implementation in a 5g sdn-based mobile network architecture. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2016.

[182] J. Qiao, Y. He, and X. S. Shen. Improving video streaming quality in 5g enabled vehicular networks. *IEEE Wireless Communications*, 25(2):133–139, April 2018.

[183] Yu Qiu, Haijun Zhang, Keping Long, Hongjian Sun, Xuebin Li, and Victor CM Leung. Improving handover of 5g networks by network function virtualization and fog computing. In *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1–5. IEEE, 2017.

[184] A. Quach and K. Lo. Automatic target detection using a ground-based passive acoustic sensor. In *Proc. IDC'99 (Cat. No.99EX251)*, pages 187 – 192, 1999.

[185] Letícia M Raposo, Mônica B Arruda, Rodrigo M de Brindeiro, and Flavio F Nobre. Lopinavir resistance classification with imbalanced data using probabilistic neural networks. *Journal of medical systems*, 40(3):69, 2016.

[186] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[187] Cristiano Rezende, Abdelhamid Mammeri, Azzedine Boukerche, and Antonio AF Loureiro. A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection. *Ad Hoc Networks*, 17:1–17, 2014.

[188] Fredrik Rusek, Daniel Persson, Buon Kiong Lau, Erik G Larsson, Thomas L Marzetta, Ove Edfors, and Fredrik Tufvesson. Scaling up mimo: Opportunities and challenges with very large arrays. *arXiv preprint arXiv:1201.3210*, 2012.

[189] MFM Sabri, KA Danapalasingam, and MF Rahmat. A review on hybrid electric vehicles architecture and energy management strategies. *Renewable and Sustainable Energy Reviews*, 53:1433–1442, 2016.

[190] Ali Safa Sadiq, Kamalrulnizam Abu Bakar, Kayhan Zrar Ghafoor, Jaime Lloret, and Rashid Khokhar. An intelligent vertical handover scheme for audio and video streaming in heterogeneous vehicular networks. *Mobile Networks and Applications*, 18(6):879–895, 2013.

[191] S. Samarah, M. Al-Hajri, and A. Boukerche. An energy efficient prediction-based technique for tracking moving objects in wsns. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5, 2011.

[192] José Santa, Antonio F Gómez-Skarmeta, and Marc Sánchez-Artigas. Architecture and evaluation of a unified v2v and v2i communication system based on cellular networks. *Computer Communications*, 31(12):2850–2861, 2008.

[193] Alois Schlögl. A comparison of multivariate autoregressive estimators. *Signal processing*, 86(9):2426–2429, 2006.

[194] Vishal Sharma, Fei Song, Ilsun You, and Han-Chieh Chao. Efficient management and fast handovers in software defined wireless networks using uavs. *IEEE Network*, 31(6):78–85, 2017.

[195] Vishal Sharma, Ilsun You, Francesco Palmieri, Dushantha Nalin K Jayakody, and Jun Li. Secure and energy-efficient handover in fog networks using blockchain-based dmm. *IEEE Communications Magazine*, 56(5):22–31, 2018.

[196] Fabrício A Silva, Azzedine Boukerche, Thais RM Braga Silva, Linnyer B Ruiz, Eduardo Cerqueira, and Antonio AF Loureiro. Vehicular networks: A new challenge for content-delivery-based applications. *ACM Computing Surveys (CSUR)*, 49(1):1–29, 2016.

[197] Kunwar P Singh, Shikha Gupta, and Premanjali Rai. Predicting carcinogenicity of diverse chemicals using probabilistic neural network modeling approaches. *Toxicology and applied pharmacology*, 272(2):465–475, 2013.

[198] Emmanouil Skondras, Angelos Michalas, and Dimitrios D Vergados. Mobility management on 5g vehicular cloud computing systems. *Vehicular Communications*, 16:15–44, 2019.

[199] Hesham Soliman, Claude Castelluccia, Karim El Malki, and Ludovic Bellier. Hierarchical mobile ipv6 mobility management (hmipv6). Technical report, 2005.

[200] Hao Song, Xuming Fang, and Li Yan. Handover scheme for 5g c/u plane split heterogeneous network in high-speed railway. *IEEE Transactions on Vehicular Technology*, 63(9):4633–4646, 2014.

[201] Donald F Specht et al. A general regression neural network. *IEEE transactions on neural networks*, 2(6):568–576, 1991.

[202] Simo Srkk. *Bayesian Filtering and Smoothing*. Cambridge University Press, New York, NY, USA, 2013.

[203] Enrique Stevens-Navarro, Yuxia Lin, and Vincent WS Wong. An mdp-based vertical handoff decision algorithm for heterogeneous wireless networks. *IEEE Transactions on Vehicular Technology*, 57(2):1243–1254, 2008.

[204] William Su, Sung-Ju Lee, and Mario Gerla. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11(1):3–30, 2001.

[205] Taewan You, Sangheon Pack, and Yanghee Choi. Robust hierarchical mobile ipv6 (rh-mipv6): an enhancement for survivability and fault-tolerance in mobile ip systems. In *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, volume 3, pages 2014–2018 Vol.3, 2003.

[206] Kenichi Taniuchi, Yoshihiro Ohba, Victor Fajardo, Subir Das, Miriam Tauil, Yuu-Heng Cheng, Ashutosh Dutta, Donald Baker, Maya Yajnik, and David Famolari. Ieee 802.21: Media independent handover: Features, applicability, and realization. *IEEE Communications Magazine*, 47(1):112–120, 2009.

[207] Hung-Yi Teng, Ren-Hung Hwang, and Chang-Fu Tsai. A context-aware seamless handover mechanism for mass rapid transit system. In Ching-Hsien Hsu, Laurence T. Yang, Jianhua Ma, and Chunsheng Zhu, editors, *Ubiquitous Intelligence and Computing*, pages 109–123, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[208] Ramona Trestian, Olga Ormond, and Gabriel-Miro Muntean. Performance evaluation of madm-based methods for network selection in a multimedia wireless environment. *Wireless Networks*, 21(5):1745–1763, 2015.

[209] Riccardo Trivisonno, Riccardo Guerzoni, Ishan Vaishnavi, and David Soldani. Sdn-based 5g mobile networks: architecture, functions, procedures and backward compatibility. *Transactions on Emerging Telecommunications Technologies*, 26(1):82–92, 2015.

[210] Elad Tzoreff, Ben Zion Bobrovsky, and Anthony J. Weiss. Single receiver emitter geolocation based on signal periodicity with oscillator instability. *IEEE Trans. Signal Process.*, 62(6):1377 –1385, MARCH 2014.

[211] Elisabeth Uhlemann. Time for autonomous vehicles to connect [connected vehicles]. *IEEE vehicular technology magazine*, 13(3):10–13, 2018.

[212] Sandesh Uppoor, Oscar Trullols-Cruces, Marco Fiore, and Jose M. Barcelo-Ordinas. Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075, May 2014.

[213] Jonathan Vestin, Peter Dely, Andreas Kassler, Nico Bayer, Hans Einsiedler, and Christoph Peylo. Cloudmac: towards software defined wlans. *ACM SIGMOBILE Mobile Computing and Communications Review*, 16(4):42–45, 2013.

[214] Alexey Vinel. 3gpp lte versus ieee 802.11 p/wave: Which technology is able to support cooperative vehicular safety applications? *IEEE Wireless Communications Letters*, 1(2):125–128, 2012.

[215] George Vosselman and Jurrien De Knecht. Road tracing by profile matching and kaiman filtering. In *Automatic extraction of man-made objects from aerial and space images*, pages 265–274. Springer, 1995.

[216] Guodong Wang, Yanxiao Zhao, Jun Huang, and Wei Wang. The controller placement problem in software defined networking: A survey. *IEEE Network*, 31(5):21–27, 2017.

[217] Shangguang Wang, Cunqun Fan, Ching-Hsien Hsu, Qibo Sun, and Fangchun Yang. A vertical handoff method via self-selection decision tree for internet of vehicles. *IEEE Systems Journal*, 10(3):1183–1192, 2014.

[218] N.V.D. Wijngaert and C. Blondia. A predictive low latency handover scheme for mobile ip. In *2005 International Conference on Mobile Computing and Ubiquitous Networking*, volume 63, 2005.

[219] Cheng-Shong Wu, Yan-San Chu, and Chia-Hung Fang. The periodic scan and velocity decision handover scheme for next generation femtocell/macrocell overlay networks. In *2013 International Conference on ICT Convergence (ICTC)*, pages 201–206. IEEE, 2013.

[220] Jin Wu, Jing Liu, Zhangpeng Huang, and Shuqiang Zheng. Dynamic fuzzy q-learning for handover parameters optimization in 5g multi-tier networks. In *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5. IEEE, 2015.

[221] Qiong Wu, Lucas CK Hui, Cheuk Yu Yeung, and Tat Wing Chim. Early car collision prediction in vanet. In *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 94–99. IEEE, 2015.

[222] Xiaoyan Wu and Qinghe Du. Utility-function-based radio-access-technology selection for heterogeneous wireless networks. *Computers & Electrical Engineering*, 52:171–182, 2016.

[223] Henk Wymeersch, Gonzalo Seco-Granados, Giuseppe Destino, Davide Dardari, and Fredrik Tufvesson. 5g mmwave positioning for vehicular networks. *IEEE Wireless Communications*, 24(6):80–86, 2017.

[224] Dionysis Xenakis, Nikos Passas, Lazaros Merakos, and Christos Verikoukis. Mobility management for femtocells in lte-advanced: Key aspects and survey of handover decision algorithms. *IEEE Communications surveys & tutorials*, 16(1):64–91, 2013.

[225] Dionysis Xenakis, Nikos Passas, and Christos Verikoukis. A novel handover decision policy for reducing power transmissions in the two-tier lte network. In *2012 IEEE International Conference on Communications (ICC)*, pages 1352–1356. IEEE, 2012.

[226] Hengheng Xie, Azzedine Boukerche, and Antonio AF Loureiro. A multipath video streaming solution for vehicular networks with link disjoint and node-disjoint. *IEEE Transactions on Parallel and Distributed Systems*, 26(12):3223–3235, 2014.

[227] Huiyuan Xiong, Huan Liu, Ronghui Zhang, Limin Yu, Zhijian Zong, Minghui Zhang, and Zhu Li. An energy matching method for battery electric vehicle and hydrogen fuel cell vehicle based on source energy consumption rate. *International Journal of Hydrogen Energy*, 44(56):29733–29742, 2019.

[228] Yi Xu, Jianhua Zhang, Wenye Wang, Avik Juneja, and Subhashish Bhattacharya. Energy router: Architectures and functionalities toward energy internet. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 31–36. IEEE, 2011.

[229] Xiaoyun Yan et al. Improving flow delivery with link available time prediction in software-defined high-speed vehicular networks. *Computer Networks*, 145:165–174, 2018.

[230] Bowei Yang, Yue Wu, Xiaoli Chu, and Guanghua Song. Seamless handover in software-defined satellite networking. *IEEE Communications Letters*, 20(9):1768–1771, 2016.

[231] Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Te-Yuan Huang, Michael Chan, Nikhil Handigol, and Nick McKeown. Openroads: Empowering research in

mobile networks. *ACM SIGCOMM Computer Communication Review*, (1):125–126, 2010.

[232] Gökhan Yavas, Dimitrios Katsaros, Özgür Ulusoy, and Yannis Manolopoulos. A data mining approach for location prediction in mobile environments. *Data Knowl. Eng.*, 54(2):121–146, August 2005.

[233] Osman NC Yilmaz, Zexian Li, Kimmo Valkealahti, Mikko A Uusitalo, Martti Moisio, Petteri Lundén, and Carl Wijting. Smart mobility management for d2d communications in 5g networks. In *2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 219–223. IEEE, 2014.

[234] Maram Bani Younes and Azzedine Boukerche. Intelligent traffic light controlling algorithms using vehicular networks. *IEEE transactions on vehicular technology*, 65(8):5887–5899, 2015.

[235] Maram Bani Younes and Azzedine Boukerche. An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wireless Networks*, 24(7):2451–2463, 2018.

[236] He-Wei Yu and Biao Zhang. A heterogeneous network selection algorithm based on network attribute and user preference. *Ad Hoc Networks*, 72:68–80, 2018.

[237] Shizhe Zang, Wei Bao, Phee Lep Yeoh, Branka Vucetic, and Yonghui Li. Managing vertical handovers in millimeter wave heterogeneous networks. *IEEE Transactions on Communications*, 67(2):1629–1644, 2018.

[238] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. In *ACM SIGSIM Simulation Digest*, volume 28, pages 154–161. IEEE Computer Society, 1998.

[239] Fan Zhang, Robson E De Grande, and Azzedine Boukerche. Accuracy analysis of short-term traffic flow prediction models for vehicular clouds. In *Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pages 19–26, 2016.

[240] Jun Zhang, Meng Ying Ren, and Houda Labiod. Performance evaluation of link metrics in vehicle networks: A study from the cologne case. In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '16, pages 123–130, 2016.

[241] Shan Zhang, Jiayin Chen, Feng Lyu, Nan Cheng, Weisen Shi, and Xuemin Shen. Vehicular communication networks in the automated driving era. *IEEE Communications Magazine*, 56(9):26–32, 2018.

[242] Shan Zhang, Ning Zhang, Sheng Zhou, Jie Gong, Zhisheng Niu, and Xuemin Shen. Energy-aware traffic offloading for green heterogeneous networks. *IEEE Journal on Selected Areas in Communications*, 34(5):1116–1129, 2016.

[243] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu. Energy-optimal mobile cloud computing under stochastic wireless channel. *IEEE Transactions on Wireless Communications*, 12(9):4569–4581, 2013.

[244] Wenyu Zhang, Zhenjiang Zhang, and Han-Chieh Chao. Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management. *IEEE Communications Magazine*, 55(12):60–67, 2017.

[245] Xin Zhang, Hongbin Li, Jun Liu, and Braham Himed. Joint delay and doppler estimation for passive sensing with direct-path interference. *IEEE Trans. Signal Process.*, 64(3):630 – 640, Feb. 2016.

[246] Yaomin Zhang, Haijun Zhang, Keping Long, Qiang Zheng, and Xiaoming Xie. Software-defined and fog-computing-based next generation vehicular networks. *IEEE Communications Magazine*, 56(9):34–41, 2018.

[247] Hong-Yan Zhao, Jun Liu, Zi-Jing Zhang, Hongwei Liu, and Shenghua Zhou. Linear fusion for target detection in passive multistatic radar. *Signal Processing*, 130:175 – 182, 2017.

[248] Peng Zhao, Xinyu Yang, Wei Yu, Jie Lin, and Duolun Meng. Context-aware multi-criteria handover with fuzzy inference in software defined 5g hetnets. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[249] Kan Zheng, Lu Hou, Hanlin Meng, Qiang Zheng, Ning Lu, and Lei Lei. Soft-defined heterogeneous vehicular network: Architecture and challenges. *IEEE Network*, 30(4):72–80, 2016.

[250] Daming Zhou, Ahmed Al-Durra, Fei Gao, Alexandre Ravey, Imad Matraji, and Marcelo Godoy Simões. Online energy management strategy of fuel cell hybrid electric vehicles based on data fusion approach. *Journal of power sources*, 366:278–291, 2017.