Fukuoka Institute of Technology
FIT 福岡工業大学

DOCTORAL THESIS

Management and Coordination of Resources in Software-Defined Vehicular Networks: Implementation and Performance Evaluation of an Integrated Fuzzy-based System and a Testbed

---

Author:
Ermioni QAFZEZI

Adviser:
Prof. Leonard BAROLLI

INA LABORATORY
GRADUATE SCHOOL OF ENGINEERING
FUKUOKA INSTITUTE OF TECHNOLOGY

2023

Contents

List of Figures

List of Tables

Abstract

In the new era of Internet of Things (IoT), different components of Intelligent Transportation System (ITS) will be linked, thus enhancing safety and effectiveness for road users. In addition, the evolution of conventional Vehicular Ad hoc Networks (VANETs) into Internet of Vehicles (IoV) and Software-Defined Vehicular Networks (SDVNs) will enable real-time data interaction among many entities such as vehicles, infrastructure, pedestrians, cloud data centers and cities. The features of these networks such as programmability and flexibility improve the performance and management of VANETs. However, there are some problems with network scalability, network connectivity, information dissemination and management, security and so on. Also, by having many entities, there are abundant available resources and the exploitation of all resources should be done without compromising the applications requirements.

In this thesis, considering these challenges, we propose an intelligent approach based on Fuzzy Logic (FL) and Software Defined Networking (SDN). We consider many parameters and implement intelligent systems that can be used in different scenarios. However, problems containing many parameters are known as NP-Hard problems. Therefore, FL is used as an adequate method for making decision in real-time, which is very important in VANETs. The proposed approach considers a cloud-fog-edge layered architecture consisting of different capabilities and makes use of an integrated fuzzy-based system implemented in the SDN controllers. The proposed system decides the best layer for a vehicle to handle a certain application, taking into consideration the application requirements and the available connections with the surrounding vehicles. We evaluate the system by extensive simulations. We design and implement a testbed to evaluate its feasibility in a real-life scenario.

The simulation results show that the network overload is decreased and the available resources of the network can be exploited. In addition, the performance of the proposed system is related to the number of considered parameters and complexity. On the other hand, the experimental results demonstrate that the proposed approach is feasible, but a larger-sized testbed is necessary to determine the exact accuracy. We also compared the simulation results with experimental results. The comparison results show that the simulation results and experimental results are close to each other.

The contributions of this thesis are as follows. 1) We give insights about VANETs and the emerging technologies supporting IoV. 2) We implement and show the performance evaluation of a resource management system based on FL for VANETs. 3) We propose and implement many subsystems to evaluate the resource capabilities of cloud-fog-edge layers separately. 4) Implementation of the proposed resource management system in a testbed. 5) Comparison of simulation results with experimental results.

The thesis structure is as follows. In Chapter 1 is presented the background, motivation and structure of the thesis. Chapter 2 are introduced different types of wireless networks. Chapter 3 presents VANETs, the radio access technologies, network architectures, data dissemination, security and privacy. Chapter 4 are described some of the emerging technologies and approaches that enable implementation of Network Function Virtualization (NFV), network slicing, cloud computing and SDN. Chapter 5 is given a detailed explanation of SDN, cloud-fod-edge computing, and their use in VANETs. Chapter 6 introduces intelligent algorithms. Chapter 7 provides basic information of FL theory, fuzzy sets and fuzzy systems. Chapter 8 presents the implemented intelligent fuzzy-based systems. In Chapter 9, we discuss the evaluation results of the proposed systems and the implemented testbed. In Chapter 10, we conclude this thesis and give directions about future work.

Chapter 1

Introduction

This chapter provides an analysis of the challenges that transportation system around the world faces today with the aim to explain how vehicular networks can solve their problems. Then, we give a description about the motivation behind this research work and the objectives it attained. In the end we describe the structure of this thesis and give a short description of each chapter.

1.1 Background

One of the biggest challenge of big cities today is facing congestion and traffic jams. According to a report of European Commission, the cost of road congestion in Europe is estimated as high as 1% of GDP [1]. Some of the countries are more problematic than others. Traffic congestion in United Kingdom has incurred a cost of 24.5 billion Euros, which is 1.6 % of their GDP; one of the highest in the region. Congestion in metropolitan cites in Japan with more than 1 million inhabitant costs about 463.8 billion Yen, which represents more than 61% of total cost of vehicular transport [2]. Another report by Texas A&M Trasportation Institute confirms the high cost of traffic in United States at around $190 billion in a year [3]. Traffic congestion comes at a big cost not only in terms of money, but also in time, environmental pollution, noise, and most importantly traffic accidents. In United States, the average auto commuter spends about 54 hours in congestion [3]. In Tokyo, the average speed on general roads is about 15.7 km/h [4]. Automobile gas emissions are important contributors to ozone layer and public health, including premature deaths, cardiovascular problems and respiratory illness. Due to nitrogen oxide emissions, roughly 160,000 people in Tokyo suffer from asthma [5]. Statistics look even more scary for traffic accidents. Every year around 1.3 million people die in the world because of traffic accidents [6].

For these reasons, Vehicular Ad Hoc Networks (VANETs) have emerged as a help to protect drivers avoid crashes everyday as they are driving on their normal routes or new routes undiscovered before, and therefore make travel safer, more efficient and convenient [7]. There exist many vehicular navigation systems, with Google Maps being the most popular, which recommend alternative routes to vehicles to avoid traffic congestion. Such navigation systems compute different alternative routes in terms of shortest driving distance, shortest driving time and lowest driving cost (in case of toll roads). However, taking into account merely these parameters does not actually achieve the goals of traffic management systems as navigation systems only serve each vehicle individually, thus offering better services only for their drivers. Vehicular Ad Hoc Networks, on the other hand, accomplish substantially bigger goals. They benefit all road users, without any exception, mostly by depending only on content awareness and inter-vehicle communications. Context-awareness is expected to play a key role in VANETs, not only by ameliorating traffic management, but also other important metrics such as the driving experience, the safety of road users, and the environmental impact. The inter-vehicle communication in VANETs enables a broader horizon of awareness for the state of other vehicles in the network and the condition of the surrounding environment. Such data include information about traffic lights, weather conditions, public safety information, and so on. Furthermore, the response time in VANETs is dramatically reduced as vehicles would be notified in real time, almost immediately after some situation occurs, or before the situation occurs even better. For example, it can predict that an accident can happen given the capacity of the route and the future amount of traffic expected.

Many might believe that maintaining and repairing the aging infrastructure is the key for improving transportation systems, but in fact making them more efficient and safer lies beyond maintenance. The evolution of Intelligent Transportation System (ITS) has gone through a long journey and can be partitioned into four significant periods during the history. The main goal of the first period was to improve civil engineering of the transportation system. During this part, importance was given to building roads, highways and sidewalks with the necessary road infrastructure. During the second period, the focus was on improving the mechanical engineering capabilities of vehicles, by producing more powerful and faster cars. The third period has been about electronics incorporated into vehicles, with intelligent sensors installed inside and outside the vehicles which make for a safer and more comfortable driving experience. As for today, having advanced technologies implemented in vehicles, the time when vehicles will continuously communicate to each other and to the transportation system is closer than ever. The focus in the near future is to improve mobile and telecommunication networks within vehicles and other systems in order to move toward a more intelligent and connected transportation system. However, the benefits of VANETs are still on hold until overcoming the technical and economical challenges of deploying them. The main technical challenges are as follows.

Mobility and high scalability management. The high mobility of vehicles and frequent changes in road condition lead to frequent network disconnections among vehicles and between vehicles and the network . On the other side, it is common to have a very large number of vehicles in a very small area. Therefore it is important to create a network that is capable to handle the large number of vehicles and maintain service stability during handover.

Lack of centralized management and coordination entity. VANETs are a subset of Mobile Ad hoc Networks (MANETs), and as such are based on inter-vehicle communication and do not rely on central coordination. Therefore, it hard to have an equitable share of resources which results in collision of transmitted packets and less efficient use of channel resources.

Hard delay constraint and providing good QoS. Different applications have different requirements such as reliability, end-to-end latency, bandwidth, data rate and so on. Given the complex architecture of these networks, the limited network bandwidth and the poor stability of the network topology, it can be hard to satisfy the necessary requirements.

Security and privacy. Having a trustworthy information is important from the receiver point of view, however sharing this trusted information can violate the privacy of the sender. On the other side, the distribution of misinformation from untrustworthy sources can lead to serious problems.

Various communication environment. The environment of vehicles can be urban, inter-urban and highways. Each of them has different characteristics. For example, the vehicles driving inside the city struggle to have a direct line of communication because of trees, buildings and other obstacles.

Resource management. Conventional VANETs, having limited resources, face various challenges in resource management. Given the big amount of data in Internet of Vehicles (IoVs) and the increasingly numbers of vehicles joining these networks, the incorporation of cloud technologies in VANETs is seen as good solution.

Interoperability. VANETs consist of heterogeneous wireless technologies and guaranteeing communication between them is a challenging task. The need for coexistence and interoperability is evident.

The research in VANETs aims to provide new ways that overcome the aforementioned challenges. Cloud computing offers unlimited storage and computational capability that can be accessed from anywhere. The integration of cloud, fog and edge computing in VANETs enables vehicles to send and retrieve huge amounts of data at any time and place, without being concerned about their limited storage capability, satisfying end-to-end latency requirements despite the complex processing of data. The evolution of the 5th Generation (5G) of Cellular Networks marks a huge leap in the advance of VANETs. 5G base stations (5G-NR gNodeB) can serve as a gateway to the Internet and therefore enable big data storage, processing, and analyzing in the cloud infrastructure. Software Defined Networking (SDN), Network Function Virtualization (NFV), Network Slicing, and Information-Centric Networking (ICN), among others, are some of the approaches and paradigms that not only promise to enable the required interoperation but also offer new ways on dealing with the massive data information of the ever-increasing number of Vehicle-to-everything (V2X) services. With the integration of Artificial Intelligence (AI) in VANETs, vehicles will be able to acquire the information from many more sources, and therefore they will be able to take better decisions to avoid potential accidents and enhance the driving experience.

We give a more detailed overview of the mentioned challenges in the following chapters, whereas in the next section we present the main objective of this thesis.

1.2 Motivation and Objectives

Even though the IoV will open many possibilities for a plethora of applications, there are still many challenges that are yet to be addressed [7,10,27,28]. Of the utmost importance is, for example, the management of the abundant information and resources available in these networks. Even a single vehicle generates huge amounts of data and considering the fact that the number of vehicles keeps increasing, managing these networks becomes more difficult. In addition, many new applications that require more and more resources come along continually, leading to increased complexity in network management.

In order to support a wide range of applications which have different requirements, VANETs must be able to manage their available resources efficiently. To cope with the challenges described in subsection 1.1, we propose in this thesis an intelligent architecture based on Fuzzy Logic (FL) and SDN approaches that can efficiently manage Cloud-Fog-Edge (CFE) storage, computing, and networking resources in VANETs. By using FL, the proposed approach can manage the resources in real-time while dealing with imprecision and uncertainty. The main research contributions are summarized as follows.

- The paper presents an integrated system, called Integrated Fuzzy-based System for Coordination and Management of Resources (IFS-CMR), which, different from existing approaches, makes a decision following a bottom-up approach in a cloud-fog-edge architecture.
- IFS-CMR considers the condition of the network created between vehicles, such as the Quality of Service (QoS) in the network and the unused amount of resources, together with the application requirements, to select the best resources for a particular situation.

- IFS-CMR is composed of three subsystems, namely Fuzzy-based System for Assess- ment of QoS (FS-AQoS), Fuzzy-based System for Assessment of Neighbor Vehicle Processing Capability (FS-ANVPC), and Fuzzy-based System for Cloud-Fog-Edge Layer Selection (FS-CFELS), each having a key role in the proposed approach.
- The feasibility of the proposed architecture in coordinating and managing the available VANETs resources is demonstrated by the results of extensive simulations.
- Testbed implementation.
- Comparison of simulation results with experimental results.

1.3 Thesis Organization

This thesis is organized into ten chapters, and a flowchart representing its structure is given in Figure 1.1. Chapter 1 conveys the focus of the thesis, the definition of the problems and identified objectives, the contributions, and the thesis' organization. A synopsis of the following chapters is as follows.

Chapter 2 gives an introduction to Wireless Networks and to the technologies that enable them to meet the demands of emerging applications.

Chapter 3 presents Vehicular Networks. The network architectures, the applications, the radio access technologies, the data dissemination, and the most important aspects of security and privacy, are among the main covered topics. The chapter explains in detail the technologies behind recent developments because not only are these emerging technologies the enablers of the state-of-the-art of vehicular networks but they also enable full implementation of our proposed system.

Chapter 4 describes the technologies that enable vehicular networks to meet the demands of emerging applications.

Chapter 5 particulary describes SDN and CFE computing, as they are the main component of our system.

Chapter 6 takes an overview of Intelligent Algorithms, which includes the principle of each algorithm, the advantages and disadvantages of their use, and several respective applications.

Chapter 7 provides fundamental information regarding FL theory, fuzzy sets, and fuzzy systems. The concept of a linguistic variable, of fuzzy operators, of Membership Functions (MFs), that of a fuzzy rule, and the inference engine process, are described in detail for a comprehensive understanding of the application of FL presented in this thesis.

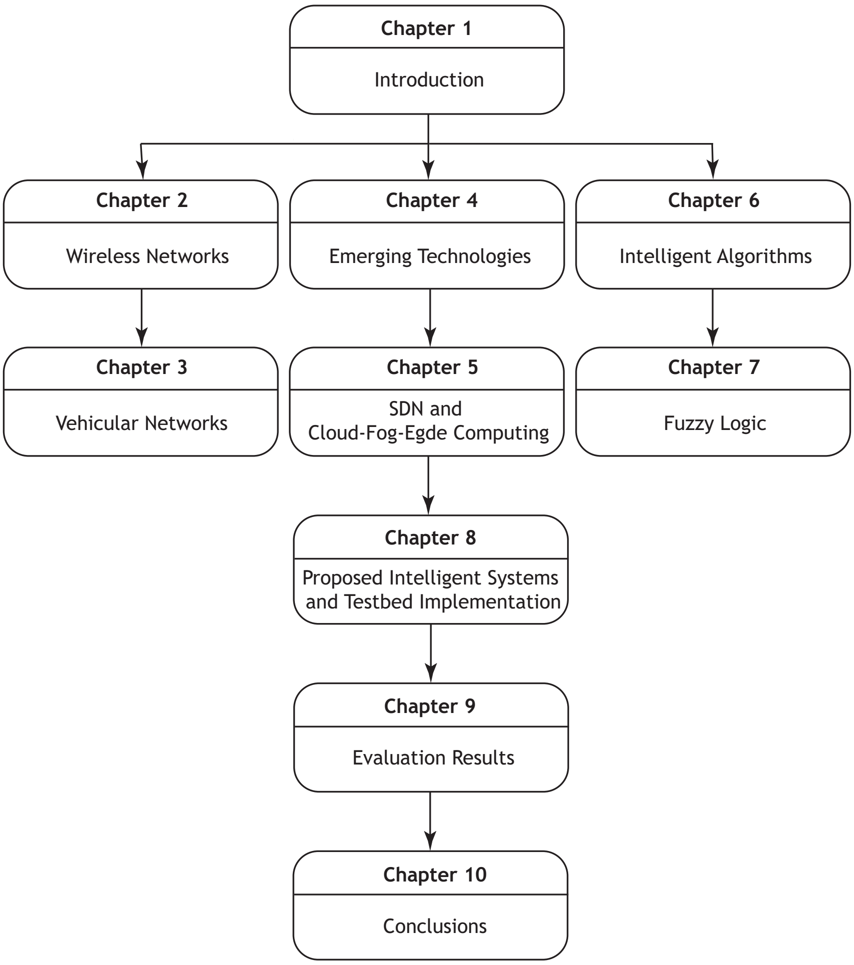


FIGURE 1.1: Thesis structure.

Chapter 8 introduces the implemented intelligent fuzzy-based systems. It includes a description of the proposed systems concerning the objectives and motivation and gives all the implementation details.

Chapter 9 discusses the evaluation results of the proposed systems and the implemented testbed. An analysis of the simulation results for each of the proposed systems is presented with the examination of the experimental results of the implemented testbed to follow. A comparison between relevant systems and between the simulation and experimental results is also drawn.

The last chapter, Chapter 10, concludes the thesis, summarizing the main research findings and future research directions.

Chapter 2

Wireless Networks

Wireless networks have dramatically changed the way we work and access information. A long time ago information was available only through books in the library or in internet coffee centers. Now we can search for anything through our mobile cellphone at no time. People on the go want to do all of the things they do in their offices and at home. With the recent advancement in wireless technologies and embedded systems, cellphone have become smaller in size but more powerful in terms of proccessing capability and the range of applications it provides. However, the increasing need for independence, self-organization and adaptability have emerged the vast expansion of ad hoc networks. Unlike cellular networks, MANETs do not have a central administration; the nodes provide a network infrastructure on an ad hoc basis themselves. In this chapter we present the evolution of wireless networks from infrastructure-based to ad hoc networks.

Wireless Networks refers to a network that makes use of radio signals and infrared signals to share information between nodes of the network. In contrast to wired networks, nodes are able to move freely without losing connection to the network, unencumbered by wires. Wired network make use of dedicated cables that run to each end user and can receive and transmit data simultaneously at any time. On the other hand, devices in a wireless network which use different radio technologies are allowed to operate on the same frequency bands. To avoid package lost and inference, they use half duplex communication–which means that only one device can transmit data at a certain time while the others are listening–in addition to various collision avoidance techniques. People often refer to wireless signals as Wireless Fidelity (WiFi), but this is partially a mistake. Wireless include a wide range of technologies in addition to 802.11, such as ZigBee, Bluetooth, 4G and 5G to name a few.

The components of a wireless networks are the end users devices and the Access Point (AP). The end users are the devices that we use in our everyday life like cellphones, laptops, printers, tablets, desktops, smart TVs and so on, which use wireless data connections. AP transmits beacon messages to the end users devices to advertise the specific networks that it offers connectivity to (known as a Service Set Identifier, or SSID). If the device wants to connect to a certain network, it sends a request to the AP to join and after completing the necessary procedures the device is able to communicate with the network.

Depending on the criteria used, network can be classified in different categories. Based on their coverage area, networks can range from small and personal to large and global. We will explain the technologies implemented in each of them.

Wireless Personal Area Network (PAN): enables users to establish ah hoc communication over the range of a room (10 meters approximately). Everything around our room that uses wireless communication is part of Wireless PAN. For example, using a remote to turn on the TV or open the blinds, computer connection with its peripherals(mouse,monitor, keyboard, speaker, microphone, scanner, printer). It is possible to connect the peripherals with cables. But users usually prefer wireless as they do not want to deal with cables, they want a tedious place to work and they want the freedom to move the devices around. Wireless network technologies used for such short distance communication are Bluetooth, ZigBee, IsDA, 6LoWPAN, NFC and so on. Bluetooth is a low-power technology with a range of 10 meters operating in the 2.4 GHz and uses a method called frequency-hopping spread spectrum for transmission. The controller device using Bluetooth dictates the other devices that are connected what addresses to use, when they can transmit, how long they can transmit, what frequencies they can use, and so on.

Wireless Local Area Network (Wireless LAN): is a local network in limited areas such as within homes, coffee shops, office buildings, factories, or in public spaces such as parks and airports. They are infrastructure-based networks in which every device has a radio modem and an antenna that it uses for communication and rely on networking infrastructures like APs and wireless routers to relay packets throughout the network or with other networks. Ethernet and Wi-Fi are the two most common technologies used in local area networks. WiFi is a IEEE 802.11 standard that runs from 11 Mbps (802.11b) to 7 Gbps (802.11ad). Whereas for wired LAN the most popular is IEEE 802.3, known as Ethernet, which uses copper, coaxial cable, and optical fiber for transmitting the signal in physical layer. Maximum speed reached by optical fibers has been 100 Gbps, which was recorded by NTT telecommunication company in Japan. reach data transmission rates of 319 Tbps.

Wireless Metropolitan Area Network (Wireless MAN): spans to a large distance over the range of several kilometers and enable users to establish broadband wireless connections. Common examples are television broadcasts systems and cellular networks that cover the locations within a city. These networks relay on pre-fixed infrastructure and have transmission lines made of copper wire, coaxial cable, optical fiber and radio links, in addition to routers which forward data packages between different networks. They can also serve as back-ups for wired networks. Popular technologies used in Wireless MAN are WiMAX (IEEE 802.16), GSM, UMTS, LTE, 5G.

Wireless Wide Area Network (Wireless WAN): are the largest distance networks covering a country, a continent or even different continents. They are also infrastructure-based networks with a similar network architecture and transmission lines as Wireless MAN, but have a lower bandwidth and data transmission rate. In addition to Wireless MAN, their network infrastructure includes also satellite links. However, maintaining Wireless WANs is difficult as they cover broad geographical areas and have higher maintenance costs. The Internet, a company that administers offices in different countries or users who want to remotely access the private network of their workplace are good examples of Wireless WAN.

Networks based on the underlined infrastructure can operate in two modes: infrastructure-based networks and ad hoc networks. A simple illustration of these two network architectures is given in Figure 2.1.

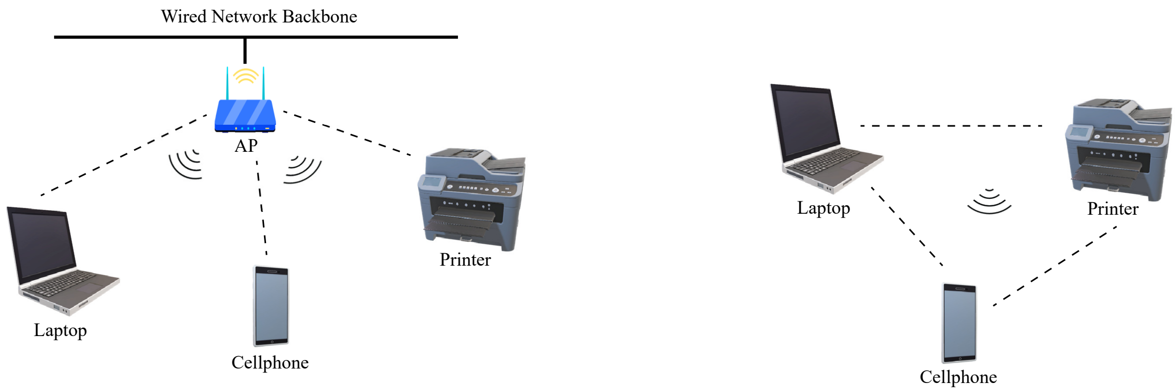


FIGURE 2.1: Comparison between an infrastructure-based network and an ad hoc architecture network.

Infrastructure-based network: Networks that are based in a fixed and wired underlined infrastructure are called infrastructure-based networks (also referred as backbone networks). In these networks, each of the nodes has a prearranged functionality and the communication among nodes follows strict rules. Even two nodes next to each other can not communicate directly, but the data goes through a central AP, a switch or Base Station (BS). For example cellular networks are typical infrastructure-based networks that rely on their core network consisting of PSTN backbone switches, Mobile Switching Centers (MSCs), and BSs.

Ad hoc network: On the other side, ad hoc networks (also referred to peer-to-peer networks) are formed dynamically through the cooperation of an arbitrary set of nodes. The nodes are independent from each other and they create temporary connections with other devices inside their communication range, without using a preconfigured network infrastructure. In other words, nodes connect directly to each other, without the need for a centralized administration like an AP. They behave either as end nodes or router by forwarding packet to other nodes in a multi-hop manner. As an autonomous self-configuring network, they can be fully deployed in any environment. MANET and Wireless Sensor Network (WSN) are typical examples of infrastructureless networks, and also closely related to vehicular networks [8]–[11].

Based of the technologies used for gaining access to network, wireless networks can be classified as mobile networks, satellite networks, MANETs and WSNs.

2.1 Mobile Networks

Mobile networks are by far one of the most successful communication technologies that have had widespread use shortly after their appearance. The current generation of cellular networks has almost no similarity with the technology that its earliest predecessor used, and only a few principles, such as the concept of dividing the area into cells, have remained the same; however, in order to put current advances into context, we will begin this section with a short review of how these networks have changed over the last four decades.

Initially, cellular systems were designed for mobile devices inside vehicles with antennas installed on their roofs. The first generation of mobile networks (1G) had base stations covering large cells and which did not use the available radio spectrum efficiently, so their capacity was very small compared to today's standards. In addition, the mobile devices were large and expensive, and affordable only by business users.

It was the development of digital wireless technologies that made it possible for cellular networks to have the services delivered to the masses through the development of the Global System for Mobile Communications (GSM). This technology marked the launch of the second generation of mobile networks (2G), which introduced increased voice capacity and smaller mobile cellphones. An enhancement of 2G networks came with the deployment of General Packet Radio Service (GPRS) technology, providing packet data capabilities over these networks, which supported new, data-oriented services and applications. Further enhancements of these networks arrived with the implementation of Enhanced Data rates for GSM Evolution (EDGE) that improved the data transmission rates of GPRS. Nevertheless, the mobile data was only in the beginning, and voice calls still dominated the traffic in 2G networks.

The growth of mobile data was slow even with the deployment of first commercial third generation (3G) networks—also known by the name Universal Mobile Telecommunications System (UMTS)—which were introduced in the early 2000s, but in the years leading up to 2010, its use started to increase considerably [12]. Two factors were behind this growth: the availability of enhanced 3G communication technologies, namely High-Speed Packet Access (HSPA) and Evolved HSPA (HSPA+), and the introduction of smartphones that supported numerous applications. Shortly after, the smartphones were everywhere, generating increased traffic, demanding more capacities, and requiring a reduced end-to-end latency. These issues drove the mobile networks to move to their fourth generation (4G), with the Long Term Evolution (LTE) technology denoting the first version of this generation.

Although the current research trend is focused on the fifth generation (5G), and even beyond it, we will also provide in this section a brief introduction to 4G/LTE, since the current standard of 3rd Generation Partnership Project (3GPP) for the deployment of vehicular networks is based on this technology.

4G/LTE

Whilst at the beginning was debated whether LTE should be considered a 4G technology, the controversy no longer exists because later International Telecommunication Union (ITU) gave consent to the use of term 4G to describe technologies whose performance is substantially better than the performance of 3G systems. For comparison, the peak data rates of LTE reach 300 Mbps in the downlink and 75 Mbps in the uplink under ideal signal conditions, as opposed to the last improvement of HSPA+, which delivered 84.4 Mbps and 23 Mbps in the downlink and uplink, respectively. The LTE-Advanced, which came as an improvement of LTE, took these figures to a different level, with downlink rates up to 3000 Mbps and uplink rates up to 1500 Mbps [12]. Moreover, this technology reduced the end-to-end latency in the network to even less than 5ms, proclaiming its feasibility of supporting vehicular communications in real-world scenarios. The standard for supporting vehicular communications was finally specified in 3GPP Release 14 [13], and it also includes the specifications of the LTE-Advanced Pro, which was first introduced in the previous release. LTE-Advanced Pro exceeds the limits of its predecessor, and it is considered a Pre-5G technology.

5G

What 4G has delivered is impressive, but there is more to come from 5G. There have been many use cases and applications whose technical requirements exceed the capabilities of 4G systems introduced over the last years, and 5G is expected to support them with ease. The ITU-R M.2083 [14] classifies them into three usage scenarios:

- Enhanced Mobile Broad-Band (eMBB) which addresses the human-centric applications for a high-data-rate access to mobile services, multi-media content, and data. This scenario fosters new services and applications over smart devices (smartphones, tablets, and wearable electronics). It emphasizes wide-area coverage to provide seamless access and high capacity in hot spots.

- Ultra-Reliable Low-Latency Communications (URLLC) which opens the possibility for mission-critical connectivity for new applications such as automatic vehicles, Smart Grid, and Industry 4.0, which have stringent requirements on reliability, latency, and availability.
- The massive Machine-Type Communications (mMTC) that supports dense connectivity with a very large number of connected devices typically deployed in Internet of Things (IoT) scenarios. The devices such as sensors are low-cost, low-power consumption but typically transmitting a low volume of delay-tolerant data.

The current standard, 5G New Radio (5G-NR), is required to reach peak data rates of 20 Gbps in downlink and 10 Gbps in uplink, but further improvements may reach even higher peak data rates. The technical performance requirements that ITU has set in the ITU-R M.2410-0 report [15] also include a minimum requirement for user plane latency of 4 ms for eMBB and 1 ms for URLLC and support for high-speed vehicular applications up to 500 km/h.

2.2 Satellite Networks

Satellite Networks provide voice, data, video transmission over large geographic areas. They are used for space research, weather prediction, navigation, and so on. For instance, television program broadcasting remotely such as journals providing live reports from war zones are a typical example of satellite system service. Satellites consist of several transponders, which listens to some portion of the spectrum for a transmitted data, they amplify the incoming signal, send back the signal at a difference frequency in order to avoid interference with the incoming signal. They can be categorized on three groups given their orbit distance from Earth: geosynchronous orbit (GEO) at 35800 km, medium-earth orbit (MEO) at 10,000 km, and low-earth orbit (LEO) at 500 to 2,000 km. GEO satellites have the largest coverage area and they move synchronously with the rotational period of Earth. The minimum number required to cover the surface of Earth is 3 GEO satellites. A modern satellite has around 40 transponders, most often with a 36-MHz bandwidth. However there are several challenges providing service by geosynchronous satellites which are: significant round trip propagation delay, high power required to transmit, low data rate which is around 500 Mbps, rain absorbs signal because of their short wavelength.

For these reasons, LEO satellites were chosen as a better alternative. Because of their close distance to Earth they do not require much power and the round trip delay propagation is much less; around 40 milliseconds. They have a fast speed and their rotation period around Earth last 90 minutes. Famous LEO constellations are Iridium and Globalstar. They are suitable for travelers who want to use only 1 phone number while roaming globally and they also cover low-population areas when no terrestrial network can be located. Their latest deployment is providing Internet service at high-speed around the Earth.

2.3 MANETs

MANETs are self organizing networks consisting of a group of mobile nodes that are formed dynamically, in an arbitrary manner, connected via wireless links. Nodes in MANET are autonomous and do not rely on a central administration nor on a pre-existing network infrastructure. They do not have only one specific role on the network, but operate in distributed peer-to-peer mode, generate independent data, act as an independent router and organize themselves according to the circumstances. The mobile nodes are free to move randomly, causing unpredictably changes of topology of the network and continuous changes of intercommunication patterns among nodes. In order to communicate with other nodes residing outside their communication range, it is common to have multi hop communication through intermediate nodes until reaching destination node. Such a network may operate in a standalone fashion, or may be connected to a fixed-backbone network and access Internet services.

Due to their easy deployment, MANETs come in handy for providing services for mobile users in territories with no preinstalled communication infrastructure. They can be set up anywhere at any time. Historically, they have been used for tactical military applications to improve battlefield communications and survivability. The difficult terrains and destructible nature of battlefields means it is suitable to rely on a fixed preconfigured infrastructure. Moreover, wireless link are vulnerable to interference and can not propagate signal beyond line of sight. A mobile ad hoc network creates a suitable framework to address these issues, provides a mobile wireless distributed multihop wireless network without preplaced infrastructure, and provides connectivity beyond line of sight.

2.4 WSNs

Sensors offer significant help in various social problems by converting real-world events into digital data that can be processed, analyzed, stored, and acted upon. A WSN consists of a large number of sensor nodes that operate together to monitor a particular process. These sensor nodes are typically low-cost and can be deployed even in remote and rough terrains beyond human reach. A sensor node is typically composed of a power unit, a sensing module, and a low power transceiver used for data dissemination, in addition to the storage and processing capabilities that help to store, analyze, and fuse the gathered data [16]. The sensors communicate with one another and with BSs through one hop and multi-hop communications using very little power in order to increase the longevity of the network, which has the power consumption its critical challenge.

The advances of wireless communications and semiconductors have helped in increasing the network lifetime leading to widespread use of such networks. Nowadays, WSNs are deployed to monitor and model different phenomena like volcanic activity, structural health condition of tunnels and bridges, heritage buildings conservation, agriculture productivity, forecast flooding, and environmental pollutants, industrial automation process, and many more. Despite their huge potential in many applications, current WSN must deal with challenges such as limited storage and processing capability, limited communication bandwidth, and single-purpose design [17].

WSN will continue to give momentum to many new applications due to the features it provides. In ITSs and vehicular networks, WSNs are seen as a key component of heterogeneous systems cooperating along with other technologies employed in vehicular scenarios; especially, due to the little installation and maintenance costs [18]. They can be deployed along urban roads and highways, intersections, and in parking areas to constantly obtain information and inform the driver beforehand about the weather and road condition, the traffic state, and so forth. Moreover, by connecting the sensed data to the Internet, WSN will enable numerous applications in the IoT, enhancing interaction between humans and the environment.

Chapter 3

Vehicular Networks

Nowadays, there is a high demand from society for more dynamic and adaptive intelligent transportation systems. The ultimate goal of vehicular networks is achieving an accident free environment of drivers and passengers, increase the vehicle utilization, manage high traffic flow, enable green transportation and much more. A significant role will play V2X communication technology, enabling emerging use cases, coorporation with other enabling technologies and networks; for which we will explain in detail in this chapter.

3.1 VANET and MANET

Originally VANET were seen as a sub-category of MANET. Even though, VANET hold some of the characteristics of ad hoc network, they also bring about many new challenges and opportunities. As it was investigated it was realized that they have unique characteristics different from MANET that make us consider them as a separate form of study.

VANETs in contrast to MANETs are able to handle a large network scalability [19], [20]. They consist of a great number of vehicles connected spontaneously via wireless link within an area. During heavy traffic, the distribution of nodes will be very dense. Only a system than can effectively handle such large networks is able to properly cope with a VANET environment. In addition, the environment is much more dynamic, with vehicles entering and exiting the network rapidly. Making these network very hard to predict and having frequent disruption in communication. On the other hand, the high density in VANETs serves to achieve better performance. The vehicles are more likely to form wide clusters which creates a more evenly distribution pattern of nodes across the network, and avoid having local clusters and therefore isolated information spread. Compared to MANETs environment, the cluster of vehicles in VANETs exchange a greater number of messages; hence, provide a better knowledge of the condition of the network.

Another distinct characteristic of VANET is multihop communication. In order to reach the vehicles that are outside the sender's communication range, relaying messages through intermediate nodes is essential,so that information gets propagated even throughout the distant nodes. Due to the high topology change and the high movement speed of vehicles, it is very hard to identify the intermediate nodes that will stay within the network for the needed time to accomplish the desired transmission. This challenge is considered harder in VANET than in MANET as it is very common that the nodes in such dynamic environment will move unexpectedly, and therefore the transmission will not be able to be performed.

VANETs are more prone to security issues compared to MANET. Malicious users can get access to the vehicle, get confidential information of the car and the driver, track the route of the vehicle or even try to change its trajectory. Moreover, they are able to modify messages that are spread throughout the network. Therefore, security and privacy in VANETs need to make improvements to prevent information from being leaked and provide a safe environment.

In VANETs the devices connected do not depend on a battery, instead to maintain their power supply devices are connected directly to the vehicle, which generates electricity as it runs. This allows more computations to be performed by the devices as they will not drain the limited power resource for performing heavy processing tasks related to VANETs. As a result, the system improves its performance by exploiting all the available processing capability offered in the system while not being concerned about suddenly disconnection of devices due to overdrawn batteries. The same holds true also for accessing GPS data. As MANET rely on limited battery resources, accessing GPS data constantly will consume a large amount of their low reserve battery. As a consequence, MANET have access to less accurate location data [21].

3.2 Network Architectures

In the following, we describe the most popular terms in literature that researchers use to describe different vehicular networks architectures.

3.2.1 Vehicular Ad Hoc Networks

It has been two decades since VANETs were first mentioned, and they have been of chief interest to many researchers ever since. VANETs were proposed as a case of MANETs but with the distinctive characteristic that the mobile nodes are vehicles, which, on the other hand, have high mobility and tend to follow organized routes instead of moving randomly.

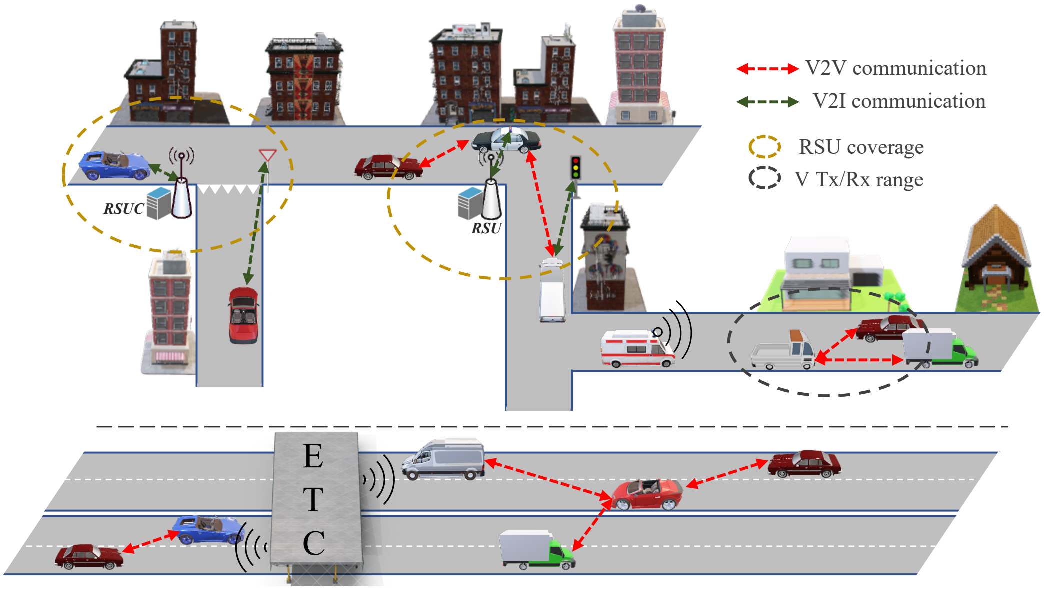


FIGURE 3.1: Illustration of a typical vehicular ad hoc network.

Two types of communications take place in VANETs: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. The infrastructure includes RSUs, road signs, traffic lights, Electronic Toll Collection (ETC), among others. We illustrate the typical VANET communication scenarios in Figure 3.1. These types of communications support many applications spanning from road safety and traffic optimization to rural and post-disaster scenarios connectivity.

3.2.2 Internet of Vehicles

The advances in vehicle manufacturing and communication technology have made it possible for vehicles to be equipped with various sensing platforms, computing facilities, storage, and control units while being connected to any entity (surrounding vehicles, RSUs, pedestrians, network, cloud, and so on) via vehicle-to-everything (V2X) communications [22], [23]. With all these entities connected through vehicles, and with many of them being designed for other purposes as well, the term ad hoc was considered obsolete by many researchers as it does not comprehensively cover the wide range of the technologies integrated within/connecting these entities. Driven by IoT, researchers gave rise to the concept of IoV as a broader concept to better represent the new era of vehicular networks.[1] While there are many types of communications envisioned in IoV, only V2V, V2I, Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) have made their way to get standardized by far. We illustrate these communications with a typical IoV scenario in Figure 3.2.

[1]However, it should be noted that many researchers still use the term VANETs while detailing the differences between conventional and novel VANETs architectures, with the latter being equivalent to IoV.

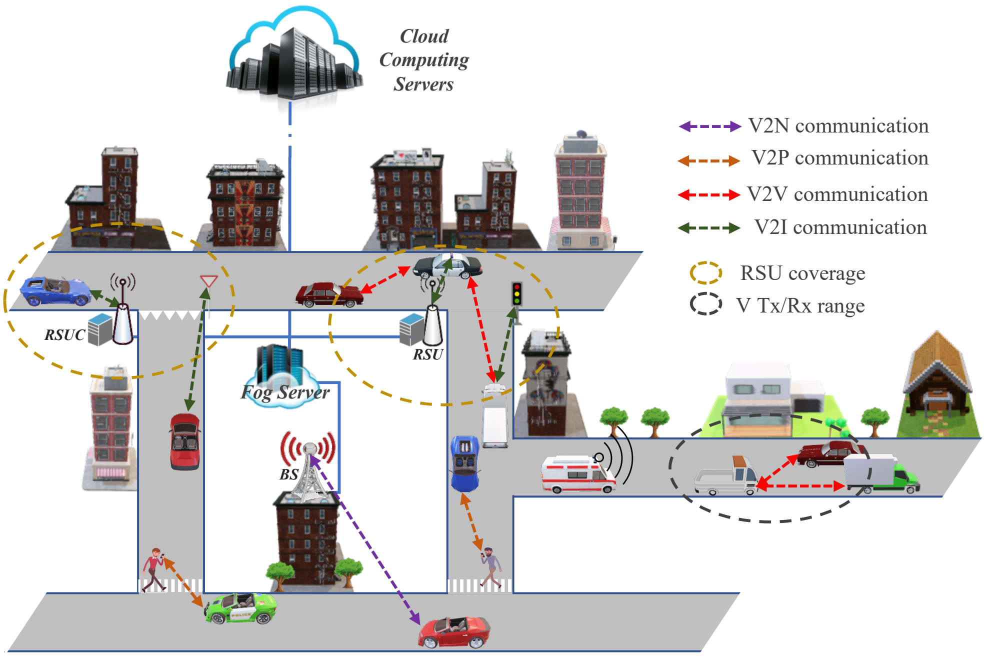


FIGURE 3.2: Illustration of a typical IoV scenario.

3.2.3 Cellular V2X

V2X has evolved from just a concept referring to communication types that take place in vehicular networks to a vehicular communication system that is analogous to these networks on its own [24]. As described above, the former concept relates to the connection of vehicles to all nearby nodes that are equipped with networking functions. The latter represents the technology that can offer a multitude of services characterized by divergent requirements, ranging from a fully automated vehicle traveling in a smart city to streaming 8K videos on an in-vehicle infotainment system. This technology provides an integrated operation between V2V, V2I, V2P, and V2N by leveraging the infrastructure of Cellular Networks, e.g., 4G/LTE, 5G [13], [25], [26]. A counterpart communication technology of C-V2X was IEEE 802.11p [27], but it reached a dead-end in the US as U.S. Federal Communications Commission (FCC) decided to move on from DSRC [28].

Connected Vehicles or *Connected Cars* is a very popular term used in the commercial world to express the new generation of cars that can provide all services described above. In essence, this term is equivalent to the Internet of Vehicles or C-V2X since all these three terms refer to the concept of new cars that can connect and interact with their surroundings while offering a myriad of practical applications and services that bring safety and ease to our daily life.

3.3 Applications

The technology of vehicular networking has seen growth not only in the number of available applications but also in the features they provide, and yet it seems to be an even more enormous potential development ahead. These applications have been seeking to find certain features that can be provided in some particular situations with specific purposes. While the technology to support many use cases of these applications is already available, many other use cases need yet to be supported properly. In addition, new concepts for new applications come along continually as the society is eager for innovation and change.

We surveyed the work of different standardization bodies, including 3GPP [13], [25], [29], ITU [30], European Telecommunications Standards Institute (ETSI) [31], and other launched projects of The 5G Infrastructure Public-Private Partnership (5GPPP) such as Fifth Generation Communication Automotive Research and innovation (5GCAR) [32] and 5G for Connected and Automated Road Mobility in the European UnioN (5GCARMEN) [33] to make an up-to-date list of the applications and use cases that have emerged over the years.

The applications are loosely classified for discussion in this thesis into three main use case groups: 1) Road safety and advanced vehicle control, 2) Advanced traffic management, and 3) Comfort and infotainment services. There is some overlap between the three groups of applications because some use cases have features that belong to more than one of the groups; that is why we differentiate them by considering their primary purpose.

3.3.1 Road Safety and Advanced Vehicle Control

Road safety and advanced vehicle control applications and use cases are presented in Table 3.1. Their purpose is to reduce the risk of driving and save the lives of road users either by directly/indirectly preventing collisions or by mitigating the severity of the accident if for any reason that accident cannot be avoided.

3.3.2 Advanced Traffic Management

The main objective of advanced traffic management applications presented in Table 3.2 is to optimize traffic flow and provide more efficacy for the road systems for the vehicles, pedestrians, and commuters using public transportation. However, these applications may offer secondary benefits, which are not directly associated with traffic management; but, as stated earlier in this section, the classification is made by considering their primary purpose. Correspondingly, several applications belonging to road safety and advanced vehicle control group might as well be listed

TABLE 3.1: Road safety and advanced vehicle control use cases.

No.	Use Case	Source
1	Vehicle Platooning	3GPP/ETSI
2	Automated cooperative driving for short distance grouping	3GPP
3	Cooperative collision avoidance of connected automated vehicles	3GPP
	Longitudinal collision avoidance	ITU-R/ETSI
4	Changing driving-mode	3GPP
5	Emergency trajectory alignment	3GPP
6	Cooperative lane change of automated vehicles / Lane merge	3GPP/5GCAR
7	Road safety services / V2X Road safety service via infrastructure	3GPP
8	Curve Speed Warning	3GPP
9	Enhancing Positional Precision for traffic participants	3GPP
10	Teleoperated support	3GPP
11	eV2X Remote driving	3GPP
	Remote driving for automated parking	5GCAR
12	V2V Emergency Stop / V2I Emergency Stop	3GPP
	Stationary vehicle - vehicle problem	ETSI
13	Emergency notification and personal security/SOS service	ITU-R/ETSI
14	Forward Collision Warning	3GPP
15	Intersection safety information provisioning for urban driving	3GPP
	Intersection collision avoidance	ITU-R/ETSI
16	Collective perception of environment	3GPP
17	Cooperative Adaptive Cruise Control	3GPP/ETSI
18	Video data sharing for assisted and improved automated driving	3GPP
	See-through	5GCAR
	Vision Enhancement Systems	ITU-R
19	V2V Emergency vehicle warning	3GPP
	Back-situation awareness of an emergency vehicle arrival	5GCARMEN
	Emergency vehicle management	ITU-R/ETSI
20	Wrong way driving warning	3GPP/ETSI
21	Pre-crash Sensing Warning / Pre-crash restraint deployment	3GPP/ITU-R
22	Control Loss Warning	3GPP
23	Warning to Pedestrian against Pedestrian Collision	3GPP
24	Vulnerable Road User Safety	3GPP
25	Pedestrian Road Safety via V2P awareness messages	3GPP
	Network assisted vulnerable pedestrian protection	5GCAR
	Vehicle-pedestrian accident avoidance	ITU-R
26	Motorbike awareness	5GCARMEN
	Motorcycle approaching indication	ETSI
27	Event horizon	5GCARMEN
28	Lateral collision avoidance	ITU-R
29	Mixed Use Traffic Management	3GPP
	Vehicle sensors and state sharing	5GCARMEN
	Safety readiness	ITU-R
	Decentralized floating car data	ETSI
30	Public travel security	ITU-R
31	Slow vehicle indication	ETSI
32	Signal violation warning	ETSI
33	Roadwork warning	ETSI

TABLE 3.2: Advanced traffic management use cases.

No.	Use Case	Source
1	Sensor and state map sharing / Map download and update	3GPP/ETSI
	High definition local map acquisition / Route guidance	5GCAR/ITU-R
2	Queue Warning	3GPP
3	V2N Traffic Flow Optimization	3GPP
	Traffic network monitoring and control	ITU-R
4	3D video composition for V2X scenario	3GPP
5	Green Driving	5GCARMEN
6	Travel demand management	ITU-R
7	Incident detection and management	ITU-R
8	Emissions testing and mitigation / Environment analysis	ITU-R/5GCARMEN
9	Electric zones	5GCARMEN
10	Dynamic Speed Limit	5GCARMEN
11	Pre-trip travel information	ITU-R
12	En-route transit information	ITU-R
13	Dynamic ride sharing / Ride matching and reservation	3GPP/ITU-R
	Car rental/sharing assignment/reporting	ETSI
14	Public transportation management	ITU-R
15	Pedestrians route guidance	ITU-R
16	Regulatory/contextual speed limits notification	ETSI
17	Traffic light optimal speed advisory	ETSI
18	Limited access warning, detour notification	ETSI
19	In-vehicle signage	ETSI
20	Vehicle and RSU data calibration	ETSI

here as advanced traffic management applications since they indirectly, often even directly, facilitate traffic management, too.

3.3.3 Comfort and Infotainment Services

The purpose of comfort and infotainment services is to enrich the travel experience by providing on-demand information to the drivers and passengers. Applications such as vehicle/service life cycle management and other applications that are intended to improve the efficiency and productivity of commercial vehicle operations are also included in this group of use cases. All the applications that fall into this definition are presented in Table 3.3.

3.4 Radio Access Technologies

Every application has its own QoS requirements to be satisfied in order to run smoothly and serve its purpose. The exponential growth of the proposed applications has proportionally increased the complexity of meeting their QoS demands due to the limited allocated radio spectrum. Various radio access technologies have been proposed over the past years to exploit every single slot of the allocated radio

TABLE 3.3: Comfort and infotainment services use cases.

No.	Use Case	Source
1	Automated Parking System	3GPP
	Parking management	ITU-R/ETSI
2	Video streaming	5GCARMEN
3	En-route driver information	ITU-R
4	Personalized public transportation	ITU-R
5	Vehicle administration	ITU-R
6	Fleet management	ITU-R/ETSI
7	Vehicle preclearance	ITU-R
8	Automated roadside safety inspections	ITU-R
9	Electronic payment services / Electronic toll collect	ITU-R/ETSI
10	Point of Interest notification	ETSI
11	Insurance and financial services	ETSI
12	Media downloading	ETSI
13	Remote diagnosis and just in time repair notification	3GPP/ETSI
14	Vehicle software/data provisioning and update	ETSI
15	Stolen vehicle alert	ETSI
16	Tethering via Vehicle	3GPP
17	Personal data synchronization	ETSI

spectrum, and more are under development. The radio access technologies range from WiFi and DSRC to cellular networks and cognitive radio and are discussed in the following subsections.

3.4.1 Short-Range Radio Technologies

For many years short-range radio technologies were considered to be the right track for successfully deploying vehicular networking applications. Not only were technologies such as WiFi [34], Bluetooth [35], Zigbee [36] that use the Industrial, Scientific and Medical (ISM) frequency bands put forward for such purpose, but there was also a dedicated spectrum for vehicular communications known as DSRC spectrum for Wireless Access in Vehicular Environments (WAVE). DSRC/WAVE is defined in IEEE 802.11p [37] and in the IEEE 1609 family of standards [38], and in the last two decades, it was seen as one of the wireless technologies that could potentially meet the low latency requirement for safety applications. However, this technology saw its end in the United States as FCC decided to split the 75 MHz of DSRC spectrum (5.850 - 5.925 GHz), allocating the lower 45 MHz of the band for Wi-Fi and other unlicensed uses and the upper 30 MHz for ITS that must use C-V2X technology [28]. Europe, on the other hand, did not abolish this technology. Instead, they adopted a technology-neutral approach that supports both Cooperative ITS (C-ITS) and C-V2X [39].

3.4.2 Cellular Networks

Although the use of cellular networks infrastructure for vehicular networks was proposed since when 3G was becoming a reality, or even earlier when GSM made its appearance and some basic applications such as tracking of stolen vehicles were believed to be successful by leveraging this technology, it is just recently that cellular networks have come into play as a true game-changer in enabling present and future vehicular networking applications. The shift started with the LTE and with 5G now, it is closer than ever. This technology is named C-V2X and is specified in the 3GPP Release 14 [40] for LTE, whereas the support of 5G and the specification of the service requirements are specified in the 3GPP Release 15 [41]. While the releases provide necessary technical details for the deployment of this technology, 5GAA is the player making indispensable efforts to make it a reality.

C-V2X supports both short- and long-range communications. The former is known as the direct mode or Sidelink and uses the PC5 interface, whereas the latter is known as network mode or UP/Downlink and is implemented over the Uu interface for LTE and 5G NR Uu URLLC [29]. Direct mode is essentially V2V, V2I, and V2P, while network mode is V2N communication[2]. These communication modes are introduced at the beginning of this chapter and illustrated in Figure 3.2.

3.4.3 White Spaces and Cognitive Radio

White spaces usually refer to the band of frequencies of the radio spectrum that are unused due to technical reasons, such as the purpose of acting as a guard band between used radio bands to avoid interference. However, this term refers to all the unutilized radio bands, including the radio which has never been used or other frequency bands left unused due to technical changes, e.g., abolition of some radio communication technology.

TV white space, for instance, which is the 470-790 MHz frequency band in the European Union, is currently unused because of abandoned TV channels and is considered for wireless broadband access since it holds rich potential for expanding broadband capacity due to superior propagation characteristics and large penetration performance. The use of TV white space is also investigated for vehicular networks, and the research results promote it as a way forward for the development of connected vehicles [42]–[46].

Cognitive radio is another promising idea envisioned to solve the problem of scarce spectrum. It is a radio technology that allows dynamic programming, configuration, and management of the radio spectrum to enable users to communicate

[2]Network mode also includes other types of communications where the vehicles are not a factor, such as Pedestrian-to-Network (P2N) and Infrastructure-to-Network (I2N).

through the best radio channels in their vicinity while avoiding interferences and congestions. The novel approaches and the research challenges associated with the use of cognitive radio technologies in vehicular networks are surveyed and presented in several research papers [47]–[49].

3.5 Information Dissemination

When vehicles are within the range of cellular antennas, communication is done via the network. However, as we mentioned in the above sections, vehicles can also communicate with one another or with other entities directly, especially when there is no possibility of connecting to a BS. The communication is done via the PC5 interface following the 3GPP specification. This communication type is considered ad hoc mode, and as such, it requires communication protocols that can provide the resiliency and robustness requested in the information dissemination process.

Many communication methods and protocols are proposed over the years, grouped mainly according to the strategy of transmitting the information to the desired destination. For example, these methods can be distinguished as transmitting data from a single source to a single destination node, to a set of nodes inside an area, or to all nodes in the network. These transmission strategies are known as unicast, multicast/geocast, and broadcast, respectively.

- The unicast strategy refers to point-to-point communications, which means that data is transmitted from a single source to a single destination node. This strategy is achieved through a hop-by-hop greedy forwarding mechanism that relays the information immediately or a carry-and-forward mechanism that stores the data in case of a lack of continuous connectivity and forwards it when a decision is made.
- The multicast/geocast strategy is used to deliver data from a single node to a set of nodes that lie within a specific geographical region, also known as zone of relevance. Geocast is considered among the most feasible approaches for safety-related applications in vehicular networks since it can inform all nodes traveling close to the event location. Beacon messages with Cooperative Awareness Message (CAM) [50] and Decentralized Environmental Notification Message (DENM) [51] are typical examples. The former provides periodical information of a vehicle to its neighbors about its presence, position, speed, etc. whereas the latter is an event-triggered message delivered to alert road users of a hazardous event.

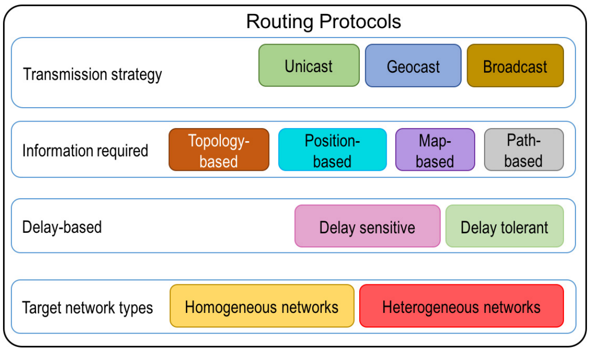


FIGURE 3.3: Taxonomy of routing protocols in vehicular networks.

- Broadcast strategy disseminates information to all vehicles in the network without exception. This approach can be used for data sharing, weather information, road condition, traffic, entertainment, and advertisement announcements.

The classification based on the transmission strategy is the most common way to distinguish the communication protocols, yet it is not the only manner. Cheng et al. [52] present other categories of routing protocols based on different perspectives such as the information required, delay-based information, and the network type target. A summary of these classification manners is given in Figure 3.3 [53]. The information required category is based on the type of information required to perform routing, which can be information related to topology, position, path- or map-based. Delay-based information disseminates data in real-time for critical applications or uses a carry-and-forward mechanism that stores the data in the buffer and transmits it once a vehicle comes within the transmitter's communication range. Whereas, the target network can be categorized as homogeneous network where the communication relies on single network architecture and heterogeneous network which includes various radio access technologies.

3.6 Security, Privacy and Trust

Developing and implementing security, privacy, and trust management solutions is a challenging task for all kinds of computer networks and doing so for vehicular networks is no exception. In fact, the unique characteristics of these networks make the development and implementation of these solutions an even more challenging issue. Many authors have researched different methods that meet the needs of security, privacy, and trust management of vehicular networks. Several surveys covering vehicular networks' requirements, challenges, existing and possible types of threats, and corresponding solutions have been published over the years. These surveys are listed in Table 3.4 for further reading since the goal of this section is

TABLE 3.4: Recent surveys related to security, privacy, and trust.

Survey	Aspect(s)	Contribution	Year
[54]	Privacy	Comprehensive overview of existing privacy-preserving schemes	2015
[55]	Trust	An adversary-oriented survey of different trust models and their evaluation against cryptography	2016
[56]	Security	Overview of security challenges and requirements, and a novel classification of different attacks along with their corresponding solutions	2017
[57]	Security	A review of services, challenges and security threats evolving from the software-defined approach	2018
[58]	Security, Privacy, Trust	A Survey covering well-studied topics of security, authentication schemes, location privacy protection mechanisms, and existing trust managements models	2019
[59]	Security	Deep analysis of various security aspects such as requirements, challenges, and attacks together with the evaluation of their respective solution	2019
[60]	Security, Privacy	A comprehensive review of various research works that address privacy, authentication and secure data dissemination	2020
[61]	Security, Privacy	Introduces security and privacy aspects of C-V2X, and discusses open security challenges and issues	2020
[62]	Trust	A review, analysis, and comparison of current trust establishment and management solutions, and a discussion of future opportunities	2021

to provide basic notions related to security, privacy, and trust rather than a deep analysis of research conducted in this direction.

3.6.1 Security

Secure communication between all involved entities in vehicular networks is of the utmost importance since any successful attack from a malicious hacker could result in a traffic accident. Therefore, a good security framework must consider the basic security requirements in terms of authentication, availability, confidentiality, integrity, non-repudiation, etc., and protect these services from different threats and attacks. There are different ways to classify the attackers and the attacks. The attackers are classified as, for example, active/passive, malicious/rational, insider/outsider, or local/extended attacker. Attacks, on the other hand, can be grouped based on the targeted service, based on the communication mode they hit (V2V, V2I, V2P, V2N), or even based on the entities (hardware or software, authorities, vehicles, infrastructure, or other members) they affect. Some of the most hazardous attacks in vehicular networks are Denial of Service, Distributed Denial of Service, Sybil attack, Black and Gray Hole, Wormhole, Injection of erroneous messages, Replay, and Eavesdropping.

Details of these attacks, their respective solutions, and also many other identified attacks can be found in the surveys presented in Table 3.4.

3.6.2 Privacy

Only dedicated individuals should have the right to access and control the vehicle information and decide what will be communicated to others. One of the most common methods to protect the privacy of individuals is anonymity. Anonymity is defined as the state of being not identifiable within a set of subjects, which can be provided by employing an authentication scheme. Based on the cryptographic mechanisms, the following categories of anonymous authentication schemes can be identified: public-key infrastructure, symmetric cryptography, identity-based, group, and certificateless signatures.

3.6.3 Trust

Trust management deals with the trustworthiness of the received information and the trustworthiness of the vehicles that have sent that information. There are three popular trust management models: entity-centric, data-centric, and combined trust models. In order to establish an effective trust management model, some properties need to be considered, such as decentralization, real-time constraints, information sparsity, scalability, and robustness. Looking at the growth and feasibility of blockchain technology, perhaps vehicular networks are not that far from establishing the long-awaited trust model that addresses those existing challenges [63].

Chapter 4

Emerging Technologies

Due to intensive developments of vehicular networks, transportation will not be the same in the future. As we saw in the previews chapter, it is foreseen that vehicles will be able to communicate with each other and the surroundings in order to have a better coordination for preventing traffic and potential accidents from happening. Moreover, smart cities will enable the connection of different systems to improve the quality of life for citizens and to achieve different business objectives. However, this can not be achieved without the integration of several enabling technologies. In this chapter we will explain the main contributors of the future vehicular networks and IoT.

4.1 Virtual Machines

Ever since technology was invented, immediately became part of our life, taking over our daily lives and daily business operations. Companies rely on technology in such a way that its absence, even for a short time, causes a lot of financial damage. Companies make use of networking, computers and servers to run applications. Thirty years ago, each application required its own server to run on. When companies needed to deploy a new application, they could not use the existing servers, even if the servers had sufficient available capacity. So, they had to purchase and run it to a new server. Soon, this become a problem, as companies needed several applications and each application required a back up server in case of malfunction. This led to the proliferation of servers, which became costly and very inefficient. A server supported only one application, even when a single application did not fully occupy the capacity of the server. In fact, according to different studies, nonvirtualized servers only utilize up to 10% of their capacity, which led to an ever-rising number of underutilized servers. Actually, servers are dedicated to a specific Operating System (OS) and the only way they can manage multiple applications is when they are supported by the same version of OS. The solution to this problem is Virtual Machines (VMs).

VM, by definition, is a self-contained software which runs its own OS and applications, acting as a physical computer. In other words, VM is an exact copy of server, consisting of a certain amount of memory size and the hard drive capacity. Many VMs can run on the same physical server. Due to the fact that VMs are strictly isolated and use only a specific amount of server resources, they do not impact other VM hosted on the same server and act as individuals machines. This enabled servers to support different OS and therefore different applications on the same piece of hardware. Because VMs are softwares, it is really easy to deploy them and it does not take a lot of time to provision new applications. Moreover, you can easily spin VM up to different servers and suspend them at any time, replicate a VM to other servers as simple as coping and pasting folders, migrate server to data center in a matter of hours and no disruption in service. Whereas, without virtualization it would take weeks or months and most likely disruption in service. Because VMs act as strictly isolated machines with individual purposes, they offer a better recovery capability. If a VM and application fails, it is easy to move the other VM to another server in little to no time and continue working as normal, unaffected from other VMs. Virtualization is enabled by decoupling the server hardware resources and OS through hypervisor which is placed in-between them. In this way the hypervisor acts as a virtual gateway to the hardware resources of the server for the OS of VM. Hypervisor is placed on top of the hardware of the server, in which no server OS is loaded, and allows the servers resources to be shared between multiple VMs via the drivers. In order to run multiple VMs on the same physical server, the hypervisor manages the access requests and information flows from the VMs to the server via multiplexing, but it is important to do this in a transparent way. Which means that the active VM should not monopolize the entire system resources, therefore active VM do not affect the performance of other VMs operating in the same server, at the same time. Figure 4.1 shows a comparison between the architecture of a nonvirtualized server and the architecture of virtualized server. The implementation of VM and hypervisor increases the efficiency of servers on a very large scale (up to 80%), reduces the cost of hardware sprawl and the energy consumption costs for maintaining and cooling the servers, offers better flexibility, agility and disaster recovery capability. Moreover, the usage of VMs removes the dependency on proprietary hardware. This allows companies to avoid vendor lock in because the VMs will run on any server and can support any desired OS. In conventional servers, you could not just spin up an move an OS without a great deal of effort.

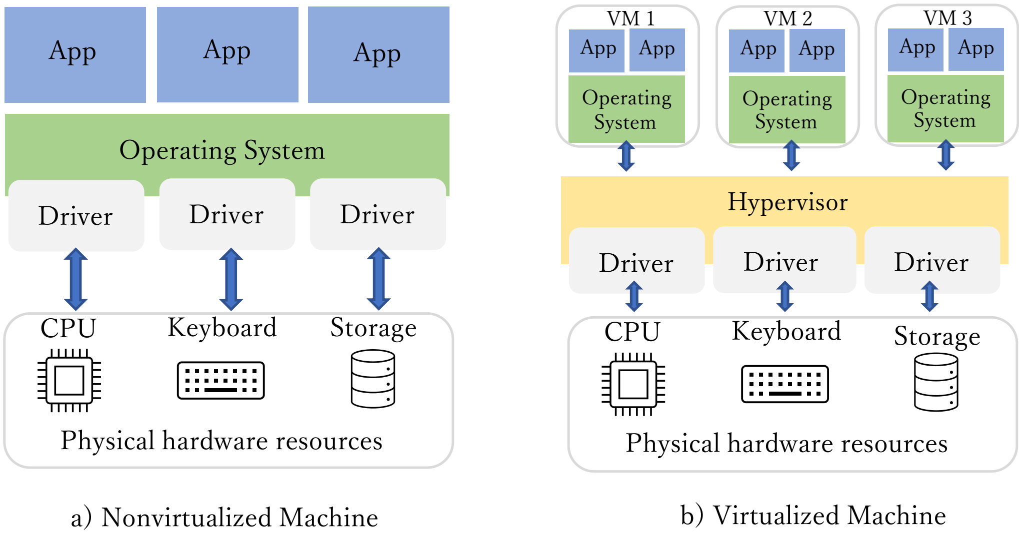


FIGURE 4.1: Comparison between the architecture of a nonvirtualized server and the architecture of virtualized server.

4.2 Cloud Computing

A cloud is on-demand access to shared computing resources – which include data, applications, or infrastructure as a service – hosted at a remote data center. Because virtualization and cloud computing are closely related, some people confuse the two concepts with each-other. However, they are two separate technologies. Virtualization is based on software, whereas cloud refers to all the services provided by a virtualized infrastructure. Virtualization is the foundation of cloud computing. It is the virtualization technology that led to cloud, which could not have existed without it.

Cloud computing provision configured computing resources such as: storage space, networks, servers, virtual machine instances, database, services, applications, and so on. According to The National Institute of Standards and Technology (NIST), cloud computing must be characterized by five essential attributes, described as below. These attributes are now widely accepted by companies and academia as an accurate representation of cloud technology.

- **On-demand Self-Service:** Cloud computing must be easy, convenient and available at all times. A client, even when he/she does not have a background on computers and information technology, must be able to provision computing resources from cloud at any time. On the other hand, even the interaction from the cloud service provider should be minimal. The client can access the resources through a web self-service portal, without the interaction of the service provider.

- **Ubiquitous Network Access:** Access to cloud must be provided by a standard platform, from whatever location and with any device. Simply put, the access must be available without the need of dedicated cables or special hardware devices. This is successfully realized through Internet, which has coverage all around the world and is accessible with a click of a button.

- **Pay Per Use:** The cloud does not belong to the customer, however he can rent it. The service is metered and billed accordingly on the amount of the usage. This eliminates the capital and management expenses of buying and maintaining servers for the client. Private clouds are an exception; they require a fixed monthly payment. Because their service is dedicated only to one customer, even if the reserved resources are idle they can not be shared with other customers.

- **Rapid Elasticity and Scalability:** Elasticity is the ability to respond to customer needs for cloud resources with ease. These needs might be fluctuating according to business demand. Cloud resources should be able to provision rapidly for a short time and on the same way deprovision when is no longer needed, with no penalty. An example is sales boost during holidays. This feature allows enterprises to take advantage of the benefits of cloud resources on an affordable price. They can perform right away intensive computational work during a project period, without having to spend for upfront costs. Then, when they are done, they can easily cancel it and do not have to pay anymore. On the same way, scalability shows the ability of cloud to scale up and down for enterprises that are planning to expand in the future.

- **Location-Independent Resource Pooling:** Cloud resources are not allocated only for a dedicated client, but they are shared among clients dynamically. When a customer is not using his resources for that moment, that resources can be lent to another client temporarily. Multiple clients share the same physical infrastructure, while retaining quality of service and privacy. In this way, by sharing the cloud resources between many clients, the provider is able to maximize the efficiency and utilization resources.

4.3 Network Function Virtualization

Network Function Virtualization (NFV) is the abstraction of Layer 4 through 7 services – such as load balancers, firewalls, intrusion detection systems and so on – from dedicated hardware devices to VMs. Traditionally these services would run only on dedicated hardware. The conversion of these proprietary network appliance into VMs allows for removing the dedicated devices and instead using software applications that run on VMs to perform the same functionalities, but now they can be deployed more easily and quickly. The virtualization allows multiple network appliance to run on a single server, which brings network consolidation with less cost and more efficiency.

NFV emerged due to inefficiencies that brought virtualization.The ease of spinning up and moving around VMs brought varied traffic in the network, which affected fixed network appliances as they had to serve the traffic. Converting fixed network appliances into virtual instances of network functions allows them for an acceleration of deployment, provision and management whenever is needed, just as a VM would do. Moreover, network configuration and management is easier with NFV. In traditional networks the network engineers had to buy dedicated hardware and configure them one by one manually. Which was a cumbersome process that required space, money and time. Whereas, NFV is controlled by the hypervisor which allows for a centralized configuration and management and can even automate the provisioning of the network functionalities. Moreover, network functionality can be changed or added easily on demand with the help of VMs. NFV allows providers to control cost as NFV runs on a standard generic server controlled of virtual machines instead of proprietary hardware servers. It takes less time as the configuration and management is orchestrated centrally. In addition, NFV offers a pay per use service, customers pay only for what they need.

NFV architecture is established by ETSI and consist of three parts:

- **Virtual Network Functions:** These are the functions of a network – which include Layer 4 through 7 services – that have been virtualized. Some of them are Firewalls and intrusion detection systems (IDS), Load balancers, Evolved Packet Core Data loss prevention systems, Secure Sockets Layer (SSL) accelerators, Security event and information managers (SEIM), Virtual private network (VPN) concentrators, and so on.

- **NFV Infrastructure:** The infrastructure that NFV runs on consist of the hardware resources (the compute, storage, networking resources of network switches and server) and the virtual resources (the abstraction of hardware with VM). The abstraction of these two layer is realized by a virtualization layer – the hypervisor – which is placed between the two. Network management is based on the hypervisor which controls the network in a centralized approach.

- **Framework for NFV Management & Orchestration:** A framework is needed to management and orchestration of the resources of the network and provision of the network functionalities. This framework is also known as MANO (management, automation and network orchestration). It handles the creation, delegation and allocation of resources in the NFV infrastructure.

4.4 SDN

In 1990 began the emergence of second generation of data centers with servers located inside a room, in proximity to the users. Network engineers made sure to design high-available data centers and increase their capability in order to guarantee that users had always access to the network. Traditional data centers worked fine for that time. However, despite the capability that network gained and the changes in operation, it never seemed to be enough. Customers raised their data consumption as they began to rely even more on these networks. The high demand created increased stress on data centers and the quality of experience offered did not satisfy customers any more. It is true that VMs offered efficiency but still they had problems when implemented in traditional data center. They had different properties, and just were created for different purposes. Traditional data centers could not handle the high load of traffic and agility required by VMs; VMs simply were not suitable to be implemented in traditional data centers. Gradually traditional data centers started to show problems. The problem was not with the network. The design was working just fine. It was the nature of resources that it was offered that changed and how people utilized these resources. It is not so long ago when a movie took hours to download, but now we complain when it takes even several minutes. Users want services and we want it right now. Moreover users want to customize these services accommodating their changing needs. In order to take full advantage of the network, changes needed to be done. The network needs to be programmable in order to adapt to dynamic changes. Software Define Networks (SDNs) and cloud computing are the result of network evolution.

Traditional networks operate in layers – referred to as OSI architecture – where each layer has a specific set of instructions how to send data across the network. The data is forwarded via the network devices such as routers and switches. When a session is initiated, the data packets are sent to the first network device, which takes the decision where to forward the data packet next. The network devices consist of two logical planes: the control plane and forwarding plane. The control plane is where all the applications and table of rules are stored. It is this plane that takes decision about the best route that the data packet should take. Whereas, the forwarding plane moves the packets from one network device to the other. The two layers are interconnected and closely tied just like the brain and the muscle where the brain is the control plane and the muscle is the forwarding plane. The network is comprised of thousands of network devices, each of them is an self-contained device, working independently and taking their own decision about the best route for data packets as the data travels from one device to another. To achieve this, each of these devices sends and receives data from the network to assure that packets are taking the best route. This increases the traffic and complexity of the network, given that the state of the network is constantly changing and each of the devices have to control the state of the network at every moment in order to take their own decisions. Moreover, when a rule set or function changes in the network, these changes have to be programmed individually into every single device. Taking into consideration that a typical long-haul network is made of hundreds and thousands of these devices, a change in the network causes tremendous amount of complexity. The best solution that will make the network more efficient and reduce complexity is separating control plane from the forwarding plane, just like separating the hardware from the OS on server virtualization. The separations creates them the ability to innovate and evolve at their own pace, independent from each-other, and paves the way for new properties and services for the network. It allows for a centralized control of the network, where the 'brain' is placed on top of the network and gives directions to all network devices, rather then all the devices trying to control the network. Having a broader visibility and knowledge of the network, provides better efficiency and intelligence over the network. Being able to program the network centrally, removes the burden from individually programming each of the nodes of the network, and therefore reduces the complexity.

SDN will be described in detail in the next chapter, however here is given a short introduction about SDN. How traditional data centers operated, the procedure how data traveled across the traditional network and the problem they faced with the increase of data demand. In the end, the concept behind SDN that brought the solution to this problem is also explained.

4.5 Network Slicing

Networks serve to many applications that assist different sectors such as healthcare, automotive, finance, logistics and others that are not fully leveraging the potential of the network yet. Each of these applications have their specific requirements. One requires high bandwidth for example, another requires high reliability and extremely low latency. But sometimes their strict requirements encounters problems. Because they rely on the same infrastructure and share the same resources, and there is no restriction in utilizing these resources, there can be conflict. Some of the applications can not satisfy their requirements, thus they can not provide the best quality of service. One solution might be creating a set of dedicated networks adapted to fit requirements of each application. The networks will operate according to the needs of each application, however this is very costly. A much more efficient solution is the creation of dedicated networks relying on a common platform. This is the notion of network slicing. Multiple virtual networks that provide services for different types of market sectors, run independently on a common physical infrastructure. Each subsytem slices the needed amount of resources in order to yield an independent network for its application. However, this approach is achieved by exploiting SDN and NFV technologies which enable network reconfiguration with software and management of network resources. One of the promising sectors to implement network slicing is vehicular networks as they need to satisfy stringent requirements of present use cases but also future ones.

Network slicing should follow the following principles:

- **Slice Isolation:** This feature enables the fair sharing of resources to multiple tenants and guarantees the performance of the services. It provides security because the tenants work independently and do not intervene with the other tenants work. Isolation impose restricions on the usage of network resource for each slice. Each slice is allocated the necessary network resources but the tenants are restricted the access and modification of resources with other tenants.

- **Elasticity:** In order to efficiently utilize the network resources, this feature allows the alteration of resources among different tenant slices. This can be achieved by scaling up and down to allocated resources of each slice and reprogramming the network functionalities. However, this should be carefully regulated in such a way that the inter-slice operation does not affect the performance of other slices.

- **End-to-end Customization:** This feature allows that the customer customize their dedicated network to their specific needs. Customization is expected to be offered in the future to vertical industries.

4.6 IoT

IoT refers to the network of everyday objects connected to the Internet to generate data that aim at enhancing the quality of life. These objects range from household tools and equipment to industrial machines and even humans. Everything traditional will be transformed into smart by integrating animate and inanimate objects into the information networks via embedded devices, actuators, sensor networks, communication technologies, etc. IoT will enable these devices to connect to the Internet, share information, communicate with each other and other networks free from human intervention.

Enterprises and consumers alike will profit from the large-scale deployment of IoT as this will enable a broad spectrum of services that will deliver efficient solutions

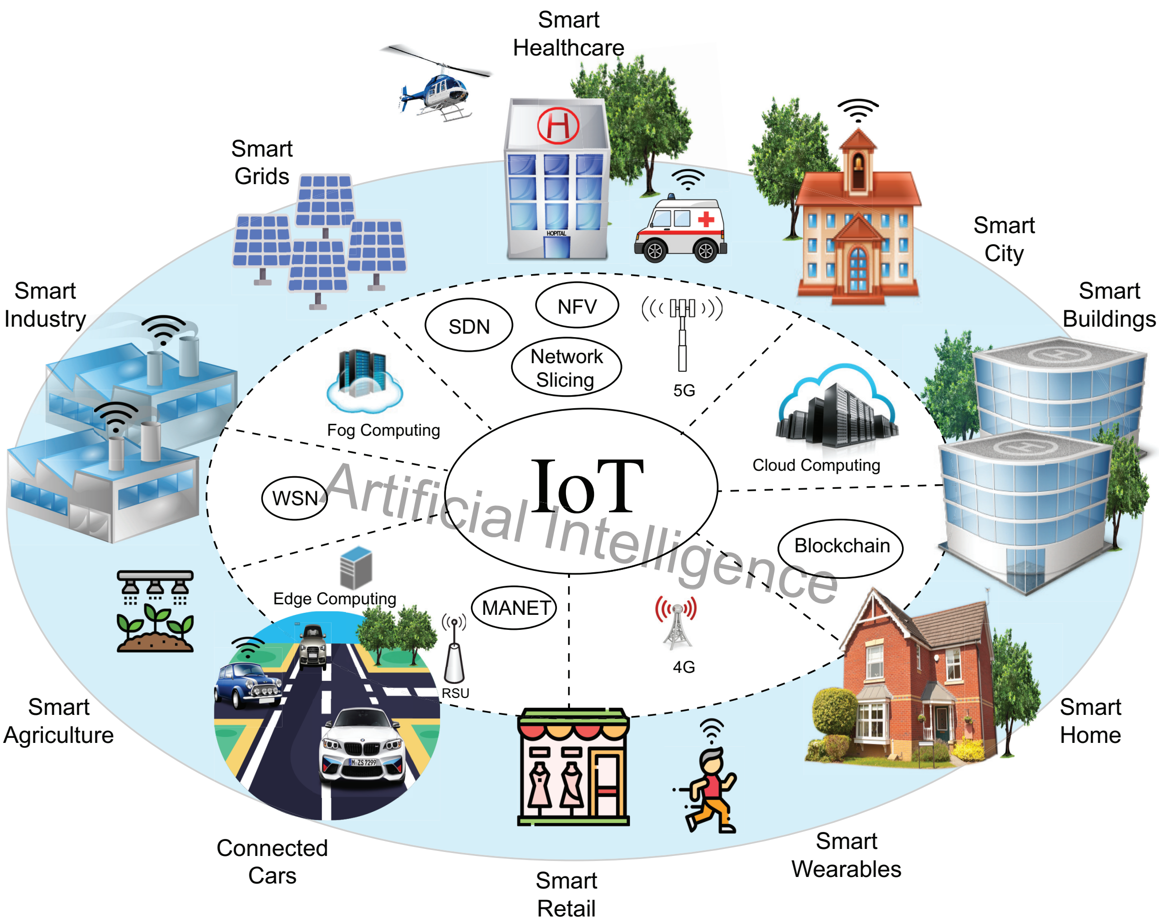


FIGURE 4.2: IoT applications and enabling technologies.

to many issues. Consumers will have a better quality of life due to the capability of IoT to improve health monitoring through wearable sensors, boost home security, time efficiency, and comfort, prevent unreasonable energy consumption, advance education, and provide many other benefits. Whereas enterprises will have more profits, will offer increased service quality, and will create new business opportunities by understanding the customers' needs, boosting productivity, ameliorating decision-making in the manufacturing process, preventing malfunctions, improving transportation and logistics, and so on.

However, these benefits will not come without a cost. In order to successfully deliver applications, many requirements must be first satisfied. For example, services related to safety and healthcare are time-critical, but they do not necessarily need high bandwidth. On the other side, surveillance systems do require high bandwidth to transmit videos, but most of the time do not need the data to be transmitted in real-time. The network must identify the service needs and must be flexible to give primacy to high-priority applications. Other important issues that IoT faces include interoperation with different types of systems, network data congestion, geo-fencing, and location accuracy.

Smooth interoperation of all the components participating in IoT, including Cellular Networks, MANETs, WSNs, and other enabling technologies such as Software Defined Networking, Cloud, Fog, and Edge Computing, is recognized as the feasible solution to these issues. We illustrate this conceptualization in Figure 4.2. Vehicular networks will be a crucial component of this concept as not only will vehicles use the information made available from the other integrated components but they will also share their resources and provide their information to the IoT, helping to better manage traffic, cut pollution, make better use of infrastructure and enable citizens to stay safe and clean.

Chapter 5

SDN and Cloud-Fog-Edge Computing

To meet the future ITSs needs it is necessary to integrate SDN and Cloud-Fog-Egde computing within vehicular networks. In this chapter, we explain in details these technologies, as our work is also based on them.

5.1 Cloud Computing and the Emergence of Edge Computing

Different services and applications are leveraging the internet resources for more storage and processing capability. Different applications have diverse requirements which often demand for customized solutions. Cloud Computing (CC) has acted as a de facto solution over the past years. CC is based on a group of data centers connected together, offering more computing and storage capability. The set of data centers work together as a unit, with a low latency communication among them, in such a way that they are seen as a singular entity, called data center network. However, the rapid growth of emerging applications like IoT, Augmented Reality (AR), e-health, V2X among others, have shown the limitations of CC. These new applications require new properties for which conventional centralized CC has not been specifically designed to accomplish. First, they require real-time processing of the data collected by sensor nodes, which are used for taking decisions and further actions without delays. In CC, even though the communication among data center devices is provided in ultra low latency, the communication with the end user is proved to be a bottleneck. Secondly, the emerging applications of vehicular networks with V2X communications require a mobility support due to their dynamic nature, which CC can not offer because of its long physical distance to the system and its centralized architecture. Moreover, the number of connected devices to CC is estimated to increase exponentially in ten years, which will impose additional load and excessive traffic in CC, resulting in increased delays and degradation of quality of service. These are some of the challenges that Edge Computing (EC) paradigm promises to overcome. EC leverages the computing resources of cloud and brings them in proximity to end users. EC can serve as an intermediate layer between the cloud and end devices and can reduce the computational load of cloud by offloading the data to the edge layer. Consecutively, this reduces the time to handle requests and resolves them with no latency. EC are a good solution for dynamic systems that suffer from intermittent connectivity, due to their geo-distributed nature and abundant availability [64]–[68].

There are three main implements of edge layer, based on system architecture, the network used and the services supported; which are the Multi-access Edge Computing (MEC), formerly known as Mobile Edge Computing, Cloudlet Computing and Fog Computing. Let's give a detailed analysis of each of them in the following sections.

5.1.1 Multi-access Edge Computing

We are going to present MEC as defined by ETSI. MEC extends cloud computing capabilities by bringing them within the Radio Area Network (RAN). MEC nodes or servers are co-located at several location on the network, such as at the Base Station (BS), at the Radio Network Controller (RNC), or at a multi-radio access technology site. This deployment option allows for a direct delivery of services accessible for a single hop from the end user [69]–[71].

MEC consists of the mobile edge host, mobile edge management, and external network related entities. The mobile edge host is comprised of mobile edge platform and the virtualization infrastructure which provision the storage and computations resources for implementation of MEC applications as software-only entities via virtual interfaces. MEC applications are running via multiple VMs on top of the virtualization infrastructure and interact with the mobile edge platform to perform procedures related to the life-cycle of the application. The virtualisation infrastructure includes a data plane which executes traffic rules from the mobile edge platform and then redirects the traffic among applications, external and internal networks. The mobile edge management supervises the mobile edge host, the resources available and the network topology and validate the MEC applications that server can support, given the application requirements and the available resources. To do so, the mobile edge management keeps a catalog of ongoing applications on the ME host and if the request for MEC application is already running, the request is forwarded to the application. If not, the application is instantiated if the system supports it and the request is accepted.

MEC reduces congestion of the core network, decreases latency, offers high bandwidth and proximity to the end user. Since MEC nodes are placed near RNC, they have a wide view of the network. They receive real time information about the network load and capacity, information about device location and improve context-awareness information over the network. This attractive features make MEC suitable for delay sensitive applications, therefore MEC is a good option for VANET applications. MEC offers a promising solution for AR application, as its requirements such as low latency and high data processing, can be supported by MEC. Video analytics for monitoring devices can be locally processed in MEC, which offers real-time data analysis to detect specific events.

5.1.2 Fog Computing

Fog computing is coined and standardized by OpenFog Consortium, an association of Cisco Systems, Microsoft, Intel, Dell and Princeton University. Fog computing is defined as a continuum of traditional cloud computing. Even though they are considered under edge computing paradigm, there are some key differences. Fog is located not only in the edge layer, but in multiple layers of a topology of a network, at any point of the architecture between cloud and end user devices and at any device that offers computing, storage, and network capability [72]–[74]. Fog works with cloud computing and complement each other in a mutual-beneficial service continuum, while edge is seen as a stand alone technology that excludes collaboration with cloud. Fog is heterogeneous and leverages the storage, computing and networking capability of legacy devices throughout the network such as routers, switches, gateways, access points, among others. As a consequence, it offers a fully interoperable system by supporting devices used at different protocol layer and with different access technologies IP based and non-IP based such as Zigbee, BLE, Bluetooth, Wi-Fi. It is the Fog abstraction layer which enables open communication of such heterogeneous devices and also multi-vendor interoperability. End users devices requires services from Fog nodes with a particular set of requirement. The requests are received by Service Orchestration Layer which supervises the Fog abstraction layer for performing functions like device management, balancing, controlling, monitoring, security and resource allocation across fog network in order to accomplish user requirements.

Fog Computing is driven by a set of core principles such as security, scalability, autonomy, reliability, availability, Serviceability, agility hierarchy, and programmability. Fog networks should offer additional scaling opportunities when is needed by increasing the individual fog node capability with additional hardware (adding more storage devices, processors, or network interfaces) or software, or by utilizing other fogs on the same level of hierarchy or in adjacent levels. Fog networks should support programming at the software and hardware layers for accommodating operational dynamics. Fog networks should ensure correct operation under normal and adverse operating conditions. Hierarchy fog architecture may not be applied in all scenarios but is offers efficiency for partitioning large scale networks with resource-intensive requirements. The network is arranged in physical or logical layer, each offering different functions that serve the upper layer.

Based on its key features, Fog computing is suitable for a diverse range of applications. As it can support a large scale of heterogeneous devices generating large volume of data, makes fog computing suitable of data mining and big data analysis for different IoT applications. Emergency, health care, public security, visual security and surveillance and other applications that are time sensitive or security sensitive can be supported by fog networks. Fog network has proven to be feasible also in VANET applications like autonomous driving, infotainment, traffic control system, collision avoidance, advanced driver assistance systems, and so on. They have been implemented either on a hierarchical architecture or having the vehicle act as a fog node.

5.1.3 Cloudlet

Cloudlet is a new paradigm in the edge computing proposed by Prof. Mahadev Satyanarayanan at Carnegie Mellon University to resolve the resource poverty constrain of mobile devices. Mobile devices are lightweight and small in size, but they lack the intensive computational capability to augment human-like cognition ability in applications like face recognition, computer vision, language translation and so on. Cloudlet promises to give solution. Cloudlet is a mobility-enhanced small-scale cloud data center (DC) that is located at the edge of the internet. They utilize dedicated standalone devices distributed geographically which are accessible through a one hop wireless local area network and provide crisp interactive response and high bandwidth. The dedicated devices have similar capabilities with data center but on a lower scale, and as such are considered as 'data center in a box'. Rapid customization of cloudlet infrastructure for supporting diverse application requirements is achieved by dynamic VM synthesis.

In order to reduce the occupation of user device storage, the mobile user carries an overlay VM which magnitude size is smaller than the complete VM. A typical VM application overlay size for a Linux application is 200 MB, instead of complete VM size of over 8 GB VM overlay image refers to the difference between a complete VM and a base VM image. The complete VM is a base VM image in which the application overlay is installed. The mobile user sends a request to the cloudlet via VM overlay (application overlay). The cloudlet receives the overlay, decrypts and decompresses it, and applies it the base VM, which it already possessed by cloudlet server, and initiates a VM instance. This launches the desired application via launch VM and brings it to a state that is ready for user interaction. In this moment the user device can perform offloading. In a language translation application, for instance, the cloudlet server receives a recorded speech from a mobile device, processes the speech recognition, realizes the translation and turn back the translated text for speech synthesis. After the session is finished and the data is proccessed, the instance is destroyed, but a copy of launched VM image is kept in cache in case it is needed for future sessions. Furthermore, to retain some training data, the cloudlets generated VM residue that are sent to user device and are incorporated to its overlay. It has been proved that using such transient customization of cloudlet infrastructure - in which cloudlet is restored to its original condition after each session, without manual intervention - has reduced the response time by 51% and energy consumption by 42% in a mobile device compared to cloud offload. This is a significant improvement, especially for resource intensive and interactive devices who have battery limitations and do not except extended delays such as wearable devices, wireless mobile nodes, smartphone, tablets, and so on.

Table 1 depicts the differences between the three implementations of edge technology.

5.2 SDN

The preceding chapter talked about the reasons why a decoupling of control plane and forward plane is necessary in networking. In doing so, we can provide centralized control and enjoy the benefits of a overarching programmable network. This pulls out the network intelligence from the individual nodes of the network and places it in the hands of a central authority, named SDN controller [75]–[79]. This architecture greatly simplified the discovery, connectivity and control, and extensibility of the network. A comparison between a traditional network and SDN network architecture is shown in Figure 5.1. In traditional network all the network nodes (routers/switches) are equipped with the control plane, forwarding plane and have applications loaded on to it. The data packets are directed from each network node individually, based on their local logic, and application changes must be programmed systematically into each device individually. Whereas, in SDN the applications and intelligence do not reside into network nodes but in the SDN controller, which makes the network programmable. The network acts like one big router controlling the network. Applications interface with the controller, and their functions are applied across the network. The decision for best route is taken from the SDN controller, which is distributed to each flow table at routers. This architecture allows changes to get updated in the network or get rid of features they no longer use much faster and simpler. We can divide SDN networks in three logical layer: the application layer, the control layer and infrastructure layer. Let's describe them in details.

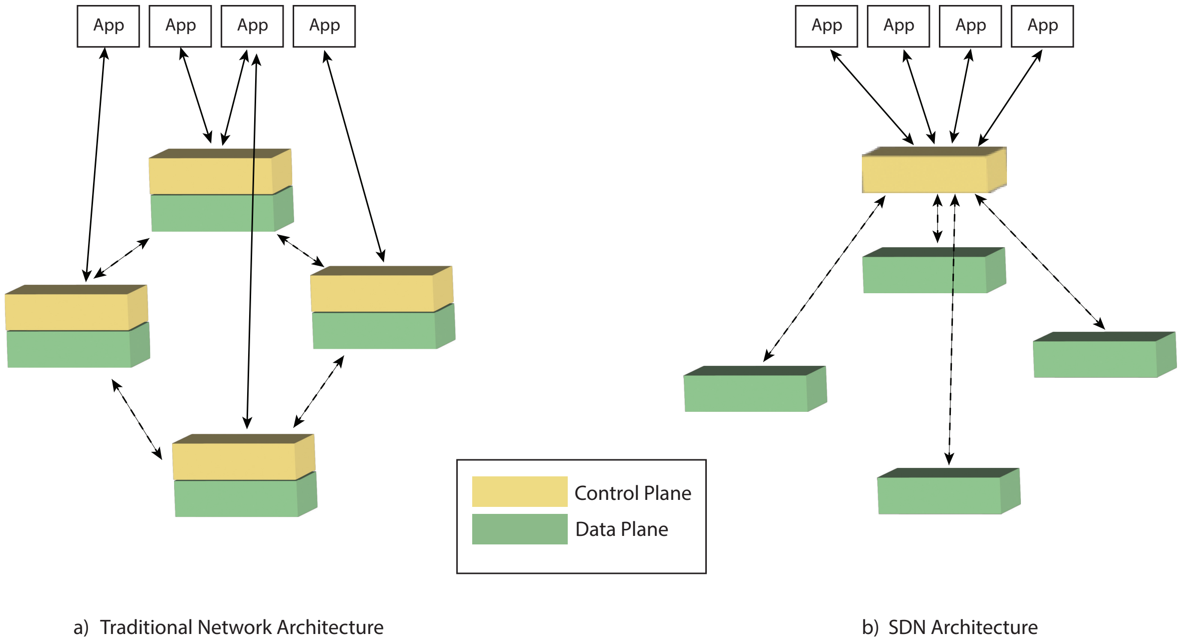


FIGURE 5.1: A comparison between a traditional network and SDN network architecture.

Application Layer

The application layer insist on business application or network services application. Network services applications include monitoring, load balancing, Voice over IP (VoIP), intrusion detection system, security applications, and many others. This layer explicitly communicates to SDN controller their requirements and the needed network resources. The communication is realized via Northbound Application Programming Interface (API). SDN contains applications that use the SDN controller to manage the forwarding plane plane behaviour, instead of using a specialized appliance (a firewall for example) like a traditional network does.

Control Layer

In SDN, the control layer functions are offloaded from the routers and switches and are handed to SDN controller which is able to manage and control the network centrally. This makes the network control directly programmable since it is separated from the forwarding plane functions. Control layer processes the requirements of the application layer and proceeds them to the networking devices in the infrastructure layer via OpenFlow protocol. More specifically, the control plane collects and analyzes information about the actual state of the forwarding plane (traffic data), and compares it with the actual state (the state that the network should be based on the application requirements). If they do not match, the SDN controller calculates the necessary changes of traffic flows and proceeds them to the forwarding plane. The changes are reflected to switch's flow table using OpenFlow protocol over a Secure Sockets Layer (SSL) channel. The controller can even determine multiple routes and load balance across them if appropriate. This comes in handy in the event of a failure, the controller instantly directs the traffic to alternative routes, making the network redundant.

Moreover, it is able to create multiple separate configurations running concurrently on a common infrastructure by creating a virtual network overlay on top of the physical layer. This means that SDN can be configured to create certain type of rules (forwarding rules in the flow table) and applications for some users, and different rules and applications for other users. By doing so, network administrator can apply different applications for different users, rather than treating all users the same. This benefits the network operators as they can provide customized and isolated services to their clients, based on their needs.

Infrastructure Layer

The infrastructure layer consists of the data and the physical network devices, switches, and routers. The latter moves the data packet across the network based on the flow table rules provided by SDN controller instructions. The infrastructure layer collect information about the network topology and network usage, in order to inform the controller about the state of the network. In addition, it also collects statistics, which report issues immediately to the controller, which, in turn, can make immediate adjustments across the entire network.

Chapter 6

Intelligent Algorithms

Nature has evolved over the years and has been a rich source of inspiration for researcher who often need to solve real world problems which are highly scalable and have proven to be nondeterministic polynomial (NP)-hard problems. Such challenging problems include various fields such as engineering, optimization, physics, medical and so on. Intelligent Algorithms (IA) have developed as a result of this nature based inspiration that have laid the foundation of today's Artificial Intelligence (AI) system. Scientists have observed and analyzed the way the elements of nature evolve and interconnect with each other, and designed different artificial systems that work on the same principles. Intelligent algorithms offer practical and simplified solution techniques on tackling challenging problems in a reasonably practical time, giving approximate solutions but satisfactory ones. This offers good-quality solutions when the complexity of the problem makes it impossible to search every possible combination. Instead it is more efficient to find nearly optimal feasible solutions that are easily reachable in a reasonable amount of time. A typical example is fuzzy logic (FL) techniques which create non linear controllers via the use of heuristic information when information on the physical system is limited. Our proposed system is implemented in FL; for this reason we are going to introduce it in detail in the next chapter. Moreover, scientists did not focus only on nature-based algorithms, but have broadened the family of IAs even with solutions that have never appeared in nature. Hill Climbing (HC), Tabu Search (TS), and Simulated Annealing (SA), for instance, are local search methods that have no inspiration from nature, and yet are ranked among the most prominent IAs.

An algorithm is a set of precise instructions of the steps a computer should follow to solve a particular problem. Algorithms start from an initial guess and aim to generate a better solution than the current one, at each iteration. We can divine algorithms in two categories: the deterministic algorithms and stochastic algorithms. Deterministic algorithms are very rigorous, they follow the same functions and the same path. So for the same starting point, the algorithm will give the same outcome. Whereas stochastic algorithm is not repeatable. Because it includes uncertainties, every time you run the algorithm, its result will differ in the end. A good example is generic algorithms. The solutions of population will be always different since it includes some pseudo-random numbers. There is also a hybrid algorithm which mixes deterministic algorithms and stochastic algorithms together. It follows the same path as the deterministic algorithm, but the initial point is chosen different every time it runs, which means some uncertainty is included. This method avoids getting stuck in the same local peak, and therefore improves the algorithm.

With so many diverse IA, a simple questions arises naturally 'which is the best algorithm?' Students should know there in no free lunch. The answer is that we can not determine the best algorithm ever, because they are created to give good results for a certain type of applications. As Wolpert and Macready has proven [80], if an algorithm A outperforms algorithm B for a certain objective function, then algorithm B will outperform A for some other objective function. There is no universal algorithm for all kinds of problems. However, for a given problem we need to choose what algorithm we are going to use. To do so, it is not an easy task. There is usually not only one algorithm but a set of algorithms that can be considered efficient for a certain problem. The choice usually depends on the expertise of person responsible for making the choice and the available resources such as software availability, computational costs, and so on. To determine the efficiencies of different algorithms it is needed to test them and compare the results. Another important component for determining the performance of an algorithm is also its complexity. The complexity is calculated by the number of iteration the algorithms needs to run for finding the optimal solution. Ideally, an algorithm should take only one iteration.

Stochastic algorithms include two types: heuristic and metaheuristic algorithms. Metaheuristic methods offer better performance than simple heuristic algorithms and they use certain tradeoffs of randomization and local search. Loosely speaking, meta means 'higher level'. However, due to the fact that there is little difference between them, and in the literature does not exist an agreed definition for both terms, they are used interchangeably. Moreover, all algorithms used today include randomization and local search, just like metaheuristic algorithms do. Two components of metaheuristic are intensification and diversification. Diversification means to explore diverse promising regions of the search space, so it can include all the regions of solutions at the range of global scale. This avoids getting trapped in a local optima. Intensification means to focus the search in one of these promising regions by exploiting the information that a current good solution is found in this region. The combination of these two components enables achieving a globally optimal solution. We have classified metaheuristic algorithms into four groups: evolutionary algorithms, swarm intelligence algorithms, human-based algorithms and physics-based algorithms. Let's analyze their key components.

6.1 Evolutionary Algorithms

Evolutionary Algorithms (EA) mimic biological evolutions, more specifically the abstraction of Darwinian evolution of biological systems, pioneered by John Holland in 1960s. Similar to the living things, which evolve over generations through the natural selection process in order to adapt to the changing environments, EAs select the fittest individuals of the population in order to have a reproduction process yield stronger offspring for the next generation. EA basic evolution cycle follow these steps: initialization, selection, genetic operators, and termination; as in Figure 6.1. The principle is that the strongest members will survive and proliferate, whether the weakest members will die off, similar to the natural evolution [81], [82].

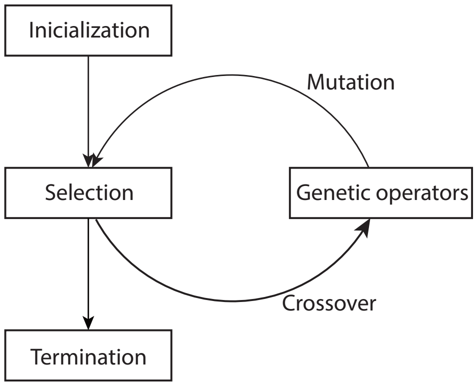


FIGURE 6.1: Evolutionary Algorithm cycle.

EA were first used for optimization problems, and later found use in many other fields. Different types of EA incorporate the same basic cycle with different model of presentations, the nature of the applied problem, or combinations of EA operator. The most typical EA algorithm is the Genetic Algorithm (GA). They also include genetic programming, evolutionary programming, evolution strategy, neuroevolution, learning classifier system, differential evolution, among others.

6.1.1 Genetic Algorithms

Before diving into GA, let us first explain some terms used. An individual is represented by a string of variables known as **chromosomes** (solutions). The smallest part comprising the chromosome is called **gene**, which are usually represent by binary values (1s and 0s). The set of individuals (chromosomes) form a **population**. Each individual is an candidate solution to the problem you want to solve. A **fitness function** evaluates how fit (good) an individual is, with respect to the problem we consider. In other words, determines how close a given solution is to the optimum solution [83].

In Figure 6.2 is shown the flowchart of a classic GA. The algorithm begins with a population of individuals that will undergo in each iteration a series of processes that allow for finding of better solutions. These processes include operations such as selection, crossover, and mutation and are implemented by means of various methods performed on the individuals that produce higher fitness values. This technique is inspired by the evolution of species, where the fittest species have higher chances of reproduction.

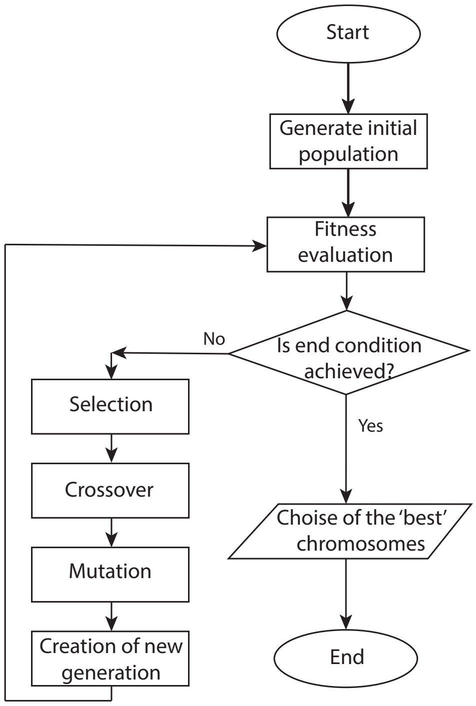


FIGURE 6.2: Flowchart of a classic GA.

The goal of GA is to select the fittest individuals for reproduction in order to inherit the best features and therefore yield the stronger offspring for the next generation. To do so, the algorithm calculates fitness functions for each individual and applies genetic operators such as selection, crossover, and mutation.

- **Selection:** The selection of the fittest individuals (parents) is done by choosing the individuals with the higher fitness score. In this way the the highest quality chromosomes with the best genes will be inherit for the next generations.

- **Crossover:** The pairing of two parents chromosomes by exchanging a random part of the genes of one parent with the other parent. The generated offspring are added the population. Crossover has a high probability (0.6 to 0.95%) to occur, which show for a high degree of mixing and exploitation.

- **Mutation:** The change of genes (a single bit or several bits) inside one chromosome, so as to generate diverse genetic characteristics within the population and avoid premature convergence. In bit strings this is done dy flipping 0 and 1 genes with each-other. Although it occurs in low random probability (around

0.001 to 0.05%), it can happen to a single individual or multiple individuals simultaneously.

6.1.2 Other Genetic Algorithms

Genetic Programming (GP) generates computer program that can program itself using Darwinian evolution theory. Contrary to genetic algorithm that has a particular objective to reach, in GP a general objective is decided, and then the system decides its own goal. This is done by continuous improvement of programs based on selection of the best fitness values evaluated from different evolutionary algorithms, crossover of different prat of the program and random mutation of the program. The most successful programs are bred to produce better generations.

Differential Evolution The algorithm starts with an initial population of candidate solutions. These candidate solutions are iteratively improved by introducing mutations into the population, and retaining the fittest candidate solutions that yield a lower objective function value. This algorithms offer a more effective and easier solution, as it can handle nonlinear and non-differentiable multi-dimensional objective functions, while requiring very few control parameters to steer the minimisation, that allow also people with low programming skills to use the algorithm.

6.2 Swarm Intelligence Algorithms

Swarm Intelligence algorithms (SI) have become very popular, especially in optimization, design applications, computer science, computational intelligence, and so on. They are inspired by biological systems such as fish and birds flocks and even by human behavior. A typical SI system is composed of many homogeneous individuals (agents) that have the same role in their community, and are able to act in a coordinated way, without the need of a coordinator. There is no centralized control dictating the group of agents behavior, however they are able to self-organize by interacting with each-other and the environment. As a result, global intelligence is reached. In the same way, SI mimics how the nature species interact with each other in order to find a solution. Due to their unique properties, SI systems offer fault tolerance, scalability, and parallel action. Fault tolerance is an inherent property due to its self-organizing and decentralized control feature. A failing agent does not affect the system. Since all the agents of the system have the same role, the failed agent can be easily substitute by another fully functioning agent. Because agents behavior is influenced only by their neighbors, the number of interactions does not increase with the overall number of members of the system. Conversely, artificial systems that deal with human behavior tend to offer better performance when the number of participants of the system increases. Whereas, parallel action is always possible because agent behavior can not be controlled externally; they can perform different actions in different places simultaneously. SI includes many types, such as particle swarm optimization, ant and bee algorithms, firefly algorithms, human swarming, and cuckoo search among others. We are going to explain in detail these algorithms, especially the particle swarm optimization which we have used in our research work in laboratory.

6.2.1 Particle Swarm Optimization

Sociobiologists believed that species moving in a group like a flock of birds can benefit from the experience of all other members. A flying bird looking randomly for food can share its experience with the other members so that they can get the best hunt. Each bird contributes to find the optimal solution in a high-dimensional solution space and the best solution found by the flock is the best solution in the space. This is a heuristic solution because it can not be proven that the global optimal is reached, however we can find that the heuristic solution in close to the global solution. The same logic is applied to Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart in 1995. The members in PSO are called particles. The particles move in steps throughout an search area, starting from an initial guess point. Each particle memorizes the best position they have found in the initialization phase. At each step (iteration) the algorithm evaluates the objective function of each particle. After the evaluation a new direction is decided for each member. The movement of particles is decided based on its own best location, the current best location reached by the swarm (global best), and a random value (stochastic component). If the particle reaches a better location from the current best locations of all the particles, it updates this location as the current best location for particle i, but also as a global best location for all the particles. The aim is to find the best location of all the current best solutions. In this way, the algorithm evaluates the possible solution at the following iterations and adjusts the particle trajectories, until the objective no longer improves or after a certain number of iterations is reached. This movement of particles is schematically shown in Figure 6.3. PSO offers many advantages such as it is easy to implement and it has only a few parameters to be set. It has shown good results for finding mid optimum, however we can not say the same for its performance in local search.

6.2.2 Other Swarm Intelligence Algorithms

SI algorithms share a few similarities. Among other characteristics mentioned in the introduction section of Chapter 6, they are population-based methods that optimize

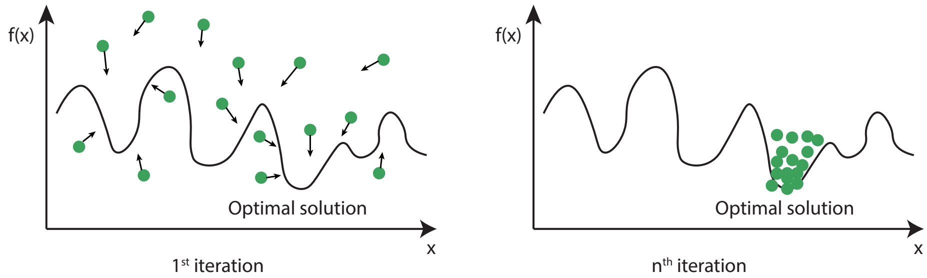


FIGURE 6.3: Arbitrary movement of particles placed at random positions in the first iteration and expected convergence of the algorithm after many iterations.

the given problem by improving the candidate solutions (simulation agents) over many iterations according to a given measure quality.

Ant Colony Optimization (ACO) imitates the real ants behavior, which explore the environment and trail pheromones to direct other ants to the discovered food resources. The artificial ants (simulation agents) explore the search space and record their coordinates and the quality of solutions they have found to enable more ants to locate better solutions in later simulation iterations. In the end the best route results to be the one that has higher pheromone concentrations. The ACO algorithm transforms the problem into a weighted graph whose possible paths would mark the set of solutions to the problem. The artificial ants can explore the search space and find their solutions by moving on the graph. The solutions are built incrementally through the modification of a set of parameters that are associated with one of the graph components (nodes or edges). The best path found at the end of all iterations is the solution that the algorithm proposes to the problem. ACO has resulted to be an effective technique for assigning aircraft arrivals to airport gates. Each pilot learns from its experience what is the best gate to land, and his experience contributes for the best solution for the airline too [84].

Artificial Bee Colony (ABC) was introduced by Karaboga in 2005. Contrary to ACO, they do not use pheromone. The model consists of three goups the employed bees, the onlooker bees which watch over employed bees and scout which look for food randomly. First, the food source is found by scout bees. The nectar in food source is exploited by employed bees and onlooker bees, which become exhausted after working for some time. For this reason, they become a scout bee in search of further food sources once again. The location of food sources represent a candidate solution to the problem, whereas the nectar of the food represents the quality (the feedback-fitness function) of the associated solution. The algorithm encourages to abandonment of food source that give negative feedback, and the recruitment of scout bees that find source food that have positive effect [85], [86].

Firefly Algorithm is inspired by the fireflies communication through flashing light among each other for mating or warn for predators. The rhythmic flash, the rate of flashing, and the amount of time between flashes form the patter of signal that brings both mates together. This phenomena has inspired Xin-She Yang to mimic fireflies behavior into a new optimization algorithm. The brightness of the fireflies is associated with the objective function. The brighter the light of the firefly, more attention attracts from mates. Thus, fireflies will move toward the brighter one [87].

Human Swarming is an artificial swarm intelligence that shows that the performance cognitive intelligence of humans increases when they are working as a team, connected in a real-time network modeled after natural based swarms, in contrast to individuals working alone. This system is distributed to different groups of people around the world and enables them to ask and answer questions, make predictions and take decisions by working together. Human Swarming has been used also in medical domain, to enables doctor around the world to diagnose illness on patients [88].

6.3 Human-based Algorithms

Human-based algorithms are meta-heuristic techniques inspired by human behavior. The agents in these algorithms are humans and algorithm mimics how they take decisions based on their knowledge, judgment, experience, the context at that moment, and their ability to analyze. The statistics of such choices gives us material to study the processes of human creativity, and their ability to innovate and adapt. Certainly humans have high level intelligence and communicate faster than animals. If a system is able to imitate human reasoning and behavior, this means that the system can be thought to be as intelligent as humans. Some of the human-based algorithms are League Championship Algorithm, Tabu Search, Exchange Market Algorithm, Imperialist Competitive Algorithm, and so on.

6.3.1 Tabu Search

Tabu Search (TS) is a local search method, which in other words explore a certain search space by moving from solution to solution in local changes, until a sufficiently good solution is found or a time-bound is elapsed. TS has integrated memory structures, which prohibit to visit a previously explored search space for finding a solution. The memory consists of short term memory and long term memory [89]. Short term memory is based on recency of occurrence which prohibit the exploration of already explored regions by keeping track of the regions in a tabu list. Whereas, long term memory is based frequency of occurrence and promotes the diversification of the search.

In contrast to other IA which accepts only better solutions, TS can replace the current solution with a less efficient solution if no improvements are available in that area. This move can be considered flexible because it allows the algorithm to escape from getting stuck at a local optimum. The algorithm then selects a new search movement by forbidding itself from going back to recently-visited solutions in order to allow other new regions of the search space to be visited. To store all the prohibited solutions, the algorithm uses a "tabu" list which is checked every time a new solution is evaluated. The algorithm continues to explore the search space until a stopping criterion forces the algorithm to end, e.g., an attempt limit has been reached, or a score threshold has been satisfied. The steps of the algorithm are shown in Figure 6.4.

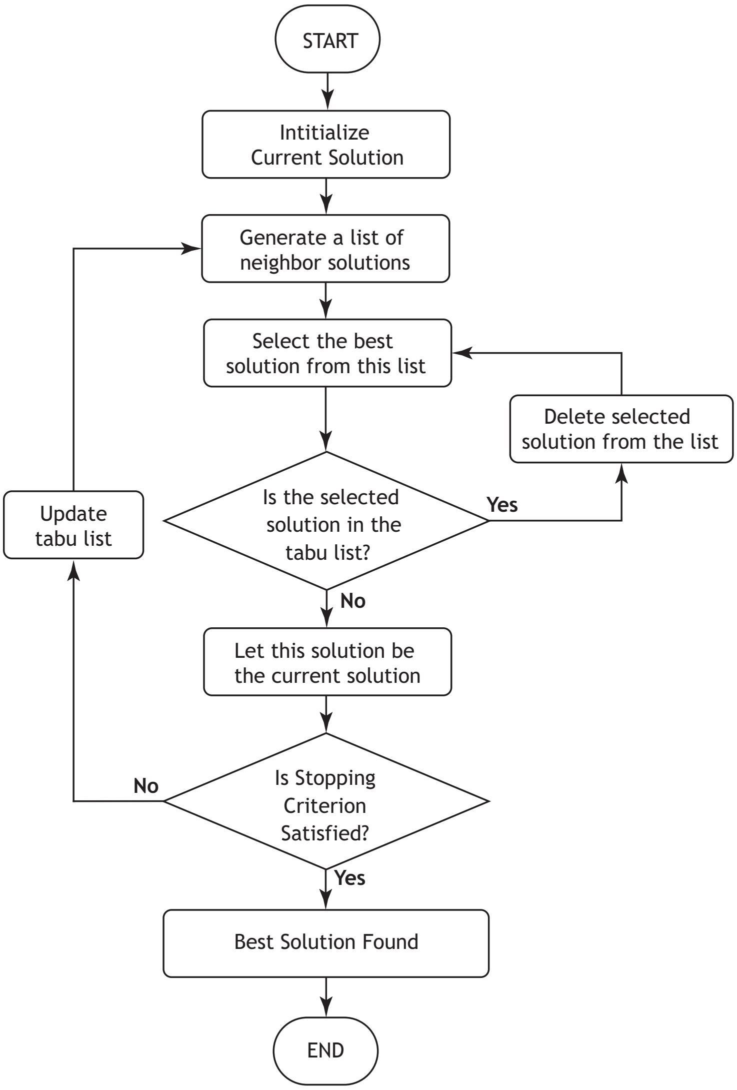


FIGURE 6.4: Flowchart of a simple TS algorithm.

6.3.2 Other Human-based Algorithms

*League Championship Algorithm*is a stochastic population based algorithm which mimics championship environment in sports. Artificial football clubs participate in the championship league for several weeks, which represent the iterations of the algorithm. They play in teams, just like normal matches and the result of the game is determined as a win/a loss or a tie, based on the playing performance (which represents the fitness value) and the team formation (represents the solution).

Exchange Market Algorithm is inspired by the behavior of shareholders in stock market. There are two states: the normal and oscillation state. The normal state does not face oscillations and in this state the algorithm searches for local optimum. The oscillation state is the most dynamic state, in which members try to take more risk in order to gain more benefits. In this state the algorithms looks for global unknown optimum points. The population of stakeholders consists of three categories based on their fitness values. The first group makes up 10-30% of the population and is the highest rank with the most amount of shares. This group does not trade any shares in neither of the states. The second group is the middle ranked dealers that consist of 20-50% of the population. They keep their total amount of shares constant while changing the number of some of each type of shares. The third group is the lowest and as such they tend to take higher risks in order to increase their value. They trade shares all the time and unlike the second group, they do not keep the total amount of shares constant.

Imperialist Competitive Algorithm is inspired by wars in the past times. The population individuals called country consist of imperialists and colonies, which because of imperialistic competition aim to take control of the country. The weak empires collapse and the powerful ones take possession of their colonies. In the end, the country is ruled by only one empire and its colonies are in the same position and have the same cost as the imperialist.

6.4 Physics-based Algorithms

These are metaheuristic algorithms inspired by physical phenomenon like gravity, big bang theory, inertia, gravitational kinematic, radiation, electromagnetism laws to solve complex problems. Even though biology based algorithms are more famous, physics based algorithms are not less effective. They have been applied to engineering, industry, business and many other areas. Some algorithms under this category are Electromagnetism-Like Algorithm, Simulated Annealing, Gravitational Search Algorithm, Space Gravitational Algorithm, Intelligent Water Drops Algorithm, Harmony Search, Big Bang-Big Crunch Algorithm, Central Force Optimization Algorithm, among others.

6.4.1 Simulated Annealing

Simulated Annealing (SA) is an algorithm that attempts to locate the best solution possible, usually a near-optimal solution, by making arbitrary adjustments to the current solution. These adjustments correspond to the other solutions in the current solution's neighborhood. At each iteration, the algorithm evaluates these candidate solutions coming from the arbitrary adjustments and assigns an acceptance probability to each of them. Then, based on the acceptance probability of the candidate solution, the algorithm decides whether to accept it or to stay in the current solution. Even if the adjustments result in a less efficient solution, the algorithm still evaluates it as a candidate solution while assigning an acceptance probability that decreases with these adjustments [90]. As it can be seen from Figure 6.5, it is the acceptance of these less efficient solutions, that similarly to TS, make SA escape the local optima to seek better solutions.

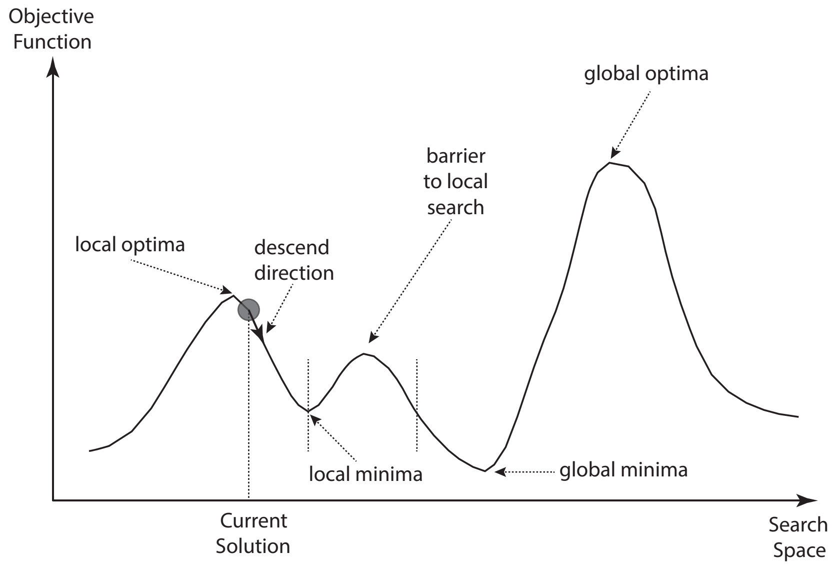


FIGURE 6.5: Search space diagram of a problem with multiple peaks and algorithm's capability to escape barriers and explore all solutions.

6.4.2 Other Physics-based Algorithms

Gravitational Search Algorithm are based on the Newton's Law. Which says that every object in the universe attract each other with a force that is proportional to the product of their masses and inversely proportional to the square of the distance between objects. In the algorithm every agent is considered as object. The fitness function of these objects are evaluated by their masses and the location of the heaviest object represents the global optima.

Big Bang-Big Crunch Algorithm is based on the evolution of universe and the theory of planets creation. The algorithm consists of two phases: the big bang phase which creates an initial population in a random manner, and the big crunch phase in which the population shrinks into a singularity. The objective function is to find the center of mass.

Electromagnetism-Like Algorithm Electromagnetism-like algorithm (EMA) is based on the electromagnetism law [91]. All the points are considered as charged particles which attract or repulse each-other. These forces move the point toward the optima. The objective function is to find high forces of attraction between points.

Chapter 7

Fuzzy Logic

"As complexity rises, precise statements lose meaning and meaningful statements lose precision. Albert Einstein."

7.1 Introduction to Fuzzy Logic

Many attempts have been made, especially in this century, for augmenting the representational capabilities of logic, or for proposing non-additive models of uncertainty. One of the most radical and fruitful of these attempts was initiated by Prof. Lotfi Zadeh in 1965 with publication of his paper "Fuzzy Sets" [92]. Fuzzy set theory has become accepted in the literature as a tool for dealing with certain forms of imprecision that frequently occur in decision making environments, but for which probability calculus is inadequate.

The main reason why fuzzy logic is so popular is indeed its resemblance with human thinking. By introduction the notion of degree on a condition and enabling the state of the condition to be other than true or false, makes fuzzy logic able to take into account uncertainties and inaccuracies, which allow for a flexible way of reasoning, similar to human reasoning. In real world, the human brain can reason with uncertainties and vagueness, we can not say definitely whether a thing is either this way or the other way. Fuzzy Logic (FL) includes a degree of answers, refers to things in imprecise terms, the same as two people do not have the same opinion over a thing. For example people do not share the same idea what temperature is considered hot. Some can say that anything over 30 degrees is considered hot (people from cold places), others can be more durable to high temperatures and consider temperatures over 35 degrees as hot (people from warm countries).

Despite the attributes it offers, FL should not be used when conventional control theory yields an acceptable result, and an adequate and solvable mathematical model already exists or can easily be created. Thus, it is noteworthy to consider the application of FL when the problem to be dealt with is a non-deterministic polynomial-time hard (NP-hard) problem. In such cases, there is no algorithm or mathematical model available that can provide a solution in polynomial time, let alone if we need to handle certain problems in real-time.

7.1.1 Fuzzy Logic and Uncertainty

In the real world uncertainty and fuzziness co-exist, but they are two contrasting concepts. The first relates to the lack of information which is the lack of knowledge of future events. Once the information is available, the uncertainty is gone. Whether, fuzziness tries to captures those uncertainties that are merely inherent in anything and deals with the nebulous terms and way of thinking. For example, in the statement ′ It will rain tomorrow with a 30% probability′ includes two types of uncertainties, a probabilistic one and fuzziness. Because of the lack of information we are not sure whether it will rain or not tomorrow. This describes the probability that the event of having rain tomorrow will occur. On the contrary, even if we refer to the current weather condition it may not be easy to determine whether it is raining or not, depending on the intensity of the water falling (there are many weather conditions: heavy fog with fine water particles condensing, drizzle, just a few raindrops) and the perception of people.

7.1.2 Classical Set Theory and Fuzzy Set Theory

The basic idea of fuzzy logic is that it associates a number to each element indicating the degree which this elements belongs to a certain set of elements. To better understand this idea, it is helpful to compare fuzzy logic with traditional logic (Boolean logic) first. To begin with, let us do a quick refresher of the Boolean logic and classical sets.

The classical set theory is the branch of mathematics that studies sets. There are a wide variety of sets; we can create set of numbers, characters, functions, set of individuals (describing a population), set of assumptions and even sets of sets. All the elements member of certain set share the same characteristics. For example {1, 4, 5, 7, 8, 43} is a set of integers and {a, e, o, u , i} is a set of vowels. Even though the order of the elements is not important, it make it easier for readability when they are in an ascending order. A classical set (also referred to as a ′crisp′ set) is denoted by a capital letter, for example A = {2, 4, 6, 8} is a set of one digit even numbers. The *universe of discourse (U)* refers to the input space of values that a certain variable is allowed to take value, thus describes the universe of all available information on a given problem. A set A is a crisp set in a given universe of discourse U when for any member of U, it can be said that A includes all members of U that meet a certain condition and excludes the rest of the members. A is defined as:

$$A = \{\ x \mid x\ meets\ a\ certain\ condition\},\ \forall x \in U$$

Other mathematical expressions which derive from the definition are:

$\overline{A} = \{ x \mid x \notin A\}, \ \forall x \in U$

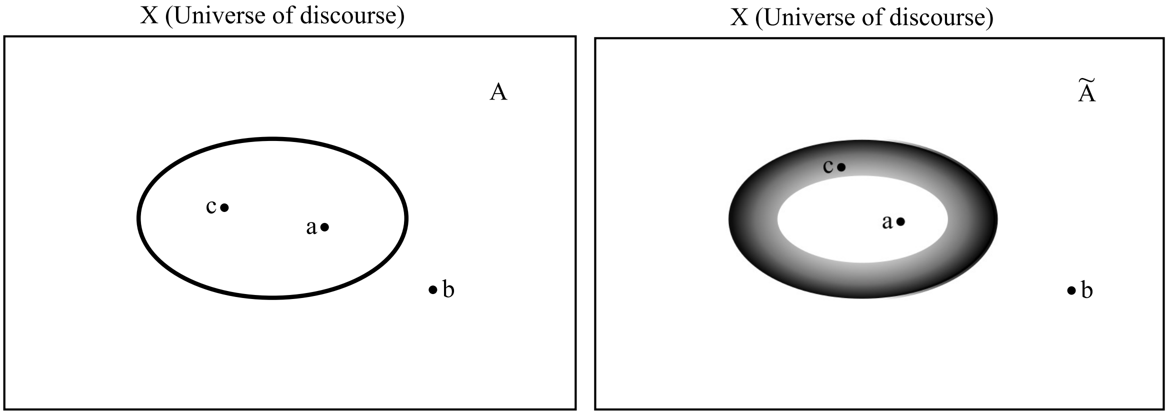
$A \cup \overline{A} = U$

$A \cap \overline{A} = \emptyset$

Alternatively, we can introduce a zero-one MF (differently named as characteristic function) for A as below:

$$A \Rightarrow \begin{cases} \mu_A(x) = 1, & \text{if } x \in A \\ \mu_A(x) = 0, & \text{if } x \notin A \end{cases}$$

The concept of belonging is important for a set as it describes whether an element is part of that set or not.



(a) Classical set theory (b) Fuzzy set theory

FIGURE 7.1: Diagrams for representing a set in (a) classical theory and (b) fuzzy theory.

Let us now compare the classical set theory with fuzzy set theory as shown in Figure 7.1. Let X be an abstraction of a universe of discourse and A a crisp (classical) set. A is a collection of elements all sharing the same characteristics. As shown in Figure 7.1(a), the classical set is defined by crisp boundaries and no uncertainty on the location of the boundaries of the set. Elements a and c are obviously a member of crisp set A and element b is not a member of set A. In classical sets no element can fall in two categories, one element is either in one category or the other, that is to say a certain point is either part of the classical set or not. On the other hand, fuzzy set is defined by ambiguous boundaries and has vague properties, as shown by the shaded boundary in Figure 7.1(b). In fuzzy sets an element can be a partial member of the set. Figure 7.1(b) shows a fuzzy set with element a as a full member of the set, and element c is a partial member of the set. The faded boundary represents the ambiguous boundary region of set A. When the point is in the central area of the fuzzy set, such as the point a, than this element is a full member of the set and such complete membership can be represented as equal to 1. When the point is outside the boundary of set A, such as point b, this element is not a member of the fuzzy set and such no-membership can be represented as equal to 0. When the point is in the boundary, such as point c its membership is ambiguous. Such partial membership is represented by an intermediate value of membership on the interval [0,1]. If point c moves closer to the central of the fuzzy set, the membership of this point approaches the value 1, and if it moves toward the fuzzy boundary the membership of point c approaches a value of 0.

For instance let us describe today's weather and decide whether it is a sunny day. By the definition of the crisp set, an element can only be or not be a part of the set. It cannot be both. So we can only express if it is a sunny day or not, no other option. But, if we were to assign a degree of membership to our concept, say, 1 to a cloud cover of 100% of the sky, 0.75 to a cloud cover of 75%, and 0 to a cloud cover 20%, then we could include sunny to our set of the weather condition to a partial degree of membership. The concept of membership is really important in fuzzy logic. It describes that an element belongs to a fuzzy set by 20%, in contrast to classical set theory where membership is either 1 (part) or 0 (not owned). In fuzzy logic the truth of any statement is a matter of a degree.

7.2 Fuzzy Logic Systems

Fuzzy logic is based on the fuzzy set theory, which is a generalization of the classical set theory. Classic set theory is a special case of fuzzy set, it is a subset of the theory of fuzzy sets. That is to say, classical sets are sets without ambiguity in their membership. Fuzzy sets are denoted by a set symbol with a tilde understrike. So, for example, $\tilde{A}$ would be the fuzzy set A. This function maps elements of a fuzzy set $\tilde{A}$ to a real numbered value on the interval [0–1]. If an element in the universe, say x , is a member of fuzzy set $\tilde{A}$, then this mapping is given by $\mu_{\tilde{A}}(x) \in [0,1]$, where $\mu_{\tilde{A}}$ expresses the membership function of set A.

Fuzzy set $\tilde{A}$ in a discrete and finite universe of discourse U is given by:
$\tilde{A} = \left\{ \frac{\mu_{\tilde{A}}(x_1)}{x_1} + \frac{\mu_{\tilde{A}}(x_2)}{x_2} + ... \right\} = \left\{ \sum_i \frac{\mu_A(x_i)}{x_i} \right\}$

Whereas, in a continuous and infinite universe of discourse U the fuzzy set $\tilde{A}$ is given by:
$\tilde{A} = \left\{ \int \frac{\mu_{\tilde{A}}(x)}{x} \right\}$

In both formulas, the horizontal line is not a quotient but rather a delimiter. The numerator in each term is the membership value in set $\tilde{A}$ associated with the element of the universe indicated in the denominator. In the first notation, the summation symbol is not for algebraic summation, but rather denotes the collection or aggregation of each element; hence, the "+" signs in the first notation are not the algebraic "add" but are an aggregation or collection operator. In the second notation, the integral sign is not an algebraic integral but a continuous function-theoretic aggregation operator for continuous variables. Both notations are due to Zadeh (1965) [92], [93].

A fuzzy logic system includes many operations and function. A fuzzy system converts a vector of linguistic inputs of data into a scalar output by the use of fuzzy rules. This process involves membership functions, if and then fuzzy rules, fuzzy logic operators, multidimensional scaling mapping of input data, aggregation and defuzzification of output sets. A typical fuzzy system is presented in Figure 7.2. The fuzzy system is comprised of four components that are the fuzzifier, rule base, inference engine, and defuzzifier. The fuzzifier takes crisp input data and convert is to fuzzy inputs by mapping it to their corresponding membership function. In this way, the fuzzifier determines to which degree the fuzzy inputs belong to fuzzy sets. Then the inference engine defines a nonlinear mapping of input fuzzy sets into output fuzzy sets. The inference engine interprets the values of the input fuzzy set, and based on the fuzzy rules, determines to what degree the antecedent is satisfied for each rule and assigns values to the output vector. If multiple conditions are applied to antecedent of a rule, fuzzy operators are used to obtain one result for that rule. It is common in many applications in fuzzy logic that multiple rules may fire simultaneously. Therefore, the outputs of all rules are aggregated and combined into a single fuzzy set. The defuzzifier coverts the fuzzy set into a single crisp value, which can be used by computers or other electronic devices.

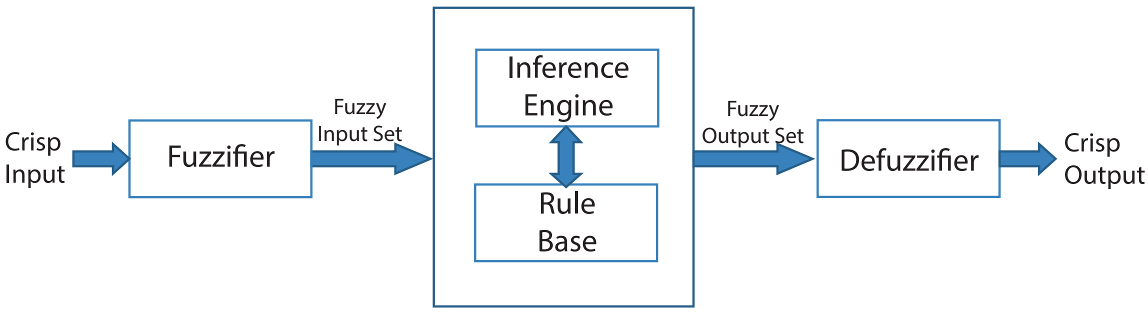


FIGURE 7.2: Block diagram of a fuzzy system.

Each of these components will be described in details in the following subsections.

Until now we have been talking about type 1 fuzzy set. Type-1 fuzzy set (T1FS) has been introduced in 1965 by prof. Zadeh. T1FS is by definition a fuzzy set in which its elements have grades of membership that are crisp, that is, their function can take any crisp value between [0,1]. In 1975, Zadeh expanded this idea as it did not exactly address the fact that people refer to the same words in different ways, and introduced type-2 fuzzy set (T2FS). T2FS is the fuzzy set whose elements have fuzzy grades of memberships. Which means that their membership values are fuzzy sets on [0, 1]. In other words T2FS is a "fuzzy-fuzzy set". This has lead to the type-n fuzzy set that is a fuzzy set in U whose membership values are type n - 1, n > 1 fuzzy sets on [0, 1]. However, for n > 3 fuzzy set is very difficult to be implemented for real-world applications, and therefore it is rarely used in practice. Even though T2FS offers better universal approximation property, higher accuracy, and is capable of handle problems with high level of uncertainty in a more complete way, T1FS on the other hand offers other advantages compared to their counterpart and has given better result satisfying the requirements of certain applications. First, because type-2 fuzzy logic systems (T2FLSs) are slower than type-1 fuzzy logic systems (T1FLSs). Due to multiple calculations and algorithms used a delay is added in the system which varies in the range of seconds. For this reason, they are not suitable for real-time applications, such are safety applications in vehicular networks, where time is the most critical component. Second, they have a very high computational cost. The extra computations require more powerful processors, which increases the cost of the overall system. Third, T1FLS have demostarted good results for applications with no/some uncertainties. For these reasons, we consider T1FLS as more suitable for vehicular network applications and the aspects where T1FLSs win over T2FLSs are aspects where we cannot make a trade-off for our system.

7.3 Fuzzification

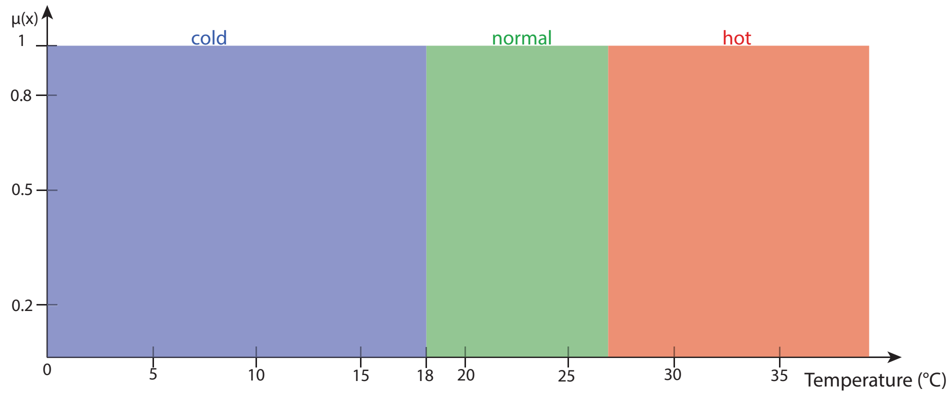
The fuzzification process includes linguistic variables and membership functions. For an sake of simplicity we will explain the terms used and fuzzy logic system implementation by taking an example that is deciding whether the weather is suitable for going out.

7.3.1 Linguistic Variables

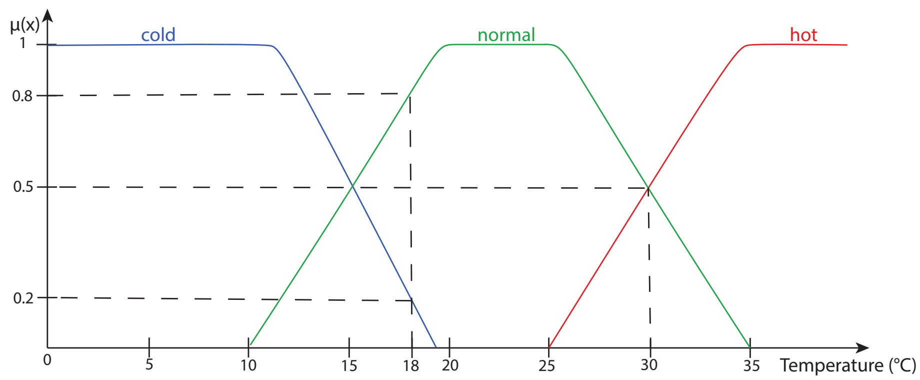
The strict defined language that we use can not describe precisely an event, a statement; therefore we lose some intent of the real statement. Moreover, a strict defined value for describing a statement is not always true for everybody. If we want to encode the opinion of people for a certain statement as a whole, we need to take into account the fuzziness or the impreciseness of language in our logical rules. Fuzzy logic uses linguistic variables to describe the control parameters. A very important property of the linguistic variables is the capability of describing imprecise parameters through words rather than strict defined values. Although words are inherently less precise than numbers, their use is closer to human intuition. For example *Environment Temperature* is the linguistic variable which contains these values: {cold, normal, hot}. The same can be done for describing *Wind Velocity*={low, moderate, high}. By using relatively simple linguistic expressions is possible to describe and grasp very complex problems. The set of values that a linguistic variable takes is identified with the terminology term set. *Environment Temperature* and *Wind Velocity* are the input variables that will be used later in the fuzzy logic system.

7.3.2 Membership Functions

Membership functions are curves that describe the degree to which a certain input variable corresponds to given set. In classical logic, if we assign elements to a crisp set membership function, we will get a very discontinuous representation of the true underlying model; therefore, we loose some of the real values. In contrary, we will achieve a smooth transition and a better representation using fuzzy logic membership functions. By definition, a fuzzy membership function defines how each point in the universe of discourse U is mapped to a membership value, and it is often associated with the notation : $U \rightarrow [\,0, 1\,]$



(a) Crisp membership function



(b) Fuzzy membership function

FIGURE 7.3: Membership function of Environment Temperature variable in (a) crisp membership function and (b) fuzzy membership function.

For example, let us consider the fuzzy set *environment temperature*. If we pool for example 100 people and ask where do they think the transition point of cold, normal, and hot temperatures is, we are going to get different values, because of course these are vague terms. Let's assume that all participants thought that a temperature more than 35 degree celcius is considered hot, but only half of them thought that 30 degree celcius is hot and none of them thought that 25 degree celcius is hot. This traces out the particular membership function for hot *environment temperature*. We can do the same for normal and cold temperature. We have now defined the membership functions of our input variable, which is the *environment temperature*. We follow the same idea to assess *wind velocity* and the output variable which in this case is deciding the *weather condition*. We will get the graphs in Figure 7.3(b) which are the sets of membership functions that indicate how people would convert a vague term of cold, normal, hot weather into a fuzzy value. We end up with these three overlapping membership functions. Now we can see that the truth of a statement does not need to belong to a single set, true or false, but rather they can belong to multiple sets based on the degree of truth. Whereas, the crisp membership function of the fuzzy set *environment temperature* is shown in Figure 7.3(a). The terms cold, normal, hot are used in a strict sense. The temperature from *18* degree celcius to *27* degree celcius are considered normal, less than *18* degree celcius is considered cold, and more than *27* degree celcius is considered hot. Where would fall the temperature of *18* degree celcius in fuzzy membership function? Intuitively we think that is mostly normal. And according to our membership function we can say that is partially true. *18* degree celcius is 80% normal and 20% cold. A temperature of *18* degree celcius may be a member of the set normal and also a member of the set cold; only the degree of membership varies with these sets. Therefore, an input value of *18* becomes the fuzzy number [0.2, 0.8, 0] which expresses that it falls 20% to the cold membership function, 80% to the normal membership function and 0% to the hot membership function. And this process of going from that crisp input value to a fuzzy value (fuzzy set) is called fuzzification. In fuzzy logic, crisp input with precise values are converted to fuzzy set through the fuzzification process. Membership functions are used in the fuzzification and defuzzification steps of an FLS to map the non-fuzzy input values to fuzzy linguistic terms and vice versa.

Different applications use various types of membership functions, including triangular, polynomial curves, trapezoidal, Gaussian curves, generalized bell shaped, and sigmoid functions.

In our system we use trapeziodal and triangular membership fuctions, because those are considered suitable for real-time operations.

A triangular MF, denoted with $\mu_f(x)$, is defined by the element x_0 which is the only element of the *core*, a left width (left boundary) w_l and a right width (right boundary) w_r.

$$\mu_f(x; x_0, w_l, w_r) = \begin{cases} 0, & \text{if } x \leq x_0 - w_l \\ 1 - \frac{x_0 - x}{w_l}, & \text{if } x_0 - w_l < x \leq x_0 \\ 1 + \frac{x_0 - x}{w_r}, & \text{if } x_0 < x < x_0 + w_r \\ 0, & \text{if } x \geq x_0 + w_r \end{cases}$$

A trapezoidal MF, denoted with $\mu_g(x)$, is defined by the elements x_0 and x_1 which are the edge elements of the *core*, a left width (left boundary) w_l and a right width (right boundary) w_r.

$$\mu_g(x; x_0, x_1, w_l, w_r) = \begin{cases} 0, & \text{if } x \leq x_0 - w_l \\ 1 - \frac{x_0 - x}{w_l}, & \text{if } x_0 - w_l < x < x_0 \\ 1, & \text{if } x_0 \leq x \leq x_1 \\ 1 + \frac{x_1 - x}{w_r}, & \text{if } x_1 < x < x_1 + w_r \\ 0, & \text{if } x \geq x_1 + w_r \end{cases}$$

7.4 Inference Engine

The inference engine defines mapping from input fuzzy sets into output fuzzy sets. Fuzzy inference system is a way to encode the known-data input into a way that computers understand in the forms of logical rules. The inference engine is comprised of fuzzy rules and determines the degree how much each particular rule impacts the final output. Interpreting the inference engine process is a four part process which includes:

1. Determine the fuzzy if-then rules in linguistic form.
2. Apply fuzzy operator if rules are comprised of multiple pars of antecedent and resolve it to a single number between 0 and 1.
3. Apply implication method to determine the degree of support of the rule for the output value.
4. Apply aggregation method if more than one rule fires at the same time. The rules are fired in parallel, therefore the order of rules does not affect the output.

7.4.1 Fuzzy Operations

Fuzzy operations are analogous to crisp logical operations. The basic crisp set operations which are also relevant to fuzzy operations include union, intersection, and complement, which essentially correspond to OR, AND, and NOT operators, respectively. The corresponding operators representing the union and intersection of two sets in fuzzy are min, max operation whereas for the not is used the complement operation, which are defined as below:

$\mu_{A \bigcup B}(x) = max[\mu_A(x), \mu_B(x)]$

$\mu_{A \bigcap B}(x) = min[\mu_A(x), \mu_B(x)]$

$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ Another way to describe the union and intersection of two sets in fuzzy is given by prof. Zadeh as:

$\mu_{A \bigcup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$

$\mu_{A \bigcap B}(x) = \mu_A(x)\mu_B(x)$

7.4.2 Fuzzy Rules

Fuzzy rules are logical rules expressed in terms of linguistic variables which specify the relationship between the input and output fuzzy sets. Using linguistic form makes them similar to human logic and therefore easier for understanding. Fuzzy relations present a degree of presence or absence of association or interaction between the elements of two or more sets. These logical rules called also fuzzy rules are decided by the expert based on his experience and knowledge in the form of linguistic rules. Rules are the encoding of knowledge, are the model that can be used for prediction. Formulating rules that are based on the existing knowledge rather than trying to develop a mathematical model that is driven my complex mechanism, makes the problem rather easy and has been proved to be accurate enough. The rules are comprised of two parts the antecedent or premise which is the if part of the rule, and the consequent or conclusion which is the then part of the rule. Interpreting an if–then rule involves two distinct steps. The first step is to evaluate the antecedent, which involves fuzzifying the input and applying any necessary fuzzy operators. The second step is implication, or applying the result of the antecedent to the consequent, which essentially evaluates the membership function. Fuzzy inference systems consist of if–then rules and uses fuzzy operation for the aggregation process. An example of fuzzy rule is given below:

> "If wind velocity is moderate and the environment temperature is hot, then the weather condition is warm."

7.4.3 Implication Method

After evaluating the antecedent and applying the fuzzy operator if the antecedent has multiple parts, we obtain a single number that represents the result of applying that rule. The implication process will assess the consequence of that rule based on this single number and will determine the weight of this rule to the output fuzzy set. In other words, the implication method applies the result of the antecedent to the consequent of the rule and determines the degree of support for the entire rule to shape the output fuzzy set. The output of this step is a fuzzy set truncated by the degree of support for the rule. There are many ways to modify the consequent, but two are most common are the min (minimum) function or prod (product) function, which are given respectively as follows:

$\mu_{A\bigcap B}(x) = min\left[\mu_A(x), \mu_B(x)\right]$

$\mu_{A\bigcap B}(x) = \mu_A(x)\,\mu_B(x)$

In our fuzzy system in this thesis we implement min method as it is the most common way to modify the output set.

7.5 Aggregation

In many applications it is common that one or more rules may fire simultaneously. In that case, the outputs for all rules fired are combined into a single fuzzy set aggregated. This process is called aggregation. Essentially, aggregation takes the outputs of each rule in the form of fuzzy sets and unifies them into a single fuzzy set. The inputs of aggregation process are the modified fuzzy sets obtained as the output from the implication method for each rule. Then, In other words, the aggregation is the modification of all consequents across the rules combined together. The output of aggregation is a single fuzzy that represents the output variable. This output is used later as input in the defuzzification process. The aggregation occurs only one time for each output variable. Since the aggregation method is commutative, the result will be the same in whatever order rules are executed. The most used functions in aggregation are max (maximum), probabilistic OR and sum (sum of the rule output sets). We use the max method because it is more straightforward and commonly used.

7.6 Defuzzification

The last step is the defuzzification method. The input of defuzzification is the fuzzy set result obtained by the aggregation process. During defuzzification the fuzzy set is mapped to a single crisp number which represent the value of the ouput variable. This crisp number can be understood by computer language and can be used for other calculations. There are many defuzzification techniques, some of them are the Center of Gravity (also known as centroid), Center of Area, Bisector of Area, Middle of Maximum, Mean of Maxima , and so on. In our fuzzy system we use the center of gravity method as this is the most commonly used method. This method is preferred over the others, because the defuzzified values tend to move smoothly around the output fuzzy region. The centroid methods finds the balance point of the area of the aggregated fuzzy set by calculating the weighted mean of that unified area of the output of the fuzzy set. The formula for centroid method is as follows:

$$y' = \frac{\int_s y_i \mu_B(y) dy}{\int_s \mu_B(y) dy}$$

We said that the output is 90% to the neutral membership function and 10% to the good, so we can chop off the top of each of these membership functions according to their percentage. And this leaves two trapezoids which we can merge into a single geometric shape. Now the defuzzified value is the x-coordinate of the centroid of this shape, which in this example is about 46%. This means that a credit score of 660 is 46% risky. So, fuzzy inference is a way of doing interpolation using vague language. In our case, we were able to approximate this true function with the interpolation of these vague terms. And the key to all of this is that baked into these terms is actual knowledge from the experience of experts.

7.7 FuzzyC Simulation Tool

Although the Fuzzy Logic Toolbox provided by MATLAB works well for analyzing, designing, and simulating FLSs, we use FuzzyC as a simulation tool for our proposed systems. FuzzyC is software written in C language that enables modeling of different FLSs with ease and accuracy, just like the MATLAB Fuzzy Logic Toolbox. It is developed in our laboratory—Information Networking and Applications (INA) Laboratory—and is maintained and improved throughout the years.

FuzzyC includes fuzzification, fuzzy operators, implication, aggregation, and defuzzification methods needed along the inference process. Even though there is not a Graphic User Interface (GUI) available for it yet, the design of the FLSs is easy and convenient. Moreover, it can work with the same file generated by the MATLAB Fuzzy Logic Toolbox, which allows a FLS designed in the latter to be executed in FuzzyC and vice versa.

There are three main reasons why we use FuzzyC over MATLAB Fuzzy Logic Toolbox. The first is that FuzzyC is more computationally efficient. It does not need powerful processors to run into, which facilitates the design of different testbeds. Second, it is faster than its counterpart, which is an aspect worth making use of in our real-time implemented systems. Last but not least, it can be integrated easily with other systems working with different intelligent algorithms; therefore, it is practical for building hybrid intelligent systems.

Chapter 8

Proposed Intelligent Systems and Testbed Implementation

This chapter describes the proposed approach and its implementation details. The first section explains the reason we choose to use FL in our system. The remainder explains in details our proposed integrated intelligent System based on Fuzzy Logic (FL) and Software Defined Networking (SDN) approach that can efficiently manage cloud-fog-edge storage, computing, and networking resources in VANETs, from a bottom-up perspective by exploiting the resources of edge layer first and then the fog and cloud resources. The integrated system is composed of three subsystems, namely Fuzzy-based System for Assessment of QoS (FS-AQoS), Fuzzy-based System for Assessment of Neighbor Vehicle Processing Capability (FS-ANVPC), and Fuzzy-based System for Cloud-Fog-Edge Layer Selection (FS-CFELS), each having a key role in the proposed approach. In addition, we propose two Fuzzy Logic Systems (FLSs) to assess the processing capability of fog layer named Fog Layer Adequacy (FLA) in SDN-VANETs. In the end, we implement a testbed with the experiment setup details to follow the system design. Each of the sections introducing the FLSs begins with a description in respect to the objectives and motivation of the system it introduces, to continue with all the implementation details one must know to obtain the same (simulation) results. The details include, more specifically, the explanation of the parameters (linguistic variables) of the implemented systems, the term set (the set of linguistic values) for each parameter, the chosen Membership Functions (MFs), and the Fuzzy Rule Base (FRB) of every Fuzzy Logic Controller (FLC) comprising the proposed FLSs.

8.1 Proposed Architecture

This section presents the architecture of our proposed approach for coordination and management of VANETs resources. The information acquired by many vehicles, not only by an individual vehicle, increases the accuracy; thus, better decisions

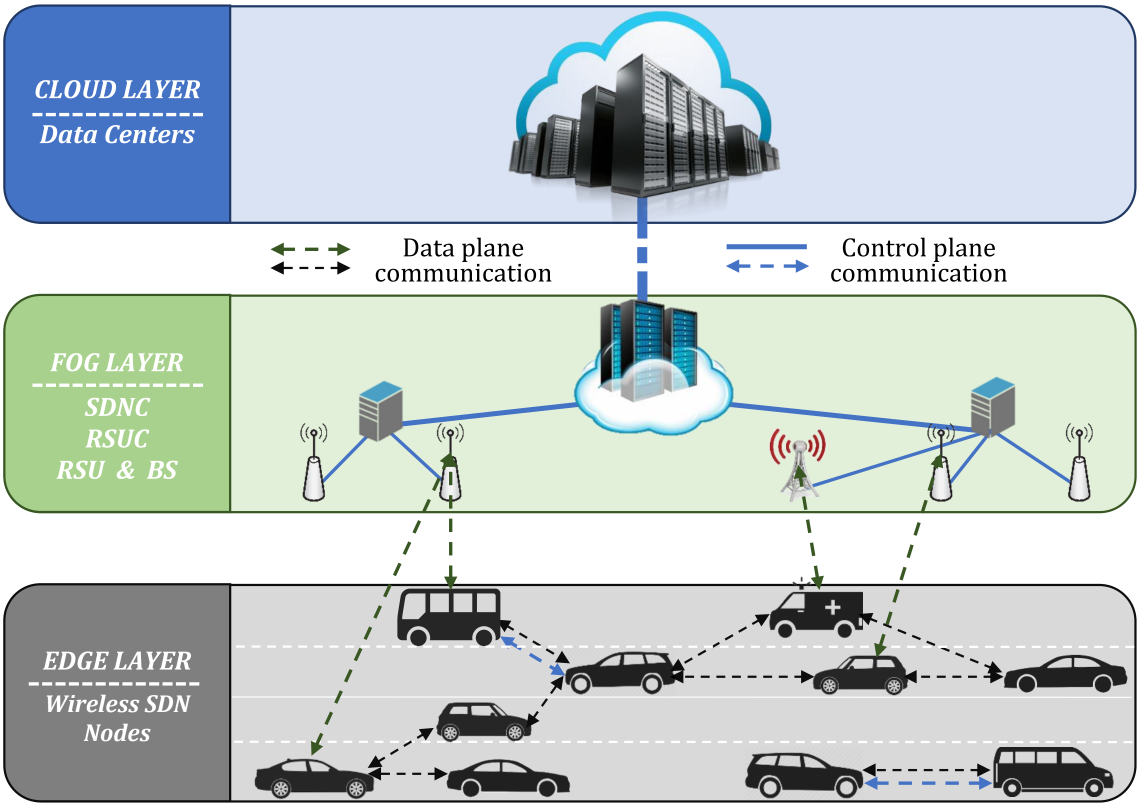


FIGURE 8.1: Layered architecture of cloud-fog-edge SDN-VANETs.

and predictions can be made. However, there is a major problem in gathering, processing, and analyzing the enormous amount of data generated while keeping the network cost at a minimum. Managing the resources of the network while providing the application's requirements is yet another challenge. Our proposed approach addresses these issues. The proposed approach considers a layered cloud-fog-edge SDN architecture that is coordinated by a fuzzy system implemented in the SDN Controller (SDNC) and SDN modules. This architecture is illustrated in Figure 8.1.

SDNC manages the resources of the edge, fog, and cloud layer and determines the appropriate layer for data storage and computing, based on the output of the fuzzy system. The edge layer includes the resources of all On-Board Units (OBUs) of the smart vehicles that are able to communicate with each other. The vehicles act not only as a relay node, but they also process and analyze the data by themselves and at the same time share their available resources with some vehicle that has a shortage of resources (hereinafter will be referred to as *the vehicle*). The fog layer consists of the RSUs, RSU Controllers (RSUCs), Base Stations (BSs), and fog servers, which are ideally only a few hops away from the vehicles. It offers more resource capabilities, compared to the edge layer, while still providing computing in real-time. Whereas the cloud layer offers abundant storage and computing facilities, located far away at the cloud data centers.

We consider this architecture from a bottom-up approach, which implies that the edge layer is the first layer considered, based on the available connections and service requirements. If the application requirements are not fulfilled or there are no or only very few available connections, then the fog layer is the layer taken into consideration. For instance, safety applications are suitable for either edge layer or fog layer, as both these layers support real-time processing and high QoS. The cloud layer is used to process applications that are delay tolerant and require long-term analytics. Through this approach, all network resources are utilized effectively and massive traffic flow in the core network is avoided.

The proposed integrates sytem called Integrated Fuzzy-based System for Coordination and Management of Resources (IFS-CMR) is implemented in the SDNC and the vehicles which are equipped with an SDN module. In case a vehicle does not have an SDN module, it sends the information to SDNC which computes the needed information and sends back its decision.

In the next sections, we give details of the composition of the proposed approach and describe the input and output parameters of each subsystem. In addition, we present the data gathering and communication module which is integrated into the vehicles and explain the FL Controllers (FLCs) of the system.

8.2 Data Gathering and Communication Module

The vehicles are equipped with various sensors (lidar, radar, ultrasonic, camera, etc.) which are placed internally and externally to acquire information about the vehicle itself (its speed, direction, steering wheel movements, tire slip, distance between the lane and other nearby vehicles, and so forth) and the condition of roads (congested roads, inadequate traffic signs, potholes, ice patches, or other hazards); a wireless transceiver device which supports different wireless technologies that enable communications with other entities; a GPS device that provides precise information about location; and an OBU which controls the communication of vehicle with other entities and offers computing, storage, and networking facilities [94]. We have included all these components in the data gathering and communication module, as shown in Figure 8.2.

Once the data is sensed and processed, the vehicles then share with one another important information through beacon messages. They broadcast these beacons periodically to other vehicles via V2V communication links to increase cooperative awareness between them. The beacon messages are critical for our approach as well. Once the vehicles receive them, they extract the information they need, which for IFS-CMR is about their neighbors' geographic position, speed, direction, transmission

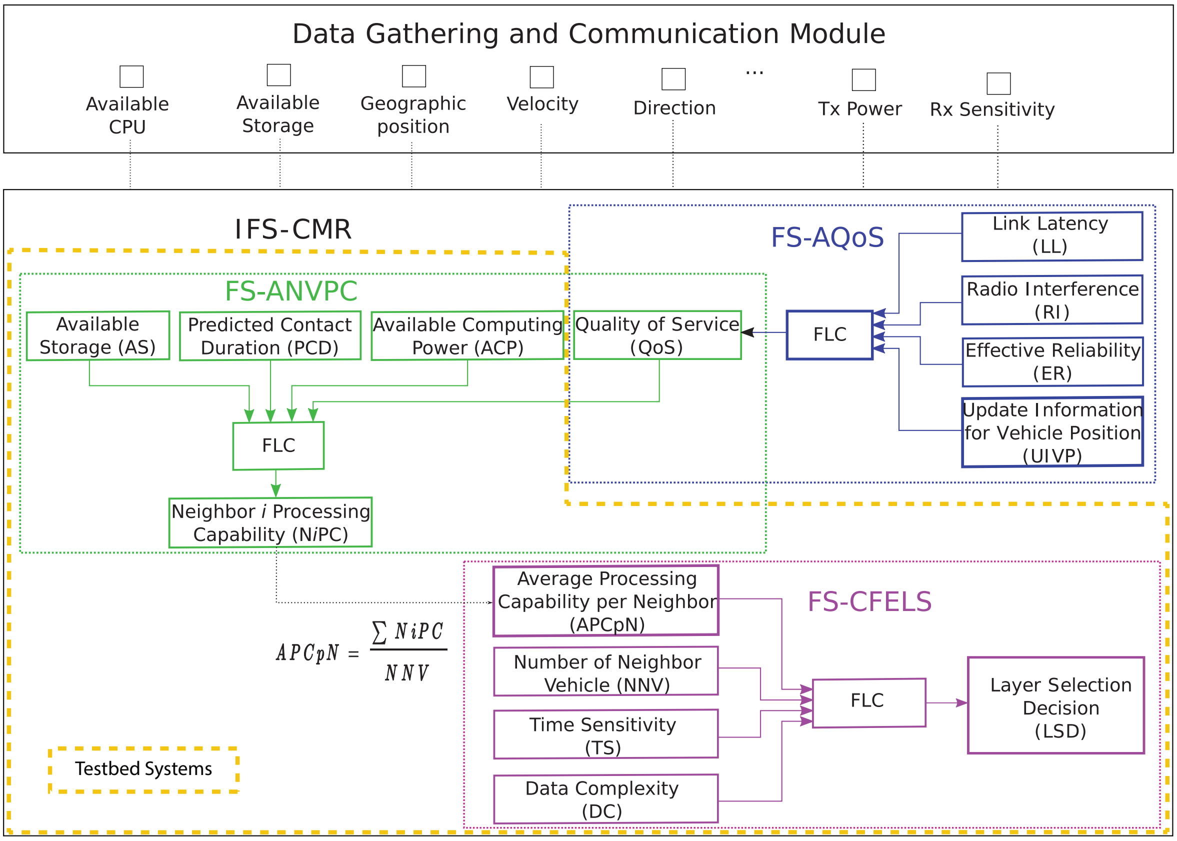


FIGURE 8.2: Structure of fuzzy integrated system and testbed.

power, available storage, available computing power, etc. Then, IFS-CMR makes the necessary calculations to update the current condition of all the input parameters.

8.3 Implementation Issues

We use FL to implement the proposed systems because the problems we have to deal with involve many uncorrelated parameters, and such problems are classified as NP-hard problems. Different approaches can be used to deal with such problems, but FL is the most efficient approach in solving decision-making and control problems in real-time. Selecting the proper type of fuzzy sets and methods—the methods used for the processes of fuzzification, implication, aggregation, and defuzzification—that suit the application requires experience and knowledge, and this is only the first step of the design of an FLS (see Chapter 6).

The next big step after the identification of the challenges is the conception of the approach that can solve those challenges. To design the best problem-solving approach, all the factors that are a cause of a problem should be pointed out and taken into consideration. In our case, the factors have to be, in some kind of way, an input of the system, with the system output having the ability to control the desired outcome. While this stands for all kinds of areas and problem-solving methods, it is necessary to mention this in the design of FLSs for the following reasons.

First, when only a factor or two are a cause of the problem, a solvable mathematical model can easily be created, and therefore there is no need for an FL approach.

Second, as opposed to the case of very few inputs, there should not be many inputs in a single FLC either because this increases the complexity.

Third, one may not want to have many separate FLSs, with one FLS solving only some issues, since that would lead to the emerging of different problems. For example, a simple question stands to the fact that if we should implement only one system, only some, or all systems at the same time. If the decision is not to implement all the proposed systems, then some issues will not be covered since one cannot predict what will the problem be and implement it right away before it happens. On the other hand, if all the systems are implemented, the systems that cover similar issues could interfere with each other. Moreover, there might be a need to solve several issues at a time, and therefore there should be a way to coordinate all the implemented systems, which is certainly not an easy task.

Once all these factors are taken into consideration (the input parameters are decided), there are other important issues to be resolved, such as:

1. Determining the suitable number of the linguistic terms for both input and output parameters,
2. Finding the appropriate type of MF that represents better its linguistic term,
3. Deciding the proper range of numerical values (granularity) for each term set,
4. Constructing a flawless rule base that can control the system in the best way.

Sorting out all these issues to construct the ideal FLS is practically impossible; however, a satisfactory, near-optimum solution is most of the time acceptable, and that could be provided with some effort. These efforts include the process of tuning all the above components until the desired results are achieved.

In our systems, as mentioned in Section 7.3.2, we use triangular and trapezoidal MFs since they are suitable for real-time operation. We carried out many simulations to decide the number of the terms for each parameter (input and output) and the granularity for each term set—which includes the core and the support for each MF. For a linguistic parameter x, the set of its linguistic values is denoted by $T(x)$, and the number of the linguistic values by $\mid T(x) \mid$. The set of the MFs associated with the term set of the parameter x is represented by $M(x)$. Based on the linguistic description of input and output parameters, we then construct the FRB. The FRB of each FLC that a FLS is comprised of forms a fuzzy set of dimensions equivalent to $\mid T(x_1) \mid \times \mid T(x_2) \mid \times \cdots \times \mid T(x_n) \mid$, where $\mid T(x_i) \mid$ is the number of terms on $T(x_i)$ and n is the number of input parameters.

8.4 IFS-CMR Parameters

The parameters of IFS-CMR are described in the following.

- **Link Latency (LL):** Latency is a strong requirement in VANETs. Many applications, especially those related to safety, must run in real-time and the network must provide a very low latency despite the rapid and high topology changes. Therefore, the time it takes the first bit to get from the sender to the destination is crucial in providing high QoS.
- **Radio Interference (RI):** Radio Interference indicates the unwanted signals that come from transmissions of adjacent vehicles and disrupt the reception of information. These signals can cause problems from low data speed transmissions to even complete loss of information.
- **Effective Reliability (ER):** We define effective reliability as the capacity of the network to successfully deliver messages to its destination. There are many factors that influence ER that include the bandwidth of transmission medium, number of collisions, and buffer size, among others.
- **Update Information for Vehicle Position (UIVP):** It is necessary for vehicles to have the coordinates of other surrounding vehicles in order to detect dangerous situations or to monitor traffic. However, too many packets occupy more bandwidth, whereas few packets cannot accurately discover the position of neighboring vehicles.
- **Quality of Service (QoS):** Each application has different requirements in terms of latency, bandwidth, throughput, and so on. Safety applications, for example, require real-time communications over reliable links. The latency constraints for such applications are in the range of a few milliseconds [95]. However, satisfying QoS requirements at all times is not a straightforward task given the highly dynamic topology and interference on these networks.
- **Available Computing Power (ACP):** Recent VANET applications require significant computational resources and real-time processing. The intelligent vehicles in the new generation of VANETs are capable of handling some of these applications. Vehicles use their computing power to run their own applications, but they can also allocate some of it for other vehicles to help them in case they need additional computing power to complete certain tasks. When vehicles are willing to share their resources, they let their neighbors know by sending them information about the amount they want to share. In other words, they decide the number of physical processor cores and the amount of memory that other vehicles can use.

- **Predicted Contact Duration (PCD):** In a V2V communication, the duration of the communication session is important since it determines the amount of data to be exchanged and the services that can be performed. A vehicle in need of additional resources (*the vehicle*) will have to set up virtual machines on the neighbors that are willing to lend their resources; therefore, the contact duration becomes even more important since much more time is needed to accomplish these tasks than just performing a data exchange. Since the vehicles change their direction or speed, we can only make a prediction of their contact duration based on the value of the parameters at the time when the beacon message was transmitted (for more accuracy, the PCD is updated each time a new beacon message from that neighbor is received). To calculate the PCD between *the vehicle* and a neighbor vehicle i (see Figure 8.3 for illustration), we first calculate the relative speed between these two vehicles using the law of cosines, as given in Equation (8.1).

$$RSV_i = \sqrt{V^2 + V_i^2 - 2VV_i \cos\theta_i} \tag{8.1}$$

where V is the speed of *the vehicle*, V_i is the speed of neighbor i, and θ_i is the angle between their directions. Then, we use the law of cosines once again to calculate the PCD, as given in Equation (8.2).

$$(RSV_i \cdot PCD)^2 + D_0^2 - 2|RSV_i| \cdot PCD \cdot D_0 \cos(\gamma_i + \beta_i) = CR^2 \tag{8.2}$$

where D_0 denotes the initial distance between the two vehicles, CR is the communication range, γ_i is the angle between the direction of *the vehicle* and D_0 imaginary line; whereas β_i is calculated with the Equation (8.3), which is derived from the law of sines.

$$\beta_i = \begin{cases} \arcsin(\frac{V_i \sin\theta_i}{|RSV_i|}), & \text{for } V_i \leq \sqrt{V^2 + RSV_i^2} \\ 180 - \arcsin(\frac{V_i \sin\theta_i}{|RSV_i|}), & \text{for } V_i > \sqrt{V^2 + RSV_i^2}, \theta \geq 0 \\ -180 - \arcsin(\frac{V_i \sin\theta_i}{|RSV_i|}), & \text{for } V_i > \sqrt{V^2 + RSV_i^2}, \theta < 0 \end{cases} \tag{8.3}$$

We posit that when two vehicles are getting farther from each other from different directions, their directions form a positive angle, whereas when the vehicles are getting closer, θ is negative.

- **Available Storage (AS):** Since vehicles generate and receive enormous amounts of data, their storage might be insufficient despite their large storage resources. If *the vehicle* needs to use also the storage of the neighbors, that should be large enough to allow *the vehicle* to run the virtual machine. This storage is used also

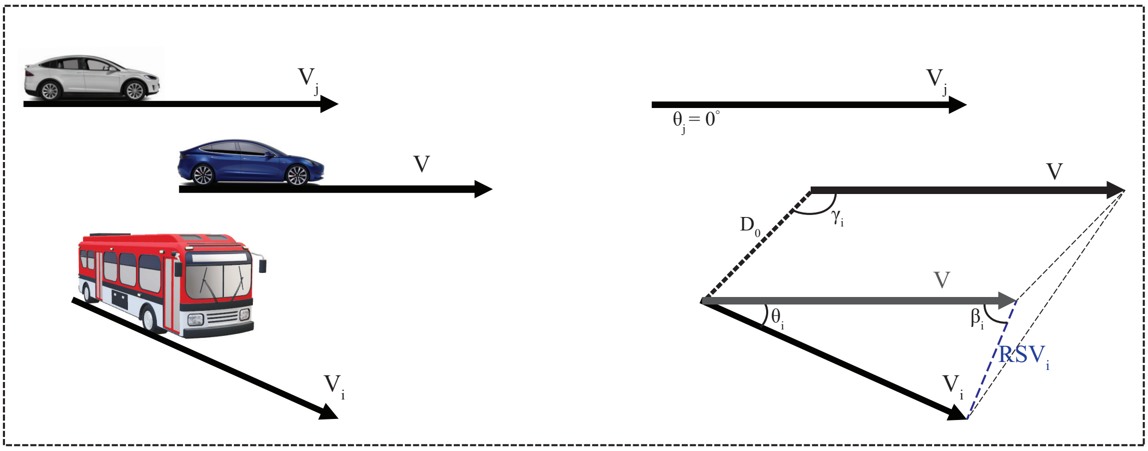


FIGURE 8.3: Graphical representation of vehicles moving at different velocities and directions.

to store data after completing specific tasks of all the tasks these neighbors are asked to accomplish.

- **Neighbor i Processing Capability (NiPC):** Describes the capability of a vehicle to help another *vehicle* that lacks the appropriate resources to accomplish certain tasks. The values of this parameter range between 0 and 1, with the value 0 implying that the neighbor cannot help at all and 1 that the neighbor is in the best condition to help out *the vehicle*.

- **Average Processing Capability per Neighbor Vehicle (APCpNV):** This parameter is the average of the Processing Capability (PC) of all neighboring vehicles within *the vehicle's* communication range. It is an important parameter that represents the capability of the edge layer, and it is calculated as the sum of the PC of each neighbor vehicle divided by the number of neighboring vehicles.

- **Number of Neighboring Vehicles (NNV):** This parameter changes continuously due to the vehicles moving out of *the vehicle's* communication range and the ones that appear. Vehicles traveling at the opposite direction lead to even more frequent changes. Since the bigger the angle between the directions, the bigger the distance created between the vehicles, we consider only the neighbors whose directions with the direction of *the vehicle* create angles that are smaller than 90°. Vehicles traveling in directions that create bigger angles move out of the communication range very quickly, making it impossible for *the vehicle* to use their resources.

- **Time Sensitivity (TS):** Different applications have different requirements in terms of latency. For instance, safety applications require a strict latency to be guaranteed, ideally <1 ms, whereas comfort and entertainment applications can tolerate latencies up to some seconds and are considered delay-tolerant [95].

System updates and the data collected for long-term analytics can tolerate even longer latencies, thus for such applications, the latency is not considered a requirement at all.

- **Data Complexity (DC):** There are many factors that dictate the data complexity, and the volume is only one of them. Even a single application might use data that differ in type and structure, not to mention that they may come from many disparate sources (e.g., sensors, cameras, radar, lidar). Besides, there are different kinds of applications that include not only VANET applications, as in the Big Data era vehicles can also be used to compute data non-related to VANETs. However, not all the data need considerable processing as some of them are in the form of small messages that are used to inform the vehicles for particular situations.

- **Layer Selection Decision (LSD):** The output parameter values, which are always between 0 and 1, denote three decisions—the interval [0, 0.3] indicates that *the vehicle* can use the edge layer resources, the values in the interval (0.3, 0.7) specify that the layer to be used is the fog layer, whereas the values in [0.7, 1] specify the cloud layer as the most appropriate layer to run their applications.

8.5 Description of IFS-CMR Subsystems

The input parameters of each subsystem of IFS-CMR do not correlate to one another, leading to an NP-hard problem. Problems with three or more uncorrelated parameters are classified as NP-hard because finding a mathematical model that can calculate an output in polynomial-time is practically impossible. Heuristic or meta-heuristic approaches are proven to provide adequate solutions for these kinds of problems, but each method has a limited scope to which it can be applied. For instance, genetic algorithms give good solutions for optimization and allocation problems. Neural networks can be applied to recognition problems and rule learning. FL, on the other hand, can be used to provide a solution for decision-making and control problems in real-time, especially when the system contends with high levels of imprecision and uncertainty [96]–[101].

Making real-time decisions while dealing with imprecision and uncertainty is the advantage of IFS-CMR. The characteristics of the network created between vehicles change continually and rapidly; therefore, the resources must be managed in real-time. Moreover, because of the rapid changes, the decision is reached in presence of much imprecision.

IFS-CMR is comprised of three integrated subsystems (FS-AQoS, FS-ANVPC, and FS-CFELS), each controlled by its respective FLC. Each subsystem has a key role

TABLE 8.1: Parameters and term sets for FS-AQoS.

Models		Parameters	Term Sets
FS-AQoS2	FS-AQoS1	LL	Low (Lo), Medium (Me), High (Hi)
		ER	Not Effective (Nef), Medium Effective (Mef), Effective (Ef)
		UIVP	Few (Fw), Moderate (Mo), Many (Ma)
		RI	Permissible (Pe), Acceptable (Ac), Harmful (Ha)
FS-AQoS1 & FS-AQoS2		QoS	Extremely Low (El), Very Low (Vl), Low (Lw), Moderate (Md), High (Hg), Very High (Vh), Extremely High (Eh)

TABLE 8.2: Parameters and term sets for FS-ANVPC.

Models		Parameters	Term Sets
FS-ANVPC2	FS-ANVPC1	ACP	Small (Sm), Medium (Me), Large (La)
		PCD	Short (Sh), Medium (Md), Long (Lo)
		AS	Small (S), Medium (M), Big (B)
		QoS	Low (Lw), Moderate (Mo), High (Hi)
FS-ANVPC1 & FS-ANVPC2		NiPC	Extremely Low (El), Very Low (Vl), Low (Lw), Moderate (Md), High (Hg), Very High (Vh), Extremely High (Eh)

TABLE 8.3: Parameters and term sets for previous models of FS-CFELS.

Models		Parameters	Term Sets
FS-CFELS2	FS-CFELS1	DS	Small (S), Medium (M), Big (B)
		TS	Low (L), Middle (Mi), High (H)
		VRSNV	Slower (Sl), Same (Sa), Faster (Fa)
		NNV	Low (Lo), Moderate (Mo), High (Hi)
FS-CFELS1		LSD	Decision Level 1 (DL1), DL2, DL3, DL4, DL5
FS-CFELS2		LSD	DL1, DL2, DL3, DL4, DL5, DL6, DL7

in the system. They have their own input parameters and their output serves the subsystem that follows. The structure of IFS-CMR is shown in Figure 8.2. The way we have built IFS-CMR has allowed us to continually improve it by discovering the implementation flaws (contrary to many AI systems, which behave as black boxes that do not provide any feedback how they reach their decision, and consequently are difficult to improve).

The term sets for the parameters of each subsystem are shown in Tables 8.1–8.4. The parameters are fuzzified using the membership functions shown in Figures 8.4–8.6. The number of terms for each parameter and the characteristics of each membership function are determined through the experience gained by running many simulations.

TABLE 8.4: Parameters and term sets for the new FS-CFELS.

Parameters	Term Sets
DC	Low (Lo), Moderate (Mo), High (Hi)
TS	Low (Lw), Middle (Md), High (Hg)
NNV	Sparse(Sp), Medium Density (Me), Dense (De)
APCpNV	Low (L), Moderate (M), High (H)
LSD	Decision Level 1 (DL1), DL2, DL3, DL4, DL5, DL6, DL7

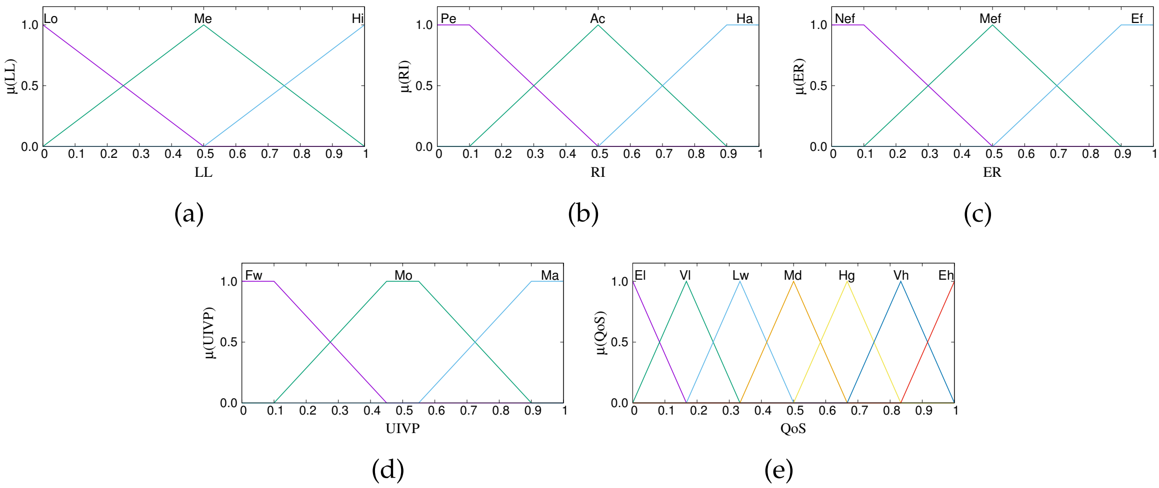


FIGURE 8.4: Membership functions of FS-AQoS. (**a**) Link Latency, (**b**) Radio Interference, (**c**) Effective Reliability, (**d**) Update Info. for Vehicle Position, and (**e**) Quality of Service.

TABLE 8.5: FRB of FS-AQoS1.

No	LL	ER	UIVP	QoS	No	LL	ER	UIVP	QoS	No	LL	ER	UIVP	QoS
1	Lo	Nef	Fw	Hg	10	Me	Nef	Fw	Vl	19	Hi	Nef	Fw	El
2	Lo	Nef	Mo	Vh	11	Me	Nef	Mo	Lw	20	Hi	Nef	Mo	Vl
3	Lo	Nef	Ma	Hg	12	Me	Nef	Ma	Vl	21	Hi	Nef	Ma	El
4	Lo	Mef	Fw	Vh	13	Me	Mef	Fw	Lw	22	Hi	Mef	Fw	Vl
5	Lo	Mef	Mo	Eh	14	Me	Mef	Mo	Md	23	Hi	Mef	Mo	Lw
6	Lo	Mef	Ma	Vh	15	Me	Mef	Ma	Lw	24	Hi	Mef	Ma	Vl
7	Lo	Ef	Fw	Eh	16	Me	Ef	Fw	Md	25	Hi	Ef	Fw	Lw
8	Lo	Ef	Mo	Eh	17	Me	Ef	Mo	Hg	26	Hi	Ef	Mo	Md
9	Lo	Ef	Ma	Eh	18	Me	Ef	Ma	Md	27	Hi	Ef	Ma	Lw

From our experience, using less than three linguistic terms for an input parameter has the risk of inefficient control and making poor decisions, whereas using more leads to redundancies and increased complexity. The same holds true for the overlap of membership functions. Less overlap results in poor decisions, more overlap brings redundancies. Regarding the shape, we use triangular and trapezoidal membership functions as they are the most suitable ones for real-time operation.

TABLE 8.6: FRB of FS-AQoS2.

No	LL	RI	ER	UIVP	QoS	No	LL	RI	ER	UIVP	QoS	No	LL	RI	ER	UIVP	QoS
1	Lo	Pe	Nef	Fw	Hg	28	Me	Pe	Nef	Fw	Vl	55	Hi	Pe	Nef	Fw	El
2	Lo	Pe	Nef	Mo	Eh	29	Me	Pe	Nef	Mo	Lw	56	Hi	Pe	Nef	Mo	Vl
3	Lo	Pe	Nef	Ma	Hg	30	Me	Pe	Nef	Ma	Vl	57	Hi	Pe	Nef	Ma	El
4	Lo	Pe	Mef	Fw	Vh	31	Me	Pe	Mef	Fw	Lw	58	Hi	Pe	Mef	Fw	El
5	Lo	Pe	Mef	Mo	Eh	32	Me	Pe	Mef	Mo	Md	59	Hi	Pe	Mef	Mo	Lw
6	Lo	Pe	Mef	Ma	Vh	33	Me	Pe	Mef	Ma	Lw	60	Hi	Pe	Mef	Ma	El
7	Lo	Pe	Ef	Fw	Eh	34	Me	Pe	Ef	Fw	Md	61	Hi	Pe	Ef	Fw	Vl
8	Lo	Pe	Ef	Mo	Eh	35	Me	Pe	Ef	Mo	Hg	62	Hi	Pe	Ef	Mo	Md
9	Lo	Pe	Ef	Ma	Eh	36	Me	Pe	Ef	Ma	Md	63	Hi	Pe	Ef	Ma	Vl
10	Lo	Ac	Nef	Fw	Md	37	Me	Ac	Nef	Fw	El	64	Hi	Ac	Nef	Fw	El
11	Lo	Ac	Nef	Mo	Hg	38	Me	Ac	Nef	Mo	Lw	65	Hi	Ac	Nef	Mo	El
12	Lo	Ac	Nef	Ma	Md	39	Me	Ac	Nef	Ma	El	66	Hi	Ac	Nef	Ma	El
13	Lo	Ac	Mef	Fw	Hg	40	Me	Ac	Mef	Fw	Vl	67	Hi	Ac	Mef	Fw	El
14	Lo	Ac	Mef	Mo	Eh	41	Me	Ac	Mef	Mo	Md	68	Hi	Ac	Mef	Mo	Vl
15	Lo	Ac	Mef	Ma	Hg	42	Me	Ac	Mef	Ma	Vl	69	Hi	Ac	Mef	Ma	El
16	Lo	Ac	Ef	Fw	Vh	43	Me	Ac	Ef	Fw	Lw	70	Hi	Ac	Ef	Fw	El
17	Lo	Ac	Ef	Mo	Eh	44	Me	Ac	Ef	Mo	Hg	71	Hi	Ac	Ef	Mo	Lw
18	Lo	Ac	Ef	Ma	Vh	45	Me	Ac	Ef	Ma	Lw	72	Hi	Ac	Ef	Ma	El
19	Lo	Ha	Nef	Fw	Lw	46	Me	Ha	Nef	Fw	El	73	Hi	Ha	Nef	Fw	El
20	Lo	Ha	Nef	Mo	Md	47	Me	Ha	Nef	Mo	Vl	74	Hi	Ha	Nef	Mo	El
21	Lo	Ha	Nef	Ma	Lw	48	Me	Ha	Nef	Ma	El	75	Hi	Ha	Nef	Ma	El
22	Lo	Ha	Mef	Fw	Md	49	Me	Ha	Mef	Fw	Vl	76	Hi	Ha	Mef	Fw	El
23	Lo	Ha	Mef	Mo	Hg	50	Me	Ha	Mef	Mo	Lw	77	Hi	Ha	Mef	Mo	El
24	Lo	Ha	Mef	Ma	Md	51	Me	Ha	Mef	Ma	Vl	78	Hi	Ha	Mef	Ma	El
25	Lo	Ha	Ef	Fw	Hg	52	Me	Ha	Ef	Fw	Lw	79	Hi	Ha	Ef	Fw	El
26	Lo	Ha	Ef	Mo	Vh	53	Me	Ha	Ef	Mo	Md	80	Hi	Ha	Ef	Mo	El
27	Lo	Ha	Ef	MaH	Hg	54	Me	Ha	Ef	Ma	Lw	81	Hi	Ha	Ef	Ma	El

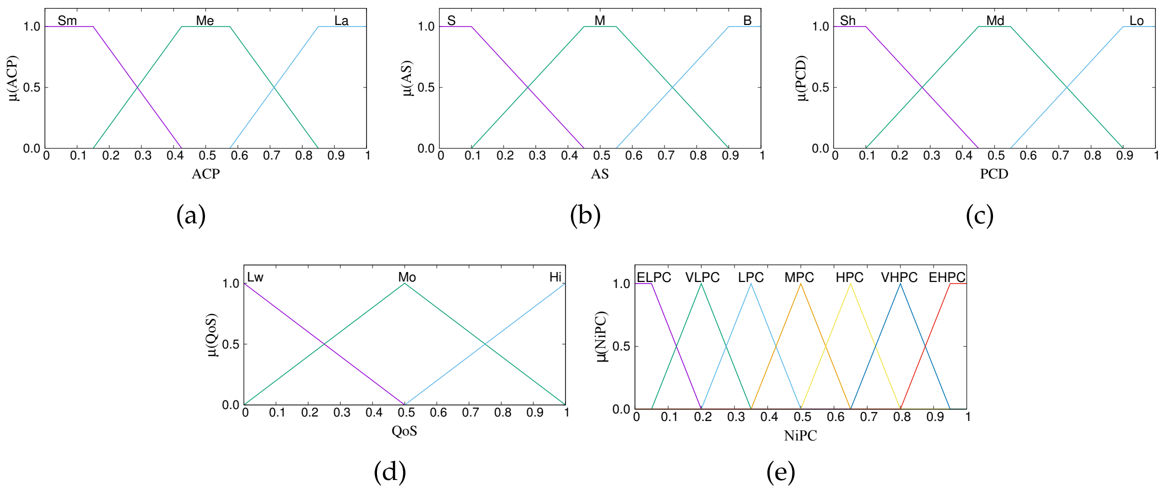


FIGURE 8.5: Membership functions of FS-ANVPC. (**a**) Available Computing Power, (**b**) Available Storage, (**c**) Predicted Contact Duration, (**d**) Quality of Service, and (**e**) Neighbor *i* Processing Capability.

In Tables 8.5–8.11, we show the FRB of FS-AQoS, FS-ANVPC, and FS-CFELS, respectively. Each subsystem has four input parameters with three linguistic terms,

TABLE 8.7: FRB of FS-ANVPC1.

No	ACP	PCD	AS	NiPC	No	ACP	PCD	AS	NiPC	No	ACP	PCD	AS	NiPC
1	Sm	Sh	S	ELPC	10	Me	Sh	S	VLPC	19	La	Sh	S	LPC
2	Sm	Sh	M	ELPC	11	Me	Sh	M	LPC	20	La	Sh	M	HPC
3	Sm	Sh	B	ELPC	12	Me	Sh	B	LPC	21	La	Sh	B	HPC
4	Sm	Md	S	VLPC	13	Me	Md	S	LPC	22	La	Md	S	MPC
5	Sm	Md	M	VLPC	14	Me	Md	M	MPC	23	La	Md	M	VHPC
6	Sm	Md	B	LPC	15	Me	Md	B	MPC	24	La	Md	B	VHPC
7	Sm	Lo	S	LPC	16	Me	Lo	S	MPC	25	La	Lo	S	VHPC
8	Sm	Lo	M	LPC	17	Me	Lo	M	HPC	26	La	Lo	M	EHPC
9	Sm	Lo	B	MPC	18	Me	Lo	B	HPC	27	La	Lo	B	EHPC

TABLE 8.8: FRB of FS-ANVPC2.

No	ACP	AS	PCD	QoS	NiPC	No	ACP	AS	PCD	QoS	NiPC	No	ACP	AS	PCD	QoS	NiPC
1	Sm	S	Sh	Lw	ELPC	28	Me	S	Sh	Lw	ELPC	55	La	S	Sh	Lw	ELPC
2	Sm	S	Sh	Mo	ELPC	29	Me	S	Sh	Mo	ELPC	56	La	S	Sh	Mo	VLPC
3	Sm	S	Sh	Hi	ELPC	30	Me	S	Sh	Hi	VLPC	57	La	S	Sh	Hi	LPC
4	Sm	S	Md	Lw	ELPC	31	Me	S	Md	Lw	VLPC	58	La	S	Md	Lw	VLPC
5	Sm	S	Md	Mo	ELPC	32	Me	S	Md	Mo	VLPC	59	La	S	Md	Mo	LPC
6	Sm	S	Md	Hi	VLPC	33	Me	S	Md	Hi	LPC	60	La	S	Md	Hi	MPC
7	Sm	S	Lo	Lw	ELPC	34	Me	S	Lo	Lw	VLPC	61	La	S	Lo	Lw	LPC
8	Sm	S	Lo	Mo	ELPC	35	Me	S	Lo	Mo	LPC	62	La	S	Lo	Mo	MPC
9	Sm	S	Lo	Hi	LPC	36	Me	S	Lo	Hi	MPC	63	La	S	Lo	Hi	VHPC
10	Sm	M	Sh	Lw	ELPC	37	Me	M	Sh	Lw	ELPC	64	La	M	Sh	Lw	VLPC
11	Sm	M	Sh	Mo	ELPC	38	Me	M	Sh	Mo	ELPC	65	La	M	Sh	Mo	LPC
12	Sm	M	Sh	Hi	ELPC	39	Me	M	Sh	Hi	LPC	66	La	M	Sh	Hi	HPC
13	Sm	M	Md	Lw	ELPC	40	Me	M	Md	Lw	VLPC	67	La	M	Md	Lw	LPC
14	Sm	M	Md	Mo	ELPC	41	Me	M	Md	Mo	VLPC	68	La	M	Md	Mo	MPC
15	Sm	M	Md	Hi	VLPC	42	Me	M	Md	Hi	MPC	69	La	M	Md	Hi	VHPC
16	Sm	M	Lo	Lw	ELPC	43	Me	M	Lo	Lw	LPC	70	La	M	Lo	Lw	MPC
17	Sm	M	Lo	Mo	VLPC	44	Me	M	Lo	Mo	MPC	71	La	M	Lo	Mo	VHPC
18	Sm	M	Lo	Hi	LPC	45	Me	M	Lo	Hi	HPC	72	La	M	Lo	Hi	EHPC
19	Sm	B	Sh	Lw	ELPC	46	Me	B	Sh	Lw	ELPC	73	La	B	Sh	Lw	LPC
20	Sm	B	Sh	Mo	ELPC	47	Me	B	Sh	Mo	VLPC	74	La	B	Sh	Mo	MPC
21	Sm	B	Sh	Hi	ELPC	48	Me	B	Sh	Hi	LPC	75	La	B	Sh	Hi	HPC
22	Sm	B	Md	Lw	ELPC	49	Me	B	Md	Lw	VLPC	76	La	B	Md	Lw	MPC
23	Sm	B	Md	Mo	VLPC	50	Me	B	Md	Mo	LPC	77	La	B	Md	Mo	VHPC
24	Sm	B	Md	Hi	LPC	51	Me	B	Md	Hi	MPC	78	La	B	Md	Hi	VHPC
25	Sm	B	Lo	Lw	VLPC	52	Me	B	Lo	Lw	LPC	79	La	B	Lo	Lw	HPC
26	Sm	B	Lo	Mo	LPC	53	Me	B	Lo	Mo	HPC	80	La	B	Lo	Mo	EHPC
27	Sm	B	Lo	Hi	MPC	54	Me	B	Lo	Hi	HPC	81	La	B	Lo	Hi	EHPC

so each FRB consists of 81 rules. The control rules have the form: IF “conditions” THEN “control action”. For instance, for FS-AQoS, for Rule 20: “IF LL is Lo, RI is Ha, ER is Nef and UIVP is Mo, THEN QoS is Md”.

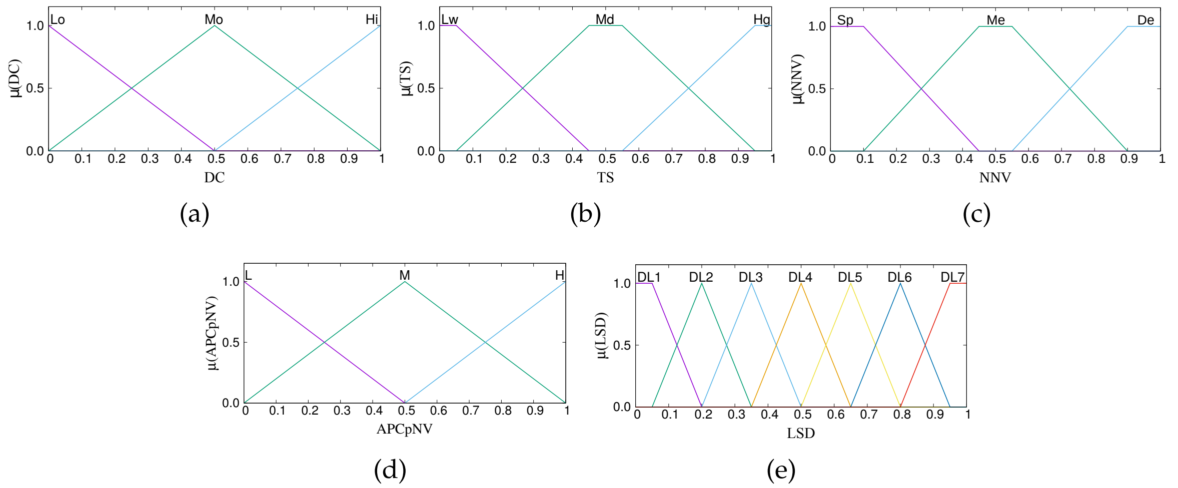


FIGURE 8.6: Membership functions of FS-CFELS. (**a**) Data Complexity, (**b**) Time Sensitivity, (**c**) Number of Neighboring Vehicles, (**d**) Avg. PC per Neighbor Vehicle, and (**e**) Layer Selection Decision.

TABLE 8.9: FRB of FS-CFELS1.

Rule	DS	TS	VRSNV	LSD	Rule	DS	TS	VRSNV	LSD	Rule	DS	TS	VRSNV	LSD
1	S	L	Sl	DL4	10	M	L	Sl	DL5	19	B	L	Sl	DL5
2	S	L	Sa	DL2	11	M	L	Sa	DL3	20	B	L	Sa	DL4
3	S	L	Fa	DL4	12	M	L	Fa	DL5	21	B	L	Fa	DL5
4	S	Mi	Sl	DL3	13	M	Mi	Sl	DL4	22	B	Mi	Sl	DL5
5	S	Mi	Sa	DL1	14	M	Mi	Sa	DL2	23	B	Mi	Sa	DL3
6	S	Mi	Fa	DL3	15	M	Mi	Fa	DL4	24	B	Mi	Fa	DL5
7	S	H	Sl	DL2	16	M	H	Sl	DL3	25	B	H	Sl	DL3
8	S	H	Sa	DL1	17	M	H	Sa	DL1	26	B	H	Sa	DL1
9	S	H	Fa	DL2	18	M	H	Fa	DL3	27	B	H	Fa	DL3

8.6 Description of Fog Layer Resources System

Vehicular cloud computing approaches are considered good solutions for handling large scales of data. However, it is essential for time-sensitive applications that a continuous low-latency service is always provided. In addition there are cases when edge layer is not always available. Therefore, fog computing is seen as a good solution because the services are migrated in close proximity of users. Fog computing is a good alternative when vehicles generate massive data, which cause increased data traffic that can not be handled by the vehicles. Fog computing leverages the processing and storage capabilities of servers located in the vicinity of vehicles, RSUs and BSs. Therefore, it reduces the transmission latency and provides a solution for high throughput demands. The proposed FLSs are implemented in the SDNC and in the vehicles equipped with SDN modules. If a vehicle does not have an SDN module, it sends a request to the SDNC, which runs the proposed system and sends back the performed evaluations.

TABLE 8.10: FRB of FS-CFELS2.

Rule	DS	TS	VRSNV	NNV	LSD	Rule	DS	TS	VRSNV	NNV	LSD	Rule	DS	TS	VRSNV	NNV	LSD
1	S	L	Sl	Lo	DL7	28	M	L	Sl	Lo	DL7	55	B	L	Sl	Lo	DL7
2	S	L	Sl	Mo	DL5	29	M	L	Sl	Mo	DL7	56	B	L	Sl	Mo	DL7
3	S	L	Sl	Hi	DL4	30	M	L	Sl	Hi	DL5	57	B	L	Sl	Hi	DL6
4	S	L	Sa	Lo	DL4	31	M	L	Sa	Lo	DL6	58	B	L	Sa	Lo	DL6
5	S	L	Sa	Mo	DL3	32	M	L	Sa	Mo	DL4	59	B	L	Sa	Mo	DL5
6	S	L	Sa	Hi	DL2	33	M	L	Sa	Hi	DL3	60	B	L	Sa	Hi	DL4
7	S	L	Fa	Lo	DL7	34	M	L	Fa	Lo	DL7	61	B	L	Fa	Lo	DL7
8	S	L	Fa	Mo	DL5	35	M	L	Fa	Mo	DL7	62	B	L	Fa	Mo	DL7
9	S	L	Fa	Hi	DL4	36	M	L	Fa	Hi	DL5	63	B	L	Fa	Hi	DL6
10	S	Mi	Sl	Lo	DL5	37	M	Mi	Sl	Lo	DL6	64	B	Mi	Sl	Lo	DL7
11	S	Mi	Sl	Mo	DL4	38	M	Mi	Sl	Mo	DL5	65	B	Mi	Sl	Mo	DL6
12	S	Mi	Sl	Hi	DL2	39	M	Mi	Sl	Hi	DL4	66	B	Mi	Sl	Hi	DL4
13	S	Mi	Sa	Lo	DL3	40	M	Mi	Sa	Lo	DL4	67	B	Mi	Sa	Lo	DL5
14	S	Mi	Sa	Mo	DL2	41	M	Mi	Sa	Mo	DL3	68	B	Mi	Sa	Mo	DL3
15	S	Mi	Sa	Hi	DL1	42	M	Mi	Sa	Hi	DL2	69	B	Mi	Sa	Hi	DL2
16	S	Mi	Fa	Lo	DL5	43	M	Mi	Fa	Lo	DL6	70	B	Mi	Fa	Lo	DL7
17	S	Mi	Fa	Mo	DL4	44	M	Mi	Fa	Mo	DL5	71	B	Mi	Fa	Mo	DL6
18	S	Mi	Fa	Hi	DL2	45	M	Mi	Fa	Hi	DL4	72	B	Mi	Fa	Hi	DL4
19	S	H	Sl	Lo	DL3	46	M	H	Sl	Lo	DL5	73	B	H	Sl	Lo	DL6
20	S	H	Sl	Mo	DL2	47	M	H	Sl	Mo	DL3	74	B	H	Sl	Mo	DL4
21	S	H	Sl	Hi	DL1	48	M	H	Sl	Hi	DL2	75	B	H	Sl	Hi	DL3
22	S	H	Sa	Lo	DL1	49	M	H	Sa	Lo	DL2	76	B	H	Sa	Lo	DL3
23	S	H	Sa	Mo	DL1	50	M	H	Sa	Mo	DL1	77	B	H	Sa	Mo	DL2
24	S	H	Sa	Hi	DL1	51	M	H	Sa	Hi	DL1	78	B	H	Sa	Hi	DL1
25	S	H	Fa	Lo	DL3	52	M	H	Fa	Lo	DL5	79	B	H	Fa	Lo	DL6
26	S	H	Fa	Mo	DL2	53	M	H	Fa	Mo	DL3	80	B	H	Fa	Mo	DL4
27	S	H	Fa	Hi	DL1	54	M	H	Fa	Hi	DL2	81	B	H	Fa	Hi	DL3

The structures of the proposed systems are shown in Figure 8.7. We focus only on the fog layer resources, for which we implement two FLSs: FLS1 and FLS2. They assess the capability of the fog layer, named Fog Layer Adequacy (FLA), to handle a certain application request from the vehicle based on different input parameters. Both FLSs determine whether the fog computing layer is appropriate and satisfies certain needs in terms of data processing. The fog layer needs to assure the requirements of real-time applications and the safety applications, in particular, in case of incapability of the edge layer. For the implementation of FLS1, we consider three input parameters: Vehicle-to-Server Latency (VSL), Server History (SH) and Current Server Load (CSL) to determine FLA. Whereas for FLS2, we include Migration Speed (MS) for assessing the fog layer processing capability. We describe the considered parameters in the following.

- **VSL**: Some applications impose strict latency requirements. For instance, platooning and remote driving require communication latency of less than 25ms and 5ms, respectively [1]. The aim of fog computing is to move data collection and processing closer to where the data is produced and used. Therefore the communication latency is maintained low and a real-time communication is possible to achieve.

TABLE 8.11: FRB of FS-CFELS.

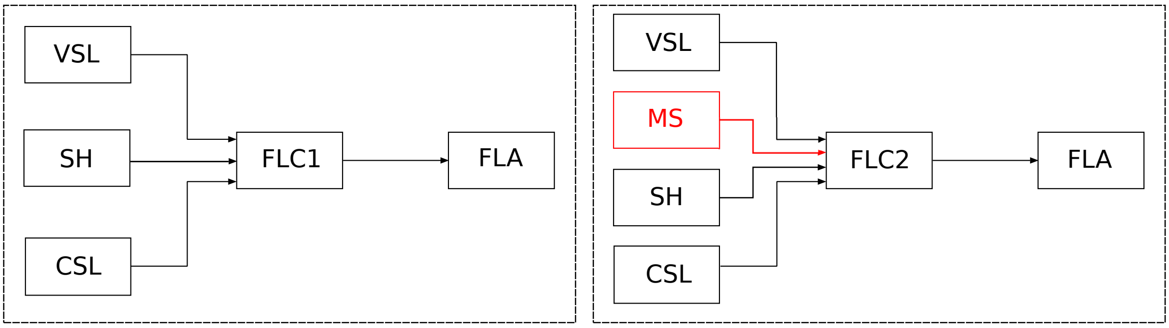
No	DC	TS	NNV	APCpNV	LSD	No	DC	TS	NNV	APCpNV	LSD	No	DC	TS	NNV	APCpNV	LSD
1	Lo	Lw	Sp	L	DL6	28	Mo	Lw	Sp	L	DL7	55	Hi	Lw	Sp	L	DL7
2	Lo	Lw	Sp	M	DL4	29	Mo	Lw	Sp	M	DL6	56	Hi	Lw	Sp	M	DL7
3	Lo	Lw	Sp	H	DL3	30	Mo	Lw	Sp	H	DL4	57	Hi	Lw	Sp	H	DL6
4	Lo	Lw	Me	L	DL6	31	Mo	Lw	Me	L	DL7	58	Hi	Lw	Me	L	DL7
5	Lo	Lw	Me	M	DL3	32	Mo	Lw	Me	M	DL5	59	Hi	Lw	Me	M	DL6
6	Lo	Lw	Me	H	DL2	33	Mo	Lw	Me	H	DL3	60	Hi	Lw	Me	H	DL5
7	Lo	Lw	De	L	DL6	34	Mo	Lw	De	L	DL6	61	Hi	Lw	De	L	DL7
8	Lo	Lw	De	M	DL2	35	Mo	Lw	De	M	DL4	62	Hi	Lw	De	M	DL5
9	Lo	Lw	De	H	DL1	36	Mo	Lw	De	H	DL2	63	Hi	Lw	De	H	DL4
10	Lo	Md	Sp	L	DL5	37	Mo	Md	Sp	L	DL7	64	Hi	Md	Sp	L	DL7
11	Lo	Md	Sp	M	DL3	38	Mo	Md	Sp	M	DL5	65	Hi	Md	Sp	M	DL6
12	Lo	Md	Sp	H	DL2	39	Mo	Md	Sp	H	DL4	66	Hi	Md	Sp	H	DL5
13	Lo	Md	Me	L	DL4	40	Mo	Md	Me	L	DL6	67	Hi	Md	Me	L	DL7
14	Lo	Md	Me	M	DL2	41	Mo	Md	Me	M	DL4	68	Hi	Md	Me	M	DL5
15	Lo	Md	Me	H	DL1	42	Mo	Md	Me	H	DL3	69	Hi	Md	Me	H	DL4
16	Lo	Md	De	L	DL3	43	Mo	Md	De	L	DL5	70	Hi	Md	De	L	DL7
17	Lo	Md	De	M	DL1	44	Mo	Md	De	M	DL3	71	Hi	Md	De	M	DL4
18	Lo	Md	De	H	DL1	45	Mo	Md	De	H	DL2	72	Hi	Md	De	H	DL3
19	Lo	Hg	Sp	L	DL4	46	Mo	Hg	Sp	L	DL5	73	Hi	Hg	Sp	L	DL5
20	Lo	Hg	Sp	M	DL3	47	Mo	Hg	Sp	M	DL4	74	Hi	Hg	Sp	M	DL5
21	Lo	Hg	Sp	H	DL2	48	Mo	Hg	Sp	H	DL3	75	Hi	Hg	Sp	H	DL4
22	Lo	Hg	Me	L	DL3	49	Mo	Hg	Me	L	DL4	76	Hi	Hg	Me	L	DL5
23	Lo	Hg	Me	M	DL2	50	Mo	Hg	Me	M	DL3	77	Hi	Hg	Me	M	DL4
24	Lo	Hg	Me	H	DL1	51	Mo	Hg	Me	H	DL2	78	Hi	Hg	Me	H	DL3
25	Lo	Hg	De	L	DL2	52	Mo	Hg	De	L	DL3	79	Hi	Hg	De	L	DL4
26	Lo	Hg	De	M	DL1	53	Mo	Hg	De	M	DL2	80	Hi	Hg	De	M	DL3
27	Lo	Hg	De	H	DL1	54	Mo	Hg	De	H	DL1	81	Hi	Hg	De	H	DL2

- **MS**: Due to VANETs dynamic characteristics and limited coverage provided by a single RSU or BS, it is inevitable that vehicles will move out of the co-located fog server service area. Therefore, the active session must be transferred to another fog server without delay while guaranteeing the quality of service. Migration speed refers to the time needed to transfer the ongoing services to the new server.

- **SH**: This parameter gives information about previous tasks accomplished by that server. SH is defined as the ratio of the successfully accomplished tasks to the overall number of tasks performed by the server.

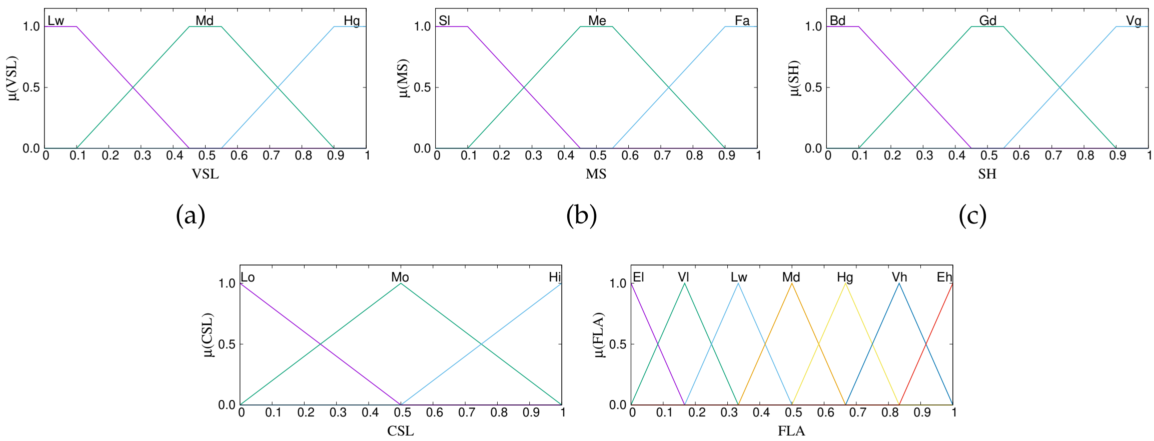
- **CSL**: Fog servers offer higher storage and computing capabilities compared to vehicles; however, they have their own limits. The server load indicates the number of processes waiting in queue to access the server. If the server is overloaded, it might affect its performance, and even make the server unresponsive.

- **FLA**: The output parameter values consist of values between 0 and 1, with the value 0.5 serving as a boundary to determine whether the fog layer is capable of handling the workload requested by the *vehicle*.



(a) FLS1 (b) FLS2

FIGURE 8.7: Proposed system structures of FLA.



(a) (b) (c)

(d) (e)

FIGURE 8.8: Membership functions of FLA. (**a**) Vehicle-to-Server Latency, (**b**) Migration Speed, (**c**) Server History, (**d**) Current Server Load, and (**e**) Fog Layer Adequacy.

TABLE 8.12: FLA System parameters and their term sets.

Models		**Parameters**	**Term Sets**
FLS2	FLS1	VSL	Low (Lw), Moderate (Md), High (Hg)
		SH	Bad (Bd), Good (Gd), Very Good (Vg)
		CSL	Low (Lo), Moderate (Mo), High (Hi)
		MS	Slow (Sl), Medium (Me), Fast (Fa)
FLS1 & FLS2		FLA	Extremely Low (El), Very Low (Vl), Low (Lw), Moderate (Md), High (Hg), Very High (Vh), Extremely High (Eh)

The input parameters are fuzzified using the membership functions shown in Fig. 8.8(a) to Fig. 8.8(d). In Fig. 8.8(e) are shown the membership functions used for the output parameter. We use triangular and trapezoidal membership functions because they are suitable for real-time operation. The term sets for each linguistic parameter are shown in Table 8.12. We decided the number of term sets by carrying

TABLE 8.13: FRB of FLS1.

No	VSL	SH	CSL	FLA	No	VSL	SH	CSL	FLA	No	VSL	SH	CSL	FLA
1	Lw	Bd	Lo	Hg	10	Md	Bd	Lo	Md	19	Hg	Bd	Lo	Lw
2	Lw	Bd	Mo	Md	11	Md	Bd	Mo	Lw	20	Hg	Bd	Mo	El
3	Lw	Bd	Hi	Vl	12	Md	Bd	Hi	El	21	Hg	Bd	Hi	El
4	Lw	Gd	Lo	Vh	13	Md	Gd	Lo	Hg	22	Hg	Gd	Lo	Md
5	Lw	Gd	Mo	Hg	14	Md	Gd	Mo	Md	23	Hg	Gd	Mo	Vl
6	Lw	Gd	Hi	Lw	15	Md	Gd	Hi	Vl	24	Hg	Gd	Hi	El
7	Lw	Vg	Lo	Eh	16	Md	Vg	Lo	Vh	25	Hg	Vg	Lo	Hg
8	Lw	Vg	Mo	Vh	17	Md	Vg	Mo	Hg	26	Hg	Vg	Mo	Lw
9	Lw	Vg	Hi	Md	18	Md	Vg	Hi	Lw	27	Hg	Vg	Hi	Vl

TABLE 8.14: FRB of FLS2.

No.	VSL	MS	SH	CSL	FLA	No.	VSL	MS	SH	CSL	FLA	No.	VSL	MS	SH	CSL	FLA
1	Lw	Sl	Bd	Lo	Hg	28	Md	Sl	Bd	Lo	Lw	55	Hg	Sl	Bd	Lo	El
2	Lw	Sl	Bd	Mo	Md	29	Md	Sl	Bd	Mo	Vl	56	Hg	Sl	Bd	Mo	El
3	Lw	Sl	Bd	Hi	Vl	30	Md	Sl	Bd	Hi	El	57	Hg	Sl	Bd	Hi	El
4	Lw	Sl	Gd	Lo	Vh	31	Md	Sl	Gd	Lo	Md	58	Hg	Sl	Gd	Lo	Vl
5	Lw	Sl	Gd	Mo	Hg	32	Md	Sl	Gd	Mo	Lw	59	Hg	Sl	Gd	Mo	El
6	Lw	Sl	Gd	Hi	Lw	33	Md	Sl	Gd	Hi	Vl	60	Hg	Sl	Gd	Hi	El
7	Lw	Sl	Vg	Lo	Eh	34	Md	Sl	Vg	Lo	Hg	61	Hg	Sl	Vg	Lo	Lw
8	Lw	Sl	Vg	Mo	Vh	35	Md	Sl	Vg	Mo	Md	62	Hg	Sl	Vg	Mo	Vl
9	Lw	Sl	Vg	Hi	Md	36	Md	Sl	Vg	Hi	Lw	63	Hg	Sl	Vg	Hi	El
10	Lw	Me	Bd	Lo	Vh	37	Md	Me	Bd	Lo	Md	64	Hg	Me	Bd	Lo	Vl
11	Lw	Me	Bd	Mo	Hg	38	Md	Me	Bd	Mo	Lw	65	Hg	Me	Bd	Mo	El
12	Lw	Me	Bd	Hi	Lw	39	Md	Me	Bd	Hi	Vl	66	Hg	Me	Bd	Hi	El
13	Lw	Me	Gd	Lo	Eh	40	Md	Me	Gd	Lo	Hg	67	Hg	Me	Gd	Lo	Lw
14	Lw	Me	Gd	Mo	Vh	41	Md	Me	Gd	Mo	Md	68	Hg	Me	Gd	Mo	El
15	Lw	Me	Gd	Hi	Md	42	Md	Me	Gd	Hi	Lw	69	Hg	Me	Gd	Hi	El
16	Lw	Me	Vg	Lo	Eh	43	Md	Me	Vg	Lo	Vh	70	Hg	Me	Vg	Lo	Md
17	Lw	Me	Vg	Mo	Eh	44	Md	Me	Vg	Mo	Hg	71	Hg	Me	Vg	Mo	Vl
18	Lw	Me	Vg	Hi	Hg	45	Md	Me	Vg	Hi	Md	72	Hg	Me	Vg	Hi	El
19	Lw	Fa	Bd	Lo	Vh	46	Md	Fa	Bd	Lo	Md	73	Hg	Fa	Bd	Lo	Vl
20	Lw	Fa	Bd	Mo	Hg	47	Md	Fa	Bd	Mo	Md	74	Hg	Fa	Bd	Mo	El
21	Lw	Fa	Bd	Hi	Md	48	Md	Fa	Bd	Hi	Lw	75	Hg	Fa	Bd	Hi	El
22	Lw	Fa	Gd	Lo	Eh	49	Md	Fa	Gd	Lo	Hg	76	Hg	Fa	Gd	Lo	Lw
23	Lw	Fa	Gd	Mo	Vh	50	Md	Fa	Gd	Mo	Hg	77	Hg	Fa	Gd	Mo	Vl
24	Lw	Fa	Gd	Hi	Hg	51	Md	Fa	Gd	Hi	Md	78	Hg	Fa	Gd	Hi	El
25	Lw	Fa	Vg	Lo	Eh	52	Md	Fa	Vg	Lo	Vh	79	Hg	Fa	Vg	Lo	Md
26	Lw	Fa	Vg	Mo	Eh	53	Md	Fa	Vg	Mo	Vh	80	Hg	Fa	Vg	Mo	Lw
27	Lw	Fa	Vg	Hi	Vh	54	Md	Fa	Vg	Hi	Hg	81	Hg	Fa	Vg	Hi	El

out many simulations. In Table 8.13 and Table 8.14, we show the FRBs of the proposed systems. FRB1 consists of 27 rules, whereas FRB2 of 81 rules. The control rules have the form: IF "conditions" THEN "control action". For instance, for FRB1, Rule 1: "IF VSL is Lw, SH is Bd and CSL is Lo, THEN FLA is Hg" or for FRB2, Rule 22: "IF VSL is Lw, MS is Fa, SH is Gd and CSL is Lo, THEN FLA is Eh".

8.7 IFS-CMR Testbed Design

In order to evaluate the simulation results of the aforementioned integrated system in Section 8.5, we implemented a testbed and carried out some experiments. IFS-CMR is comprised of two integrated subsystems (FS-ANVPC and FS-CFELS), each controlled by its respective FLC. Both subsystems have a key role in the system. The structure of IFS-CMR is shown in Figure 8.2. However for testbed experiment we considered only FS-ANVPC and FS-CFELS.

We designed and implemented a small-scale testbed using Raspberry Pis (RPi). We use five RPis that represent the vehicles moving for about 25 minutes in some urban area that expands to a number of apartment blocks. Of these five vehicles, one is *the vehicle* in need of resources whereas the other four are the vehicles that could turn into potential neighbors if they possess enough processing capability and the system decides to deploy the application in the edge layer. The size of area considered is 200m x 200m and the communication range of each vehicle is set to 50m. The movement of vehicles are generated using the *sumo* simulator and the layout is given in Figure 8.9.

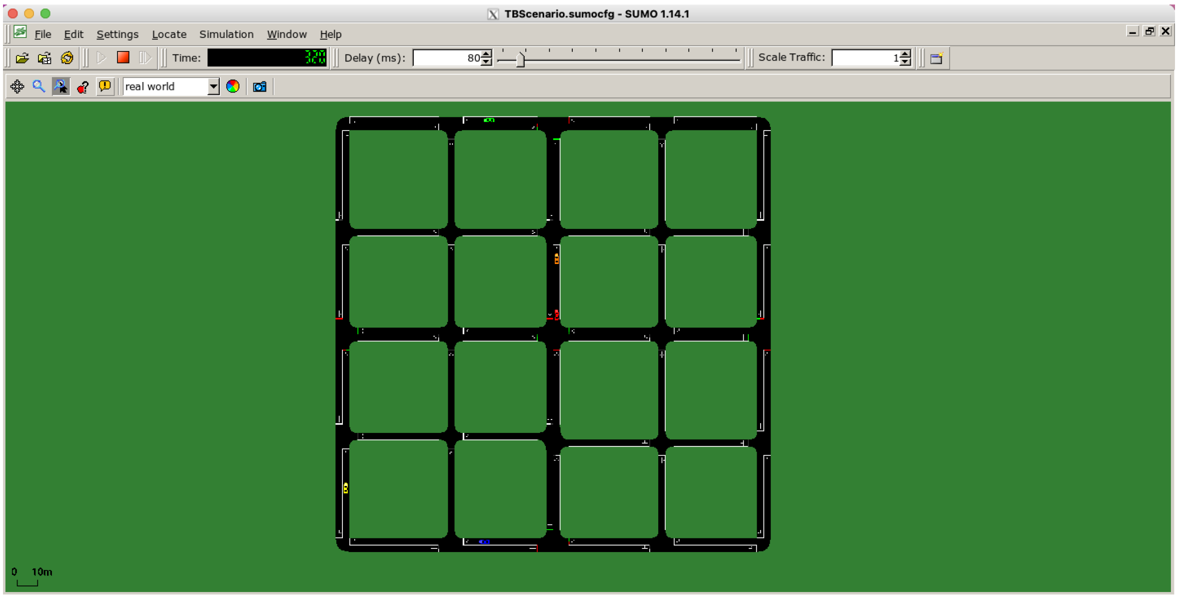


FIGURE 8.9: A screenshot of vehicles moving around the area.

The mobility trace is used to obtain the locations of all the vehicles at each time step of the experiment since the deployment of a large-scale testbed in a real environment (e.g., the RPi moving around in the neighborhood) was impossible due to various factors (i.e., increased costs, lack of human resources and time constraints). The setup of the testbed is summarized in Table 8.15.

The *vehicle* and its neighbors in the testbed communicate with one another as given in Figure 8.10. The *vehicle* broadcasts every second a help beacon containing

TABLE 8.15: Testbed setup.

Vehicles	5 RPi Model 3B+
Mobility trace generator	sumo
Area size	200m x 200m
Communication range	50m
TS	[0.1, 0.5, 0.9]
DC	[0.1, 0.5, 0.9]
Experimental time	1500 s

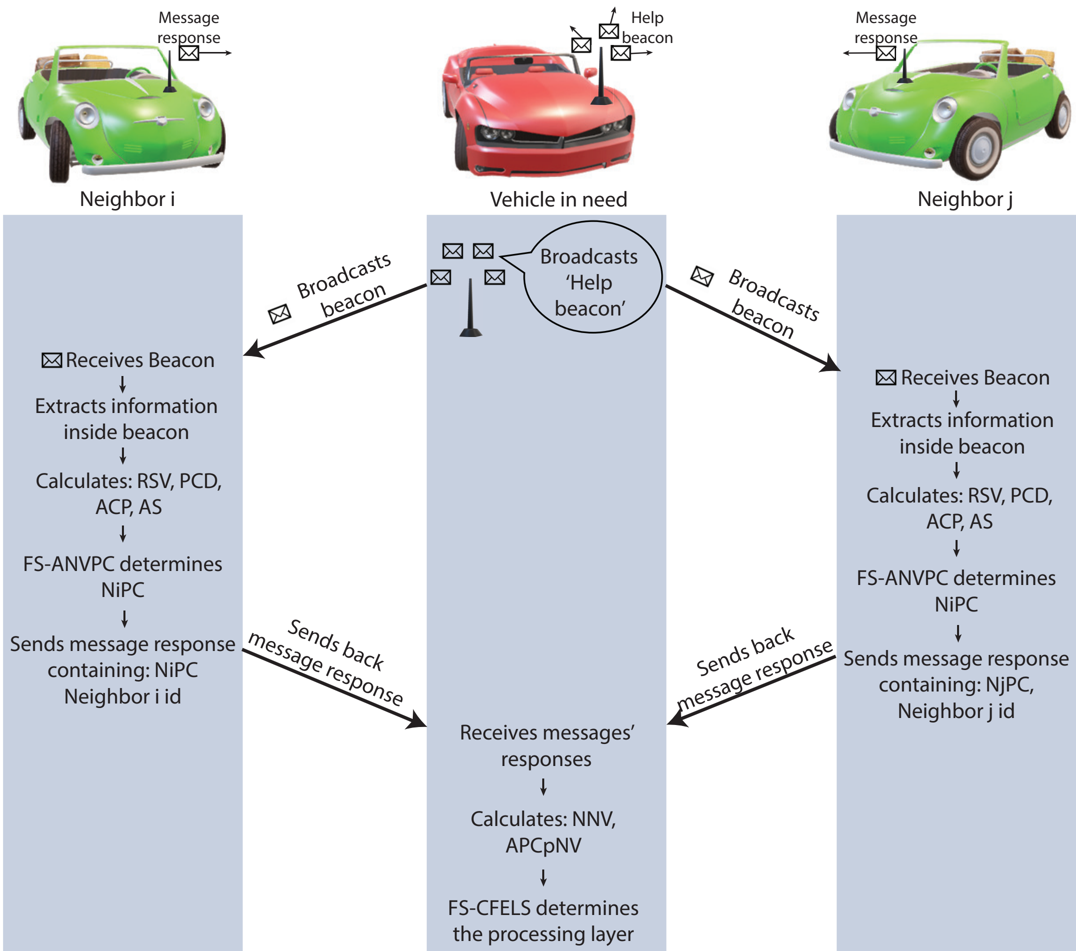


FIGURE 8.10: A scheme of the communication between the *vehicle* and its neighbors.

information about the vehicle id, speed, direction, timestep, and current and previous location. The vehicles within the communication range, also known as neighbors, receive the beacon and extract the information it contains so they can calculate the relative speed and the predicted contact duration. A RPi is considered a neighbor vehicle only if the distance, which is calculated using the coordinates obtained from the mobility trace, is shorter than their communication range. After calculating the predicted contact duration, each neighbor calculates its current available cpu and storage, which are the data they need to determine their processing capability. The processing capability is determined by running FS-ANVPC and the result is sent

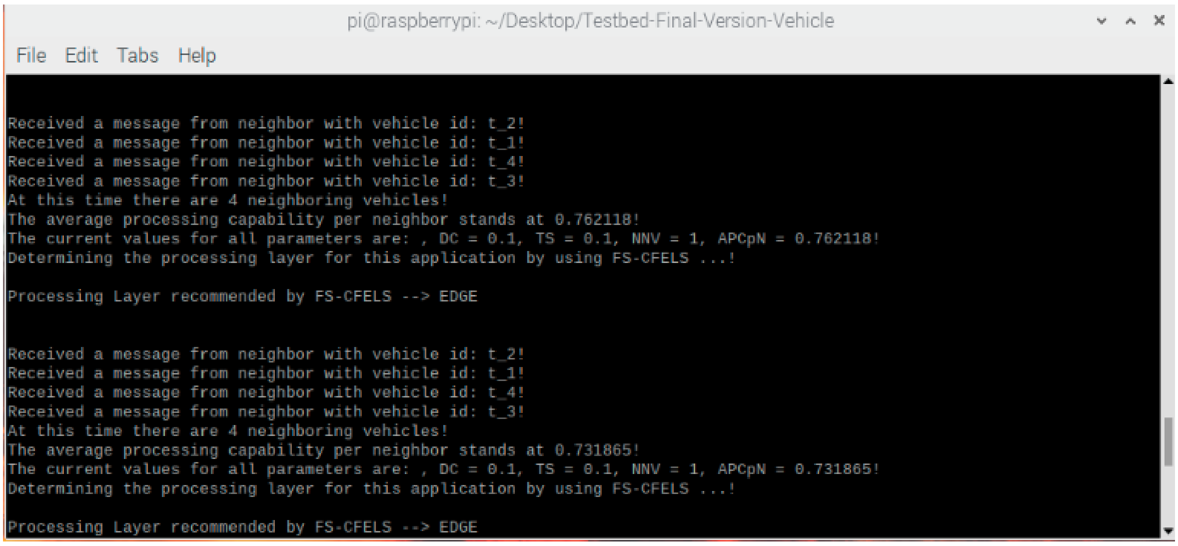


FIGURE 8.11: Snapshots from the command line of the *vehicle*.

back to the vehicle alongside its id in the form of a response message.

Based on the number of responses that the *vehicle* received at that timestep, it calculates the number of neighboring vehicles and the average processing capability per neighbor. It then runs FS-CFELS using these inputs and the application requirements (time sensitivity and complexity) and determines the recommended processing layer for that application. The application requirements are given as arguments to the program in order to investigate how IFS-CMR takes decisions for different applications under the same scenario[1]. One snapshots from the command line of the *vehicle* is shown in Figure 8.11.

[1]However not all parameters remain unchanged. While the number position of vehicles is unchanged, the same does not hold true for APC and AS. For the latter two parameters we obtain the available cpu and storage of the RPi at the moment the beacon arrives and this differs even from beacon to beacon, let alone when running the scenario again.

Chapter 9

Evaluation Results

This chapter presents the evaluation results of IFS-CMR, and it is organized into three sections, with each section presenting the results for the three subsystems of which IFS-CMR is composed. Section 9.1 presents and compares the results of the proposed FS-AQoS models, Section 9.2 presents and compares the simulation results of the FS-ANVPC models, whereas Section 9.3 shows the evaluation results of FS-CFELS and discusses the results of IFS-CMR as a whole. The results for each subsystem are separately organized to better present and understand the way IFS-CMR controls its final output, which is the selection of the resources to be used by vehicles that can best satisfy the considered requirements. Nevertheless, there is no distinctly separate discussion of results since the explanation is rather focused toward the overall purpose of system. The computer simulations for all of the proposed systems were conducted using FuzzyC[1]. Table 9.1 summarizes the simulation environment used for the simulations.

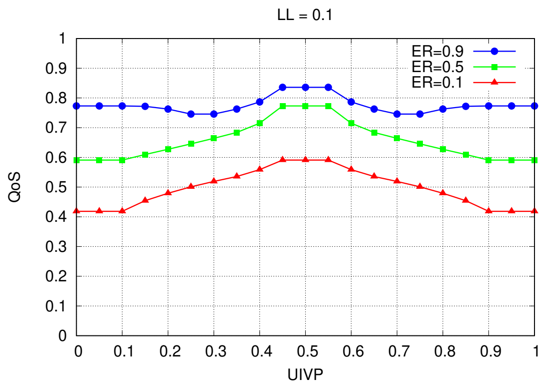
TABLE 9.1: Testing environment.

Simulator	FuzzyC
Hardware	ASRock Z77 Extreme6
OS	Ubuntu 20.04.4 LTS
CPU	Intel® Core™ i5-6500 CPU @ 3.20GHz × 4
Memory	16 GB

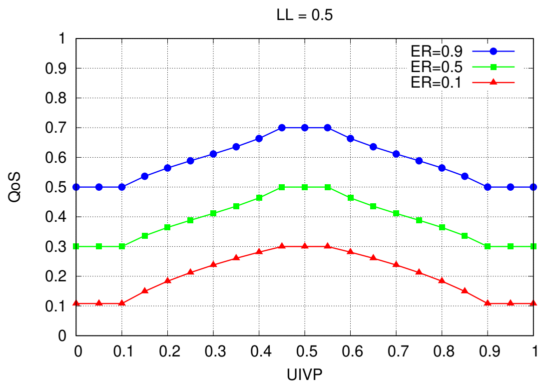
9.1 Results of FS-AQoS

In this section we discuss the simulation results for FSAQoS1 and FSAQoS2.

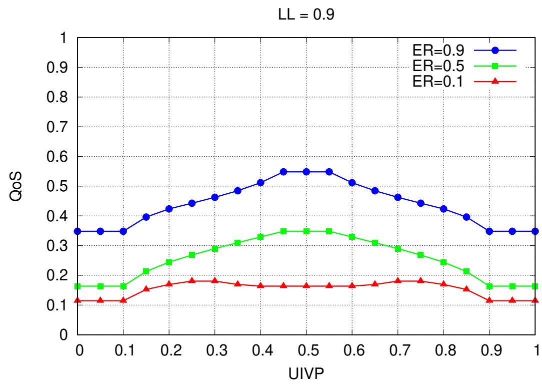
[1]More details for FuzzyC are given in Section 7.7



(a) LL = 0.1



(b) LL = 0.5



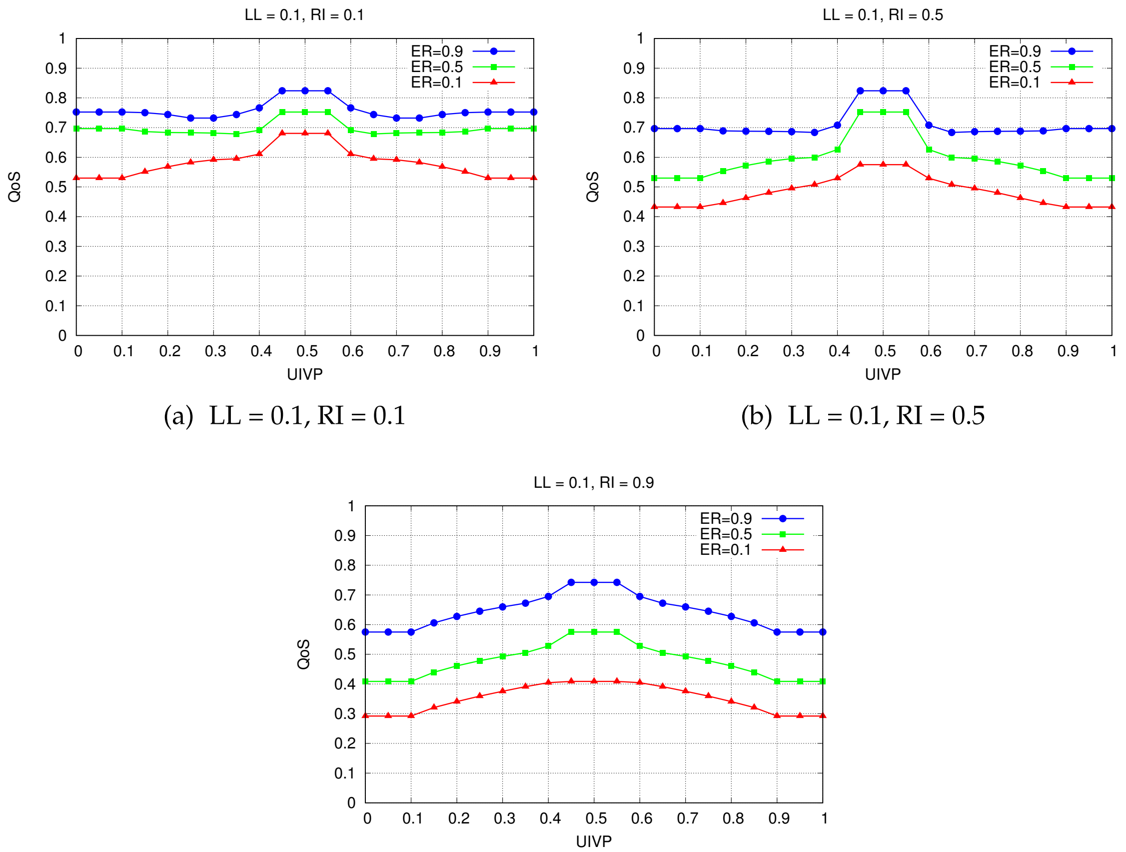
(c) LL = 0.5

FIGURE 9.1: Simulation results for FS-AQoS1.

9.1.1 Results of FS-AQoS1

The results for FSAQoS1 are shown for three scenarios: for low, medium and high link latency and are given in Figure 9.1. We show the relation between QoS and UIVP for different ER values while keeping LL values constant. The ER values 0.1, 0.5, 0.9 show a not effective, medium effective, effective link reliability, respectively.

In Figure 9.1(a), the link between two neighboring vehicles offers low communication latency. We see that most of V2V communication links provide very high QoS, especially for Ef and Mef reliability. QoS is still decided as acceptable even for links with ER=0.1. This is due to the fast responsiveness of communication links which is provided by the low latency. When the link latency increases, as shown in Figure 9.1(b), Nef communication links show a low QoS performance. We can see that only V2V links that have at least ER=0.5 and moderate UIVP can be considered as reliable. In Figure 9.1(c), we see that QoS values deteriorate when the link latency is high. A high latency means that more time is needed for the information to be transmitted between two vehicles. Only the links that offer high reliability and have not redundant beacon messages are considered as capable of offering high performance.



(a) LL = 0.1, RI = 0.1

(b) LL = 0.1, RI = 0.5

(c) LL = 0.1, RI = 0.9

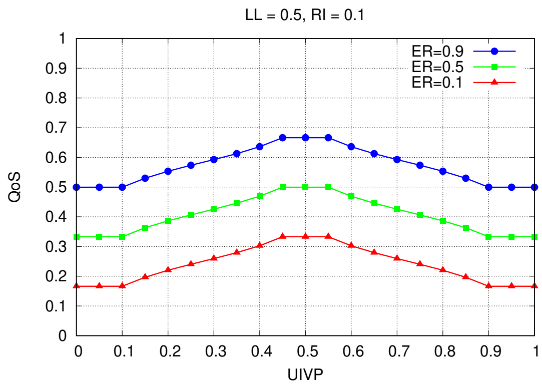
FIGURE 9.2: Simulation results for FS-AQoS2 for low link latency.

9.1.2 Results of FS-AQoS2

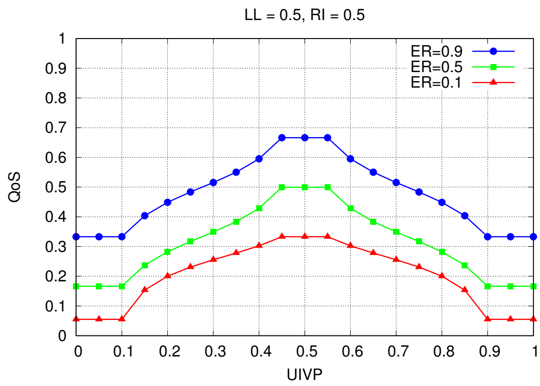
The simulation results for FS-AQoS2 are presented in Figures 9.2–9.4. We see the effect of RI on the decision for selecting the appropriate V2V links in order to accomplish a successful communication. The results are shown for low, medium and high link latency. Each of them shows the results for different cases of RI. The RI values 0.1, 0.5 and 0.9 show permissible, acceptable and harmful interference, respectively.

Figure 9.2(a) shows the scenario in which a low communication latency is supported for data transmission and the interference caused by transmissions of other vehicles is permissible. Due to this, we see that all of V2V communication links provide high QoS, despite the reliability values. Even though the reliability of the successfully transmitted packets is very low, given the low latency of the network, the vehicles can retransmit the data packets again during their contact duration, until a successful delivery is achieved. However, harmful interference will degrade the performance of links, as seen in Figure 9.2(b). In the case of high RI and not effective ER (see Figure 9.2(c)), the communication will lead to loss of information, which is indicated by the low value of QoS.

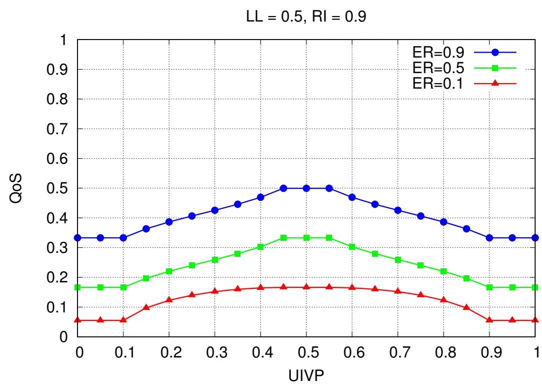
With the increase of LL, the link performance will deteriorate even more since



(a) LL = 0.5, RI = 0.1



(b) LL = 0.5, RI = 0.5

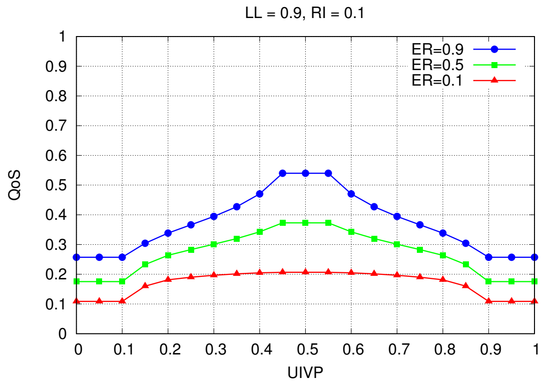


(c) LL = 0.5, RI = 0.9

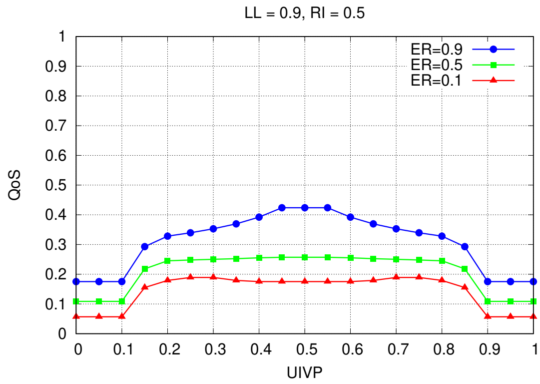
FIGURE 9.3: Simulation results for FS-AQoS2.

it takes more time for the data to be transmitted between two vehicles, as seen in Figure 6. If there is no or little interference (see Figure 9.3(a)), the QoS is always decided as acceptable for ER=0.9. The QoS is decided as acceptable also for ER=0.5, but for this case, the UIVP must be moderate. With the increase of interference, the QoS of the communication links is deteriorated. When the interference is within the permissible level (see Figure 9.3(b)), acceptable QoS values are achieved even for ER = 0.5, but this is not possible for worse interferences. For harmful interferences (see Figure 9.3(c)), the only communication scenario considered successful is when ER=0.9 and UIVP = 0.5. This is due to the fact that higher latencies and interferences lead to data exchanges that require more time for transmission and failure to reach the destination.

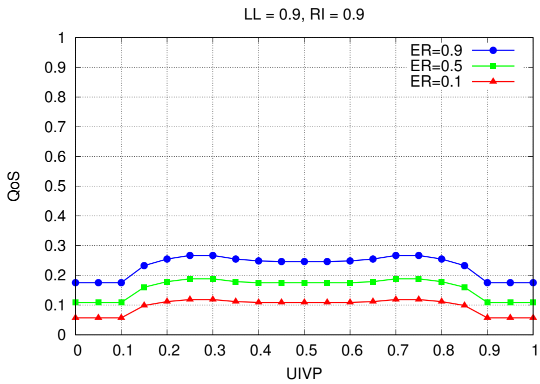
The results for high LL are shown in Figure 9.4. In such a condition, the realization of a successful communication becomes hard as a long period of time is needed to transmit messages in a dynamic environment. This can be seen even for a permissible interference. As shown in Figure 9.4(a), only the links with high ER and few UIVP are considered as capable of offering a good performance. With the increase of interference (see Figure 9.4(c)), even a reliable link cannot achieve a good communication.



(a) LL = 0.9, RI = 0.1



(b) LL = 0.9, RI = 0.5



(c) LL = 0.9, RI = 0.9

FIGURE 9.4: Simulation results for FS-AQoS2.

Having a considerable interference over a long link latency will cause attenuation of the wireless signals, and given the long duration of transmission, it increases the chance of a loss of communication. Moreover, when a harmful RI is present (see Figure 9.4(c)), none of links is capable to hold a successful communication.

9.1.3 Comparison of FS-AQoS1 and FS-AQoS2

Throughout the scenarios, we see that both systems, FSAQoS1 and FSAQoS2, reach peak values when the UIVP values are moderate. The performance of the link communication does not increase when too many UIVP packets are transmitted because more packets occupy more bandwidth in the network.

Comparing both models, FSAQoS2 gives more realistic results because it takes into consideration factors that affect the network performance such as path attenuation, obstacles, transmission distance and so on. On the contrary, FSAQoS1 gives higher QoS values because it does not consider RI effect, and therefore it might not accurately estimate the performance of the link. For example, FSAQoS1 for LL = 0.9 (Figure 9.1(c)) shows similar results with FSAQoS2 when LL = 0.9 and RI = 0.1 (Figure

9.4(a)). Nevertheless, the results change significantly for FSAQoS2 for RI = 0.5 and RI = 0.9. This is because FSAQoS1 assumes an ideal environment where communications among vehicles will always happen without interference, which is not realistic, especially for such a dynamic network like VANETs where vehicles broadcast messages concurrently all the time. V2V links carry mostly safety-critical messages, thus providing low latency, minimal interference and no packet losses becomes essential in order to guarantee a high performance of the critical applications.

9.2 Results of FS-ANVPC

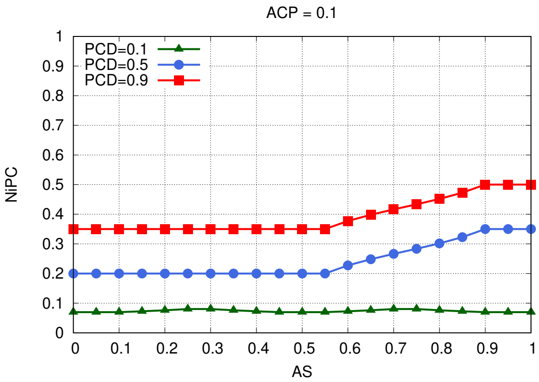
The results for FS-ANVPC1 and FS-ANVPC2 are shown in Figure 9.5 and Figures 9.6–9.6, respectively. While we have conducted many simulations for both models of FS-ANVPC, the scenarios we present in this thesis are the ones that show the relation between NiPC and AS for different PCD values. The effect of ACP for FS-ANVPC1 and ACP and QoS for FS-ANVPC2 on NiPC can be seen when a pair of subfigures are selected for comparison, provided that one of the parameters has the same value.

9.2.1 Results of FS-ANVPC1

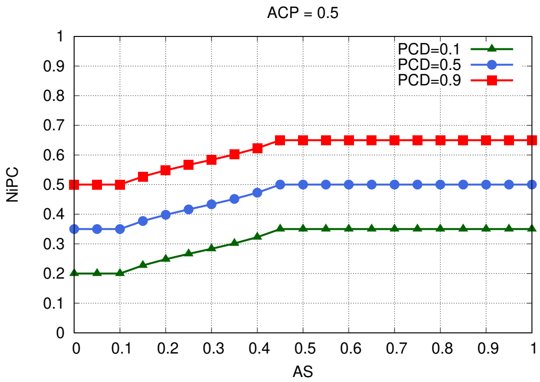
For FS-ANVPC1 we consider three scenarios, each with different ACP values: small, medium and large amount of ACP. The results are presented in Figure 9.5. In Figure 9.5(a), we consider a small amount of available computing power offered by the neighboring vehicle to help others in need for additional resources. We can see that when a long period of communication is anticipated between vehicles, a big amount of storage is needed in order to perform the required tasks. It is worth taking these vehicles in consideration, as an increase in ACP might happen, given the long period of communication. It is possible that after some time, this neighbor will complete processing its own tasks and offer more computing resources to the other vehicles in need, thus making this neighbor a prospective helper. This effect is shown in Figure 9.5(b) in which a neighbor with long PCD is capable to help, despite its available storage resources. Also, vehicles with PCD = 0.5 and a moderate amount of AS are considered as helpful. On the other hand, if a neighbor is willing to lend a large amount of computing power (see Figure 9.5(c)), even vehicles with a short PCD are considered as helpful when a minimal amount of AS is offered.

9.2.2 Results of FS-ANVPC2

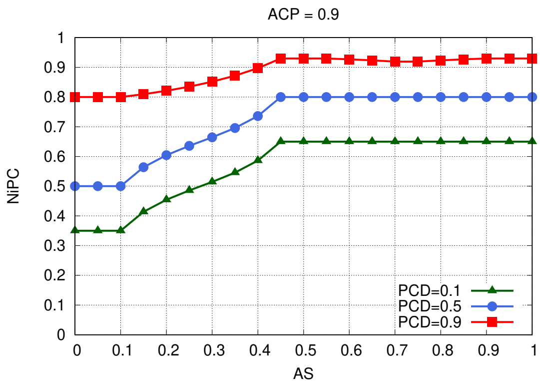
For FS-ANVPC2 we consider nine scenarios, which result from the combination of different ACP and QoS values. The results are presented in Figures 9.6–9.8. We see



(a) ACP = 0.1



(b) ACP = 0.5



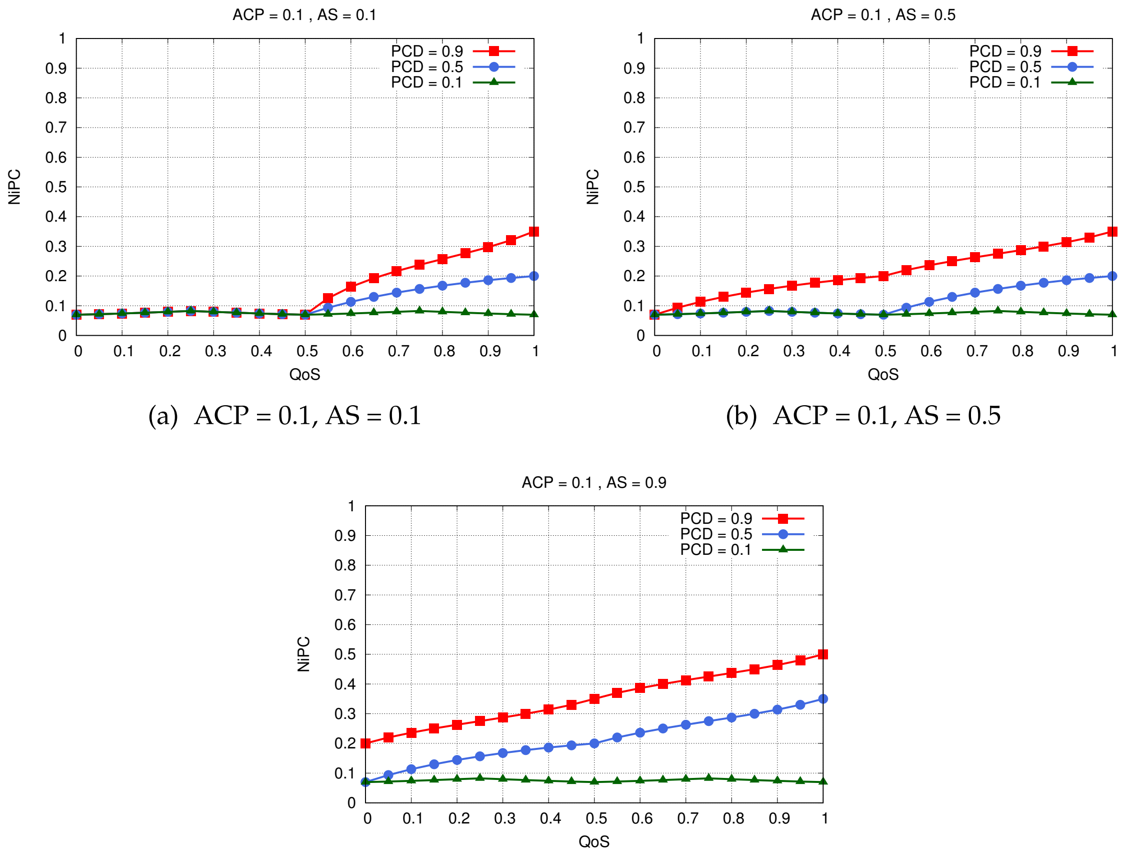
(c) ACP = 0.9

FIGURE 9.5: Simulation results for FS-ANVPC1.

the effect of QoS on the decision for selecting the appropriate neighboring vehicles to accomplish the needed applications.

We see in Figure 9.6(a) and Figure 9.6(b) that due to the low processing power and poor QoS provided, none of the vehicles is considered as helpful, even when a long predicted contact duration between vehicles is anticipated. However, for high QoS (see Figure 9.6(c)) we see that vehicles with long predicted contact duration and the highest value of AS have the potential to be helpful, even though they offer a small value of processing power. A high QoS value of the communication link between the vehicle and the neighboring vehicle indicates that the application requirements will be met, thus it significantly increases the chance for the neighbor to successfully accomplish the required tasks.

Figures 9.7(a)–9.7(c) show vehicles that are willing to lend a moderate amount of processing resources. As we can see, many vehicles can be considered helpful compared to the scenarios with small ACP. In case of a moderate QoS (see Figure 9.7(b)) we see that the vehicles with long predicted contact duration and moderate AS are capable to help the vehicle in need. Whereas, for a high QoS, the vehicles that are able to hold the communication session for a long time are considered as helpful,



(a) ACP = 0.1, AS = 0.1

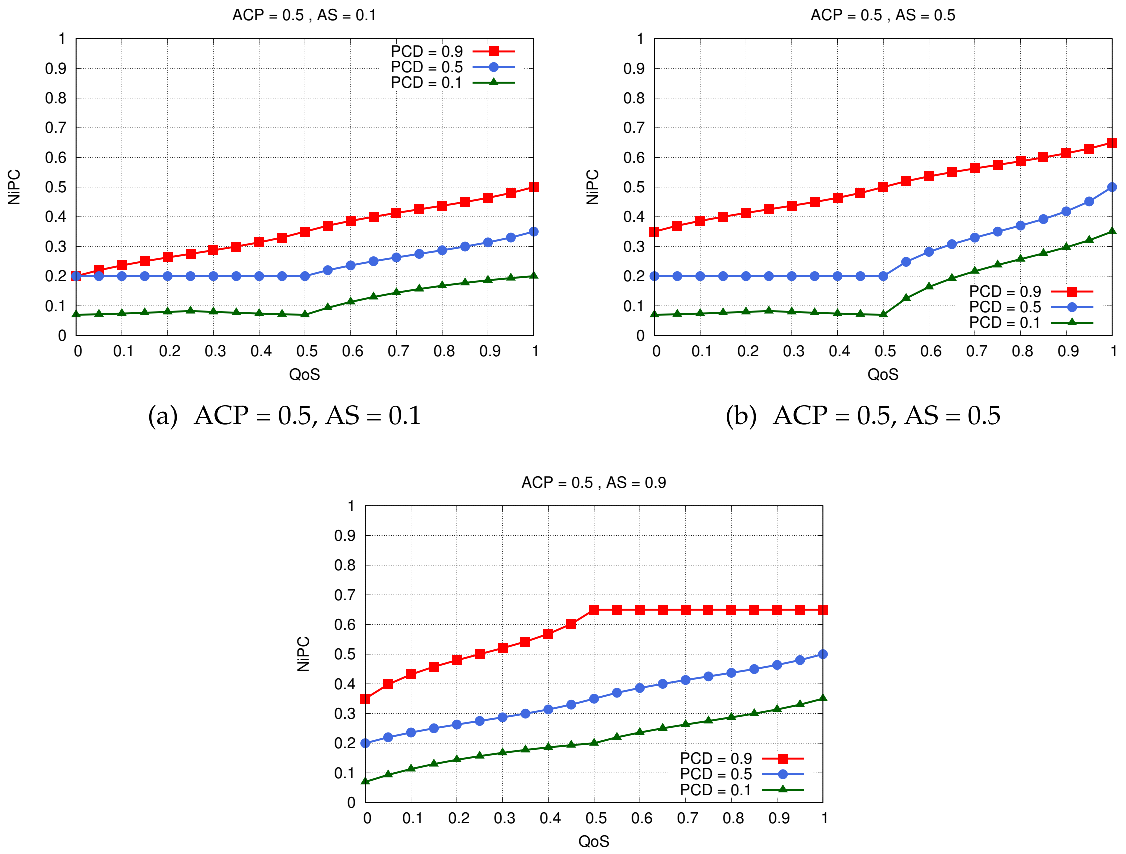
(b) ACP = 0.1, AS = 0.5

(c) ACP = 0.1, AS = 0.9

FIGURE 9.6: Simulation results for FS-ANVPC2.

given that these vehicles possess a storage with a minimum value of 17.5%. This happens because the vehicle in need can distribute the data processing to multiple neighbors, which will process the data, accomplish the tasks and send the output to the vehicle. For this to happen, a high QoS is needed to establish reliable connections so that the parallel processing data can accomplish the task successfully.

When vehicles offer large processing power but a low QoS is provided (see Figure 9.8(a)), we can see that the neighbor vehicles that stay inside the vehicle's communication range for a long period of time and offer a moderate value of AS can be considered as helpful. Although the communication link between vehicles provides a low QoS value, it is worth considering these vehicles as potential helpers as they are anticipated to stay inside the communication range of each-other for a long duration. The dynamic nature in VANET can cause temporary interference which degrade the quality of link, but it is likely that after some time this obstacle will be diminished and given the long PCD and moderate amount of storage offered, these neighbors can store the processed data in their storage and transmit them when an improved QoS is provided. With the increase of QoS (see Figure 9.8(b)) we see that even though the processing capability of the vehicles has improved



(a) ACP = 0.5, AS = 0.1

(b) ACP = 0.5, AS = 0.5

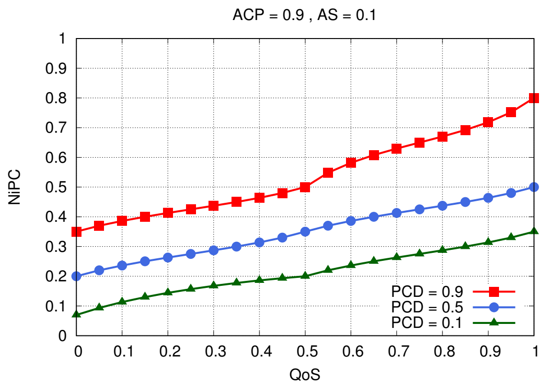
(c) ACP = 0.5, AS = 0.9

FIGURE 9.7: Simulation results for FS-ANVPC2.

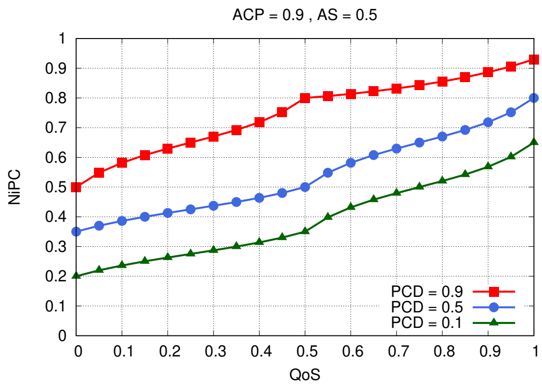
significantly, a neighbor is considered as helpful, regardless the offered AS, only when the communication session is expected to last for a while. Whereas, when a high QoS is provided, we see from Figure 9.8(c) that even the vehicles with a short PCD can be considered as helpful. Given the high QoS and the large ACP, the data will be transmitted to the neighbor at a high rate and it will be processed faster, thus this neighbor is able to accomplish the required tasks successfully and transmit the output in real-time, within a short period of time. A minimum amount of storage is needed to perform the tasks, as the neighbor losses and jitter will always happen among vehicles, which is not realistic. V2V links carry mostly safety-critical messages, thus providing low latency, high bandwidth and no packet losses becomes essential in order to guarantee a high performance of the critical applications.an transmit the output stepwise, after each single sub-task has finished processing.

9.2.3 Comparison of FS-ANVPC1 and FS-ANVPC2

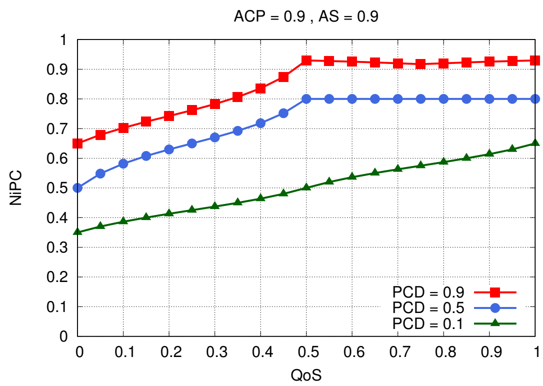
Comparing the models, FS-ANVPC2 gives more realistic results because it takes into consideration factors that affect the network performance such as path attenuation



(a) ACP = 0.9, AS = 0.1



(b) ACP = 0.9, AS = 0.5



(c) ACP = 0.9, AS = 0.9

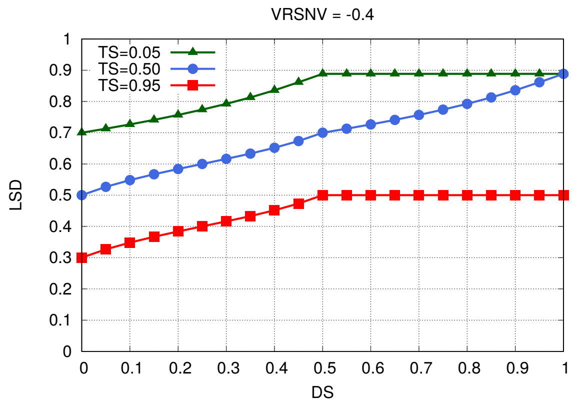
FIGURE 9.8: Simulation results for FS-ANVPC2.

and application requirements. On the contrary, FS-ANVPC1 gives higher NiPC values because it does not consider QoS requirements, and therefore it might not accurately estimate the processing capability. For example, for FS-ANVPC2 when ACP=0.5 and QoS=0.1 (see Figure 9.7(a)), we see that a neighbor with a long PCD is not consider helpful, while for FS-ANVPC1 we see that the same neighbors are helpful. This is because FS-ANVPC1 assumes an ideal environment where a communication with no packet losses and jitter will always happen among vehicles, which is not realistic. V2V links carry mostly safety-critical messages, thus providing low latency, high bandwidth and no packet losses becomes essential in order to guarantee a high performance of the critical applications.

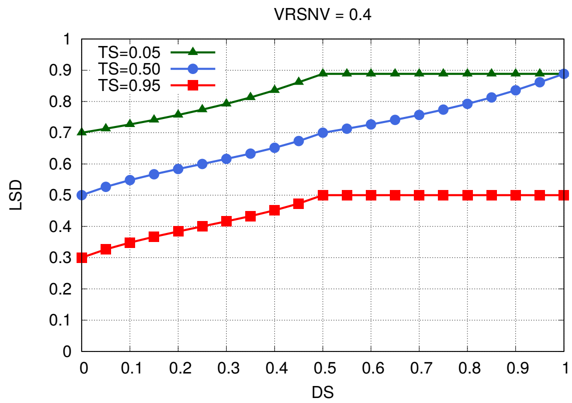
9.3 Results of FS-CFELS

9.3.1 Results of FS-CFELS1

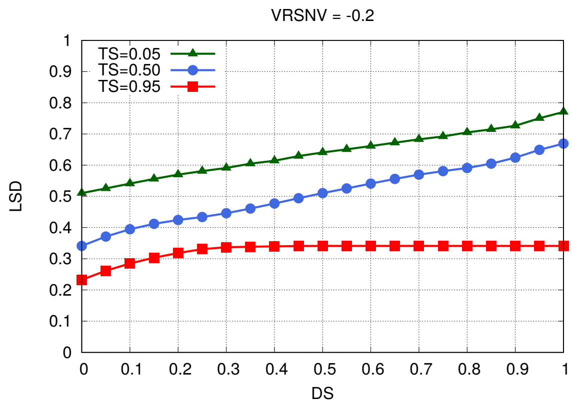
The simulation results for FSRM1 are presented in Figure 9.9. We show the relation between LSD and DS for different TS values. The TS values considered for the



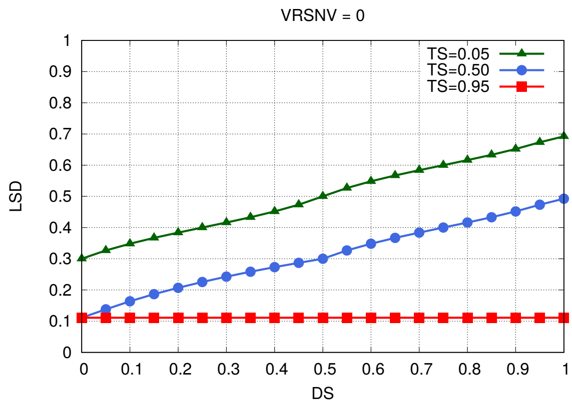
(a) VRSNV = -0.4



(b) VRSNV = 0.4



(c) VRSNV = -0.2



(d) VRSNV = 0

FIGURE 9.9: Simulation results for FSRM1.

simulations are 0.05, 0.5 and 0.95. The values 0.05 and 0.95 simulate the delay tolerant and highly time-sensitive applications, respectively. The TS=0.5 is included for the applications which are neither delay tolerant nor time-sensitive. The standard set of truth values for all fuzzy variables is [0, 1], however we have shifted these truth values for the VRSNV parameter to [-0.5, 0.5]. This change is made to show that the VRSNV is a symmetrical parameter, as the relative speed is calculated as a difference between the vehicle and the neighbors moving in the same direction. Therefore, two opposite values, e.g., -0.2 and 0.2 represent the vehicle moving slower and faster at the same rate, respectively. The simulation results for FSRM1 are presented in Figure 9.9. We show the relation between LSD and DS for different TS values.

The TS values considered for the simulations are 0.05, 0.5 and 0.95. The values 0.05 and 0.95 simulate the delay tolerant and highly time-sensitive applications, respectively. The TS=0.5 is included for the applications which are neither delay tolerant nor time-sensitive. The standard set of truth values for all fuzzy variables is [0, 1], however we have shifted these truth values for the VRSNV parameter to [-0.5, 0.5]. This change is made to show that the VRSNV is a symmetrical parameter, as the relative speed is calculated as a difference between the vehicle and the neighbors moving in the same direction. Therefore, two opposite values, e.g., -0.2 and 0.2 represent the vehicle moving slower and faster at the same rate, respectively. In the results presented in this section we use fixed values for this parameter in order to better explain its relation with LSD and to compare the results for opposite values. The considered values are -0.4, -0.2, 0 , 0.2 and 0.4. better explain its relation with LSD and to compare the results for opposite values. The considered values are -0.4, -0.2, 0 , 0.2 and 0.4.

In Figure 9.9(a) are shown the simulation results for VRSNV=-0.4. The vehicle is moving much slower than the other neighboring vehicles; therefore, it is hard to establish and maintain a connection between the vehicle and its neighbors, and from applications which need real-time computing, only the ones with the smallest data can be processed at the edge. If the amount of data is bigger, these time-sensitive applications can be processed/run at the fog servers which offer low latency anyway. If the applications are delay tolerant, we can see that these applications will be sent at the cloud servers.

From Figure 9.9(b) we can see that same results are achieved for VRSNV=0.4. The relative speed between the vehicle and its neighbors is translated into a distance between them. This distance grows equally for the vehicle moving slower or faster at a same rate, thus the contact duration is same for both considered scenarios.

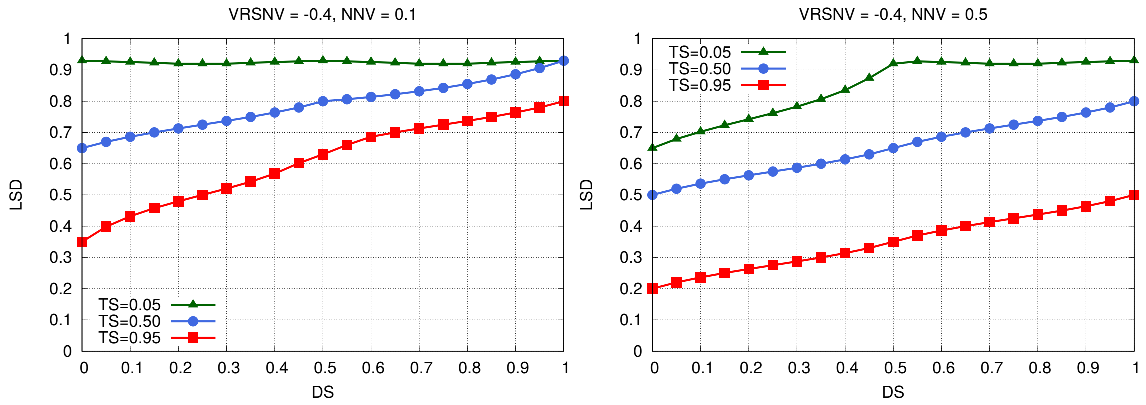
In Figure 9.9(c) we consider the vehicle moving slightly slower than its neighbors. The decrease of the relative speed between the vehicle and its neighboring vehicles increases the contact duration between them, consequently more highly time-sensitive data can be processed at the edge.

The scenario where the vehicle moves at the same speed as its neighbors is considered in Figure 9.9(d). We can see that all highly time-sensitive applications will run at the edge, even when the size of data is big. The vehicle will have the same adjacent vehicles for a while as they move with the same speed. Being in the communication range of each other for a time, creates the possibility of initiating virtual machines to these adjacent vehicles which can be used to run also a number of delay tolerant applications. Processing these data at the edge instead of cloud servers can offload a big amount of traffic flow from the core networks.

9.3.2 Results of FS-CFELS2

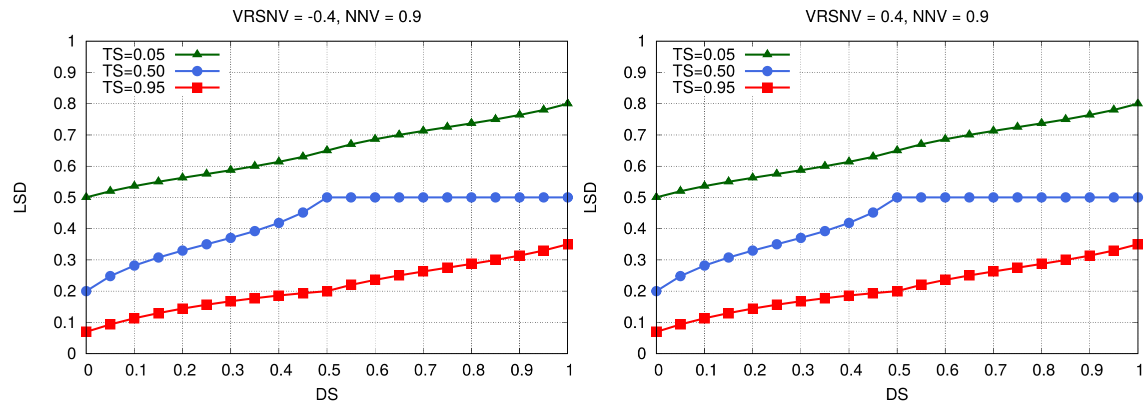
The simulation results for FSRM2 are shown in Figure 9.10, Figure 9.11, Figure 9.12 and Figure 9.13. We see the effect of NNV on the decision of LSD.

In Figure 9.10 we consider the VRSNV value -0.4 and change the NNV from 0.1 to 0.9. The results for NNV=0.1 are shown in Figure 9.10(a). Having only a few vehicles within the communication range and moving much slower than them, makes it



(a) NNV = 0.1 (b) NNV = 0.5

FIGURE 9.10: Simulation results for FSRM2. VRSNV = -0.4.



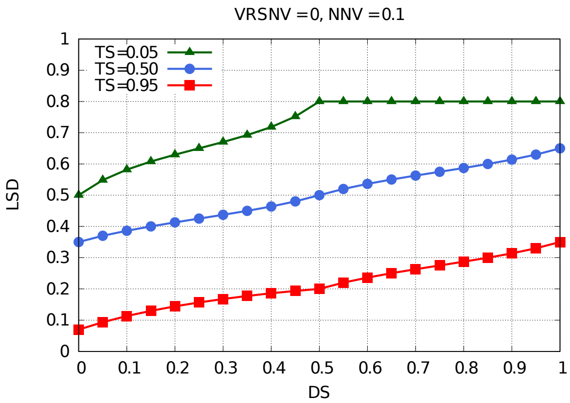
(a) VRSNV = -0.4 (b) VRSNV = 0.4

FIGURE 9.11: Simulation results for FSRM2. NNV = 0.9.

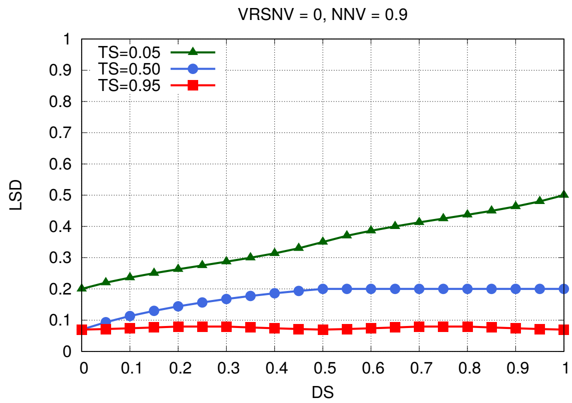
almost impossible for the vehicle to process applications through its neighbors. For this reason, only the smallest time-sensitive applications are processed at the edge. If the number of the neighbors is increased (see Figure 9.10(b)) the size of time-sensitive data that can be processed in the edge layer is also increased. This is due to the fact that the vehicle can distribute the data processing between many neighbors.

In Figure 9.11 we consider NNV=0.9 and show the results for both VRSNV=-0.4 and VRSNV=0.4. The results are exactly the same and we included the reason while demonstrating the results of FSRM1. When comparing with FSRM1, we can see that contrary to the FSRM1, FSRM2 decides that not only the smallest time-sensitive data will be processed at the edge but also other time-sensitive data no matter what their size is.

Figure 9.12 shows the results for VRSNV=0, with Figure 9.12(a) and Figure 9.12(b) showing the results for low and high number of neighboring vehicles, respectively. Moving at the same speed as its neighbors makes it possible for the vehicle to process various applications at the edge. Furthermore, if many vehicles are within

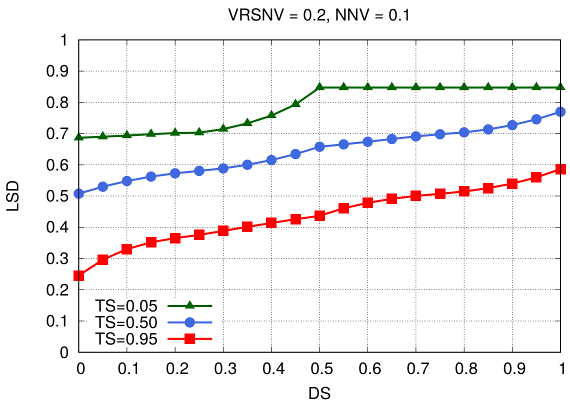


(a) NNV = 0.1

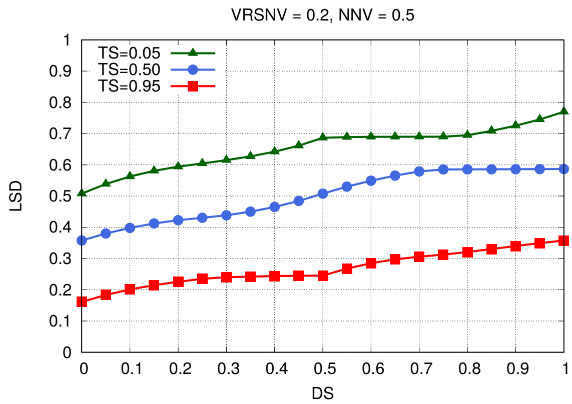


(b) NNV = 0.9

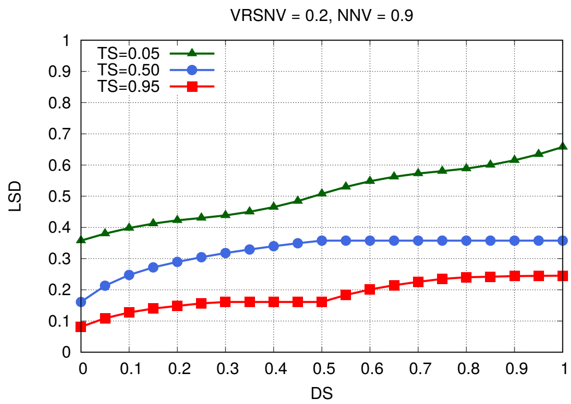
FIGURE 9.12: Simulation results for FSRM2. VRSNV = 0.



(a) NNV = 0.1



(b) NNV = 0.5

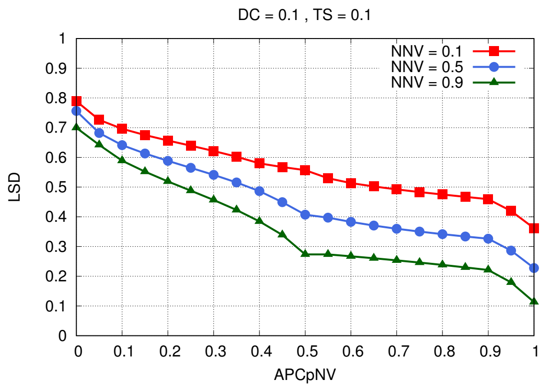


(c) NNV = 0.9

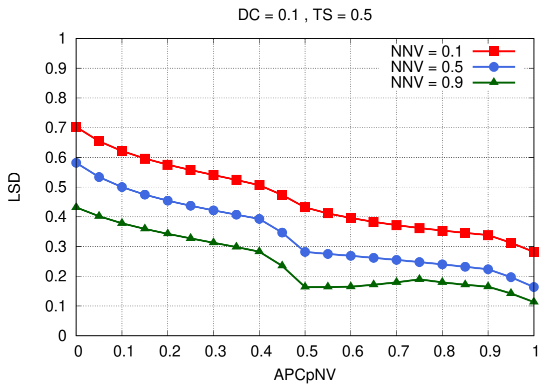
FIGURE 9.13: Simulation results for FSRM2. VRSNV = 0.2.

the communication range of the vehicle which needs additional resources (Figure 9.12(b)), we can see the possibility to process even big delay tolerant applications at the edge is increased. This results in removing the burden of having an excessive traffic flow on the core networks.

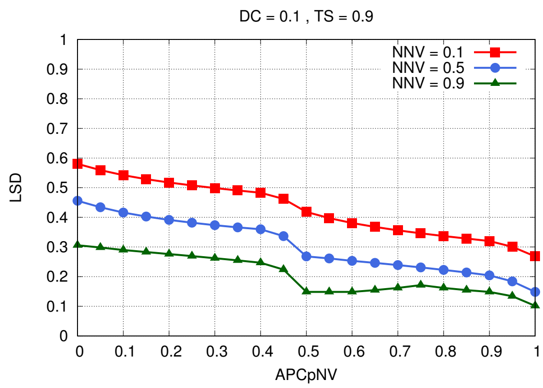
In Figure 9.13 we consider VSRNV=0.2 in which we simulate the scenario where



(a) DC = 0.1, TS = 0.1



(b) DC = 0.1, TS = 0.5



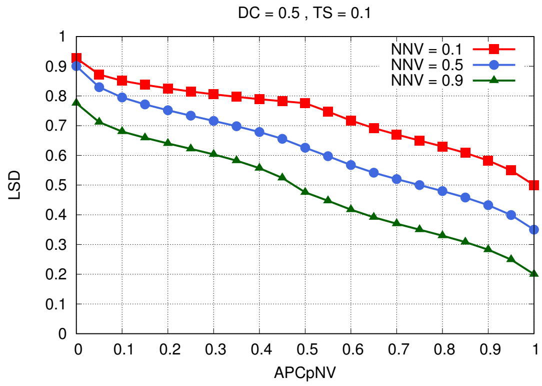
(c) DC = 0.1, TS = 0.9

FIGURE 9.14: Simulation results for FS-CFELS.

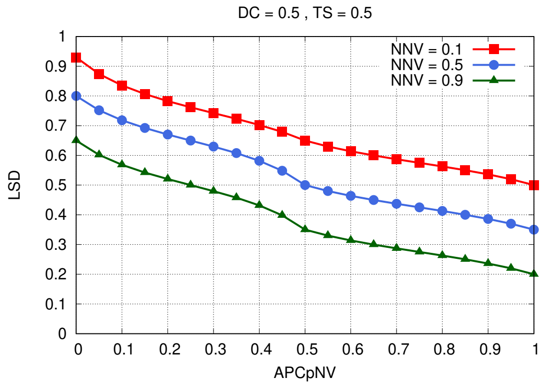
the vehicle is moving faster than the other vehicles in its vicinity. We change the value of NNV from 0.1 to 0.9. In Figure 9.13(a), we show the results for NNV = 0.1. We can see that only a few real-time data can be processed at the edge, based on their size. The vehicle will move out of the communication range of its neighbors in a short time and data distribution is not possible when only a few other vehicle are present. When the number of neighboring vehicles is increased, not only the amount of time-sensitive application data which can be processed at the edge is increased, but also the amount of non real-time data.

9.3.3 Results of FS-CFELS3

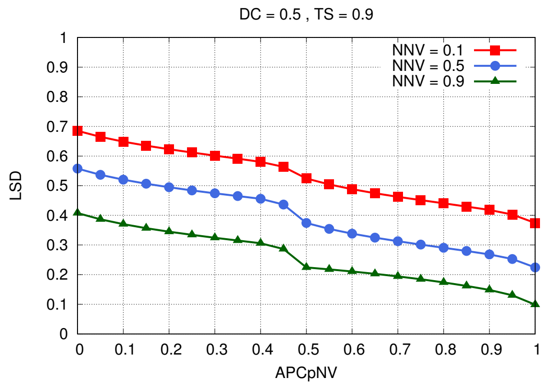
The simulation results of FS-CFELS are given in Figures 9.14 – 9.16. The parameters considered constant for presenting the results are DC and TS as these parameters represent the application requirements, and as such, they differ only from application to application. Therefore, each subfigure represents practically a different set of applications that have similar requirements. Using this configuration we can see how



(a) DC = 0.5, TS = 0.1



(b) DC = 0.5, TS = 0.5



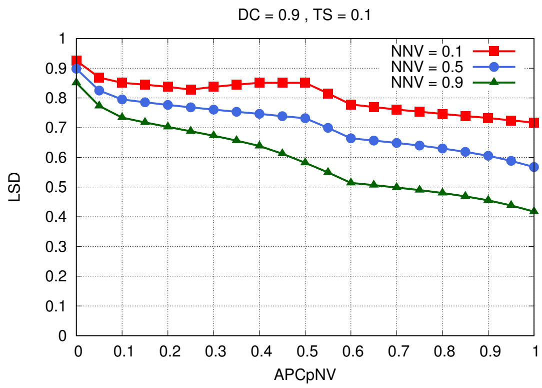
(c) DC = 0.5, TS = 0.9

FIGURE 9.15: Simulation results for FS-CFELS.

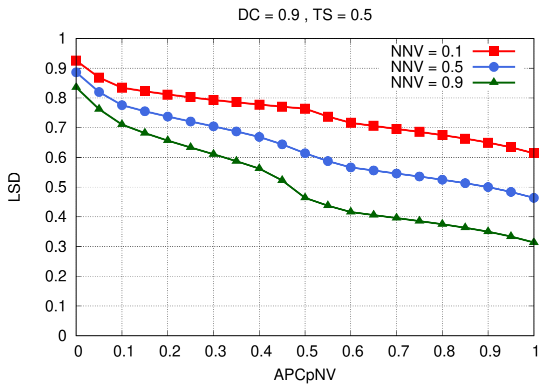
LSD relates with the changing characteristics of the edge layer, which are represented by NNV and APCpNV.

The simulation results considering a set of non-complex applications that are delay tolerant are presented in Figure 9.14(a). The results show that when *the vehicle* is surrounded by many potentially helpful neighbors, the system selects the edge layer as the most appropriate layer for *the vehicle* to run its applications. Although these applications do not require real-time processing, running them in the edge layer has two benefits: it exploits the high capacity of neighbors which at this point is being unused and it avoids unnecessary traffic being sent in the core network. On the other hand, in Figure 9.14(b) and Figure 9.14(c), we can see that the edge layer is hardly selected when the data complexity increases. Most of the data are sent in the fog or cloud layer, with the latter being used significantly more as the complexity increases. Using the cloud layer instead of fog for complex data frees the fog servers from unnecessary overload, considering the fact that these applications do not need to run in real-time.

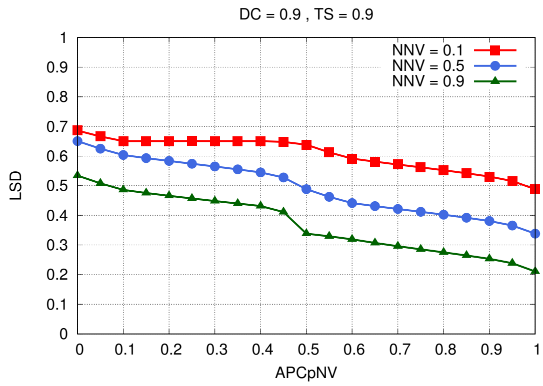
However, as we can see from the results shown in Figure 9.15, the system decides that the time-sensitive applications will be processed only in the edge and fog layer



(a) DC = 0.9, TS = 0.1



(b) DC = 0.9, TS = 0.5



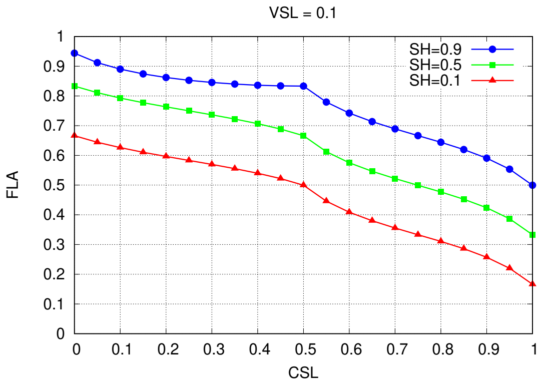
(c) DC = 0.9, TS = 0.9

FIGURE 9.16: Simulation results for FS-CFELS.

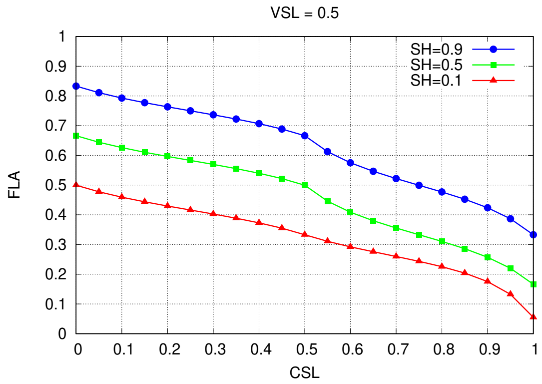
and never in the cloud. This decision fulfills such a strong requirement like latency. When the applications are not too complex, *the vehicle* is suggested to use mostly the resources of the neighboring vehicles, provided that there is a considerable number of them in its vicinity. When the number of neighboring vehicles is not very high or they are not prospective helpers, the system suggests the fog layer as the appropriate layer, especially for complex data. Fog servers have more powerful computing capabilities, and since they offer low latency as well, they can handle these data better while still ensuring real-time processing.

9.4 Results of FLA

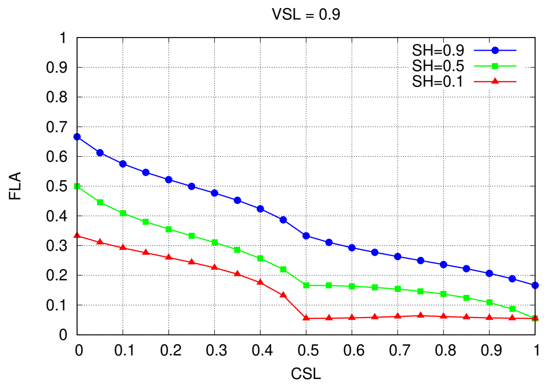
In this section we discuss the simulation results for FLS1 and FLS2. The simulations were conducted using FuzzyC and the results are shown for three scenarios for each model.



(a) VSL = 0.1



(b) VSL = 0.5

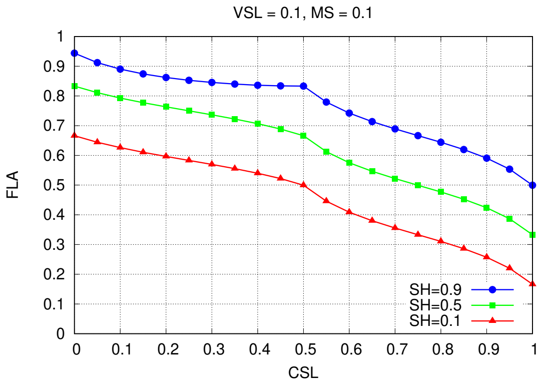


(c) VSL = 0.9

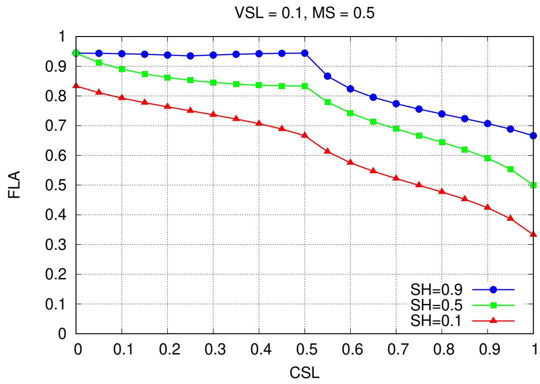
FIGURE 9.17: Simulation results for FLS1.

9.4.1 Results of FLS1

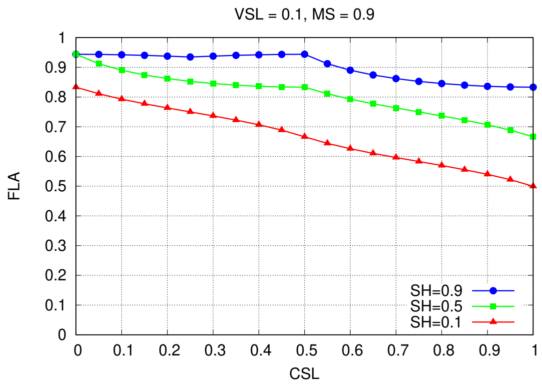
Fig. 9.17 shows the result of FLS1 for low, moderate and high vehicle-to-sever latency communication. Fig. 9.17(a) shows the scenario when a vehicle-to-server communication is supported with a low latency. We see that due to the low latency value, the fog layer is considered suitable for processing applications, but only until the server load becomes high. Once the server load is overloaded, there will not be sufficient space to run other processes and applications. Even for good SH, an overloaded server will experience performance problems and decline other access requests. The increase of VSL indicates that the fog server is located farther and therefore more time is needed for the communication. The results for moderate and high VSL are shown in Fig. 9.17(b) and Fig. 9.17(c), respectively. We can see that for high VSL, the servers can be considered as suitable for helping the vehicle in need only when the server has a good SH and CSL is low.



(a) VSL = 0.1, MS = 0.1



(b) VSL = 0.1, MS = 0.5



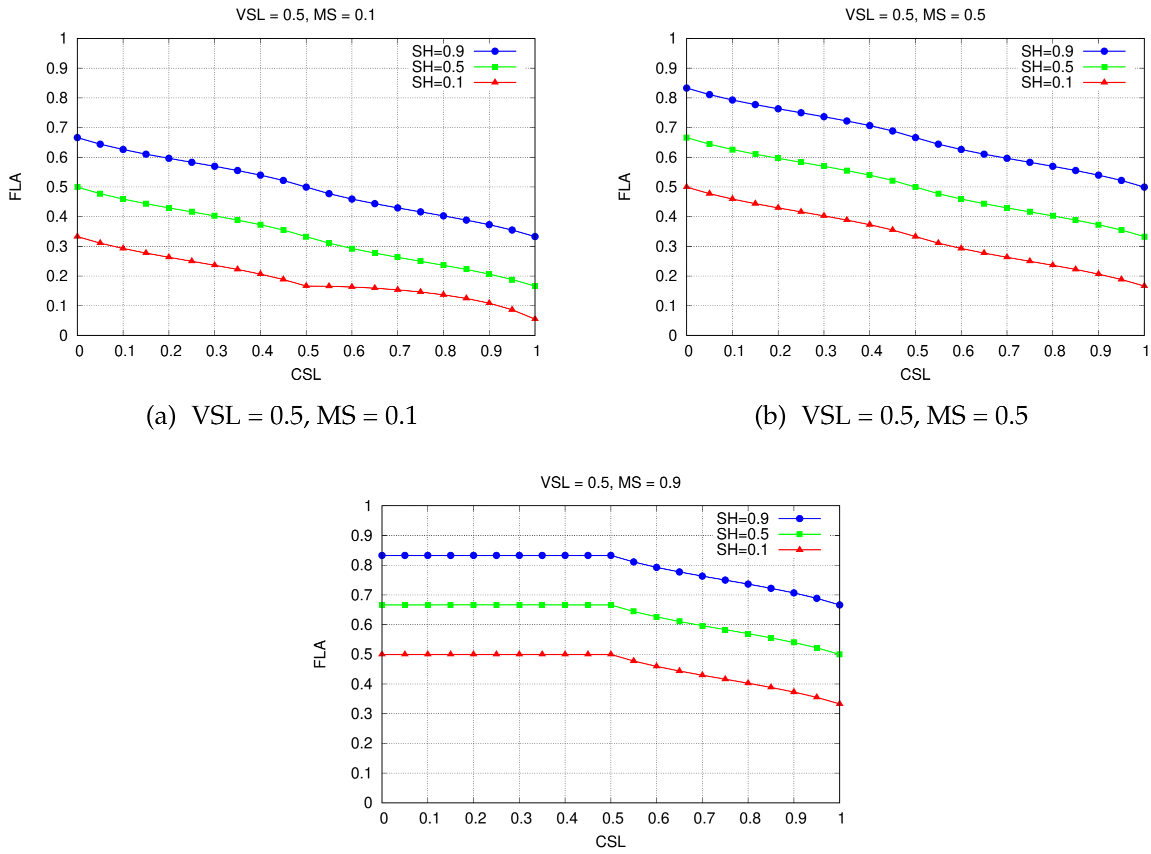
(c) VSL = 0.1, MS = 0.9

FIGURE 9.18: Simulation results for FLS2: VSL = 0.1.

9.4.2 Results of FLS2

Fig. 9.18, Fig. 9.19 and Fig. 9.20 show the results of FLS2 for low, moderate and high vehicle-to-sever communication latency, respectively. We show the relation between FLA and CSL for different SH values while keeping VSL and MS values constant. Fig. 9.18(a) shows the scenario when a low vehicle-to-server communication latency is supported for data transmission, while the time required for handover is long. We can see that in the case of a high server load, the fog layer is not considered appropriate to process an application, because there is no sufficient space to run other processes and applications. With the increase of MS (see Fig. 9.18(b) and Fig. 9.18(c)) the chances of the server to be considered suitable are increased. When MS is fast, the fog layer is considered suitable for processing applications regardless of the CSL and SH values.

The results for moderate VSL are shown in Fig. 9.19. In Fig. 9.19(a) it is shown that servers that have a bad reputation are not considered helpful. When MS increases to 0.5 (see Fig. 9.19(b)), the chances of fog layer to handle applications successfully are increased because more fog servers are considered suitable. In Fig. 9.19(c), we see that when MS is fast, only the servers with bad history are considered incapable of



(a) VSL = 0.5, MS = 0.1

(b) VSL = 0.5, MS = 0.5

(c) VSL = 0.5, MS = 0.9

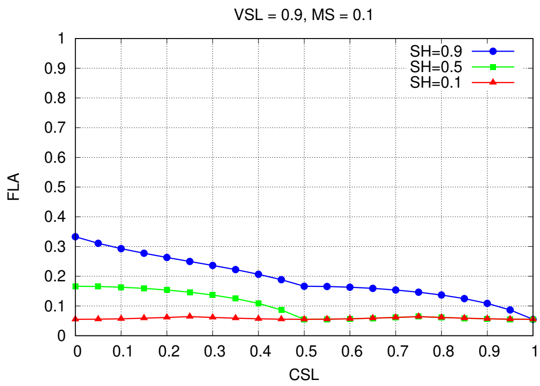
FIGURE 9.19: Simulation results for FLS2: VSL = 0.5.

processing the data.

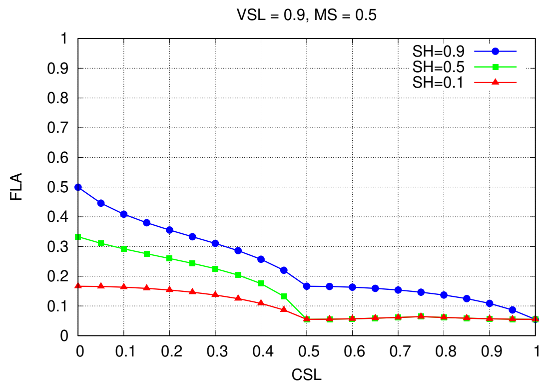
In Fig. 9.20 we consider the case of high VSL. Even when MS is fast, all values of FLA are lower than 0.5, thus the fog servers can not provide services to the *vehicle* in need.

9.4.3 Comparison of FLS1 and FLS2

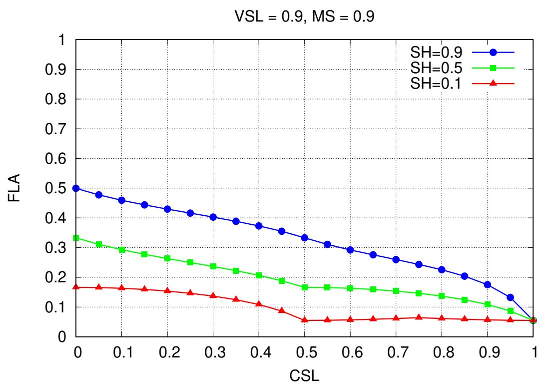
Comparing both models, FLS2 is more complex, but it takes into consideration the migration speed, which can be a determining factor considering the high mobility that characterizes VANETs. Each fog server provides services only to vehicles traveling nearby, and since vehicles may move out of their service area, it is critical to ensure a fast transfer of ongoing sessions and the continuity of services. On the contrary, FLS1 gives higher FLA results because it does not consider the effect of MS, and therefore it might not accurately estimate the fog layer adequacy. For example, FLS1 for VSL = 0.9 (Fig. 9.17(c)) shows similar results with FLS2 when VSL = 0.9 and MS = 0.1 (Fig. 9.20(a)). Nevertheless, the results change significantly for FLS2 for MS = 0.5 and



(a) VSL = 0.9, MS = 0.1



(b) VSL = 0.9, MS = 0.5



(c) VSL = 0.9, MS = 0.9

FIGURE 9.20: Simulation results for FLS2: VSL = 0.9.

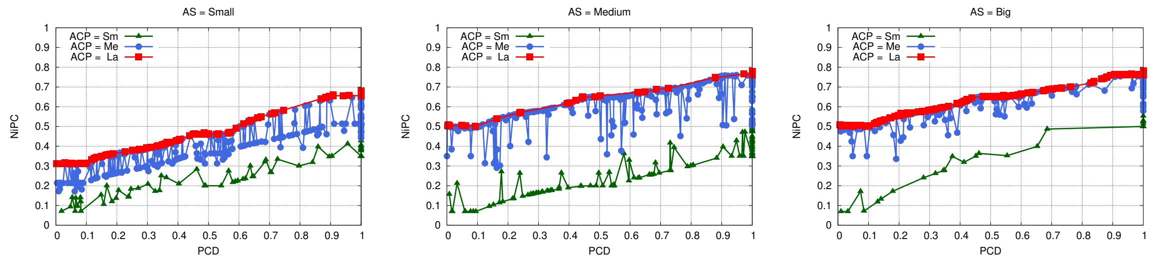
MS = 0.9. This is because FLS1 assumes an ideal environment without link failures occurring during the handover.

9.5 Experimental Results

In this section we discuss the experimental results for FS-ANVPC and FS-CFELS. The simulations were conducted by the communication of five RPis moving randomly, with one of them representing the vehicle in need of resources.

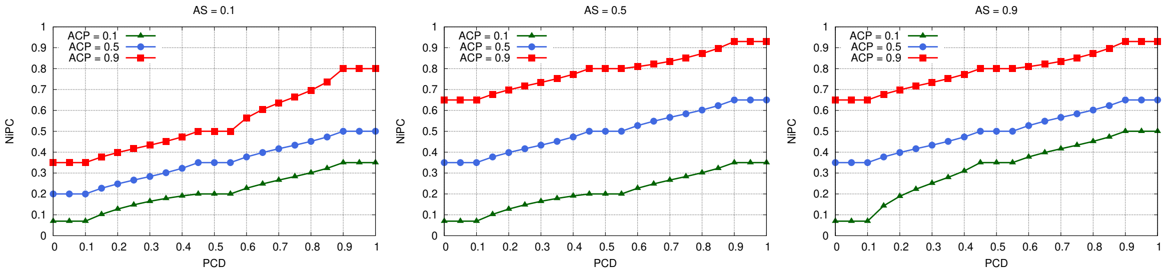
9.5.1 Experimental results of FS-ANVPC

The experimental results of FS-ANVPC are given in Figure 9.21. Let us compare them with the simulation results of FS-ANVPC1, given in Figure 9.22. We show the results for three scenarios: for small, medium and big AS. We can see that both the simulation and experimental results follow the same trend. For testbed results we can see more oscillations, especially for ACP small and medium. This is due to the fluctuations of the ACP of RPi while running many other applications simultaneously,



(a) AS = 0.1 (b) AS = 0.5 (c) AS = 0.9

FIGURE 9.21: Testbed results for FS-ANVPC1.



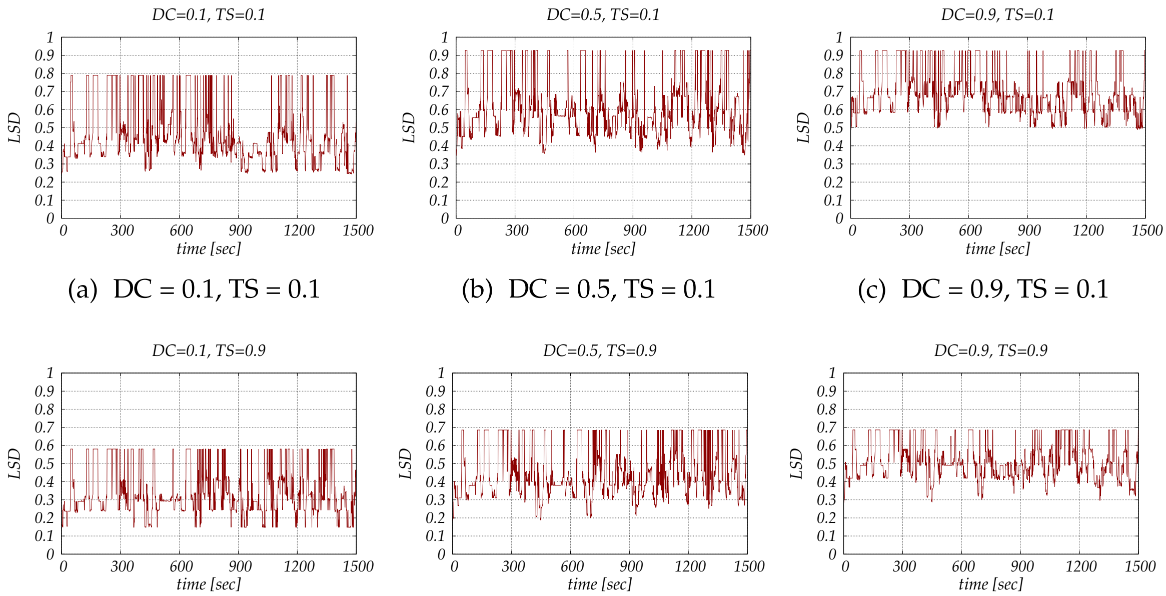
(a) AS = 0.1 (b) AS = 0.5 (c) AS = 0.9

FIGURE 9.22: Simulation results for FS-ANVPC1.

thus imitating a similar scenario of a vehicle operating its own applications and the applications of *vehicle* in need for additional resources. For small AS (see Figure 9.21(a)), vehicles that are considered helpful are the ones with large ACP and long PCD. Whereas, for medium and big AS (see Figure 9.21(b) and Figure 9.21(c)) vehicles with large ACP are considered helpful, despite the PCD value.

9.5.2 Experimental results of FS-CFELS

The experimental results of FS-CFELS are given in Figure 9.23. The parameters considered constant for presenting the results are DC and TS as these parameters represent the application requirements, and as such, they differ only from application to application. Therefore, each subfigure represents practically a different set of applications that have similar requirements. Using this configuration we can see how LSD relates with the changing characteristics of the edge layer, which are represented by NNV and APCpNV. The same idea is applied to testbed results too, which are shown in Figure 9.23. DC and TS are kept constant. Whereas, NNV changes in accordance with the random movement of neighbor vehicles in and out the communication range of the *vehicle* in need. APCpNV also changes in accordance with the condition of neighbor vehicles at a certain time, depending on the changing condition of their AS, ACP, and their movement in relation to the vehicle in need. In the testbed, the results are not distinguished as in the simulation graphs. The single



(a) DC = 0.1, TS = 0.1 (b) DC = 0.5, TS = 0.1 (c) DC = 0.9, TS = 0.1

(d) DC = 0.1, TS = 0.9 (e) DC = 0.5, TS = 0.9 (f) DC = 0.9, TS = 0.9

FIGURE 9.23: Testbed results for FS-CFELS.

red line represents the results for different NNV and APCpNV at the same time. Let us analyze and compare the simulation and testbed results.

In Figure 9.23(a) shows the results for non-complex applications that are delay tolerant. We can see that LSD values fall into the interval of [0.24, 0.78]. LSD values more than 0.7 are achieved when *the vehicle* has no available connection and therefore there is no processing capability of neighbor vehicles. In this case, the most appropriate layer is the cloud layer. The results are similar with the simulation results in Figure 9.14(a) for NNV=0.1 and APCpNV=0. In Figure 9.23(a), many of the LSD values fall between [0.3, 0.7] interval, which show the selection of fog layer for processing data. The characteristic of this vehicle network environment is having moderate APCpNV value. We can see that the same holds true for simulation results. Whereas, for the diapason of LSD values less than 0.3, the most appropriate layer for *the vehicle* to run its applications is considered the edge layer. In testbed, such results occur when APCpNV values is high and NNV=1. We can see that the same trend is followed also by simulation results for NNV=0.9 in Figure 9.14(a). The results show that when *the vehicle* is surrounded by many potentially helpful neighbors, even though the applications do not require real-time processing, the edge layer is decided as most suitable. Such decision has two benefits: it exploits the high capacity of neighbors which at this point is being unused and it avoids unnecessary traffic being sent in the core network.

With the increase of data complexity in Figure 9.23(b), LSD values are increased which denote for the selection of fog and cloud layer. Whereas in and Figure 9.23(c), as a results of having very complex data, the cloud layer is most appropriate. In both cases, we see that the edge layer is hardly selected as the data are very complex to be handled and real-time processing is not a requirement. This decision hold true also for simulation results shown in Figure 9.15(a) and Figure 9.16(a).

Time sensitivity applications are shown in Figure 9.23(d) - 9.23(f). For low data complexity (see Figure 9.23(d)) we can see that LSD values (which are less than 0.6) decide edge layer and fog layer as suitable to handle these applications. The increase of data complexity (see Figure 9.23(e) and Figure 9.23(f)) increases LSD value, however never reaches the cloud layer. Instead, it chooses the fog layer, as this layer provides powerful computing capabilities and real-time processing at the same time. We can see the same results also in Figure 9.15(c) and Figure 9.16(c).

Chapter 10

Conclusions

Vehicular networks are a critical component of the future smart cities. They are expected to lead to improved traffic management and road safety, lower air pollution and so on. First implementations are promising, but still many problems need to be tackled for a full deployment. Much remains to be done since the ultimate goal of this technology is to have the number of road fatalities from traffic crashes reduced to zero.

10.1 Summary of Thesis

After presenting the background, objectives, and main contributions of this thesis, in Chapter 2, we presented Wireless Networks. We discussed recent advances that are expected to support the trend of upcoming applications. The concepts were also described in regard to how these technologies are expected to assist in the widespread deployment of vehicular networks.

Then, in Chapter 3, we investigated in detail different applications and use cases of vehicular networks, introduced their network architectures and radio access technologies, and discussed aspects of security, privacy, and trust mechanisms. We highlighted the limitations of the current architectures and explored some of the emerging solutions.

In Chapter 4 we described the emerging technologies that enable the future use cases of vehicular networks. We start form the basic technologies such as virtualization to state-of-the-art technologies that enable the implementation of vehicular networks, such as SDNs, NFV, network slicing, and so on.

Extensive information about SDN and cloud, fog, edge computing is given in Chapter 5. We talk about different implementation of edge/fog computing technologies and explain the main differences between traditional networks and SDNs and the advantages it provides. In addition, the architecture integrating vehicular networks with cloud fog and edge computing and content flow is also explained.

Chapter 6 is dedicated to Intelligent Algorithms and their applications. Neural networks, genetic algorithms, and the most common techniques and methods of swarm intelligence and simulated annealing are among the intelligent approaches that are reviewed throughout the chapter.

In Chapter 7, we gave a thorough description of theoretical concepts underlying fuzzy logic, fuzzy sets, and fuzzy systems. The provided concepts are discussed in regards to the application of fuzzy logic presented in this thesis in order to enable a complete understanding of the proposed systems, which are detailed in Chapter 8.

In this Chapter 8 are given details of all fuzzy based systems. All the implemented models and their parameters were described by delineating the reason behind their selection. The set of linguistic values for each parameter, the chosen membership functions, and the fuzzy rule base of every fuzzy logic controller comprising the proposed systems, were also presented as information that is needed if one wants to obtain the same simulation results. After explaining each model, we introduced the testbed that we implemented to carry out experiments and to evaluate the systems experimentally.

Both simulation and experimental results for each system are given in Chapter 9. A comparison between simulation and experimental results is also discussed. The concluding remarks regarding the proposed systems and the results are given in the following section.

10.2 Concluding Remarks

The proposed intelligent fuzzy logic systems take into consideration the challenges that vehicular networks are currently facing. Of the utmost importance is the management of the abundant information and resources available in these networks. To do so, we proposed a fuzzy based integrated system, called Integrated Fuzzy-based System for Coordination and Management of Resources, consisting of three subsystems.

For first subsystem, Fuzzy-based System for Assessment of QoS, we proposed two fuzzy-based approaches that deal with criteria that data communication channels should meet in order to achieve reliable V2V communication links. Such communication issues are caused by the rapid and constant change of the network topology which are presented in our approach by input parameters that are the latency of message transmission between two neighboring vehicles, the reliability of data exchange, as well as the beacon signals disseminated across the network that inform vehicles about the state and location of their neighbors. In addition to these parameters, the second approach considered the interference of the concurrent transmission of nearby vehicles. The simulation results showed that the proposed approaches could effectively decide appropriate communication links between vehicles in order to exchange data packets successfully. V2V links carry mostly safety-critical messages, thus providing low latency, minimal interference and no packet losses becomes essential in order to guarantee a high performance of the critical applications. In addition, both systems reach peak values when beacon packets do not overload the network but have a moderate value. Comparing the models together, the second system makes a better assessment of the QoS of V2V communication links, since it takes into consideration factors that affect the network performance, although its complexity is higher than the first system, but still run in real-time.

The second subsystem, Fuzzy-based System for Assessment of Neighbor Vehicle Processing Capability, assessed the neighboring vehicles resource capability for which were considered two fuzzy-based approaches. Both approaches included the available resources of the neighboring vehicles and the predicted contact duration between them and the vehicle in need, while the second approach included in addition the QoS of the communication link among vehicles. The output of second subsytem served as an input parameter for the following subsystem to estimate the processing capability of the edge layer.

The third subsystem, Fuzzy-based System for Cloud-Fog-Edge Layer Selection, managed the resources of the edge, fog, and cloud layer and determined the appropriate layer for data storage and computing, based on the output of the fuzzy system. We considered this architecture from a bottom-up approach, which implies that the edge layer is the first layer considered based on the available connections and service requirements. If the application requirements are not fulfilled or there are no or only very few available connections, then the fog layer is taken into consideration.

It is essential for time-sensitive applications that a continuous low-latency service is always provided, but there are cases when edge layer lacks the needed resources or is not available for providing resources. As it was mentioned before, in this case, the fog layer is considered as a good solution since it processes the applications with no delay and provides high throughput. For assessing the fog layer we implemented two fuzzy based systems. As explained in Chapter 8, both systems considered vehicle-to-server latency, server history and current server load, while second system included migration speed in addition to other parameters. This system is not included on the integrated fuzzy system, but works alone in case an assessment of the fog layer processing capability is needed.

For the implementation of Integrate System for Coordination and Management of Resources in a testbed, we considered the following input parameters: neighbor vehicle available storage, neighbor vehicle available computing power, predicted contact duration between the *vehicle* and neighbor vehicle, processing capability of all neighbors inside the communication range of *the vehicle*, the density of neighbor vehicles that can maintain a communication with *the vehicle* for awhile, time sensitivity set by application requirements and data complexity of the application. We showed through simulations and experiments the effect of these parameters on the determination of appropriate layer for processing a certain application. The comparison made in Chapter 9 proves that experimental results follow the results of the simuation. The experimental results demonstrated that the proposed approach was feasible.

10.3 Future Directions

In order to take full advantage of the system, it is noteworthy to determine the accuracy of the system as it can show which parameters lead mostly to a false positive output and what should be improved to reduce the false-negative outputs. In the following are shown some future aspects regarding improvements that we aim to make in the proposed system.

In our testbed we used *sumo* simulator to generate the movement of vehicles. In the future, we aim to improve the testbed by implementing mobile RPis moving randomly in designated roads, equipped with GPS which enable real-time localization of vehicles (RPis).

Second, in our testbed we use only five RPis which represent the moving vehicles. We get an understanding of how the proposed system and the testbed performs, but we would like to implement a large size network with many more vehicles, making it more similar to a real life scenario. A large size network will show how the proposed system responds to data congestion, interference and many other problems that arise that can have a profound effect on the network performance.

Third, during the vehicle communication sessions in experiment no application is running, only small data size packets are exchanged. In the future, we intend to evaluate the performance by running a real application in *the vehicle*. In this way, we can prove whether the application in the end is performed successfully via the decision taken by the proposed system. For example, the fuzzy-based system decides that edge layer is capable to execute the application, when in fact the application might not be successfully accomplished in the edge layer. Such false positives/negatives determine the accuracy of the system, therefore it is important to investigate which parameters lead to these false results, in order to improve the system.

References

[1] C. P and I. R. JN, "Measuring road congestion," Luxembourg (Luxembourg), Scientific analysis or review LF-NA-25550-EN-N, 2012. DOI: 10.2791/15282.

[2] F. Mizutani, Y. Suzuki, and H. Sakai, "Estimation of social costs of transport in japan," *Urban Studies*, vol. 48, no. 16, pp. 3537–3559, 2011, ISSN: 00420980, 1360063X. (visited on 05/31/2022).

[3] B Schrank, L Albert, and B. a. Eisele, "Urban mobility report," *The Texas A&M Transportation Institute with cooperation from INRIX*, June, 2021.

[4] G. Heaton, "Traffic jams in japan: Does public transit have an impact?" Ph.D. dissertation, 2015.

[5] R Anderson, "The united states experience with economic incentives for protecting the environment," United States Enviromental Protection Agency, 2001.

[6] World Health Organization (WHO), *Global status report on road safety 2018: summary*. Geneva, Switzerland, 2018, (WHO/NMH/NVI/18.20). Licence: CC BY-NC-SA 3.0 IGO).

[7] S. Jat, R. S. Tomar, and M. Satya Prakash Sharma, "Traffic analysis for accidents reduction in vanet's," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2019, pp. 115–118. DOI: 10.1109/ICCIKE47802.2019.9004270.

[8] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, "Multihop Ad Hoc Networking: The Evolutionary Path," in *Mobile Ad Hoc Networking: The Cutting Edge Directions*. 2013, pp. 1–33. DOI: 10.1002/9781118511305.ch1.

[9] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005. DOI: 10.1017/CBO9780511841224.

[10] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003. DOI: https://doi.org/10.1016/S1570-8705(03)00013-1.

[11] Federal Highway Administration (FHWA), "FHWA White Paper On Mobile Ad Hoc Networks," White Paper FHWA-HRT-18-027, 2018.

[12] C. Cox, "Chapter I - Introduction," in *An Introduction to LTE: LTE, LTE-Advanced, SAE, VoLTE and 4G Mobile Communications*. 2014, pp. 1–20. DOI: 10.1002/9781118818046.ch1.

[13] 3rd Generation Partnership Project (3GPP), "Technical Specification Group Services and System Aspects; Study on LTE Support of Vehicle to Everything (V2X) Services (Release 14)," Technical Report 22.885, Dec. 2015, V14.0.0.

[14] International Telecommunication Union. Rec. ITU-R M.2083-0, *MT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond*, Mobile, radiodetermination, amateur and related satellite services (M Series), Geneva, Sep. 2015.

[15] International Telecommunication Union. Report ITU-R M.2410-0, *Minimum requirements related to technical performancefor IMT-2020 radiointerface(s)*, Mobile, radiodetermination, amateur and related satellite services (M Series), Geneva, Nov. 2017.

[16] W. Dargie and C. Poellabauer, "Motivation for a Network of Wireless Sensor Nodes," in *Fundamentals of Wireless Sensor Networks: Theory and Practice*. 2011, pp. 1–16. DOI: 10.1002/9780470666388.ch1.

[17] Y. Wang, G. Attebury, and B. Ramamurthy, "A Survey of Security Issues in Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 8, no. 2, pp. 2–23, 2006. DOI: 10.1109/COMST.2006.315852.

[18] F. Losilla, A.-J. Garcia-Sanchez, F. Garcia-Sanchez, J. Garcia-Haro, and Z. J. Haas, "A Comprehensive Approach to WSN-Based ITS Applications: A Survey," *Sensors*, vol. 11, no. 11, pp. 10 220–10 265, 2011. DOI: doi.org/10.3390/s111110220.

[19] M. Sharma, M. Singh, K. Walia, and K. Kaur, "A comprehensive study of performance parameters for manet, vanet and fanet," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 0643–0646. DOI: 10.1109/IEMCON.2019.8936159.

[20] A. J. Maidamwar and R. V. Sadakale, "Comprehensive study for localization techniques in manet and vanet," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 2018, pp. 349–352. DOI: 10.1109/ICACCT.2018.8529557.

[21] S. M. AlMheiri and H. S. AlQamzi, "MANETs and VANETs Clustering Algorithms: A Survey," in *IEEE 8th GCC Conference Exhibition*, 2015, pp. 1–6. DOI: 10.1109/IEEEGCC.2015.7060048.

[22] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China Communications*, vol. 11, no. 10, pp. 1–15, 2014. DOI: 10.1109/CC.2014.6969789.

[23] W. Zhang and X. Xi, "The innovation and development of internet of vehicles," *China Communications*, vol. 13, no. 5, pp. 122–127, 2016. DOI: 10.1109/CC.2016.7489980.

[24] H. Zhou, T. Ma, Y. Xu, N. Cheng, and X. Yan, "Software-defined multi-mode access management in cellular v2x," *Journal of Communications and Information Networks*, vol. 6, no. 3, pp. 224–236, 2021. DOI: 10.23919/JCIN.2021.9549119.

[25] 3rd Generation Partnership Project (3GPP), "Technical Specification Group Services and System Aspects; Study on Enhancement of 3GPP Support for 5G V2X Services (Release 15)," Technical Report 22.886, Sep. 2018, V15.3.0.

[26] 5G Automotive Association (5GAA), *Explore the technology: C-V2X*, https://5gaa.org/5g-technology/c-v2x/, (Last Accessed: 15-05-2021).

[27] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments," in *VTC Spring 2008 - IEEE Vehicular Technology Conference*, 2008, pp. 2036–2040. DOI: 10.1109/VETECS.2008.458.

[28] Federal Communications Commission (FCC), "FCC 20-164. FCC Report and Order on Use of the 5.850-5.925 GHz Band," Report and Order, 2020.

[29] 3rd Generation Partnership Project (3GPP), "Technical Specification Group Services and System Aspects; Architecture Enhancements for 5G System (5GS) to Support Vehicle-to-Everything (V2X) Services (Release 16)," Technical Report 23.287, Dec. 2020, V16.5.0.

[30] International Telecommunication Union. Rec. ITU-R M.1890-1, *Operational radiocommunication objectives and requirements for advanced Intelligent Transport Systems*, Mobile, radiodetermination, amateur and related satellite services (M Series), Geneva, Jan. 2019.

[31] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions," Technical Report 102 638, Jun. 2009, V1.1.1.

[32] Fifth Generation Communication Automotive Research & innovation (5GCAR), "5GCAR Scenarios, Use Cases, Requirements and KPIs," The 5G Infrastructure Public Private Partnership (5GPPP), Deliverable D2.1 V2.0, Feb. 2019.

[33] 5G for Connected and Automated Road Mobility in the European UnioN (5GCARMEN), "5G CARMEN Use Cases and Requirements," The 5G Infrastructure Public Private Partnership (5GPPP), Deliverable D2.1, May 2019.

[34] IEEE Computer Society, *802.11-2020 - 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*, Feb. 2021. DOI: 10.1109/IEEESTD.2021.9363693.

[35] Christiane Köllner, *Bluetooth will network millions of vehicles in the future*, https://www.bluetooth.com/news/bluetooth-will-network-millions-of-vehicles-in-the-future/, Bluetooth SIG, Inc. (Last Accessed: 15-05-2021).

[36] U. Mohammad, N. Al-Holou, and C. Balas, "Performance Evaluation of IEEE 802.15.4/ZigBee Protocol for Automotive Applications," SAE, Technical Paper 2008-01-0278, 2008. DOI: 10.4271/2008-01-0278.

[37] IEEE Computer Society, *802.11p-2010 - IEEE Standard for Information Technology–Local and Metropolitan Area Networks– Specific Requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, Jul. 2010. DOI: 10.1109/IEEESTD.2010.5514475.

[38] IEEE Vehicular Technology Society, *1609.3-2020 - IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Networking*, Mar. 2021. DOI: 10.1109/IEEESTD.2021.9374154.

[39] C-ITS Deployment Group, *C-ITS deployment takes off, increasing road safety and decreasing congestion*, https://c-its-deployment-group.eu/mission/statements/december-2019-constitutive-act-of-c-its-deployment-group/, (Last Accessed: 15-05-2021).

[40] 3rd Generation Partnership Project (3GPP), "Technical Specification Group Radio Access Network; Study on LTE-Based V2X Services (Release 14)," Technical Report 36.885, Jun. 2016, V14.0.0.

[41] 3rd Generation Partnership Project (3GPP), "Technical Specification Group Services and System Aspects; Enhancement of 3GPP Support for V2X Scenarios; Stage 1 (Release 15)," Technical Report 22.186, Sep. 2018, V15.4.0.

[42] J. Chen, B. Liu, H. Zhou, L. Gui, N. Liu, and Y. Wu, "Providing Vehicular Infotainment Service Using VHF/UHF TV Bands via Spatial Spectrum Reuse," *IEEE Transactions on Broadcasting*, vol. 61, no. 2, pp. 279–289, 2015. DOI: 10.1109/TBC.2015.2400819.

[43] O. Altintas, Y. Ihara, H. Kremo, *et al.*, "Field Tests and Indoor Emulation of Distributed Autonomous Multi-Hop Vehicle-to-Vehicle Communications over TV White Space," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12, Istanbul, Turkey: Association for Computing Machinery, 2012, 439–442. DOI: `10.1145/2348543.2348605`.

[44] H. Zhou, N. Zhang, Y. Bi, Q. Yu, X. S. Shen, D. Shan, and F. Bai, "TV White Space Enabled Connected Vehicle Networks: Challenges and Solutions," *IEEE Network*, vol. 31, no. 3, pp. 6–13, 2017. DOI: `10.1109/MNET.2017.1600049NM`.

[45] J.-H. Lim, W. Kim, K. Naito, J.-H. Yun, D. Cabric, and M. Gerla, "Interplay Between TVWS and DSRC: Optimal Strategy for Safety Message Dissemination in VANET," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 11, pp. 2117–2133, 2014. DOI: `10.1109/JSAC.2014.1411RP02`.

[46] M. Fadda, M. Murroni, and V. Popescu, "Interference Issues for VANET Communications in the TVWS in Urban Environments," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 4952–4958, 2016. DOI: `10.1109/TVT.2015.2453633`.

[47] F. B. de Carvalho, W. T. Lopes, M. S. Alencar, and J. V. Filho, "Cognitive Vehicular Networks: An Overview," *Procedia Computer Science*, vol. 65, pp. 107–114, 2015, International Conference on Communications, Management, and Information technology (ICCMIT'2015). DOI: `10.1016/j.procs.2015.09.086`.

[48] J. Eze, S. Zhang, E. Liu, and E. Eze, "Cognitive Radio Technology Assisted Vehicular Ad Hoc Networks (VANETs): Current Status, Challenges, and Research Trends," in *2017 23rd International Conference on Automation and Computing (ICAC)*, 2017, pp. 1–6. DOI: `10.23919/IConAC.2017.8082035`.

[49] K. D. Singh, P. Rawat, and J.-M. Bonnin, "Cognitive Radio for Vehicular Ad Hoc Networks (CR-VANETs): Approaches and Challenges," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1–22, 2014. DOI: `10.1186/1687-1499-2014-49`.

[50] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," Technical Report 102 637-2, Mar. 2011.

[51] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," Technical Report 102 637-3, Sep. 2010.

[52] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in Internet of Vehicles: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2339–2352, 2015. DOI: 10.1109/TITS.2015.2423667.

[53] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, Protocols, and Security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2018. DOI: 10.1109/JIOT.2017.2690902.

[54] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym Schemes in Vehicular Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 228–255, 2015. DOI: 10.1109/COMST.2014.2345420.

[55] C. A. Kerrache, C. T. Calafate, J.-C. Cano, N. Lagraa, and P. Manzoni, "Trust Management for Vehicular Networks: An Adversary-Oriented Overview," *IEEE Access*, vol. 4, pp. 9293–9307, 2016. DOI: 10.1109/ACCESS.2016.2645452.

[56] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANET Security Challenges and Solutions: A Survey," *Vehicular Communications*, vol. 7, pp. 7–20, 2017. DOI: 10.1016/j.vehcom.2017.01.002.

[57] H. Shafiq, R. A. Rehman, and B.-S. Kim, "Services and Security Threats in SDN Based VANETs: A Survey," *Wireless Communications and Mobile Computing*, vol. 2018, p. 8 631 851, 2018. DOI: 10.1155/2018/8631851.

[58] Z. Lu, G. Qu, and Z. Liu, "A Survey on Recent Advances in Vehicular Network Security, Trust, and Privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 760–776, 2019. DOI: 10.1109/TITS.2018.2818888.

[59] S. Sharma and B. Kaushik, "A Survey on Internet of Vehicles: Applications, Security Issues & Solutions," *Vehicular Communications*, vol. 20, p. 100 182, 2019. DOI: 10.1016/j.vehcom.2019.100182.

[60] D. Manivannan, S. S. Moni, and S. Zeadally, "Secure Authentication and Privacy-Preserving Techniques in Vehicular Ad-hoc NETworks (VANETs)," *Vehicular Communications*, vol. 25, p. 100 247, 2020, ISSN: 2214-2096. DOI: 10.1016/j.vehcom.2020.100247.

[61] C. Lai, R. Lu, D. Zheng, and X. Shen, "Security and Privacy Challenges in 5G-Enabled Vehicular Networks," *IEEE Network*, vol. 34, no. 2, pp. 37–45, 2020. DOI: 10.1109/MNET.001.1900220.

[62] R. Hussain, J. Lee, and S. Zeadally, "Trust in VANET: A Survey of Current Solutions and Future Research Opportunities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2553–2571, 2021. DOI: 10.1109/TITS.2020.2973715.

[63] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-Based Decentralized Trust Management in Vehicular Networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2019. DOI: 10.1109/JIOT.2018.2836144.

[64] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *MCC Workshop on Mobile Cloud Computing*, ACM, 2012, pp. 13–16. DOI: 10.1145/2342509.2342513.

[65] David Linthicum, *Edge computing vs. fog computing: Definitions and enterprise uses*, https://www.cisco.com/c/en/us/solutions/enterprise-networks/edge-computing.html, Cisco, (Last Accessed: 2019-07-30).

[66] G. Samara, M. Rasmi, N. A. Sweerky, E. A. Daoud, and A. A. Salem, "Improving vanet's performance by incorporated fog-cloud layer (fcl)," in *2021 22nd International Arab Conference on Information Technology (ACIT)*, 2021, pp. 1–5. DOI: 10.1109/ACIT53391.2021.9677345.

[67] H.-J. Hong, "From cloud computing to fog computing: Unleash the power of edge and end devices," in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2017, pp. 331–334. DOI: 10.1109/CloudCom.2017.53.

[68] A. Waheed, M. A. Shah, S. M. Mohsin, A. Khan, C. Maple, S. Aslam, and S. Shamshirband, "A comprehensive review of computing paradigms, enabling computation offloading and task execution in vehicular networks," *IEEE Access*, vol. 10, pp. 3580–3600, 2022. DOI: 10.1109/ACCESS.2021.3138219.

[69] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017. DOI: 10.1109/COMST.2017.2705720.

[70] Z. Xu, Y. Zhang, X. Qiao, H. Cao, and L. Yang, "Energy-efficient offloading and resource allocation for multi-access edge computing," in *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2019, pp. 1–2. DOI: 10.1109/ICCE-TW46550.2019.8991706.

[71] C.-M. Huang and C.-F. Lai, "The delay-constrained and network-situation-aware v2v2i vanet data offloading based on the multi-access edge computing (mec) architecture," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 331–347, 2020. DOI: 10.1109/OJVT.2020.3028684.

[72] D. S. Linthicum, "Connecting fog and cloud computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 18–20, 2017. DOI: 10.1109/MCC.2017.37.

[73] N. K. Giang, R. Lea, M. Blackstock, and V. C. Leung, "Fog at the edge: Experiences building an edge computing platform," in *2018 IEEE International Conference on Edge Computing (EDGE)*, 2018, pp. 9–16. DOI: 10.1109/EDGE.2018.00009.

[74] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in *2017 IEEE International Conference on Edge Computing (EDGE)*, 2017, pp. 17–24. DOI: 10.1109/IEEE.EDGE.2017.12.

[75] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, "Towards Software-defined VANET: Architecture and Services," in *Annual Mediterranean Ad Hoc Networking Workshop*, IEEE, Jun. 2014, pp. 103–110. DOI: 10.1109/MedHocNet.2014.6849111.

[76] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software Defined Networking-Based Vehicular Adhoc Network with Fog Computing," in *International Symposium on Integrated Network Management*, IEEE, May 2015, pp. 1202–1207. DOI: 10.1109/INM.2015.7140467.

[77] D. King, C. Rotsos, A. Aguado, N. Georgalas, and V. Lopez, "The software defined transport network: Fundamentals, findings and futures," in *2016 18th International Conference on Transparent Optical Networks (ICTON)*, 2016, pp. 1–4. DOI: 10.1109/ICTON.2016.7550669.

[78] M. S. Olimjonovich, "Software defined networking: Management of network resources and data flow," in *2016 International Conference on Information Science and Communications Technologies (ICISCT)*, 2016, pp. 1–3. DOI: 10.1109/ICISCT.2016.7777384.

[79] A. Bhatia, K. Haribabu, K. Gupta, and A. Sahu, "Realization of flexible and scalable vanets through sdn and virtualization," in *2018 International Conference on Information Networking (ICOIN)*, 2018, pp. 280–282. DOI: 10.1109/ICOIN.2018.8343125.

[80] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997. DOI: 10.1109/4235.585893.

[81] X.-S. Yang, "Chapter 1 - Introduction to Algorithms," in *Nature-Inspired Optimization Algorithms (Second Edition)*, Academic Press, 2021, pp. 1–22. DOI: 10.1016/B978-0-12-821986-7.00008-1.

[82] X. Ma, X. Li, Q. Zhang, K. Tang, Z. Liang, W. Xie, and Z. Zhu, "A survey on cooperative co-evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 421–441, 2019. DOI: 10.1109/TEVC.2018.2868770.

[83] X.-S. Yang, "Chapter 6 - Genetic Algorithms," in *Nature-Inspired Optimization Algorithms (Second Edition)*, Academic Press, 2021, pp. 91–100. DOI: 10.1016/B978-0-12-821986-7.00013-5.

[84] M. Dorigo, "Optimization, Learning and Natural Algorithms (in Italian)," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.

[85] K. S. Kaswan, S. Choudhary, and K. Sharma, "Applications of artificial bee colony optimization technique: Survey," in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015, pp. 1660–1664.

[86] Y. Liu, L. Ma, and G. Yang, "A survey of artificial bee colony algorithm," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2017, pp. 1510–1515. DOI: 10.1109/CYBER.2017.8446301.

[87] C. Kalpana and B. Booba, "Bio-inspired firefly algorithm a methodical survey – swarm intelligence algorithm," in *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, 2021, pp. 1–7. DOI: 10.1109/ICCICA52458.2021.9697171.

[88] L. Rosenberg, G. Willcox, D. Askay, L. Metcalf, and E. Harris, "Amplifying the social intelligence of teams through human swarming," in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, 2018, pp. 23–26. DOI: 10.1109/AI4I.2018.8665698.

[89] A. Amuthan and K. Deepa Thilak, "Survey on tabu search meta-heuristic optimization," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, pp. 1539–1543. DOI: 10.1109/SCOPES.2016.7955697.

[90] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated Annealing: From Basics to Applications," in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds. Cham: Springer International Publishing, 2019, pp. 1–35. DOI: 10.1007/978-3-319-91086-4_1.

[91] D. Oliva and E. Cuevas, "Electromagnetism—like optimization algorithm: An introduction," in *Advances and Applications of Optimised Algorithms in Image Processing*. Cham: Springer International Publishing, 2017, pp. 23–41. DOI: 10.1007/978-3-319-48550-8_3.

[92] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965. DOI: 10.1016/S0019-9958(65)90241-X.

[93] T. J. Ross, "Classical sets and fuzzy sets," in *Fuzzy Logic with Engineering Applications, Third Edition*. John Wiley Sons, Ltd, 2010, pp. 25–42. DOI: 10.1002/9781119994374.

[94] K. Bylykbashi, E. Qafzezi, P. Ampririt, M. Ikeda, K. Matsuo, and L. Barolli, "Performance Evaluation of an Integrated Fuzzy-Based Driving-Support System for Real-Time Risk Management in VANETs," *Sensors*, vol. 20, no. 22, p. 6537, 2020. DOI: 10.3390/s20226537.

[95] X. Ge, Z. Li, and S. Li, "5g software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, 2017, ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1601144.

[96] L. A. Zadeh and J. Kacprzyk, *Fuzzy Logic for the Management of Uncertainty*. New York, NY, USA: John Wiley & Sons, Inc., 1992, ISBN: 0-471-54799-9.

[97] A. Kandel, *Fuzzy Expert Systems*. Boca Raton, FL, USA: CRC Press, Inc., 1992, ISBN: 0-8493-4297-X.

[98] F. M. McNeill and E. Thro, *Fuzzy Logic: A Practical Approach*. San Diego, CA, USA: Academic Press Professional, Inc., 1994, ISBN: 0-12-485965-8. DOI: 10.1016/C2013-0-11164-6.

[99] H.-J. Zimmermann, "Fuzzy control," in *Fuzzy Set Theory—and Its Applications*, Springer, 1996, pp. 203–240.

[100] T. Munakata and Y. Jani, "Fuzzy Systems: An Overview," *Communications of the ACM*, vol. 37, no. 3, pp. 69–77, 1994. DOI: 10.1145/175247.175254.

[101] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Upper Saddle River, NJ, USA: Prentice Hall, Inc., 1987, ISBN: 0-13-345984-5.

List of Publications

International Journals (First Author)

1. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "An Intelligent Approach for Cloud-Fog-Edge Computing SDN-VANETs Based on Fuzzy Logic: Effect of Different Parameters on Coordination and Management of Resources," Sensors, MDPI, Vol. 22, No. 3, 878; DOI: 10.3390/s22030878, February 2022. IF: 3.847, CiteScore: 6.4.

2. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "FSAQoS: A Fuzzy-Based System for Assessment of QoS of V2V Communication Links in SDN-VANETs and Its Performance Evaluation," International Journal of Distributed Systems and Technologies (IJDST), IGI Global, Vol. 13, No. 1, p. 7, DOI: 10.4018/IJDST.300338, 2022. Indexed in: [Web of Science], CiteScore: 1.2.

3. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A QoS-Aware Fuzzy-Based System for Assessment of Edge Computing Resources in SDN-VANETs: System Implementation and Performance Evaluation," International Journal of Mobile Computing and Multimedia Communications (IJMCMC), IGI Global, Vol. 12, No. 4, pp. 1-16, DOI: 10.4018/IJMCMC.289161, October 2021. Indexed in: [Web of Science], CiteScore: 1.8

4. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-based Approach for Resource Management in SDN-VANETs: Effect of Trustworthiness on Assessment of Available Edge Computing Resources", Journal of High Speed Networks (JHSN), IOS Press, Vol. 27, No. 1, pp. 33-44, DOI: 10.3233/JHS-210650, 2021. Indexed in: [Web of Science], CiteScore: 1.3.

5. **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "Coordination and Management of Cloud, Fog and Edge Resources in SDN-VANETs Using Fuzzy Logic: A Comparison Study for Two Fuzzy-based Systems", Internet of Things (IoT), Elsevier, Vol. 11, Article 100169, 12 pages, DOI: 10.1016/j.iot.2020.100169, September 2020. IF: 5.711, CiteScore: 10.2.

International Journals (Co-author)

1. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "Implementation and evaluation of a fuzzy-based system for determining stress feeling level in VANETs: Effect of driving experience and history on driver stress," Journal of High Speed Networks (JHSN), IOS Press, Vol. Pre-press, No. Pre-press, pp. 1–13, DOI: 10.3233/JHS-220693, 2022.

2. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, "A hybrid intelligent system based on particle swarm optimization and distributed genetic algorithm for WMNs: A comparison study of boulevard and stadium distributions considering different router replacement methods and load balancing," Wireless Networks (WINE), Springer, DOI: 10.1007/s11276-022-03050-5, First Online: 27 June 2022.

3. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy Logic Approach for Determining Driver Impatience and Stress Leveraging Internet of Vehicles Infrastructure," Vehicles, MDPI, Vol. 4, No. 2, pp. 553-566, DOI: 10.3390/vehicles4020032, June 2022.

4. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "FSSLA: A Fuzzy-based Scheme for Slice Service Level Agreement in 5G Wireless Networks and Its Performance Evaluation ," Journal of High Speed Networks (JHSN), IOS Press, Vol. 28, No. 1, pp. 47–64, DOI: 10.3233/JHS-220678, February 2022.

5. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, "A Comparison Study of Weibull, Normal and Boulevard Distributions for Wireless Mesh Networks Considering Different Router Replacement Methods by a Hybrid Intelligent Simulation System," Journal of Ambient Intelligence and Humanized Computing (JAIHC), Springer, DOI: 10.1007/s12652-021-03680-1, First Online: 13 January 2022.

6. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "IFACS-Q3S: A New Admission Control System for 5G Wireless Networks Based on Fuzzy Logic and Its Performance Evaluation," International Journal of Distributed Systems and Technologies (IJDST), IGI Global, Vol. 13, No. 1, p. 8, DOI: 10.4018/IJDST.300339, 2022.

7. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, "Performance Evaluation of Chi-Square and Normal Distributions of Mesh Clients for WMNs Considering Five Router Replacement Methods ,"

International Journal of Distributed Systems and Technologies (IJDST), IGI Global, Vol. 13, No. 1, p. 2, DOI: 10.4018/IJDST.296247, 2022.

8. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, Makoto Takizawa, "A Comparison Study of Chi-square and Uniform Distributions of Mesh Clients for Different Router Replacement Methods Using WMN-PSODGA Hybrid Intelligent Simulation System," Journal of High Speed Networks (JHSN), IOS Press, Vol. 27, No. 4, pp. 319-334, DOI: 10.3233/JHS-210670, October 2021..

9. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "Application of Fuzzy Logic for Slice QoS in 5G Networks: A Comparison Study of Two Fuzzy-based Schemes for Admission Control," International Journal of Mobile Computing and Multimedia Communications (IJMCMC), IGI Global, Vol. 12, No. 2, DOI: 10.4018/IJMCMC.2021040102, April 2021.

10. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "Effect of Vehicle Technical Condition on Real-time Driving Risk Management in Internet of Vehicles: Design and Performance Evaluation of an Integrated Fuzzy-based System", Internet of Things (IoT), Elsevier, Vol. 13, p. 100363; DOI: 10.1016/j.iot.2021.100363, March 2021.

11. Phudit Ampririt, Seiji Ohara, **Ermioni Qafzezi**, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "An Integrated Fuzzy-based Admission Control System (IFACS) for 5G Wireless Networks: Its Implementation and Performance Evaluation", Internet of Things (IoT), Elsevier, Vol. 13, p. 100351; DOI: 10.1016/j.iot.2020.100351, March 2021.

12. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "Performance Evaluation of an Integrated Fuzzy-Based Driving-Support System for Real-Time Risk Management in VANETs", Sensors, MDPI, Vol. 20, No. 22, p. 6537; DOI: 10.3390/s20226537, November 2020.

13. Seiji Ohara, **Ermioni Qafzez**i, Admir Barolli, Shinji Sakamoto, Yi Liu, Leonard Barolli, "WMN-PSODGA - An Intelligent Hybrid Simulation System for WMNs Considering Load Balancing: A Comparison for Different Client Distributions", International Journal of Distributed Systems and Technologies (IJDST), IGI Global, Vol. 11, No. 4, pp. 39-52, DOI: 10.4018/IJDST.2020100103, 2020.

14. Yi Liu, **Ermioni Qafzezi**, Phudit Ampririt, Seiji Ohara, Leonard Barolli, “FBPCQS-Fuzzy-Based Peer Coordination Quality Systems for P2P Networks: Implementation and Performance Evaluation”, International Journal of Mobile Computing and Multimedia Communications (IJMCMC), IGI Global, Vol. 11, No. 3, pp. 22-37 DOI: 10.4018/IJMCMC.2020070102, 2020.

15. Yi Liu, **Ermioni Qafzezi**, Seiji Ohara, Kevin Bylykbashi, Leonard Barolli, “A Comparison Study of Two Fuzzy-based Decision Visiting Systems (FBDVSs) for Sightseeing Spots”, International Journal of Web Information Systems (IJWIS), Emerald Publishing Limited, Vol. 16 No. 2, pp. 249-262, DOI: 10.1108/IJWIS-10-2019-0046, 2020.

16. Kevin Bylykbashi, **Ermioni Qafzezi**, Makoto Ikeda, Keita Matsuo, Leonard Barolli, “Fuzzy-based Driver Monitoring System (FDMS): Implementation of Two Intelligent FDMSs and a Testbed for Safe Driving in VANETs”, Future Generation Computer Systems (FGCS), Elsevier, Vol. 105, pp. 665-674, DOI: 10.1016/j.future.2019.12.030, April 2020.

International Conferences (First Author)

1. **Ermioni Qafzezi**, Kevin Bylykbashi, Elis Kulla, Makoto Ikeda, Keita Matsuo, Leonard Barolli, “Implementation of a Fuzzy-based Testbed for Coordination and Management of Cloud-Fog-Edge Resources in SDN-VANETs, ” Proc. of EIDWT-2023, Semarang, Indonesia, February 23-25, 2023.

2. **Ermioni Qafzezi**, Kevin Bylykbashi, Admir Barolli, Makoto Ikeda, Keita Matsuo, Leonard Barolli, “A Comparison Study of Two Fuzzy-based Systems for Assessment of Fog Computing Resources in SDN-VANETs,” Proc. of 3PGCIC-2022, Tirana, Albania, October 27–29, 2022.

3. **Ermioni Qafzezi**, Kevin Bylykbashi, Admir Barolli, Makoto Ikeda, Keita Matsuo, Leonard Barolli, “A Fuzzy-Based System for Assessment of Fog Computing Resources in SDN-VANETs Considering Service Migration Speed as a New Parameter,” Proc. of INCoS-2022, pp. 140-149, Hyogo, Japan, DOI: 10.1007/978-3-031-14627-5_14_18, September 7–9, 2022.

4. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, “A Fuzzy-Based System for Assessment of Fog Computing Resources in SDN-VANETs,” Proc. of IMIS-2022, pp. 1–9, Kitakyushu, Japan, DOI: 10.1007/978-3-031-08819-3_1, June 29–July 1, 2022.

5. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Survey on Advances in Vehicular Networks: Problems and Challenges of Architectures, Radio Technologies, Use Cases, Data Dissemination and Security," Proc. of AINA-2022, Vol. 3, pp. 602-613, Sydney, Australia, DOI: 10.1007/978-3-030-99619-2_56, April 13–15, 2022.

6. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Assessment of QoS of V2V Communication Links in SDN-VANETs," Proc. of EIDWT-2022, pp. 153–162, Okayama, Japan, DOI: 10.1007/978-3-030-95903-6_17, March 2–4, 2022.

7. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Assessment of Quality of Service Communication Links in SDN-VANETs," Proc. of BWCCA-2021, pp. 120–128, Fukuoka, Japan, DOI: 10.1007/978-3-030-90072-4_12, October 28–30, 2021.

8. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A QoS-aware Fuzzy-based System for Assessment of Edge Computing Resources in SDN-VANETs," Proc. of AINA-2021, Vol. 1, pp. 63-72, Toronto, Canada, DOI: 10.1007/978-3-030-75100-5_6, May 12–14, 2021.

9. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Leonard Barolli, Makoto Takizawa, "Resource Management in SDN-VANETs Using Fuzzy Logic: Effect of Average Processing Capability per Neighbor Vehicle on Management of Cloud-Fog-Edge Resources," Proc. of EIDWT-2021, pp. 155-167, Chiang Mai, Thailand, DOI: 10.1007/978-3-030-70639-5_15, February 25–27, 2021.

10. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Leonard Barolli, Makoto Takizawa, "Assessment of Available Edge Computing Resources in SDN-VANETs by a Fuzzy-Based System Considering Trustworthiness as a New Parameter," Proc. of 3PGCIC-2020, pp. 102-112, Yonago, Japan, DOI: 10.1007/978-3-030-61105-7_11, October 28–30, 2020.

11. **Ermioni Qafzezi**, Kevin Bylykbashi, Phudit Ampririt, Makoto Ikeda, Leonard Barolli, Makoto Takizawa, "A Fuzzy-Based System for Assessment of Available Edge Computing Resources in a Cloud-Fog-Edge SDN-VANETs Architecture," Proc. of NBiS-2020, pp. 10–19, Victoria, Canada, DOI: 10.1007/978-3-030-57811-4_2, August 31 – September 2, 2020.

12. **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, Makoto Takizawa, "Resource Management in SDN-VANETs Using

Fuzzy Logic: Effect of Data Complexity on Coordination of Cloud-Fog-Edge Resources," Proc. of CISIS-2020, pp. 498–509, Lodz, Poland, DOI: 10.1007/978-3-030-50454-0_52, July 1–3, 2020.

International Conferences (Co-author)

1. Admir Barolli, Kevin Bylykbashi, Leonard Barolli, **Ermioni Qafzezi**, Shinji Sakamoto, Makoto Takizawa, "A Comparison Study of LDVM and RDVM Router Replacement Methods by WMN-PSODGA Hybrid Simulation System Considering Two Islands Distribution of Mesh Clients," Proc. of EIDWT-2023, Semarang, Indonesia, February 23-25, 2023.

2. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, Makoto Takizawa, "A Comparison Study of UNDX and UNDX-m Methods for LDVM and RDVM Router Replacement Methods by WMN-PSODGA Hybrid Intelligent System Considering Stadium Distribution," Proc. of BWCCA-2022, Tirana, Albania, October 27–29, 2022.

3. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Admir Barolli, Elis Kulla, Leonard Barolli, "Performance Comparison of Roulette Wheel and Random Selection Methods by WMN-PSODGA Simulation System Considering Stadium Distribution and LDIWM," Proc. of BWCCA -2022, Tirana, Albania, October 27–29, 2022.

4. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-based System for Handover Decision in 5G Wireless Networks: A Performance Evaluation for Two Fuzzy-based Systems," Proc. of BWCCA -2022, Tirana, Albania, October 27–29, 2022.

5. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, Makoto Takizawa, "Mesh Routers Placement by WMN-PSODGA Hybrid Intelligent System Considering Stadium Distribution and RDVM: A Comparison Study for Different Crossover Methods," Proc. of INCoS-2022, pp. 199-211, Hyogo, Japan, DOI: 10.1007/978-3-031-14627-5_19_18, September 7–9, 2022.

6. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Handover in 5G Wireless Networks Considering Different Network Slicing Constraints: Effects of Slice Load Parameter on Handover Decision," Proc. of NbiS-2022, pp. 152-162, Hyogo, Japan, DOI: 10.1007/978-3-031-14314-4_15_18, September 7–9, 2022.

7. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Admir Barolli, Elis Kulla, Leonard Barolli, "A Fuzzy-Based System for Determining Driver Impatience in VANETs Considering Number of Forced Stops, Unnecessary Maneuvers, Time Pressure and Task Importance," Proc. of NbiS-2022, pp. 163-171, Hyogo, Japan, DOI: 10.1007/978-3-031-14314-4_16_18, September 7–9, 2022.

8. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Elis Kulla, Leonard Barolli, "Unnecessary Maneuvers as a Determinant of Driver Impatience in VANETs: Implementation and Evaluation of a Fuzzy-based System," Proc. of CISIS-2022, pp. 1-9, Kitakyushu, Japan, DOI: 10.1007/978-3-031-08812-4_1, June 29–July 1, 2022.

9. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Handover in 5G Wireless Networks Considering Network Slicing Constraints," Proc. of CISIS-2022, pp. 180–189, Kitakyushu, Japan, DOI: 10.1007/978-3-031-08812-4_18, June 29–July 1, 2022.

10. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, Makoto Takizawa, "Mesh Routers Placement by WMN-PSODGA Simulation System Considering Stadium Distribution and RDVM: A Comparison Study for UNDX and UNDX-m Methods," Proc. of IMIS-2022, pp. 184–195, Kitakyushu, Japan, DOI: 10.1007/978-3-031-08819-3_18, June 29–July 1, 2022.

11. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based Scheme for Slice Priority Assessment in 5G Wireless Networks," Proc. of AINA-2022, Vol. 1, pp. 651-661, Sydney, Australia, DOI: 10.1007/978-3-030-99584-3_56, April 13–15, 2022.

12. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Determining Driver Stress in VANETs Considering Driving Experience and History," Proc. of AINA-2022, Vol. 2, pp. 1-9, Sydney, Australia, DOI: 10.1007/978-3-030-99587-4_1, April 13–15, 2022.

13. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Safe Driving in VANETs Considering Impact of Driver Impatience on Stress Feeling Level," Proc. of EIDWT-2022, pp. 236–244, Okayama, Japan, DOI: 10.1007/978-3-030-95903-6_-25, March 2–4, 2022.

14. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Slice Service Level Agreement in 5G Wireless Networks: Effect of Traffic Load Parameter," Proc. of EIDWT-2022, pp. 272–282, Okayama, Japan, DOI: 10.1007/978-3-030-95903-6_-29, March 2–4, 2022.

15. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for Deciding Driver Impatience in VANETs," Proc. of 3PGCIC-2021, pp. 129–137, Fukuoka, Japan, DOI: 10.1007/978-3-030-89899-1_13, October 28–30, 2021.

16. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "An Intelligent System for Admission Control in 5G Wireless Networks Considering Fuzzy Logic and SDNs: Effects of Service Level Agreement on Acceptance Decision," Proc. of 3PGCIC-2021, pp. 185–196, Fukuoka, Japan, DOI: 10.1007/978-3-030-89899-1_19, October 28–30, 2021.

17. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, "A Comparison Study of Chi-square and Uniform Distributions of Mesh Clients by WMN-PSODGA Simulation System for RIWM and LDIWM Router Replacement Methods," Proc. of BWCCA-2021, pp. 1-13, Fukuoka, Japan, DOI: 10.1007/978-3-030-90072-4_1, October 28–30, 2021.

18. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-Based System for User Service Level Agreement in 5G Wireless Networks," Proc. of INCoS-2021, pp. 96-106, Taichung, Taiwan, DOI: 10.1007/978-3-030-84910-8_10, September 1–3, 2021.

19. Admir Barolli, Kevin Bylykbashi, **Ermioni Qafzezi**, Shinji Sakamoto, Leonard Barolli, "A Hybrid Intelligent Simulation System for Node Placement in WMNs: A Comparison Study of Chi-Square and Uniform Distributions of Mesh Clients for CM and LDVM Router Replacement Methods," Proc. of NBiS-2021, pp. 117-130, Taichung, Taiwan, DOI: 10.1007/978-3-030-84913-9_11, September 1–3, 2021.

20. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A New Scheme for Slice Overloading Cost in 5G Wireless Networks Considering Fuzzy Logic," Proc. of CISIS-2021, pp. 110-120, Asan, Korea, DOI: 10.1007/978-3-030-79725-6_11, July 1–3, 2021.

21. Phudit Ampririt, **Ermioni Qafzezi**, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-based Scheme for Admission Control in

5G Wireless Networks: Improvement of Slice QoS Considering Slice Reliability as a New Parameter," Proc. of AINA-2021, Vol. 3, pp. 17-29, Toronto, Canada, DOI: 10.1007/978-3-030-75078-7_3, May 12–14, 2021.

22. Phudit Ampririt, Seiji Ohara, **Ermioni Qafzezi**, Makoto Ikeda, Leonard Barolli, Makoto Takizawa, "Effect of Slice Overloading Cost on Admission Control for 5G Wireless Networks: A Fuzzy-Based System and Its Performance Evaluation," Proc. of EIDWT-2021, pp. 24-35, Chiang Mai, Thailand, DOI: 10.1007/978-3-030-70639-5_3, February 25–27, 2021.

23. Kevin Bylykbashi, **Ermioni Qafzezi**, Makoto Ikeda, Keita Matsuo, Leonard Barolli, Makoto Takizawa, "Effect of Vehicle Technical Condition on Real-Time Driving Risk Management in VANETs," Proc. of EIDWT-2021, pp. 143-154, Chiang Mai, Thailand, DOI: 10.1007/978-3-030-70639-5_14, February 25–27, 2021.

24. Kevin Bylykbashi, **Ermioni Qafzezi**, Makoto Ikeda, Keita Matsuo, Leonard Barolli, Makoto Takizawa, "An Integrated Fuzzy-Based Simulation System for Driver Risk Management in VANETs Considering Relative Humidity as a New Parameter," Proc. of BWCCA-2020, pp. 233-243, Yonago, Japan, DOI: 10.1007/978-3-030-61108-8_23, October 28–30, 2020.

25. Phudit Ampririt, Seiji Ohara, **Ermioni Qafzezi**, Makoto Ikeda, Leonard Barolli, Makoto Takizawa, "Integration of Software-Defined Network and Fuzzy Logic Approaches for Admission Control in 5G Wireless Networks: A Fuzzy-Based Scheme for QoS Evaluation," Proc. of BWCCA-2020, pp. 386-396, Yonago, Japan, DOI: 10.1007/978-3-030-61108-8_38, October 28–30, 2020.

26. Kevin Bylykbashi, **Ermioni Qafzezi**, Phudit Ampririt, Keita Matsuo, Leonard Barolli, Makoto Takizawa, "An Integrated Fuzzy-Based Simulation System for Driving Risk Management in VANETs Considering Road Condition as a New Parameter," Proc. of INCoS-2020, pp. 15-25, Victoria, Canada, DOI: 10.1007/978-3-030-57796-4_2, August 31 – September 2, 2020.

27. Kevin Bylykbashi, **Ermioni Qafzezi**, Makoto Ikeda, Keita Matsuo, Leonard Barolli, Makoto Takizawa, "A Fuzzy-Based Simulation System for Driving Risk Management in VANETs Considering Weather Condition as a New Parameter," Proc. of IMIS-2020, pp. 23-32, Lodz, Poland, DOI: 10.1007/978-3-030-50399-4_3, July 1–3, 2020.