Instituto Tecnologico y de Estudios Superiores de Monterrey

Campus Estado de México

School of Engineering and Sciences

A Deep Learning-based Algorithm for the Routing Problem in Vehicular Delay-Tolerant Networks

A dissertation presented by

Roberto Hernández Jiménez

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Engineering Sciences

Atizapán, Estado de México, June 11th, 2020

Dedication

I want to dedicate this work to God, in the first place, because nothing would have been possible without him.

This thesis is dedicated to my family and friends. First and foremost, to my loving wife, Guille Cruz, because she helped me get through the hard times and was always there to patiently support me and help me when I was about to give up. To my mother, Ángela Hernández, because undoubtedly, she let me see that education is the main avenue for progress both individually and as part of a society. To my sister Ana and my brothers Jorge and Manuel, for being always there for me with their unconditional support. To my parents-in-law, Guille Miranda and Facundo Cruz, for their guidance and inspiration to be the best version of myself, and for the whole Cruz Miranda, Hernández Cruz (primeros), Treviño Miranda, Varela, Cruz Gutiérrez and Cruz Luna families, for inspiring me to be a better person and for reminding me of the importance of family values. To my many friends, colleagues and people that move me to be a better person every day, my warmest gratitude to them all.

Acknowledgements

I want to express my deepest gratitude to my advisor, Dr. César Cárdenas Pérez, for giving me his support and guidance through this process and to Dr. David Muñoz Rodríguez, because their feedbacks and suggestions were key in this thesis and related works. Also, I want to thank the committee members, Dr. Juan Humberto Sossa Azuela, Dr. Martin Rogelio Bustamante Bello and Dr. Miguel González Mendoza, for taking the time to read and review this report, and certify that it is fully adequate in scope and quantity as a partial requirement for the degree of Doctor of Philosophy in Engineering Sciences. Finally, special thanks to Tecnologico de Monterrey and Consejo Nacional de Ciencia y Tecnología (CONACYT) for their academic and financial support.

A Deep Learning-based Algorithm for the Routing Problem in Vehicular Delay-Tolerant Networks

by

Roberto Hernández

Abstract

The exponential growth of cities across the world has brought along important challenges such as waste management, pollution and overpopulation, and transportation administration. To mitigate these problems, the idea of Smart City was born, seeking to provide robust solutions integrating sensors and electronics, information technologies and communication networks. More particularly, to face transportation challenges, Intelligent Transportation Systems are a vital component in this quest. Intelligent Transportation Systems are intelligent systems that aim at providing the best solution to transportation-related matters, with the aid of information technologies, electrical and electronics and communication networks. In this context, communication networks are called Vehicular Networks, and they offer a communication framework for moving vehicles, road infrastructure and pedestrians. The extreme conditions of vehicular environments, nonetheless, make communication between high-speed moving nodes very difficult, so non-deterministic approaches are necessary to maximize the chances of packet delivery. In this work, this problem is addressed using Artificial Intelligence from a hybrid perspective, focusing on both the best next message to replicate and the best next hop in its path in the network. Furthermore, DLR+ is proposed, a router with a prioritized type of message scheduler and a routing algorithm based on Deep Learning. Simulations done to assess the router performance show important gains in terms of network overhead and hop count, while maintaining an acceptable packet delivery ratio and delivery delays, with respect to other popular routing protocols in vehicular networks.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This first chapter begins with a preamble to the methodology used in the research process, and then follows directly into the main topic, which is routing protocols in vehicular delay tolerant networks.

Research Methodology

The methodology used in this research is based on the Design Science in Information Systems Research Framework proposed by Hevner et al. [31]. Such methodology (fig. 1.1) identifies two science paradigms based on behavioral and design science, both of which are essential to the Information Systems (IS) and can be used to make significant contributions to IS Research. The behavioral-science paradigm seeks to develop and verify theories or predict human or organizational behavior, whereas the design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts.

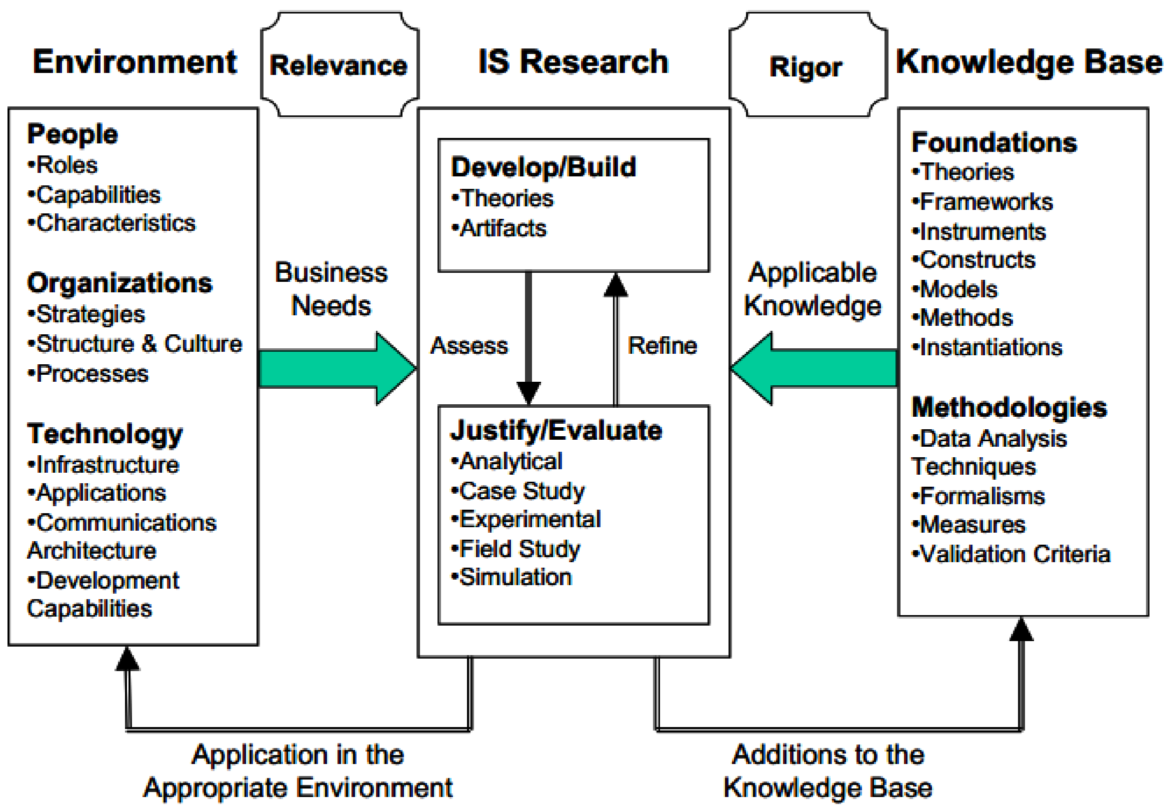


Figure 1.1. Design-science for Information Systems Research Framework.

According to the authors, the design-science paradigm has its roots in engineering and the science of the artificial, and it is fundamentally a problem-solving paradigm, seeking to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management and use of IS can be effectively and efficiently accomplished. The research in this paper is based precisely on the design of a new artifact that represents a significant contribution to the solution to an identified problem. As mentioned in [31], artifacts may vary from software, formal logic, and rigorous mathematics to informal natural language description, whose evaluation may include optimization proofs, analytical simulation, and quantitative comparisons with alternative designs. Figure 1.1 depicts this IS Research Framework.

In this framework, the need or research problem is addressed from the two mentioned perspectives: the behavioral-science (used to develop and justify the theories that explain or predict phenomena related to the identified problem; it seeks truth) and the design-science (used to build and evaluate artifacts designed to meet that need; it seeks utility). The whole research process referred to in this paper will make use of this assessment approach in such a way that the *justify/evaluate* activities can result in the identification of weaknesses in the theory or artifact and thus refine and reassess as needed. Relevance and rigor are accomplished as the proper foundations (theories, frameworks, models, etc.) and methodologies (data analysis techniques, validation criteria, etc.) are applied to the research process to solve the identified problem in the environment of interest.

As design-science is a problem-solving process, there are seven guidelines to follow through to guarantee a rigorous and relevant research in IS (Table 1.1). As stated by Hevner et al [31], the research artifact must meet the following criteria: have a purpose, be problem-relevant, be evaluated, be innovative, be rigorous, be properly selected, and be properly communicated. The design-science research requires the creation of an innovative purposeful artifact (guideline 1) to a specified problem domain (guideline 2), in which evaluation (guideline 3) is crucial to validate the purposefulness and utility of the artifact.

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 1.1. Detailed description of the guidelines of the Design-Science in IS Research framework.

Also, innovation is crucial (guideline 4) in order to be able to differentiate the design-science from the practice of design. Rigor, presentation and consistency (guideline 5) are characteristics of the artifact, which must be properly selected (guideline 6) from a variety of available means to reach the desired end. And finally, the results of the design-science research must be effectively communicated to the proper audience, both technical and managerial (guideline 7). Table 1.1 provides a more detailed view of the mentioned guidelines.

Being packet routing one of the main challenges in such environments, in the context of Smart Cities (guideline 2), in this research, a method for the packet routing in Vehicular Delay Tolerant Networks using a deep learning architecture to optimize the quality of delivery is presented (guidelines 1, 4 and 6). Furthermore, the efficacy of the proposed routing protocol was evaluated through suitable and rigorous simulations of the vehicular environment (guidelines 3 and 5). Finally, the research process and the obtained results have been properly presented in several research papers (Appendix) and in this report (guideline 7).

The rest of this Chapter focuses on the introduction of background concepts essential for the understanding of the identified problem, which leads to a more detailed discussion in Chapter 2.

Smart Cities

In the last decades cities around the world have grown formidably, concentrating a big percentage of the total population. Globally, more people live in urban areas than in rural areas, with 55.3% of the world's population residing in urban areas in 2018, according to the United Nations [73]. By 2030, urban areas are projected to house 60 per cent of people globally and one in every three people will live in cities with at least half a million inhabitants and it is estimated that by 2050 this number will rise to 68 percent of the world's population to be urban, with over 2.5 billion people living in cities.

This exponential growth of cities is considered in the 2030 Agenda for Sustainable Development, including Sustainable Development Goal 11, to make cities and human settlements inclusive, safe, resilient and sustainable [73]. One of the biggest concerns regarding the fast growth of cities is their administration and the capacity to give the best attention to their residents. Sustainability, security, waste management, information, traffic, etc. are important issues that need to be addressed in the administration of big cities. One of the most general ideas to address these concerns derived from the huge growth of urban areas mainly focus on applying the next-generation information technology to all walks of life, embedding sensors and equipment to hospitals, power grids, railways, bridges, tunnels, roads, buildings, water systems, dams, oil and gas pipelines and other objects in every corner of the world, and forming the "Internet of Things" via the Internet [70] This paradigm was defined as a "Smart City", representing the use of information and communication technology to sense, analyze and integrate the key information of core systems in running cities. At the same time, a smart city can make intelligent responses to different kinds of needs, including daily livelihood, environmental protection, public safety and city services, industrial and commercial activities [70]. In general, there are many definitions of Smart City, but they all encompass the same idea of sustainable development of cities with the aid of technology. An acceptable definition of Smart City considers that a city can be defined as 'smart' when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic development and a high quality of life, with a wise management of natural resources, through participatory governance [5]. Moreover, the Smart City is based on six key components including economy, governance, people, mobility, living and environment (Figure 1.2) [5][44].

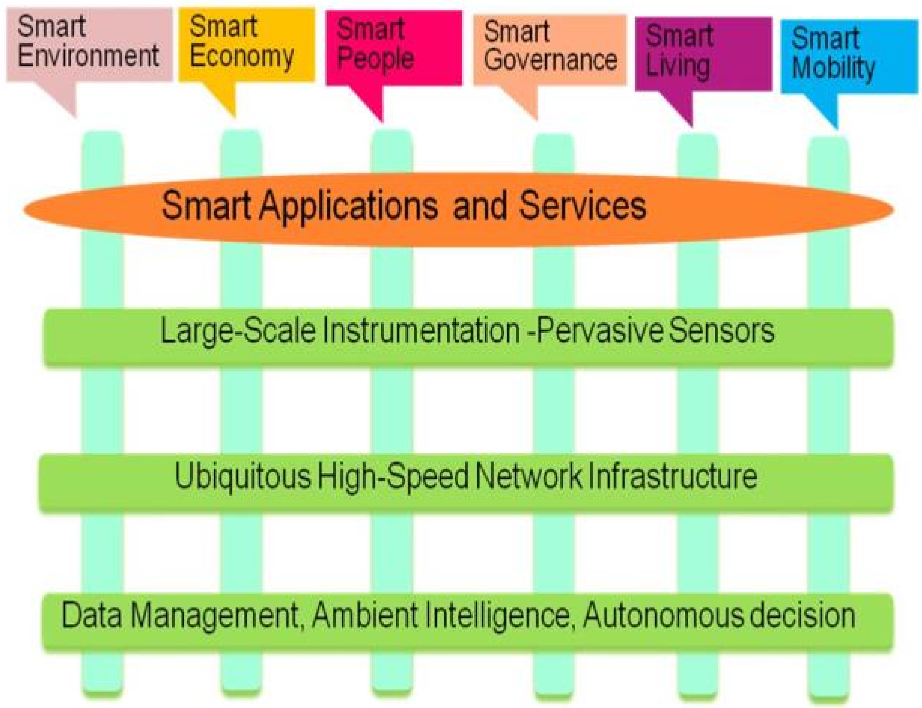


Figure 1.2. Building blocks of Smart City Architecture.

Having such a model of city come to life can make the future world increasingly appreciable and measurable, increasingly interconnected and interoperable and increasingly intelligent [70]. As Smart City is considered to be the future trend of urban development, its construction can generally be divided into three levels, including the construction of public infrastructure, construction of public platform for smart city and the construction of application systems, being the later typically applied to several aspects including the construction of Smart Medical Treatment, the construction of Smart Tourism, the construction of Green City, the construction of Smart Urban Management, the construction of Smart Public Services and Social Management, the construction of a Wireless City, the construction of Smart Home and the construction of Smart Transportation (Figure 1.3)[70].

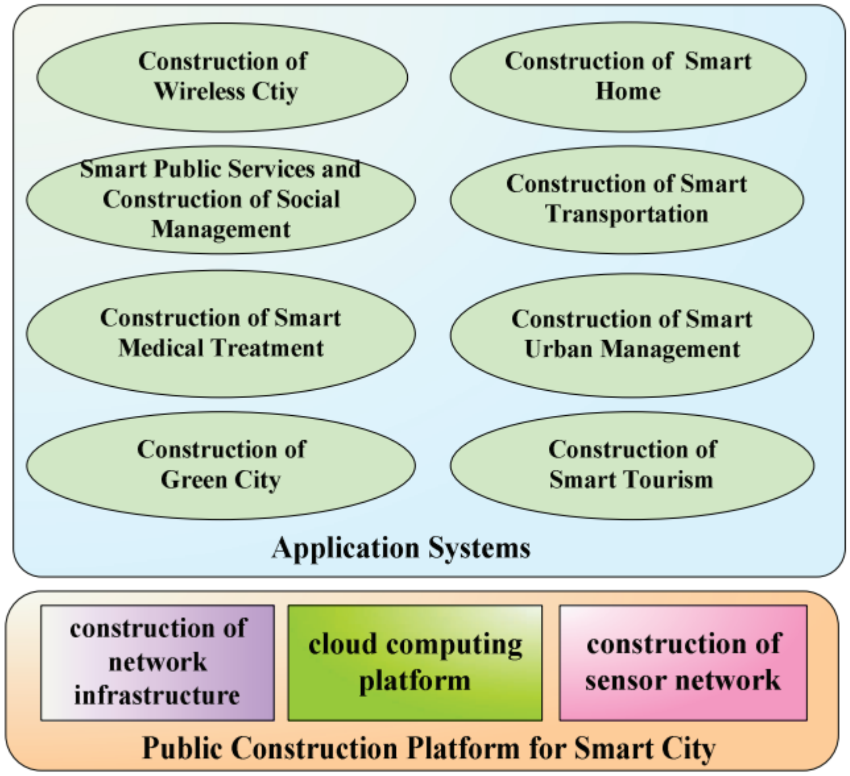


Figure 1.3. Construction frame of Application Systems for Smart cities.

Another popular proposal for the building blocks of the architecture of a Smart City is shown in Figure 1.4. [59] We can see that in the two first blocks Transportation Systems are considered, as "Transport Network" (bottom block) and "Transport Services" (second block) mention it as key components in the architecture. Thus, Transportation is an important element in this paradigm, and in the next section we are going to dig deeper in a second paradigm within Smart Cities.

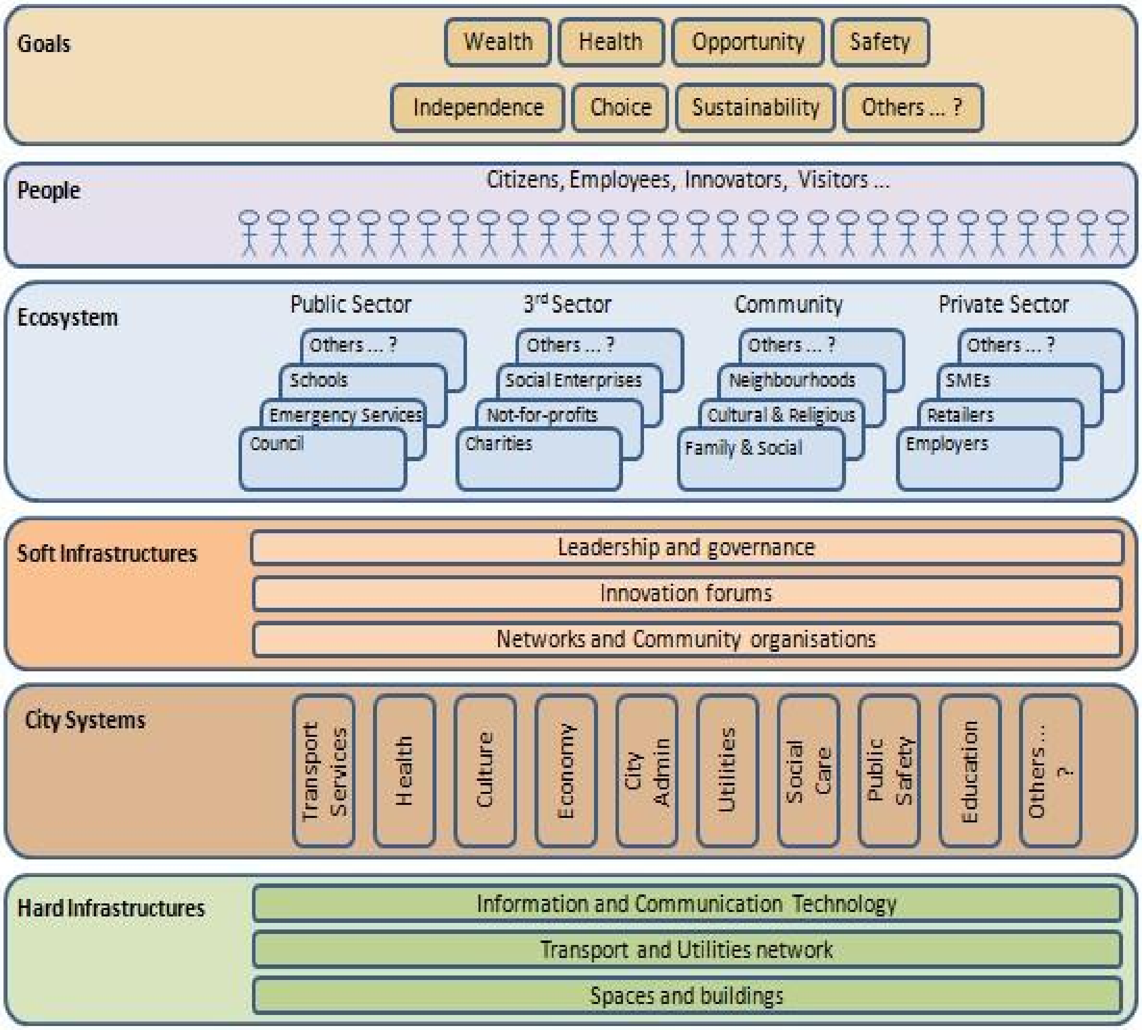


Figure 1.4. Building blocks of a Smart City, as proposed in [59].

ITS: a transportation paradigm for big cities.

The expansion of cities directly impacts the mobility and brings along an increase in the number of vehicles and the need of more communication means or an alternate transport system that makes infrastructure more efficient. Transport in big cities is one of the main concerns inherited from the accelerated growth of these metropolis and the number of inhabitants. More and more vehicles are sold every year and the streets suffer severe traffic jams, especially in large cities, where distances are bigger and consequently, vehicles are even more necessary for transportation purposes [13]. In particular, traffic congestions, emergencies and accidents reveal inefficiencies in transportation infrastructures. As part of the goals of a Smart City, Smart Transportation has a key role addressing this urban problematic. Even before this idea of Smart Cities was born, the concept of Intelligent Transportation System (ITS) has been developed since the beginning of 1970s, which makes human, vehicles and roads united and harmonic and establishes a wider range, fully efficient, real-time and accurate information manage system [56]. Such intelligent systems bring an increase in safety, efficiency and reliability to the actual systems, and make them more environmentally friendly [56][78]. A wide number of services and transport applications that provide security and comfort to the passengers can be enabled thanks to the ITS, including but not limited to electronic toll collection, traffic surveillance and management, collision avoidance, dynamic traffic optimization, information, news, online games, music, movies and others [13][71][78][56]. In fact, ITS general applications can be divided into the following [8]:

- Advanced Travel Information System (ATIS), designed to make travel more efficient and safer with information on congestion, navigation and location, weather and traffic conditions, and alternative routing.
- Advanced Traffic Management Systems (ATMS), designed for highway management and traffic control systems.
- Advanced Public Transportation System (APTS), designed to improve the operation of mass transit services.
- Advanced Vehicle Control and Safety System (AVCSS), designed to achieve efficiency and safety inside the vehicle.
- Commercial Vehicle Operation (CVO), designed to effectively manage taxi and truck fleet operations by controlling alternative routing and time of transport delivery system

Three key components are needed in order to sustain the functionality of such intelligent systems: sensors (required for data collection), communications (data, audio and video to and from vehicle to other units in the network) and Information Technologies (software, hardware, database management, etc.). However, in order for ITS to provide support for such a wide range of services, the communication systems must allow communication between vehicles (V2V) and between vehicles and infrastructure (V2I) [13]. Furthermore, more recently, with the rise of the Internet of Things (IoT), vehicle to anything (V2X) and vehicle to pedestrian (V2P) communications are also considered key components of communication in a fully connected transportation system (Figure). Thus, the aim for communication systems is to be very complete and complex, but at the same time they must be safe, stable and reliable [52], which is naturally difficult to achieve due to the high mobility constrains that vehicles present with respect to each other and with respect to fixed infrastructure, but there is where other disciplines come in into play.

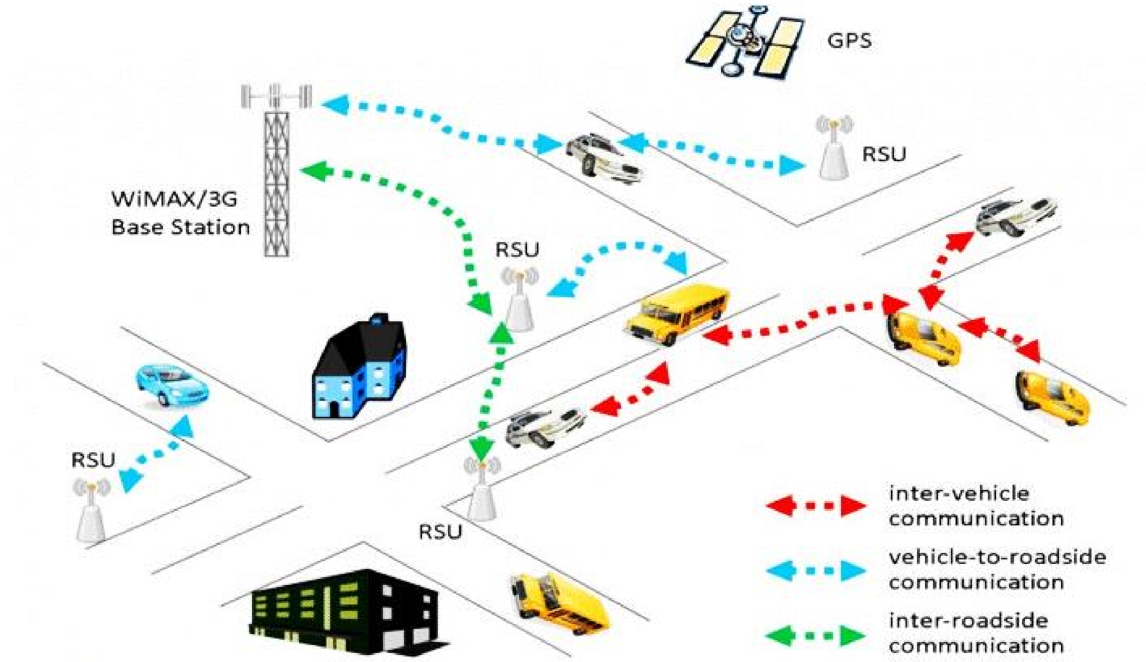


Figure 1.5. Vehicular communications in an ITS.

Communication Networks in ITS

Vehicular Communication is seen as a key technology for addressing problems in ITS due to its potential to improve the safety and comfort through various applications built upon it [52]. In order to allow that, seamless and heterogeneous communication must be possible, between any component in the network, namely, cars (V2V), infrastructure (V2I), pedestrians (V2P), and virtually any object capable of transmission or reception of data (V2X).

Applications and services in vehicular networks

Applications and services in vehicular networks aim at utilizing the information coming from vehicles to improve the efficiency in transport systems as well as security and safety of the passengers. In general, experts in the field have classified V2X services and applications in safety-related use cases, and non-safety related use cases, both with particular conditions, such as frequency and minimum latency. Table 1.2, summarizes safety-related use cases, and Table 1.3 lists the main uses cases for non-safety-related use cases.

Safety Service	Use-case	Type	Communication Mode	Minimum Frequency	Minimum Latency
Vehicle status warning	Emergency electronic brake lights	DEN/V2X	Time limited, event-based, periodic broadcast	10 Hz	100 ms
	Abnormal condition warning	DEN/V2X	Time limited, event-based, periodic broadcast	1 Hz	100 ms
Vehicle type warning	Emergency vehicle warning	CAM/V2X	Periodic broadcast, vehicle-mode dependent	10 Hz	100 ms
	Slow vehicle warning	CAM/V2X	Periodic broadcast, vehicle-mode dependent	2 Hz	100 ms
	Motorcycle warning	CAM/V2X	Periodic broadcast	2 Hz	100 ms
	Vulnerable road user warning	CAM/ VRU2X	Periodic broadcast	1 Hz	100 ms
Traffic hazard warning	Wrong way driving warning	DEN/V2X	Time limited, event-based, periodic broadcast	10 Hz	100 ms
	Stationary vehicle warning	DEN/X2X	Time limited, event-based, periodic broadcast	10 Hz	100 ms
	Traffic condition warning	DEN/X2X	Time limited, event-based, periodic broadcast	1 Hz	100 ms
	Signal violation warning	DEN/I2X	Time limited, event-based, periodic broadcast	10 Hz	100 ms
	Roadwork warning	DEN/I2X	Time limited, event-based, periodic broadcast	2 Hz	100 ms
Dynamic vehicle warning	Overtaking vehicle warning	DEN/V2X	Time limited, event-based, periodic broadcast	10 Hz	100 ms
	Lange change assistance	DEN/V2X	Time limited, event-based, periodic broadcast	10 Hz	100 ms
	Pre-crash sensing warning	DEN/V2X	Time limited, event-based, periodic broadcast	10 Hz	50 ms
	Co-operative glare reduction	DEN/V2X	Time limited, event-based, periodic broadcast	2 Hz	100 ms

Table 1.2. Safety-related use-cases. DEN stands for Decentralized Environmental Notification, CAM for Cooperative Awareness Message (adapted from [18]).

Non-safety services	Use-case	Type	Communication mode	Minimum frequency	Minimum latency
Traffic Management	Speed limits	I2V	Periodic broadcast	1 Hz	100 ms
	Traffic light optimal speed advisory	I2V	Periodic broadcast	2 Hz	100 ms
	Intersection management	I2V	Periodic broadcast	1 Hz	500 ms
	Co-operative flexible line change	I2V	Periodic broadcast	1 Hz	500 ms
	Electronic toll collection	I2V	Periodic broadcast	1 Hz	500 ms
Infotainment	PoI notification	I2V	Periodic broadcast	1 Hz	500 ms
	Local Electronic commerce	I2V, V2I	Duplex, internet access	1 Hz	500 ms
	Media download	I2V	Duplex, internet access	1 Hz	500 ms
	Map download and update	I2V	Duplex, internet access	1 Hz	500 ms

Table 1.3. Non-safety-related use-cases (adapted from [18]).

As can be seen, one of the main differences between both groups of applications is the latency: while in non-safety applications and services the latency can be as big as 100 to 500 ms, in safety applications is 100ms. This is relevant for certain vehicular scenarios, particularly when modeling and testing.

Physical conditions, topology and performance

In vehicular networks, the main representation of nodes is hold by vehicles, although pedestrians and infrastructure are also normally considered as part of the network. As such, both vehicles and pedestrians move following a movement pattern which of often subject to certain conditions such as road topology, weather and day and time [52]. Nonetheless, the highly dynamic and mobile nature of vehicular networks makes them very unstable and unreliable, resulting in unknown and constantly changing network topology (although sometimes can be partially know, if the presence of relay nodes, such as those form infrastructure or roadside units, is considered). In fact, connections between nodes are very intermittent (unstable), and normally there is a lack of and end-to-end (E2E) path between two different nodes due mainly to the node’s high speed in a VANET, and to the presence of obstacles like buildings [29][36][52][72], which reduces the reliance of communication and adds high error rates and communication delays. As a summary, vehicular networks present the characteristics listed in Table 1.4, which make them unique and require them to have special treatment [29][36][50][52][68].

Physical aspects	Vehicles are the nodes in the network Nodes have high mobility The network has specific mobility patterns Nodes move with random and high velocities
Topology	Sparse and intermittent connectivity among nodes Highly variable network topology Lack of E2E connectivity among nodes
Performance	High latency Long and variable delays Asymmetric data rate High error rates

Table 1.4. Main characteristics of vehicular networks.

Perhaps the most unique characteristic of these networks is the fact that nodes must withstand the harsh environment and yet, try to communicate with other nodes. In this sense, the network must be able to resist delays and disruptions. Indeed, often vehicular networks are precisely called Vehicular Delay-Tolerant Networks (VDTN, for short). The main purpose of Delay-Tolerant Networks is to guarantee communication (i.e. reliable message propagation) in environments where otherwise it would be impossible, while maintaining an acceptable quality of service (e.g. low error rates and satisfactory delivery ratio) [1][7][29]. In vehicular networks it is the same case, and as such, they deserve a special treatment, particularly in the directives, rules and resources they use to communicate.

Main challenges in vehicular networks

Vehicular networks can be seen as a subgroup of Delay-Tolerant Networks (DTN), which are a kind of network with special characteristics whose main objective is to guarantee communication in very harsh conditions that prevent links between nodes to establish in a permanent way (they have to “tolerate” delays and disruptions). Due to the particular conditions of these environments, there are some challenges specific to this kind of network.

In [68], the authors expose the following research opportunities: network architecture (naming and addressing), node design (power, storage capacity, range, speed, physical link), node type (mobile, stationary), node interactions, node cooperation, network topology (known, partially known, unknown), mobility pattern (deterministic, stochastic, predictable, etc.), scheduling, traffic (static, dynamic), routing protocols, bundle format, caching mechanisms, security, and supported applications.

According to [47][68], major research issues in DTN include routing protocols, the investigation of DTN in various applications, and the performance analysis, as well as the queuing model, buffer management mechanism and the interaction among the traffic sources.

Other authors claim that mobility models (useful to evaluate routing protocols), routing and scheduling decisions are the areas that need to be paid attention to, although they also mention network architecture, aggregation and disaggregation algorithms (buffer management), routing protocols, scheduling and dropping policies, fragmentation mechanisms, network monitoring, and tools for the performance evaluation as well as applications for VDTN [36][72].

In [37], the authors hold that routing, network architecture, scheduling, forwarding issues and application designs, multicasting, delay and buffer management, congestion and flow control, cooperative schemes and mathematical modeling present research opportunities. A more essential challenging problem, according to the same authors, is the analytical modeling and performance evaluation of DTNs, since DTN characteristics vary from one environment to another, which difficult the development of a generalized DTN model. Also, none of the routing protocols proposed in the DTN open literature specifies a clear-cut procedure for setting up paths between communicating nodes, and the design of more intelligent, efficient and chattiness free network learning procedures is of particular interest, useful when there is little to no link information available (e.g. highly deterministic nodal contacts to absolutely unknown opportunistic encounters). Another problem arises when a bundle is received by its ultimate destination and its remaining replicas become useless, so it is important to know how to get rid of these additional unusable copies and free up resources. Finally, security issues are still at their early stages, since no security standards have been defined yet. Vehicular Networks, being a kind of Delay-Tolerant Networks, naturally inherit the same problems, with their particular conditions.

As can be seen, most authors agree on physical design, data management, services and application deployment, security-related and performance evaluation aspects as research fields in vehicular networks. Table 1.5 summarizes these challenges identified in the literature.

Physical design	Naming, addressing (network architecture)
	Node design (power, storage capacity, range, speed, physical link)
	Node type (mobile, stationary)
	Node interaction and cooperation
	Network topology (known, partially known, unknown)
	Mobility patterns and models (deterministic, stochastic, predictable, etc.)
Data management	Scheduling and queuing modeling
	Buffer management mechanisms (aggregation, disaggregation, etc.)
	Traffic (static, dynamic)
	Routing protocols (forwarding decisions)
	Bundle format
	Caching mechanisms
	Dropping policies
	Fragmentation mechanisms
	Congestion and flow control
	Network overhead control (useless msg copies elimination)
Services and applications	Supported applications (development, test and deploy)
	Non-safety applications (infotainment, ecommerce, etc.)
	Safety-related applications (crash assistance, collision avoidance, etc.)
Security-related	Privacy and confidentiality issues
	Accessibility
	Network and data integrity
Performance evaluation	Performance analysis (analytical modeling, tools, scenarios, data sets)

Table 1.5. Summary of research opportunities for VDTN.

Wireless Access for Vehicular Environments standards

The IEEE Society and the Standards Coordinating Committee of the IEEE Standards Association (IEEE-SA) Standards Board has developed a set of communication standards to provide support for Wireless Access for Vehicular Environment (WAVE) systems [33]. As the name suggests, the WAVE system as presently envisaged is designed to meet the communication needs of mobile elements in the transportation sector. While in many of the usage scenarios at least one of the devices engaged in WAVE communications is expected to be associated with a vehicle, other devices, both fixed and portable (e.g., roadside and pedestrian) are envisaged as well. As its mission, the WAVE standards enable the development of interoperable low-latency, low overhead WAVE devices that can provide communications in support of transportation safety, efficiency and sustainability, and that can enhance user comfort and convenience.

The WAVE set of standards are included as part of the IEEE 609 family of standards and the IEEE 802.11 – 2012 standard, through the amendment IEEE 802.11p. The full-use WAVE standards are depicted in Figure 1.6, and briefly explained below.

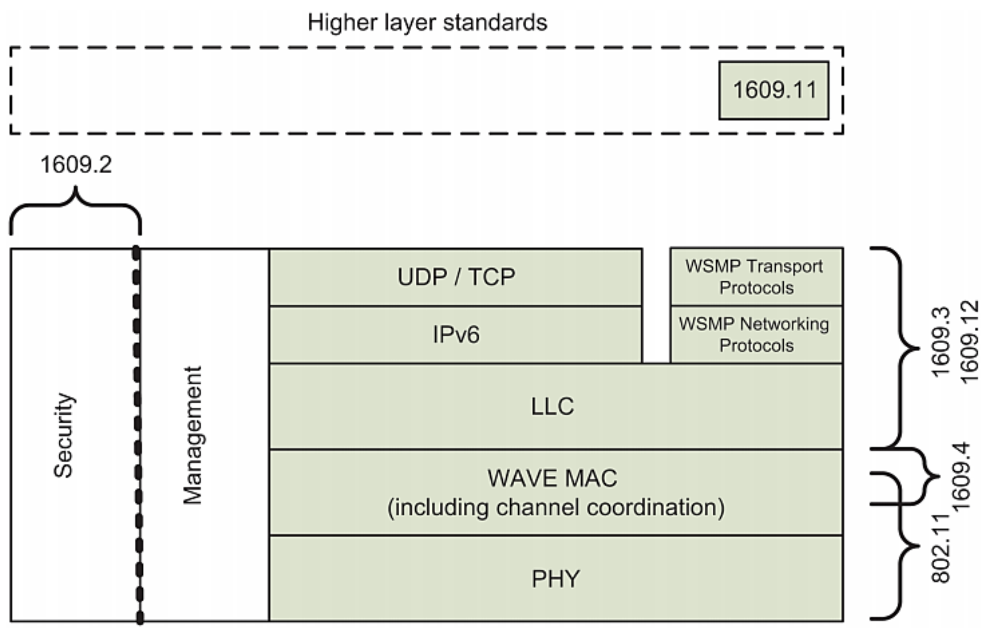


Figure 1.6. Full-use WAVE standards [33].

IEEE Std 1609.4 (Multi-Channel Operations) specifies extensions to the IEEE 802.11 MAC layer protocol and includes the following features:

— Channel coordination and routing, multi-channel synchronization
— Use of IEEE Std 802.11 facilities [e.g., channel access, Enhanced Distributed Channel Access (EDCA)] outside the context of a BSS
— Use of IEEE Std 802.11 Timing Advertisement frames in a WAVE system
— MAC-layer readdressing in support of pseudonymity

— Management information base (MIB) maintenance (contains configuration and status information)

IEEE Std 1609.3 (Networking Services) includes the following features:

- — WSA transmission and monitoring, channel access assignment
- — WSMP
- — Use of the local link control (LLC) sublayer and EtherType Protocol Discrimination (EPD)
- — Use of Internet Protocol version 6 (IPv6), including streamlined IPv6 configuration
- — Exchange of specific management information between WAVE devices
- — MIB maintenance (contains configuration and status information)

IEEE Std 1609.2 (Security Services for Applications and Management Messages) specifies communications security for WAVE Service Advertisements and WAVE Short Messages and additional security services that may be provided to higher layers.

IEEE Std 1609.11 (Over-the-Air Electronic Payment Data Exchange Protocol for ITS) is the first application level IEEE 1609 standard and specifies a payment protocol referencing ISO standards. An example use case illustrating electronic fee collection is provided in D.2.

IEEE Std 1609.12 (Identifiers) records the allocations of some identifiers used by the WAVE standards, including object identifier (OID), EtherType, and Management ID. PSID usage and encoding rules are also described.

As for the spectrum allocation, it is expected that WAVE systems are deployed to use the channel allocation depicted in Figure 1.7, or a subset of them.

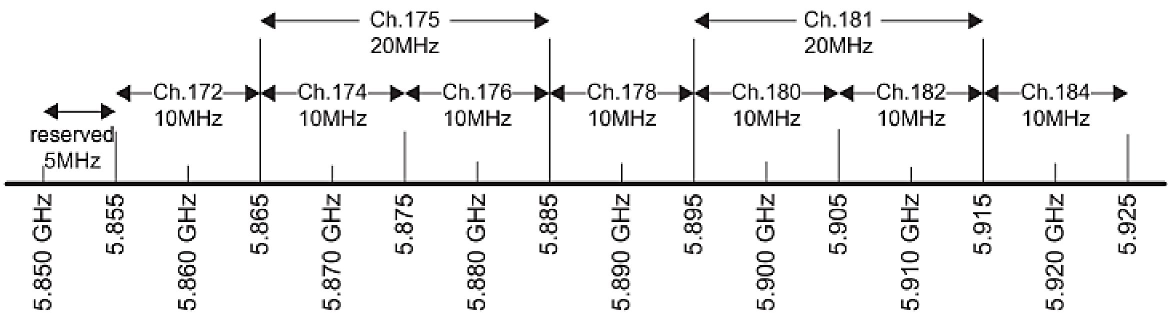


Figure 1.7. FCC Channel allocation.

The spectrum is allocated as follows:

- 5.850 GHz to 5.855 GHz is held in reserve.
- Channel 178 is the Control Channel
- Channels 172, 174, 176, 180, 182, and 184 are Service Channels.
- Channels 174 and 176 and channels 180 and 182 could be combined to produce two 20 MHz channels, channels 175 and 181, respectively.
- Channels 172 and 184 are designated for public safety applications involving safety of life and property. Specifically, channel 172 is dedicated "exclusively for vehicle-to-vehicle safety communications for accident avoidance and mitigation, and safety of life and property applications." Channel 184 is dedicated "exclusively for high-power, longer distance communications to be used for public safety applications involving safety of life and property, including road intersection collision mitigation."

5G technologies for V2X communications

The 5G technology for vehicles, as it stands out today, has yet to be standardize, deployed and adopted. There are some releases, however, that represent some efforts towards that technology. Currently, Cellular Vehicle-to-vehicle technology (C-V2X), also known as LTE-vehicle (LTE-V), is available as a fierce competitor of the IEEE 802.11p-based technology. C-V2X is a V2X radio access technology developed in its release 14 [19]. Perhaps the main advantage of C-V2X is the existing cellular infrastructure, but it lacks full support for low latency applications.

There is currently a fierce fight between the WAVE technology and its competitor 5G (under the name of C-V2X technology). The IEEE 802.11p standard was designed, from the beginning, to meet every V2X application requirement, including both safety-related and non-safety-related applications (Tables 1.2 and 1.3, respectively), with the most stringent performance specification [18], which was approved in 2009 and has had numerous field trials since then, and several semiconductor companies, such as Autotalks, NXP semiconductors, and Reneseas, have also designed and tested 802.11-compliant products [17][18]. According to the same authors, the IEEE 802.11 is ready to roll, and the market is expected to pick up significantly since 2016 in the US.

The cellular technology, on the other hand, being the most widely adopted standard for cellular communications, addresses only basic V2X use-cases and lacks support for low-latency and high mobility cases, closely associated with safety-related applications. The 5G technology, nonetheless, as it stands today, is well suited for non-safety related applications, associated with non-safety related use-cases, which involve infrastructure (V2I and I2V), where content originates or is processed on the cloud. It is unclear, however, how the network would perform in very congested scenarios (for instance, messages for traffic management are particularly relevant to highly congested and highly mobile urban scenarios). A comparison between WAVE and C-V2X (LTE rel. 14) technologies is summarized in Table 1.6.

Parameters	WAVE	C-V2X (LTE Rel. 14)	Future 5G
Currently available technology	Yes	Yes	No
Field trials (+10 years)	Yes	No	No
Applications	V2V, V2I	V2V, V2I, V2N	V2V, V2I, V2N
Latency	5 ms	20 ms	<5ms
Data rate	3-27 Mbps	150 Mbps	10 Gbps
Multimedia and cloud services support	No	Yes	Yes

Table 1.6. WAVE (802.11p) and 5G-related standards (adapted from [74]).

As can be seen, both existing technologies are currently available, but IEEE 802.11p could be the preferred standard for V2V communications, because it supports full coverage of all applications and has the work of 10+ years in field trials to backup this affirmation, whereas the 5G standard could be the preferred technology for non-safety applications, due to the current latencies. According to the timeline presented in [18] (fig. 1.8), the 5G technology could be ready to support V2X applications by 2023, but as shown in the table above C-V2X (LTE rel. 14) is ready, but does not achieve a performance in latencies less than 20ms, which for some safety applications is critical.

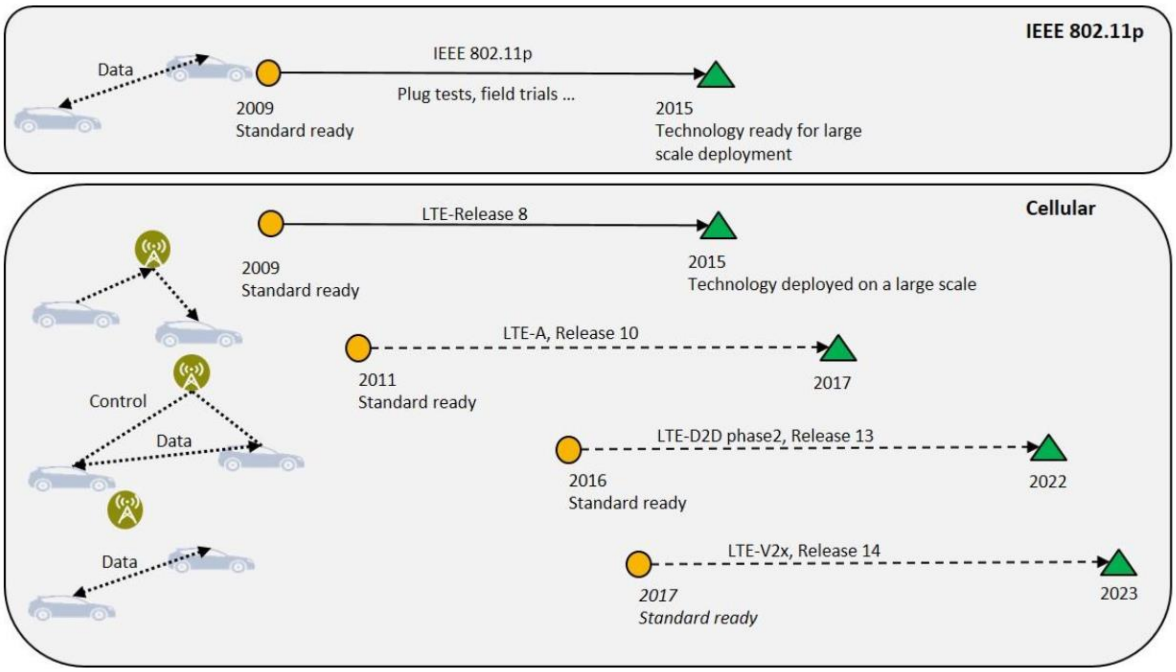


Figure 1.8. Timeline of IEEE 802.11p versus cellular technologies for supporting V2V (adapted from [18]).

Despite the common sense that 5G technologies can make use of existing cellular infrastructure, this is not as simple as it sounds, because today's infrastructure is not equipped to support the many V2X use cases that require very low latency in situations of high mobility and congestion. At the end of the day, nonetheless, as it stands out today, there is the need for broader compatibility in both technologies, and even though the IEEE 802.11p could be ready to roll, it's deployment and adoption will take time, as well as the 5G technology in order to be able to support most V2X applications. Also, the trend is to adopt both technologies and let the market decide which one is more suited for commercial deployment and evolution [12]. Furthermore, the path to fully automated vehicles will require coexistence for a period of time between vehicles with no active control systems, different levels of automated vehicles and fully automated vehicles, and therefore the coexistence in both V2X technologies is also a possibility, at least in the short term, when both of them are running towards its maturity [12][17].

Summary of Chapter 1

In this Chapter, the research methodology used in this work was presented, as well as the background of the research problem.

The Design Science in Information Systems Research Framework proposed by Hevner et al., in its design-science paradigm, seeks to extend the boundaries of human and organizational capabilities by additions to the knowledge base and applications of existing theories and artifacts in the appropriate environments. As part of Intelligent Transportation Systems, in the context of Smart Cities, vehicular networks, also known as Vehicular Delay-Tolerant networks, are a key component in these ecosystems, and for their physical implementation, current efforts towards achieving V2X communication include the IEEE 802.11p standards, which are preferred for safety-related applications and the C-V2X technology, based on cellular communications. The former has advantages like it is ready to be implemented and has more than 10 years in field trials, although the main downside is that major changes in the infrastructure are needed, whereas the later has its strengths precisely in the infrastructure, although it is not preferred for safety-critical applications, such as crash warnings, but for non-critical use-cases including traffic-related services and infotainment. As time goes by, the market will determine which technologies ultimately will be used for V2X applications, or if a combination of both is possible.

Research opportunities identified in the literature include node design (power, node capabilities, etc.), data management (queuing and buffer management, scheduling and routing), security and supported applications. Of them, one of the main challenges to achieve seamless and reliable communication is routing protocols. In this research, a new packet routing protocol for vehicular networks is presented, based on a deep learning architecture, to explore the impact of this machine learning technique in the routing process. Furthermore, how the research framework and the research process fit together has been briefly explained. In the following chapters, this alignment will be discussed in more detail.

Chapter 2

Research Problem

In this section, the research problem is discussed in more detail.

Problem statement

As discussed in previous paragraphs, routing is one of the research challenges identified in the field of VDTN. As a key component in the functioning of this kind of networks, the lack of efficiency in routing brings delays in the unleash of their full potential. In this thesis, this problematic is addressed, and it is presented as:

The routing algorithms in vehicular networks are not fully efficient

The research problem has intrinsic theoretical relevance, and because

i. Theoretically, communication in vehicular networks is possible but is not enough to offer superior communication services. This is due to the natural existence of big data loses, very long delays and thus very small packet delivery ratios.
ii. Routing is the core of every communications network. The protocols used to achieve data transmission highly determine the efficiency of the whole communication process.

The practical relevance of the research problem is met by the following affirmations:

i. The relevance of having a highly efficient algorithm to route in vehicular networks is intrinsic since it helps to achieve the best performance in communication process in Intelligent Transportation Systems.
ii. Having more efficient communications in ITS, particularly in vehicular networks, will make possible to have more and better services in different areas, such as security services, information services, entertainment services, transactional services (e-commerce, automated payments, etc.) and any other service derived from V2X communication (vehicle to anything), including V2P (vehicle to pedestrian).

iii. The successful implementation of routing algorithms in VDTN can serve as a model to improve similar networks; for instance, other kind of DTN network: sensor networks, underwater networks, etc.

Research question

The main idea behind the purpose of VDTN is that they must be able to provide seamless communication in very harsh conditions; given the intermittent connections and frequent disruptions in the network, their goal is to guarantee communication (e.g., delivery of messages) with the less possible amount of delays. When assessing the performance of routing protocols, the most popular metrics are precisely delivery rates, delays and network overhead and number of hops [72]. These metrics are summarized in Table 2.1 and are more broadly explained in the next chapter.

METRIC	DEFINITION
Packet Delivery Ratio (PDR)	Ratio of delivered messages to created messages.
Average Delivery Delay (ADD)	Elapsed time since a message is created until it reaches its destination.
Networks Overhead (OVH)	Ratio of useless copies in the network, with respect to the amount of delivered messages.
Hop Count (HOP)	Number of nodes that a message traversed to get to its destination.

Table 2.1. Summary of metrics for performance evaluation in VDTN.

Even though the routers in the network must be able to guarantee communication, the balance in performance of these parameters is also important, because they all reflect important aspects of the network. Also, state of the art optimization techniques includes Artificial Intelligence, and more particularly, Deep Learning is gaining quite a lot of attention with recent advances in the mater. Hence, the research question is formulated as follows:

To what extent can a routing algorithm based on Deep Learning influence the performance of routing in VDTN?

The goal in this research is then to find out to what extent the use of Deep Learning influences the performance of routing in VDTN, reflected by changes in the metrics when evaluated. For this, a router based on this AI technique was implemented, including its architecture and its routing algorithm, and its performance was compared with other known routing algorithms.

Proposed Solution

Nowadays and for the past few years, Artificial Intelligence has become an important field as it the advances in the matter have resulted in the solution of a wide variety of very complex problems, using this approach of leveraging the information in the environment to actually learn from it and make smart, accurate predictions. In this work, a Deep Learning – based Router that is capable of learning to make intelligent decisions based on local and global conditions from a dual perspective, called DLR+, is proposed. The goal with this solution is to provide an algorithm that results in improvements in at least one of the four more used metrics (namely, delivery ratio, delivery delay, network overhead and hop count), to be able to respond to the research question. As envisioned, the proposed solution has two main advantages listed below:

i. The algorithm can be further refined harnessing the power of Deep Learning
ii. The algorithm will make effective use of available resources in the network
iii. The algorithm reduces the risk of security leaks, as sensitive and private information is processed locally

The broadness of the routing problem and its environment of application make it difficult to address explicitly all their aspects. To be more specific, it is impossible to consider absolutely all the performance metrics due to the complexity of the so-resulting model. This research work is limited to the following metrics:

i. Network Overhead
ii. Hop Count
iii. Average Delivery Delay
iv. Packet Delivery Ratio

Finally, the proposed solution is tested in a synthetic scenario, as is typically done by researchers in this field due to the high complexity of real-life implementations.

Related work

There exist several routing algorithms for VDTN [7][11][34][72]. Some examples of routing algorithms in DTN include the Epidemic Protocol (probably the most known routing protocol) and the Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET protocol) [72], as well as different classifications and taxonomies proposed by different authors based on different parameters such as the objective of the protocol, the amount of information required by each protocol, the availability of information regarding the actual state of the topology and its future evolution (deterministic or stochastic routing)[7], and the destination of a single packet (unicast, multicast and anycast routing) [11]. Some other protocols referring to opportunistic networks encountered in the literature include the Heterogeneous Context-aware Routing protocol (HCR) [77], BUBBLE Rap [2], Predict and Spread [49], the Epidemic Routing Protocol, the MV Routing Protocol, the Network-coding-based Routing, the Context-aware Routing (CAR) and many others [54]. More specific to VDTN, there are also some proposed routing algorithms such as CONHIS [58], GeoDTN+Nav and Fast-Ferry Routing in DTN-enabled Vehicular ad hoc networks (mentioned in [9]), GVGrid, GSR, GyTAR (Greedy Traffic Aware Routing), TAPR (Traffic Adaptive Packet Relaying) and some others surveyed in [1],as well as proposed classifications and taxonomies for VDTN routing algorithms [1][7][51]. DTN protocols, however, need to be properly adapted to vehicular constrains in order to get better outcomes. For instance, unlike other types of DTN where the mobility patterns are purely random-based models, the mobility patterns of vehicles on highways are often times predictable due to the restrictions imposed by roads, traffic, intersections, etc. [1][55]. Another unique characteristic of vehicular networks is the existence of roadside infrastructure which can be leveraged in order to improve the efficiency of routing mechanisms [1]. Other than the network characteristics, the applications (e.g. safety related applications, traffic monitoring, etc.) which are expected to run on top of vehicular networks also make it a unique environment [1]. Also, the node's high mobility (the network topology changes rapidly because of vehicle speeds), the inconstant topology in time and space (the network topology evolves depending on time (e.g., traffic jams) and location (urban, rural)), the large scale possibilities (all vehicles are potential nodes), and the no significant power constrains (cars can generate sufficient power) are specific conditions of vehicular networks that make them special [51][55]. As a result, VDTN do present some key differences from traditional DTN, so they justify special protocols to leverage those particular characteristics and improve the network performance.

Many dissemination algorithms are based on the popular epidemic dissemination principle (also called Epidemic Routing or Random Spray, depicted in Figure 2.), in which a node replicates the message to all contacted nodes that do not have the packets yet. Compared to other solutions, this one wastes much storage and bandwidth, though it is one of the simplest to implement and could guarantee the maximum delivery ratio regardless of the buffer space exhaustion [11][26][37][47][51][58][72].

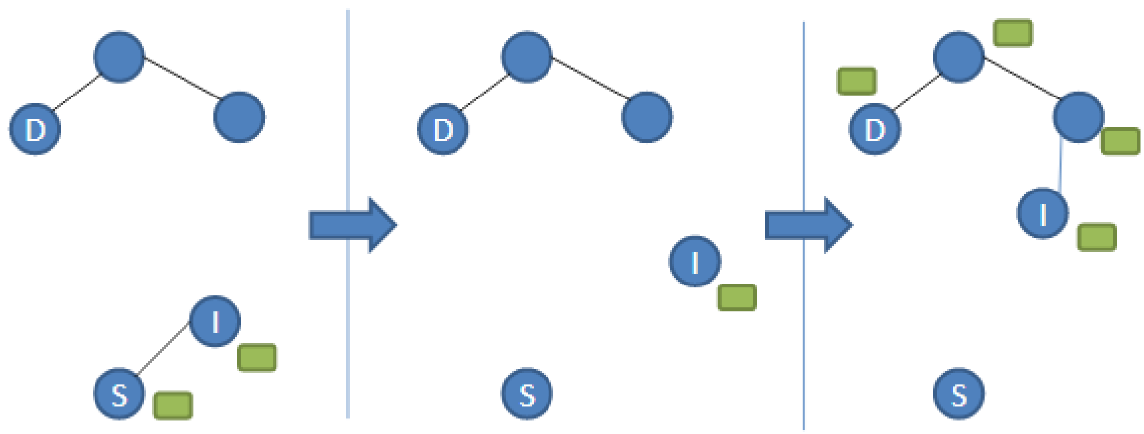


Figure 2.1. Flooding principle in epidemic-based routing protocols.

Among routing algorithms derived from Epidemic Routing are Spray-and-Wait [11][47][51], Two-Hope-Relay [11], Network Coding-based Epidemic Routing (NCER) [37], NECTAR [11], Distance-Aware Epidemic Routing (DAER) [11], Unicast Based Routing (UBR) [11], Message Suppression RSUs (MSRs) [34], DAWN (Density Adaptive routing With Node awareness) [26], and DARCC [26]. Besides the epidemic dissemination principle, these epidemic-based algorithms use additional information to get a better performance than the pure Epidemic Routing algorithm, such as topology information, history of connections in each node, distance between nodes and number of packets to disseminate.

Other kind of algorithms consider different aspects for the dissemination process. For example, geographical routing is another approach for efficient routing, which takes the location information of the vehicle into consideration. With this idea in mind, GeOpps (Geographical Opportunistic Routing for Vehicular Networks) aims to enhance the performance of single-copy routing protocol in VDTNs as it exploits the geo-location of vehicles to forward the geographical bundle opportunistically towards the final destination location [26][72]. Greedy DNT and MoVe are also location-based algorithms such as GeOpps [72]. Others take into account some kind of history of encounters or connections in order to estimate the probability of a node to be able to deliver the message to the final destination, such as PRoPHET [26][72], ZebraNet [58] and CAR [58]. Finally, some other algorithms include social metrics, such as the number of links in the social graph or their centrality to choose the next forwarding node, such as in ZOOM and SinBet [72].

One last branch is the use of Machine Learning in routing algorithms. Machine Learning is a branch of Artificial Intelligence that aims at processing huge amounts of data to find and apply patterns [25]. In [62] the authors developed a routing algorithm based on Fuzzy Logic that reactively searches for the best route in a fixed graph, based on distance, overhead, power consumption and remaining active time of the route. In [61], an algorithm based on the Simulated Annealing technique was developed. In that work, the authors focus in finding the most suitable message to be transferred next in each connection. This algorithm will be explained in more detail at the end of this section. The authors in [66] propose KNNR, a router based on the KNN optimization algorithm; they focus on finding the best next hop in the route of a message framed as a binary classification algorithm. It is worth noticing that, to the best of our knowledge, none of the proposed algorithms use Deep Learning techniques in a realistic VDTN scenario. There are some efforts oriented in that direction, though. In [63], the authors present a routing strategy based on a neural network. The objective is to find the next node and the next instant of contact, but they consider the "predefined" path of buses and history of contacts, which is a very particular scenario of a vehicular network, with low density and predefined path. Also, they use a neural network only 1 hidden layer, which is not considered deep learning. In [41], the authors work on a routing strategy using neural networks as well, but they focus on security aspects and not on other performance metrics, such as delivery ratio and delay. Finally, the authors in [43] use a Radial Basis Function neural network (whose functioning is similar to that of the KNN technique). They try to fragment the network applying clustering in a straight highway. The clustering is done using the Simulated Annealing technique, and a Radial Basis Function Neural Network to determine the Cluster Head using velocity and free buffer size as the parameters for the decision. To the best of our knowledge, none of the existing routing algorithms in VDTN use DL.

Table 2.2 summarizes the most popular protocols found in the literature.

NAME	TYPE	DESCRIPTION
Epidemic	Flooding	**[75]** Replicates each packet to nodes in range, which in turn replicate further to each new connection.
MaxProp	Controlled flooding	**[79]** Uses frequency of contacts (hop count) as metric to schedule the incoming packages. Then, the algorithm uses epidemic transmissions until the TTL of the messages expire.
Spray and Wait	Controlled flooding	**[69]** Controlled version of Epidemic: for a given packet in the queue, "sprays" a given number of copies the first time, and each subsequent node only delivers the msg to its final destination.
PRoPHET	Probabilistic	**[41]** Uses history of previous encounters to compute the delivery predictability of each node and select the next hop.
GeOpps	Location	**[47]** Uses the location of the nodes to find the best route based on the shortest path.
Fuzzy based Routing Protocol	Fuzzy Logic	**[31]** Reactively searches the best route in a fixed graph based on distance, overhead, power consumption and remaining active time of the route.
SeeR	ML	**[61]** Based on the Simulated Annealing ML algorithm. They focus on the message with higher probability to be delivered.
NN Routing strategy for buses	ML	**[63]** The objective is to find the next node and the next instant of contact (they consider the "predefined" path of buses and history of contacts). They use only 1 hidden layer.
Trust-based Routing with NN	ML	**[80]** Focusses on security aspects in terms of "trust": their goal is to find the most "trustable" next node
KNNR	ML	**[42]** Use KNN for binary classification to find the next hop
SA-RBFNN RP	ML	**[4]** Applies clustering in a straight highway. Clustering: SA; Cluster Head: RBFNN using velocity and free buffer size.

Table 2.2. Summary of most popular routing protocols for VDTN. ML stands for Machine Learning.

In practice, there is not a clear-cut way to know the performance of the existing algorithms. The fact that authors use different scenarios in their experiments, with different simulators and different metrics, and that they do not provide useful information for reproducibility make infeasible the reproduction of the algorithms. To be best of my knowledge, there is no complete study of performance of all existing algorithms that includes evaluation of the main metrics among all of them even for a single particular scenario. For instance, in existing surveys and comparative analysis [83-87], the authors focus on different aspects such as type of router, simulation scenario, simulator, vehicle density estimation, if they use store-carry-forward, if the routers handle network disconnections, if the routers are traffic-aware, if they require maps or not, if they use realistic scenarios or not and the complexity of computation, and they even propose different taxonomies and classifications but do not include how the routers' performance is with respect to their counterparts in delivery ratio, delivery delay, network overhead or any other metric, and I argue that the main reason for that is the lack of reproducibility. Indeed, one of the open issues is the need of a standard tool and procedure for evaluation of these protocols [84][85]. Those are the main reasons why in this research, the routing protocols used for comparative purposes were the following routers, which were already defined in the simulation environment (The ONE simulator): the Epidemic routing (probably the most widely known routing protocol for delay-tolerant networks), the Spray and Wait algorithm (a controlled version of the Epidemic protocol), the PRoPHET (the most popular probabilistic routing protocol), and the SeeR routing protocol (and AI-based algorithm for VDTN whose authors give the complete code for reproducibility). Next, such algorithms used in the experiment are explained in a broader way.

Epidemic Routing.

This routing protocol was proposed in 2000 by Vahdat, A. and Becker, D. [75] With this protocol, they aim at maximizing the message delivery rate (ADR) and minimizing the message latency (ADD), as well as minimizing the total resources consumed in message delivery. In the approached proposed in this router, they distribute packages to other hosts, called *carriers*, who in turn carry the messages and further replicate them when they are in contact with other nodes (see fig. 2.1). The hope is that with this flooding mechanism, the message will eventually reach its final destination.

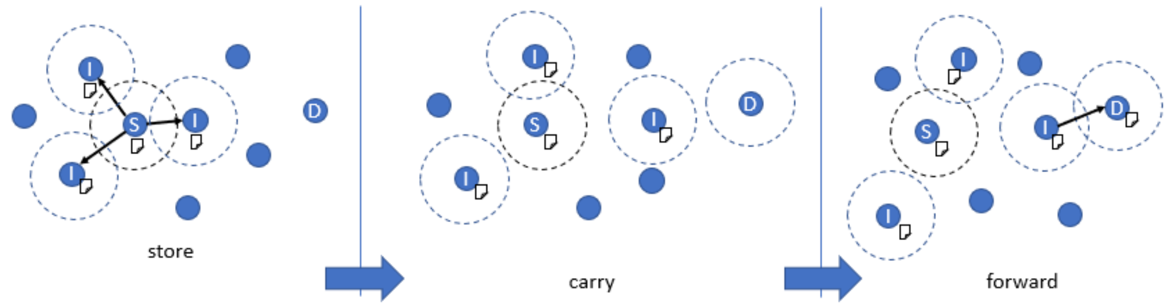


Figure 2.1. Flooding principle used in the Epidemic Routing.

While there is a high probability that the packets will eventually make it to their destination, providing that the routers have enough buffer space, that is not always the case. In fact, not only the buffer resources may be limited, but also the bandwidth resources will be compromised, since with this uncontrolled spreading a lot of useless copies will be left in the process.

Spray and Wait.
This routing protocol is a controlled version of the Epidemic Routing. It was proposed in [69] by Spyropolus, T., et al. in 2005. The main purpose behind this algorithm is to have a routing protocol that reduced the network overhead, optimizing resource usage. The proposed algorithm consists in two stages: the *Spray* and the *Wait* (fig. 2.2).

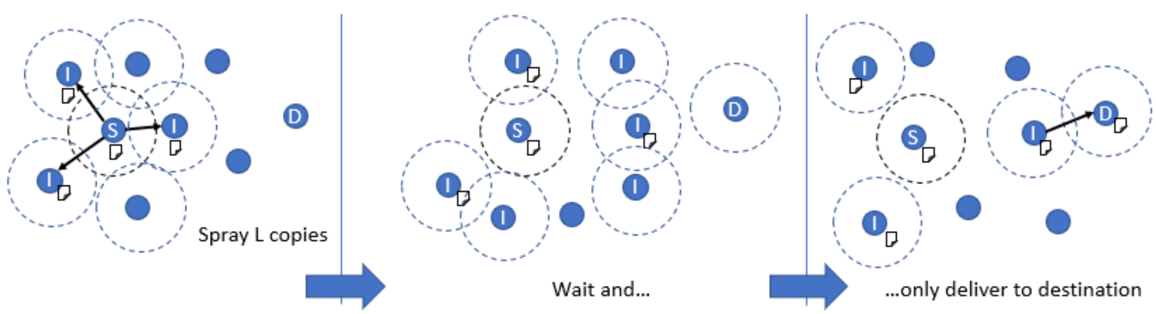


Figure 2.2. Graphical description of the Spray and Wait routing protocol.

In the Spray Phase, a router that creates a message spreads L copies of that message to some other hosts. During the Wait Phase, if the destination node is not found in the Spraying phase, each of the L nodes will carry the message and transmit it only to its final destination. This can be seen as a trade-off between a single copy and a multi-copy scheme, which ends up in a reduction of the number of copies in the network, thus reducing the network overhead. The performance in this algorithm shows high delivery rates and acceptable delivery delays, with respect to the Epidemic Routing, and better performance in delivery rate for scenarios with high node density due to severe contention.

PRoPHET protocol.

This protocol, Probabilistic Routing Protocol using History of Encounters and Transitivity, was proposed by Lindgren, A. Doria, A. and Schelén, O. in 2004 [41] and is the most popular protocol that bases its routing decisions in a probabilistic mechanism. The goal in this algorithm is to improve the delivery rate while maintaining the network overhead and buffer space usage at low level. They define a metric called delivery predictability $P(a, b) \in [0,1]$ at every node a for each known destination b, indicating how likely is it for a to deliver a packet to b. When there is a connection between two nodes, they exchange information to update their delivery predictability tables (see fig. 2.3), and a transmits to b only if $P(a, X) < P(b, X)$.

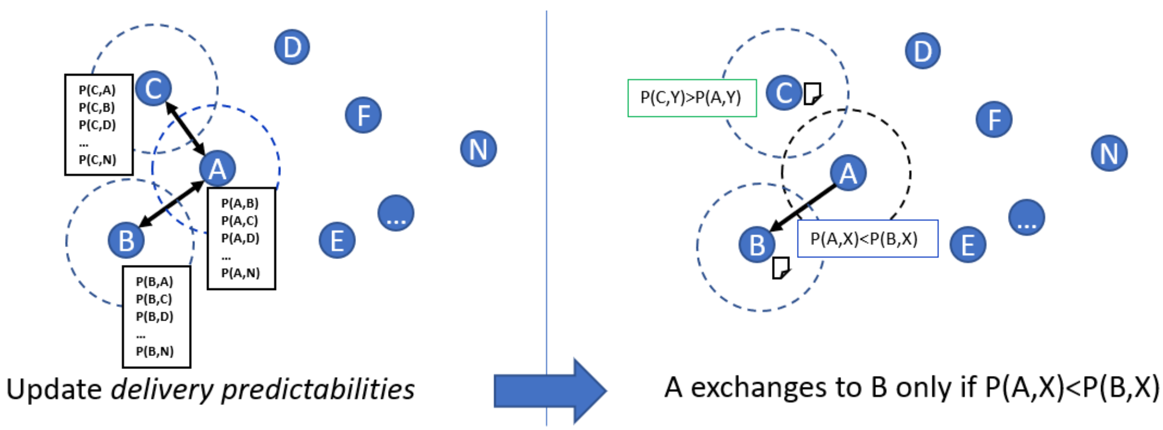


Figure 2.3. Graphic description of the PRoPHET protocol.

The update of their delivery predictabilities is done according to Equations 2.1 and 2.2, where $P_{init} \in [0,1]$ is an initialization constant, and $\gamma \in (0,1)$ is the *aging* constant used to *age* the probabilities of two nodes if they do not meet in k time units.

$$P(a, b) = P(a, b)_{old} + P_{init} \cdot (1 - P(a, b)_{old}) \tag{2.1}$$

$$P(a, b) = \gamma^k P(a, b)_{old} \tag{2.2}$$

Furthermore, they consider that all nodes have a transitive property based on the observation that, if node A frequently encounters node B, and node B frequently encounters node C, then node A frequently encounters node C. This is shown in

Equation 2.3, where $\beta \in [0,1]$ is a scaling constant that decides how large is the impact of this transitivity.

$$P(a,c) = P(a,c)_{old} + (1 - P(a,c)_{old}) \cdot P(a,b) \cdot P(b,c) \cdot \beta \tag{2.3}$$

The authors tested some synthetic scenarios with PRoPHET and Epidemic, and they found that, under some conditions, the proposed protocol outperformed the Epidemic Protocol, though that was not always the case.

SeeR routing

This router was proposed by Saha, B. K., Misra, S., and Pal, S., in 2017. This algorithm is based on the Simulated Annealing, and they focus on trying to determine the messages best suited for a transmission at that moment. Each message has a cost function associated to it, that depend on three locally observed variables: estimated inter-contact time (ICT) of a node, time-to-live (TTL) of a message and current hop count of a message. Inter-contact time of a node $\hat{\tau}_i$ at time t is computed as in Equation 2.4, where T_{ij} is the instant when the previous contact of i with j terminated and $\alpha \in (0,1)$ determines how much weight should be given to the historical estimate of ICT.

$$\hat{\tau}_i = \alpha\hat{\tau}_i + (1 - \alpha)(t - T_{ij}) \tag{2.4}$$

The residual TTL $\rho(m,t)$ of a message $m \in M$ at an instant t is determined by Equation 2.5:

$$\rho(m,t) = m.ttl - (t - m.createdAt) \tag{2.5}$$

where $m.createdAt$ is the time instant where m was created. Finally, the cost function of having a message m at node i, $C(m,i)$ is given by Equation 2.6, where $h(m,i)$ is the number of nodes that message m traversed at that point, including node i.

$$C(m,i) = \hat{\tau}_i(1 + h(m,i)) \tag{2.6}$$

Similarly, if there is a transmission from node i to node j, the message m traversed one more node, $h(m,j) = h(m,i) + 1$. Consequently, the cost function at j changes according to Equation 2.7:

$$C(m,j) = \hat{\tau}_j(2 + h(m,i)) \tag{2.7}$$

Based on this function, the simulated annealing process consists of a *cooling* stage, where the initial temperature θ_0 of a message $m \in M$ changes over time. More particularly, at the k^{th} replication attempt of m, its current temperature $\theta(m)$ is decreased according to Equation 2.8, where γ is usually taken between 0.85 and .99.

$$\theta(m) = \gamma^k \theta_0 \tag{2.8}$$

The replication of m is attempted at every contact until its current temperature exceeds a lower threshold ϵ. With these considerations, the algorithm is rather simple: the objective herein is to have a subset of the messages at any node that can be replicated to the other nodes when a contact is established. From them, the goal is to evaluate the *fitness* of every message to have the "best" message to replicate. Furthermore, the replication of message m takes place from node i to node j only if its temperature θ (m) is greater than the threshold ϵ and having the message at j is less "expensive" (i.e., if the cost function is smaller) than having the message at i. Figure 2.4 depicts the SeeR routing algorithm.

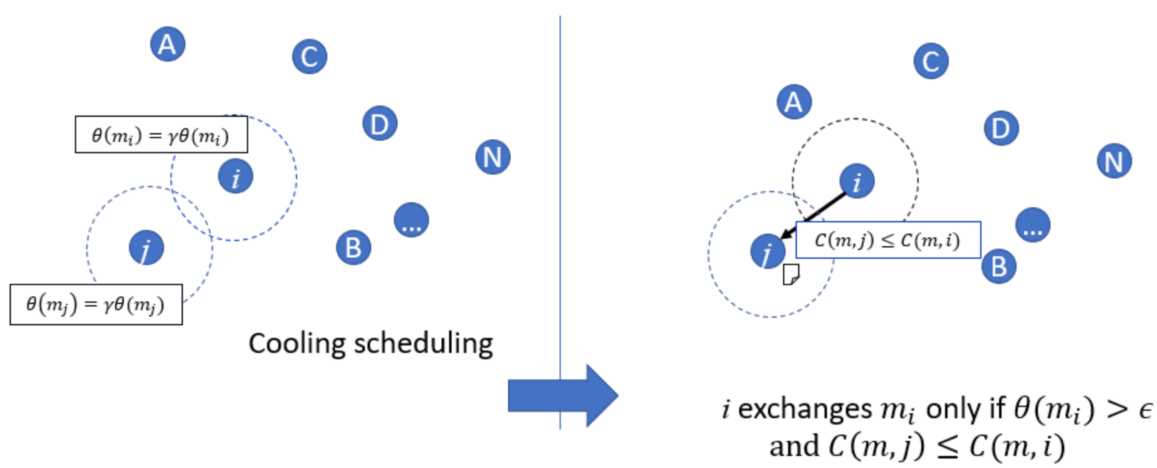


Figure 2.4. Graphic representation of the SeeR algorithm.

Metrics for performance evaluation

Finally, different protocols take into consideration different metrics in order to improve their efficiency, but none of them take into account absolutely all of the parameters that play a role in the performance, as some assumptions have to be made in order to simplify the algorithm and make it cost-effective. Many protocols are based on the criteria of how to select the forwarding node, but this is not the only aspect that can make a difference in the performance of different protocols. Indeed, very rarely do researchers evaluate the same metrics under the same scenarios, when comparing performance of different protocols [72]. Undoubtedly all VDTNs have in common some conditions such as opportunistic encounters, changing (though somehow predictable) topology, node's high speeds and particular mobility models within the network (i.e. how the traffic flows within specific road topologies). As a result, a set of mechanisms that define the hop-by-hop and the end-to-end communication schemes can heavily influence the delivery ratio, the delivery delay or other performance metrics. Generally, these mechanisms can be applied to any utility-based protocol. Some of the most representative mechanisms available in the bibliography are reliability, redundancy, path diversity and message priority [72].

One critical factor when proposing a new protocol is its evaluation, and since developing and conducting real implementation and tests for vehicular networks is very expensive in terms of time, human resources and money, simulation is the alternative used by researchers, although there is a lack of balance among different simulations that complicate the comparison of different results, being the following metrics the most used: delivery ratio, average delay, delay cumulative distribution function, overhead and average number of hops priority [72]. These metrics are described in more detail below.

Packet Delivery Ratio

We will call this metric PDR, for short. This value is defined as in Equation 2.9 and is a value that is desired to be maximized, which would mean that a great amount of the messages that were created were successfully delivered to its destination.

$$PDR = \frac{\#\ of\ delivered\ msgs}{\#\ of\ created\ msgs} \tag{2.9}$$

Ideally, we would like this number to be 1, but in practice this seems rather impossible, since there are other constrains in the network, such as buffer size and message TTL, resulting in dropping or destruction policies which prevent some of the messages to get to its destination. Because the resources in the network are limited, that is precisely why they must be optimized. This parameter shows the fraction of created messages that got to its destination.

Average Delivery Delay

Also known as latency, this parameter is the elapsed time since a message is created until it reaches its destination. In other words, this number shows how long it takes for a message to be delivered. Ideally, we would like this value to be 0, but this is obviously impossible. Instead, the minimization of this parameter is pursued. We will call this parameter ADD, for short.

Network Overhead Ratio

This parameter (that we will call OVH, for short), shows the ratio of the messages that were relayed to the network that did not reach their destination with respect to the number of messages that did do it. Equation 2.10 shows this definition:

$$OVH = \frac{\#relayed\ msgs - \#delivered\ msgs}{\#delivered\ msgs} \tag{2.10}$$

The impact of OVH in the network is directly in the resource usage on the entire network. Ideally, this value should be minimized to reduce the problems related to poor bandwidth allocation, such as network congestions and consequential delays and disruptions.

Hop Count

HOP, for short, this parameter shows the . The smaller this parameter is the less administrative overhead in the previous hosts this message may have caused, so it is ideal to keep this value low.

All of the above described metrics are desired to be optimized, since all of them offer some advantages in the overall performance of the network, which can be critical under particular environments. For instance, a low OVH would be desired in networks with hosts with low buffer capacity, such as sensor networks.

Summary of Chapter 2

In this chapter, the research problem has been stated, introducing its theoretical and practical relevance, and the proposed solution has been broadly explained, including the advantages and limitations in this research. In addition, existing work towards a solution to the research problem has been presented, ranging from the simplest flooding routing protocols, like the Epidemic Protocol, to controlled flooding protocols (such as Spray and Wait) to the ones that use probabilistic approaches (like PRoPHET), to the ones that consider physical aspects such as geographic location, to others whose goal is to provide the router to be more capable of taking the routing decisions, and use heuristic approaches, such as SeeR and other machine learning -based algorithms. Nonetheless, although some protocols use Machine Learning and other Artificial Intelligence techniques, none of the existing routing protocols use Deep Learning as its core.

Also, different preliminary considerations on how to improve the efficiency of routing protocols are discussed, such as next hop selection and message scheduling. We found that routing is one of the primary concerns in VDTN, since a poor routing technique results in long delays and low delivery rates, let alone the network resource consumption. Finally, the use of simulations as testbeds is also informed, stressing the lack of a unified simulator and the fact that the proposed algorithms are not implemented as to be part of the public domain so more researchers can make use of them, compare against them and even improve them. Most researches do not provide technical details on the implementations, which difficult the replicability of their proposals and experiments.

In practice, there are many aspects that need to take into consideration in a network design and as for the metrics used to evaluate the performance of routing algorithms, the most widely used are packet delivery ratio, average delivery delay and network overhead.

In the following chapter, the research problem is formalized, and the proposed routing architecture and protocol are introduced and explained in detail.

Chapter 3

Solution

In this section, the design of the proposed solution is explained, including the formalization of the research problem as well as the proposed routing architecture and protocol. Also, the experiments carried out to evaluate the performance of the proposed solution are explained.

Problem framing

The VDTN scenario consists, in general, of nodes (vehicles) which move in random directions according to some mobility rules, and in their way, they can produce, carry and deliver data packets to other nodes in the network, making communication possible. In this section the routing problem is introduced and put in a more formal way.

Formalization

Let $N = \{N_i | 1 \leq i \leq L_N\}$ be the set of available nodes in a vehicular network with constant disruptions and non-fixed topology, and let $A \in N$ be a given node in that set (Figure 3.). Given the fact that there are no predefined paths and the connections are intermittent, the nodes in the network must act opportunistically, taking advantage of any node that gets into their communication range, because whenever these encounters happen, the opportunity of replicating a message arises. In those situations, A has to decide on a node to start a transfer, and several criteria can be used for this decision, but ultimately, A would like to choose the node with better capabilities of further spreading the messages until hopefully they get to their destination. Following this approach, the routing problem can then be expressed as finding the *best next hop* (BNH) for the messages. This is, from all k nodes that A is connected to in a given moment, the one, N_x, with better fitness f_x must be determined, in terms of its current features $x_1, \dots x_n$. Furthermore, in order to optimize the communication conditions, not only the best next hop B must be selected, but we can also detect the *best next message* (BNM) to be transferred. This is, based on its current attributes $y_1, y_2, \dots, y_m$, we must be able to select from the message queue $M = \{M_i | 1 \leq i \leq L_M\}$ the message $M_y \in M$with the best fitness f_y. Because neural networks have the power to learn very complex non-linear patterns, they are the perfect fit for what we are traying to achieve here, so we can model both optimization scenarios as binary classification tasks to allow us to precisely quantify the capabilities of such nodes N_i as a function F of some of their characteristics x_i as $f_x = F(x_1, x_2, \dots x_n)$ and the capabilities of such messages M_i as a function G of some of their characteristics y_i as $f_y = G(y_1, y_2, \dots y_n)$.

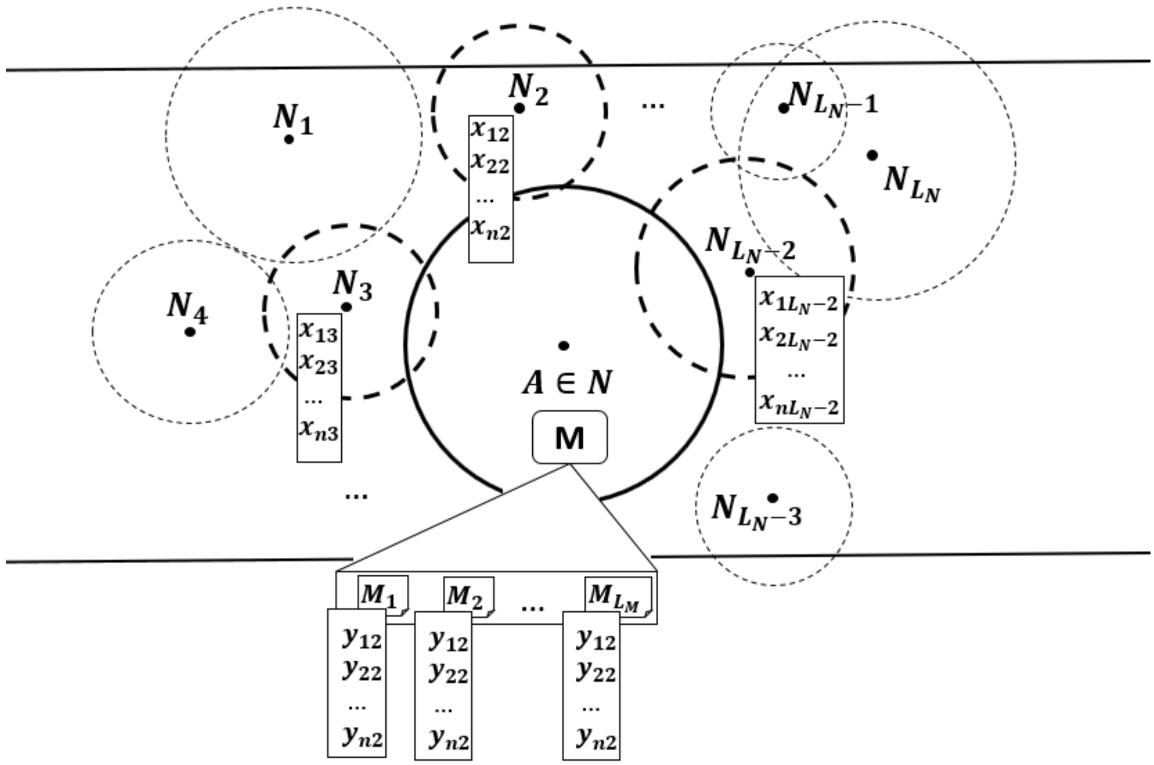


Figure 3.1. Opportunistic encounters for routing in VDTN and the message queue in a host.

DLR+ Router Overview

In this section we describe in more detail the fundamental principle and architecture of DLR+, the router in the proposed solution. The main idea is to have a router capable of learning from the conditions of its environment and use such information to make smart forwarding decisions. Those conditions in the environment are given by the current features of the messages in the msg queues of the nodes, and the current attributes of the nodes themselves. In order to achieve that, the router uses two pre-trained feed forward neural networks to process the information from both its neighbors and the messages in their queues in real time and selects, when a connection is formed (i.e., when a transmission is possible), from them the best next hop for the best next message, according to their current fitness. More details are given in the following subsections.

Router Architecture

The core of the router has three fundamental modules that allow the router, upon a connection-up event, to choose the best next hop from its current connections and the best next message to send from its queue, but also to share information to other nodes (upon request), so they can decide whether or not to pass a packet to it. Such modules are called, respectively, the Connections Manager and the Fitness Center, which in turn has two independent modules for the messages and for the host itself (Figure 3.).

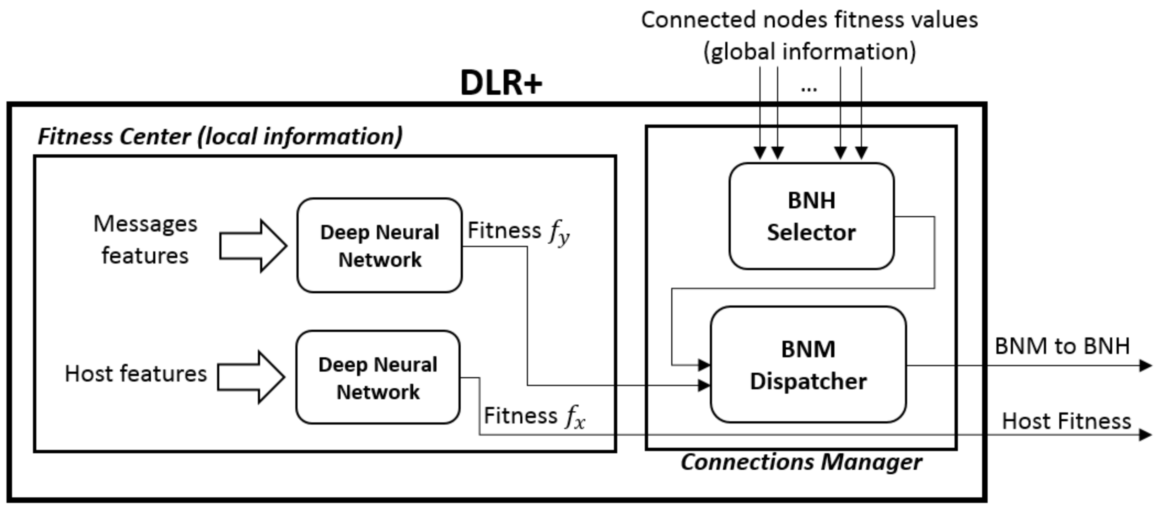


Figure 3.2. The fundamental routing architecture of DLR+.

The Fitness Center

This part of the router has two pre-trained deep feed forward neural networks that use the available local information to compute the router's current fitness f_x, defined as the value that determines its ability to correctly deliver data packets to the final destination, and the fitness f_y for each message in the queue, with $f_x, f_y \in R, 0 \leq f_x, f_y \leq 1$. The closer these values are to 1, the fitter their owners are. More details on how to get these numbers are given in section 4.2 – The Neural Network. These values are automatically updated in each router right after a connection is ended and right after a new message has been received, so they are available and ready to be used at any moment.

The Connections Manager

The function that this module has is vital in the selection of the best next message for the best next hop. This module manages the incoming connections, requesting their f_x values in order to select the fittest node. After this, if available, the message scheduler will send the fittest msg to such node.

The Neural Networks

The problems of finding the BNH and BNM is treated as binary classification problems, given that the main goals are to know if the node and messages are in best conditions (i.e., *fit*) to carry and deliver the messages, or not. Thus, the neural networks used in the Fitness Center are Feed Forward Neural Networks with several hidden layers each and one output. The general architecture of these neural networks is presented in Figure 3..

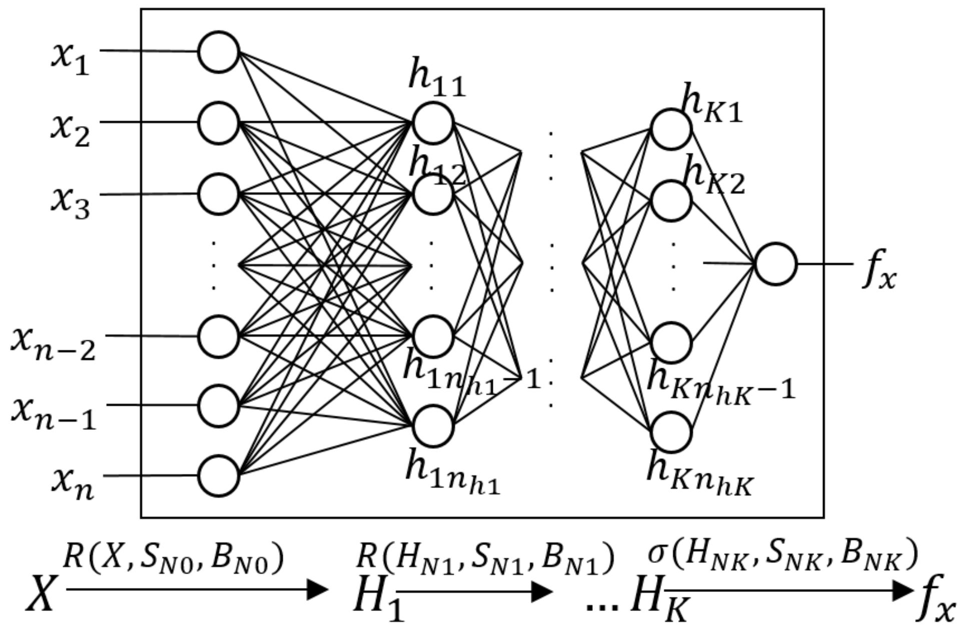


Figure 3.3. Architecture of the neural network used in the host's fitness center to calculate the host's fitness.

Here, $X \in R^n$ is the set of n input values $x_i, \forall i \in \{1,2,\dots,n\}$ that reflect some of the characteristics of the host at that moment, such as its speed and buffer occupancy; $H_i \in R^{n_{hi}}$ is the vector that contains the values h_i(computed according to Equation 3.3) of the n_{hi} neurons in the hidden layer number $i, \forall i \in \{1,\dots,K\}$, where K is the number of hidden layers in the network; and f is the resulting fitness value of the host in the given conditions. The set of weights (synapsis) of the neural network, without its bias values, is given by $S_{N0} \in R^{n \times n_{h1}}$ for the connections between the input layer and the hidden layer 1, and $S_{Ni} \in R^{n_{hi}}$ for the connections between the hidden layer i and the next hidden layer $i+1$, for all $1 \leq i \leq K$, including the connections from the last hidden layer to the output layer. Finally, the bias values of each synapsis are given by $B_{Ni} \in R^{n_{hi}}, \forall i \in [0,K]$. Similarly (see fig. 3.4), $S_{M0} \in R^{m \times m_{h1}}$ is the synapsis for the connections from the input layer to the first hidden layer, and $S_{Mi} \in R^{m_{hi}}$ are the synapsis for the connections from the i-th hidden layer to the next one, including the connections from the last hidden layer to the output layer, and the bias values of each synapsis are given by $B_{Mi} \in R^{n_{hi}}, \forall i \in [0, K]$.

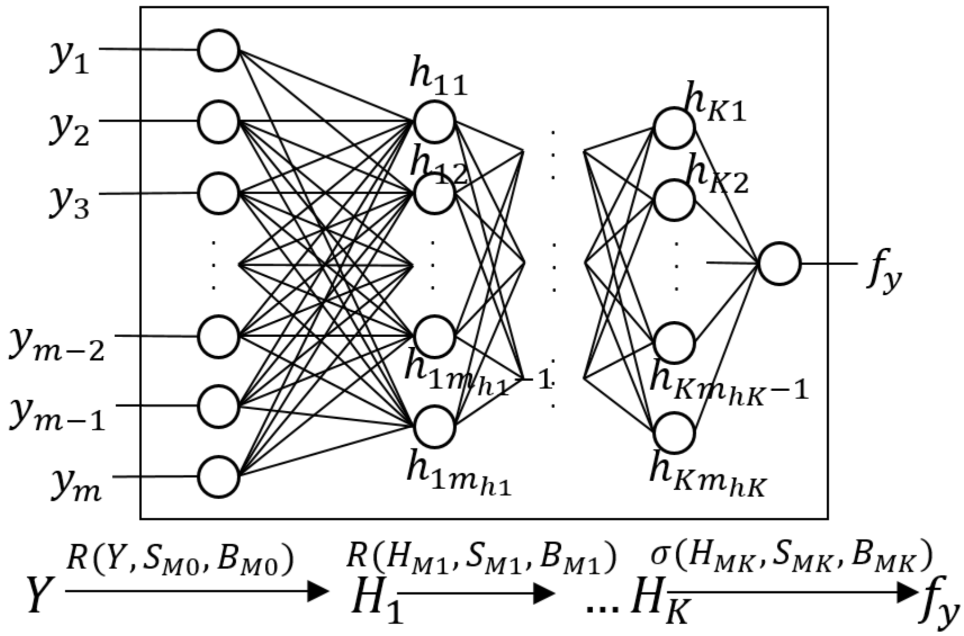


Figure 3.4. Architecture of the neural network used in the host's fitness center to calculate the messages' fitness.

The activation functions in a neural network are functions that transform an input value to another, which determines if it is "passed" to the rest of the network. In an analogy, these *activation* process can be seen as what neurons fire (e.g., are transmitted) to the rest of the network. The Rectified Linear Unit (ReLU, for short) was used as activation function for the neurons in the hidden layers (Equation 3.1), and the sigmoid function $\sigma(z)$ (defined in Equation 3.2) as activation function for the neuron in the output layer. The ReLU function was used because it helps the neural networks have a faster conversion during the training stage (offline), which is necessary in and the sigmoid function was used because we want this value to reflect the fitness of the hosts, and the nature of this function returns values between 0 and 1. Given the nature of the sigmoid function, the closer to 1 is a value f, the fitter the host will be, and vice versa.

$$R(z) = \begin{cases} 0, & z \leq 0 \\ z, & z > 0 \end{cases} \tag{3.1}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.2}$$

Finally, the fitness value for the host is computed taking the current set of features X of the host and making a forward pass through the neural network, as is shown mathematically by Equations 3.3 and 3.4, where $P \cdot Q$ denotes the dot product between P and Q.

$$H_i = R(H_{i-1} \cdot S_{i-1} + B_{i-1}), \forall i \in \{1, \dots, K\} \tag{3.3}$$

$$f = \sigma\,(H_K \cdot S_K + B_K) \tag{3.4}$$

Notice that there a parameter, vector B, called the Bias vector. This set of values is important because it helps the output to adjust along the weighted sum of the inputs to the neuron (it is like the y-intercept in a linear equation). Without this vector, the learning model is not guaranteed to generalize the predictions to unseen data.

The routing algorithm

There are basically two stages in the proposed DLR+ router. The first stage is carried out in the Fitness Center module and is executed whenever the host receives a message and when a connection was terminated, and the second stage is executed when a new connection is established between the host an at least one other router.

In the first stage, the former condition implies that if there is a new message in the queue, then there is the need of recomputing its fitness value, according to its current attributes, to ensure that the conditions in the environment are always up to date. This update is done by making a forward pass through the neural network in charge of the computation of the *fitness* Lack of E2E connectivity among nodes value of messages (there are dedicated neural networks for both the host's fitness computation and the messages' fitness computation). The second condition states that if a connection was just terminated, then very likely the attributes of the host changed, so there is the need of an update of its *fitness* value as well, which is done with a forward pass through its corresponding neural network in the fitness center.

In the second stage, the selection of the best host and best message to start a transmission is performed, by asking for the current connection's fitness and selecting the healthiest message in the message queue of the host. The routing algorithm is summarized below in Algorithm 1 and explained in more detail in the rest of this subsection.

Algorithm 1: DLR+ algorithm. Actions in node c connected to a set of nodes C and having a queue of messages M.

Message received event – msg *fitness* update

Inputs:

m : incoming message

M: c's message queue

Outputs:

M_o: c's message queue, ordered by fitness value

Steps:

1. $Y \leftarrow$ current features y_i of m
2. Normalize Y according to Equation 3.5
3. Compute the value f_y of m according to Equations 3.3 and 3.4
4. **Insert** m in M, in descending order
5. **Return** M

Connection down event – host fitness update

Inputs:

X: the set of features x_i of c

Outputs:

f_x: the updated fitness value of c

Steps:

1. $X \leftarrow$ current features x_i of c
2. Normalize X according to Equation 3.5
3. Compute the value f_x of c according to Equations 3.3 and 3.4

Connection up event – Selection of BNH and BNM dispatch

Inputs:

C: the set of nodes connected to c at that moment

M: c's message queue

Outputs:

C_o: the set of connection tuples ordered by fitness

Steps:

1. Exchange messages whose final destination is in C
2. **Do:**
   for each $\boldsymbol{c_i} \in \boldsymbol{C}$:
   get $\boldsymbol{f_{xi}}$
   if $\boldsymbol{f_{xi}} \geq \boldsymbol{\alpha}$:
   store tuple (c_i, f_{xi}) in C_o
3. **Sort** C_o in descending order
4. **Do:**
   for each $\boldsymbol{m_i} \in \boldsymbol{M}$:
   get $\boldsymbol{f_{yi}}$
   if $\boldsymbol{f_{yi}} \geq \boldsymbol{\beta}$:
   for each $\boldsymbol{c_i} \in \boldsymbol{C_o}$:
   replicate m_i to c_i

To have some sensitivity with respect to other node's fitness, DLR+ uses the parameter α, with $0 \leq \alpha \leq 1$, named as the host fitness threshold, that determines the fitness limit over which the incoming connections may be directly ignored. This value is a key component in the routing protocol in DLR+, because different threshold values result in different dynamics in the opportunistic environment.

In a similar way, the message fitness threshold β was introduced, which determines a limit of fitness for the messages in the queue, above which they can be directly ignored by the message dispatcher. Algorithm 1 summarizes the routing protocol as explained in the following subsections.

f-value update

This first stage takes place each time a connection between the host and another node in the vehicular network has ended. Since some of the host's features may have changed (such as buffer occupancy, dropping rate and others), its fitness value has to be recomputed as well. For this, the considered features x_i are obtained in the Fitness Center, and they are made pass through a process of normalization to obtain normalized features x_i', according to Equation 3.5, where x is a feature that is being transformed, and x_m and x_M are the minimum and maximum registered values of that feature.

$$x' = \frac{x - x_m}{x_M - x_m} \tag{3.5}$$

This will give final input values x_i', with $0 \leq x_i' \leq 1$, which in turn will make the prediction process more reliable. These normalized values are forward passed through the network, according to Equations 3.3 and 3.4 to get the final updated f value.

A similar process is executed each time a message is received by the host. Whenever this happens, the f value of the incoming message is computed according to Equations 3.3 and 3.4 in its corresponding neural network. Finally, the message is put in the queue according to its fitness. This way, the message queue is always ready with the messages ordered by the fittest message first.

BNH selection and packet forwarding

The second stage of the routing process occurs when a link is established between the current host and some of its neighbor nodes. At that moment, the router will attempt to exchange deliverable messages (i.e., messages whose final destination is among the current connections), if any. Then, the host router asks the connected nodes for their fitness values (which, thanks to their Fitness Center, are always up to date). After that, before entering the final selection, the router directly discards those connections whose f value is not at least the fitness threshold α, and orders in descending order the remaining connections, according to their fitness. With a complete list of fit candidates, the selection process is straight forward: the best next hop will be the fittest node (the one with the higher f value), so the router will attempt to replicate a data package to the nodes in that order.

Summary of Chapter 3

In this chapter, the research problem has been formalized, presented as a binary classification task to quantify the capabilities of the available nodes and messages in a given connection as candidates for the next transmission. Also, the routing architecture and the routing protocol have been introduced, detailing each of the modules, stages and steps in the transmissions. In summary, the parameters (i.e., *fitness* values) that help the router know which message to transmit from its queue and to which node from the incoming connections are computed doing a forward pass through the corresponding neural networks in the *fitness center* upon a "message received" and "connection down" events, respectively. In this way, these decision values are always ready to be used when there is a new transmission or when a neighbor host asks for the host fitness value. Furthermore, the router parameters α and β (fitness threshold values) assist the router in the final decision as to what message from its queue transmit and to which of the routers connected to it at that moment. As we can intuitively infer, different values of these threshold result in different dynamics in the vehicular network and in the performance of the routers as well.

In the next chapter, the design, setup and run of the experiment will be introduced and explained in more detail.

Chapter 4

Experiment

As explained in former sections of this report, the lack of common testbeds and a unique simulator besides the incomplete technical specifications given by many authors of the proposed routing algorithms found in the literature obstructs the implementation process of their algorithms, severely affecting reproducibility. In this Chapter, the design and execution of the experiment to validate the proposed solution is presented. First, the general setup is explained, and following the router and neural networks tuning as well as the evaluation metrics considered in this experiment. Also, for the benefit of the readers, technical aspects and datasets are given to ensure reproducibility of the experiment.

Simulation setup

The Opportunistic Network Environment simulator (The ONE simulator [39]), which is a virtual environment based on Java designed to test opportunistic networks, was used as the main tool to replicate a synthetic scenario for the experiment. This test scenario was a portion of Queretaro City, a medium sized state in Mexico, with little over 2 million inhabitants. The scenario was delimited by a 1000m by 1200m squared terrain (fig. 4.1).

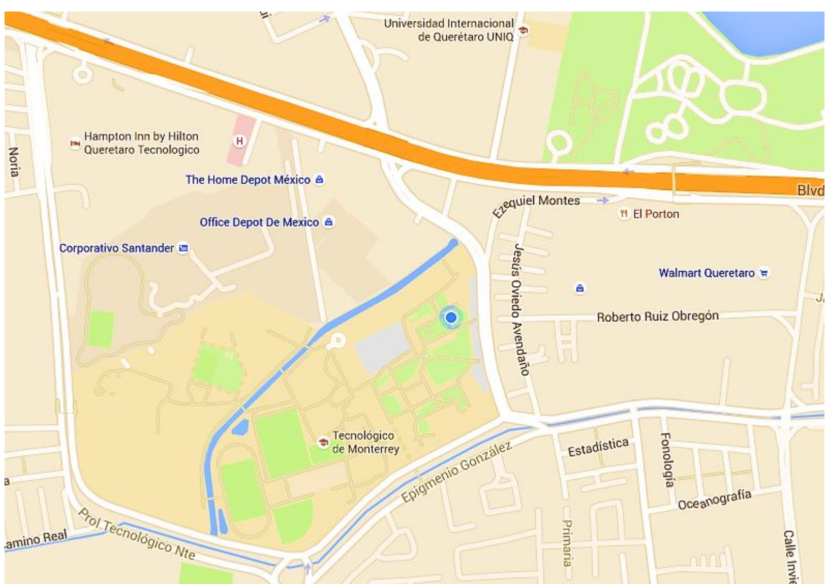


Figure 4.1. The roads and streets used in the simulation.

The simulation time was 43200 seconds (12 hrs.), which is considered as a typical span of a working day, and even though the dynamics of the vehicles may be different at different hours, the same scenario was used for all the routers, so the influence that these selection has in the final routing performance of the routers is irrelevant.

Mobility Model

One of the features that helps make the simulation more realistic is the model that governs the movement of the nodes in the vehicular network, providing coordinates, speeds and pause times for the nodes. Popular models include [64] random waypoint (nodes move randomly in arbitrary direction with random speeds), map-based movement (nodes move based on predefined paths, such as for streets and avenues), and shortest path map-based movement (nodes move based on predefined paths in a map, following the shortest path between origin and destination; see fig. 4.2).

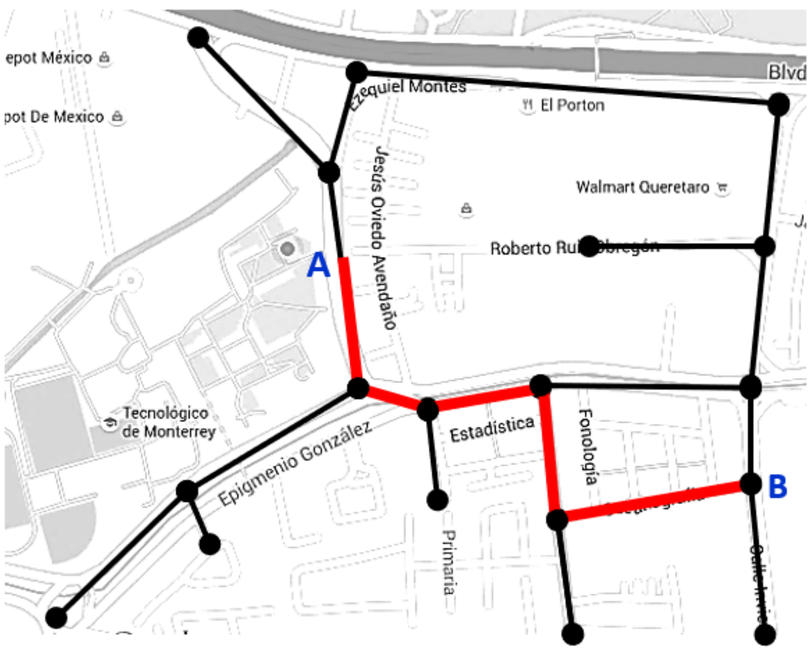


Figure 4.2. Shortest Path Map-based Movement mobility model. The model takes picks two random points A and B in the graph, and makes the vehicles move from A to B following the shortest route in the connected graph; when the vehicle gets to B, pauses for a random amount of time, and the process starts again.

Even though this experiment was carried out in a synthetic scenario, having as many real life similarities helps to capture the dynamics of a more realistic environment, so the later model was used for the simulation, which constrains the node movement to predefined paths, using Dijkstra's shortest path algorithm to find its way through the map area. Under this model, once one node has reached its destination, it waits for a pause time, and then another random map node is chosen, and the node moves there repeating the process. The same mobility model was used for all the routers in the simulation, and the comparative analysis of the influence of mobility models in the routing dynamics is left as part of the future work (several analysis of that influence are already in the literature, such as in [28] and [67], where they show that this model, the Shortest Path Map-based Movement model, results in higher packet delivery ratio than other popular mobility models).

Host groups and routers

It's worth noticing that there is no clear-cut procedure to choose the number of vehicles and their features in a vehicular scenario, and the characteristics may influence the final performance of any router put to work under those conditions. For instance, the speed range at which vehicles move, the number of vehicles in the simulation (i.e., node density in the VDTN), and the transmission interface can heavily have an impact in the vehicular network dynamics. For this simulation, representative samples of agents in a vehicular network were used, including different moving speeds, transmission ranges, and transmission interface were used, but these conditions were chosen in a subjective way. Nonetheless, these conditions are the same for all routers in the experiment, so all of them are influenced in the same way (this synthetic environment was the same for each node and for each type of router).

In the experiment, there was a total of 85 nodes, divided into 8 different groups, each with particular characteristics. The Wireless Access for Vehicular Environment (WAVE) IEEE 802.11p Standard [76] established a minimum of 3Mbps and a maximum of 27Mbps speeds for wireless communications. Thus, we decided to include connections at 6Mbps, 12Mbps and 24Mbps. Also, we included some Bluetooth connections at 2Mbps. Higher connection speeds where not considered in this experiment. That is an important research opportunity, i.e. the comparison between the IEEE 802.11p and C-V2X or even 5G standards when it comes to high transmission speeds and very high node densities (e.g., when there are extremely large number of vehicles), but that is left as part of future work. Finally, the buffer size, maximum node speed and number of nodes of each type are shown in Table 4.1., along with the rest of the network simulation parameters.

Group	Nodes	ID	Buffer size (MB)	Speed range (m/S)	Interface	Description
1	10	p1	5	0.5 – 1.5	Bluetooth	A group of pedestrians
2	10	p2	5	0.5 – 1.5	WAVE 802.11p@6Mbps	Another group of pedestrians
3	5	b1	10	2.7 – 16.7	WAVE 802.11p@6Mbps	A group of buses
4	10	b2	10	2.7 – 16.7	WAVE 802.11p@12Mbps	Another group of buses
5	15	c1	15	5.5 – 22.22	WAVE 802.11p@12Mbps	A group of low-speed cars
6	15	c2	15	5.5 – 22.22	WAVE 802.11p@24Mbps	Another group of low-speed cars
7	10	c3	20	8.3 – 30.56	WAVE 802.11p@12Mbps	A group of high-speed cars
8	10	c4	20	8.3 – 30.56	WAVE 802.11p@24Mbps	Another group of high-speed cars

Table 4.1. Group of nodes in the simulation.

As it is intuitively thought of, the time-to-live (TTL) of a message has a direct influence on the dynamics of the VDTN, because of the direct relationship to the availability of the messages in the network. This is, after certain amount of time messages are destroyed, regardless of whether they have reached their destination or not, so they are no longer available for transmission, and it also has an impact on the buffer availability. In this experiment, the TTL, in minutes, was iterated from the list TTL= {0, 25, 50, 75, 100, 150, 200, 300} to have a broader understanding of the behavior of the router.

As for the routers, the main simulation was done with DLR+, and tests against Epidemic, Spray and Wait, PRoPHET and SeeR were performed. The selection of the routers is based on the fact that the aforementioned routers are the most popular routers in each category, as presented in the Related Work section. One issue of concern with most of the routing proposals in the literature is that the authors do not include enough technical details to guarantee reproducibility of their work, and that makes the replication of the experiments and the utilization of their proposals in comparative analysis rather difficult. As for machine learning-related routers, for starters, none of the routers propose a Deep-Learning architecture, and for the routers that include significant traces of neural networks [4][63], unfortunately, the authors do not provide enough technical specifications to reproduce their models, so those routers were not included in the simulation.

Finally, the metrics used in the experiment were Packet Delivery Ratio (PDR), Average Delivery Delay (ADD), Network Overhead (OVH) and Hop Count (HOP), which are the most popular metrics used for routing protocol evaluation [57]. These routers and metrics were explained in former sections in this report.

The Neural Networks in DLR+

The Neural Networks general architecture used in DLR+ was presented in the previous section. As noted, all the neural network parameters were left as variables, meaning that they can be adjusted in future versions as desired. In this simulation, the following considerations were made.

Layers

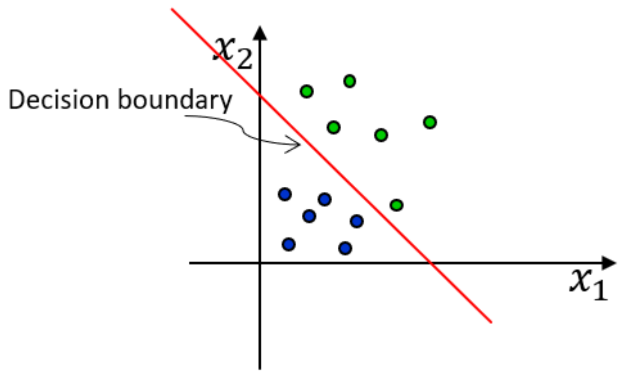
The neural networks considered in this work are deep feed forward neural networks with 2 hidden layers, which provide the capability to capture complex non-linearities in the system. This way, the networks consisted in an input layer, two hidden layers, and an output layer.

The input layers. These layers are the door of the model. Here, the data that reflects the current conditions of the host (features) is entered to be processed by the neural network and come up with a prediction (in the output layer) to see the *fitness* of the node. Feature selection is a key task in the prediction model, because such features will have an impact in the results of the predictive model. However, it's not a not a trivial task when modeling systems, and a lot of the times it comes down to intuition and trial and error [53]. In the literature, the most used features of nodes in vehicular networks are the *physical speed* of the vehicles, the *transmission speed,* the *transmission range* and the *buffer size* [61][66] [81]. These features were included in the proposed model. Additionally, another feature that can intuitively influence the dynamics of the packet forwarding decisions is the number of connections over time (*average number of connections*), because it somehow reflects the number of encounters, and therefore the probability of forwarding to another hosts. One more feature in this direction is the *buffer occupancy,* which provides information not only on the total buffer size, but how much of it is available; this feature may have an influence in the forwarding decisions because it has an impact on the incoming packets: it there's enough buffer available, then they are accepted, or discarded otherwise. Finally, the *dropping rate* and the *abort rate* are also feature that may throw information on how the packets are normally treated (e.g., how frequently the packets are discarded, and that might mean issues such as lack of buffer space or poor memory allocation, or how frequently does the host abort transmissions, and that might mean issues on connectivity, such as short transmission ranges or inability to handle high speed connections). We have included all those features in the model. The number of neurons in the input layers is the number n of features to process from each sample in the classification process, and thus, for this version of DLR+, for the host's fitness, 8 different features x_i were considered. Additionally, in order to help in the capture of the non-linearities in the system, 8 extra features $x_j = x_i^2, 1 \leq i \leq 8$ were introduced in the model, for a total of $n = 16$ input features, listed in Table 4.2.

Feature	**Name**	**Description**
x_1	Host speed	Speed (m/S) at which the vehicle is moving
x_2	Transmission speed	Transmission speed of the communications link (Mbps)
x_3	Transmission range	Maximum radial distance (m) at which the host can connect to other nodes
x_4	Avg number of connections	The number of connections, on average, that a host handle
x_5	Buffer size	Buffer size (MB)
x_6	Buffer occupancy	Percentage of buffer occupancy
x_7	Dropping rate	Rate at which a host drops packets
x_8	Abort rate	Rate at which a host aborts packet transmissions
$x_i = x_{i-8}^2$, for $9 \leq i \leq 16$		Composite features to help capture non-linearities

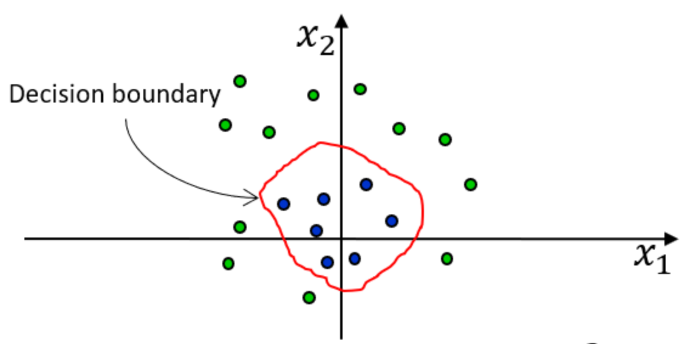
Table 4.2. Features considered in the first neural network (for host fitness) in DLR+.

Non-linear systems are systems whose output is not proportional to the change of the inputs, which is very common in systems of very complex dynamics such as the one we are trying to model here, where vehicles move in a very unpredictable manner. Including non-linearities in a model results in a better prediction, because that means that the decision boundaries are not purely linear [82]. To illustrate this, in Figure 4.3 we can see an example of a linearly separable system (two features), with its corresponding equation, whereas in Figure 4.4 a non-linearly separable system (two features) is presented as an example, with its corresponding equations. Something similar is expected to happen in the model of the vehicular system, where having a function $F(x_i)$ that depends not only in linear terms of the parameters, but also in non-linear terms can increase the precision in the predictions.



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Figure 4.3. Example of a linearly separable system (adapted from [82]). Notice that the classification function only depends on linear terms of the features.



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Figure 4.4. Example of a non-linearly separable system (adapted from [82]). Notice that the classification function not only depends on linear terms of the features x_1 and x_2, but also in higher order terms, such as quadratic terms x_1^2 and x_2^2. This makes possible to have decision boundaries with different shapes other than just a straight line.

For the second neural network (the one that takes care of the messages fitness), a total of $m = 3$ different features were used, the features most used in the literature [61]. Those features are briefly described in Table 4.3. We also included the squared features but did not notice any gains in accuracy during the training process, so we decided to just take them away in the final version of the proposed router.

Feature	Name	Description
y_1	Residual TTL	Also known as time-out ratio, is the ratio of the remaining TTL to the initial TTL
y_2	Message size	Size of the message (Bytes)
y_3	Hop count	The number of nodes that the message has traversed so far

Table 4.3. Features considered in the second neural network (for message fitness) in DLR+.

The hidden layers. The Universal Approximation Theorem [6] establishes that "a neural network with a single hidden layer with a finite number of neurons can approximate any continuous function on compact subsets in R^n"; this implies that, finding the appropriate parameters, a neural network with one single hidden layer is enough to represent a great amount of problems. Nonetheless, the width of such layer might become exponentially big. Indeed, Ian Goodfellow, a pioneer researcher on Deep Learning, holds that "a neural network with a single layer is enough to represent any function, but the layer can become infeasibly large and fail to learn and generalize correctly" [20]. On the other hand, while not having hidden layers at all in the neural network would only serve to represent linearly separable functions, a hidden layer can approximate functions with a continuous mapping from a finite space to another, and two layers can represent an arbitrary decision boundary with any level of accuracy [27]. In summary, this means that one hidden layer helps to capture non-linear aspects from a complex function, but two layers help generalize and learn better. In fact, the authors hold that one rarely needs more than two hidden layers to represent a complex non-linear model. With this information, we opted for two hidden layers in this version of DLR+. As for the number n_{hi} of neurons in each hidden layer H_i , there is no formula to have an exact number, although some empirical rules can be used [32]. The most common assumption is that the optimal size of the hidden layers is in general between the size of the input layer and the size of the output layer. For this module in DLR+, this would mean that $n = 16 \geq n_{hi} \geq 1$. Another suggestion is to keep this number as the mean between the number of neurons in the input and output layers and from here start decreasing the number of neurons in each subsequent layer without falling below 2 neurons in the last hidden layer. For this module in DLR+, this would imply that $n_{h1} = 8$ and $2 \leq n_{h2} \leq 8$. One last suggestion to avoid overfitting during the training process (which would mean that the neural network would have great memory capacity, but no prediction capabilities over unseen data) is to keep the number of neurons in the hidden layers as in Equation 3.6, where N_s is the number of simples in the training set, N_i is the number of neurons in the input layer, N_o is the number of neurons in the output layer, and γ is an arbitrary scaling factor, generally with $2 \leq \gamma \leq 10$.

$$n_{hi} < \frac{N_s}{\gamma\ (N_i\ +\ N_o)} \tag{3.6}$$

In this context, this would mean:

$$n_{hi} < \frac{N_s}{\gamma\ (10\ +\ 1)} = \frac{N_S}{11\gamma} \tag{3.7}$$

Ultimately, nonetheless, the number of neurons in the hidden layers comes down to trial and error. Following these suggestions and seeking a short computational time, we opted for $n_{h1} = 14$ and $n_{h2} = 10$. In a similar way, we decided to use $m_{h1} = 5$ and $m_{h2} = 3$ for the messages' neural network.

The output layer. The output layer in both neural networks (the one for the host fitness and the one for the messages) has a single neuron, that, according to Equation 3.2, will have a value between 0 and 1. During the training process, this value is further converted to a digital value, so each sample has a unique label $l \in \{0,1\}$, given by Equation 3.8, where f is the value returned by the sigmoid function in the last part of the forward pass.

$$l = round\left(\frac{f\ +\ 0.5}{2}\right) \tag{3.8}$$

This labeling process is used to compare and evaluate the prediction class during training. However, we have to remember that during the application of the neural networks in the VDTN environment this labeling process must not be done, because we are only interested in identifying the simples with the best fitness (this is, the sample with the highest f value), which is directly given after the forward pass by the sigmoid function (see Equations 3.3 and 3.4).

Training

DLR+ uses $K + 1$ synapses matrixes S_i with their corresponding bias vectors B_i, with $i \in \{0, \ldots, K\}$, where K is the number of hidden layers of the deep neural networks, as introduced before in section IV-B. These matrixes are obtained during the training process by using a dataset with samples gotten from a simulation scenario with the conditions defined in section VI-A. More particularly, the hosts were configured to be one of the three popular routers PRoPHET, Spray and Wait and SeeR, and a total of 11,016,000 sample vectors $X = [x_1, x_1, \dots, x_8]$ were obtained from a simulation with a simulation time of 43200 seconds (12 hrs.), gathering the current features x_i of each of the 85 hosts each second. The labels l for each sample were directly obtained from the feature final delivery rate (FDR), considering that the more messages a host delivers to a final destination, the closes to a fit node it must be. For this, the samples were passed through a standardization process and the ones that got a positive z-score were considered as "fit" ($l = 1$) according to Equation 3.9, where x is the value of the aforementioned feature FDR, $\bar{x}$ is the mean of all those FDR values in the data set, and σ is the sample standard deviation.

$$l = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases}, \text{ with } z = \frac{x - \bar{x}}{\sigma} \tag{3.9}$$

In preprocessing, all duplicated records were deleted from the original data set, and all remaining values were normalized for each feature x_i/y_i, according to Equation 3.5, to have a better mapping and a faster convergence during training; finally, the final dataset was randomly permuted. From this, the resulting data set was split into two subsets for real training (80% of the data) and validation (20%), to assess the learning process and generalization. Other hyperparameters of the neural networks were Adam optimizer (faster than the traditional stochastic gradient descent, [40]) and binary cross-entropy as an error function. This way, we got 90.12% accuracy in the training set and 90.55% in the validation set. This is how synapses and bias matrixes S_i and B_i used in DLR+ were obtained, whose final values are included in the source files.

Last but not least, the training of the neural networks was done in Python 3.7, using the Spyder environment from the Anaconda 3 distribution. For the benefit of the reader, the python scripts, the whole dataset used for training and the implementation of the router in The ONE are provided in the following link:

https://bit.ly/sourceFilesDLRplus

The fitness thresholds in DLR+

As described at the beginning of section 5, the fitness threshold $\alpha \in [0,1]$ is a router parameter used to discriminate "bad" from "good" nodes as explained in the routing algorithm definition. This value can be any real number between 0 and 1, each possibility resulting in a different router performance, as can be seen in the results section (Chapter 5). It was found that α=0.65 offered the optimal performance, so that's the default value for this parameter in DLR+. As for the β value, no significant differences for values different than 0 were observed, so it was decided to use $\beta = 0$ as de default value. It is worth emphasizing, though, that different values of these parameters result in different dynamics and response of the router, in a given environment; however, the dynamics, and, therefore, the values of these parameters, might not be the optimal for other scenarios (e.g., because of the influence of TTL of the messages, node density or other parameters in the environment).

Summary of Chapter 4

In this Chapter, the design, setup and run of the experiment was presented in detail, explaining each of the features considered in the scenario, such as the mobility model, the TTL of the messages and the host groups. Additionally, the characteristics of the neural networks used as core of the proposed router architectures were explained, such as layers, features and activation functions. Finally, the training stage of the learning model and the corresponding evaluation metrics were also presented. As we can see, in Deep Learning techniques, such as Deep Neural Networks, one of the main tasks in the design is the parameter tuning, and many times different scenarios require different hyperparameter values of the neural network, and there is no clear-cut procedure to know this beforehand, and ultimately most of the design comes down to trial and error in order to get the best performance in the training data. Some rule of thumbs can be used, nonetheless, for the number of layers (a minimum of 2 hidden layers is recommended, so the model can capture the non-linearities of the system) and for the number of neurons in each hidden layer.

In the next chapter, the results obtained in the experiment are discussed.

Chapter 5

Results

In this section, a description and comments on the simulation results are provided.

Effect of TTL

As can be seen in the subsequent plots, the time to live of the messages has a significant impact on the metrics to a certain extent, as the longer a message exists, the higher the probability it has to be delivered to its final destination. Any metric value, however, tends to plateau as more TTL is granted. It was found that the TTL value at which the metrics began to settle in a notable way is around 300 min. This means that adding more time-to-live to the messages will not add any significant improvements in the performance. Also, depending on the router, some of them will exhibit a better performance when the TTL is smaller than that of the settling point, as we see that, for instance, in the PRoPHET and Epidemic Routers (figs. 5.5 and 5.6). Therefore, at least a minimum of TTL=300 min is advised when evaluating router performance to capture the complete behavior. These threshold value, however, might be slightly different for different scenarios, but given the dependence of all the parameters in the vehicular networks, such as the buffer size of the hosts, the node density and even the map used for the physical movement of the vehicles, it is guaranteed that the TTL has a strong influence in the final performance of the routers, regardless of their type, so when running simulations, it is advised to include a sweep of TTL values up to around 300 minutes.

Effect of the fitness thresholds

As describe in the previous section, the α parameter is a value that determines to what extent some of the connections are immediately discarded as next hop candidates. Intuitively, a very small value would mean that only a small portion of the current connections are discarded, so most of them have a chance to be chosen (although in descending order with respect to their fitness values). The limit is $\alpha = 0$, and since $1 \geq f \geq 0$, the condition $f \geq \alpha$ means in this case that all the connections are considered as potential candidates. Similarly, a very large value of α will result in a strong limiting condition, meaning that only the very best hosts (the ones with considerably large fitness) will be considered as possible next hops. This can be counterproductive, nonetheless, especially in very early stages of the simulation, because at the beginning there might not be a lot of "healthy" nodes to carry the messages, and thus a lot of them are prone to be skipped, resulting in larger delays. As we can infer from this explanation, the dynamics of the environment are strongly influenced by the α value. To better understand the effect of this fitness threshold, simulations were run changing this parameter with $\alpha = \{0, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5, 0.65, 0.8, 0.95, 1.0\}$. Also, the TTL of the messages varying from $TTL = \{10,25,50,75,100,150,200,300\}$.

Finally, a similar reasoning than that for α was made for the β fitness threshold, so we considered $\beta = \{0, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5, 0.65, 0.8, 0.95, 1.0\}$ in the simulations as well (all the plots can be found in Appendix 6-9).

We distinguished two main differentiators in both the α and β values: $\alpha = 0$ and $\alpha > 0$, and $\beta = 0$ and $\beta > 0$. In the first case, with $\alpha = 0$, we can see that the cases $\beta = 0$ and $\beta > 0$ resulted in noticeable different dynamics (see figs. 5.1-2). We notice that for $\alpha = 0$, for TTL values smaller than 60, the performance of DLR+ is better with $\beta = 0$ for PDR.

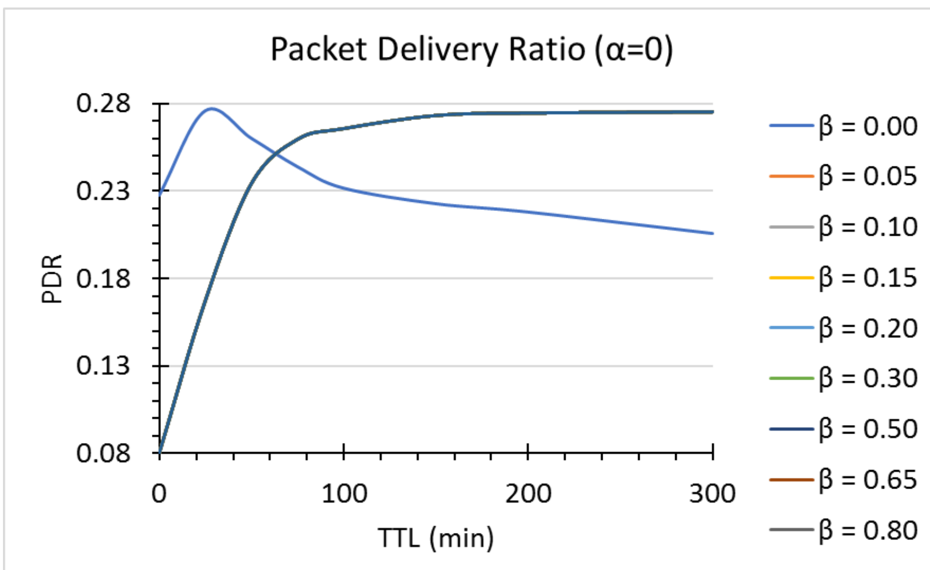


Figure 5.1. Effect of the fitness thresholds in Packet Delivery Ratio.

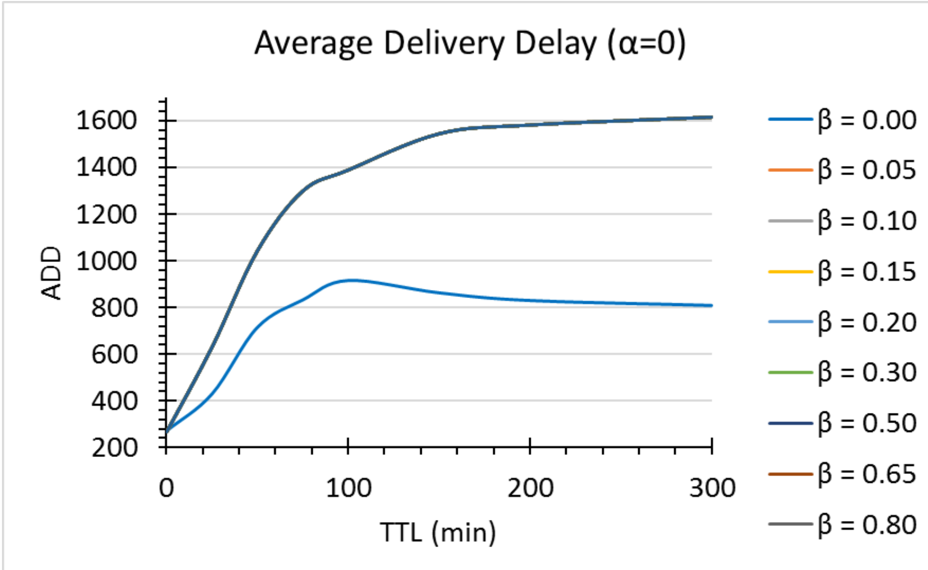


Figure 5.2. Effect of the fitness thresholds in Average Delivery Delay.

For ADD, in turn (fig. 5.2), $\beta = 0$ is the choice, as it showed better results than for other β values. In any case, however, for OVH and HOP the choice is any value different than 0 for β (figs. 5.3 and 5.4). As we can see, there is a tradeoff mainly between network overhead and delivery ratio or delivery delays, and the final choice of the parameters ultimately depends on the final application of the router in delay-tolerant networks (i.e., if we are interested in minimizing latency, at the expenses of some overhead, or we have limited resources, such as in mobile sensor networks).

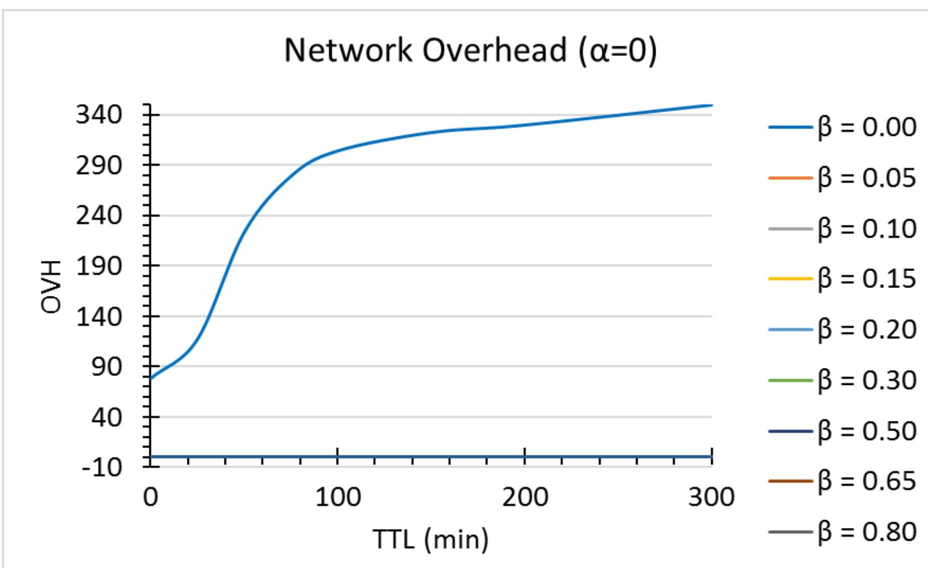


Figure 5.3. Effect of the fitness threshold in Network Overhead.

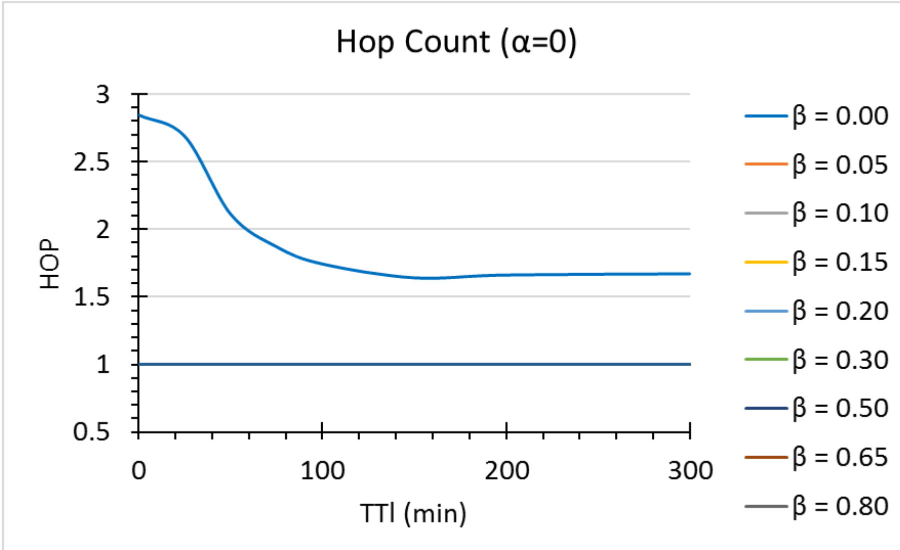


Figure 5.4. Effect of the fitness threshold in Hop Count.

For $\alpha > 0$, we did not notice any significant difference in the values of β. Finally, for $\alpha > 0.5$ there was a slightly improvement in overhead and number of hops. Based on these results, for this version of DLR+, we decided to use $\alpha = 0.65$ and $\beta = 0$, although these parameters can be tweaked, depending on the scenario in which they are used.

Performance of DLR+

In this subsection we discuss the final performance of DLR+ ($\alpha = 0.65/0, \beta = 0$) and compare it against other well-known routers (figs. 5.5-5.8).

Performance on Packet Delivery Ratio

As can be seen in Figure 5.5, DLR+ ($\alpha = 0.65$) offers a greater PDR than the Epidemic router and PRoPHET for TTL greater than 60 and 130, respectively. And although its performance on this metric is not the best, it is very close to those who offer the best values, only about 6.07% below its better counterparts. On the other hand, with $\alpha = 0$, DLR+ outperforms all routers in PDR for $TTL < 25$. This reflects an interesting dynamic in the response of DLR+ for this case, in contrast with other routers: the more TTL is provided, the more inefficient the router becomes; however, as TTL is smaller, the response of the proposed router increases, outperforming the other routers in this and other metrics, and particularly having a high PDR (very close to the ones shown by the best routers in this metric), and a low ADD, very similar to the one from the other routers at this point. There is a tradeoff, nonetheless, in this range of operation, because in this part DLR+ ($\alpha = 0$) does not have the best performance in network overhead and hop count (figs. 5.7 and 5.8), although it shows acceptable values, very close to the ones shown by other routers.

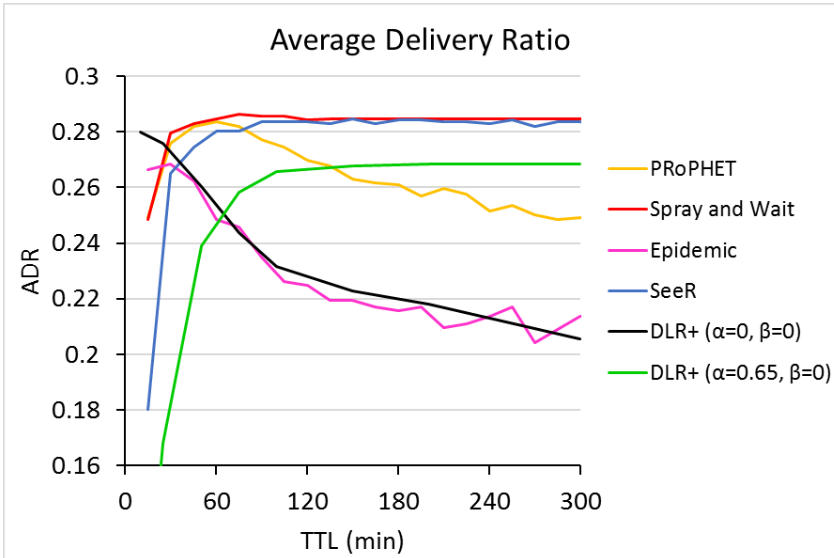


Figure 5.5. Performance of DLR+ in Packet Delivery Ratio.

Performance on Average Delivery Delay

In the long run, DLR+ does not provide the best performance on Average Delivery Delay (fig. 5.6). We can see that as the TTL increases, so do the delivery delay values, and although they tend to stabilize at some point, there are significant differences with respect to otter router performances. The proposed router, however, performs fairly well for small TTL values, laying in points very close to those resulted from their counterparts, with roughly the same ADD values than those of other routers for TTL≤25.

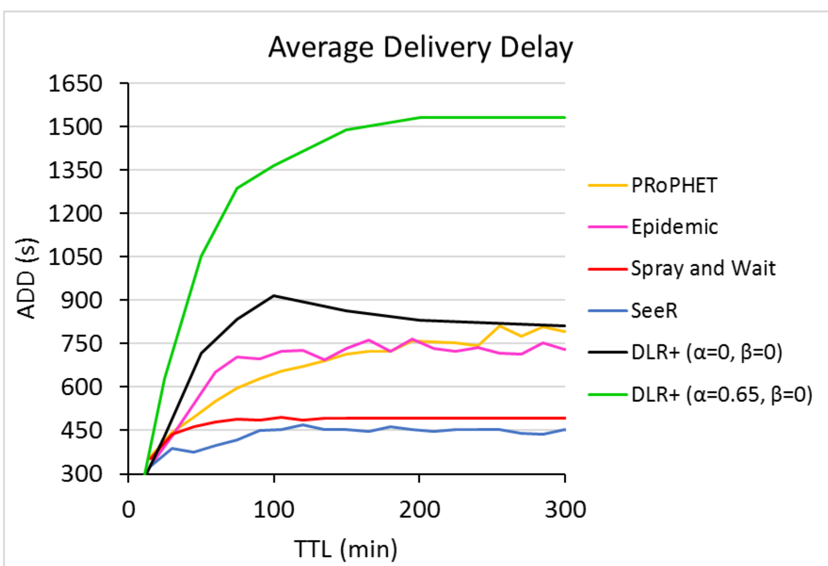


Figure 5.6. Performance of DLR+ in Average Delivery Delay.

Performance on Network Overhead and Hop Count

As can be seen in fig. 5.7, DLR+ ($\alpha = 0$) did not have the best results in Network Overhead, with significant differences with respect to their counterparts, closely resembling the Epidemic routing. For $\alpha = 0.65$, however, DLR+ had the best performance, with nearly zero overhead, which means extremely efficient resource usage, way below the OVH values returned by other routers.

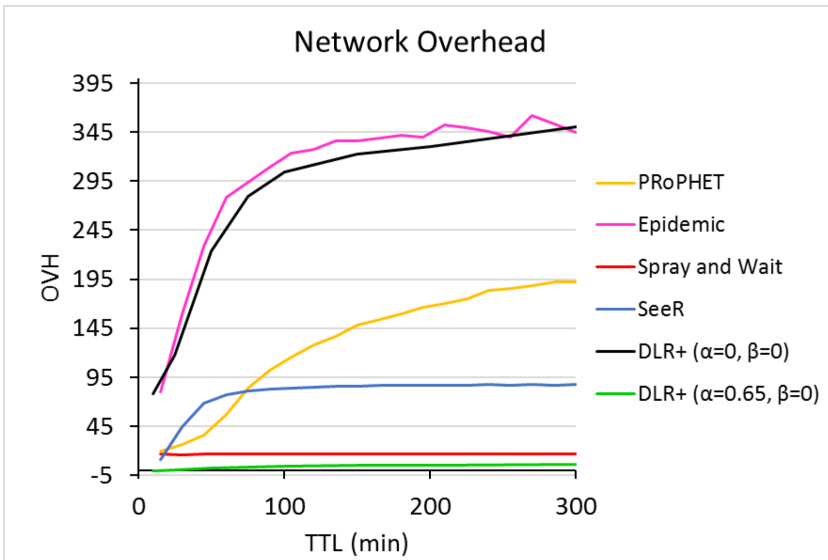


Figure 5.7. Performance of DLR+ in Network Overhead.

In hop count (fig. 5.8), on the other hand, with $\alpha = 0$ the number of hops used by DLR+ is very close to a constant 1.6 in the long run, which shows better values than other routers. Indeed, for TTL>50 the proposed router ($\alpha = 0$) outperforms all other routers in the experiment, but even for TTL values smaller than 50, the number of hops used by DLR+ is between 2.2 and 2.8, which is a range in which all other routers lie as well. For $\alpha = 0.65$, however, the proposed router shows an impressive HOP of nearly 1, which is a very significant difference with respect to the rest, confirming the highly efficient usage of network resources. The intrinsic dependence of all the metrics, nonetheless, let us see that even though the performance in this metric was very good, that may have an influence in the performance in the other metrics, such as network overhead (to its favor) and packet delivery rates and average delivery delays, which in turn show a performance close to the maximum observed in other routers (for PDR) and a bad performance when it comes to ADD, with respect to other routers. Intuitively, an adjustment in this trade-off can result in better performance in the first two metrics.

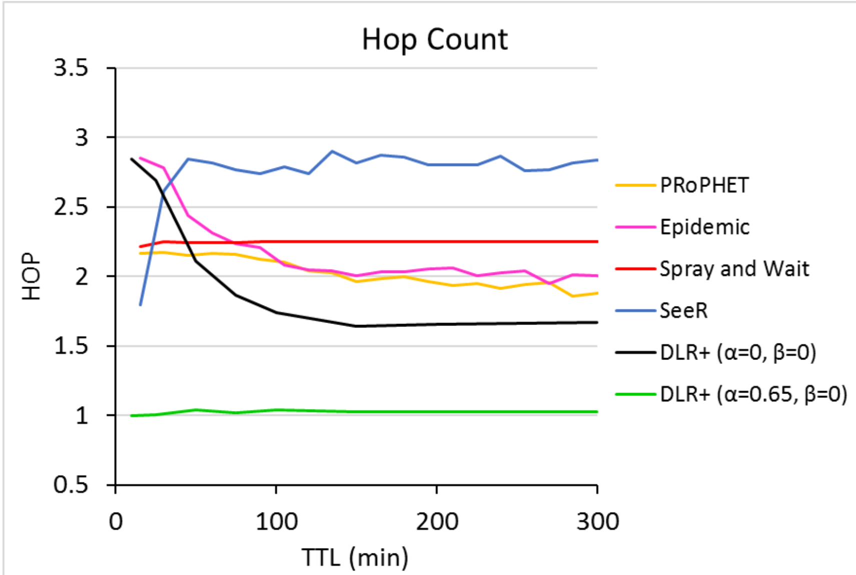


Figure 5.8. Performance of DLR+ in Hop Count.

Table 5.1 summarizes the information explained above, separated into the two ranges of TTL observed in the experiment ($TTL \leq$ 25 min and $TTL >$ 25 min). And as can be seen, there are certain regions in the TTL spectrum where DLR+ stands out with respect to the other routers, or is close to the top performers.

	TTL<25		TTL>>25	
PDR	$\alpha = 0$	Better than all routers	$\alpha = 0.65$	Close to the top routers
ADD	$\alpha = 0$	Similar to all routers	$\alpha = 0$	Close to PRoPHET and Epidemic
OVH	$\alpha = 0.65$	Better than all routers	$\alpha = 0.65$	Better than all routers
HOP	$\alpha = 0.65$	Better than all routers	$\alpha = 0$, 0.65	Better than all routers

Table 5.1. Summary of performance of DLR+.

Finally, in Figure 5.9, the strengths in the long run (for very large TTL values) of each router against their counterparts is shown, where ✓ means, that the router was better in the corresponding metric with respect to the given router.

DLR+ (α=0.65)

	PDR	ADD	OVH	HOP
DLR+ (α=0)	-	-	✓	✓
Epidemic	✓	-	✓	✓
Spray & Wait	-	-	✓	✓
PRoPHET	-	-	✓	✓
SeeR	-	-	✓	✓

Epidemic

	PDR	ADD	OVH	HOP
DLR+ (α=0.65)	-	✓	-	-
DLR+ (α=0)	-	✓	-	-
Spray & Wait	-	-	-	✓
PRoPHET	-	-	-	-
SeeR	-	-	-	✓

Spray & Wait

	PDR	ADD	OVH	HOP
DLR+ (α=0.65)	✓	✓	-	-
DLR+ (α=0)	✓	✓	✓	-
Epidemic	✓	✓	✓	-
PRoPHET	✓	✓	✓	-
SeeR	-	-	✓	✓

DLR+ (α=0)

	PDR	ADD	OVH	HOP
DLR+ (α=0.65)	-	-	-	✓
Epidemic	✓	-	✓	✓
Spray & Wait	-	-	-	✓
PRoPHET	-	-	-	✓
SeeR	-	-	-	✓

PRoPHET

	PDR	ADD	OVH	HOP
DLR+ (α=0.65)	-	✓	-	-
DLR+ (α=0)	✓	✓	✓	-
Epidemic	✓	-	✓	-
Spray & Wait	-	-	-	✓
SeeR	-	-	-	✓

SeeR

	PDR	ADD	OVH	HOP
DLR+ (α=0.65)	✓	✓	-	-
DLR+ (α=0)	✓	✓	✓	-
Epidemic	✓	✓	✓	-
Spray & Wait	-	✓	-	-
PRoPHET	✓	✓	✓	-

Figure 5.9. Summary of strengths of each router compared to the other routers.

It can be seen that none of the routers is better than all the others in every metric. Furthermore, none of the routers is better than the others in a given metric, except SeeR, which is the best than everyone else when it comes to delays, and the proposed router, DLR+ (α = 0.65), which beats all the other routers in network overhead and hop count. This clearly reflect that there must be a trade-off between the metrics in a given router. Normally, it is possible to have a better packet delivery ratio or a better average delivery delay at the expense of network overhead and hop count. With this in mind, DLR+ can be used in situations where the resources are limited or the network suffers high congestions, whereas in critical, safety-related services and applications SeeR or Spray & Wait are preferred. It's worth emphasizing, however, that these comparisons are true when TTL values are considerably big, which is a general assumption because in reality TTL values offer different dynamics for smaller values, as we compared before for TTL<25 minutes, so ultimately the final decision as to what router to use will depend on the particular scenario being addressed.

Summary of Chapter 5

In this chapter, the results obtained in the experiment were explained in detail. One of the main observations is that the TTL has an important effect on the performance of the network, and it is observed that for TTL close to 300 min, the values on the evaluation metrics tend to plateau. It will be interesting, though, to see if for very large values of TTL, the ADD of the proposed router improves, given that a lot more of information can the hosts have access to of the environment during that time (this is, the hosts could learn in a more deepest way the dynamics in the very long run). Also, the effect of the fitness thresholds was discussed individually for each of the metrics in the simulation, and the values of $\alpha = 0.65$ and $\alpha = 0$ were the ones that show different responses with respect to other values of α, but for β there were no significant differences in the performance, so $\beta = 0$ was used for this synthetic scenario. It's worth noticing that the values of these parameters can be different for other scenarios, and they can be tweaked as needed. Finally, the performance of the proposed solution was assessed according to each of the metrics, and one key conclusion is that there must be a trade-off between the metrics that are in play, due to an intrinsic dependency of them. For instance, when getting the minimum hop-count, which is 1, the network overhead naturally decreases, too, but there's an increase in delivery delay, because the nodes have to wait until the conditions for direct delivery are present, which may not always be the case, thus leading to very large delays or a decrease in delivery rates, if the time-to-live of the messages is short. These and other intrinsic relationships in the environment make getting the maximum of minimum values of all the metrics at the same time virtually impossible, and that is why the trade-off is necessary. For this scenario, however, we see that the proposed router ($\alpha = \beta = 0$), outperforms any other router for small TTL values (for TTL values less than around 25 minutes) in delivery ratio (PDR) while having delivery delays (ADD) very close to the ones shown by the other routers, though slightly larger network overhead. In the long run, however (for TTL values larger than 25, $\alpha = 0.65$ is preferred than $\alpha = 0$, having a PDR very close to the maximum value obtained, although more delay than other routers showed. In Network overhead, nonetheless, is where the proposed router excels.

In the following chapter, the final thoughts on this research are provided.

Chapter 6

Summary and conclusions

This chapter closes this research report.

Summary

The integration of vehicular networks in Intelligent Transportation Systems, in the context of Smart Cities, will bring a vast set of novel services in areas such as traffic management, security and safety, e-commerce and entertainment, resulting in a global evolution of cities as we know them. More particularly, there is the paradigm of data networks where vehicles themselves are the nodes in the network, and the system itself provides valuable information to sustain services and applications such as assistance in traffic jams, collision warnings, pre-crash warnings, lane assistance and electronic brake notifications, among others, which in turn will help the future of transportation system safer and greener. Such networks, where the nodes are vehicles, are called Vehicular Delay-Tolerant Networks, because they have to withstand the harsh conditions of vehicular environments, where the high speeds and high mobility of the nodes provokes very frequent disruptions and a non-fixed topology.

Several research opportunities are identified in the literature, such as node design (power consumption, communication range, etc.), buffer management (queuing, buffer allocation, scheduling, etc.) and routing (routing protocols, security in the transmissions, etc.). Routing in Vehicular Delay-Tolerant Networks is a research challenge that requires special attention, since their efficiency will ultimately dictate when these networks become real life implementations. In this paper and following the Design Science in Information Systems Research Framework, we have modeled a solution to the routing problem in VDTN and presented a router based on deep learning which uses an algorithm that leverages the power of neural networks to learn from local and global information to make smart forwarding decisions on the best next hop and best next message.

As discussed in the previous section, the proposed router presents improvements in network overhead and hop count over some popular routers, while maintaining an acceptable delivery rate and delivery delay. For TTL≤ 25, if resources are not a problem, it is recommended to use DLR+ with α=β=0, as it will provide the highest delivery ratio. On the contrary, if network resources are a concern, the proposed router is recommended to use with α=65 and set the message scheduler to β=0, so it has the highest performance despite the resource limitation. All in all, the proposed router with its architecture and routing protocol can be seen as a starting point to further research the routing problem in VDTN using Deep Learning.

Conclusions on the research question

The routing problem in VDTN is a research challenge that still needs to be addressed, as there is not yet an optimal solution that takes into consideration all the metrics in the communication process. Furthermore, as earlier discussed, there has to be a trade-off between some of the metrics that are sought to be optimized to achieve an overall better performance in the VDTN, and the quest for this continues. Ultimately, the corresponding trade-offs depend on the particular application of the network; for instance, in mobile sensor networks the delays may not be an important thing, but the limited resources might be, whereas in VDTN there can be a certain level of flexibility depending on even more specific applications, such as e-commerce transactions versus entertainment applications.

As for the research question, we have discussed in more detail the results and performance that a deep-learning solution like the proposed router has in the routing process in a VDTN. As conclusions, the use of Deep Learning in the proposed router to address the routing problem presents some advantages, as can be inferred from the results section. From the perspective used to address the routing problem in VDTN in this work, Deep Learning techniques can be used to minimize the search for the best conditions for transmissions (namely, the "healthier" nodes and the "best" candidate messages), but this can in turn naturally influence the delays in the delivery process, and therefore the delivery rate, depending on the TTL used. This makes sense, because the router has to wait until it comes into contact with those "healthy" nodes, and this is particularly true in early stages of the simulations, because none of the vehicles has enough history as to improve its conditions, and offer better features to their neighbors so they are considered potential candidates in the transmissions. As time elapses, nonetheless, it is expected that those conditions improve, and the router offers a better performance not only in network overhead and hop count, but also in delivery rate and delays. In the future, a lot of improvements to this router can be done, such as fine-tuning the features (e.g., feature engineering), additional control mechanisms to avoid getting trapped in direct delivery (e.g., spreading metrics similar to the ones used in Spread and Wait) and even the adaptation of different neural network architectures to the router architecture proposed in this work. Finally, very high vehicle densities and very high transmission speeds (e.g., C-V2X and 5G) can significantly impact the performance of the routers. As part of future work, these scenarios are yet to be explored, which could change the balance in the performance of a DL-based solution. Table 6.1 summarizes the impacts that a Deep Learning-based approach can have in the routing protocols.

<table>
<tr><th></th><th>Packet Delivery Ratio</th><th>Average Delivery Delays</th><th>Network Overhead</th><th>Hop Count</th></tr>
<tr><td>Conclusion</td><td>Acceptable, needs improvement</td><td>Needs improvement</td><td>Is better (highly reduced)</td><td>Is better (highly reduced)</td></tr>
<tr><td rowspan="4">Comments</td><td>The PDR is close to the maximum presented by other routers</td><td>ADD is the metric with worst performance</td><td colspan="2">A DL-based solution offers the best performance in this metrics</td></tr>
<tr><td>Improvements could be achieved by adjusting the trade-off between the metrics</td><td colspan="3">As the approach tends to find the fittest node and message to continue the transmission, high delays are possible, especially in the early stages of the interactions in the network</td></tr>
<tr><td colspan="4">Additional control mechanisms can be implemented in the forwarding decisions to give away on network overhead and hop-count in order to achieve improvements in PDR and ADD.</td></tr>
<tr><td colspan="4">Very high vehicle densities and very high transmission speeds (e.g., C-V2X and 5G) can significantly impact the performance of the routers. As part of future work, these scenarios are yet to be explored, which could change the balance in the performance of a DL-based solution.</td></tr>
</table>

Table 6.1. Summary of the impacts on the metrics of a DL-based routing solution.

It is true that there is a lot of work to be done when it comes to the application of Deep Learning to the routing problem in this kind of networks, and some points about this are given in the rest of this section, as well as some directions for future work. All in all, the DLR+ router provides an insight into how deep neural networks can be used to make smarter routers, and this work provides a framework than can serve as a starting point to build more intelligent routing algorithms, and there's still a lot of work ahead in this matter.

Future work and extended applications

In the future, the DLR+ router can be further developed, including the full integration of the neural network to work in real time and automatic online parameter tuning to increase the overall performance. Also, more features of the host and messages can be added to the paradigm, so the router gets an even better understanding of its environment, and additional control mechanisms can be included in the routing protocol to avoid falling into direct delivery, which can heavily affect the delivery rate and delays. Feature engineering is a field in this arena that can be exploited to guarantee that the selection and explosion of features will result in optimal values. Different mechanisms like feature partitioning, grouping, explosion, split can be used to further refine feature selection [15][53].

Furthermore, different neural network architectures can be explored as candidate approaches to solve the optimization problem in determining the best route either by finding the best next hop or finding the best next message (message scheduling) or a combination of both. On the different existing architectures of neural networks, Recurrent Neural Networks (RNN) are a kind of network that is optimized to work with sequential modeling. Also, other types of neural networks, such as the Generative Adversarial Networks (GAN), can help to deal with the lack of enough data to accurately train a deep learning model, which is particularly important towards a physical implementation. If the routing problem is modelled as finding the "cheapest" path in a space of possible paths (sequences), then these two neural networks might be an interesting approach, and the DLR+ architecture can be used as a starting point in the quest in this direction, even though slight changes in the approach (such as the problem framing) might be necessary to adequately adapt the RNN or GAN models for the predictions.

The influence of other variables in the environment is also a research opportunity. For instance, the effect of TTL of the messages was explored in this work, but the influence of node density (i.e., highly congested scenarios) and the presence of relay nodes in the vehicular network (i.e., fixed, road-side units) is yet to be explored. It is a fact that these and other variables heavily influence the dynamics of the vehicular environments, and therefore the dynamics and response of the routers themselves with particular settings. It would be interesting to see the performance of the proposed router in highly dense vehicular networks, because neural networks are prone to perform better as much more data is fed into the systems.

Another aspect that can heavily influence the performance of the routers and the VDTN dynamics is the presence of very high transmission speeds, which can enable high data consuming applications, like infotainment, which are higher than those offered by the IEEE 802.11p standard (up to 27 Mbps), like C-V2X technologies (up to 150 Mbps) or 5G (10 Gbps). These technologies were discussed in more detail in the Introduction chapter, but the influence of such very high speeds in the dynamics was not explored in the experiments and is left as future work.

Also, as the years go by, vehicles are equipped with better processing technology, with complete Operating Systems or ECUs (Engine Control Units) that are mostly AUTOSAR or OSEK-based, for safety and critical applications, or Linux (GENIVI), QNX or Android for ADAS and infotainment [57]. Thus, a research opportunity is also the implementation of such routing algorithms in the latest vehicle processing software, as well as the implementation of the aforementioned algorithms on top of a Software-Defined Network (SDN), which is a recent paradigm that provides a programmable network through decoupling the data, control and application planes [38].

In reality, routing in VDTN is a very vast problem, and a lot of aspects that can affect the performance and dynamics of the networks can be explored. One last area that offers opportunities for research is the analysis of use-cases, namely those that fall into the safety-critical use-cases and those who are in the non-safety use-cases set. While both groups of services and applications are part of the general applications of VDTN, the main difference between them is the minimum latency required for success execution. The first group deals with critical, safety-related applications, and the requires very low latencies from 1 to 10 milliseconds, while the second group can have longer latencies, above 10 and to 100 or more milliseconds. These two scenarios were not specifically addressed in this work, and it is a research opportunity also, since variables such as *message drops* and *delivery delays* have a critical dependence of the minimum allowed latencies in the services that the routers have to support, and the correlation with the rest of the variables and metrics can be affected, resulting in different levels of performance. It's worth noticing, though, that in order to model the first scenarios, the safety use-cases, one must pay attention to the broadcasting that a particular node starts to their immediate nodes, which in turn will replicate the warning messages to neighboring vehicles. Thus, in those cases the focus must be in transmissions that are within a physical radius from the emitter, regardless of whether or not the final delivery is done using direct delivery (e.g., directly to the destination) or with minimum hop count, as to minimize the latencies. In that sense, different more specific scenarios can be tested, such as modeling of crashes at intersections, or any other of the particular use-cases considered for safety-critical services listed in Table 1.2 in chapter 1.

Finally, another field of research is the application of the same router to other scenarios can also be explored, such as in UAV networks or autonomous robot networks. As we approach to a technological era where autonomous and self-adapting systems become a reality, self-organizing networks will be an asset for different kinds of delay-tolerant networks in transportation, logistics, natural disaster response and manufacturing. Table 6.2 summarizes these research opportunities.

Router	Additional routing control mechanisms
	Combination of buffer management and scheduling mechanisms (dropping and abort policies)
Neural Networks	Fine-tuning and additional features
	Feature engineering
	Automated parameter tuning (unsupervised and reinforcement learning)
	Different architectures (RNNs, GANs, etc.)
Vehicular scenario	Influence node density (e.g., highly congested scenarios)
	Effect of relay nodes (presence of road-side units)
	Effect of mobility patterns (i.e., performance under different models)
	Influence of high transmission speeds (e.g., C-V2X and 5G)
	Particular scenarios (safety-related scenarios, specific patterns (bus, working day, etc.))
Other applications	UAV networks
	Cooperative autonomous robots
	General DTNs

Table 6.2. Summary of research opportunities derived from this work.

As can be seen, there's still a lot to be explored in this field and the use of Deep Learning is a very promising approach to these research opportunities. All in all, the paradigm of deep learning itself is yet to unleash its full potential, but the most recent advances in the matter could be used to address any optimization problem, like the routing problem in VDTN, in unimagined and promising ways.

References

[1] Ahmed, S., & Kanhere, S. S. (2011). Message Dissemination in Vehicular Networks.

[2] Asandei, A., Dobre, C., & Johnson, P. (2012). An Analysis of Techniques for Opportunistic Networking.

[3] Asgari M., Jumari K. & Ismail M. (2011). Analysis of Routing Protocols in Vehicular Ad Hoc Network Applications. In: Software Engineering and Computer Systems. Communications in Computer and Information Science, 2011. Volume 181. Springer, Berlin, Heidelberg.

[4] Baguerlou, H. & Ghaffari, A. (2018). A routing protocol for vehicular ad hoc networks using simulated annealing algorithm and neural networks.

[5] Balakrishna, C. (2012). Enabling Technologies for Smart City Services and Applications. Sixth International Conference on Next Generation Mobile Applications, Services and Technologies.

[6] Balázs Csanád Csáji. (2001). Approximation with Artificial Neural Networks; Faculty of Sciences; Eötvös Loránd University, Hungary, 2001.

[7] Benamar, N., Benamar, M., & Bonnin, J. M. (2012). Routing Protocols for DTN in Vehicular Environment.

[8] Blanes, R., Paton, R. & Docherty, I. (2015). Public Value of Intelligent Transportation System. 48th Hawaii International Conference on System Sciences, 2015.

[9] C. Lee, K., & Gerla, M. (2010). Opportunistic Vehicular Routing. 2010 European Wireless Conference.

[10] Canziani, A., Paszke, A. & Culurciello, E. (2017). An Analysis of Deep Neural Network Models for Practical Applications. In: arXiv preprint at arXiv:1605.07678, 2017.

[11] Cao, Y., & Sun, Z. (2013). Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges. IEEE Communications Surveys & Tutorials Vol. 15, No. 2, Second Quarter 2013.

[12] Constandoiu, A. & Leba, M. (2019). Convergence of V2X Communication Systems and Next Generation Networks.

[13] Dimitrakopoulos, G. (2011). Intelligent Transportation Systems based on Internet-Connected Vehicles: Fundamental Research Areas and Challenges. 11th International Conference on ITS Telecommunications.

[14] Dua A., Kumar N. & BawaA S. (2014). Systematic review on routing protocols for Vehicular Ad Hoc Networks. Vehicular Communications, 2014, Volume 1, Issue 1, pp 33-52.

[15] Emre, R. (2019). Fundamental Techniques of Feature Engineering for Machine Learning. Available online at https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114

[16] Fengxiao T. , Bomin, M., Fadlullah, Z., Kato, N., Akashi, O., Inoue, T. & Mizu, K. (2018). On Removing Routing Protocol from Future Wireless Networks: A Real-time Deep Learning Approach for Intelligent Traffic Control. IEEE Wireless Communications, 2018, Vol. 25, no. 1.

[17] Filippi, A., et al. (2017). IEEE802.11p ahead of LTE-V2V for safety applications. Autotalks, NXP.

[18] Filippi, A., et al.Why 802.11p Beats LTE and 5G for V2X. Available online at http://www.eenewsautomotive.com/design-center/why-80211p-beats-lte-and-5g-v2x.

[19] Gaurang, N., Biplav, Ch. and Jung-Min, P. (2020) IEEE 802.11bd & 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications.

[20] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press, 2016.

[21] Goyal, E. M., & Chaudhary, E. M. (2013). Ensuring Privacy in Opportunistic Network. IOSR Journal of Computer Engineering (IOSR-JCE).

[22] Greede, A., M. Allen, S., & M. Whitake, R. (2012). RDD: Repository-based Data Dissemination Protocol for Opportunistic Networks. 2012 International Conference on Selected Topics in Mobile and Wireless Networking.

[23] Gupta, A., Bansal, A., Naryani, D. & Sharma, D.K. (2017). CRPO: Cognitive Routing Protocol for Opportunistic Networks. Proceedings of the International Conference on High Performance Compilation, Computing and Communications; vol. 1, pp. 121-125, March 2017.

[24] Hanif Halim, A., & Ismail, I. (2014). Bio-Inspired Optimization Method: A Review. NNGT International Journal of Artificial Intelligence.

[25] Hao, K. (2018). What is Machine Learning. MIT Technology Review. Available online at https://www.technologyreview.com/2018/11/17/103781/what-is-machine-learning-we-drew-you-another-flowchart/

[26] Hassan Ahmed, S., Kang, H., & Kim, D. (2015). Vehicular Delay Tolerant Network (VDTN): Routing Perspectives.

[27] Heaton, J. (2018). Introduction to Neural Networks for Java. Heaton Research, Inc., 2008.

[28] Heena and Garg, V. (2015). Performance Analysis of Various Movement Models in DTN. National Conference on Advances in Engineering, Technology & Management. IOSR Journal of Computer Engineering (IOSR-JCE).

[29] Hernández, R., Cárdenas, C., & Muñoz, D. (2014). On the Importance of Delay-Tolerant Networks for Intelligent Transportation Systems in Smart Cities. Mexican International Conference on Computer Science (ENC 2014), Nov 3-5, 2014; Oaxaca, México.

[30] Hernández, R., Cárdenas, C., & Muñoz, D. (2015). Epidemic Routing in VDTN: the use of Heterogeneous Conditions to Increase Packet Delivery Ratio. IEEE International Smart Cities Conference, 2015. [CrossRef]

[31] Hevner, A., March, S., Park, J. & Ram, S. (2004). Design Science in Information Systems Research. MIS Quarterly, (28: 1).

[32] Hyndman, R. (2019). How to choose the number of hidden layers and nodes in a feedforward neural network? Retrieved on August, 2019, from https://stats.stackexchange.com/q/181

[33] IEEE Standards Commitee. IEEE Vehicular Technology Society. (2019). IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture.

[34] Ikeda, M., Honda, T., Ishikawa, S., & Barolli, L. (2014). Performance Comparison of DTN Routing Protocols in Vehicular-DTN Environment. 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications.

[35] Ioan Ciobanu, R., Dobre, C., & Cristea, V. (2013). Reducing Congestion for Routing Algorithms in Opportunistic Networks with Socially-Aware Node Behavior Prediction. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications.

[36] Isento, J. N., J. P. C Rodrigues, J., Dias, J. A., Paula, M. C., & Vinel, A. (2013). Vehicular Delay-Tolerant Networks - A Novel Solution for Vehicular Communications. IEEE IntellIgent Transportation Systems Magazine.

[37] J. Khabbaz, M., M. Assi, C., & F. Fawaz, W. (2012). Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges.

[38] Jingjing, Z., Di, C., Weiming, W., Rong, J., & Xiaochun, W. (2014). The Deployment of Routing Protocols in Distributed Control Plane of SDN. The Scientific World Journal, 2014, 1–8.

[39] Keränen, A., Ott, J. & Kärkkäinen, T. (2009). The ONE Simulator for DTN Protocol Evaluation. 2nd International Conference on Simulation Tools and Techniques; SIMUTools 2009, Rome, Italy. [CrossRef]

[40] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2014.

[41] Lindgren A., Doria A. & Schelén O. (2004). Probabilistic Routing in Intermittently Connected Networks. In: Service Assurance with Partial and Intermittent Resources. SAPIR. Lecture Notes in Computer Science, 2004, Volume 3126. Springer, Berlin, Heidelberg [CrossRef]

[42] Lindgren, A., & Hui, P. (2009). The Quest for a Killer for Opportunistic and Delay Tolerant Networks. CHANTS '09.

[43] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. & Alsaadi, F.E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 2017, Vol 234, pp. 11-26. [CrossRef]

[44] Mardacany, E. (2014). Smart Cities Characteristics: Importance of Built Environment Components.

[45] Meisel, M., Pappas, V., & Zhang, L. (2009). A Taxonomy of Biologically Inspired Research in Computer Networking. Computer Networks.

[46] N. G. J. Soares, V., Farahmand, F., & J. P. C. Rodrigues, J. (2009). A Layered Architecture for Vehicular Delay-Tolerant Networks.

[47] N. G. J. Soares, V., J. P. C. Rodrigues, J., & Dias, J. A. (2012). Performance Analysis of Routing Protocols for Vehicular Delay-Tolerant Networks.

[48] Neumann, F., & Witt, C. (2010). Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity.

[49] Niu, J., Guo, J., Cai, Q., Sadeh, N., & Guo, S. (2011). Predict and Spread: An Efficient Routing Algorithm for Opportunistic Networking. IEEE WCNC 2011 - Network.

[50] Niyato, D., Wang, P., & Ming Teo, J. C. (2009). Performance Analysis of the Vehicular Delay Tolerant Network. WCNC 2009.

[51] Palma, A., Rogério Pereira, P., & Casaca, A. (2012). Multicast Routing Protocol for Vehicular Delay-Tolerant Networks. 2012 Fifth International Workshop on Selected Topics in Mobile and Wireless Computing.

[52] Patil, P. (2012). Towards Reliable Communication In Intelligent Transportation Systems. 31st International Symposium on Reliable Distributed Systems.

[53] Patrick, H. (2019). The importance of Feature Engineering and Selection. Available online at https://www.rittmanmead.com/blog/2019/02/the-importance-of-feature-engineering-and-selection/

[54] Pelusi, L., Passarella, A., & Conti, M. (2006). Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks.

[55] Pereira, R., Casaca, A., Rodrigues, J., Soares, V., Triay, J., & Cervelló-Pastor, C. (2011). From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks. IEEE Communications Surveys & Tutorials.

[56] Qi, L. (2008). Research on Intelligent Transportation System Technologies and Applications. 2008 Workshop on Power Electronics and Intelligent Transportation System.

[57] Quora. What Operating Systems are used in cars? Available online at https://www.quora.com/What-operating-systems-are-used-in-cars

[58] R. Penurkar, M., & A. Deshpande, U. (2014). CONHIS: Contact History-based Routing Algorithm for a Vehicular Delay Tolerant Network. 2014 Annual IEEE India Conference (INDICON).

[59] Robinson, Rick. (2012). The New Architecture of Smart Cities. Available online at https://theurbantechnologist.com/2012/09/26/the-new-architecture-of-smart-cities/

[60] S., B., & Sathya, S. S. (2012). A Survey of Bio inspired Optimization Algorithms. International Journal of Soft Computing and Engineering (IJSCE).

[61] Saha, B. K., Misra, S. & Pal, S. (2017). SeeR: Simulated Annealing-based Routing in Opportunistic Mobile Networks. IEEE Transactions on Mobile Computing, 2017. Vol. 16, no. 10, pp. 2876-2888.

[62] Saha, S., Roy, U., & Sinha, D. (2015). Neural Network based modified AODV Routing Protocol in VANET.

[63] Segundo, F. R., Silveira, E. & Silva, JM. F. (2016). A DTN Routing Strategy Based on Neural Networks for Urban Bus Transportation System.

[64] Shahzamal, M., Pervez, M. F., Zaman, M. A. U. & Hossain, M. D. (2014). Mobility Models for Delay Tolerant Networks: A Survey. International Journal of Wireless and Mobile Networks, 2014, Vol. 6, No. 4. [CrossRef]

[65] Sharma, D. K., Dhurandher, S. K., Woungang, I., Srivastava, R. K., Mohananey, A., & Rodrigues, J. J. P. C. (2018). A Machine Learning-Based Protocol for Efficient Routing in Opportunistic Networks. IEEE Systems Journal, 2018, Vol. 2, no. 3, pp. 2207-2213. [CrossRef]

[66] Sharma, D.K., Aayush.,Sharma, A. & Kumar, J. (2017). KNNR: K-nearest neighbor classification-based routing protocol for opportunistic networks. 10th International Conference on Contemporary Computing (IC3), 10-12 August 2017, Noida, India.

[67] Soares, V. N. G; Rodrigues, J., and Farahmand, Farid. (2009). Impact analysis of the shortest path movement model on routing strategies for VDTNs in a rural region.

[68] Soares, V. N., Farahmand, F., & J. P. C. Rodrigues, J. (2009). A Layered Architecture for Vehicular Delay-Tolerant Networks.

[69] Spyropoulos, T., Psounis, K. & Raghavendra, C.S. (2005). Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In: ACM SIGCOMM'05 Workshops, 2005. [CrossRef]

[70] Su, K., Li, J., & Fu, H. (2011). Smart City and the Applications.

[71] Tewolde, S. G. (2012). Sensor and Network Technology for Intelligent Transportation Systems.

[72] Tornell, S., Calafate, C., Cano, J-C. & Manzoni, P. (2014). DTN Protocols for Vehicular Networks: An Application Oriented Overview. IEEE Communications Surveys & Tutorials, Vol. 17, No. 2. 2014.

[73] United Nations, Department of Economic and Social Affairs, Population Division (2018). The World's Cities in 2018—Data Booklet (ST/ESA/ SER.A/417).

[74] Usman, M., et al. (2020). A Business and Legislative Perspective of V2X and Mobility Applications in 5G Networks. IEEE Access.

[75] Vahdat, A. & Becker, D. (2000). Epidemic Routing for Partially-Connected Ad Hoc Networks. In: Handbook of Systemic Autoimmune Diseases. Technical Report, 2000.

[76] Wang, Y. Duana, X., Tiana, D., Lu, G. & Yu, H. (2013). Throughput and Delay Limits of 802.11p and Its Influence on Highway Capacity. In: Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013), Special Issue. Procedia - Social and Behavioral Sciences; vol 96, pp. 2096-2104, Nov. 2013. [CrossRef]

[77] Xie, X., Zhang, Y., Dai, C., & Song, M. (2010). Heterogeneous Context-aware Routing Protocol for Opportunistic Network.

[78] Yan, X., Zhang, H., & Wu, C. (2012). Research and Development of Intelligent Transportation Systems. 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science.

[79] Burgess, J. et al. (2006). MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. IEEE INFOCOM 2006 Proceedings.

[80] Singh, A. V., Juyal, V., & Saggar, R. (2016). Trust based Intelligent Routing Algorithm for Delay Tolerant Network using Artificial Neural Network.

[81] Paul, A., Chilamkurti, N., Daniel, A., & Rho, S. (2017). Evaluation of vehicular network models. Intelligent Vehicular Networks and Communications, 77–112.

[82] NG, A. Machine Learning course. Coursera. Available online at https://www.coursera.org/learn/machine-learning/

[83] Karimi, R. et al. (2011). DTN Routing Protocols for VANETs: Issues and Approaches. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011.

[84] Kamran, K. et al. (2015). A Comparative Survey on Vehicular Ad-hoc Network (VANET) Routing Protocol using Heuristic and Optimistic Techniques.

[85] Benamar, N. et al. (2014). Routing Protocols in Vehicular Delay-Tolerant Networks: A Comprehensive Survey. Computer Comunications.

[86] Tripp-Barba, C. et al. (2019). Survey on Routing Protocols for Vehicular Ad Hoc Networks Based on Multimetrics. Electronics, 2019.

[87] Amin, M. et al. (2016). A survey of Routing Protocols for Delay-Tolerant Networks in VANETs.

Appendix

Appendix 1 – Research paper 1. Hernández-Jiménez, R., Cárdenas-Pérez, C. & Muñoz-Rodríguez, D. (2019) *Modeling and Solution of the Routing Problem in Vehicular Delay-Tolerant Networks: a Dual, Deep Learning Perspective.* Applied Sciences. 2019, 9(23), 5254; DOI: 10.3390/app9235254 (**Impact Factor 2.217**).

Appendix 2 – Research paper 2. Hernández-Jiménez, R., Cárdenas-Pérez, C. & Muñoz-Rodríguez, D. (2019). *Towards the Optimal Solution for the Routing Problem in Vehicular Delay Tolerant Networks: A Deep Learning Approach.* IEEE Latin America Transactions, Special Issue on Deep Learning. DOI: 10.1109/TLA.2019.9011548 (**Impact Factor 0.0804**).

Appendix 3 – Research paper 3. Hernández, R. Cárdenas, C. & Muñoz, D. (2017). *Game theory applied to transportation systems in Smart Cities: analysis of evolutionary stable strategies in a generic car-pooling system*. Virtual Concept Workshop 2016. Intelligent Transport Systems and Data Science. Guadalajara, Mexico. May 30-31, 2016. DOI: 10.1007/s12008-017-0373-4

Appendix 4 – Research paper 4. Hernández, R., Cárdenas, C. & Muñoz, D. (2015). *Epidemic Routing in Vehicular Delay-Tolerant Networks: the use of Heterogeneous Conditions to Increase Packet Delivery Ratio.* IEEE International Smart Cities Conference 2015. **DOI:** 10.1109/ISC2.2015.7366199

Appendix 5 – Research paper 5. Hernández, R., Cárdenas, C. & Muñoz, D. (2014). *On the Importance of Delay-Tolerant Networks for Intelligent Transportation Systems in Smart Cities.* Mexican International Conference on Computer Science (ENC 2014), Nov 3-5, 2014; Oaxaca, México.

Appendix 6 – Performance of DLR+ on Packet Delivery Ratio with different threshold values (α, β)

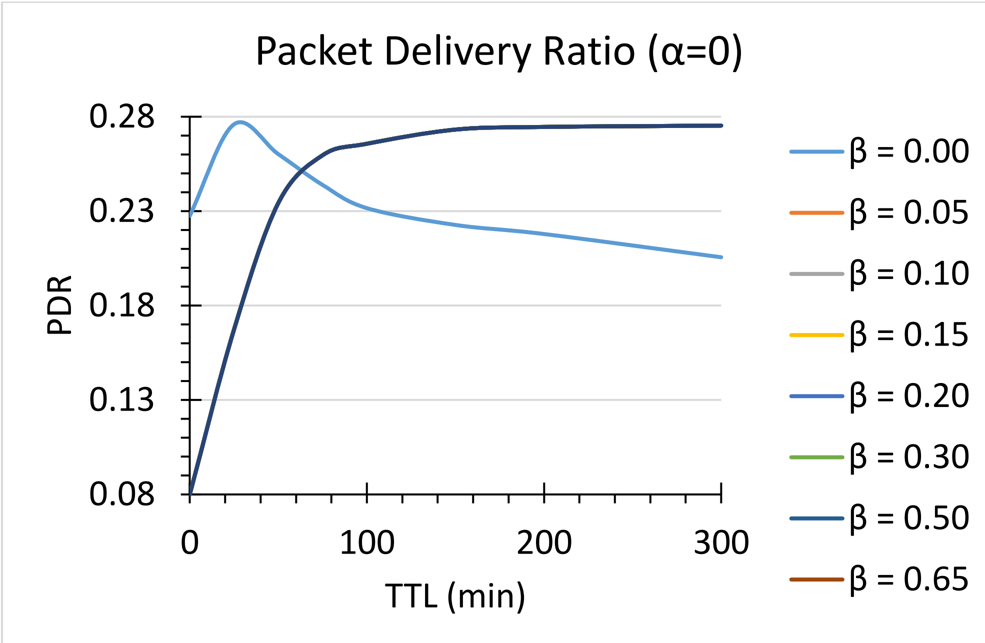


Figure A-6.1. PDR with $\alpha = 0$.

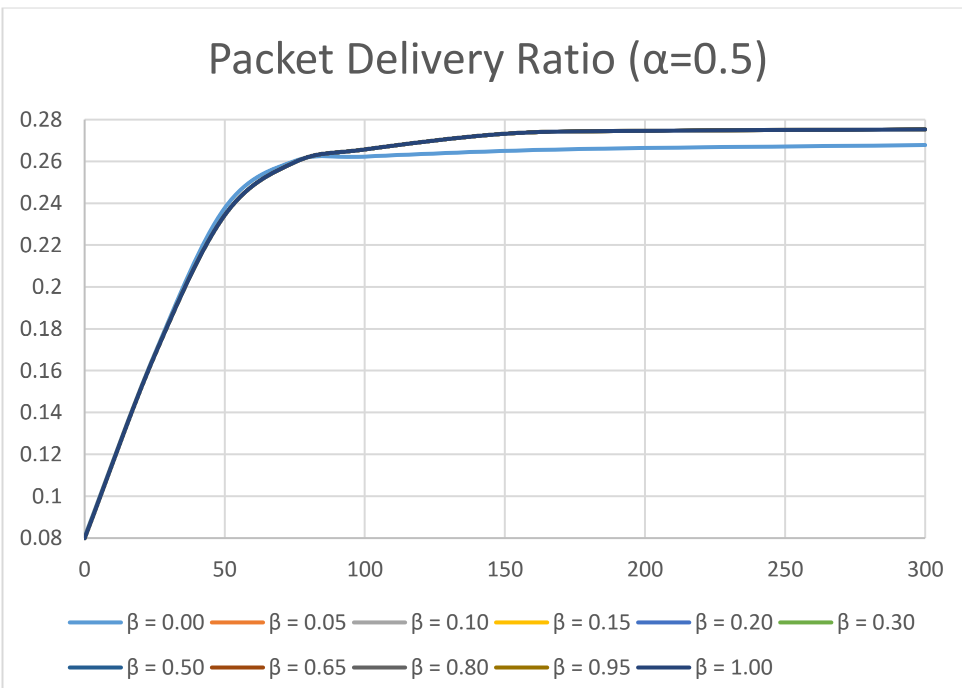


Figure A-6.2. PDR with $\alpha = 0.5$.

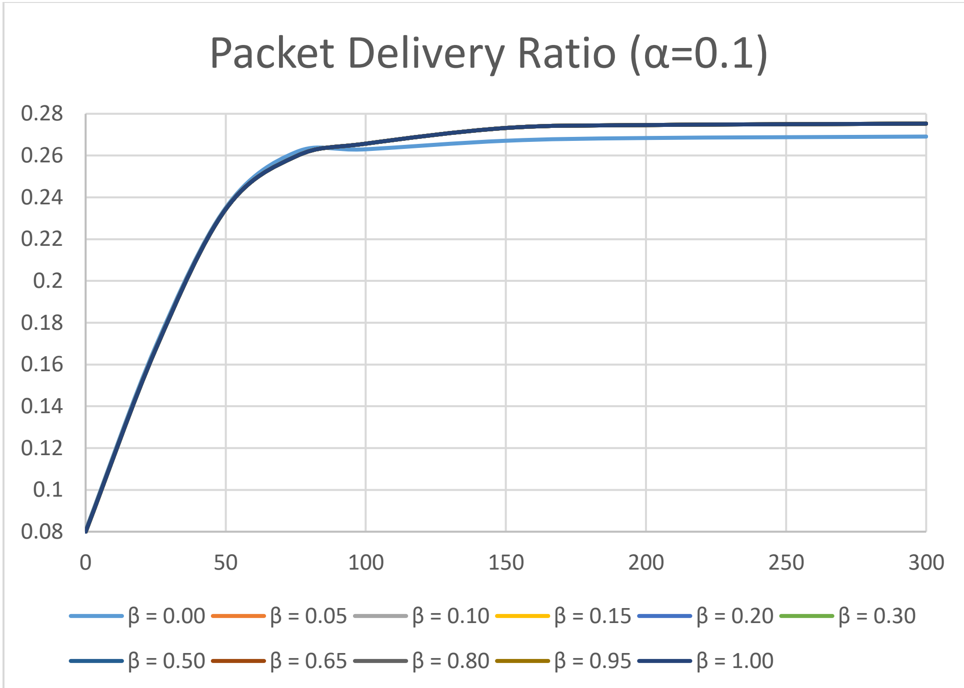


Figure A-6.3. PDR with $\alpha = 0.1$.

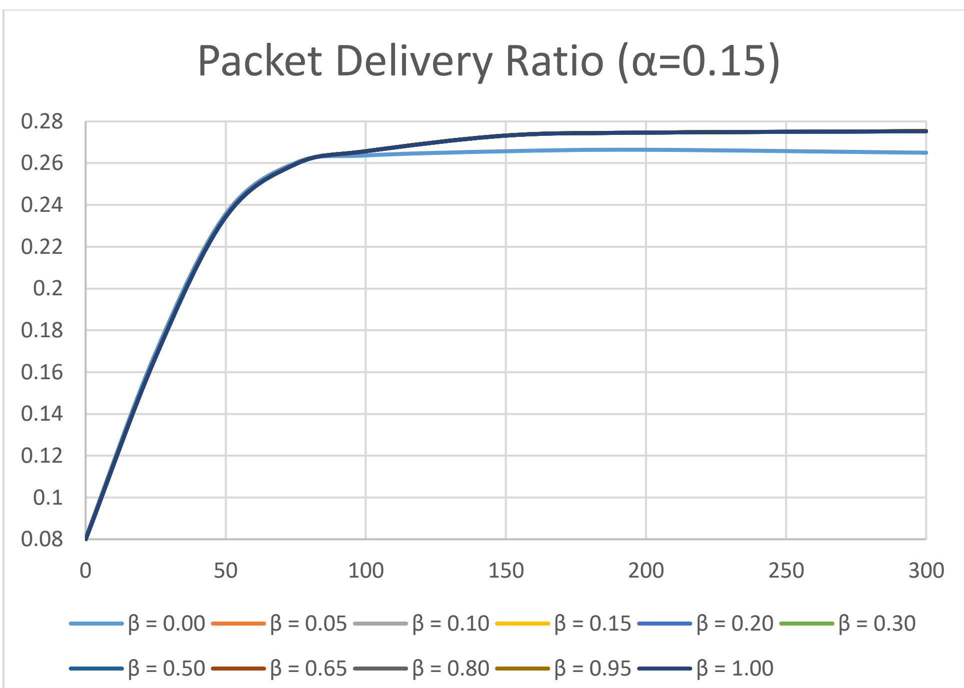


Figure A-6.4. PDR with $\alpha = 0.15$.

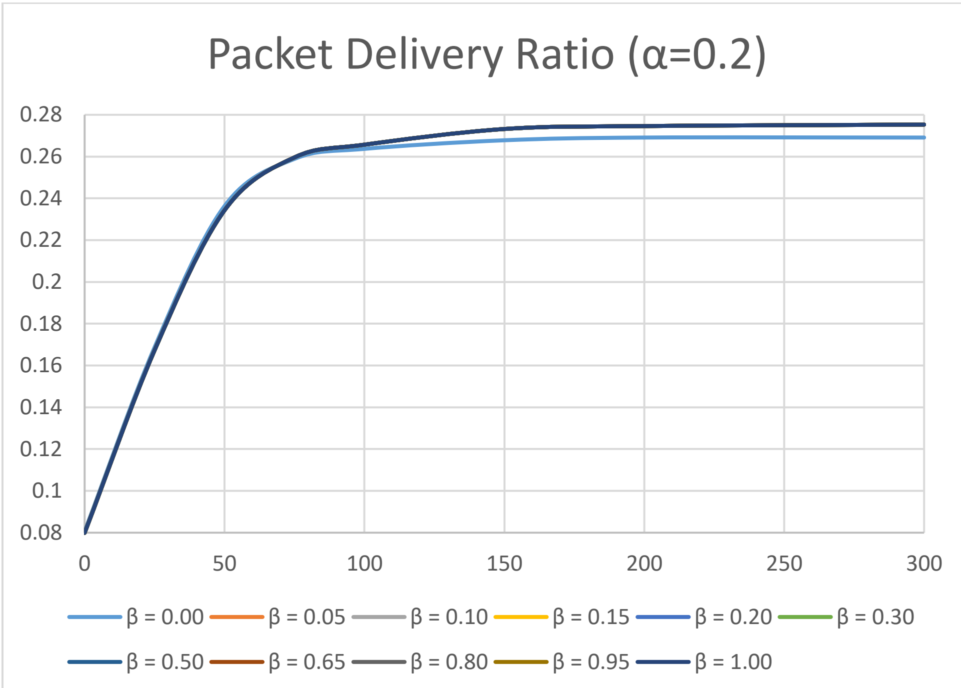


Figure A-6.5. PDR with $\alpha = 0.2$.

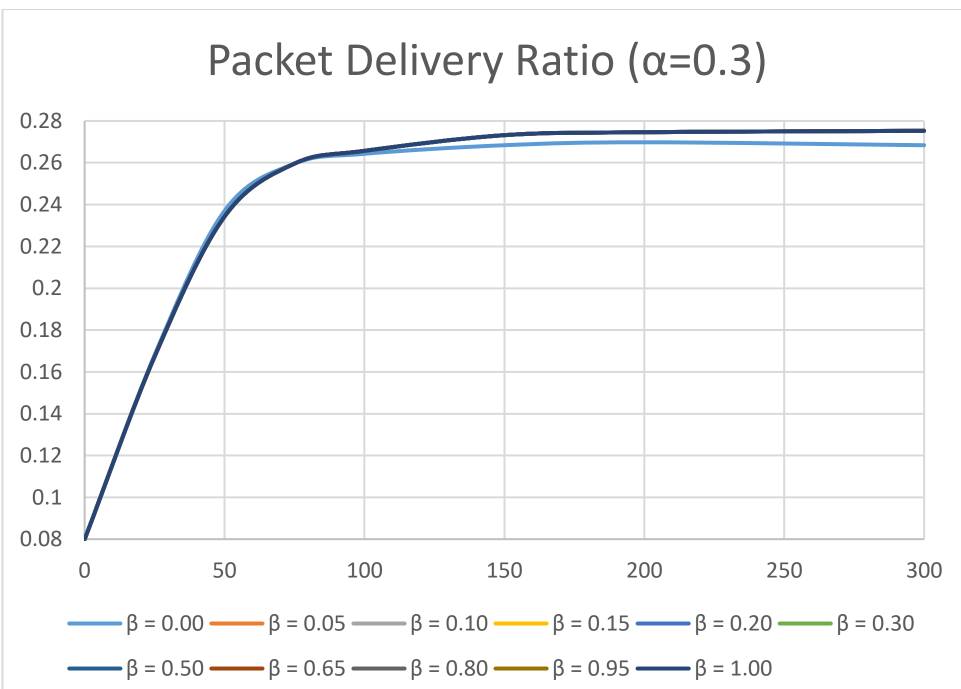


Figure A-6.6. PDR with $\alpha = 03$.

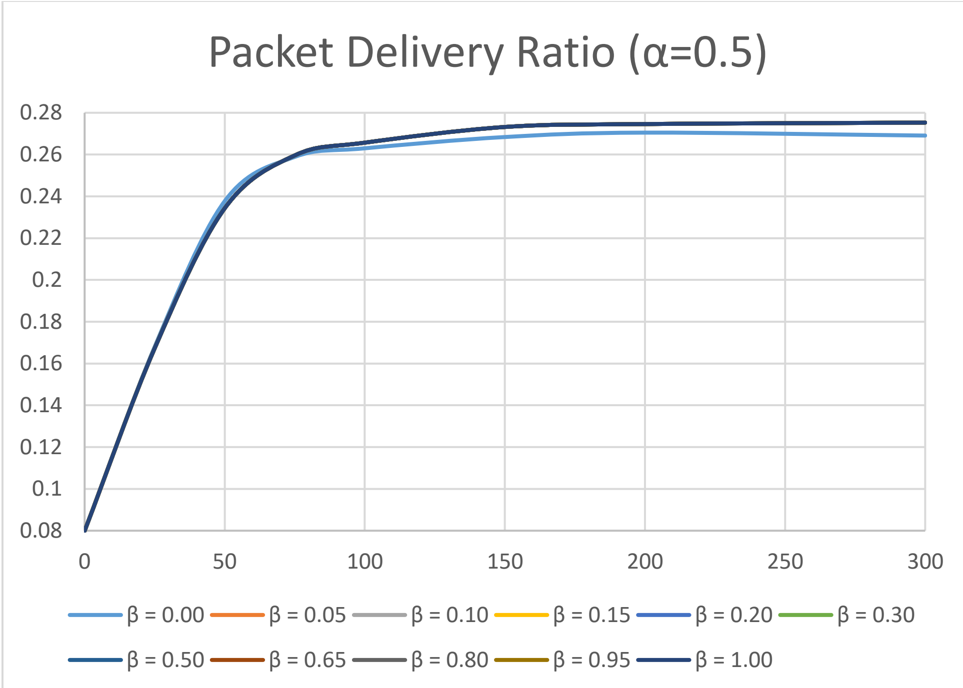


Figure A-6.7. PDR with $\alpha = 0.5$.

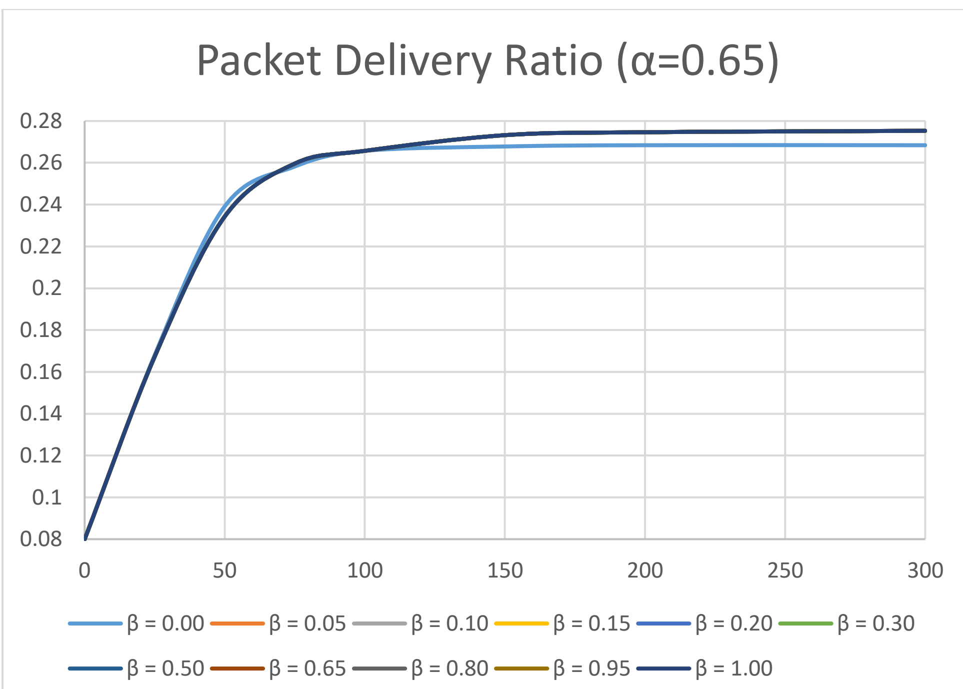


Figure A-6.8. PDR with $\alpha = 0.65$.

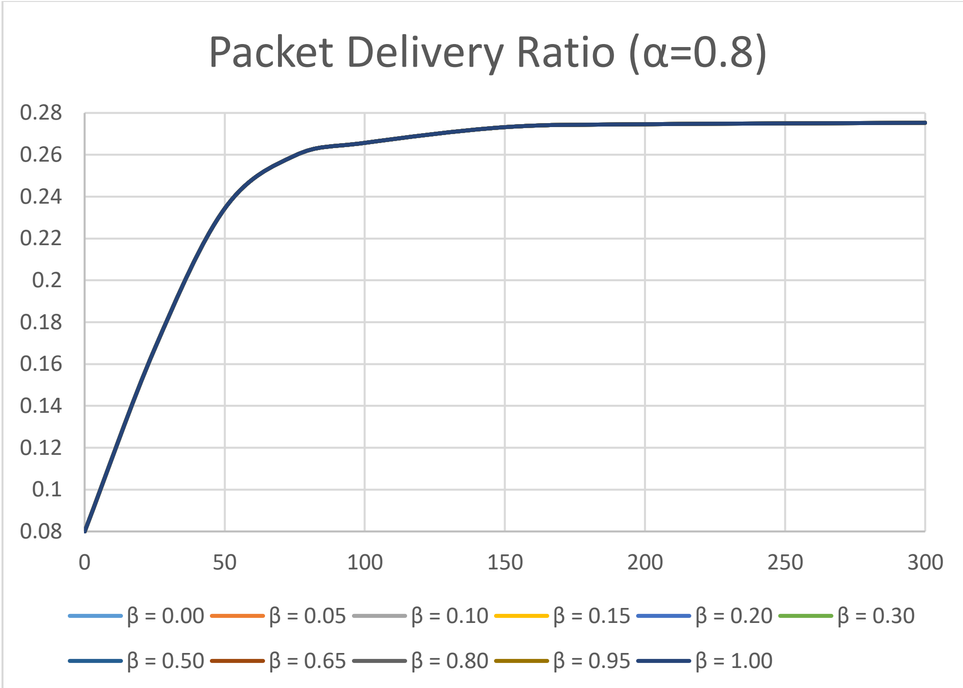


Figure A-6.9. PDR with $\alpha = 0.8$.

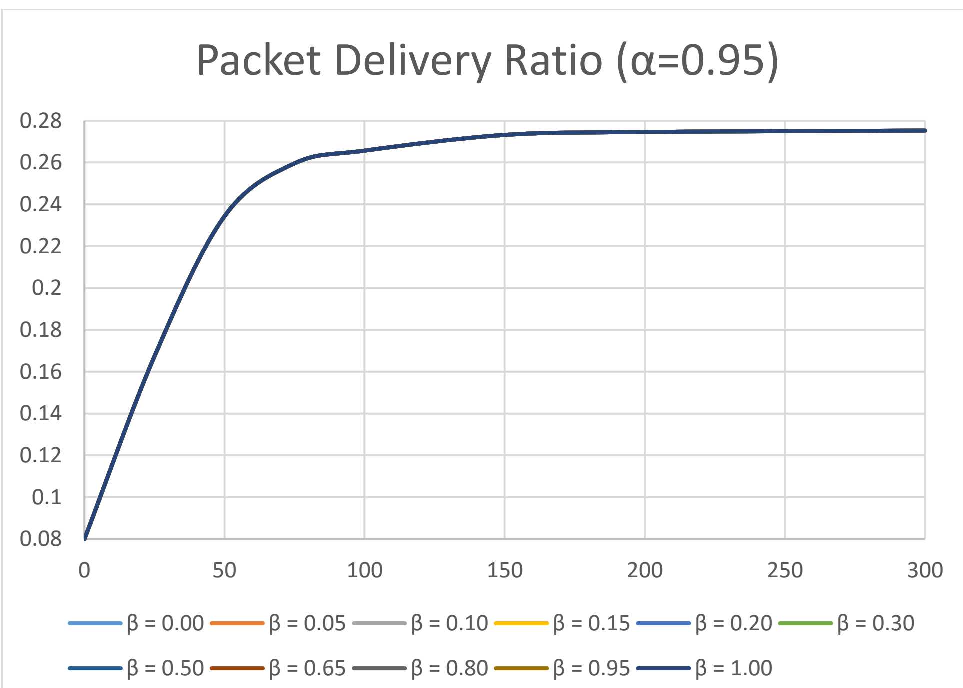


Figure A-6.10. PDR with $\alpha = 0.95$.

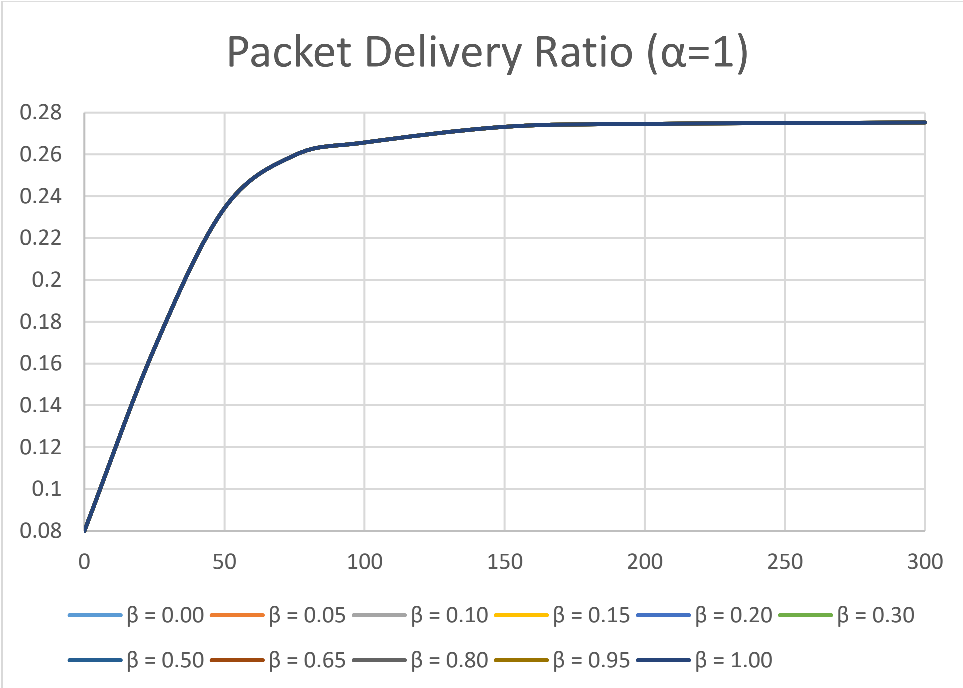


Figure A-6.11. PDR with $\alpha = 1$.

Appendix 7 – Performance of DLR+ on Average Delivery Delay with different threshold values (α, β)

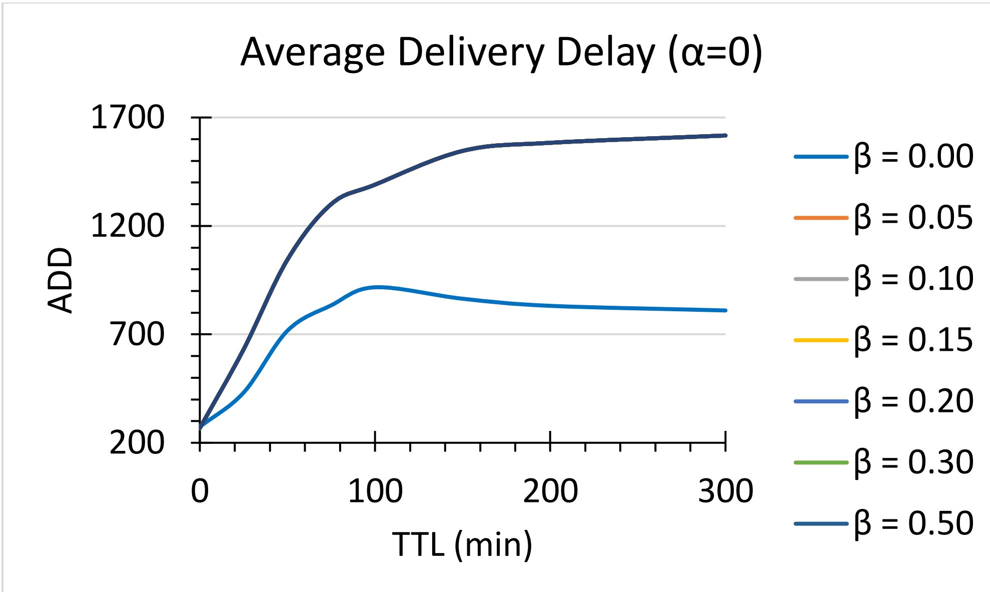


Figure A-7.1. ADD with $\alpha = 0$.

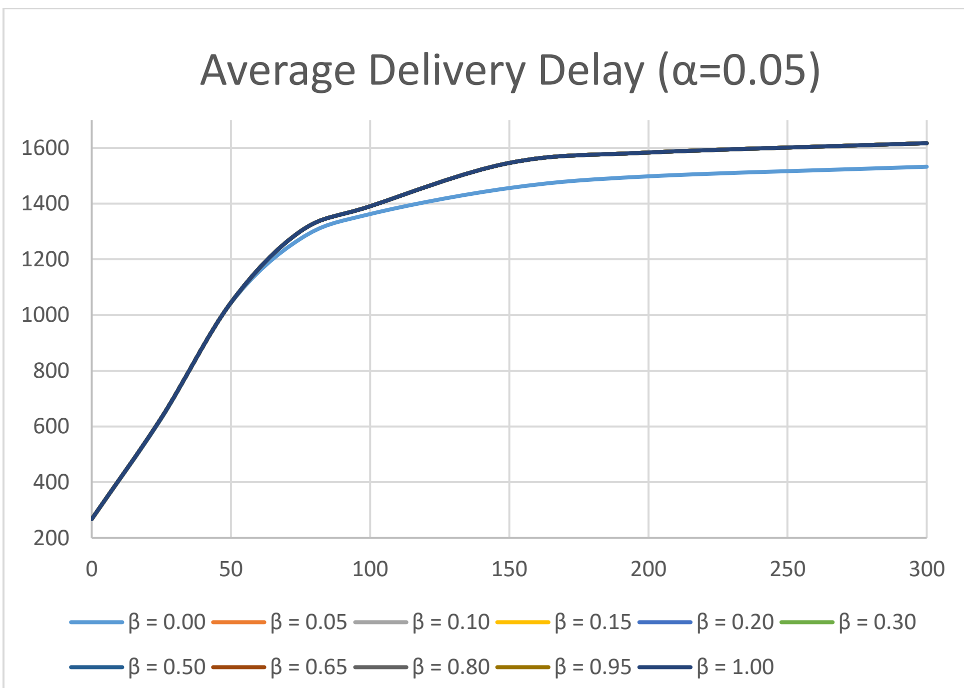


Figure A-7.2. ADD with $\alpha = 0.05$.

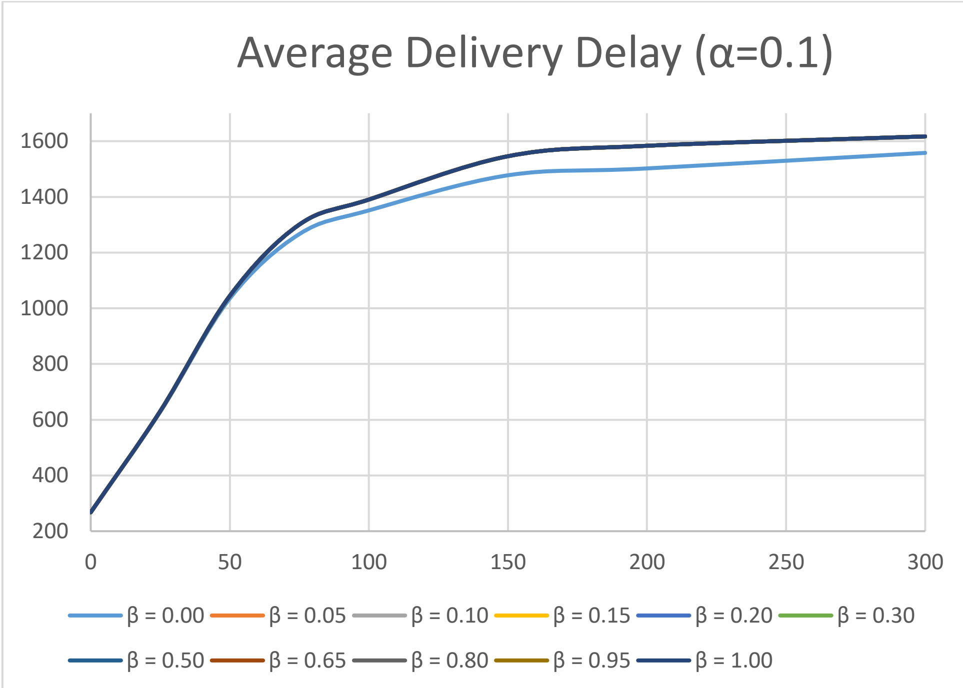


Figure A-7.3. ADD with $\alpha = 0.1$.

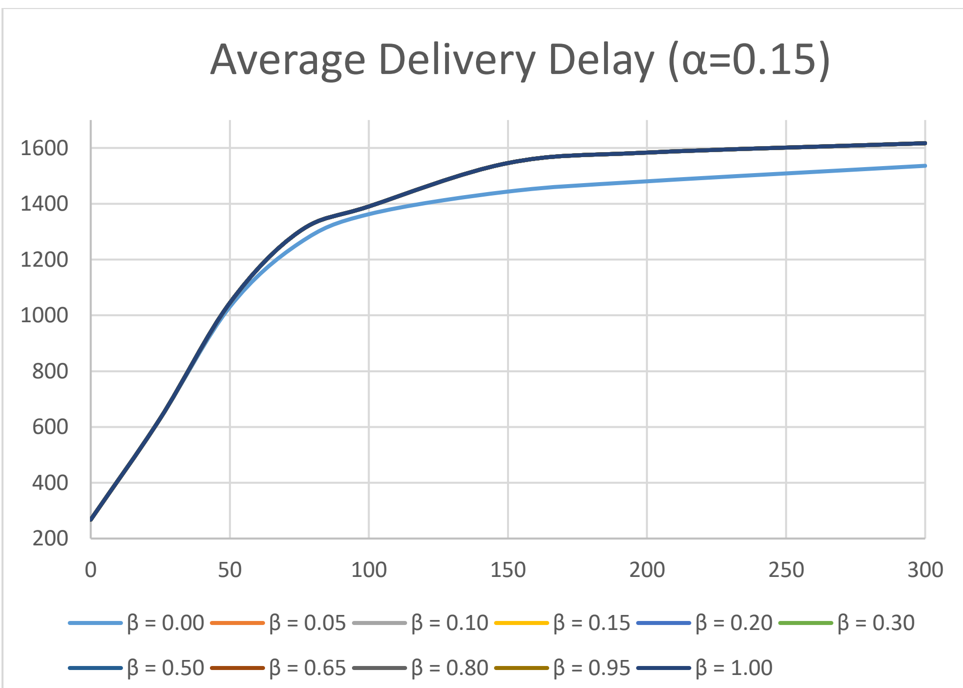


Figure A-7.4. ADD with $\alpha = 0.15$.

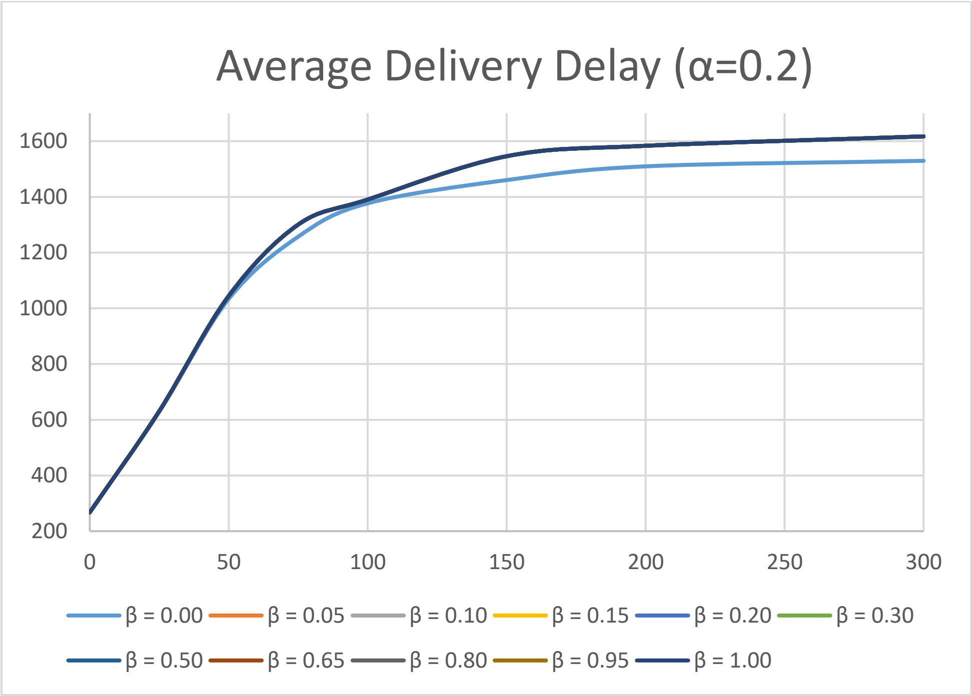


Figure A-7.5. ADD with $\alpha = 0.2$.

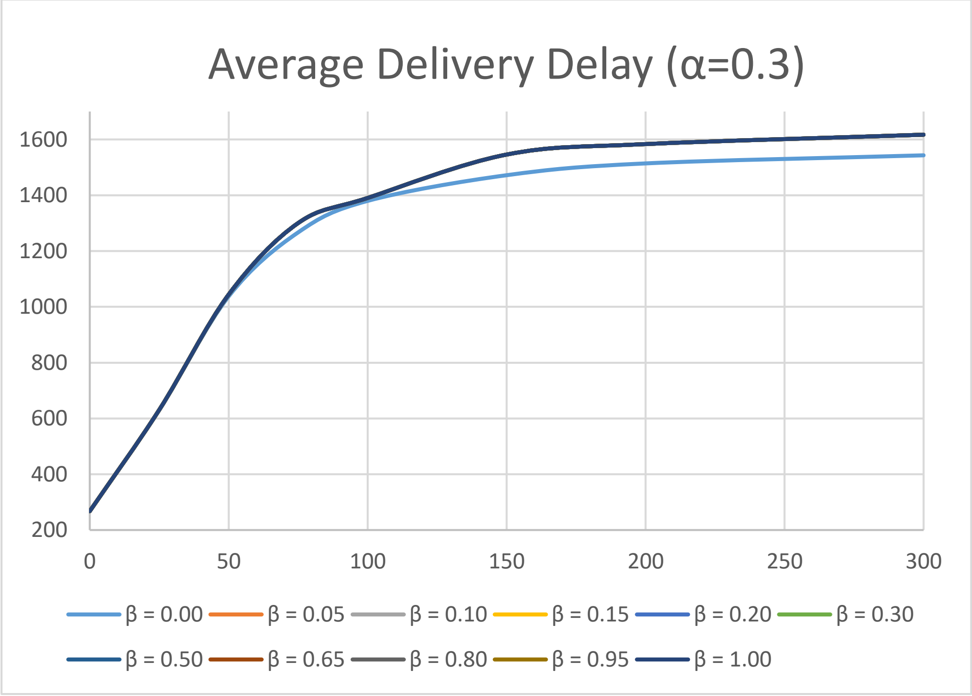


Figure A-7.6. ADD with $\alpha = 0.3$.

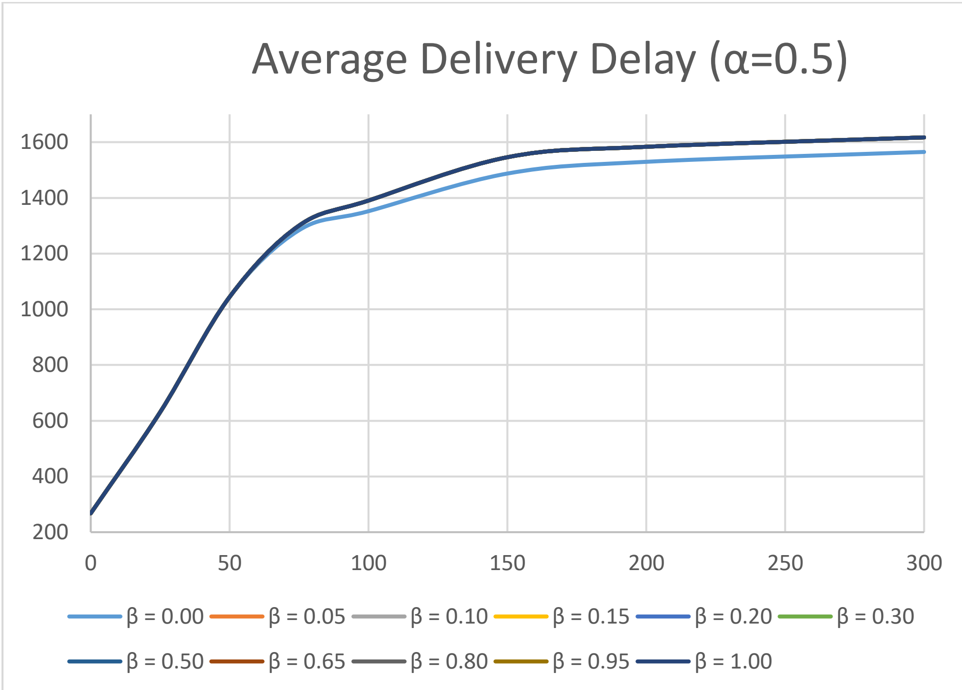


Figure A-7.7. ADD with $\alpha = 0.5$.

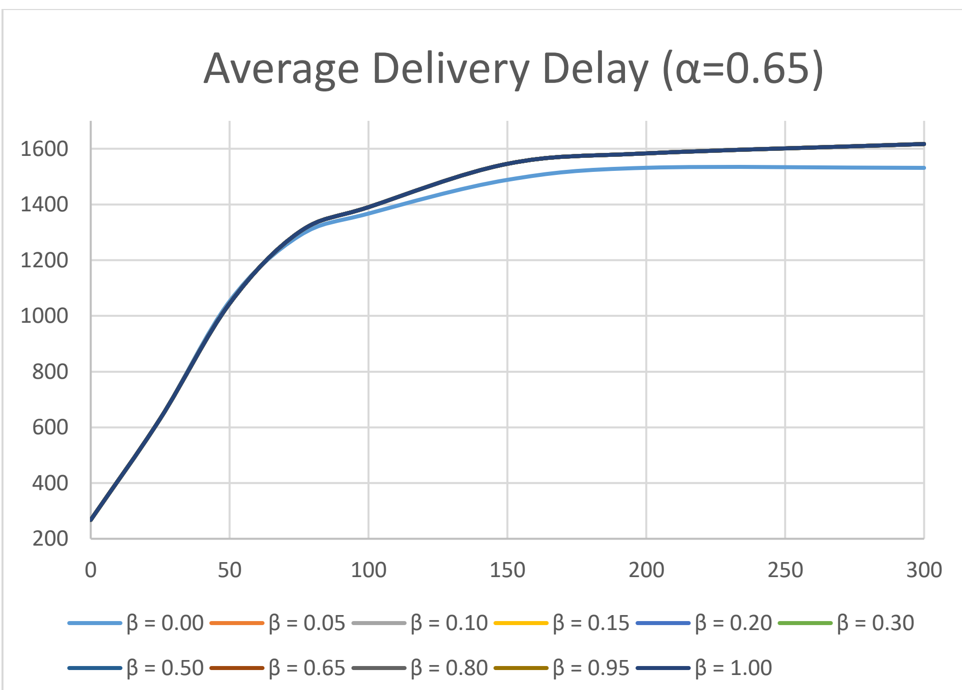


Figure A-7.8. ADD with $\alpha = 0.65$.

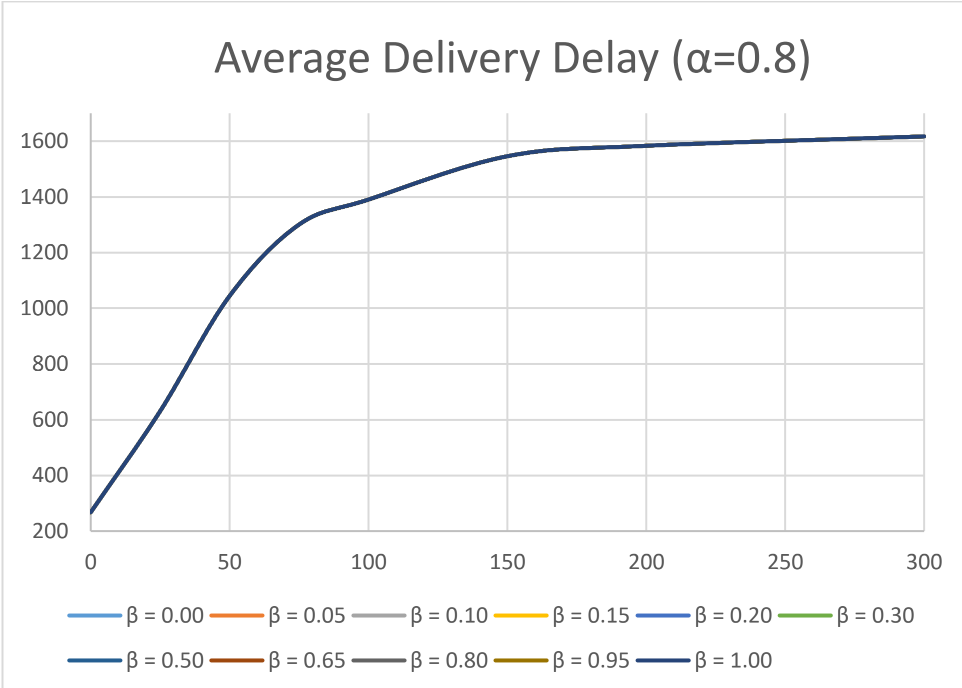


Figure A-7.9. ADD with $\alpha = 0.8$.

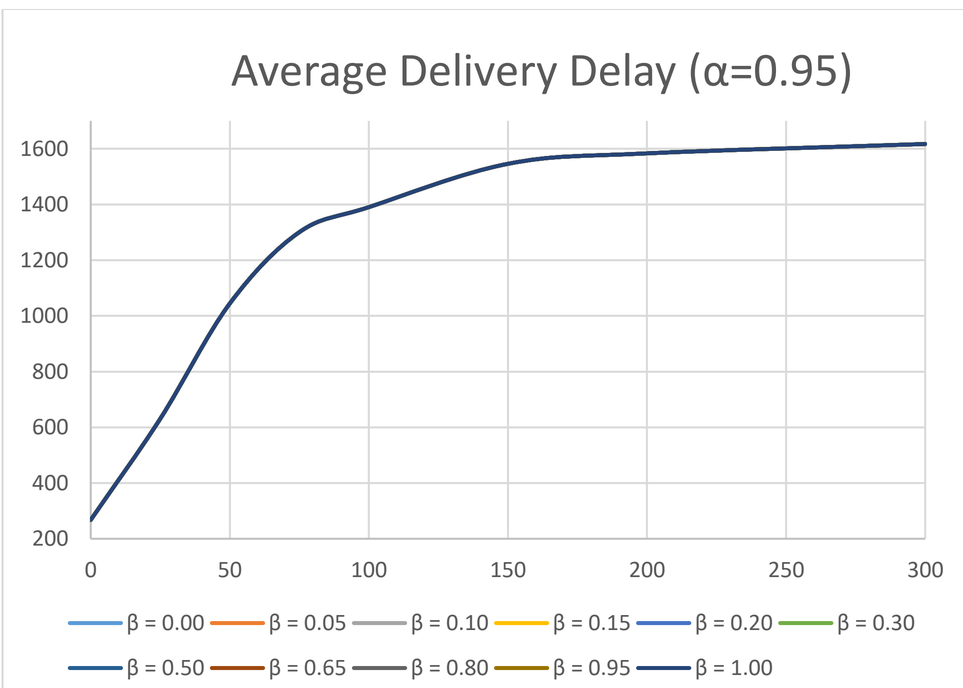


Figure A-7.10. ADD with $\alpha = 0.95$.

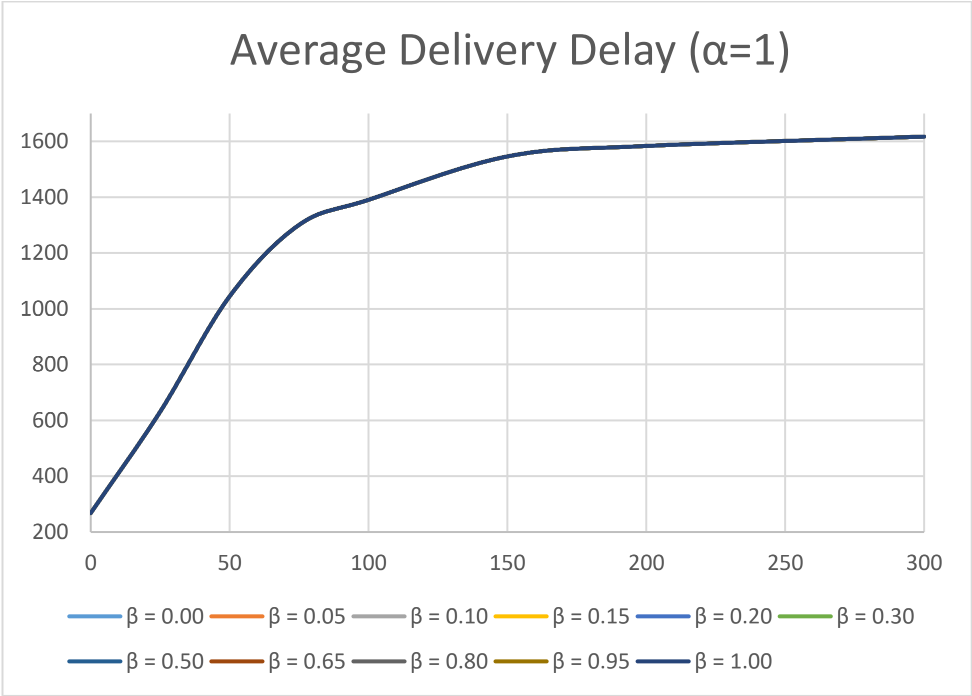


Figure A-7.11. ADD with $\alpha = 1$.

Appendix 8 – Performance of DLR+ on Network Overhead with different threshold values (α, β)

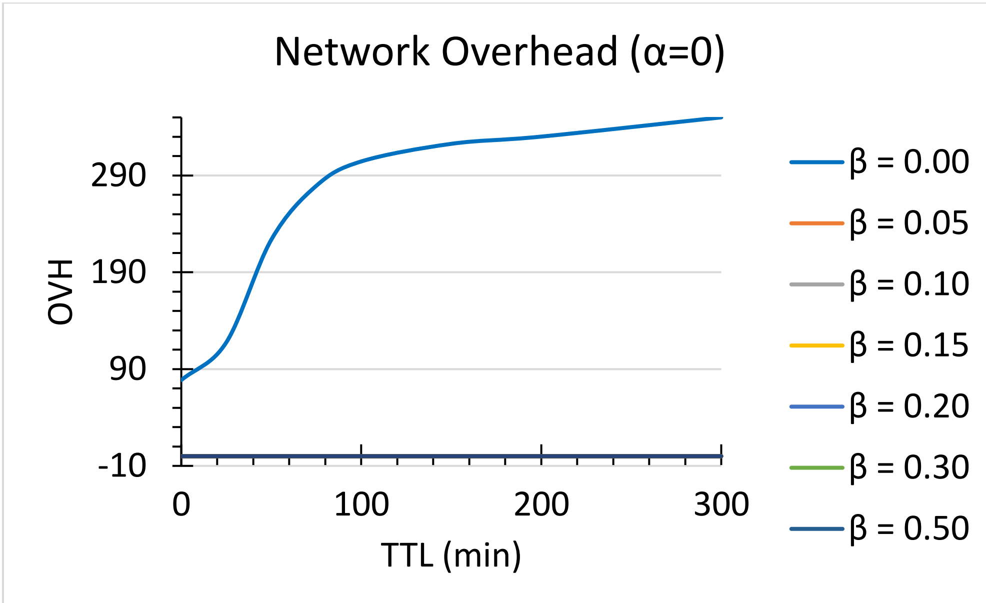


Figure A-8.1. OVH with $\alpha = 0$.

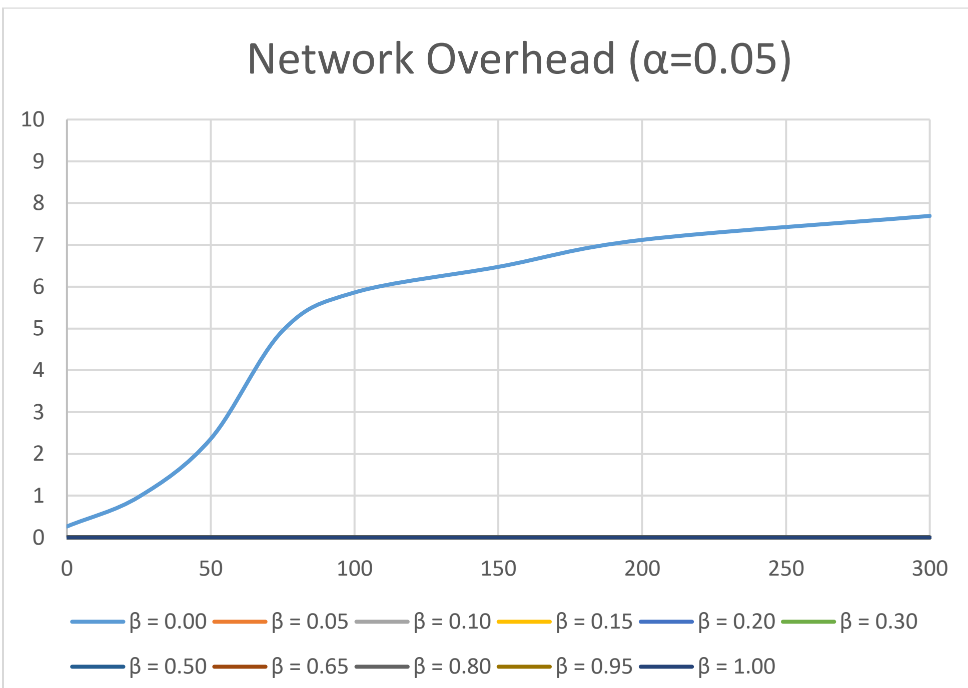


Figure A-8.2. OVH with $\alpha = 0.05$.

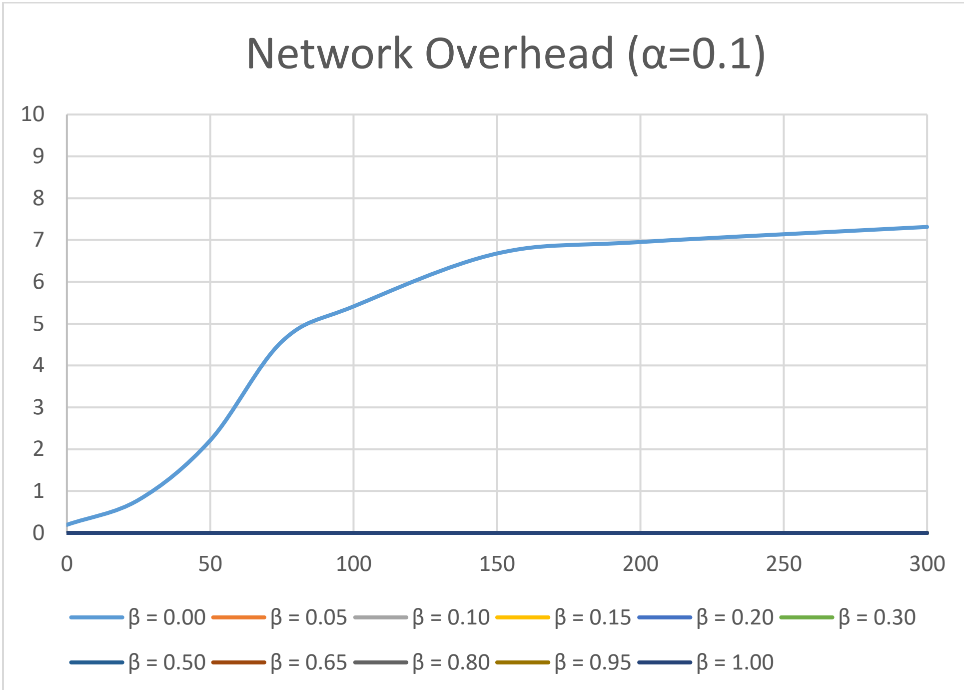


Figure A-8.3. OVH with $\alpha = 0.1$.

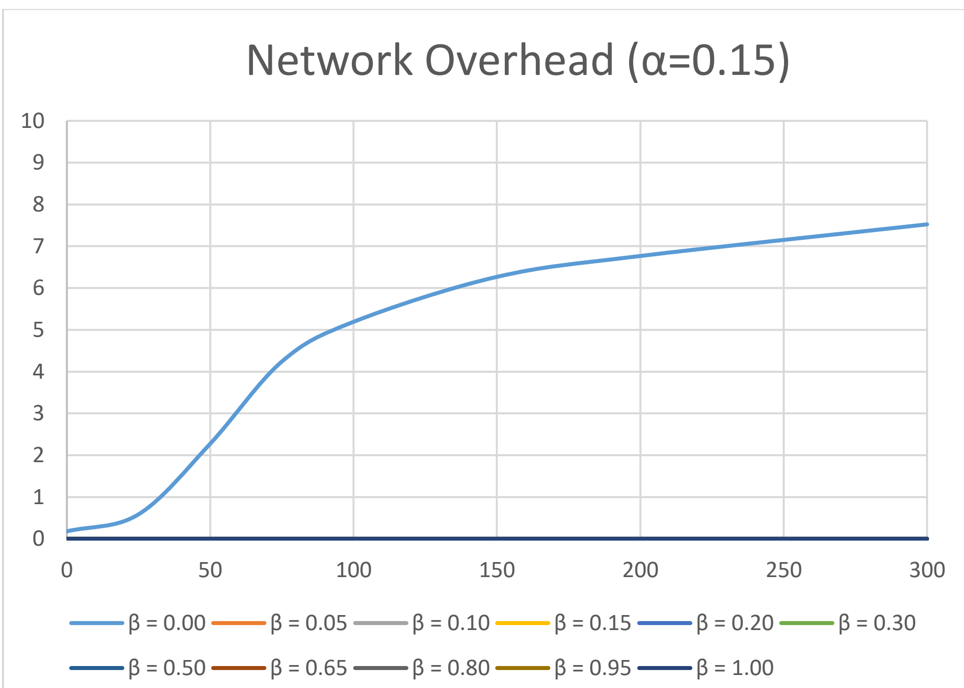


Figure A-8.4. OVH with $\alpha = 0.15$.

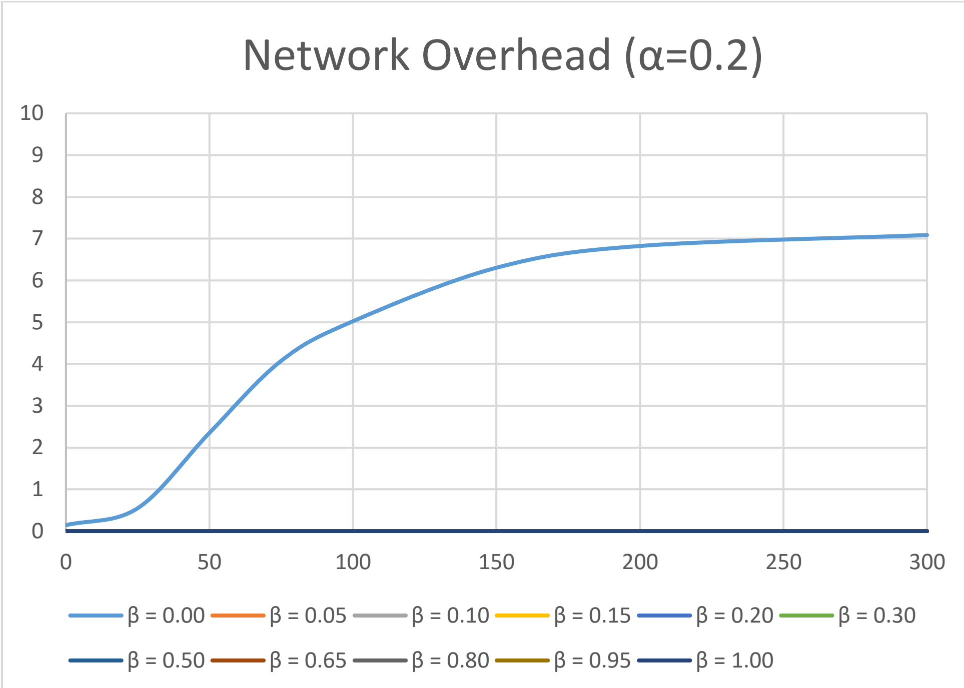


Figure A-8.5. OVH with $\alpha = 0.2$.

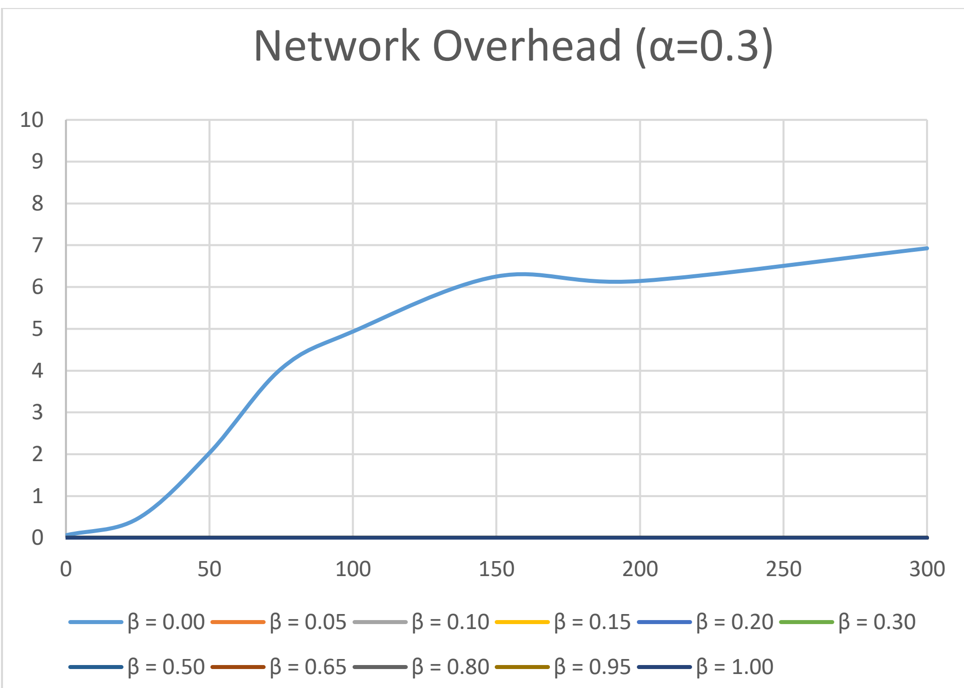


Figure A-8.6. OVH with $\alpha = 0.3$.

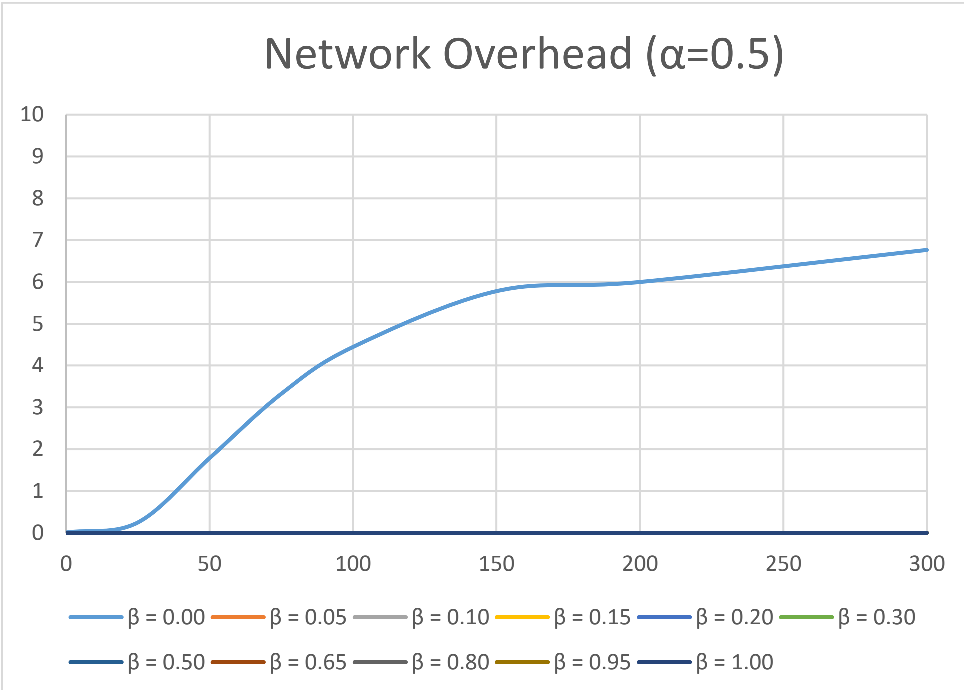


Figure A-8.7. OVH with $\alpha = 0.5$.

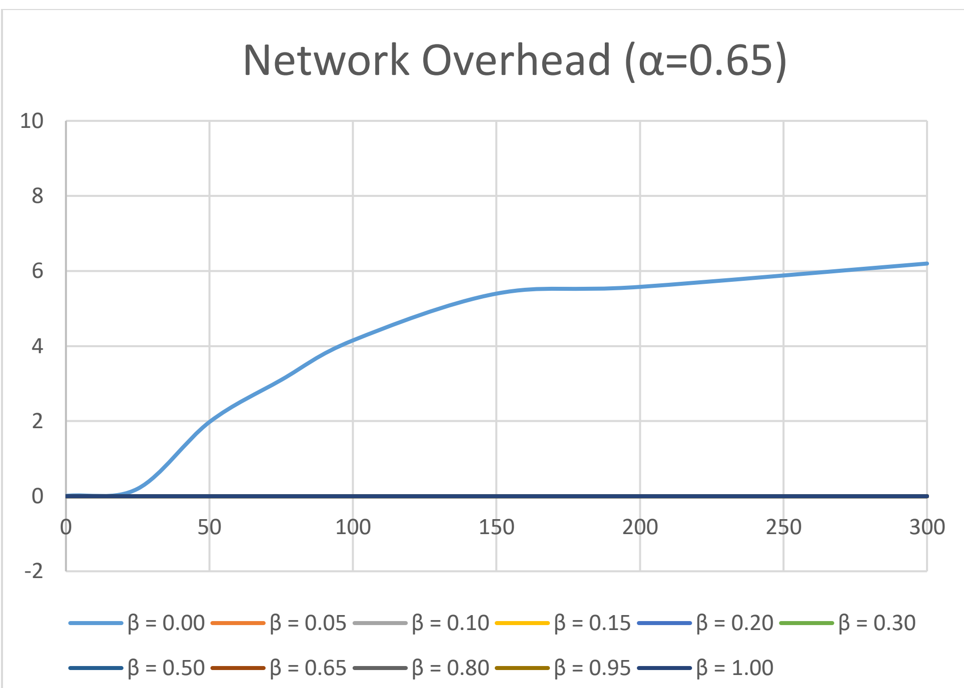


Figure A-8.8. OVH with $\alpha = 0.65$.

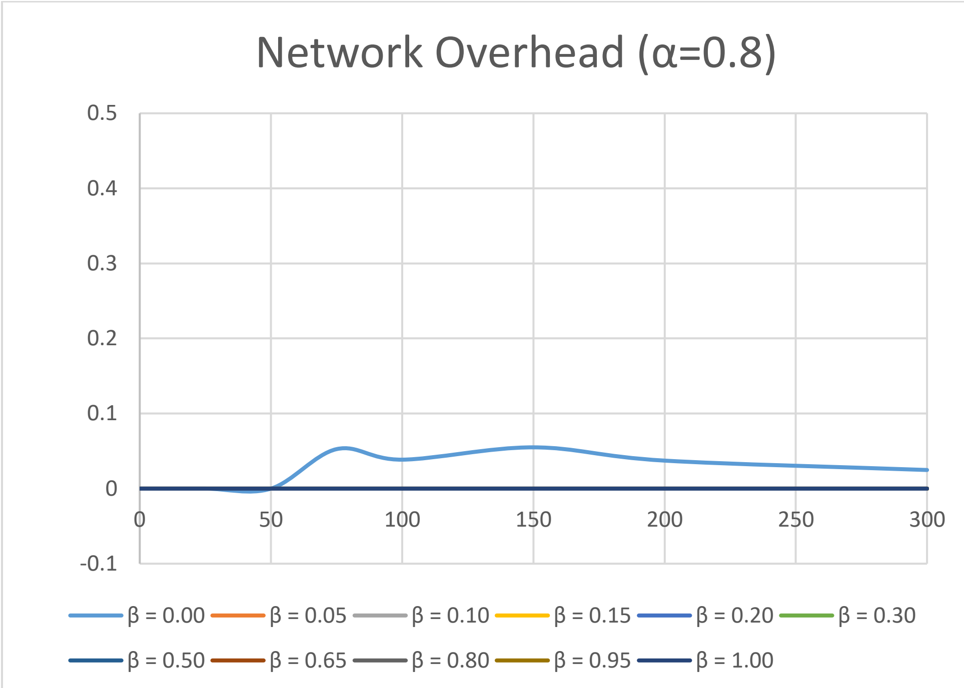


Figure A-8.9. OVH with $\alpha = 0.8$.

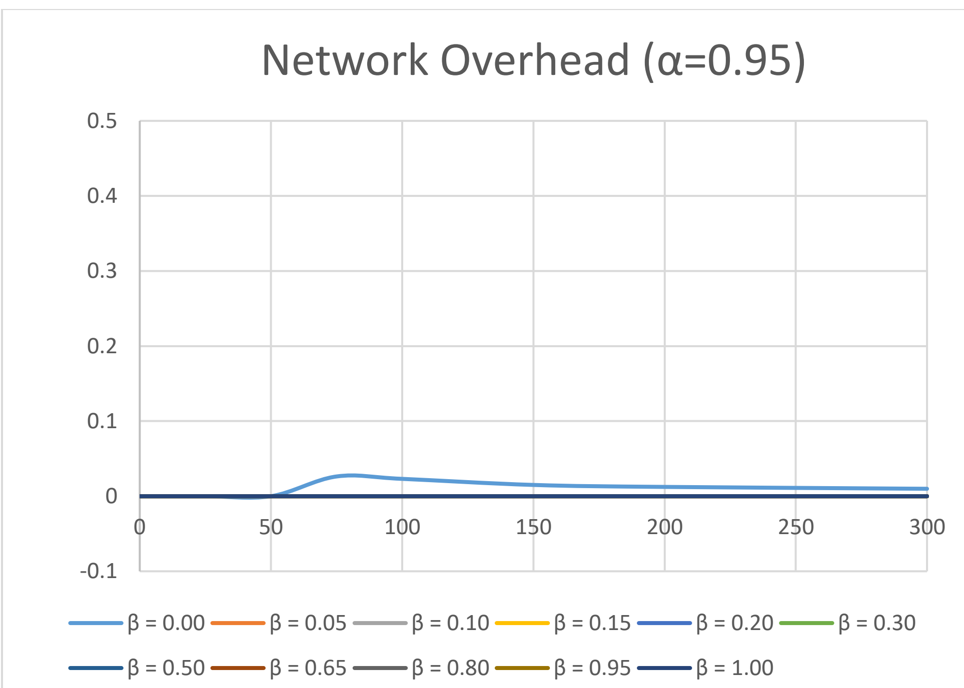


Figure A-8.10. OVH with $\alpha = 0.95$.

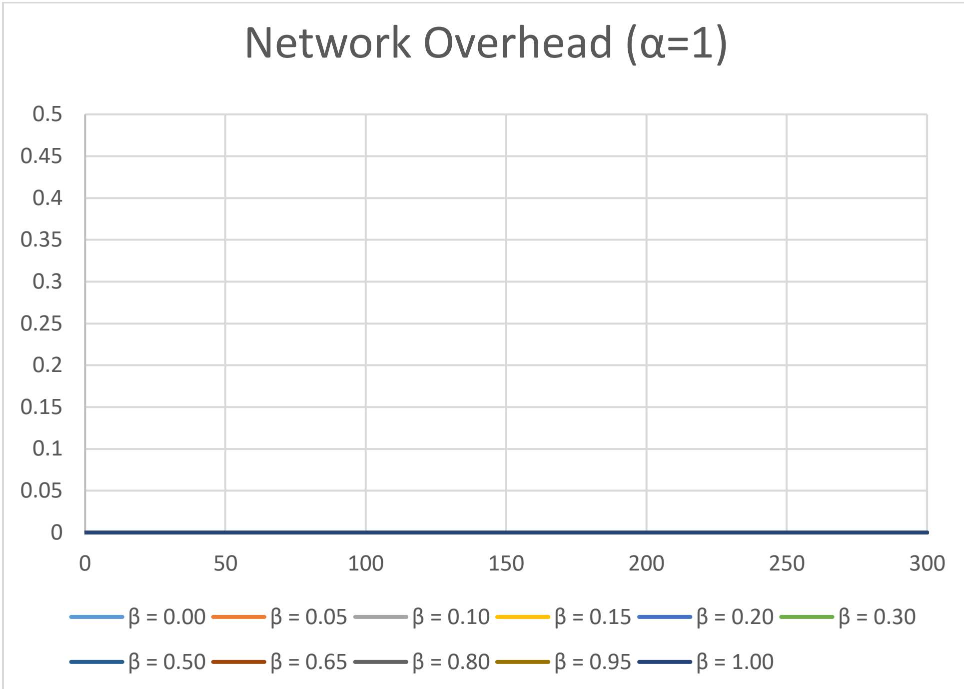


Figure A-8.11. OVH with $\alpha = 1$.

Appendix 9 – Performance of DLR+ on Hop Count with different threshold values (α, β)

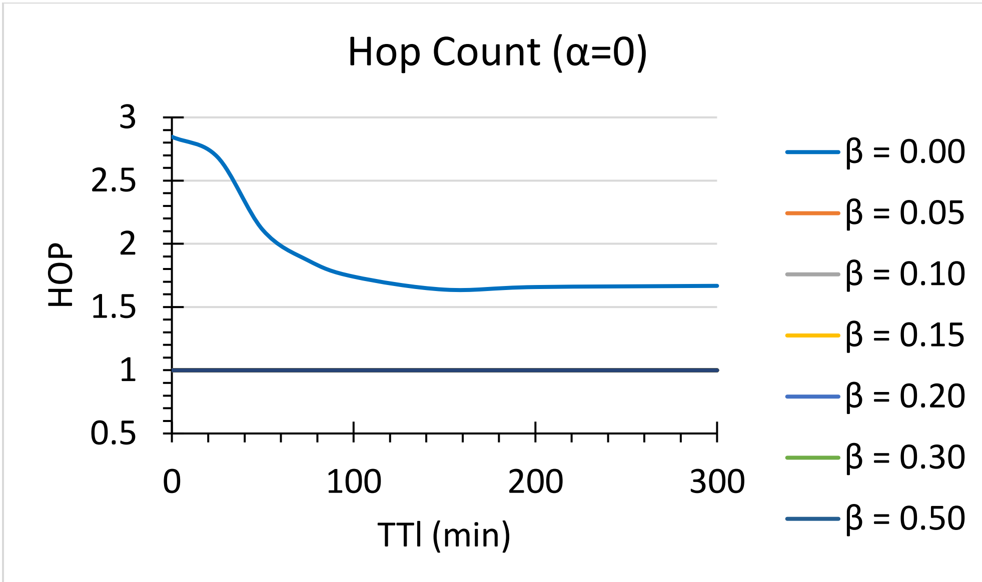


Figure A-9.1. HOP with $\alpha = 0$.

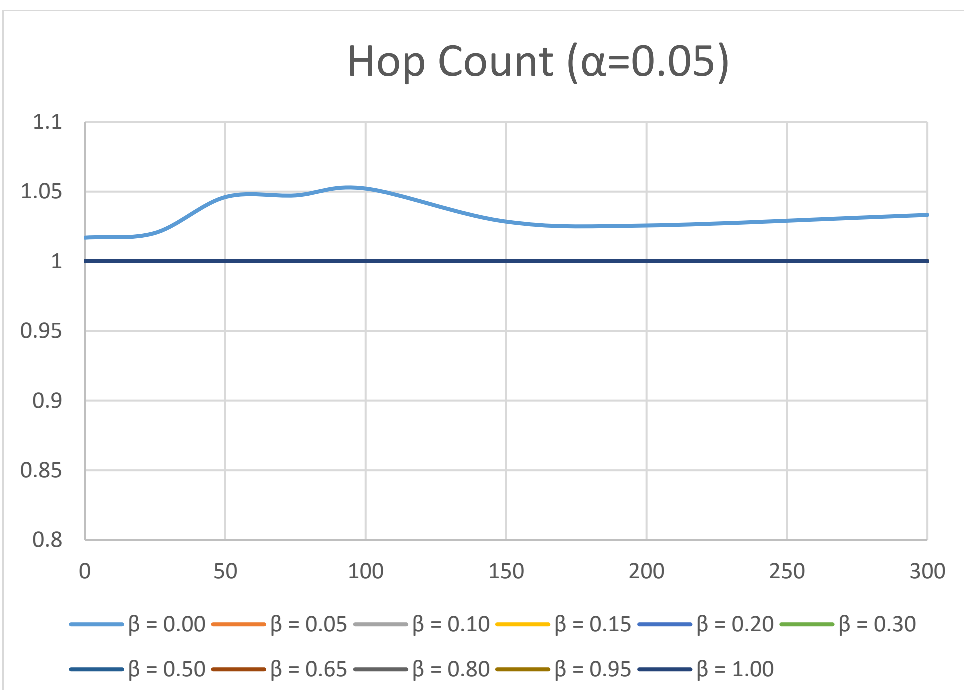


Figure A-9.2. HOP with $\alpha = 0.05$.

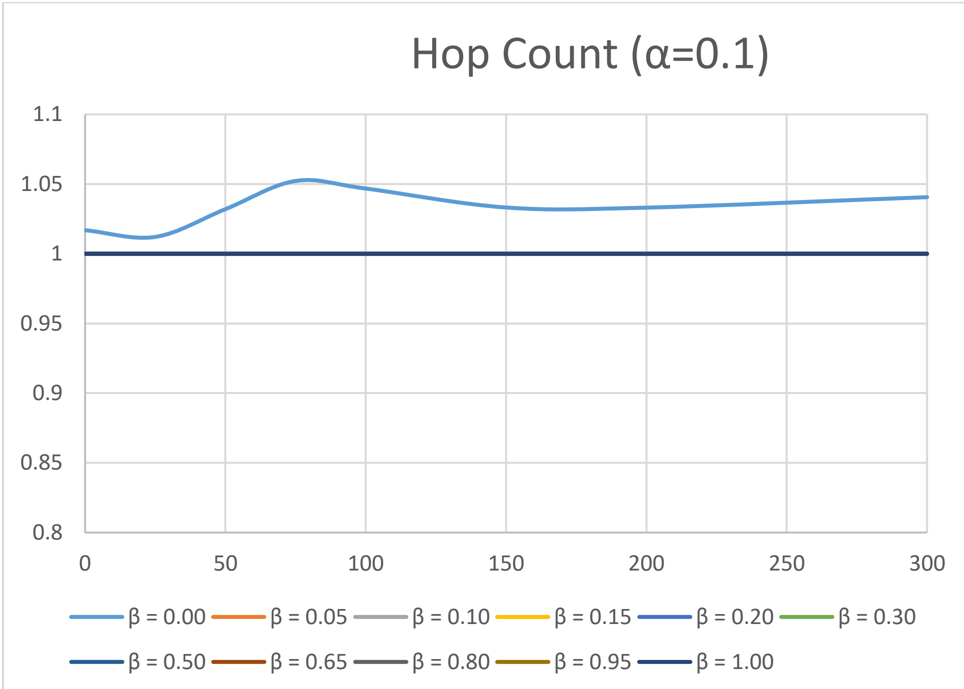


Figure A-9.3. HOP with $\alpha = 0.1$.

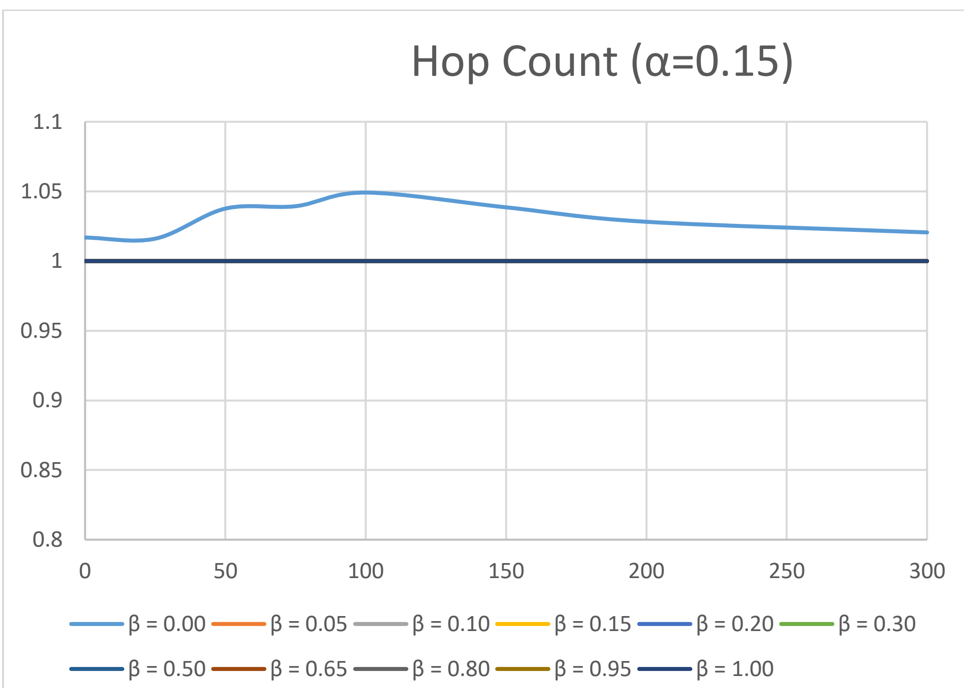


Figure A-9.4. HOP with $\alpha = 0.15$.

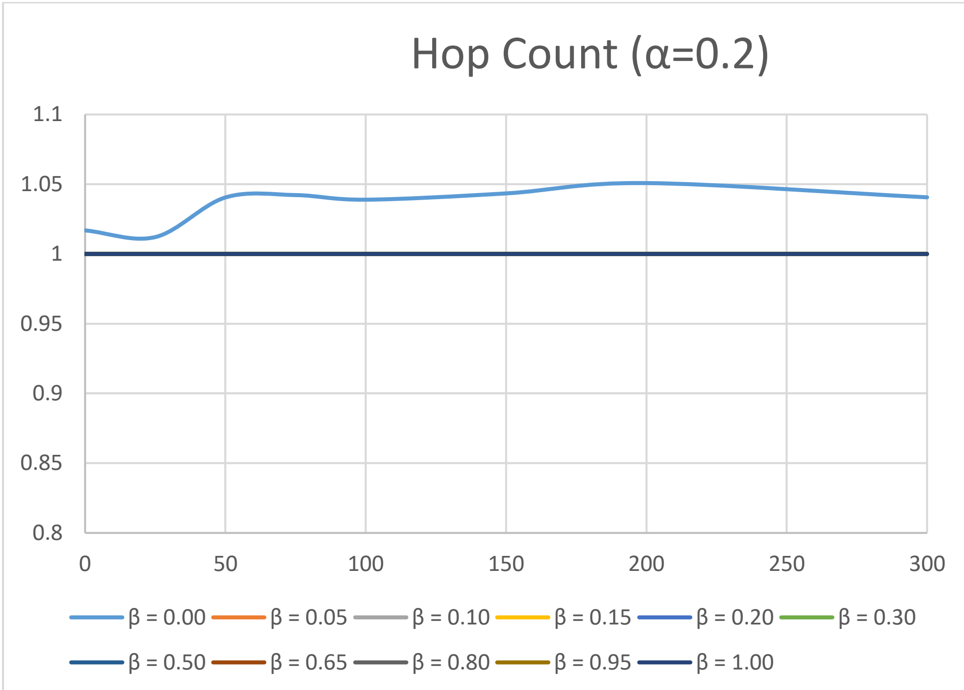


Figure A-9.5. HOP with $\alpha = 0.2$.

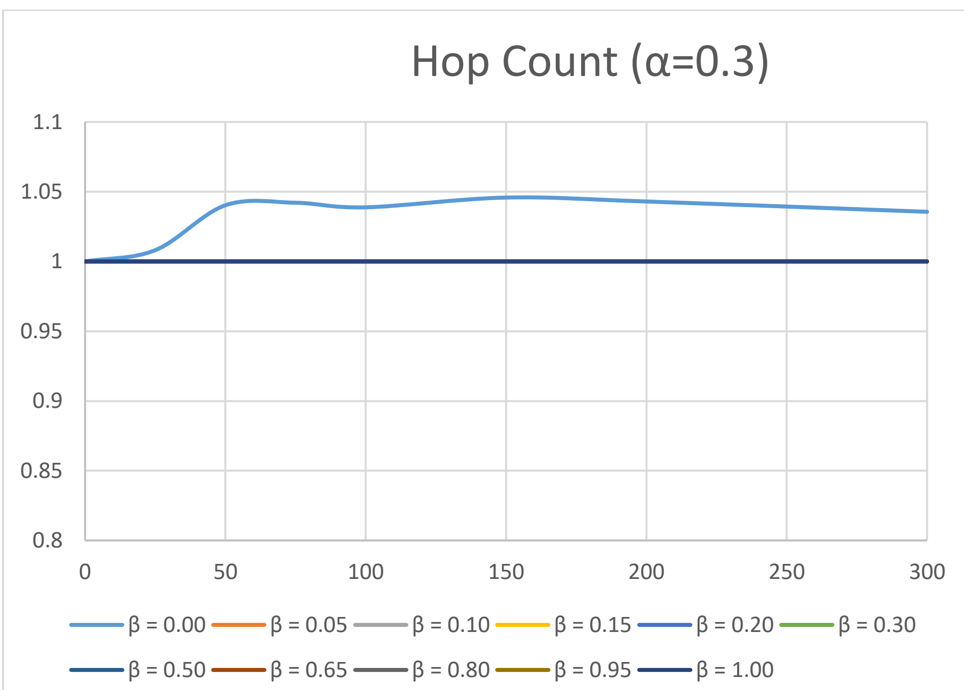


Figure A-9.6. HOP with $\alpha = 0.3$.

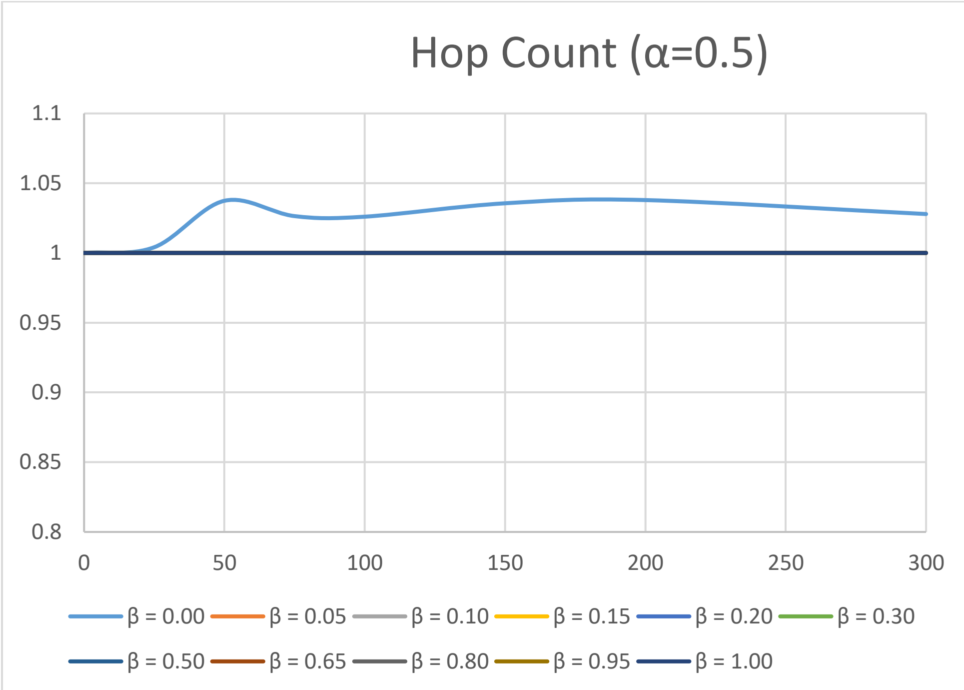


Figure A-9.7. HOP with α = 0.5.

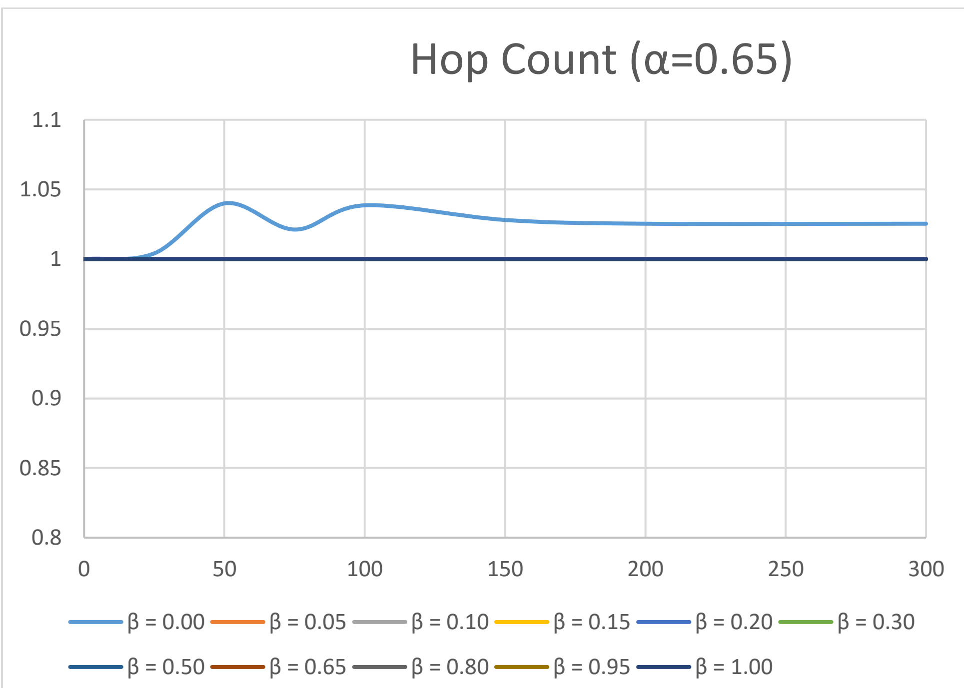


Figure A-9.8. HOP with α = 0.65.

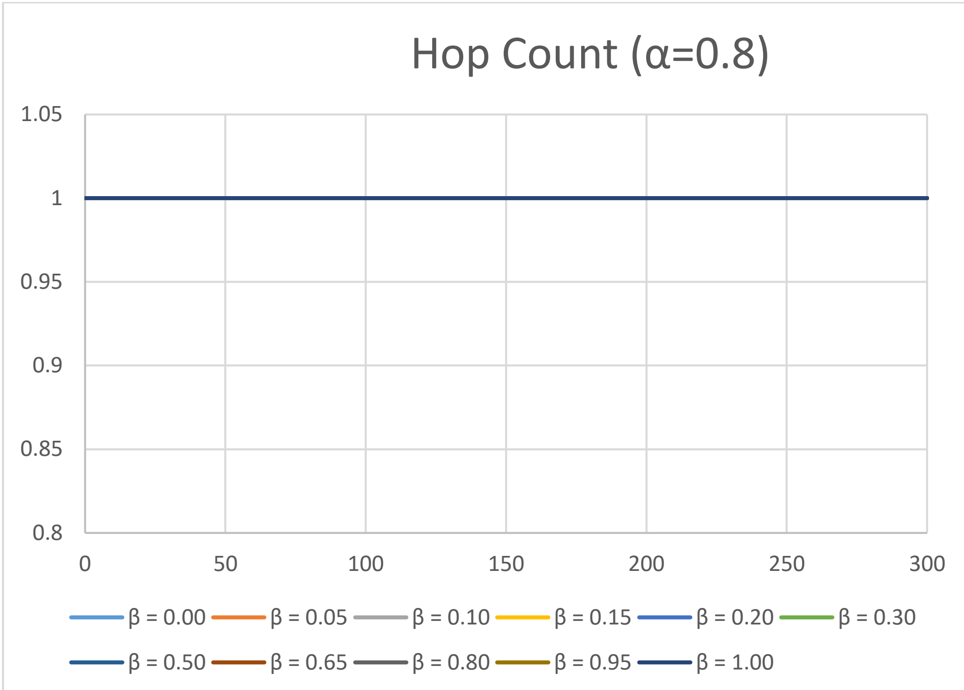


Figure A-9.9. HOP with $\alpha = 0.8$.

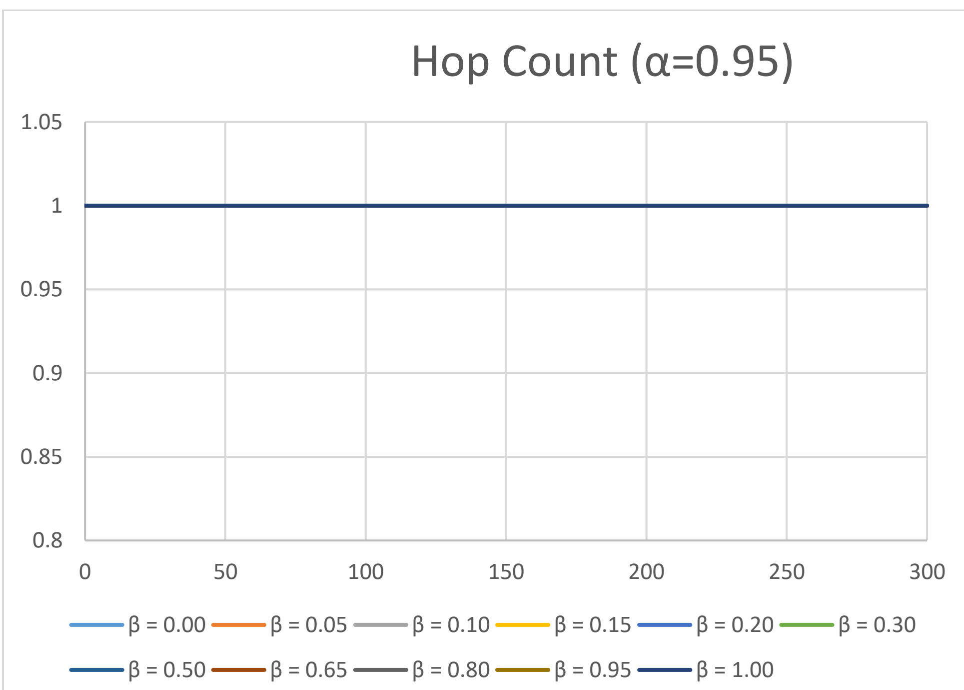


Figure A-9.10. HOP with $\alpha = 0.95$.

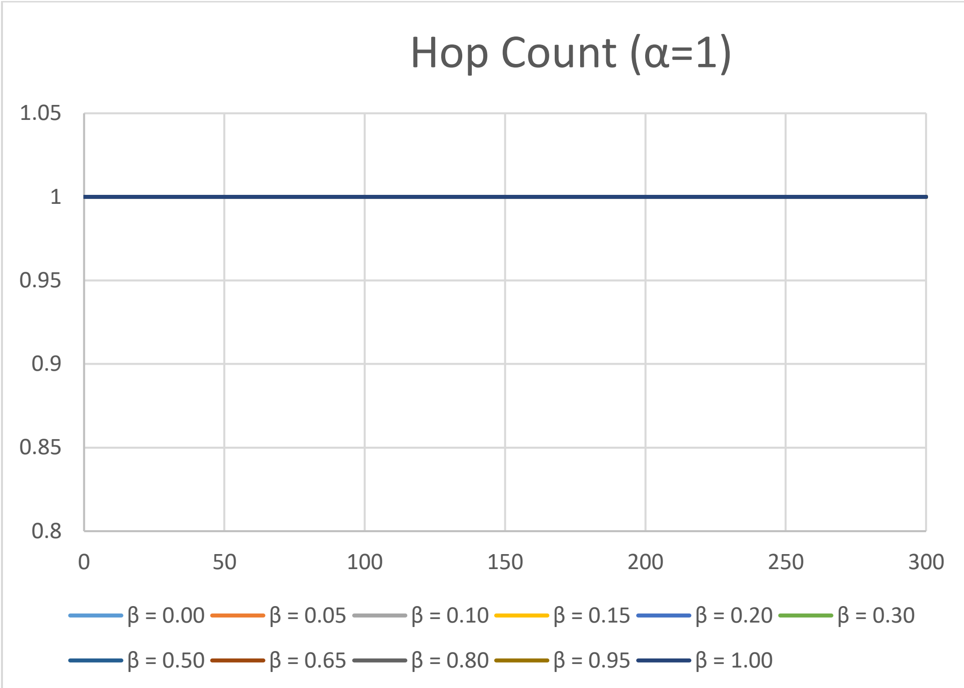


Figure A-9.11. HOP with $\alpha = 1$.