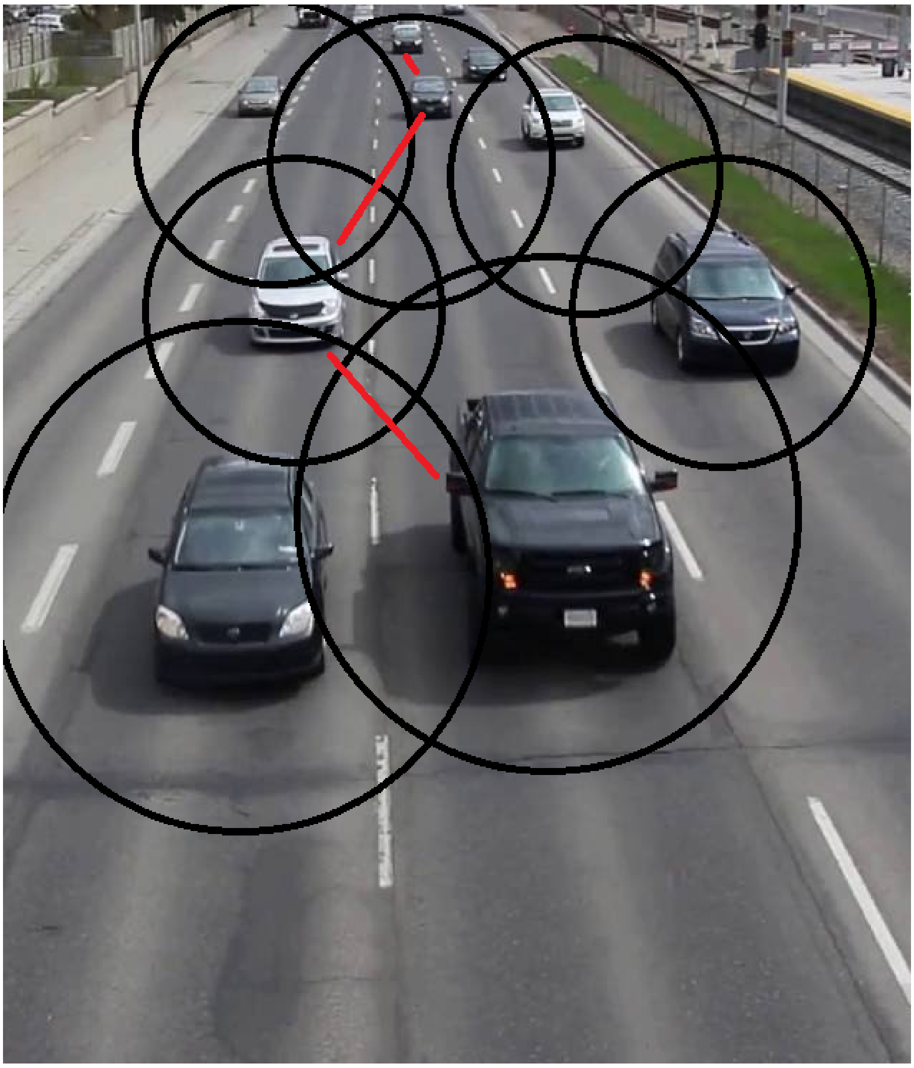
Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2023



Project Title: **Topology changes prediction for proactive location-based routing in VANETs**

Student: **Zhengyu Chen**

CID: **01736885**

Course: **EIE3**

Project Supervisor: **Dr Javier Barria**

Second Marker: **Dr Cong Ling**

Abstract

Vehicular Ad Hoc Networks (VANETs) have gained significant popularity as a network structure for communication among highly mobile vehicles. Numerous algorithms have been developed in the past for establishing communication links between vehicles in VANETs. Examples of such algorithms include Localised Power (LP), Power Progress (PP), Iterative Power Progress (IPP), Maximum Path Expiration Time (MPET), and Maximum Path Affinity (MPA). The first part of this project aims to provide a comprehensive review and establish the background related to VANET networks. It will examine the existing algorithms and their performance within VANET environments. In the second part of the project, the focus will be on analyzing and improving the MPA algorithm. This improvement will be based on incorporating the location and velocity information of nodes within the VANET environment. Additionally, a new algorithm called Automatically Selectable Maximum Path Affinity (ASMPA) will be designed. ASMPA combines the benefits of Dijkstra's algorithm and the greedy algorithm to enhance the performance and time delay of the MPA algorithm.

Overall, the project will contribute to the advancement of communication algorithms in VANETs by evaluating existing approaches, enhancing the MPA algorithm, and proposing a novel algorithm that combines different algorithmic benefits.

Acknowledgements

I would like to extend my heartfelt gratitude to all individuals who have played a significant role in the successful conclusion of this research endeavour. First and foremost, I am grateful to my advisor, Dr Javier Barria, for his unwavering guidance, expertise, and constant support throughout this endeavour. Their valuable insights and constructive feedback have been instrumental in shaping this research.

I would also like to acknowledge the invaluable contributions of my colleagues and peers who have supported me throughout this research journey. Their assistance, fruitful discussions, and shared ideas have broadened my understanding of the subject matter and enriched this study. I would like to express my heartfelt gratitude to Yang Pu for his invaluable guidance and support throughout the course of this research project.

I am grateful to Imperial College London for providing the necessary resources and opportunities that have facilitated this research. Their commitment to academic excellence has been an inspiration throughout my academic journey.

Lastly, I would like to express my gratitude towards my family and friends for their unwavering support and understanding throughout this research project. Their continuous encouragement, patience, and unwavering belief in my abilities have been instrumental in my achievements.

I want to emphasize that any oversights or omissions in this acknowledgement are unintentional, and the contributions of individuals not mentioned here are not overlooked but rather deeply appreciated.

Plagiarism Statement

I affirm that I have submitted, or will submit, an electronic copy of my final year project report to the provided EEE link.

I affirm that I have submitted, or will submit, an identical electronic copy of my final year project to the provided Blackboard module for Plagiarism checking.

I affirm that I have provided explicit references for all the material in my Final Report that is not authored by me, but is represented as my own work.

I have used ChatGPT as an aid in the preparation of my report. I have used it to improve the quality of my English throughout, however all technical content and references comes from my original text.

Contents

1 Introduction

1.1 Project Motivation

Since 1899, Italian famous physicist Guglielmo Marconi[1] established the world's first wireless station at South Foreland, England, for communicating with Wimereux in France, the wireless communication has been in development for over a century ago. Over the years, wireless communication has been integrated into people's lives along with continuous development. Wireless communication enabled the connection of billions of people to the Internet which significant improvement in quality of life. Also, the pervasive use of wireless technologies has permeated virtually every sector of the economy, playing a fundamental role in critical industries such as banking, agriculture, transportation, and healthcare[2]. According to recent data[3], the global count of individuals using mobile internet exceeded 4.2 billion in 2021. The economic worth generated by mobile technologies and services has already amounted to $4.5 trillion, which corresponds to approximately 5% of the worldwide gross domestic product (GDP). As more countries adopt mobile services, this figure is expected to rise by over $400 billion by 2025, bringing the total value to nearly $5 trillion. This growth is attributed to the increased productivity and efficiency resulting from the widespread use of mobile technology. The development of wireless communication has also brought certain benefits to transport as an area called Vehicular Ad hoc Network (VANET)[4].

The development of VANET stems from the frequent occurrence of traffic. Based on data from a reliable source[5], road traffic accidents are the primary cause of mortality among individuals aged 5-29 years. These incidents lead to approximately 1.3 million fatalities annually. Furthermore, these accidents impose a financial burden of around 3% of the gross domestic product on many countries. In Great Britain, there were still 137,013 reported casualties of all severities that happened in 2022 which is already a 12% decrease compared with 2019 because of the benefit from the improving transport policy[6]. Road accidents are the accidents that happen around our life, which need to be treated seriously

There are various intelligent transport systems that can potentially offer a solution to this issue. For example, Advanced driver assistance systems (ADAS) are essential components inside cars and trucks that aid drivers in maintaining safe driving practices. ADAS can assist drivers in maintaining safe speeds and distances, driving within their lane, avoiding critical overtaking situations, and safely navigating intersections in today's complex driving environment. These systems are crucial in ensuring safer driving practices and preventing accidents. Recent studies on these safety assistance systems have demonstrated their high potential for improving road safety. An example from a specific source[7] highlights the effectiveness of various intelligent transport systems in addressing the issue. According to eImpact's 2020 high scenario, Electronic Stability Control (ESC) is expected to have the most significant impact. It has the potential to prevent approximately 3,000 fatalities (-14%) and 50,000 injuries (-6%) annually. Other systems like Speed Alert (with active gas pedal) (-5%), eCall (-4%), and Lane Keeping Support (-3%) also demonstrate considerable effectiveness in reducing fatalities. According to the Regulation (EC) No 661/2009[8] implemented on 13 July 2009, there are certain mandatory safety features that motor vehicles must be equipped with. These include:

- Electronic Stability Control Systems, which became mandatory for all new types of vehicles starting from 1 Nov 2011 and for all new vehicles from 1 Nov 2014.
- Advanced Emergency Braking Systems and Lane Departure Warning Systems are required for heavy-duty vehicles, with the mandate starting from 1 Nov 2013 for new types of vehicles and 1 Nov 2015 for all new vehicles.

These measures will reduce fatal casualties in traffic by an estimated 5,000 per year[7].

Moreover, these systems also have the potential to reduce congestion on roads, as accidents are a leading cause of traffic congestion in Europe, accounting for approximately 15% of all congestion. The implementation of ADAS can thus help to improve traffic flow and reduce the overall societal and economic costs of road accidents. These days, traffic congestion plays a large part in societal and economic costs and people's quality of life. The United States experienced significant traffic congestion according to the 2018 Global Traffic Scorecard[10], which analyzed congestion levels in the top 60 urban areas. In cities like Boston, drivers lost approximately 164 hours due to congestion, while Washington D.C. experienced 155 hours of lost time. The financial impact of congestion was substantial, with Boston drivers losing up to $2,291 per year, followed by Washington D.C. ($2,161), Seattle ($1,932), Chicago ($1,920), and New York City ($1,859). This issue is not limited to the U.S., as the United Kingdom also faced a significant problem with congestion[11]. On average, British drivers wasted around 115 hours in congestion in a year, costing the country £6.9 billion in 2019, equivalent to an average of £894 per driver.

Vehicular ad-hoc network (VANET) are a type of wireless communication network that enables vehicle-to-vehicle and vehicle-to-infrastructure communications using wireless local area network technologies. VANET provide a specific range of potential applications[4, 9], including collision warnings and localized traffic information for drivers. Additionally, VANET possesses unique assets such as licensed spectrum and rechargeable power sources. VANET enables the exchange of real-time traffic information, such as vehicle speed, location, and road conditions, among vehicles in the network. This information can be utilized to enhance road safety by providing warnings about accidents, road hazards, or traffic congestion, allowing drivers to make informed decisions and take appropriate actions. VANET is also an integral part of Intelligent Transportation Systems, which aim to improve transportation efficiency, reduce traffic congestion, and enhance overall traffic management. By facilitating communication between vehicles, traffic infrastructure, and traffic management systems, VANET enables the implementation of various applications such as traffic signal optimization, adaptive traffic routing, and intelligent parking systems. Furthermore, VANET is essential for the development and deployment of cooperative driving systems and autonomous vehicles. Vehicles can exchange information about their speed, position, and intentions, enabling cooperative manoeuvres, such as cooperative merging and platooning. The environment in which VANETs operate, including vehicular traffic flow patterns and privacy concerns, also makes it a unique area of wireless communication.

Currently, VANETs are gaining increased attention due to advancements in related fields like low-cost GPS(global-positioning system) receivers and WLAN(wireless local area network) transceivers[9]. These developments contribute to making VANET systems more practical and valuable in enhancing road safety and improving transportation efficiency. VANET's contribution to safety and congestion is huge because traffic safety and congestion are interrelated areas, and if VANET can effectively improve traffic safety, then it will also have a good effect on traffic congestion.

VANET networks do have some limitations[4], one of which is the frequent and rapid changes in the network topology due to the high mobility of vehicles acting as nodes. This poses a challenge for VANET network algorithms, as they need to efficiently handle the dynamic nature of the network with minimal time delay and establish stable communication links. Therefore, this project focuses on addressing these challenges by improving communication link stability and reducing algorithmic time delays in VANETs.

1.2 Project Aim

The Aim of this project is to survey the background and research related to the VANET network, and also improve the performance of algorithms used in the previous project[12] within a VANET environment using data from NGSIM[14]. For comparison purposes, this project will use the similar simulation software MATLAB and data-set NGSIM. Also, the comparison between this project and the result from the last project will be collected and analyzed using EXCEL, which includes Average Path Lifetime and Percentage Path Availability. On the other side, the limitation and the future improvement of this project will be explored.

The core of this project is separated into three part, first part focus on analyzing the parameter of the algorithms, and see how we can improve the performance, delay and generality of the algorithm. The second part is about a new method 'Up-Edge Detection' used in the algorithms which can improve the accuracy of the algorithm. Third part design a new algorithm Automatically Selectable Maximum Path Affinity Algorithm(ASMPA) which combines the benefit of algorithms used in the previous project. The table below will list the project's aim in a more clear way.

	Project Aim
1	Survey the background and research related to the VANET network, and review it in the project report.
2	Review the Algorithms and Performance Metrics from the previous project.
3	Analyzing and suggesting the use of the Parameters in the algorithms.
4	Design a new calculation method 'Up-Edge Detection' used in algorithms.
5	Design new algorithm Automatically Selectable Maximum Path Affinity Algorithm(ASMPA)
6	Simulate and compare the performance result between previous algorithms and ASMPA.
7	Evaluate constraints on the project's efforts and suggest opportunities for enhancing them in the future.

2 Background

This section will provide background on Vehicular ad-hoc network(VANET) and mobile ad hoc network(MANET) and will cover the basics associated with them, including wireless network, ad hoc network, infrastructure mode network, single and multi-hop ad hoc networks, etc.

2.1 Mobile Ad Hoc Network(MANET)

2.1.1 Wireless Network

Wireless network[15] is a form of computer network that establishes connections between network nodes using wireless data connections instead of wired connections. Unlike wired networks, which rely on physical cables to connect devices, wireless networks use radio waves or infrared signals to transmit data. Wireless networking is commonly used in homes, businesses, and public places to provide access to the internet or to share files and other resources between devices.

Wireless networks are typically implemented using devices such as wireless routers, access points, and wireless network adapters. These devices use radio waves to communicate with each other and establish connections between devices. Wireless networks are often more flexible and easier to install than wired networks since they do not require the installation of physical cables.

Some common types of wireless networks include cellular networks, Wi-Fi networks, Bluetooth networks, and satellite networks. Each of these wireless network types has its unique characteristics and is designed for specific applications.

2.1.2 Benefits and drawbacks of utilizing Wireless Networks

Wired networks have several advantages[17] including faster data transfer speeds in comparison to wireless networks and increased immunity to interference. They also provide enhanced security since unauthorized access to data is more challenging. However, wired networks can be costlier to install or modify, and relocating devices may pose challenges due to limited network connectivity in different locations.

Wireless networks offer several benefits to users, such as the ability to add new nodes easily without interrupting the network and the freedom to move around while staying connected. However, they have some drawbacks, including limited range, electromagnetic interference, and signal blockage. Bandwidth limitations can arise in wired networks as each access point has a finite amount of bandwidth to distribute among connected nodes. Consequently, communication may slow down when more nodes are connected. Moreover, the use of radio signals in wireless networks introduces a security vulnerability, as unauthorized users can intercept these signals. To mitigate this issue, message encryption is necessary to ensure secure communication.

The convenience and simplicity of wireless networking have resulted in its widespread adoption for connecting nodes to networks. Devices like laptops, tablets, smartphones, interactive TVs, media centres, game consoles, and security cameras can effortlessly connect to a network without the need for individual cables for each device.

Wireless networks offer the distinct advantage of unrestricted mobility, making them highly popular in various environments such as homes, schools, and organizations where the number of connected nodes fluctuates. For the purpose of this project, only wireless networks will be taken into account for vehicular communication.

2.1.3 Wireless Ad Hoc Network

A wireless ad hoc network (WANET)[16] or mobile ad hoc network (MANET) is a type of wireless network that functions in a decentralized manner, without the need for any pre-existing infrastructure such as routers or wireless access points. In an ad hoc network, each node is responsible for routing data for other nodes, each wireless device serves as both a node and a router, enabling data to be transmitted between devices in a peer-to-peer fashion, which enables the network to be highly flexible and adaptable to changing conditions. The decision regarding which nodes are responsible for forwarding data is made dynamically, taking into consideration factors such as network connectivity and the specific routing algorithm employed.

In a MANET, every device has the freedom to move independently in any direction, resulting in frequent changes to its connections with other devices. These wireless networks are advantageous in that they eliminate the need for complex infrastructure setup and administration, allowing devices to form and join networks spontaneously. MANET can operate autonomously or be connected to the Internet, with various transceivers facilitating communication between nodes. This dynamic and self-governing nature gives rise to a network structure where devices can establish connections and communicate on the go. Additionally, MANET typically employs a network routing system built on top of a link layer ad hoc network.

2.1.4 Infrastructure and Ad-Hoc Network

In the majority of Wi-Fi networks, an infrastructure network configuration is utilized. This means that all devices within the network communicate through a single access point, typically a wireless router. Even if two laptops are physically close to each other, they still communicate indirectly by connecting through the access point. In this mode, a central access point is necessary for all devices to connect to.

In contrast, an ad-hoc network, also known as a peer-to-peer network, operates without the need for a centralized access point. Devices within the wireless network can connect directly to each other, bypassing the requirement for a central access point. For instance, if two laptops are configured in ad-hoc wireless mode, they can establish a direct connection without relying on a centralized access point. While the infrastructure mode is suitable for most Wi-Fi networks, the ad-hoc mode is beneficial in situations where a centralized access point is unavailable or impractical, such as in disaster relief scenarios or remote locations with limited infrastructure.

2.1.5 Advantage and Disadvantage of Infrastructure and Ad-Hoc Network

Ad-hoc network can be useful[18] when you need to connect two devices directly without a centralized access point. In such cases, an ad-hoc network enables the creation of a temporary Wi-Fi network without the need for a router. The emergence of the Wi-Fi Direct standard further builds upon the ad-hoc mode, allowing devices to communicate directly through Wi-Fi signals.

On the other hand, infrastructure networks are more suitable for establishing long-term or permanent networks. Wireless routers, which act as access points in infrastructure networks, are equipped with higher-powered wireless radios and antennas that enable them to provide coverage over larger areas. On the other hand, the wireless radio of a laptop, for example, may not possess the same level of strength as a router, resulting in a limited range for the network.

One disadvantage of ad-hoc networks is the increased demand for system resources due to the dynamic nature of the network layout as devices move. In contrast, the access point in an infrastructure network typically remains stationary. Moreover, when numerous devices are connected to an ad-hoc network, there can be more wireless interference because each computer must establish direct connections with every other computer instead of utilizing a single access point. If a device is out of the range of another device it wishes to connect to, the data must be relayed through other devices, potentially slowing down the network. Additionally, ad-hoc networks do not scale well, making them unsuitable for larger networks.

However, in this project, we are simulating a scenario that involves vehicular communication, which is a situation where ad-hoc networks demonstrate exceptional efficiency. This specific scenario showcases two key advantages of ad-hoc networks.[19]:

- Mobility
  - Ad hoc networks can be easily established in almost any situation that involves multiple wireless devices, allowing for their creation on the spot.
- Speed
  - Setting up an ad hoc network requires minimal configuration adjustments and does not necessitate the use of additional hardware or software. When there is a need to swiftly and effortlessly connect multiple computers, an ad hoc network proves to be an excellent solution.

2.1.6 Single and Multi-hop Ad Hoc Networks

Ad hoc networks can be further classified into single-hop and multi-hop networks[20] based on how the nodes communicate with each other.

In a single-hop ad hoc network, all nodes are within the transmission range of each other, so they can communicate directly with each other without the need for any intermediate nodes.

On the other hand, in a multi-hop ad hoc network, nodes communicate with each other through intermediate nodes, also known as relays. In this case, nodes that are not within each other's transmission range can communicate by forwarding their packets through other nodes. This means that nodes in a multi-hop network can be located at different levels, creating a hierarchical network topology.

Multi-hop ad hoc networks have some advantages over single-hop networks. For instance, they can cover a larger geographical area and can support more nodes since the transmission range of each node is extended through relays. They are also more resilient to node failure since there are multiple paths for data transmission. In certain cases[21], multi-hop routing can be more energy-efficient compared to single-hop routing because the transceiver, which is the primary power consumer in a radio node, requires high power for long-distance transmission. Nevertheless, there are certain drawbacks associated with multi-hop ad hoc networks, including increased latency caused by the additional hop and the requirement for efficient routing protocols to handle network topology. The research on multi-hop ad hoc networks and their security remains an active area of study. Conversely, single-hop ad hoc networks are currently being utilized in commercial applications, with Bluetooth emerging as one of the most widely adopted single-hop ad hoc wireless standards.

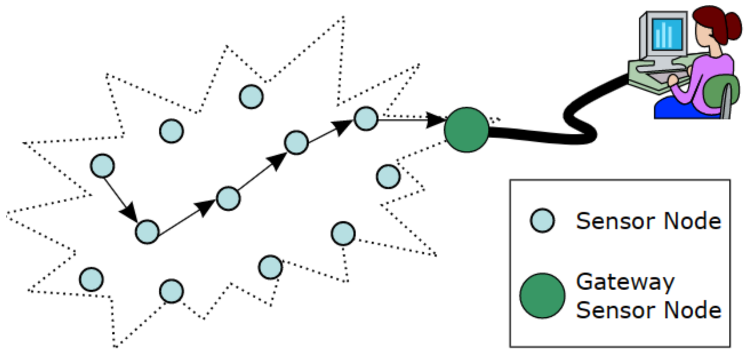


Figure 1: Multi-hop ad hoc network[21]

2.1.7 Applications of MANET

Ad hoc networks find diverse applications in industries and businesses that necessitate collaborative mobile data exchange. In military environments, there is a demand for robust and IP-compliant data services in mobile wireless communication networks, which often comprise dynamically changing topology segments. The advanced capabilities of mobile ad hoc networks, including support for multimedia data rates, global roaming capabilities, and integration with other network structures, facilitate the emergence of novel applications. Some of the common applications[22, 23] are listed below:

- Military and defence applications:
  - Ad hoc networks are particularly well-suited for military and defence applications that prioritize rapid communication deployment and resilient, IP-compliant data services within mobile wireless communication networks.
- Crisis management:
  - Ad hoc networks can restore communication quickly in the aftermath of natural disasters, when traditional communication infrastructure may be disrupted.
- Commercial and civilian Environment

- Mobile ad hoc networks are used in commercial and civilian environments for various purposes such as facilitating e-commerce transactions anytime and anywhere, providing road and weather information, enabling inter-vehicle communication and taxi cab networks, guiding sports stadium visitors, and serving as networks for visitors at airports, trade fairs, and shopping malls.

- Body Area Networks (BANs)
  - A health monitoring system can be created by a wireless network of medical sensors placed on or around a human body.
- Telemedicine:
  - Ad hoc networks can facilitate telemedicine applications, allowing paramedics and medical personnel in remote locations to access medical records and seek emergency assistance from surgeons.
  - The medical sensors can form a BAN (Body Area Network) which is used in telemedicine systems.
- Tele-geoprocessing:
  - The integration of GPS (Global Positioning System), GIS (Geographic Information System), and high-capacity wireless mobile systems allows for tele-geoprocessing, which involves the real-time processing and analysis of geographic data.
- Virtual navigation:
  - Ad hoc networks can be used to provide access to graphical representations of buildings and streets, including emergency rescue plans and points of interest, through virtual navigation.
- Wireless Sensor Networks (WSNs)
  - This technology is employed to establish connections between multiple cost-effective and energy-efficient sensor devices that are either deployed in the environment or attached to animals. These devices are commonly integrated into various structures such as buildings, bridges, and streets, as well as onto animals and mountains, with the aim of monitoring the environment, industries, and observing events and phenomena in general.
- Wireless Personal Area Networks (PANs)
  - This technology facilitates the sharing of information among nearby electronic devices such as cameras, storage devices, televisions, mobile phones, or laptops through a self-contained network within a home.
- Education:
  - Ad hoc networks can be utilized to offer educational opportunities in remote regions where conventional internet access may be economically impractical.
- Vehicular area networks:
  - Ad hoc networks can provide emergency services and other information through vehicular area networks, which are equally effective in both urban and rural settings.
  - Wireless interfaces are incorporated into vehicles to facilitate communication between them (V2V) as well as with fixed infrastructure located along the roadsides (V2I).
  - V2V communication allows vehicles to engage in coordination platforms and facilitate the transmission of other communications.
  - V2I connectivity enables vehicles to receive crucial information regarding road conditions, traffic congestion, or alerts regarding accidents.
  - The IEEE 1609 standards establish a standardized framework, services, and interfaces for ensuring secure wireless communications in both V2V and V2I environments.
  - The main goal of planning for InVANETs is to prevent collisions among vehicles and ensure the safety of passengers.
  - Vehicles establish a mobile ad hoc network using wireless technologies such as WiMax IEEE802.16 and WiFi 802.11 for communication purposes.

2.2 Routing Protocol for MANET

The following will show the algorithms I investigated for MANET, which can also be found in [24]. They mostly can be separated into three parts: Pro-active routing protocols, Reactive routing protocols, and Hybrid Routing protocol.

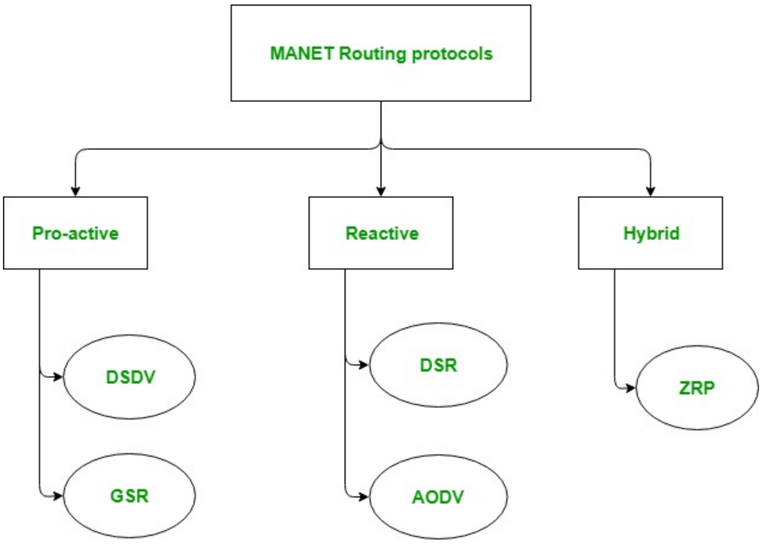


Figure 2: Routing Protocol for MANET[24]

2.2.1 Pro-active routing protocols

Also known as table-driven or link state routing protocols, these routing protocols involve each mobile node maintaining its own routing table, which stores information about potential routes to other mobile nodes. Due to the dynamic nature of mobile ad-hoc network topology, these routing tables are updated regularly. However, in larger networks, these protocols may not be ideal as the routing tables can become excessively large, requiring them to contain information about all possible nodes.

- Destination Sequenced Distance Vector Routing Protocol (DSDV)
  - DSDV, a proactive or table-driven routing protocol, was specifically designed for mobile ad-hoc networks and is based on the Bellman-Ford algorithm[25]. It extends the distance vector routing protocol used in wired networks. The count-to-infinity problem posed a challenge for distance vector routing protocols in mobile ad-hoc networks. DSDV addresses this issue by introducing a destination sequence number, which is assigned to each routing entry in the routing table maintained by every node. When a new update with a higher sequence number for a destination is received, the node updates its routing table with the new route.
- The Global State Routing Protocol (GSR)
  - GSR is a proactive or table-driven routing protocol that builds upon the link state routing protocol used in wired networks. It utilizes Dijkstra's algorithm[26] for routing decisions. However, the traditional link-state routing protocol is not suitable for mobile ad-hoc networks due to the global flooding problem, where each node floods the link-state information throughout the network, leading to congestion of control packets. GSR addresses this issue by avoiding the global flooding of link state packets. Instead, each mobile node in GSR maintains essential data structures such as an adjacency list, topology table, next hop table, and distance table to facilitate efficient routing within the network.

2.2.2 Reactive routing protocols

Also called on-demand routing protocols, do not maintain a continuous record of the network topology. Instead, routes are only discovered and established when they are needed. This process involves broadcasting route request packets across the network to find a suitable route. Reactive routing consists of two primary phases: route discovery, where a route is actively searched for, and route maintenance, where the established route is maintained and kept operational.

- Dynamic Source Routing protocol (DSR)
  - DSR is a reactive/on-demand routing protocol that is employed when a route is needed for communication. In this type of routing, route request packets are flooded throughout the mobile network to discover an appropriate route. DSR consists of two essential phases: Route Discovery and Route Maintenance. During Route Discovery, the most efficient path between the source and destination mobile nodes is identified. Route Maintenance ensures the proper functioning of the network and manages situations where links break, potentially causing network failures due to the dynamic nature of mobile ad-hoc networks.
- Ad-Hoc On-Demand Vector Routing protocol (AODV)
  - AODV is an on-demand/reactive routing protocol that offers improvements over the DSR protocol. In DSR, the complete path from the source to the destination is included in the data packet header, leading to increased header size and slower performance as the network size grows. AODV addresses this concern by storing the path information in the routing table instead. Similar to DSR, AODV operates in two phases: Route Discovery and Route Maintenance. During the Route Discovery phase, it determines the most efficient path for transmitting data between the source and destination nodes. The Route Maintenance phase ensures the proper functioning of the established route by handling link breakages in the dynamic network.

2.2.3 Hybrid Routing protocol

A Hybrid Routing protocol combines reactive and proactive routing protocols and adjusts its behavior based on the location of the source and destination mobile nodes. One well-known example of a hybrid routing protocol is the Zone Routing Protocol (ZRP). In ZRP, the network is divided into zones, and the positions of the source and destination nodes are considered. If both nodes are within the same zone, proactive routing is employed to transmit data packets. However, if the nodes are located in different zones, reactive routing is utilized for data packet transmission.

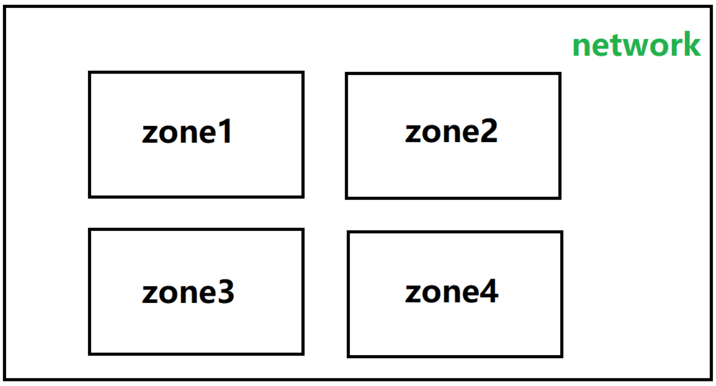


Figure 3: Zone Routing Protocol (ZRP)[24]

2.3 Challenges and Issues of MANET

MANET has been used in many applications and there has been a lot of research on the subject, but there are still many challenges and issues[23] in practice, which I will discuss below.

- Limitation Bandwidth
  - Wireless networks still have much lower capacity compared to wired networks. Moreover, the actual data transfer speed in wireless communication is often much lower than the theoretical maximum due to factors such as multiple access, fading, noise, and interference.
- Routing
  - In MANET, the nodes constantly move and the network topology changes frequently, making it prone to generating outdated routes in the routing table. This can cause unnecessary routing overhead. To address this issue, an efficient and intelligent routing protocol is needed to handle the highly dynamic and ever-changing network conditions.
- Battery Constraints
  - The devices used in such networks have power limitations to maintain their portability, lightweight nature, and compact size. However, in the context of this project (VANET), the power and computing constraints for vehicles are not as stringent as those for small, handheld mobile devices. This is because vehicles have larger batteries that can be recharged by the engine.
- Security
  - The mobile and wireless nature of MANETs introduces novel security issues for network design. Since wireless transmission is susceptible to eavesdropping and the functionality of ad hoc networks relies on node cooperation, MANETs are inherently vulnerable to a range of security attacks.
- Nodes Update
  - To facilitate the automatic selection of the optimal route, there is a need for dynamic updates to identify newly arrived nodes and inform the network about their presence.
- TCP/UDP
  - TCP and UDP are the established protocols employed in the Internet for communication. In the context of MANET, applications such as HTTP and real-time audio/video necessitate the use of transport layer protocols like TCP and UDP to facilitate the transmission of packets over the network links.
- IP Address
  - A crucial concern in MANETs is the IP addresses assigned to the network, which has led to a significant focus on IP addressing and address auto-configuration.
- Radio Interface
  - In MANETs, the transmission of packets between mobile nodes is carried out through the radio interface or antenna, which makes it important to study techniques related to packet forwarding and receiving.
- Packet Loss
  - Packet loss is a common occurrence in Ad hoc wireless networks and is caused by a variety of factors such as the presence of hidden terminals, interference, uni-directional links, and frequent changes in the network topology due to node mobility. This leads to a higher rate of packet loss compared to other types of networks.
- Robustness and Reliability
  - Nodes that exhibit improper behaviour and unreliable links can have a significant impact on the overall performance of the network. Since there are no centralized monitoring and management mechanisms, it is difficult to quickly detect and isolate such misbehaviour. This leads to a significant increase in the complexity of network design.

2.4 Vehicular Ad Hoc Networks (VANET)

VANET stands for Vehicular Ad-Hoc Network. It is a variant of a mobile ad-hoc network (MANET) specifically designed for vehicles. It consists of a group of vehicles that establish wireless communication among themselves.

The main purpose of VANET is to enable communication between vehicles and infrastructure components, such as traffic lights, road signs, collision avoidance and other road infrastructure, in order to improve road safety, traffic efficiency, and passenger comfort.

VANETs have a unique set of characteristics that make them different from other ad-hoc networks, including high mobility of nodes, varying network topology, limited communication range, and changing network conditions due to environmental factors such as terrain, weather, and other vehicles on the road. These characteristics pose several challenges to designing efficient and reliable routing protocols and security mechanisms for VANETs.

2.4.1 Components of VANET

VANET is a self-organizing wireless network that consists of three main components. The first component is the vehicles themselves, which serve as the network nodes and enable wireless communication between them (V2V) as well as with infrastructure access points (V2I). The second component is the infrastructure, which includes roadside base stations strategically placed at specific locations such as junctions or parking areas. These base stations extend the communication range of the ad hoc network and support safety applications like low bridge warnings and accident alerts. The third component is the communication channels, which utilize radio waves, a form of electromagnetic radiation with longer wavelengths than infrared light. The frequencies of these radio waves range from 190 GHz to 3 kHz. The radio propagation model used plays a crucial role in the performance of the network protocols and determines the number of nodes that can be part of a single collision domain.

2.4.2 Characteristics of VANET[4]

- High Mobility
  - In VANET, vehicles have the capability to travel at high speeds and exhibit frequent changes in their positions and directions.
- Dynamic Network Topology
  - The topology of VANET is highly dynamic, as vehicles move in and out of communication range of each other.
- Heterogeneous Networks
  - VANETs are heterogeneous networks composed of vehicles with different communication devices, and their communication links may vary in terms of quality and reliability.
- Ad-hoc Connectivity
  - The connectivity between vehicles in VANET is established on an ad-hoc basis, i.e., without relying on a pre-existing infrastructure.
- No Power constraints
  - While power constraints are present in many networks, VANETs have the advantage that vehicles can supply power to the on-board unit (OBU) using long-life batteries. Therefore, energy constraints are not always a significant challenge as in MANETs.
- High Computability Ability:
  - The presence of computational resources and sensors enhances the computational capacity of the nodes.
- Variable Network Density
  - The density of the VANET network varies based on the density of traffic, with high density in congested areas and low density in suburban areas.

2.5 Difference between VANET and MANET

Node Characteristics: In MANET, the nodes are typically mobile devices, such as smartphones, laptops, or tablets, that can communicate with each other using wireless links. In VANET, the nodes are vehicles equipped with communication devices that can establish a wireless communication network with other vehicles and infrastructure.

Mobility: MANET nodes can move freely in any direction, and their movement is not constrained to a particular environment. In contrast, VANET nodes are constrained to a road network and move along predefined routes. The mobility of vehicles in VANET is highly dynamic and can change rapidly.

Network Topology: The topology of MANET changes frequently due to the mobility of the nodes, and the network is typically characterized by a large number of nodes with low density. In contrast, the topology of VANET changes less frequently due to the constrained movement of the vehicles, and the network is typically characterized by a lower number of nodes with high density in specific areas.

Communication Range: In both MANET and VANET, the communication range is constrained by factors such as the power of wireless devices and the surrounding environment. However, in VANET, the communication range is further influenced by additional factors like vehicle speed and road conditions.

Applications: MANET is used in a variety of applications, such as military operations, disaster response, and mobile communication.VANET is predominantly utilized in transportation applications, particularly in Intelligent Transportation Systems (ITS), to enhance traffic flow, alleviate congestion, and bolster road safety through effective communication between vehicles and roadside infrastructure. These networks provide drivers with real-time traffic updates and navigation information, expedite emergency services by enabling swift message transmission, facilitate vehicle-to-vehicle and vehicle-to-infrastructure communication for cooperative driving and platooning, and offer entertainment and infotainment services to passengers. In summary, VANET aim to improve driving experiences, increase road safety, and facilitate various intelligent transportation applications.

Routing Protocols: The routing protocols used in MANET and VANET are different due to the unique characteristics of each network. MANET routing protocols focus on finding the most efficient path to the destination based on the network topology, while VANET routing protocols take into account additional factors such as vehicle mobility, road conditions, and safety.

The table below will clearly show the difference between VANET and MANET[27].

No.	MANET	VANET
1	The reorientation of network topology occurs at a slow pace.	Network topology changes frequently and rapidly.
2	The production cost of MANET is relatively lower in comparison to VANET.	Significantly higher.
3	The density of nodes is low or sparse.	The density of nodes varies frequently.
4	The limited mobility of MANET creates difficulties for network providers in determining the exact location of a mobile subscriber.	High mobility makes it easier for serving networks to locate the position of a mobile subscriber.
5	The range of MANET extends up to 100 meters.	The range of VANET extends up to 500 meters.
6	The lifetime of nodes in MANET is determined by their power resources.	The lifetime of nodes in VANET is dependent on the lifetime of the vehicle.
7	An addressing scheme based on attributes is used.	An addressing scheme based on location is used.
8	MANET provides a bandwidth of 100 Kbps.	VANET offers a bandwidth of 1000 Kbps.
9	Availability of Multi-hop Routing	Weakly available of Multi-hop Routing.
10	The movement of nodes has an impact on the functionality of MANETs, as they require resilient routing protocols to handle node mobility. MANETs are characterized by random node movements.	Regular, moving pattern of nodes.
11	Ultrasonic technology is utilized to acquire the position.	Ultrasonic technology is utilized to acquire the position.
12	High reliability of VANET.	VANET has a high level of reliability.

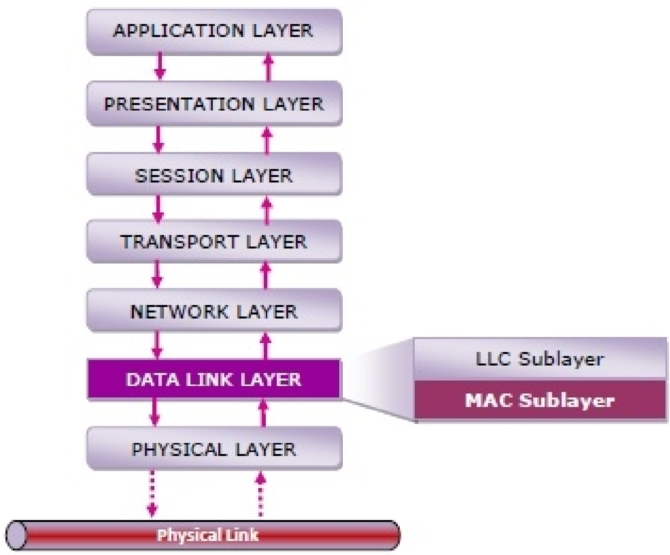
Table 1: Extracted from [27]

2.6 MAC Protocols for VANET

2.6.1 MAC Protocols

The MAC protocol is a crucial element of the data link layer in the OSI reference model, responsible for managing data transmission. It manages the flow of data and enables multiple data transmissions on shared communication channels. The MAC protocol controls the transfer of data packets through network interface cards.

The MAC layer plays a vital role within the OSI network architecture as it acts as an intermediary between the physical layer and higher layers of the network. It is responsible for preparing data frames for transmission through the physical medium by encapsulating them. The MAC layer also handles the addressing of source and destination stations or groups of destination stations. When multiple data frames need to be transmitted, the MAC layer performs multiple access resolutions and decides on channel access methods. In case of collisions, it performs collision resolution and initiates retransmission. Additionally, the MAC layer generates frame check sequences to ensure data transmission accuracy. The figure in the right shows the position of the MAC layer in the OSI Model[28].



2.6.2 MAC Protocols used in VANET

In VANET, the MAC (Media Access Control) protocol is responsible for managing access to the wireless communication channel. Its main function is to ensure that multiple vehicles in the network can share the wireless channel efficiently and fairly.

In VANETs, there are multiple MAC protocols utilized[29], among which the IEEE 802.11p protocol is the prevalent choice for vehicular communications. This protocol employs a contention-based method for accessing the wireless channel, where vehicles contend for access by transmitting data request signals.

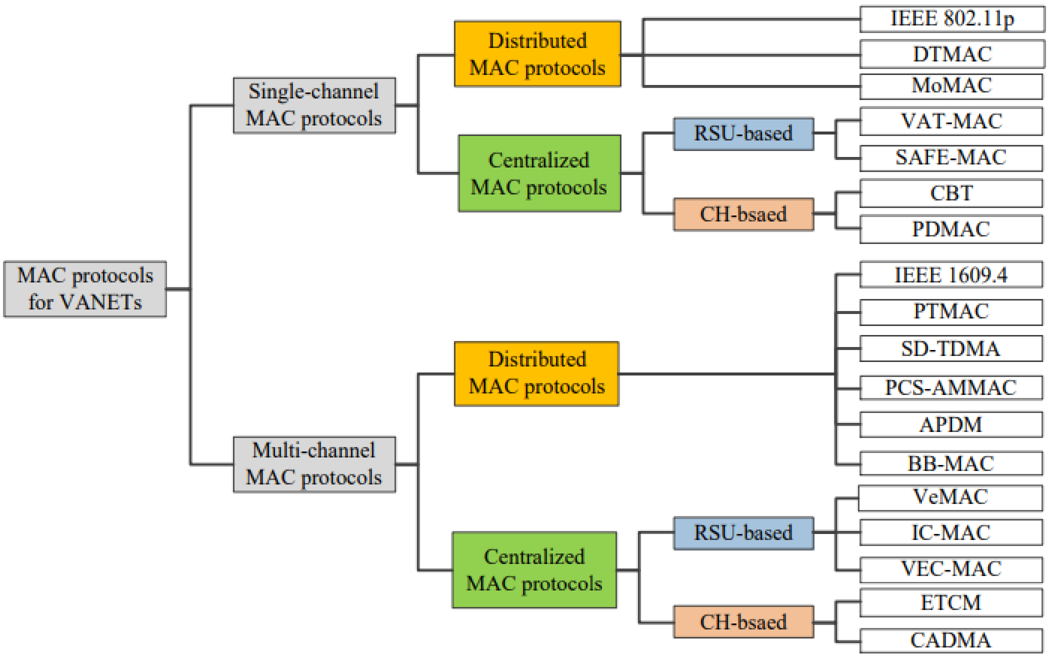


Figure 4: MAC Protocols for VANET[29]

- IEEE 802.11p
  - The IEEE 802.11p standard is a wireless communication protocol developed specifically for vehicular environments. It is an extension of the widely used IEEE 802.11 standard, commonly known as Wi-Fi, which is primarily used for wireless local area networks (WLANs). IEEE 802.11p operates in the 5.9 GHz frequency band and facilitates communication between vehicles (V2V) and between vehicles and roadside infrastructure (V2I).

    The IEEE 802.11p standard defines the physical (PHY) and media access control (MAC) layers for wireless communication in vehicular environments. The PHY layer specifies the modulation and coding schemes, as well as the signal bandwidth, while the MAC layer defines the access mechanisms for the wireless medium.

    The MAC protocol specified by IEEE 802.11p employs a contention-based mechanism for accessing the wireless channel in VANETs. In this approach, when a vehicle intends to transmit data, it first examines the channel to determine if it is unoccupied. If the channel is found to be busy, the vehicle will defer its transmission and select a random time interval before attempting again. On the other hand, if the channel is determined to be idle, the vehicle initiates the transmission process by sending a request to transmit (RTS) packet, which contains information about the size of the data packet to be transmitted.

    If the receiving vehicle is ready to receive the data, it sends a clear-to-send (CTS) packet back to the transmitting vehicle, which reserves the channel for the duration of the transmission. Once the transmission is complete, the receiving vehicle sends an acknowledgement (ACK) packet to the transmitting vehicle to confirm that the data was received successfully.

In addition to the basic access mechanism, the IEEE 802.11p MAC protocol also includes several other features that are specific to vehicular environments. These include a priority scheme for emergency messages, a mechanism for beaconing to exchange information between vehicles and roadside infrastructure, and a mechanism for authentication and security.

- IEEE 1609.4
  - IEEE 1609.4 is a standard that specifies the architecture and protocol requirements for enabling secure communication among multiple wireless access technologies in vehicular environments. It defines a standard interface between the upper layers of a protocol stack and the lower layers of the communication system. The primary goal of IEEE 1609.4 is to enable seamless communication between different wireless networks that may have different data rates, transmission ranges, and Quality of Service (QoS) requirements.

    One of the main reasons why IEEE 1609.4 was developed even though IEEE 802.11p already existed is because IEEE 802.11p alone could not provide the required level of QoS and reliability for Vehicular Ad-hoc Networks (VANETs). IEEE 1609.4 extends the capabilities of IEEE 802.11p by providing additional functionality and protocols that address the specific needs of VANETs. For example, IEEE 1609.4 provides QoS mechanisms for prioritizing different types of traffic, security mechanisms for protecting against cyber attacks, and network discovery mechanisms for identifying neighbouring nodes.

    The main difference between IEEE 802.11p and IEEE 1609.4 is that IEEE 802.11p is a physical layer protocol that defines the transmission of data over the wireless medium, while IEEE 1609.4 is a higher-layer protocol that defines the communication between different networks and the coordination of multiple access technologies. In other words, IEEE 802.11p is responsible for the wireless transmission of data packets, while IEEE 1609.4 is responsible for ensuring that the data packets are transmitted securely and reliably between different networks. Additionally, IEEE 1609.4 provides additional functionality such as network discovery, session management, and security mechanisms that are not provided by IEEE 802.11p alone.

2.7 Greedy Algorithm and Dijkstra Algorithm

For calculating and selecting the next hop as mostly VANET is a situation of multi-hop network, the most commonly used basic algorithm is the Greedy algorithm and Dijkstra algorithm[26, 30], which is also considered as the most important part in this project, following will introduction this two algorithm in detail and explain how they work.

2.7.1 Greedy Algorithm

In a Vehicular Ad-hoc Network (VANET), a greedy algorithm is a type of routing algorithm that makes decisions based on local information, without considering the global network state. The goal of a greedy algorithm is to find the most optimal path between a source and a destination by selecting the next hop that is closest to the destination from the current node.

The basic principle of the greedy algorithm[30] is to select the neighbour vehicle that is closest to the destination as the next hop, and forward the message to that vehicle. This process continues until the message reaches its destination. Therefore, the greedy algorithm always chooses the best choice as the current node. The algorithm does not take into account the distance between the source and the next hop, or any other potential paths that may exist in the whole global network.

The advantage of using a greedy algorithm in VANETs is its simplicity and low computational cost. As the algorithm only considers local information, it can quickly make decisions without requiring global network knowledge. However, the disadvantage of using a greedy algorithm is that it may not always find the most optimal path, especially in situations where there are obstacles or congested areas that prevent direct communication between the source and destination.

To overcome the limitations of the greedy algorithm, other routing algorithms, such as the A* algorithm or the Dijkstra algorithm[38], can be used. These algorithms take into account the entire network topology and attempt to find the most optimal path based on various metrics, such as distance, bandwidth, or delay.

Following is an example of how the Greedy Algorithm accurately work.

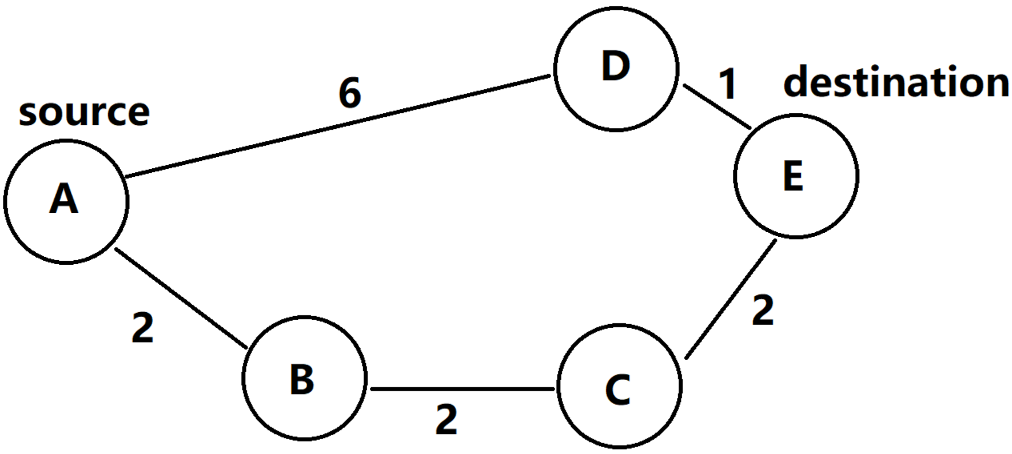


Figure 5: Example of Greedy Algorithm

From source node A, the greedy algorithm will choose the longest path from the current node(A). Therefore, node D is chosen because path length 6 is longer than the path from A to B(path length 2). Now the current node is D, and the only choice left is node E. Finally, using the greedy algorithm we have the longest path from source to destination is A-D-E, and the path length is 6+1=7.

In this case, the greedy algorithm works pretty well, it does find the longest path, but in some other cases, the greedy algorithm doesn't always produce the optimal solution. The following will show another example of how the greedy don't produce the optimal solution

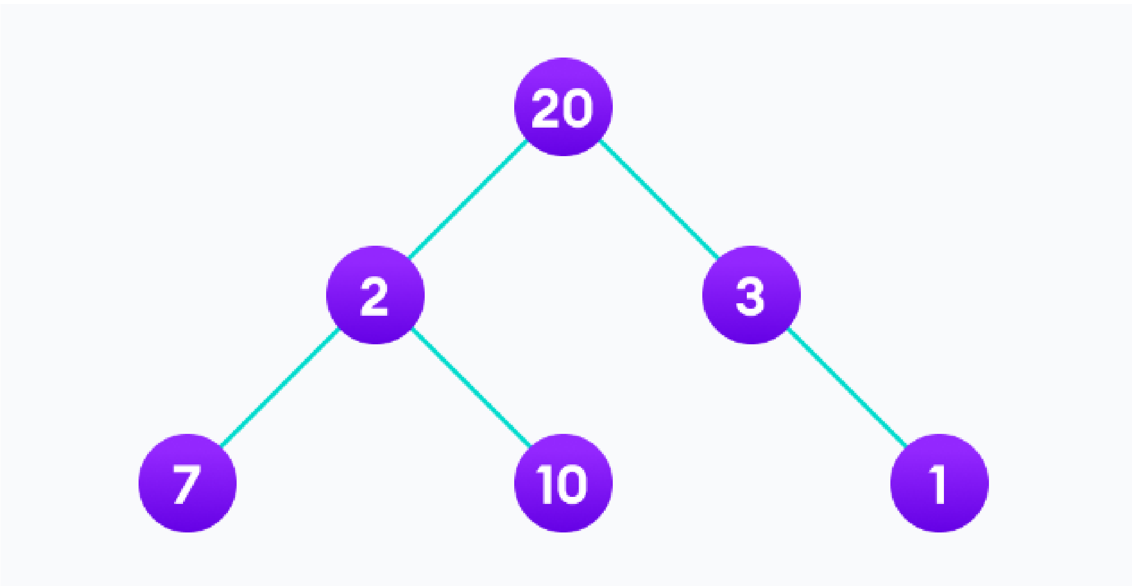


Figure 6: Example2 of Greedy Algorithm[30]

- The starting node is 20 and it has a right child with a weight of 3 and a left child with a weight of 2.
- The goal is to find the largest path, and the current best solution is 3, so the greedy algorithm selects 3.
- However, the final result of 20 + 3 + 1 = 24 is not optimal because there exists another path with a higher weight of 20 + 2 + 10 = 32.
- The optimal path is not chosen because the greedy algorithm only considers the immediate next step and does not take into account other potential paths.

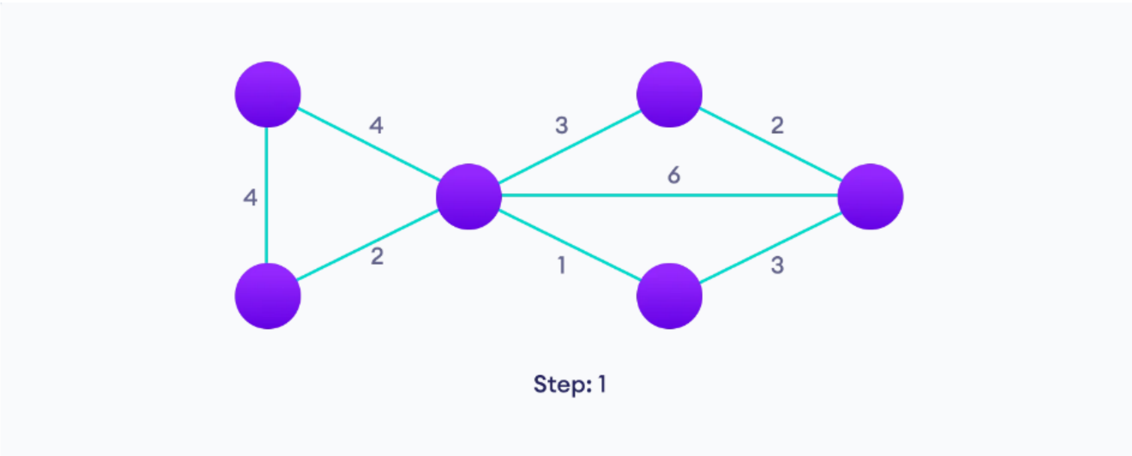
2.7.2 Dijkstra Algorithm

The Dijkstra algorithm[26], a popular routing algorithm employed in Vehicular Ad-hoc Networks (VANETs), is utilized to determine the shortest path between a source and a destination node. This algorithm constructs a shortest-path tree with the source node as the root and progressively updates the tree to incorporate neighbouring nodes that offer the shortest distances from the source.

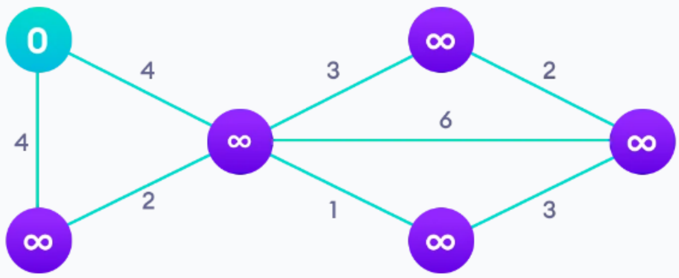
The Dijkstra Algorithm operates as follows:

- Initialization: The algorithm initializes a set of unvisited nodes, distances from the source node to all other nodes, and a set of preceding nodes. The distance from the source node is set to 0, while the distances to all other nodes are initially set to infinity. The preceding nodes are initially set to NULL.
- Selection: The algorithm chooses the unvisited node with the minimum distance from the source node as the current node and designates it as visited.
- Update: The algorithm examines all the neighbouring nodes of the current node that have not been visited, and updates their distances if a shorter path through the current node is found. If an update occurs, the preceding node for the neighbour is set to the current node.
- Repeat: Steps 'Selection' and 'Update' are repeated until the destination node is visited or all nodes have been visited.
- Path reconstruction: After visiting the destination node, the algorithm retraces its steps from the destination node back to the source node by following the preceding nodes. This process allows it to construct the shortest path between the source and destination nodes.

Following is an example of how Dijkstra Algorithm works.

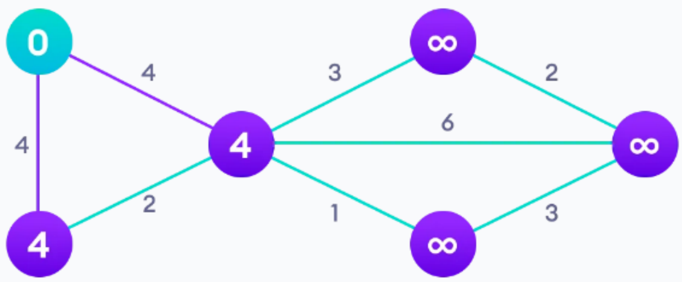


Start with a weighted graph



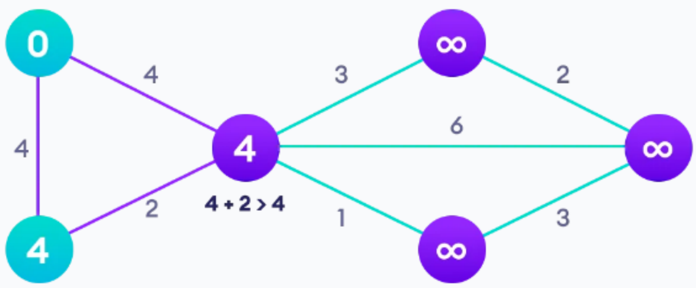
Step: 2

Choose a starting vertex and assign infinity path values to all other devices



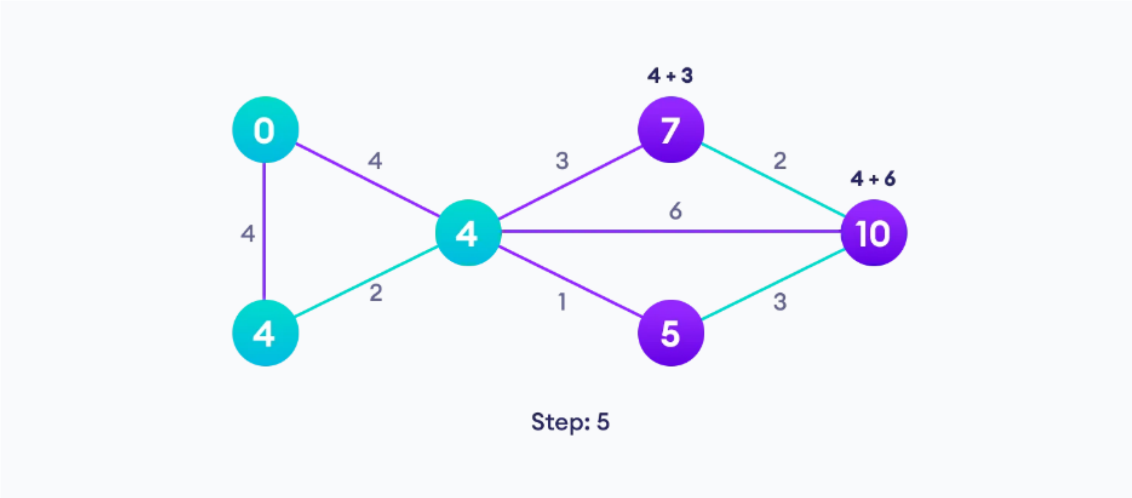
Step: 3

Go to each vertex and update its path length

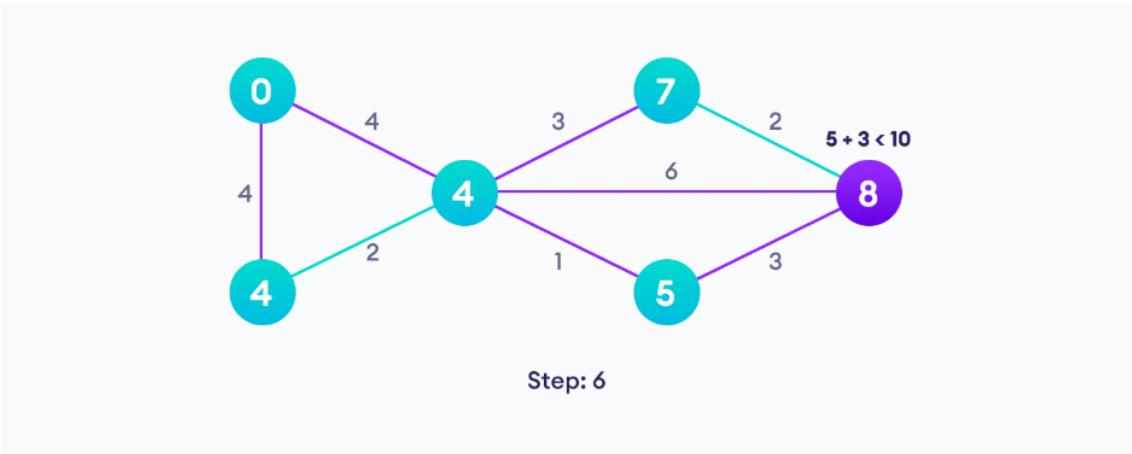


Step: 4

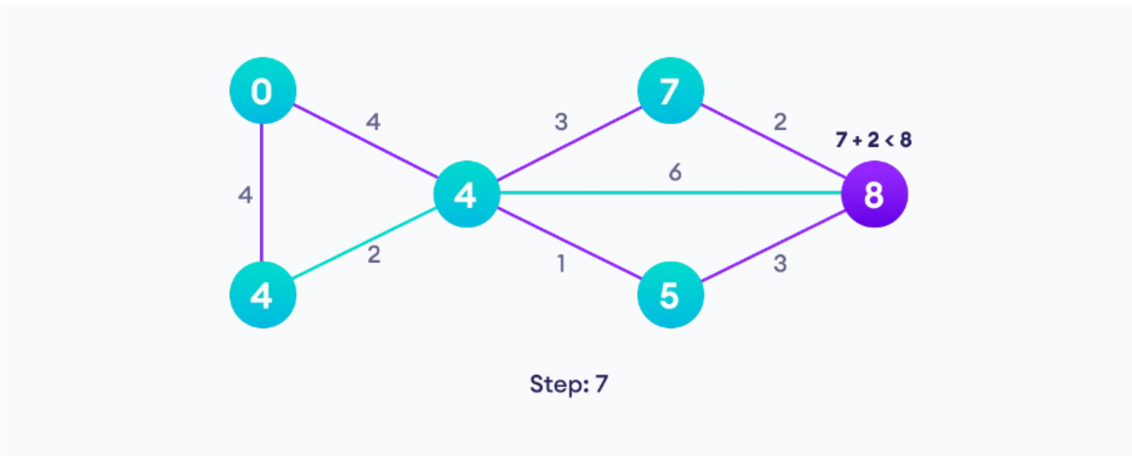
If the path length of the adjacent vertex is lesser than new path length, don't update it



Avoid updating path lengths of already visited vertices



After each iteration, we pick the unvisited vertex with the least path length. So we choose 5 before 7



Notice how the rightmost vertex has its path length updated twice

Figure 7: Example of Dijkstra Algorithm[26]

In VANETs, the Dijkstra algorithm can be used to find the shortest path between a source and a destination node, taking into account various metrics such as distance, bandwidth, or delay. The algorithm can be modified to handle the dynamic nature of VANETs by updating the distances and preceding nodes based on the current state of the network.

However, the Dijkstra algorithm does have limitations, particularly when applied to large-scale networks. Its computational cost increases due to the requirement of maintaining and updating the shortest-path tree. That is precisely why, in this project, both the greedy algorithm and Dijkstra's algorithm are being considered.

2.7.3 A* Algorithm

The A* algorithm[38] is a widely used path-finding algorithm in computer science and artificial intelligence. It is an extension of Dijkstra's algorithm, with the addition of heuristic information to guide the search and improve efficiency.

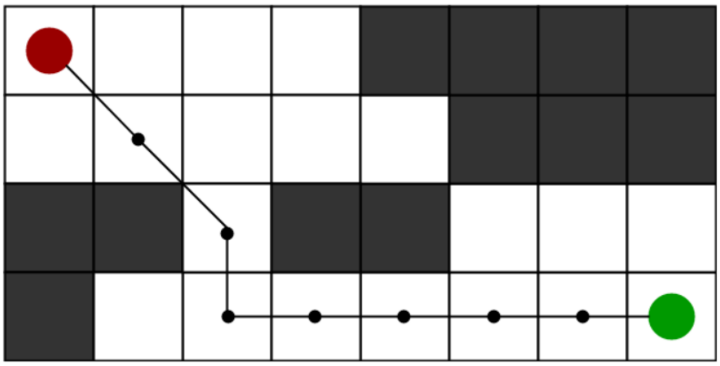


Figure 8: A* algorithm[39]

The A* algorithm is typically used to find the shortest path between two nodes in a graph. It takes into account both the cost of reaching a node from the start node (g-value) and the estimated cost to reach the goal node from the current node (h-value). The sum of these two costs, known as the f-value, guides the algorithm's search process. A* uses a priority queue to explore nodes with lower f-values first, making it more efficient than Dijkstra's algorithm.

The heuristic function used in A* provides an estimate of the cost from a given node to the goal node. This heuristic is problem-specific and can be as simple as the Euclidean distance between two points or more complex based on domain knowledge. The heuristic guides the algorithm

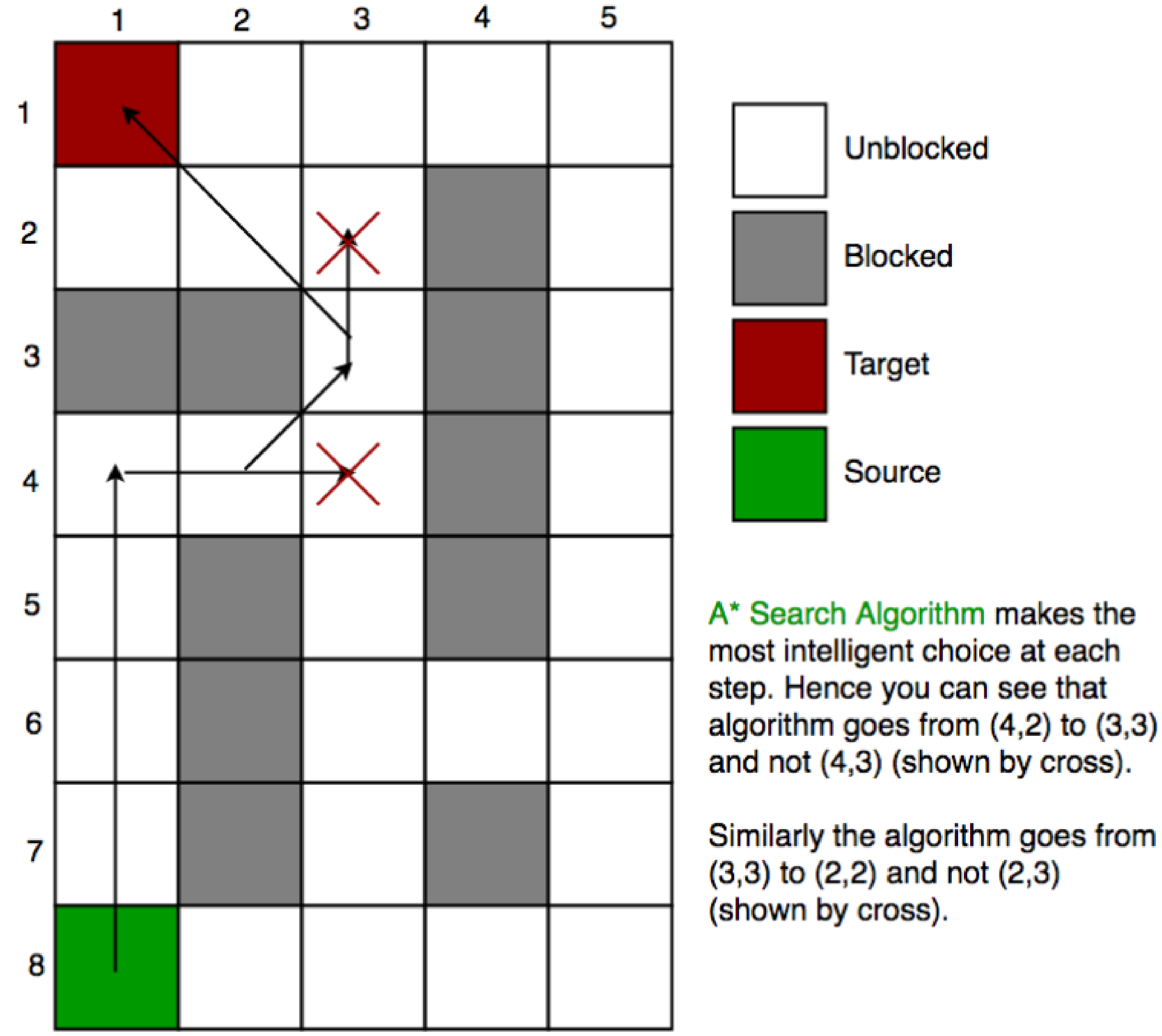


Figure 9: A* algorithm[39]

to prioritize exploring paths that seem more promising, resulting in a more focused search.

In Vehicular Ad Hoc Networks (VANETs), the choice between Dijkstra's algorithm and A* for routing purposes depends on the specific characteristics and requirements of the VANET scenario. While A* is generally more efficient than Dijkstra's algorithm, there are a few reasons why Dijkstra's algorithm might be preferred in certain VANET situations:

Dynamic and changing network topology: VANETs often have a dynamic network topology due to the mobility of vehicles. Nodes (vehicles) move, enter, and leave the network frequently, resulting in frequent changes in connectivity. A* relies on precomputed or estimated heuristics that assume a static environment. In a dynamic VANET scenario, the heuristics used in A* may quickly become outdated and lead to suboptimal routing decisions. Dijkstra's algorithm, which recalculates paths based on the current state of the network, can be more suitable in such dynamic environments.

Limited and unreliable information: In VANETs, the availability and accuracy of information about the network can be limited and unreliable. A* relies on accurate and up-to-date heuristic information to guide the search process efficiently. If the available information is incomplete or unreliable, the benefits of using A* diminish. Dijkstra's algorithm, which considers only the current costs of reaching nodes, can be more robust in scenarios with uncertain or incomplete information.

Safety and reliability considerations: In certain VANET scenarios, safety and reliability are critical factors. Dijkstra's algorithm guarantees to find the shortest path, given non-negative edge weights. This property can be valuable in situations where safety-critical applications or reliable communication are essential. A* does not provide such a guarantee unless an admissible heuristic is used. If finding the optimal path is a priority in VANETs, Dijkstra's algorithm may be preferred.

It's important to note that the choice between the Dijkstra algorithm and A* in VANETs is not universally fixed and depends on the specific requirements, characteristics, and constraints of the VANET scenario. Other variations and optimizations of these algorithms, such as modified versions of A* to account for dynamic environments, may also be considered in VANET routing protocols.

In this project, more discussion on A* will not take place as it is an extension of the Dijkstra algorithm. This project is focused on Dijkstra and the greedy algorithm, the choice between Dijkstra and A* algorithm can be considered as the future work for this project, this involves a further choice of different algorithms in different situations.

3 Simulation

3.1 Simulation Software

This project will use MATLAB as Simulation Software, there were two primary reasons for selecting MATLAB as the simulation software. Firstly, since the existing code and algorithms in the previous project[12] were already developed in MATLAB, the use of the same MATLAB software in both the previous project and the development of new algorithms allowed for a fair comparison of performance between the existing algorithms and the newly developed ones.

Additionally, MATLAB provides a user-friendly environment that allows for rapid algorithm development and prototyping. This is especially useful when experimenting with different algorithms in VANET. Also, MATLAB provides easy-to-use visualization tools that allow for quick and effective analysis of simulation results. This is particularly important when adjusting and fine-tuning algorithms in VANET. On the other side, MATLAB can easily interface with other programming languages, which allows for the seamless integration of algorithms developed in different languages into a single VANET simulation, which is advantageous for further work for this project. The version of MATLAB in this project is chosen to be R2016a, 64-bit.

3.2 Simulation Data

Same as the previous project[12], to create a realistic VANET environment in the simulations, real-world VANET data was employed. The United States Department of Transportation's Federal Highway Administration (FHWA) website[14] was the source of the real-world VANET data used in the simulations. FHWA has been at the forefront of developing traffic simulation models since the 1970s when there were no commercially available simulation packages. The Traffic Analysis Tools Program of the Federal Highway Administration (FHWA) initiated the Next Generation SIMulation (NGSIM) program with the aim of promoting the adoption and accuracy of microsimulation systems.

NGSIM (Next Generation Simulation) is a program developed by the US Federal Highway Administration (FHWA) which was published in 2007, that aims to provide traffic researchers and engineers with access to real-world traffic data. The program collects detailed traffic measurements from various sources, including video cameras, GPS devices, and radar sensors, in order to develop traffic models and improve traffic management strategies. The NGSIM data sets include information such as vehicle trajectories, speed, acceleration, and lane position, which can be used for the simulation and analysis of traffic behaviour in various scenarios, including on highways and in urban environments. The NGSIM program has contributed significantly to the development and evaluation of intelligent transportation systems (ITS) and connected vehicle technologies.

There are three datasets published in the NGSIM, Interstate 80 Freeway Dataset, Lankershim Boulevard Dataset, and US Highway 101 Dataset. While the main focus of this project is on the Interstate 80 Freeway Dataset (I80) for simulation tests, the other datasets will also be introduced below as they are all sourced from NGSIM, and offer various road situations which can be utilized for future work.

US Highway 101 Dataset[31]: As part of the NGSIM program, researchers gathered detailed vehicle trajectory data on the southbound US 101, known as the Hollywood Freeway, in Los Angeles, CA, on June 15th, 2005. The study area spanned approximately 640 meters (2100 feet) and encompassed five mainline lanes with an additional auxiliary lane in certain sections. To capture the vehicles' movements, eight digital video cameras were deployed, and a customized software application called NG-VIDEO was utilized to extract the trajectory data from the video footage. The collected data provided precise information on each vehicle's location within the study area every one-tenth of a second, including lane positions and their positions relative to other vehicles.

The complete dataset comprises 45 minutes of data, divided into three 15-minute segments representing different traffic conditions: 7:50 a.m. to 8:05 a.m., 8:05 a.m. to 8:20 a.m., and 8:20 a.m. to 8:35 a.m. In addition to the trajectory data, the dataset includes various supplementary files such as computer-aided design and geographic information system files, aerial photos, loop detector data, raw and processed video footage, weather data, and aggregate data analysis reports.

The provided figure depicts the US Highway 101, specifically the study area of interest. The aerial photograph showcases the study area and highlights the location of the building where the digital video cameras were installed. It also displays the coverage area of each of the eight cameras used. The bottom schematic drawing illustrates the position of the on-ramp at Ventura Boulevard and the off-ramp at Cahuenga Boulevard. Additionally, it provides information on the number of lanes within the study area.

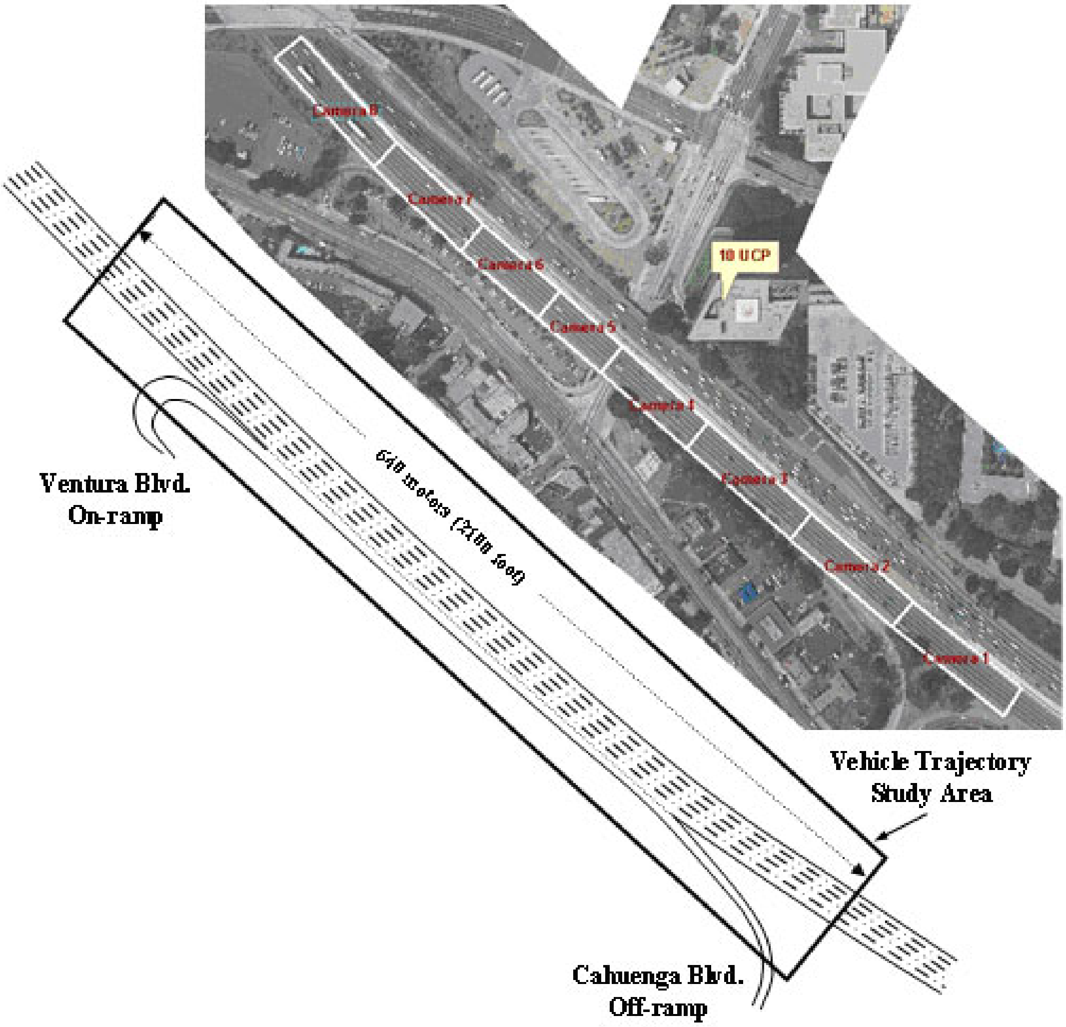


Figure 10: US 101 Study Area[31]

Interstate 80 Freeway Dataset: The NGSIM program researchers obtained comprehensive data on the movement of vehicles on the eastbound lanes of I-80 in Emeryville, CA, in the San Francisco Bay area. This study area extended for around 500 meters and consisted of six lanes, including an HOV lane, along with an on-ramp. The researchers placed seven digital video cameras on top of a 30-story building near the freeway to record the movement of vehicles through the study area.

The complete dataset for Interstate 80 (I-80) consists of 45 minutes of data, divided into three 15-minute time periods, which are from 4:00 p.m. to 4:15 p.m., 5:00 p.m. to 5:15 p.m., and 5:15 p.m. to 5:30 p.m. These time periods capture different traffic conditions, including the onset of congestion, the transition between uncongested and congested conditions, and the peak congestion period. In addition to the vehicle trajectory data, the dataset includes various types of information such as computer-aided design and geographic information system files, aerial orthorectified photos, loop detector data within and around the study area, video data (both raw and processed), signal timing settings on adjacent arterial roads, traffic sign information and locations, weather data, and aggregate data analysis reports.

The provided illustration depicts the layout of US Highway 101. The aerial photograph presents the coverage area of seven video cameras positioned atop a building to capture vehicle movements in the I-80 study area located in Emeryville, CA. Additionally, the photograph showcases the study area's extent in relation to the building. On the right side, an image depicts the position of the Powell Street onramp and the count of lanes present within the I-80 study area.

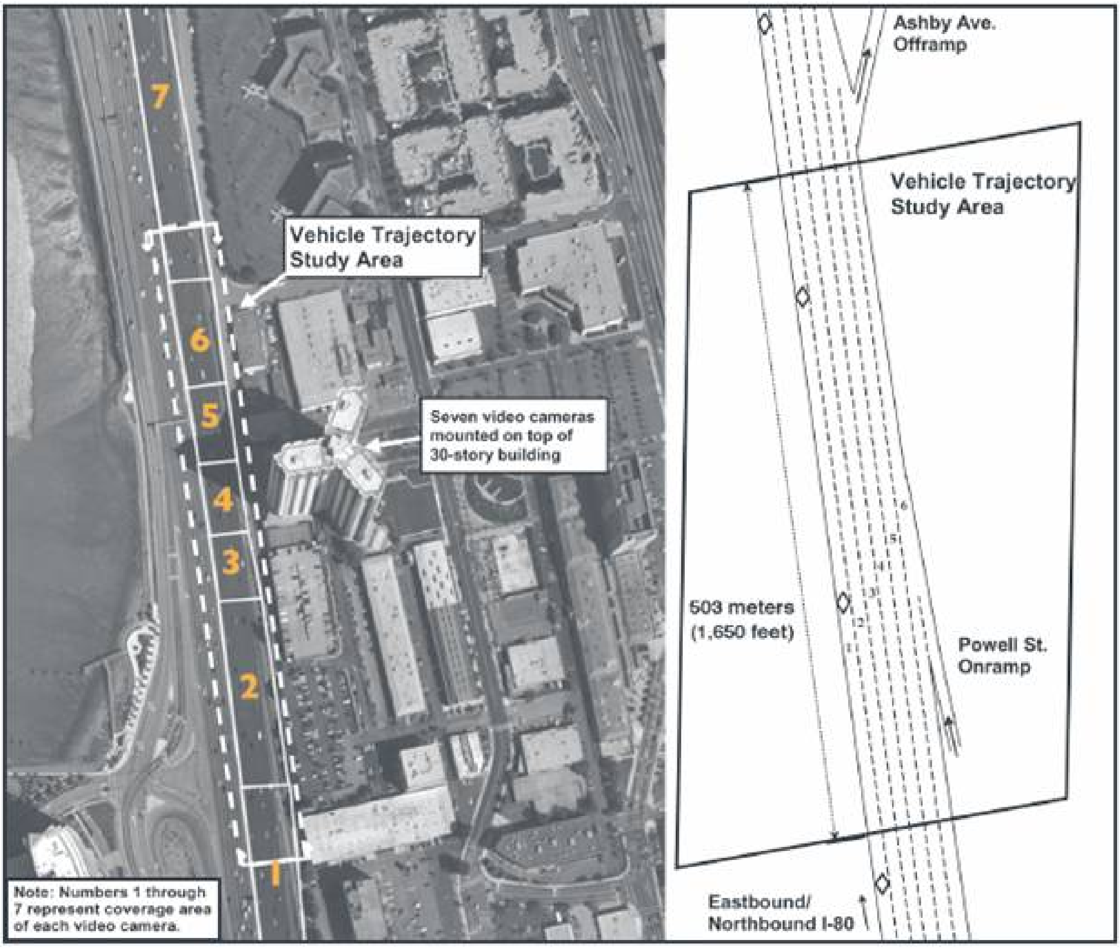


Figure 11: I80 Study Area[33]

Lankershim Boulevard Dataset: On June 16, 2005, researchers involved in the NGSIM program collected comprehensive vehicle trajectory data on Lankershim Boulevard in the Universal City neighbourhood of Los Angeles, CA. The study area spanned approximately 500 meters (1,600 feet) and covered bidirectional data of the three to four lane arterial segments, along with complete coverage of three signalized intersections. The data collection process involved the use of five video cameras mounted on the roof of a 36-story building situated near the interchange of U.S. Highway 101 and Lankershim Boulevard. Specifically, the researchers utilized NG-VIDEO, a custom software application developed for the NGSIM program, to transcribe the vehicle trajectory data from the video footage. This provided precise location information for each vehicle within the study area at intervals of one-tenth of a second, including detailed lane positions and relative locations to other vehicles. The full dataset encompasses 30 minutes of data, divided into two 15-minute periods from 8:30 a.m. to 8:45 a.m. and 8:45 a.m. to 9:00 a.m.

The following figure shows the situation of Lankershim Boulevard, the diagram displays the direction of lanes, signal lights, intersections, and streets that intersect with the study area of Lankershim Boulevard.

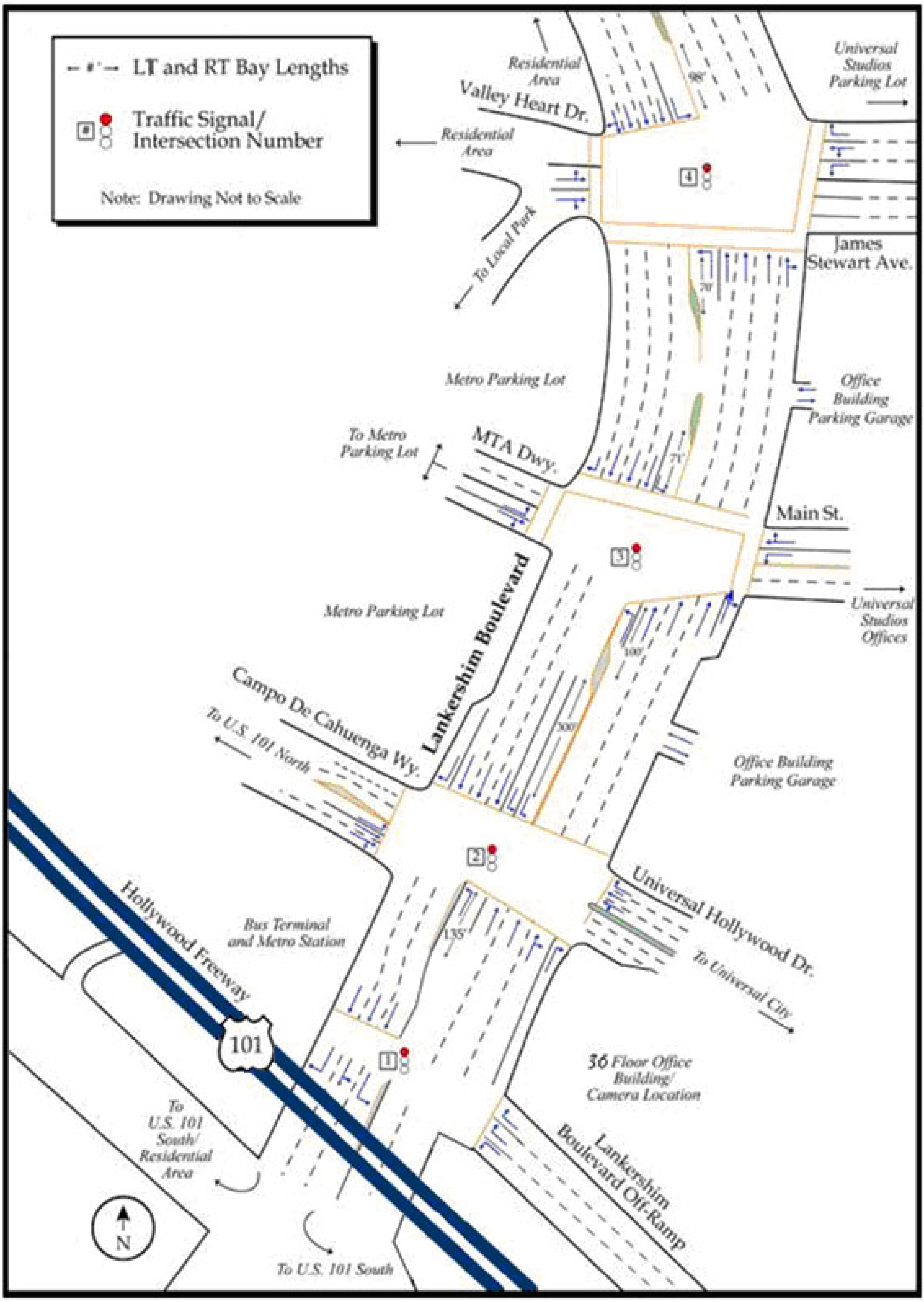


Figure 12: Lankershim Boulevard Study Area[32]

The I-80 dataset's vehicle trajectory file data was organized in a text file format with rows and 18 columns. Each row of the file represented a single data point of a vehicle at a specific time, while the columns contained information about the vehicle at that instance. The following table will show the information in each column.

Column Number	Name	Description	Units
1	Vehicle ID	The vehicle identification number is a unique number assigned to each vehicle in the dataset, and it is sorted in ascending order based on the time the vehicle entered the section.	Number
2	Frame ID	This refers to a number assigned to each recorded frame in ascending order based on the time it started.	1/10 second
3	Total Frames	The total count of frames in which the vehicle is present in the dataset.	1/10 second
4	Global Time (Epoch Time)	Elapsed time since Jan 1, 1970.	Milliseconds
5	Local X	The position of the front centre of the vehicle in the lateral(horizontal) direction, measured from the left-most edge of the section in the direction the vehicle is travelling.	Feet
6	Local Y	The position of the front centre of the vehicle in the longitudinal(vertical) direction, measured from the left-most edge of the section in the direction the vehicle is travelling.	Feet
7	Global X	X Coordinate the front centre of the vehicle based on CA State Plane III in NAD83.	Feet
8	Global Y	Y Coordinate the front centre of the vehicle based on CA State Plane III in NAD83.	Feet
9	Vehicle Length	Length of vehicle	Feet
10	Vehicle Width	Width of vehicle	Feet
11	Vehicle Class	Vehicle type: 1 - motorcycle, 2 - auto, 3 - truck	Text
12	Vehicle Velocity	Instantaneous velocity of vehicle	Feet per second
13	Vehicle Acceleration	Instantaneous acceleration of vehicle	Feet per second square
14	Lane Identification	Current lane position of the vehicle. Lane 1 is the farthest left lane; Lane 6 is the farthest right lane. Lane 7 is the on-ramp at Powell Street, and Lane 9 is the shoulder on the right-side.	Number

Column Number	Name	Description	Units
15	Preceding Vehicle	Vehicle ID of the vehicle that is ahead of the current vehicle in the same lane. If there is no vehicle ahead, the value is '0'.	Number
16	Following Vehicle	Vehicle ID of the vehicle that is behind the current vehicle in the same lane. If there is no vehicle ahead, the value is '0'.	Number
17	Spacing (Space Headway)	Distance between the front-centre of a vehicle and the front-centre of the vehicle immediately preceding it in the same lane.	Feet
18	Headway (Time Headway)	The headway is the time taken for a vehicle to travel from its front-centre to the front-centre of the vehicle in front of it. When a vehicle is not moving, the headway value is set to 9999.99.	second

Table 2: Extracted from NGSIM[14], also used in previous project[12]

For a better comparison with the previous project[12], the same first five minutes from each of the three 15-minute intervals were used and the same number of columns were kept. The dataset I80set1 includes vehicle trajectory information between 4:00 p.m. and 4:05 p.m., while I80set2 covers data from 5:00 p.m. to 5:05 p.m. Another dataset, I80set3, includes data from 5:15 p.m. to 5:20 p.m. For columns, as there are lots of columns which are not important for this project, only column 1, 2, 5, 6, 12, and 13 was used. Similar to the previous project[12], the data points in the rows with odd-numbered values in column 2 were removed, resulting in a time interval of 0.2 seconds between each data point. Column 2's numerical values were adjusted by scaling them such that the first data point was set to 0, and subsequent values increased by 0.2 seconds. Also instead of the raw data from I-80 dataset's vehicle trajectory file data, the first column and second column are exchanged to make the data more obvious to analyze.

3.3 Simulation Machine

As the time delay of the algorithm is analysed in this project, it is floating depending on the different simulation machines. This project is based on TUF Gaming FX505GT_FX95GT, BIOS: FX505GT.310, Processor: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz (8 CPUs), 2.40 GHz. RAM: 8192MB.

4 Design and Implementation

This Project is focused on analyzing the affection of different parameters and improving the algorithm used in the previous project[12], to optimise the performance, delay, stability and generality of the algorithm. First, the algorithm used in the previous will be explained, and then the improvement of both parameters and algorithms will be discussed.

4.1 Performance Features

There are two performance features that are used in the algorithm: Link Expiration Time Predictor(LET) and Link Affinity Predictor(LA). Both features describe how long a communication link stays or how good this link is. Both features use the relative velocity and the distance between two nodes to describe the link, following will discuss two features in detail.

4.1.1 Link Expiration Time Predictor(LET)

Link Expiration Time Predictor(LET) is also used in the previous project[12], the main idea is to calculate the distance between two nodes in the next timestamp, by limiting the largest distance it can be for the link, we can calculate how long this link can maintain. The following will show how the distance between two nodes in the next timestamp is calculated.

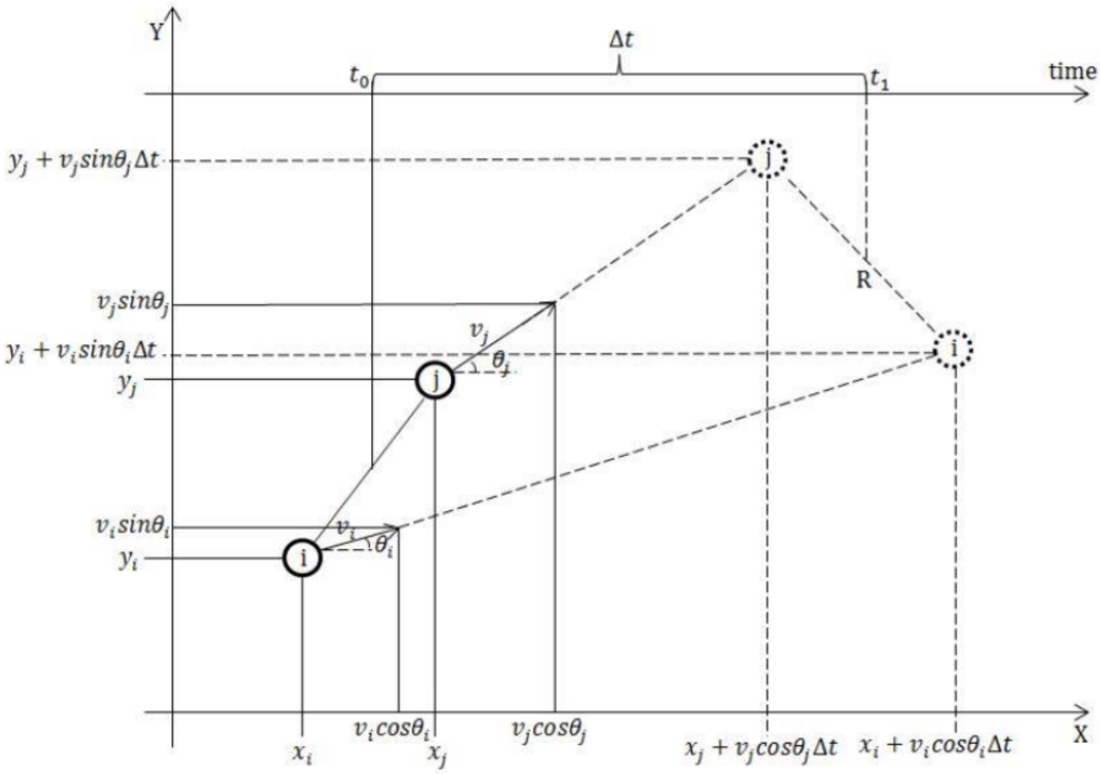


Figure 13: The geometric model of two moving object

From the figure above, we can see there are two moving objects i and j, and each object has its current position (x_i, y_i) and (x_j, y_j), velocity v_i, v_j, direction angle of movement θ_i θ_j, and time between the current time and next timestamp Δt. Using Pythagorean theorem[34], we can calculate the distance between two objects in the current timestamp using the formula:

$$Distance = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

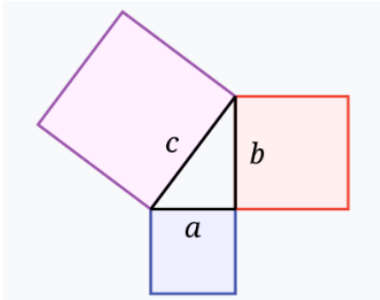


Figure 14: Pythagorean theorem: $a^2 + b^2 = c^2$

For finding the distance between two nodes in the next timestamp we just need to find out each node's position in the next timestamp, which using the Trigonometric functions[35]

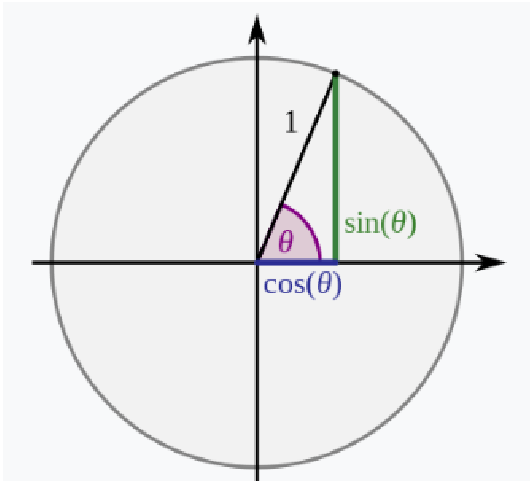


Figure 15: Trigonometric function[35]

Therefore we can calculate both X position and Y position of the next timestamp by adding the corresponding X velocity and Y velocity to the current position. For the next timestamp we will have $((x_i+v_i cos\theta_i \Delta t), (y_i+v_i sin\theta_i \Delta t))$ for moving object i, and $((x_j+v_j cos\theta_j \Delta t), (y_j+v_j sin\theta_j \Delta t))$ for moving object j. Now the distance between two objects in the next timestamp can be calculated using the formula:

$$R = \sqrt{[(x_j + v_j cos\theta_j \Delta t) - (x_i + v_i cos\theta_i \Delta t)]^2 + [(y_j + v_j sin\theta_j \Delta t) - (y_i + v_i sin\theta_i \Delta t)]^2}$$

Solving the equation as Δt, we can find the formula[36]:

$$\Delta t = \frac{-(ab+cd)+\sqrt{(a^2+c^2)R^2-(ad-bc)^2}}{a^2+c^2}$$

where

$$a = v_i \cos\theta_i - v_j \cos\theta_j$$
$$b = x_i - x_j$$
$$c = v_i \sin\theta_i - v_j \sin\theta_j$$
$$d = y_i - y_j$$

Δt is the Link Expiration Time(LET) we used to describe the link. Notes, once $a^2 + c^2 = 0$, which is the case of two objects' relative velocity is 0, we need to prevent 'divide by 0' problem, that's where the parameter 'Velocity Threshold' is introduced, and will be discussed in the following Analyzing Parameter section. The formula changing to[12]:

$$\Delta t = \frac{-(ab+cd)+\sqrt{(a^2+c^2)R^2-(ad-bc)^2}}{a^2+c^2+vel\_threshold}$$

The Path Expiration Time (PET) is determined by the minimum Link Expiration Time (LET) of the routing path, as the failure of any of its component links will result in the failure of the entire routing path.

4.1.2 Link Affinity Predictor(LA)

Another feature Link Affinity(LA) is used in this project same as the previous project, created by J. A. Barria and R. Lent in [37]. Link Affinity calculates the difference between the largest transmission range and the distance between two nodes in the current state. By diving the relative velocity from the distance difference, we can describe the affinity of the link. When the difference between the largest transmission range and the distance between two nodes is smaller, it means the link between two vehicles is closer to touching the limitation of transmission range, closer to breaking the link. Also, the larger the relative is, it means the link between two vehicles is closer to breaking, which is a lower affinity. Formula[37, 12] below shows how it works.

$$LA = \frac{R - D_{ij}}{V_{ij}}$$

where

R = Transmission Range Limitation
D_{ij} = distance between two nodes i and j
V_{ij} = relative velocity between nodes i and j

Similar to the LET, in LA we still need to prevent the divide 0 problem. In this case, if the relative velocity is lower than the parameter 'Velocity Threshold', we need to convert the relative velocity to Velocity Threshold. In the normal case, when the 'divide by 0' problem does not happen, the relative velocity keeps itself.

```
If V_ij < VelocityThreshold
    V_ij = VelocityThreshold
end
```

4.1.3 Stability Function

For the algorithm MPET and MPA developed in the previous project, the final routing decision was made on Stability Function[12]. This is because the stability function is much more sensitive to analyzing and simulating for the algorithm than LET and LA itself.

$$S_{LET} = 1 - e^{\frac{-LET}{a}}$$
$$S_{LA} = 1 - e^{\frac{-LA}{a}}$$

where

S = Stability feature
a = Stability Coefficient

By adjusting the parameter 'Stability Coefficient', the sensitivity of the cost function can be changed, resulting in a change in the marginal cost.

Need to notice, Dijkstra algorithm and greedy algorithm are used in MPET and MPA to find the maximum path expiration time and maximum path affinity. Although the greedy algorithm can be used to find the maximum path, the Dijkstra algorithm always finds the minimum path. Once the Dijkstra algorithm is used, we must use Cost Function as the inverse of the Stability function to find the maximum path we want.

$$Cost = \frac{1}{S}$$

4.2 Algorithms from Previous Project

There are two algorithms developed in the previous project, the analysis and improvement of these algorithms is the focus of this project, following will discuss the detail of these algorithms.

4.2.1 Maximum Path Expiration Time Algorithm(MPET)

The MPET algorithm selects a routing path based on the link expiration times of its constituent links, with the goal of choosing the path with the longest path expiration time. During link finding, there are two ways to find the maximised Path Expiration Time(PET): greedy algorithm and Dijkstra, both are mentioned in the project already.

Algorithm MPET greedy will choose the largest LET at each node for the next hop provided the node is located between the current node and destination. Algorithm MPET Dijkstra is always choosing the shortest path. Thus, for MPET Dijkstra, instead of LET, a reverse of LET will be used.

4.2.2 Maximum Path Affinity Algorithm(MPA)

Algorithm MPA chooses the routing path with the maximum path affinity of its constituent links. Similar to MPET, there are also two sub-algorithm for MPA, called MPA greedy and MPA Dijkstra which do similar works as MPET but with the maximum feature Path Affinity.

4.2.3 Predictor

In algorithms MPA and MPET, the predictor can be used to increase the accuracy and anticipatory of the algorithms.

MPA and MPET are algorithms that calculate the cost and stability of Link Affinity and Link Expiration Time respectively, using the Dijkstra algorithm and greedy algorithm to maximise the Link Affinity and Link Expiration Time. The predictor is working after the link is already found by the Dijkstra and greedy algorithm. The predictor calculated the LET and LA respectively again for the communication link used or created to predict the lowest LA and LET in this communication link. If the lowest LA and LET are less than the threshold, the algorithm will treat the communication as a disconnection imminent link. In this condition, the algorithm will start to find a new link before the in-using link break.

Therefore, the algorithm with predictor will take up more resources, cost more time delay, but as a result, have much better performance. In this project, although the algorithm without the predictor will be considered and simulated, but the algorithm with the predictor will be focused.

4.3 Performance Metrics

The Performance Metrics are used to describe the performance of routing algorithms. Because this project is focused on analyzing and improving the previous algorithm, two Performance Metrics used before will continue used in this project: Average Path Lifetime and Percentage Path Availability[12]. One new Performance Metric is introduced, called: Time Delay.

4.3.1 Average Path Lifetime

The Average Path Lifetime refers to the length of time that a routing path is available continuously, on average. When considering network conditions with similar parameters, having a longer average path lifetime is advantageous as it allows for more information transmission across the network compared to a shorter path lifetime. Furthermore, a longer path lifetime signifies a reduced number of overhead packets associated with path discovery and establishment.

$$\text{Average Path Lifetime} = \frac{T}{N}$$

where

T = Total time available (second)
N = Total number of the available periods (number)

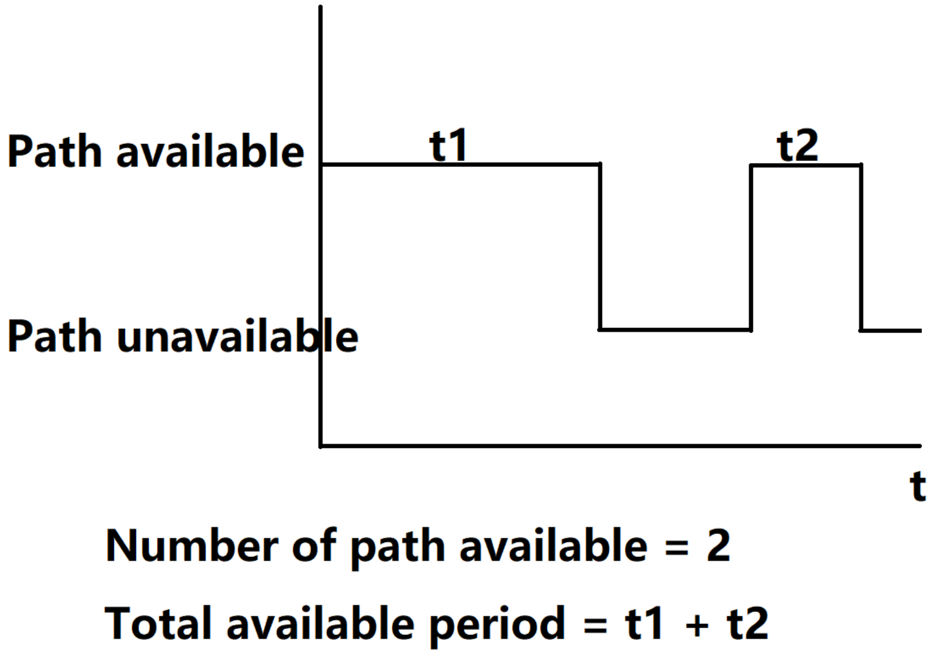


Figure 16: Average Path Lifetime

Using the algorithm above we can find:

$$\text{Average Path Lifetime} = \frac{T}{N} = \frac{t1+t2}{2}$$

4.3.2 Percentage Path Availability

Percentage Path Availability is a metric that measures the frequency with which a path is accessible when it is needed for data transmission. It indicates how successful a routing algorithm is in identifying a suitable routing path that remains viable for the entire duration it is needed. This is a critical performance indicator for routing algorithms, with those achieving a higher percentage of path availability being considered better, assuming other factors are equal.

$$\text{Percentage Path Availability} = \frac{T}{T_t} * 100\%$$

where

T = Total time available(second)
T_t = total amount of time a routing path was required(second)

4.3.3 Time Delay

The metric of Time Delay is subject to variability due to factors such as the specific running machine and environmental conditions. It measures the time taken for the algorithm to execute on a particular machine and can be influenced by variables like temperature and time of execution. To ensure the reliability of this metric, the project will measure Time Delay on the same machine consistently. Additionally, when comparing different datasets, the Time Delay measurements will be conducted within adjacent time intervals. The Time Delay measurement will be implemented using code inserts such as 'tic' and 'toc' within the algorithm's code. The Time Delay metric will be selectively utilized when it holds significance. In cases where the differences in Time Delay are negligible or primarily caused by external environmental factors, they will be disregarded. The focus will be on instances where Time Delay variations have substantive implications for the analysis being conducted. This approach ensures that Time Delay is only considered when it provides meaningful insights and excludes instances where it is predominantly influenced by external factors.

4.4 Parameter Analysis

In the previous project, the parameter used is chosen casually and only simulated well for the particular situation, which is not a good algorithm for both generically and for further work. Therefore, this project will analyze the differences in performance for algorithms with the changing of parameters, and choose the more general parameter values. The analyzing part with the result of choosing parameters will show in the following part in 'Analysis and Result'.

In this project the Algorithm Maximum Path Affinity Algorithm(MPA) will be forced during the simulation test. There are 4 parameters in Maximum Path Affinity Algorithm: Stability Coefficient, Danger Timeout, Velocity Threshold, and Transmission Range.

- Stability Coefficient: Affect the calculation of stability function.
- Danger Timeout: The threshold of the lowest link affinity can be tolerated when using the predictor in the algorithm
- Velocity Threshold: The lowest relative velocity can be tolerated for the calculation of features LET and LA, to avoid 'divided by 0' situation.
- Transmission Range: The limitation assumed for the longest distance between two nodes can maintain link communication.

With the changing of these parameters, the performance of the algorithms will be affected obviously, and for parameter 'Danger Timeout' and 'Transmission Range', the running time (delay) of the algorithms will be affected.

We want the algorithms to have more generality, which can work well for most situations. Better performance, which will be closer to the real situation. Lower delay, as no one wants to use an algorithm with very high delay although the performance is good.

4.5 Up-Edge Detection

The performance metric Average Path Lifetime is used to describe the algorithm, but in the previous project, there are some problems with it. The algorithm to calculate the Average Path Lifetime is not completed, it only counts the number of down edges as the total number of available period numbers. It does well sometimes on comparison between different datasets, but it does not describe the Truth value of the Average Path Lifetime in real work in some special situations, which is not a good option as the dataset used is real work data from NGSIM [14]. The problem will show in the following figure.

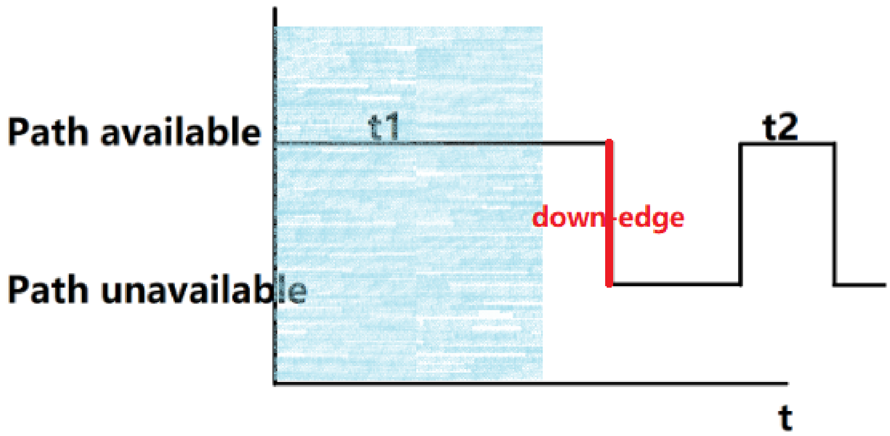


Figure 17: Average Path Lifetime Problem 1

The blue part is the observable area for the algorithm to calculate the Average Path Lifetime, there is zero down edge that can be detected in the observable area. Thus, the previous algorithm will calculate as:

$$\text{Average Path Lifetime} = \frac{t1}{0} = Inf$$

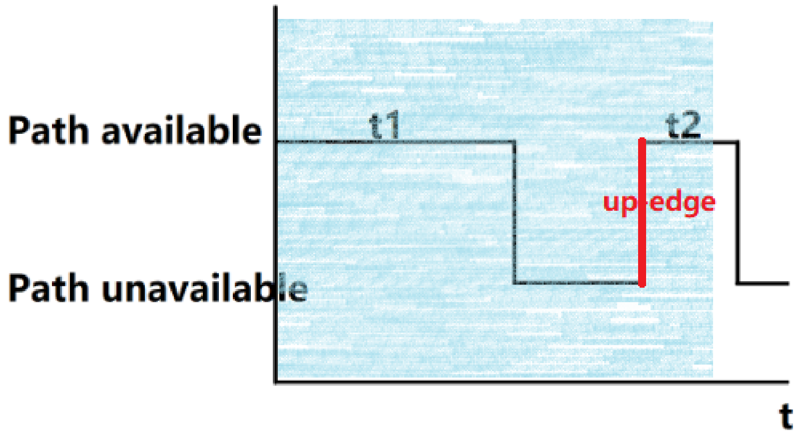


Figure 18: Average Path Lifetime Problem 2

The blue part is the observable area, only 1 down edge is detected, but should have 2 available periods.

Both situation above is erroneous for calculating the Average Path Lifetime. To avoid these problems, a new calculation algorithm: Up-Edge Detection is used. Instead of detecting down edge, we should detect the up-edge. Once up-edge is detected, we can make sure, there is a new available period. But this algorithm still has a problem: there is no up-edge if the available period is starting at the beginning. Therefore, one more task is taken in the algorithm to detect the beginning period separately.

```
count = 0;
if path_av(1) == 1 %detect the beginning period
    count = count + 1;
end

for i = 1:L-1
if path_av(i)==0 && path_av(i+1)==1 % detect up-edge
    count = count + 1; % count the number of available period
end
end
```

Following figure(the figure above is the previous one, the figure below is the new calculation algorithm: up-edge detection) shows clearly how the new calculation algorithm is working better on respecting from reality dataset and handling special situations. Using down-edge detection, at danger_timeout=0.2, we can see the average path lifetime is lower than the real value. Once over 0.3 danger_timeout, the down-edge method will calculate the average path lifetime as **Inf(infinity)** when no down-edge is detected which can not show appropriately in the graph. However, this is not accurate or suitable for calculation and comparison because there are no communication links that remain connected indefinitely.

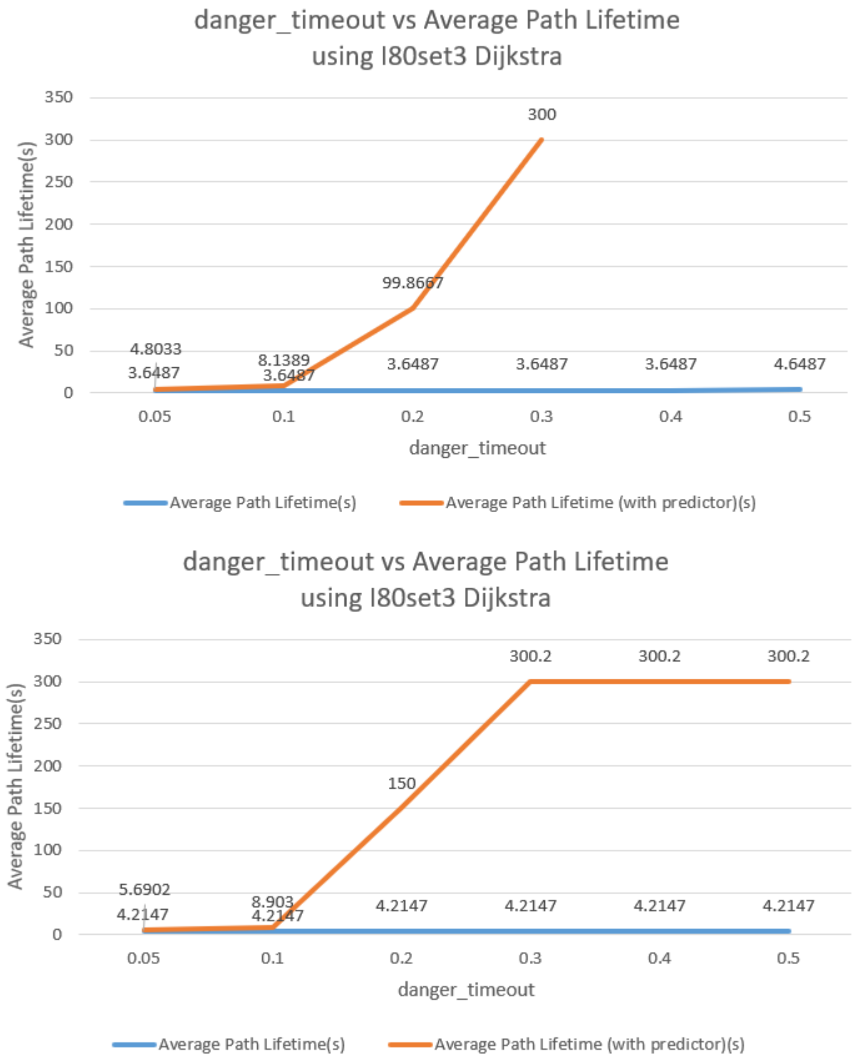


Figure 19: Comparison of new calculation algorithm: up-edge detection

4.6 Automatically Selectable Maximum Path Affinity Algorithm(ASMPA)

From the previous explanation, we can find out that the algorithm Maximum Path Affinity Algorithm(MPA) which was developed in the previous project is limited by the algorithms it used, Greedy Algorithm and Dijkstra Algorithm. But both algorithms MPA Dijkstra and MPA greedy have their own problem.

Although MPA Dijkstra has better performances, but takes much longer running time(delay). MPA greedy has a lower running time, as the price, having the worse performance of the most time. Also, it is necessary for users to understand and select the different algorithms in different situations, which is not convenient and general. In this project Automatically Selectable Maximum Path Affinity Algorithm(ASMPA) is designed, which depends on the different road situations to select the better algorithm. Thus, ASMPA will combine the advantages of both MPA Dijkstra and MPA greedy, with better performance and lower time delay.

The performance of the MPA Dijkstra and MPA greedy is depending on the road congestion situation, with higher congestion, the cars are moving much slower, and the topology of communication link is changing slower and less complicated. Therefore, the MPA algorithm can work with similar performance and lower time delay using MPA greedy. Thus, ASMPA will automatically select the MPA greedy algorithm, and vice versa.

```
total_vel = 0;

%% Calculate the average velocity
NumOfData = length(Rundata);

for i = 1:NumOfData;
    total_vel = total_vel + Rundata(i,5);
end
average_vel = total_vel / NumOfData;

%% Using average velocity to choose method

if average_vel < Parameter Threshold
    MPA_greedy;

else
    MPA_Dijkstra;

end
```

The analysis of average vehicle velocity vs. Performance and Time Delay will be shown in the following ' Analysis and Result' part, further explanation can also be found. In conclusion, ASMPA is an algorithm that combines the benefit of both MPA Dijkstra and MPA greedy, lower Time Delay and better Performance. This is achieved by using greedy during low average velocity, and Dijkstra during high average velocity.

5 Analysis and Result

5.1 Analyzing of Parameters

5.1.1 Stability Coefficient

Stability Coefficient is a parameter used to calculate the cost function of the Dijkstra algorithm, hence Stability Coefficient affects the sensitivity of the Dijkstra Algorithm, which will affect the performance of the algorithm. The figure below shows how the changing of the Stability Coefficient affects the performance of the MPA Dijkstra algorithm for each of the datasets I80set1, I80set2 and I80set3.

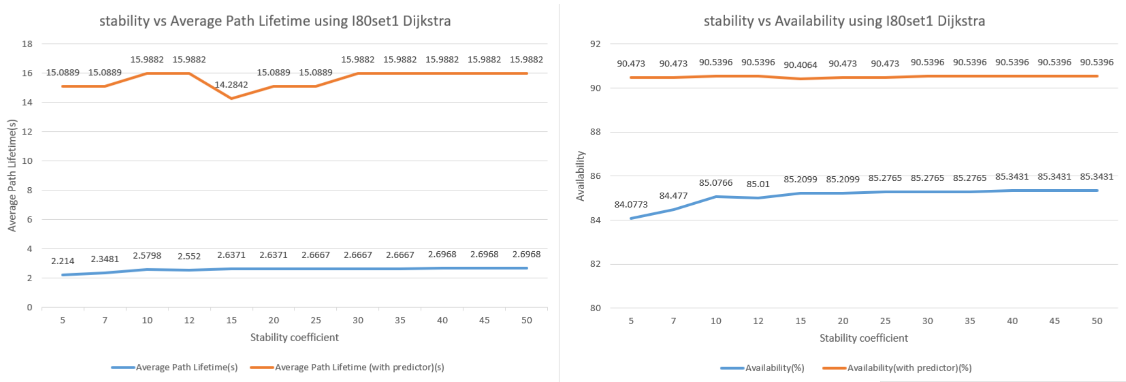


Figure 20: stability coefficient vs. Average Path Lifetime and Percentage Path Availability using dataset I80set1 with algorithm MPA Dijkstra

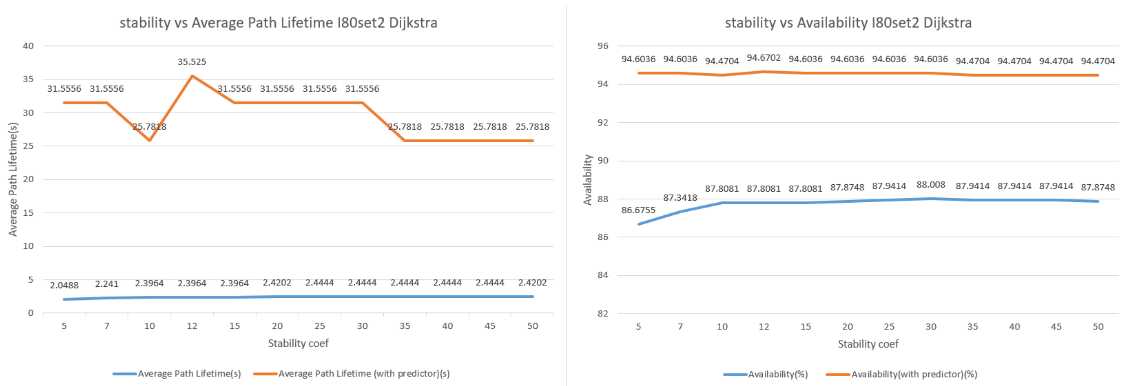


Figure 21: stability coefficient vs. Average Path Lifetime and Percentage Path Availability using dataset I80set2 with algorithm MPA Dijkstra

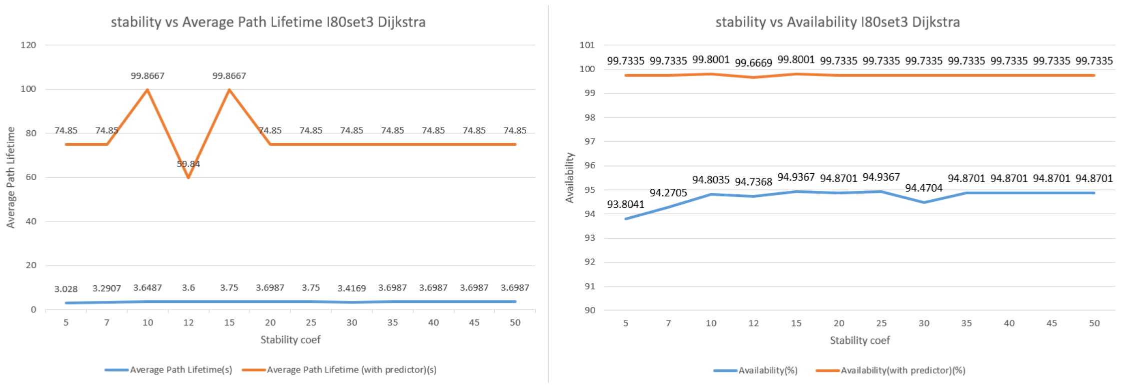


Figure 22: stability coefficient vs. Average Path Lifetime and Percentage Path Availability using dataset I80set3 with algorithm MPA Dijkstra

From the result above, it is obvious that the stability coefficient is very sensitive for feature Average Path Lifetime using the Dijkstra algorithm with predictor(red line) during stability coefficient between 7 to 20 in general, except for I80set1. But the stability coefficient is not very sensitive for Percentage Path Availability, because within the enormous dataset, the Dijkstra with predictor already works very well on finding the next hop in the VANET network. Although the stability coefficient does affect the Percentage Path Availability, for most timestamps it is still available to create or maintain the connecting.

Considering the Dijkstra algorithm without predictor(blue line), although the Average Path Lifetime is not sensitive in dataset I80set2, from two other datasets I80set1 and I80set3, we can see the stability coefficient go through a big improvement of both Average Path Lifetime and Percentage Path Availability at 10, and do not sensitive as before if the stability coefficient is increasing over 10.

On the other side, the figure below shows that the stability coefficient is not affecting the greedy algorithm at all, if you want to see more datasets about the stability coefficient vs. performance to make sure it is not affecting greedy you can find in 'Appendix'. Therefore, choosing a good stability coefficient is totally decided by the Dijkstra algorithm. For the algorithm to be more general and compatible with other data, instead of 10 in the previous project, the stability coefficient should be set to 20 or 25.

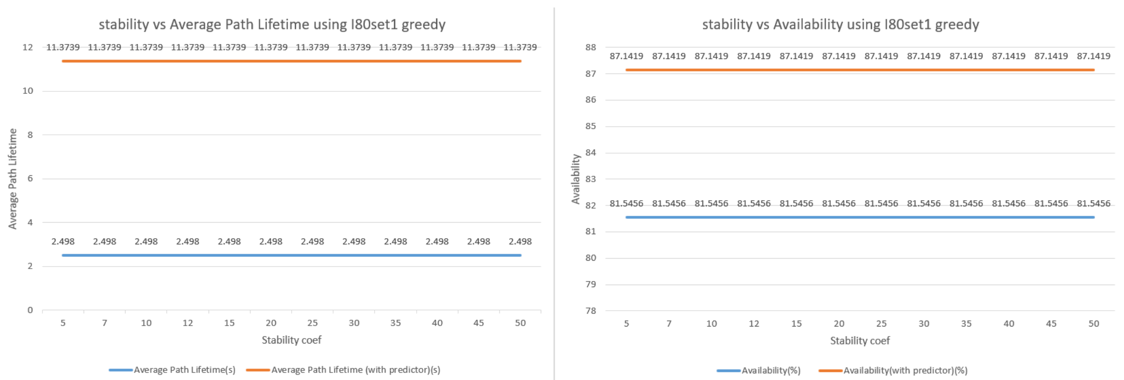


Figure 23: stability coefficient vs. Average Path Lifetime and Percentage Path Availability using dataset I80set1 with algorithm MPA greedy

5.1.2 Danger Timeout

Danger Timeout is a parameter that is used to check whether the link affinity is too low to maintain the connecting link anymore when using the predictor. If the link affinity is lower than the Danger Timeout, the new link connecting should be found as a replacement in the next timestamp before the link is already broken. Thus, the parameter Danger Timeout will not affect the algorithm without the predictor. The following will show how Danger Timeout affect the Performance of both algorithm MPA Dijkstra and MPA greedy, also the affection on the time cost of the algorithm will be considered and shown below, the reason will be discussed after the result.

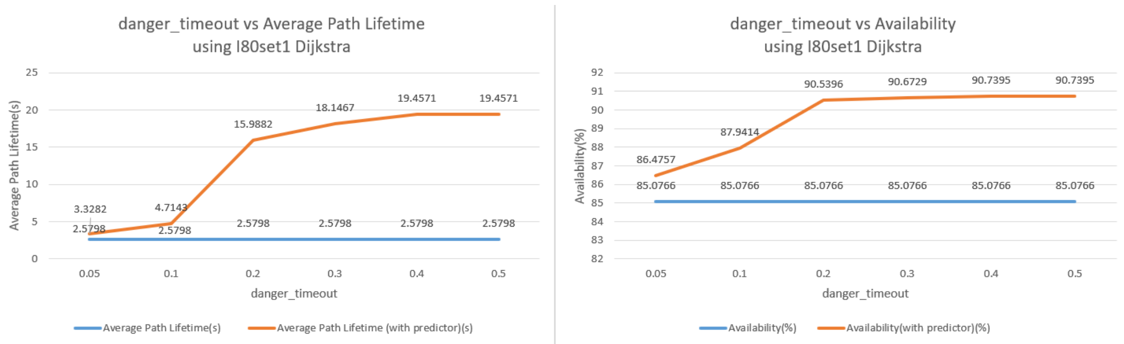


Figure 24: Danger Timeout vs. Average Path Lifetime and Percentage Path Availability using dataset I80set1 with algorithm MPA Dijkstra

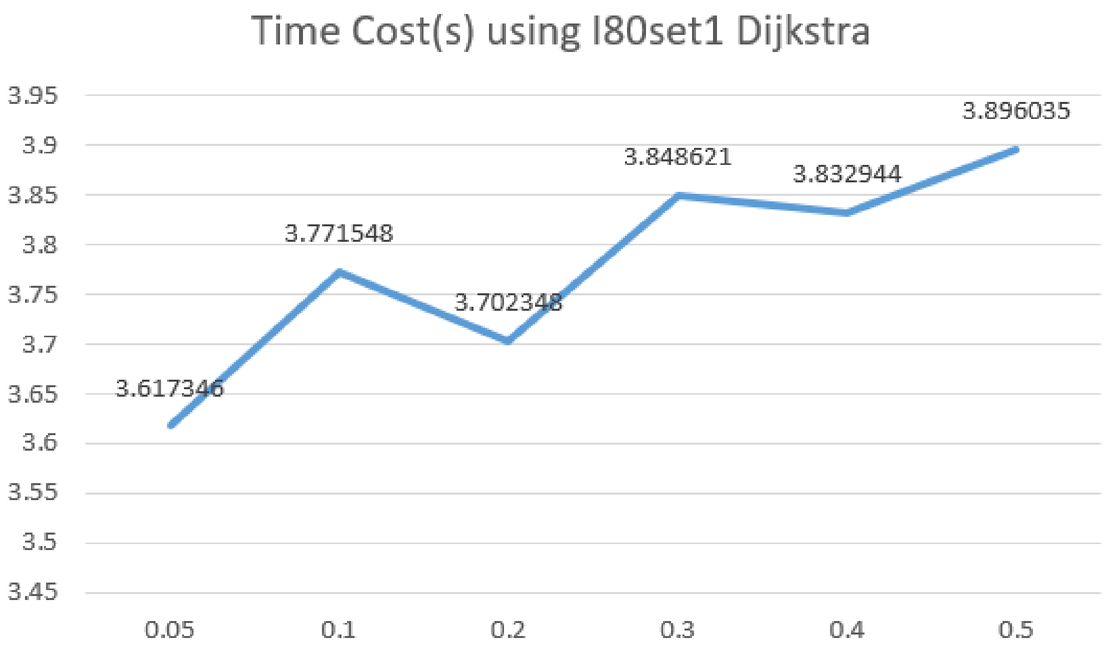


Figure 25: Danger Timeout vs. Time Cost using dataset I80set1 with algorithm MPA Dijkstra

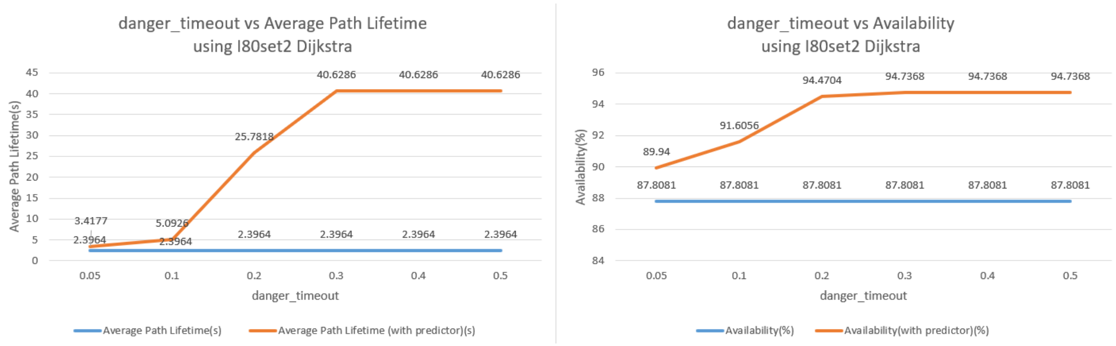


Figure 26: Danger Timeout vs. Average Path Lifetime and Percentage Path Availability using dataset I80set2 with algorithm MPA Dijkstra

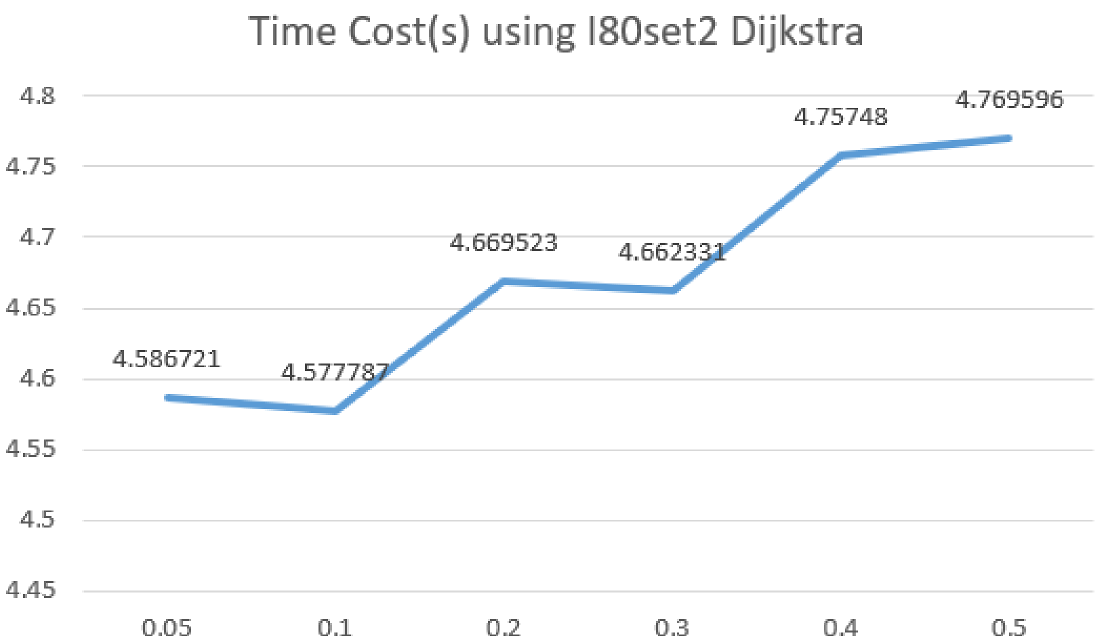


Figure 27: Danger Timeout vs. Time Cost using dataset I80set2 with algorithm MPA Dijkstra

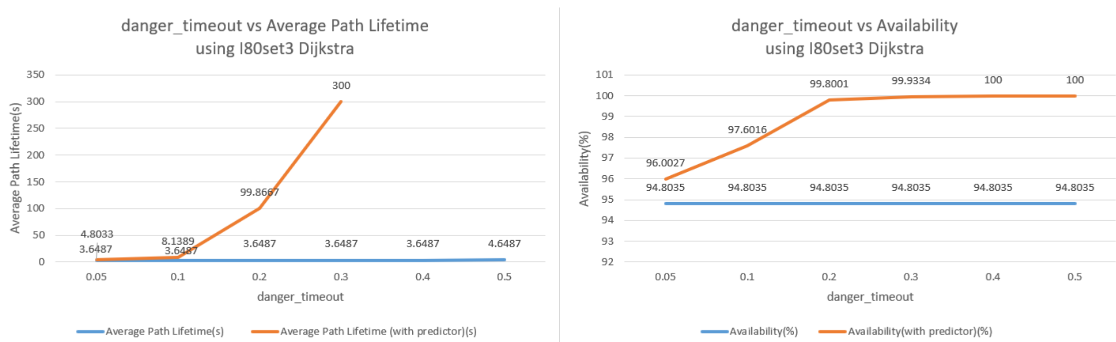


Figure 28: Danger Timeout vs. Average Path Lifetime and Percentage Path Availability using dataset I80set3 with algorithm MPA Dijkstra

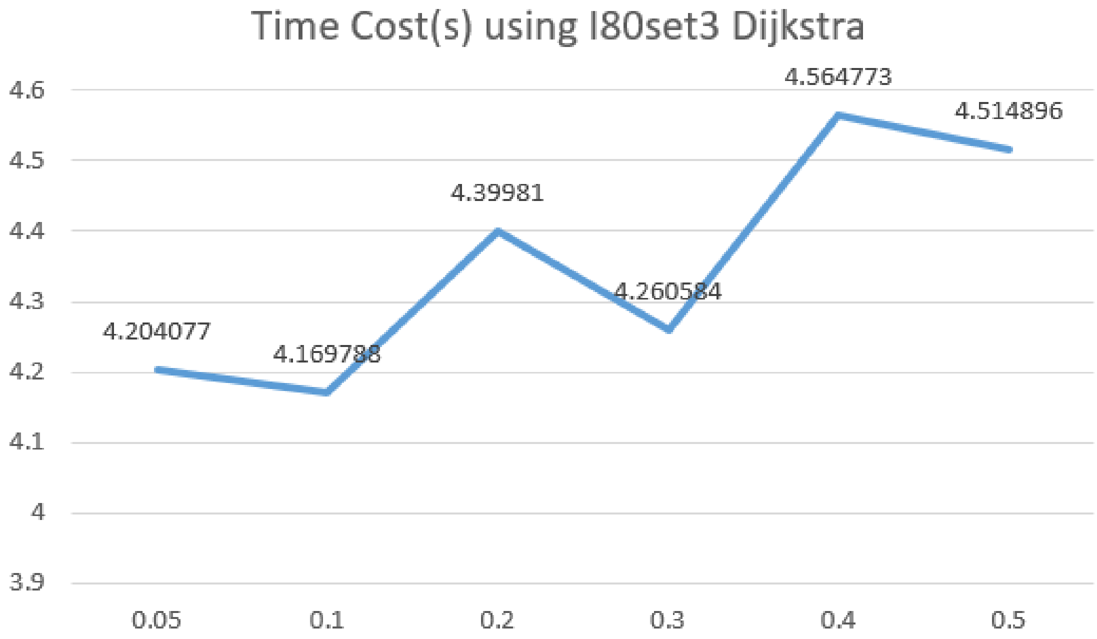


Figure 29: Danger Timeout vs. Time Cost using dataset I80set3 with algorithm MPA Dijkstra

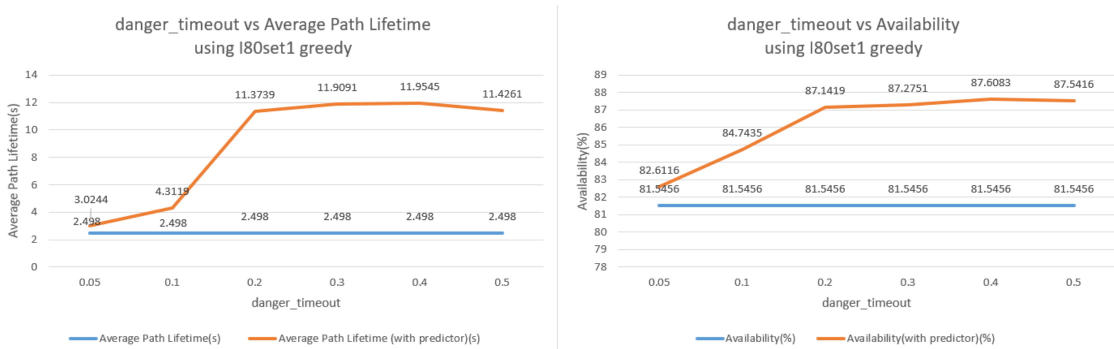


Figure 30: Danger Timeout vs. Average Path Lifetime and Percentage Path Availability using dataset I80set1 with algorithm MPA greedy

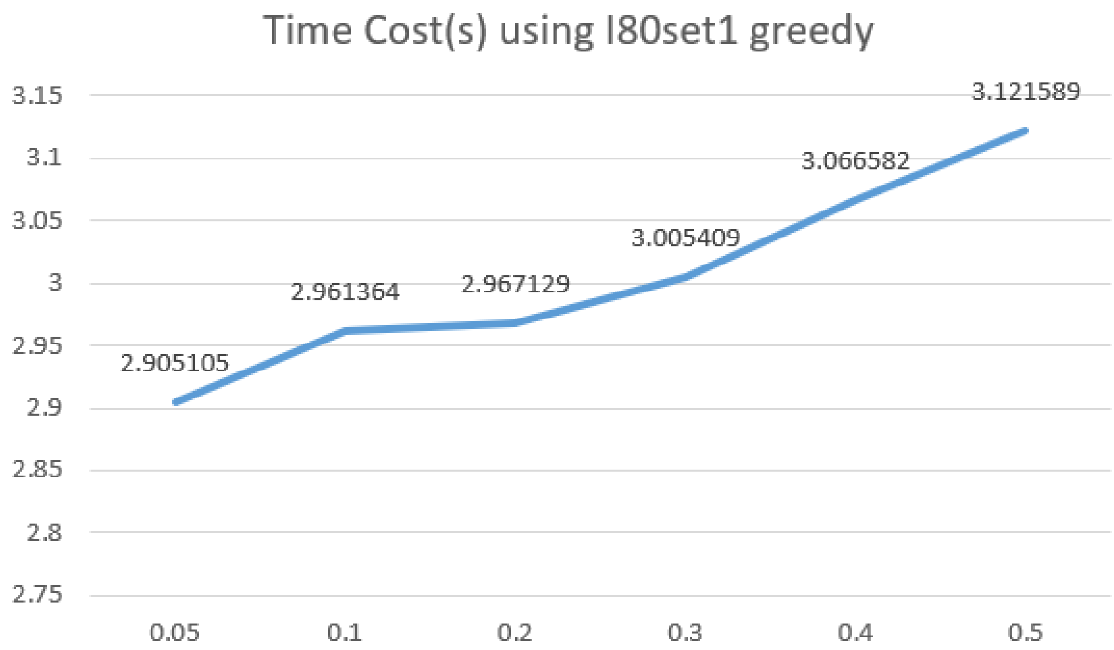


Figure 31: Danger Timeout vs. Time Cost using dataset I80set1 with algorithm MPA greedy

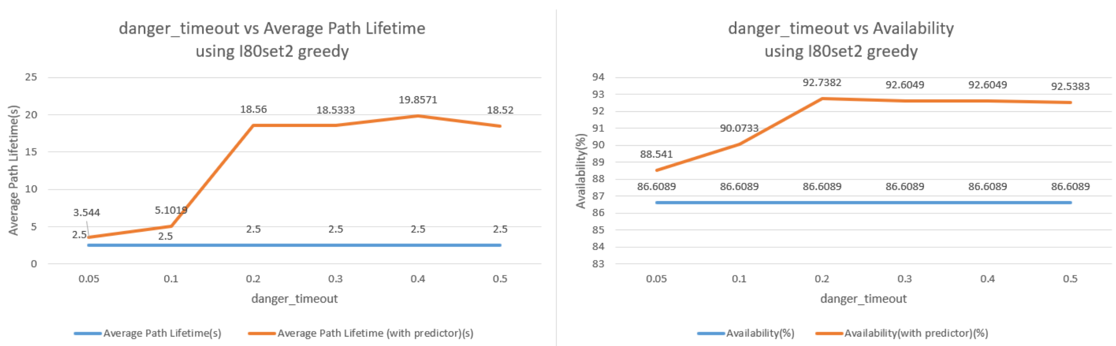


Figure 32: Danger Timeout vs. Average Path Lifetime and Percentage Path Availability using dataset I80set2 with algorithm MPA greedy

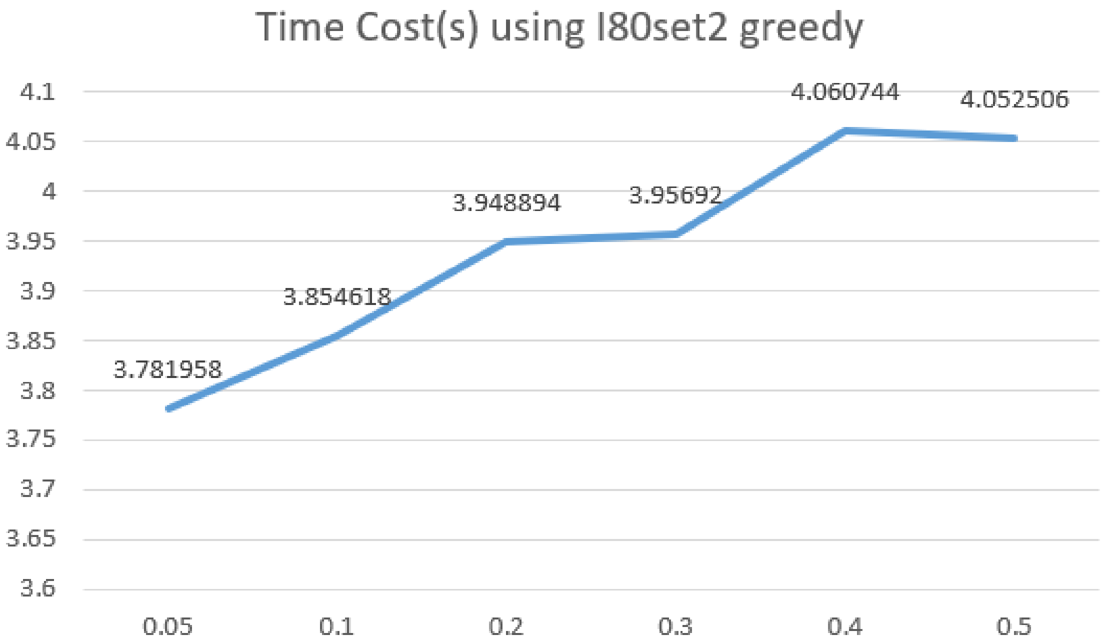


Figure 33: Danger Timeout vs. Time Cost using dataset I80set2 with algorithm MPA greedy

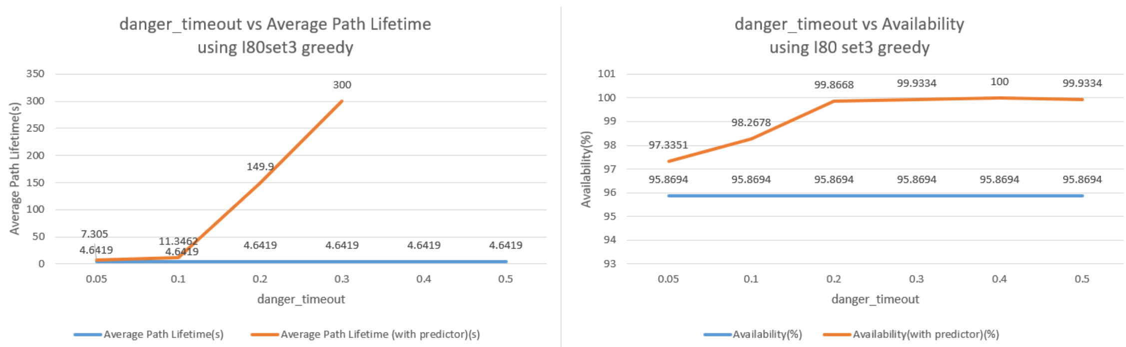


Figure 34: Danger Timeout vs. Average Path Lifetime and Percentage Path Availability using dataset I80set3 with algorithm MPA greedy

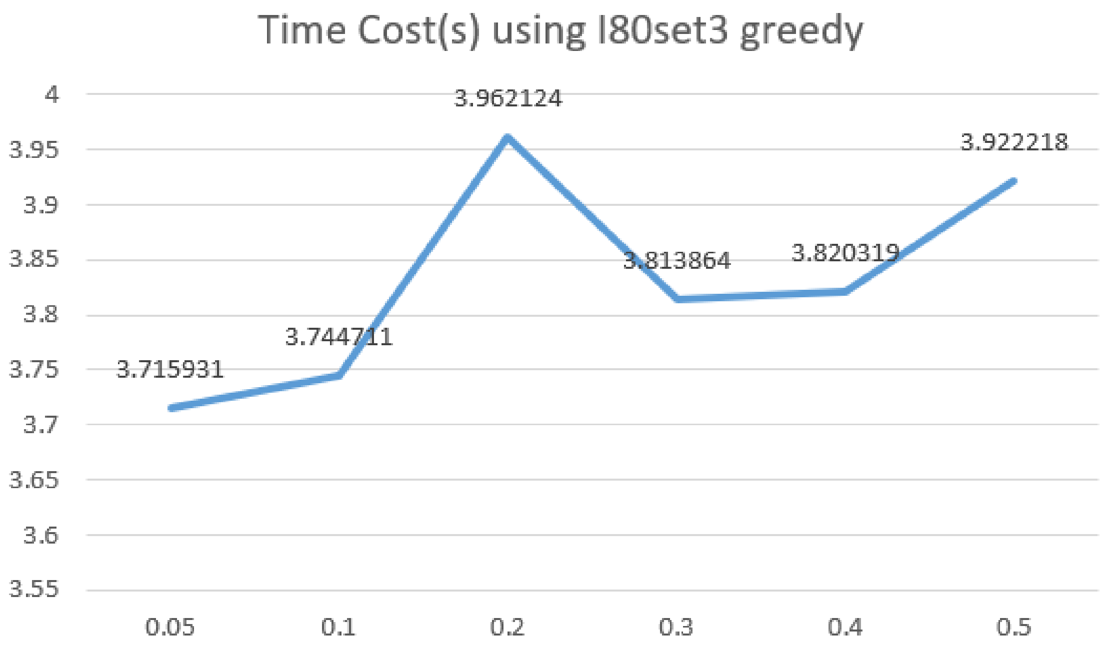


Figure 35: Danger Timeout vs. Time Cost using dataset I80set3 with algorithm MPA greedy

From the result above shows that the increasing of parameter Danger Timeout has a significant improvement of both Average Path Lifetime and Percentage Path Availability, but also increases the time cost of the algorithm in every simulation condition. Although Time Cost have some floating value, as the Time Cost itself is a floating value, this project will consider the Time Cost as the general trend.

The reason Time Cost increasing is that with the increasing of Danger Timeout, the link affinity of the current link is easier to touch the threshold, which makes the new link are recreating frequently. With lower Danger Timeout, the algorithm finds a new link less frequently. Also, a higher Danger Timeout does make the algorithm works much better than a lower Danger Timeout which we can see from the result above, but the enormous increase in Time Cost is normally not good, especially in the VANET network. Thus, the balance between Time Cost and Performance should be considered.

From the result above shows, in most cases, the results before 0.3 will significantly improve with an increase in the value of Danger Timeout, but improvement will mostly reach maximum performance after that as the time cost will still increase quickly. Therefore, considering both Performance and Time Cost, for the generality of the algorithm Danger Timeout should be chosen as 0.3.

Note: The Time Cost of the algorithm is totally depending on the machine used to run this code, and every time the result will be different depending on the situation of the machine, even with the same machine, at the same temperature, the Time Cost will always different. In this project, The most representative number was taken as every code is running several times for every dataset.

5.1.3 Velocity Threshold

Velocity Threshold is a parameter that is used when calculating the features LET and LA to avoid 'divide by 0 'problem. Therefore, changing the velocity threshold is not a big effect on the performance of the algorithms. The following figure can prove it. Thus, the velocity threshold is not a parameter we need to force on for the performance of the algorithms, choosing a suitable value would be enough for the generality of the algorithms.

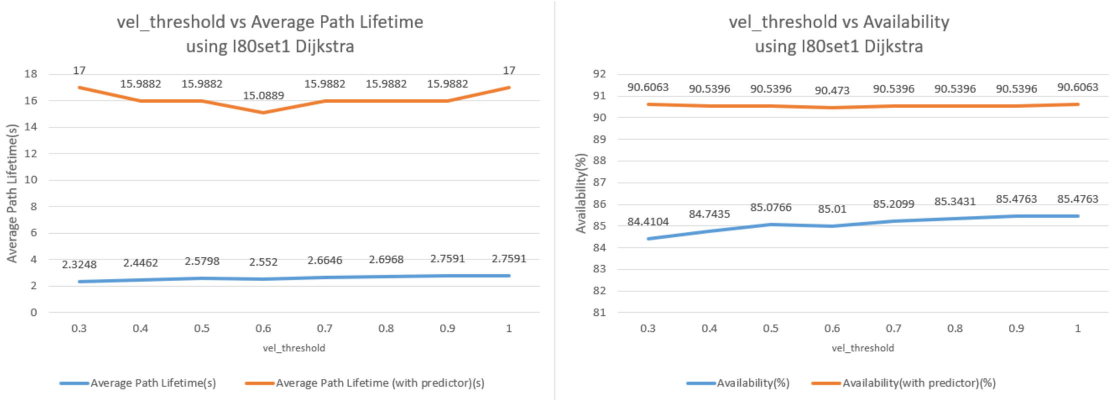


Figure 36: velocity threshold vs performance

Others analyzing figure of the velocity threshold will show in the 'Appendix' section as this parameter is not an important and interesting point for this project.

5.1.4 Transmission Range

Transmission Range is a parameter that controls the limitation of the longest distance the communication link between two vehicles can maintain. This is depending on the development of the equipment for communication. From the previous project[12], the transmission range is set to be 65 feet (about 20 metres), which is much lower than the maximum range of 3,280 feet (about 1km) for IEEE 802.11p vehicular variant. 65 feet was chosen because the study area in I80 is just 500 meters (1,640 feet), but since this project's aim is to make the algorithm much more general and compatible with other datasets, we need to consider the changing limitation of the Transmission Range, as the development of equipment is rapid progress since the previous project was written in 2014. Therefore, the analysis of the parameter Transmission Range is necessary and interesting for this project.

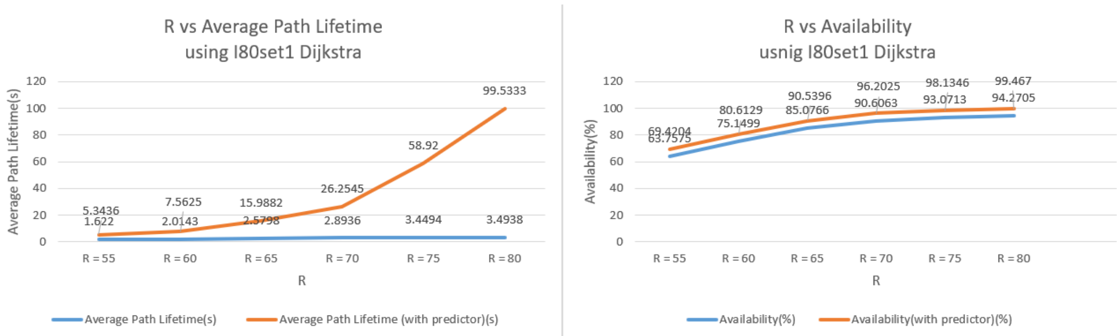


Figure 37: Transmission Range using I80set1 with MPA Dijkstra

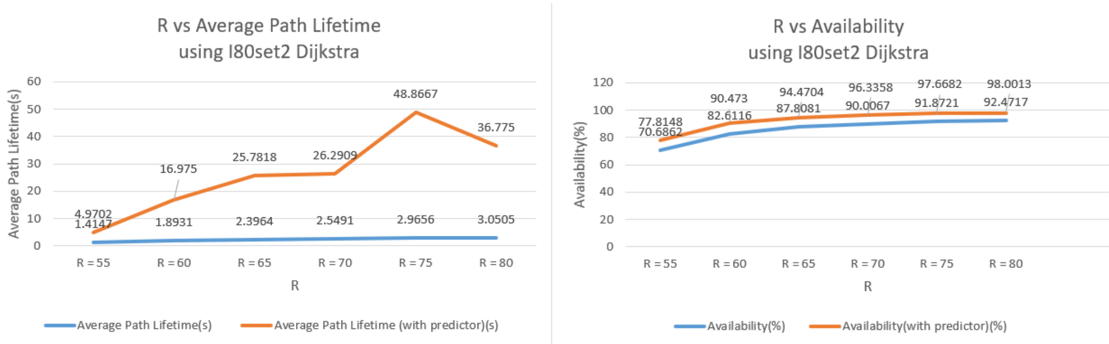


Figure 38: Transmission Range using I80set2 with MPA Dijkstra

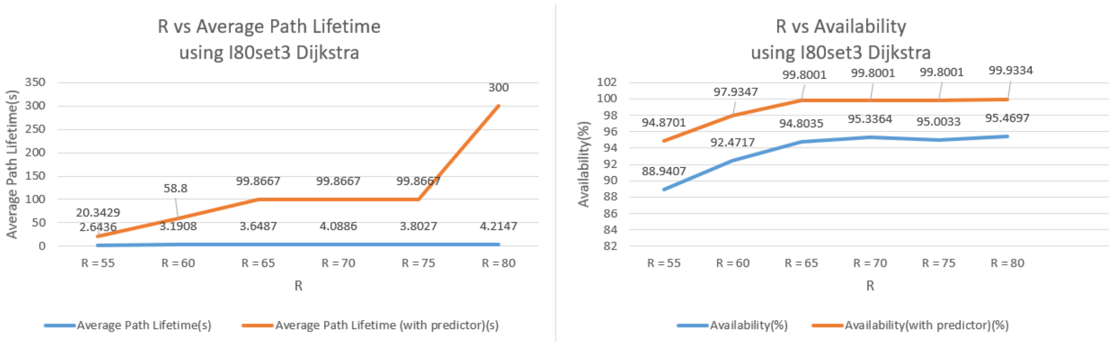


Figure 39: Transmission Range using I80set3 with MPA Dijkstra

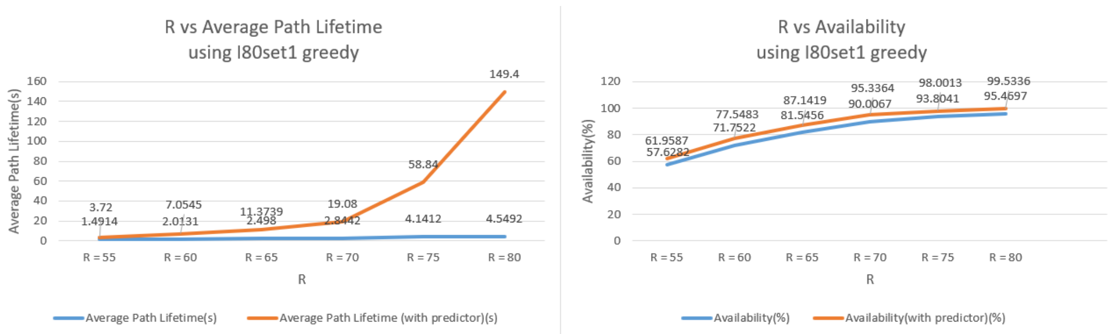


Figure 40: Transmission Range using I80set1 with MPA greedy

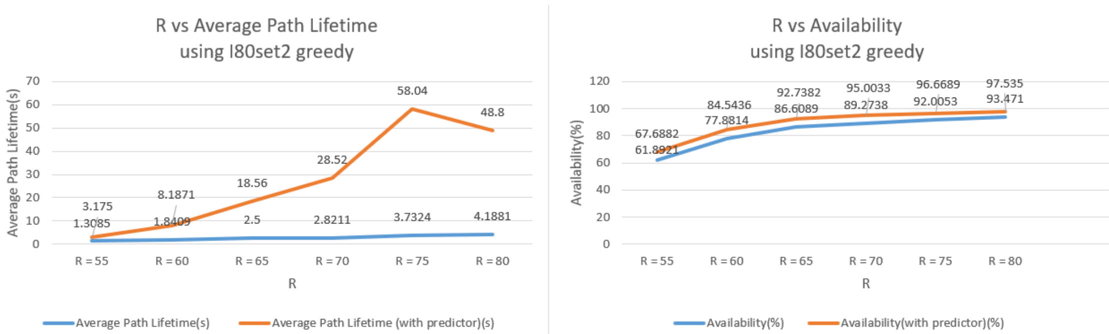


Figure 41: Transmission Range using I80set2 with MPA greedy

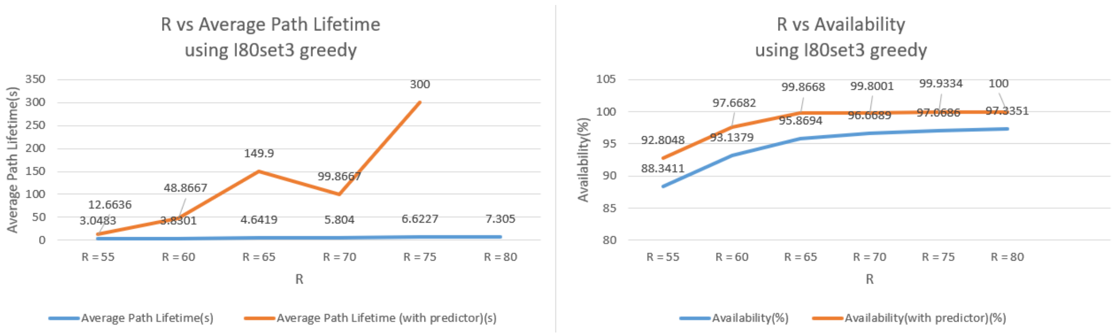


Figure 42: Transmission Range using I80set3 with MPA greedy

From the result above we can find out that the increasing transmission range improves the performance of the communication link enormously, which is a great benefit for VANET communication. We can imagine the advantage of the increasing transmission range easily, as the transmission range increase, the link between two vehicles can be maintained much longer, less new links need to find since the link is not easy to break, thus the average path lifetime increasing quickly.

Also, with the higher transmission range, less hop need to pass through from the source to the destination node, which increases the safety of the algorithm and decreases the delay of communication between each node. Also with the decrease of the requirement to find new communication link, the algorithm itself have a much lower delay. The following figure is the analysis of the algorithm delay vs Transmission Range.

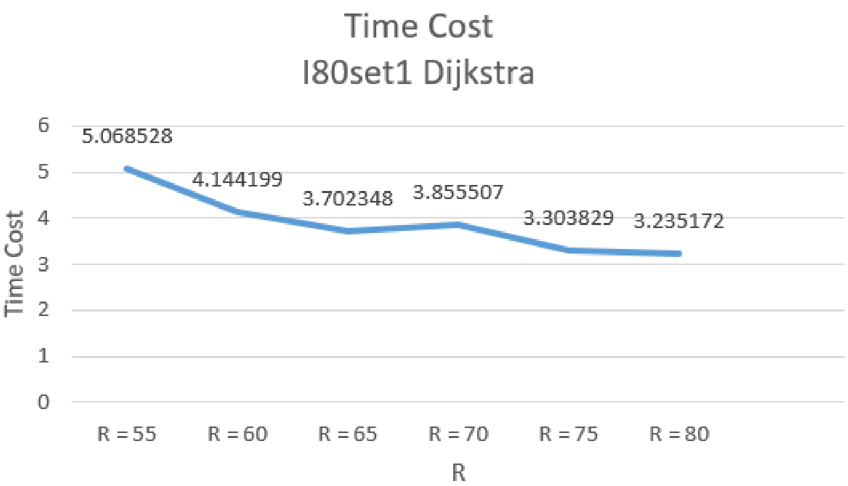


Figure 43: Transmission Range vs Time Cost using I80set1 with MPA Dijkstra

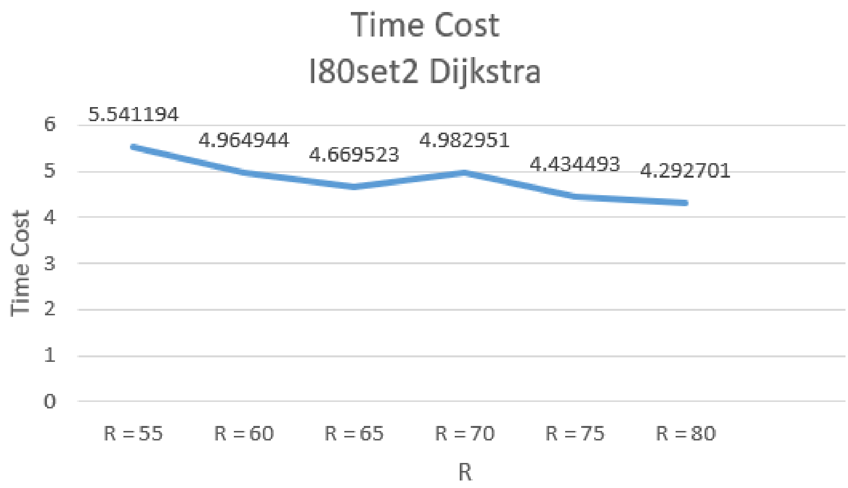


Figure 44: Transmission Range vs Time Cost using I80set2 with MPA Dijkstra

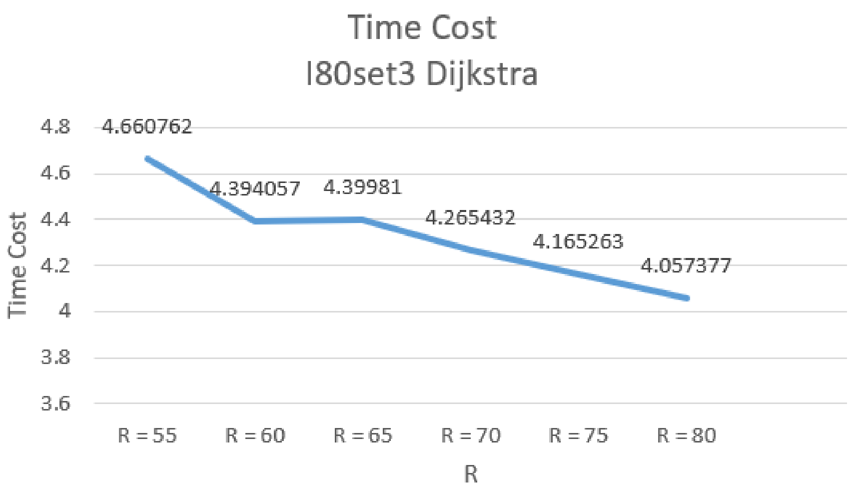


Figure 45: Transmission Range vs Time Cost using I80set3 with MPA Dijkstra

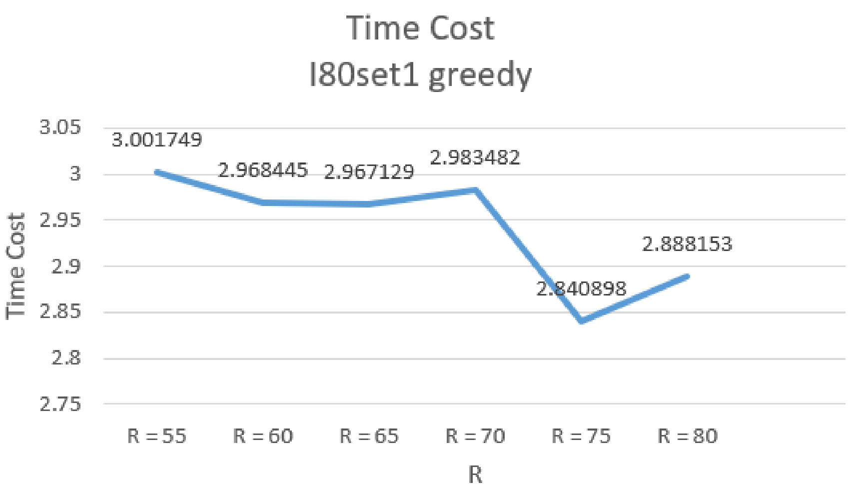


Figure 46: Transmission Range vs Time Cost using I80set1 with MPA greedy

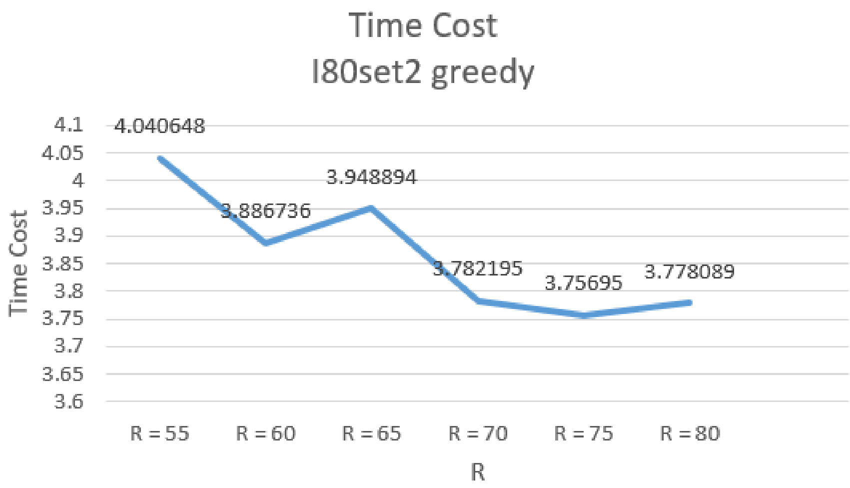


Figure 47: Transmission Range vs Time Cost using I80set2 with MPA greedy

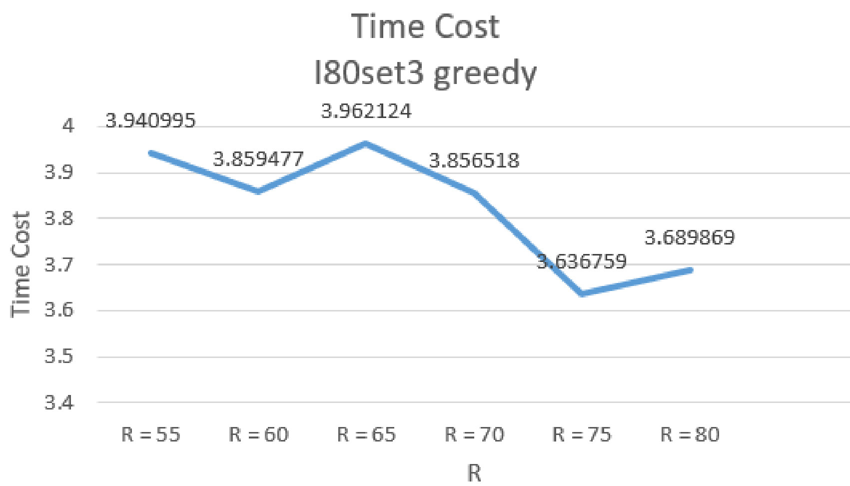


Figure 48: Transmission Range vs Time Cost using I80set3 with MPA greedy

From the result above, although the time cost of the algorithm is depending on the running machine and a lot of other conditions, the trending of the time cost is obviously decreasing with the increasing of Transmission Range. That proves our opinion, that we should choose the Transmission Range carefully when simulation on the different datasets depending on the different situations. The smaller or higher transmission range will not describe the correct performance of the dataset.

Also, the increasing of the Transmission Range between two nodes needs to be treated as one of the most important features for VANET communication in future works.

5.1.5 Conclusion on Parameters

The Parameter Stability Coefficient only has an impact on the Dijkstra algorithm. When it comes to the algorithm's performance, a Stability Coefficient below 20 is considered sensitive and unstable. To ensure the algorithm's general applicability, it is recommended to set the Stability Coefficient to 20 or 25.

The Parameter Danger Timeout, a higher Danger Timeout bring better Performance but comes with a higher Time Delay. Considering the balance of Performance and Time Delay, it is recommended to set the Danger Timour to 0.3.

The Parameter Velocity Threshold does not have a significant impact on the algorithm's performance or time delay. Therefore, selecting a suitable value for this parameter would be sufficient, as it does not require specific recommendations for optimization.

The selection of the Parameter Transmission Range depends on the advancements in communication equipment. A higher Transmission Range can greatly enhance performance values and substantially reduce Time Delay. Careful consideration should be given to choosing the Transmission Range for different datasets. In general, it is desirable to have the Transmission Range as large as possible in order to achieve optimal results.

5.2 Average Velocity vs. Algorithm Performance

As mentioned in the previous section, the difference between velocities is representative of road congestion. When the average velocity is higher, the mobility of the vehicles is higher, which means the road is less congested, and vice versa. Therefore, we can use average velocity to analyse the performance of the algorithm during different congestion, and decide whether Dijkstra or greedy algorithm works better during different situations.

This is the core idea for Automatically Selectable Maximum Path Affinity Algorithm(ASMPA) to always select the 'better' algorithm.

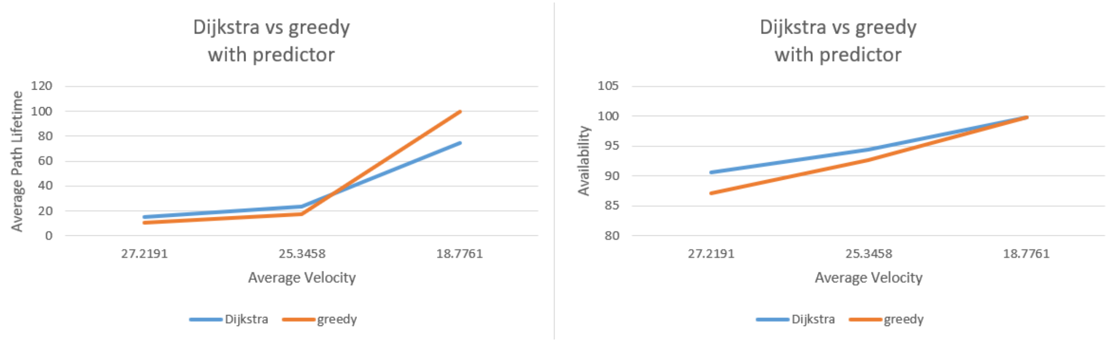


Figure 49: Average Velocity vs. Performance

From the result above, it is clear that with lower average velocity, the greedy algorithm works closer to the performance of the Dijkstra algorithm. Dijkstra algorithm always finds the best path in the whole global VANET communication network, but as the price, the Dijkstra algorithm always takes a larger time cost. Thus we want the Automatically Selectable Maximum Path Affinity Algorithm(ASMPA) algorithm select MPA greedy at low average velocity, and MPA Dijkstra at high average velocity. In this way, we can minimise the time cost without losing too much on performance, as in low average velocity, MPA greedy have similar performance with MPA Dijkstra, but a much lower time cost than MPA Dijkstra.

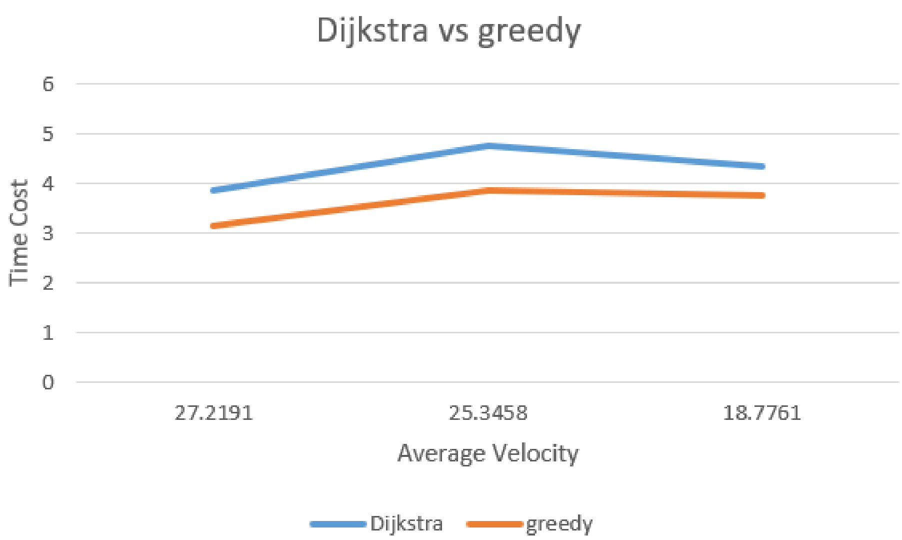


Figure 50: Average Velocity vs. Time Cost

In this project, this threshold is chosen to be 25 feet/second, lower than 25 feet/second, ASMPA will treat the road congestion as busy-situation, and over the threshold will be treated as relaxed-situation. In busy-situation, the topology of the VANET network is changing slowly, the communication link is not easy to break. Thus, using MPA greedy, although the established link affinity is not good as MPA Dijkstra, it does not affect a lot for the performance of the algorithm. However, in relaxed-situation, the topology of the network changes quickly, and the link is much easier to break. Therefore, choosing the global optimum solution is important for the percentage of availability. This is obviously in the result above, in the relaxed-situation, the percentage of availability for the MPA Dijkstra is much higher than MPA greedy. When the situation is closer to busy-situation, the percentage of availability for both algorithms is much more similar. Therefore, ASMPA should select the algorithm with lower time delay and similar performance during busy-situation using MPA greedy, and better performance during relaxed-situation using MPA Dijkstra.

There is one exception, when the average velocity is 18 feet/second, the average path lifetime of MPA greedy is higher than MPA Dijkstra, this is not normal because Dijkstra always finds the global best solution. This suggests that the current stage of the Dijkstra algorithm still does not simulate perfectly as the theoretical Dijkstra. The stability function still needs to be fine-tuned in cases the algorithm can describe which link should be selected more accurately.

5.3 Performance of ASMPA

The following figures will show the comparison of the ASMPA with algorithm MPA Dijkstra and MPA greedy separately, and it can show the benefit of the ASMPA clearly.

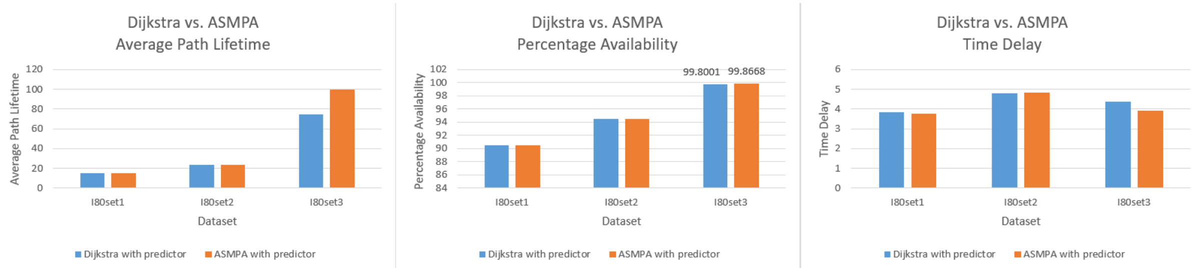


Figure 51: ASMPA vs. MPA Dijkstra

Compared with MPA Dijkstra, the performance of the ASMPA is not worse than MPA Dijkstra, also the delay of the ASMPA is lower than the MPA Dijkstra in situations that do not decrease the performance of the algorithm. We can say the ASMPA algorithm reduced time delay while maintaining performance than MPA Dijkstra.

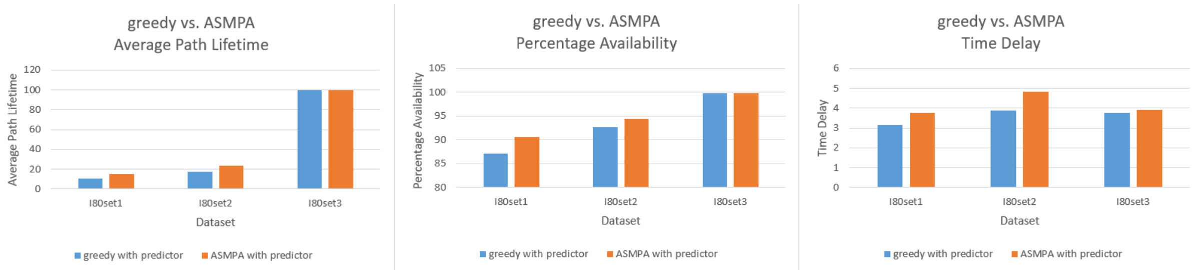


Figure 52: ASMPA vs. MPA greedy

The performance of the ASMPA is much better than MPA greedy in most cases, and although the time delay is higher than the MPA greedy, trying the best to keep the difference as low as possible. Once the performance cannot increase a lot, ASMPA will reduce the time delay. As a conclusion, ASMPA has better performance while controlling the time delay than MPA greedy.

In this case, we can say the ASMPA have better performance than the MPA greedy and lower time delay than the MPA Dijkstra. This is automatically selecting the algorithm that works better in general cases by combining the benefit of both algorithms. The following figure shows that the ASMPA always choose a more suitable algorithm.

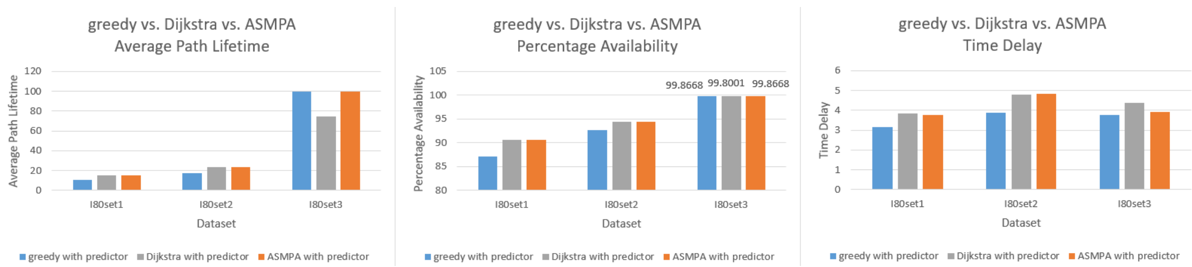


Figure 53: greddy vs. Dijkstra vs.ASMPA

The improvement of the ASMPA in different datasets I80set1, I80set2, and I80set3 over MPA DIjkstra and MPA greedy is summarised in the figure below. Min improvement is defined as the MINIMUM(MAXIMUM[(performance of ASMPA - performance of MPA Dijkstra), (performance of ASMPA - performance of MPA greedy)]). Using MAXIMUM to select the higher difference is because the ASMPA is based on MPA DIjkstra and MPA greedy. Thus, only the maximum difference should be considered as an improvement.

Similarly, Max improvement is defined as MAXIMUM(MAXIMUM[(performance of ASMPA - performance of MPA Dijkstra), (performance of ASMPA - performance of MPA greedy)]).

Average improvement is defined as AVERAGE(improvement in different dataset).

The improvement of ASMPA in Time Delay is similar but in the reverse calculation. The improvement of time delay is shown as the decrease in time delay. Therefore, Improvement of time delay is defined as MAXIMUM[(time delay of MPA Dijkstra - time delay of ASMPA), (time delay of MPA greedy - time delay of ASMPA)].

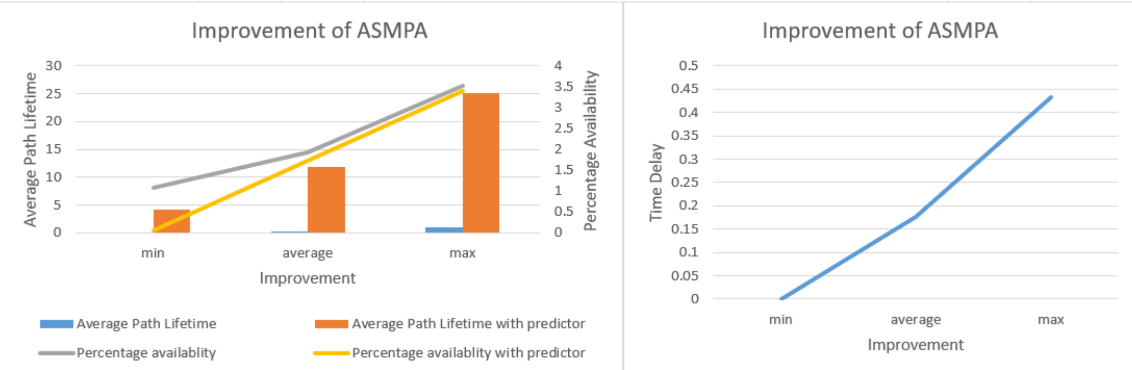


Figure 54: Improvement of ASMPA

6 Conclusion and Future Work

6.1 Conclusion

This project successfully investigates the background of VANET networks, conducts a comprehensive analysis of algorithm parameters, and proposes the Up-Edge Detection method, highlighting its significance for algorithms. Furthermore, the design of the Automatically Selectable Maximum Path Affinity Algorithm (ASMPA) achieves the intended objectives and surpasses previous algorithms. Comparative results between ASMPA and existing algorithms support this claim.

Based on the results, several suggestions and conclusions can be drawn. Firstly, the analysis of algorithm parameters provides valuable insights into parameter selection and emphasizes the importance of Transmission Range for algorithm performance and time delay.

Secondly, the Up-Edge Detection calculation method effectively addresses the issues encountered in previous methods, enhancing the reliability of algorithm results.

Finally, the newly developed ASMPA algorithm demonstrates the potential of combining the advantages of both Dijkstra and greedy algorithms. ASMPA introduces the concept of selecting different algorithms based on varying road congestion levels, which proves to be both feasible and beneficial.

6.2 Future Work

Due to time constraints in this project, there are several potential future works that can be pursued based on the findings. These include:

- Expanding the dataset: The project only utilizes a portion of the I80 dataset, specifically I80set1, I80set2, and I80set3. To obtain more reliable and generalizable conclusions, it is recommended to simulate the algorithms using the remaining data from the I80 dataset as well as other datasets mentioned, such as the US Highway 101 Dataset and Lankershim Boulevard Dataset. Utilizing a wider range of datasets will provide a more comprehensive analysis of parameters and allow for better threshold selection in the ASMPA algorithm.

- Improving the MPET algorithm: While the focus of this project is on the MPA algorithm, similar methods can be applied to enhance the MPET algorithm as well. By implementing comparable enhancements, the performance of the MPET algorithm can be further improved.

- Exploring additional algorithms in ASMPA: Although ASMPA currently selects between Dijkstra and greedy algorithms, there are numerous other algorithms that can be incorporated into the selection process, such as the A* algorithm. Including a broader range of algorithms in ASMPA would provide more flexibility and potentially lead to even better performance outcomes.

- Extending the concept of algorithm selection: The idea of selecting different algorithms based on varying road congestion levels, as demonstrated in ASMPA, can be applied to other algorithms such as selecting algorithm MPA and MPET depending on different situations. By analyzing and identifying situations where specific algorithms are more effective, this approach can be expanded to improve the performance of various algorithms.

- Introducing additional performance metrics: While the project evaluates the algorithms using metrics like Average Path Lifetime, Percentage Path Availability, and Time Delay, it is beneficial to include other metrics for a more comprehensive evaluation. Metrics such as throughput can be introduced to provide a broader understanding of the algorithms' effectiveness.

7 Reference

References

[1] Reginald Leslie Smith-Rose, Guglielmo Marconi. Available from: *https://www.britannica.com/biography/Guglielmo-Marconi*

[2] Why wireless standards are so important in today's world, 15 Mar 2019. Available from: *https://www.itu.int/hub/2019/03/why-wireless-standards-are-so-important-in-todays-world/*

[3] Kenechi Okeleke, Stefano Suardi, The Mobile Economy 2022, Mar 2022. Available from: *https://data.gsmaintelligence.com/research/research/research-2022/the-mobile-economy-2022*

[4] Divya Chadha, Reena, Vehicular Ad hoc Network (VANETs): A Review, March 2015. Available from: *https://www.rroij.com/open-access/vehicular-ad-hoc-network-vanets-a-review.pdf*

[5] Road traffic injuries, 20 June 2022. Available from: *https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries*

[6] Reported road casualties in Great Britain, provisional estimates: year ending June 2022, 24 November 2022. Available from: *https://www.gov.uk/government/statistics/reported-road-casualties-in-great-britain-provisional-estimates-year-ending-june-2022/reported-road-casualties-in-great-britain-provisional-estimates-year-ending-june-2022*

[7] Vehicle Safety Systems. Available from: *https://transport.ec.europa.eu/transport-themes/intelligent-transport-systems/road/application-areas/vehicle-safety-systems_en*

[8] Regulation (EC) No 661/2009. Available from: *https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32009R0661*

[9] H. Hartenstein, A tutorial survey on vehicular ad hoc networks, IEEE, 06 June 2008. Available from: *https://ieeexplore.ieee.org/document/4539481*

[10] INRIX: Congestion Costs Each American 97 hours, $1,348 A Year. Available from: *https://inrix.com/press-releases/scorecard-2018-us/*

[11] INRIX Global Traffic Scorecard: Congestion cost UK economy £6.9 billion in 2019. Available from: *https://inrix.com/press-releases/2019-traffic-scorecard-uk/*

[12] Xun Yong Lee, Location-based Routing Algorithms for Mobile Ad Hoc Networks, Imperial College London, 2014

[13] Regulation (EU) 2016/679, General Data Protection Regulation (GDPR). Available from: *https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02016R0679-20160504*

[14] FHWA. Traffic Analysis Tools Program: Next Generation Simulation (NGSIM) Available from: *http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm*

[15] Wireless network, Wikipedia. Available from: *https://en.wikipedia.org/wiki/Wireless_network*

[16] Wireless ad hoc network, Wikipedia. Available from: *https://en.wikipedia.org/wiki/Wireless_ad_hoc_network*

[17] Wired and wireless connectivity. Available from: *https://www.bbc.co.uk/bitesize/guides/zj88jty/revision/4*

[18] CHRIS HOFFMAN, What's the Difference Between Ad-Hoc and Infrastructure Mode Wi-Fi? 2016. Available from: *https://www.howtogeek.com/180649/htg-explains-whats-the-difference-between-ad-hoc-and-infrastructure-mode/*

[19] By James T Wood, The Advantages of Ad Hoc Networks. Available from: *https://itstillworks.com/advantages-ad-hoc-networks-6793071.html*

[20] Embedded Staff, Securing wireless ad hoc networks: Part 1 – single and multi-hop ad hoc networks, 2007. Available from: *https://www.embedded.com/securing-wireless-ad-hoc-networks-part-1-single-and-multi-hop-ad-hoc-networks/l*

[21] Multi-hop routing, Wikipedia. Available from: *https://en.wikipedia.org/wiki/Multi-hop_routing*

[22] Mobile Adhoc Network (MANET). Available from: *https://www.javatpoint.com/mobile-adhoc-network*

[23] Diaa Eldein Mustafa Ahmed, Othman O. Khalifa, An Overview of MANETs: Applications, Characteristics, Challenges and Recent Issues, ISSN: 2249 − 8958, Volume-6 Issue-4, April 2017. Available from:
https://www.researchgate.net/publication/316257334_An_Overview_of_MANETs_Applications_Characteristics_Challenges_and_Recent_Issues

[24] raman_257, MANET Routing Protocols, Mar, 2023. Available from:
https://www.geeksforgeeks.org/manet-routing-protocols/

[25] Bellman Ford's Algorithm. Available from: *https://www.programiz.com/dsa/bellman-ford-algorithm*

[26] Dijkstra's Algorithm. Available from: *https://www.programiz.com/dsa/dijkstra-algorithm*

[27] kaurbal1698, Difference between MANET and VANET. Available from: *https://www.geeksforgeeks.org/difference-between-manet-and-vanet/*

[28] Samual Sam, Medium Access Control Sublayer (MAC sublayer), 2019. Available from: *https://www.tutorialspoint.com/medium-access-control-sublayer-mac-sublayer*

[29] Mengyuan Ma, Kai Liu, Xiling Luo, Tao Zhang, Feng Liu, Review of MAC Protocols for Vehicular Ad Hoc Networks, 2020. Available from: *https://mdpi-res.com/sensors/sensors-20-06709/article_deploy/sensors-20-06709-v2.pdf?version=1606299093*

[30] Greedy Algorithm. Available from: *https://www.programiz.com/dsa/greedy-algorithm*

[31] FHWA. US Highway 101 Dataset, FHWA-HRT-07-030, 2007. Available from: *https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm*

[32] FHWA. Lankershim Boulevard Dataset, FHWA-HRT-07-029, 2007. Available from: *https://www.fhwa.dot.gov/publications/research/operations/07029/index.cfm*

[33] FHWA. Interstate 80 Freeway Dataset, FHWA-HRT-06-137, 2006. Available from: *https://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm*

[34] Pythagorean theorem, Wikipedia. Available from: *https://en.wikipedia.org/wiki/Pythagorean_theorem*

[35] Trigonometric functions, Wikipedia. Available from: *https://en.wikipedia.org/wiki/Trigonometric_functions*

[36] William Su, Sung-Ju Lee, Mario Gerla, Mobility prediction and routing in ad hoc wireless networks, 2001. p.2 Available from: *https://onlinelibrary-wiley-com.iclibezp1.cc.ic.ac.uk/doi/pdfdirect/10.1002/nem.386*

[37] J.A. Barria, R. Lent, MANET Route Discovery using Residual Lifetime Estimation, 2006. ISBN:0-7803-9410-0. Available from: *https://ieeexplore.ieee.org/document/1613589*

[38] Rachmawati, Dian ; Gustin, Lysander, Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem, 2020. ISSN: 1742-6588. Available from: *https://iopscience.iop.org/article/10.1088/1742-6596/1566/1/012061*

[39] Rachit Belwariar, A* Search Algorithm, 2023. Available from: *https://www.geeksforgeeks.org/a-search-algorithm/*

8 Appendix

8.1 Necessary Material

You can find the working file on GitHub using the provided link URL: *https://github.com/czyhq/Topology-changes-prediction-for-proactive-location-based-routing-in-VANETs.git*. The provided link URL contains all the necessary materials required to execute the simulation.

8.2 Stability coefficient vs. Performance using MPA greedy

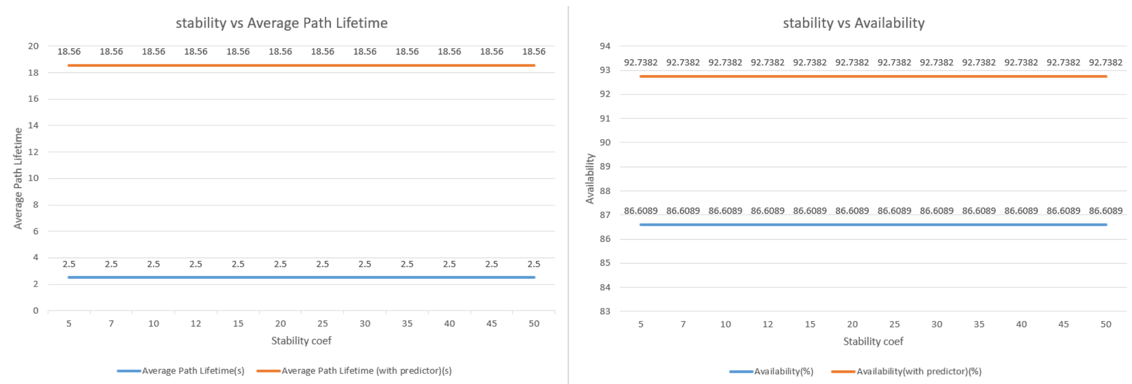


Figure 55: Stability coefficient vs. Performance using dataset I80set2 with MPA greedy

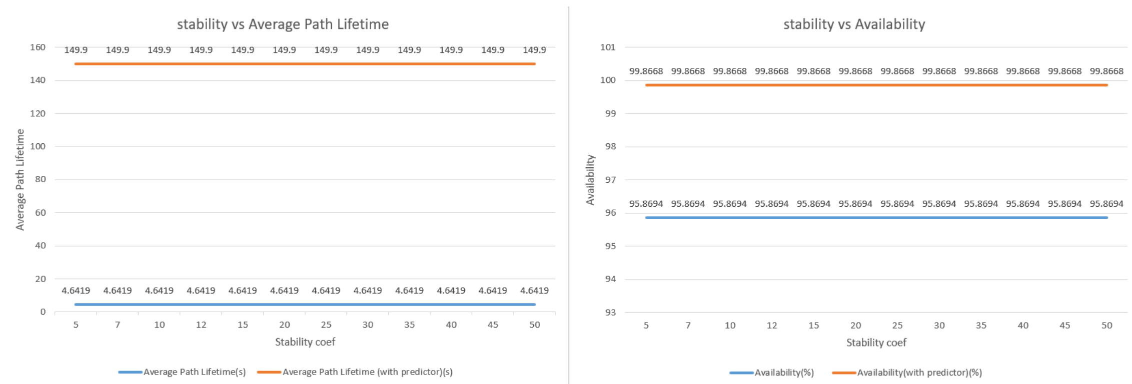


Figure 56: Stability coefficient vs. Performance using dataset I80set3 with MPA greedy

8.3 Analysis of Velocity Threshold

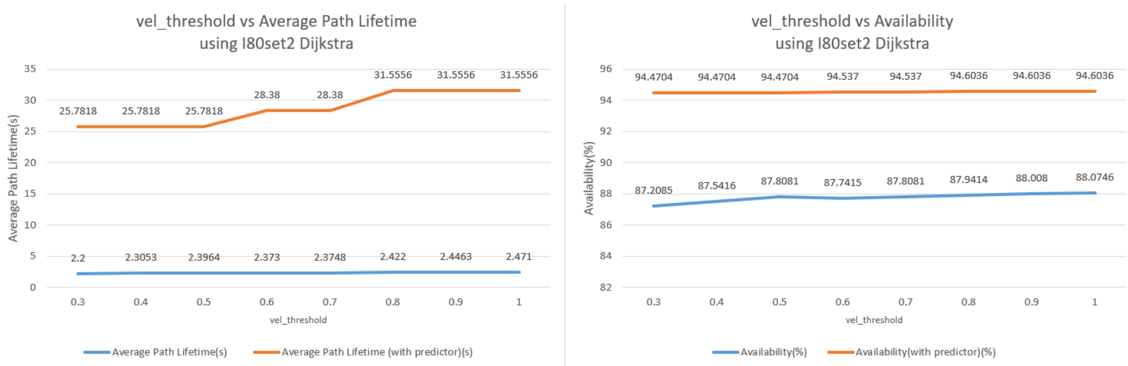


Figure 57: Velocity Threshold vs Average Path Lifetime and Percentage Path Availability using Dataset I80set2 with algorithm MPA Dijkstra

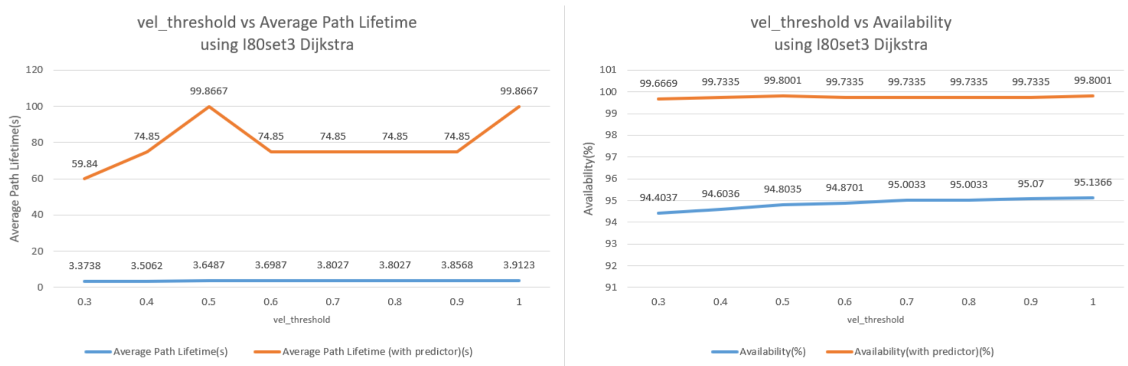


Figure 58: Velocity Threshold vs Average Path Lifetime and Percentage Path Availability using Dataset I80set3 with algorithm MPA Dijkstra

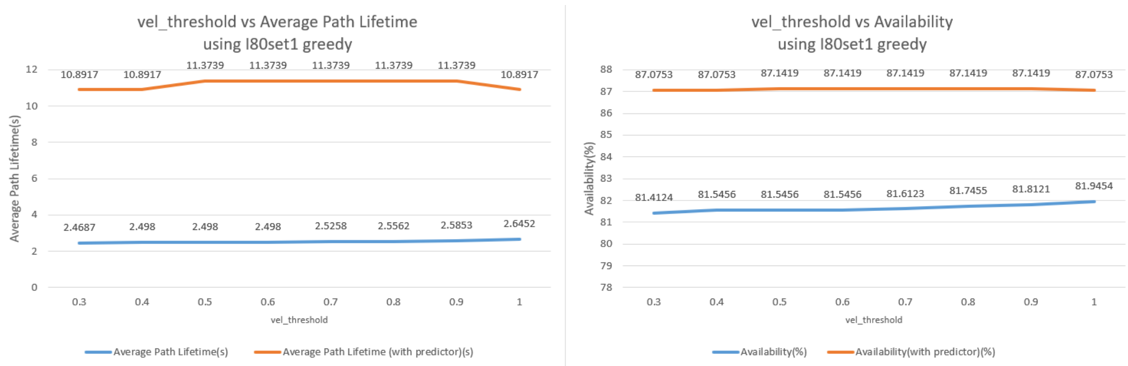


Figure 59: Velocity Threshold vs Average Path Lifetime and Percentage Path Availability using Dataset I80set1 with algorithm MPA greedy

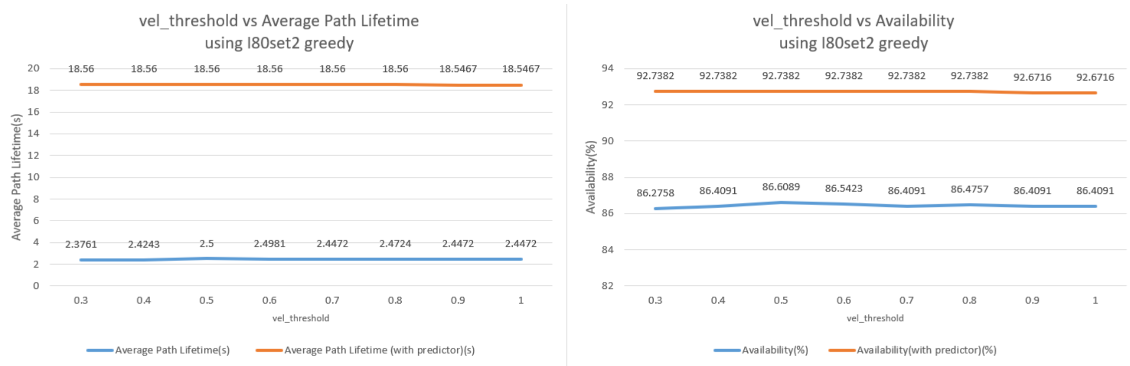


Figure 60: Velocity Threshold vs Average Path Lifetime and Percentage Path Availability using Dataset I80set2 with algorithm MPA greedy

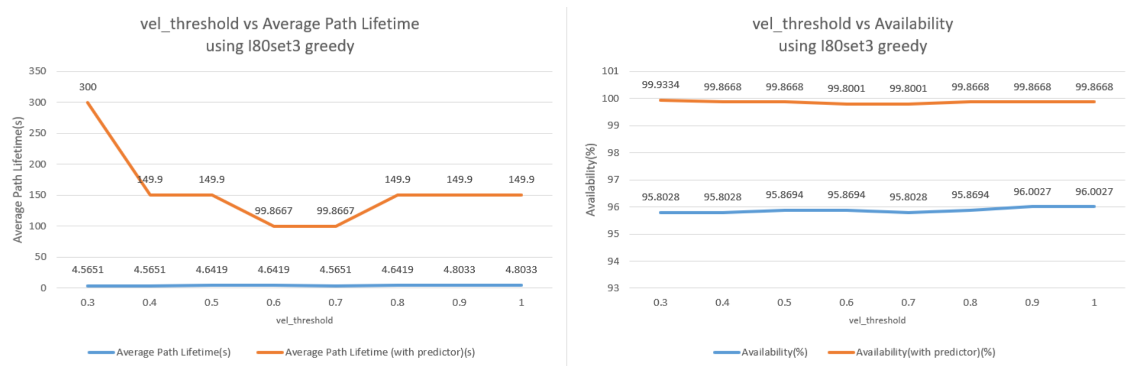


Figure 61: Velocity Threshold vs Average Path Lifetime and Percentage Path Availability using Dataset I80set3 with algorithm MPA greedy