UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Matemáticas y Ciencias Computacionales

TÍTULO: Simple Protocol based on Adaptive Data Dissemination for Urban VANETs

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información

Autor:
Cárdenas Ponce Andrés Fernando

Tutor:
Iza Cristhian, PhD.

Urcuquí, Agosto 2021

Urcuquí, 22 de julio de 2021

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
ACTA DE DEFENSA No. UITEY-ITE-2021-00023-AD

A los 22 días del mes de julio de 2021, a las 14:30 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.
Miembro No Tutor	Dr. AMARO MARTIN, ISIDRO RAFAEL , Ph.D.
Tutor	Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.

El(la) señor(ita) estudiante **CARDENAS PONCE, ANDRES FERNANDO**, con cédula de identidad No. **1805369194**, de la **ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES**, de la Carrera de **TECNOLOGÍAS DE LA INFORMACIÓN**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-43-No.496-2014**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **Simple Protocol based on Adaptive Data Dissemination for Urban VANETs.** , previa a la obtención del título de **INGENIERO/A EN TECNOLOGÍAS DE LA INFORMACIÓN.**

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	Docente	Calificación
Miembro Tribunal De Defensa	Dr. AMARO MARTIN, ISIDRO RAFAEL , Ph.D.	10,0
Tutor	Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.	10,0
Presidente Tribunal De Defensa	Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.	9,0

Lo que da un promedio de: **9.7 (Nueve punto Siete)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

Certifico que *en cumplimiento del Decreto Ejecutivo 1017 de 16 de marzo de 2020, la defensa de trabajo de titulación (o examen de grado modalidad teórico práctica) se realizó vía virtual, por lo que las firmas de los miembros del Tribunal de Defensa de Grado, constan en forma digital.*

CARDENAS PONCE, ANDRES FERNANDO
Estudiante

Firmado electrónicamente por:
ANDRES FERNANDO CARDENAS PONCE

Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.
Presidente Tribunal de Defensa

Firmado electrónicamente por:
JULIO JOAQUIN ARMAS ARCINIEGA

Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.
Tutor

Firmado electrónicamente por:
CRISTHIAN RENE IZA PAREDES

Dr. AMARO MARTIN, ISIDRO RAFAEL , Ph.D.
Miembro No Tutor

ISIDRO RAFAEL AMARO MARTIN
Firmado digitalmente por ISIDRO RAFAEL AMARO MARTIN
Nombre de reconocimiento (DN): c=EC, o=BANCO CENTRAL DEL ECUADOR, ou=ENTIDAD DE CERTIFICACION DE INFORMACION-ECIBCE, l=QUITO, serialNumber=0000228903, cn=ISIDRO RAFAEL AMARO MARTIN
Fecha: 2021.07.26 17:06:38 -05'00'

TORRES MONTALVÁN, TATIANA BEATRIZ
Secretario Ad-hoc

TATIANA BEATRIZ TORRES MONTALVAN
Firmado digitalmente por TATIANA BEATRIZ TORRES MONTALVAN
Fecha: 2021.07.26 16:19:54 -05'00'

Autoría

Yo, **Andrés Fernando Cárdenas Ponce**, con cédula de identidad **1805369194**, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el autor del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Agosto del 2021.

ANDRES FERNANDO CARDENAS PONCE

Firmado digitalmente por ANDRES FERNANDO CARDENAS PONCE
Fecha: 2021.08.12 14:17:39 -05'00'

Andrés Fernando Cárdenas Ponce
CI: 1805369194

Autorización de publicación

Yo, **Andrés Fernando Cárdenas Ponce**, con cédula de identidad **1805369194**, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Urcuquí, Agosto del 2021.

ANDRES FERNANDO CARDENAS PONCE

Firmado digitalmente por ANDRES FERNANDO CARDENAS PONCE
Fecha: 2021.08.12 14:17:53 -05'00'

Andrés Fernando Cárdenas Ponce
CI: 1805369194

Dedication

To my mother Paquita,
for your love, constant support, teachings and advices that have formed the person I am now. I love you!.

To my sister Diana,
for your love and constant support. I will always be grateful!.

To my family,
for their unconditional love.

Thanks for always believing in me.

Andrés Fernando

Acknowledgments

A deep thanks to all the academic staff of the Universidad de Investigación de Tecnología Experimental Yachay for having provided me with an international quality education.

Thank you very much to my advisor Cristhian Iza, for all the teachings and the help provided in the process of developing this thesis.

Thanks to my professors, for instilling in me this great passion for computing and programming.

I thank my great friends Marcos, Mauro, Henry, Bryan, Pedro, Jhonatan, Steeven and Karen for all the joys and teachings that they left me at this stage of my life. I will miss you very much, hope to see you soon.

Abstract

The advancement of wireless network technologies has opened many doors to the development of novel applications in diverse areas of research. Vehicular Ad-Hoc networks (VANETs) represent a clear example of the evolution of wireless communications. Their implementation facilitates road safety and comfort for both the driver and passengers. In this type of communication, the dissemination of emergency messages is vital for the proper functioning of most of the applications used in this technology. This is why a great variety of information dissemination schemes and protocols have been developed, which face challenges such as broadcast storms and disconnected networks. These problems do not allow the correct dissemination of the data in the vehicular network and are the main objective to overcome in this research work. In this thesis, we present the adaptive probabilistic protocol for the dissemination of data in real-time RTPAD. Our proposal seeks to solve the challenges that arise in the dissemination of data in-vehicle networks. In this document, we explain its implementation, operation, and evaluation in comparison with other dissemination protocols.

Keywords: VANETs, Dissemination, Emergency Messages, RTPAD, Adaptive, Probabilistic.

Resumen

El avance de las tecnologías de redes inalámbricas ha abierto muchas puertas al desarrollo de aplicaciones novedosas en diversas áreas de investigación. Las redes vehiculares Ad-Hoc (VANET) representan un claro ejemplo de la evolución de las comunicaciones inalámbricas. Su implementación facilita la seguridad vial y la comodidad tanto para el conductor como para los pasajeros. En este tipo de comunicación, la difusión de mensajes de emergencia es vital para el correcto funcionamiento de la mayoría de las aplicaciones utilizadas en esta tecnología. Es por esto que se han desarrollado una gran variedad de esquemas y protocolos de difusión de información, que enfrentan desafíos como tormentas de transmisión y redes desconectadas. Estos problemas no permiten la correcta difusión de los datos en la red vehicular y son el principal objetivo a superar en este trabajo de investigación. En esta tesis presentamos el protocolo probabilístico adaptativo para la difusión de datos en tiempo real RTPAD. Nuestra propuesta busca dar solución a los retos que surgen en la difusión de datos en redes de vehículos. En este documento explicamos su implementación, funcionamiento y evaluación en comparación con otros protocolos de difusión.

Palabras Clave: VANETs, Diseminacion, Mensajes de Emergencia, RTPAD, Adpativo, Probabilstico .

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

One of the long-standing dreams of human beings is to travel safely and comfortably. Currently, the advancement of technology in vehicles has allowed us to pose new challenges to fulfill this dream [4]. Thus, vehicle companies, government institutions, and tech visionaries have been involved in the development and evolution of transportation systems that reduce the risk of accidents on the roads [5]. Many of these systems are already in place, but they have provided limited safety, focusing on the impact of the accident and not its prevention [6]. Likewise, the progress in the research and development of the networks was a great help to reach what is now the intelligent vehicle networks called VANET [7]. This technology allows vehicles that are circulating to share information with other vehicles in order to collect and better understand the environment where they are. Indeed, thanks to this information that is generated, the driver can be warned and an accident can be prevented by taking the appropriate measures [8].

Currently, there is a wide variety of large and small-scale projects around the world for the development of different applications offered by vehicular networks [9]. Among the main applications are traffic control, route optimization, road safety and comfort, internet connection, real-time information exchange, commercial services, among many others[10]. The applications developed in VANET are mainly based on the transmission or multiple broadcasts of messages in a specific area[11]. Faced with a dangerous situation or an accident, it is best to inform neighboring vehicles of the event and if this information is required they can reach other vehicles located in more distant places in the shortest possible time. In this research, we are interested in proposing an adaptive dissemination protocol in a vehicular network optimizing the use of the communication channel. The contribution of this study ranges from the evaluation of existing dissemination protocols in the literature to the presentation of a novel data dissemination protocol that exceeds the performance of current dissemination protocols.

1.2 Main Objective

The main objective of this research is to propose, design and evaluate an adaptive data dissemination protocol for vehicle networks that achieves successful dissemination and optimizes the communication channel resources. Based on the scope described, the research question for this work is: *How to achieve an efficient and optimal diffusion of emergency messages for vehicular networks?*

1.3 Specific Objectives

- Evaluate several data dissemination protocols in terms of informed vehicles, number of messages received and number of collisions at MAC-layer.
- Propose a Real-time Probabilistic Adaptive Dissemination Protocol to perform data dissemination.
- Compare our dissemination scheme with other protocols in terms of informed vehicles and collisions at MAC-layer.

1.4 Organization of the Thesis

- Chapter 2 shows the literature review of articles that present the main concepts about vehicle networks, their communication and their characteristics in general. We also cover the applications in VANETs, and the main problems that we face in this technology. We talk about the advantages of V2V communication and what we assume for the correct evaluation of our proposal.
- Chapter 3 presents all the tools we use for the development and evaluation of data dissemination protocols. We describe their main characteristics and how they interact with each other.
- Chapter 4 explains the state of the art in our field of study. Here, we describe several existing data dissemination protocols.
- Chapter 5 presents the methodology used in this thesis. We explain in detail our adaptive data broadcast protocol, its main concepts, and how it works. The scenario where our proposal is evaluated is also considered.
- Chapter 6 shows the results obtained by the simulations.
- Chapter 7 presents the conclusions of this research work.

Chapter 2

Vehicular Adhoc Networks

During the last decade, the development and implementation in real life of Intelligent Transportation Systems (ITS) grew exponentially [4]. Ad-hoc Vehicle Networks (VANETs) are a promising technology that opens a new world of possibilities to improve the efficiency and security of Intelligent Transportation Systems (ITS) through the development of a wide variety of applications. VANET is one of the types of MANET (mobile ad-hoc networks) that allow vehicles to communicate with each other where each node acts as an independent and self-organizing network [12]. The main objective of the VANETs is to guarantee road safety and traffic management to offer a better driving experience, and also to avoid dangerous situations for the driver and pedestrians[4]. One of the main requirements in VANETs is the design of mechanisms to achieve efficient and optimal data dissemination. However, some of the characteristics of VANETs such as highly dynamic topology, the frequently disconnected network, dynamic network density, have made data dissemination a really difficult task. Over the years, there have been endless proposals trying to tackle this challenge in very ingenious ways. Despite this, most of the proposed solutions do not completely solve the problem, since they do not efficiently solve data dissemination problems such as broadcast storm, network partition and temporary network fragmentation [13].

2.1 Vehicular Networks

Ad-hoc vehicle networks are made up of vehicles equipped with wireless radio interfaces, with which they can communicate with nearby vehicles; as long as they are in your radio transmission range. Vehicle networks are also made up of Roadside Units (RSU), which fulfill the function of intermediate nodes to connect distant nodes or capture specific information according to the objective of interest. In the following, characteristics of a vehicular network are detailed [14].

2.1.1 Definition

Vehicle networks are considered a particular class of Mobile Ad-Hoc Networks (MANET) and their main difference is that nodes are restricted to moving only along the length and width of the tracks [15]. Ad-hoc vehicle networks consists of two types of nodes, static nodes, and mobile nodes. Static nodes or stationary access points called RSU (Road-Side Unit) are fixed elements, located along the roads, whose function is to send, receive, and re-transmit packets to increase the range of network coverage. On the other hand, mobile nodes are vehicles that are equipped with an on-board communication unit called OBU (On-Board Unit) and an application unit called AU (Application Unit). The OBU's role is to exchange information with other vehicles or with stationary access points (RSU) located around the roads; while AUs refer to devices that display information to the user. Generally, this name is given to devices such as laptops, smartphones, or screens located inside the node and that are connected to the OBU. Besides, OBU is a device that has wireless or wired communication capabilities [16]. Figure 2.1 shows an example of an overview of VANET network structure.

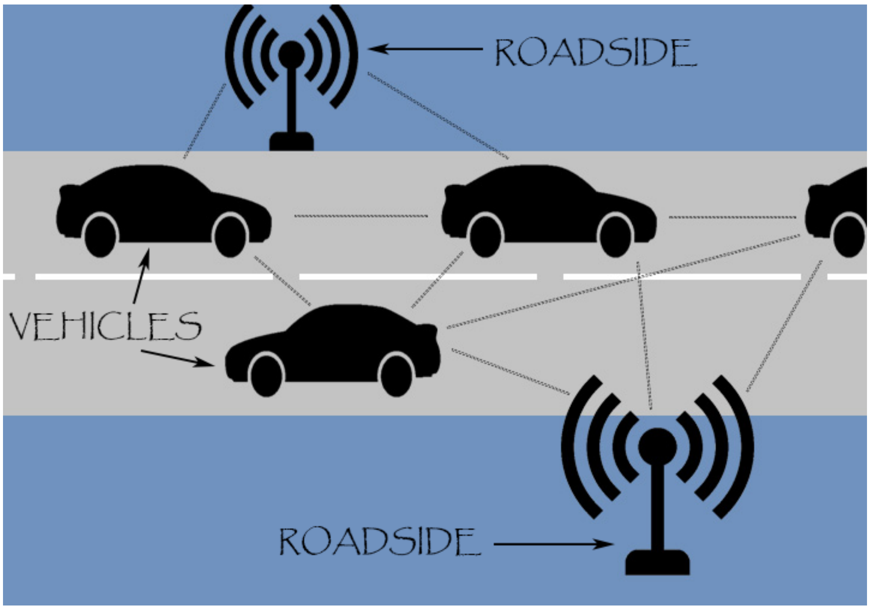

Figure 2.1: An overview of a VANET network

To be part of this network, a vehicle must be equipped with communication tools, both wireless and geographic positioning systems (GPS) [17] allowing the interaction of both input and output data, to later analyze and process the information. The vehicle can send both point-to-point messages for its neighbors, as well as alert messages in the event of an accident. Figure 2.2 illustrates in a general way some equipment that a smart car should have to be part of the network.

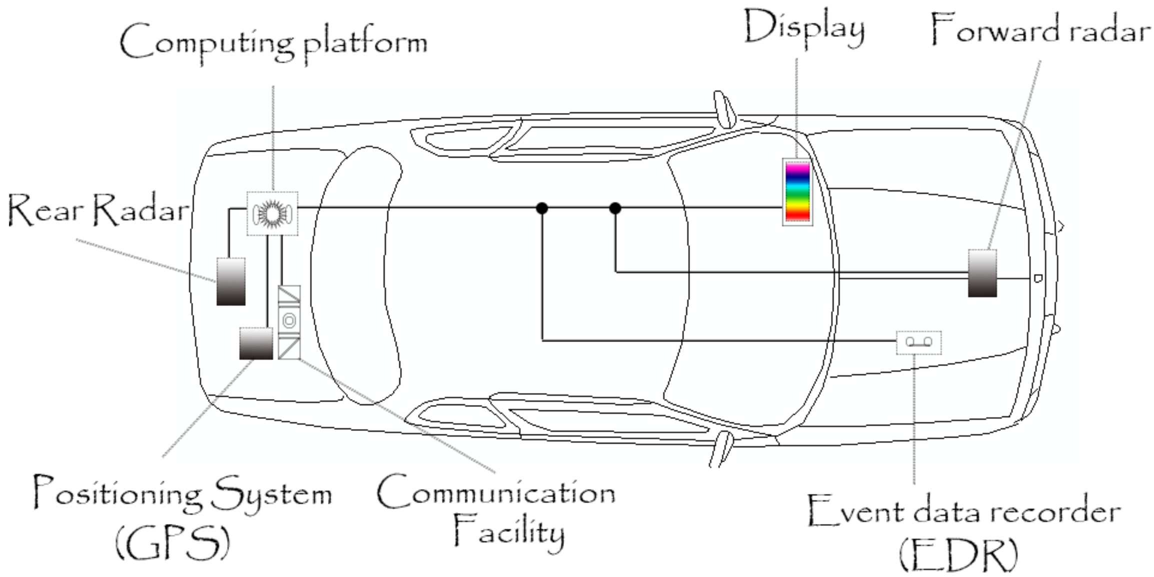

Figure 2.2: Different equipment present on a VANET vehicle

One of the main objectives of Intelligent Transport Systems (ITS) is to provide better road knowledge for drivers, to reduce the number of accidents and provide a better driving experience [18]. The applications that VANETs have are from a simple exchange of information between their nodes (vehicles) to the use of these networks focused on road safety and traffic efficiency. Through VANETs, we can obtain a set of data for each node such as geographical position, speed, direction, among others. These data will allow us to know the state of the traffic, vehicles that exceed speed, and accidents in different geographical areas, also to give us warnings to drivers of possible collisions. Despite the great advancement of VANET networks, they present a number of problems, among which the following can be highlighted: the limited coverage offered by wireless networks and the high mobility of nodes. Due to this, the links that are created during communication have a very limited lifetime. Therefore, this makes it difficult to exchange data and, in turn, the efficient routing of packets in a vehicular network becomes a highly complex task[18].

2.1.2 General Characteristics

VANETs are similar to MANETs in that they both allow ad hoc communications between nodes in constantly changing environments. However, the investigations carried out in the field of MANETs are not applicable in vehicular networks due to some specific properties that they have [19] [20]. These are some properties that differentiate these network structures.

- **Network density and topology:** VANETs are characterized by their highly mobile and constantly changing environment. A vehicle can join and leave the network in a short time and this changes the topology. The density of VANETs can vary according to the number of vehicles in a given area. They range from low densities (few vehicles) to high densities (traffic in urban areas) [21] [22]. This means that VANET protocols must address this density variability in different environments.

- **Network distribution and Mobility model:** The topology constantly changes but the distribution of vehicles is not random because they follow patterns or paths (roads, streets) as seen in Figure 2.3. The mobility of the vehicles in the network is governed by the predefined routes, where the street, direction, the number of lanes, etc. are specified. [16]. The environments of the vehicular networks can be a highway, urban or rural and thus their trajectory can be predicted. In the environment, limitations are defined to obtain a more realistic scenario, such as obstacles, speed limits, traffic lights and parked vehicles which affect the vehicular density and the mobility model. Due to this, the protocols must take into account these characteristics for their correct performance.

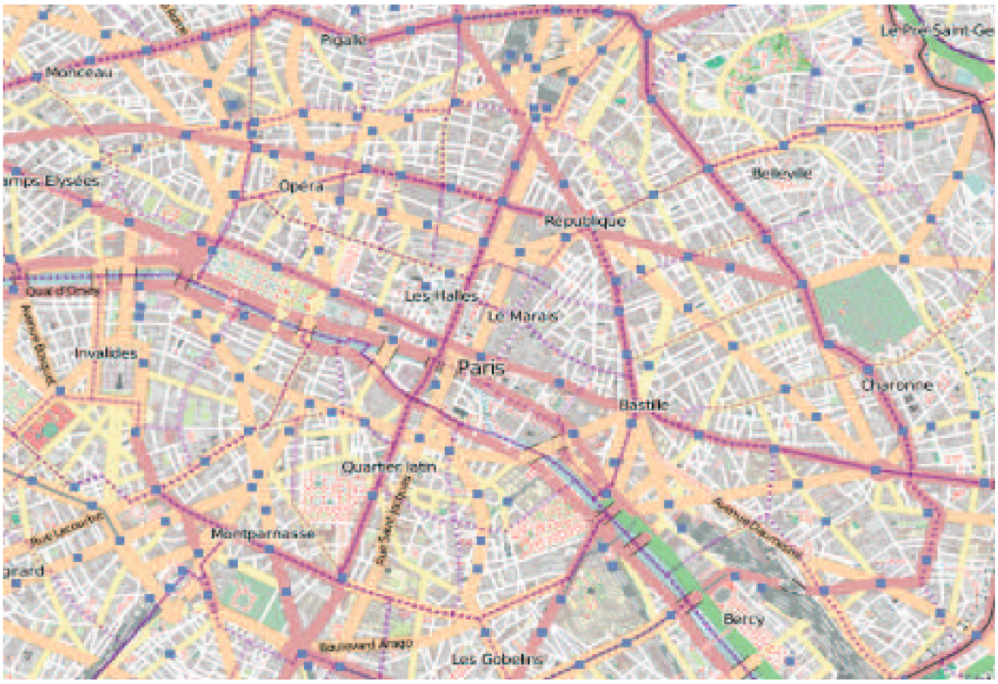

Figure 2.3: Example of a road network

- **Energy capacities and Processing:** Vehicles in motion provide high computing and electrical power. Because of this, energy use is not an important factor in VANET. Furthermore, all nodes have an on-board unit (OBU) which allows simultaneous execution of applications for a large number of wireless communication technologies, such as Bluetooth [23], WiMax6 [24], WIFI5 [25], etc.

- **Scalability:** A VANET tends to grow exponentially, this happens more frequently in urban areas comprised of multi-lane highways, intersections or high density of vehicles. Therefore, the data dissemination protocols must face the existence of a large number of interferences between vehicles and possible wireless collisions during their communication or participation in the vehicle network. [6].

- **Security and anonymity in the network:** It is important to guarantee the privacy and security of the vehicles participating in the network. The type of communication used in VANET is vulnerable because an attacker could enter the network in the same way as a legitimate node. This is why there are many specialized security and privacy protocols for vehicle networks. [26] [27].

2.1.3 Related Projects

So far, a number of projects have been registered that are based on vehicle networks, since this is considered a technology on the rise [15]O. In Europe and USA many research projects have been developed based on the intelligent transportation system.

- **CVIS[28]:** This research project aims to develop, test, and implement the technologies necessary for automobiles to have connections with each other, and with the closest infrastructure.

- **Car2Car Consortium[29]:** This is an international project supported by the European Telecommunications Standards Institute (ETSI). Its main objective is to improve road safety and effectively manage traffic through the use of communications between vehicles.

- **Projects in USA:** Here we can talk about the Automated Highway System (AHS) project [30]. The main objective was to reduce accidents on the roads by helping drivers to reduce risks and losses. Another project may be the Intelligent Vehicle Initiative (IVI) [31]. Its objective is to offer an active cooperative insurance between moving vehicles and to develop especially transport systems that help the driver. These systems have a common purpose such as route indications and road collision warnings.

2.2 Vehicular communications

The communication in VANETs can be classified in the categories: Vehicle to Vehicle(V2V), and Vehicle to Infrastructure(V2I). In Figure 2.4, we show this classification, and we can see how this type of communication is applied for a number of protocols based on infotainment, driver assistance, traffic efficiency, as well as safety.

- **V2V:** It allows direct vehicular communication without dependence on fixed infrastructure support. It is mainly used for security and broadcast applications. V2V communication tolerates failures in the distributed environment as vehicular networks are highly dynamic in nature [32].

- **V2I:** It allows a vehicle to communicate with the road infrastructure. Primarily, this mode of communication is used for information and data collection applications. The network infrastructure is also composed of a telecommunications network with road side units (RSUs). They are located at strategic points in the area of interest and connected through a wired network [32].

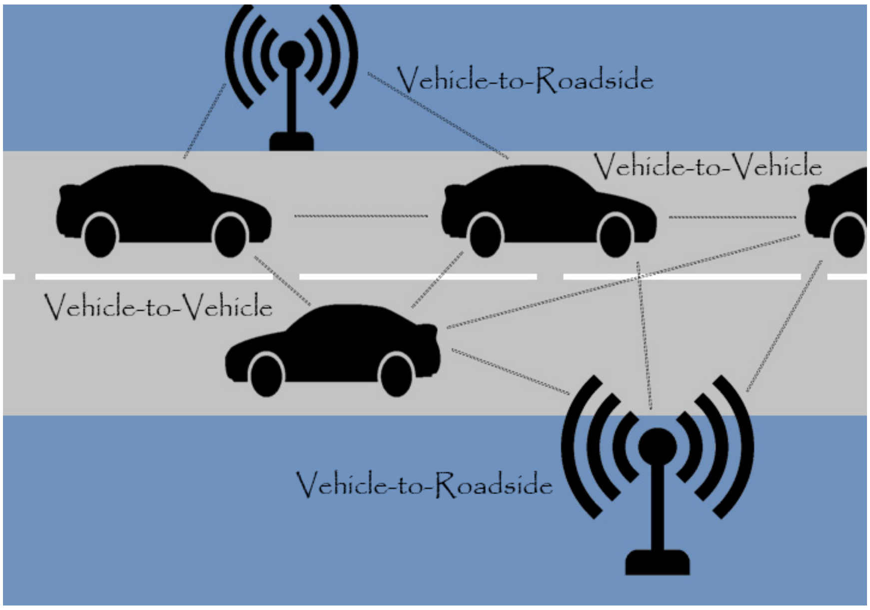

Figure 2.4: Vehicular ad hoc network (VANET) communication architecture.

To create communication between the vehicles that make up the network, dedicated short-range communications devices (DSRC) are used, these allow high-frequency communication between infrastructure and vehicles. The use of the 5.9 GHz frequency with a wide spectrum of 75 MHz for DSRC was granted by the Federal Communications Committee (FCC) this for priority uses of public safety [33]. The maximum range that a DSRC device can have is up to 1000 meters, this would mean that it is a free environment without obstacles. If we have a network in which there are many obstacles, wireless connectivity is around 300 meters. DSRCs provide low latency and high data throughput in a vehicular network. Because of this, DSRC communication gives vehicles the ability to transmit and receive information with vehicles that are out of sight distance or not in line of sight. Thanks to data dissemination protocols, vehicles can forward and re transmit messages throughout the vehicle network; achieving a much greater reach than radio. The IEEE 802.11p standard. specifies the media access control layers (MAC) and physical layers (PHY) for vehicular communication. The standard defines data rates from 3 to 27 Mbps. The combination of IEEE 1609 and the IEEE 802.11p protocol suite is denoted as wireless access in vehicular environments (WAVE)[34].

2.3 VANETs Applications

Vehicle networks and communications have proven to be a powerhouse in driving awareness and road safety.It is a fact that more than half of traffic accidents could have been avoided if the driver had been alerted before the accident [35]. However, this technology is not limited only to road safety, but there are new scopes and benefits that can be achieved by its application. Thus, VANET applications can be established in three categories: traffic coordination, infotainment and security. The main ideas of these categories are explained here.

- **Traffic Coordination Applications.** Vehicle communications are essential for this type of application since vehicles share the traffic information that they collect individually in order to optimize traffic flow and improve the driver's experience. An automated way of traffic control would be with infrastructure that allows to store traffic data in a dynamic way, monitor the streets and predict the congestion of the roads in a specific area of interest. RSUs use the information collected to inform nearby vehicles about local traffic in their area. The information received by the vehicles is used to take available optimal routes and alert drivers according to traffic conditions [36]. V2V communication allows vehicles to join a moving traffic without disrupting the flow traffic. It means, at moment that a vehicle merges into the roadway, the vehicle gets maneuvers to accomplish a not disruptive merge.

- **Infotainment.** This category focuses on providing drivers with information and especially entertainment during their trip. This application of VANETs allows drivers to connect to the internet, exchange files, download and upload videos and information services [37]. This type of service is delayed and may fail in different areas where the connection is not stable. This category is one of the most promising for future work and research.

- **Safety Applications:** Security applications are responsible for granting both the driver and the passenger a safe trip. The vehicle receives relevant information from the environment to inform the driver. This kind of applications is usually delay-sensitive. Table 2.1 presents a summary of the main applications in safety

Safety Applications	**Features**		
	Communication	**Latency[ms]**	**Messaging Type**
Emergency Electronic Brake Lights	V2V	100	Event-triggered, time-limited broadcast
Slow Vehicle Warning	V2V	100	Periodic permanent broadcast
Pre-Crash Sensing	V2V	50	Periodic permanent broadcast, unicast
Lane Change Warning	V2V	100	Periodic broadcast
Intersection Collision Warning	V2V-V2I	100	Periodic permanent broadcast
Hazardous Location Warning	V2V-V2I	100	Event-triggered, time-limited Geocast

Table 2.1: Security applications in VANETs [1]

2.4 Data Dissemination in VANETs

Currently, the dissemination of data in vehicle networks is a well-known topic, and even more well-known are the problems involved in this process. The information that is handled in theA VANET is oriented to the diffusion in all the nodes that make up a group within an area of interest. In other words, the information to be disseminated is sent to everyone by means of multi-hop re transmission. This is different from the routing technique, which is to share data from a specific source to a specific destination [38] [39]. One of its main advantages is that it does not need an IP address or the route to take to that node [40] [41]. Figure 2.5 shows an example of how a Dissemination protocol works in an urban scenario where cars are colored when they received the warning message. The most important challenges for data dissemination are broadcast storm, network partition, and temporal network fragmentation.

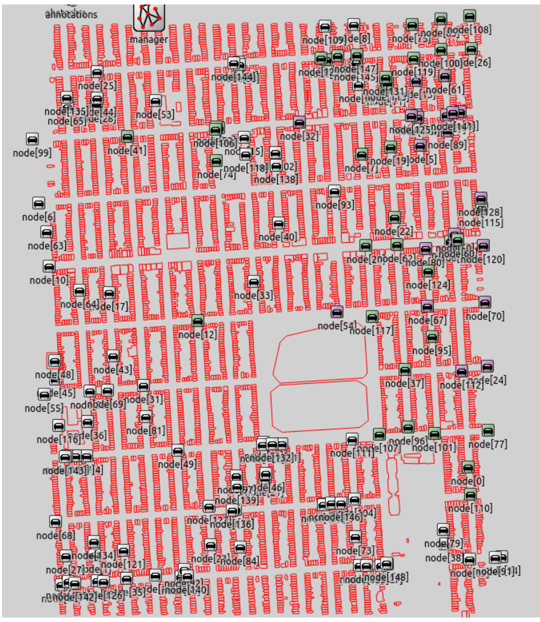

Figure 2.5: Dissemination Protocol in urban scenario

- **Broadcast storm problem:** It occurs when multiple vehicles transmit data at the same time. Therefore, this provokes network congestion, packet collision, and transmission delay. It is generated in high traffic densities due to the excessive forwarding of messages in the network. Due to this, contention and collision occur in the MAC layer, which leads to the collapse of the service in the vehicle network [42]. Figure 2.6 shows an example of a transmission storm problem in vehicular networks. This serious problem occurs on urban roads, during hours of high traffic flow.

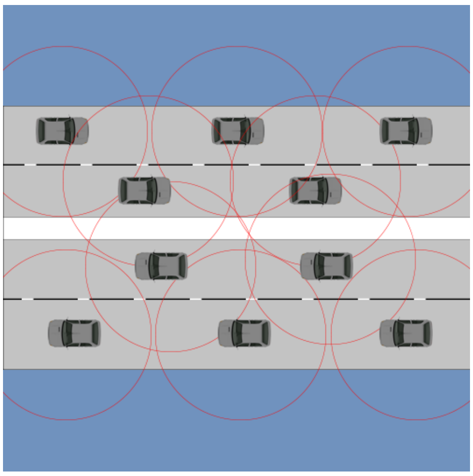

Figure 2.6: Broadcast Storm Problem

- **Network Partition problem:** It occurs when a number of vehicles in an area of interest is not enough to spread the data across a relatively close group of vehicles. This problem is frequent in low densities of urban or highway scenery where the delivery of messages cannot be done easily. [43]. Figure 2.7 shows an example of Network Partition problem in vehicular networks.

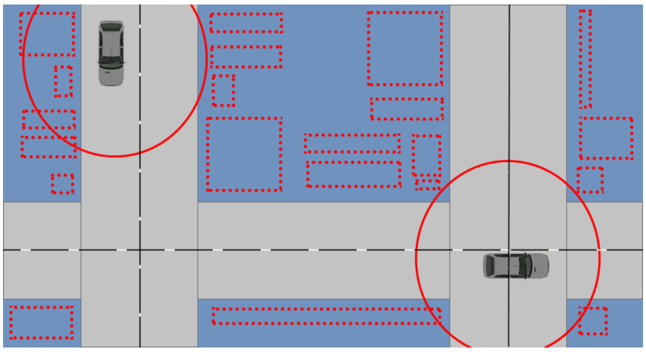

Figure 2.7: Network Partition problem

- **Temporal Network Fragmentation problem:** In the temporary network fragmentation problem, the network partition is temporary and short-lived. This problem occurs due to the high mobility of these networks, due to the loss of packets and collisions in the mac layer. This problem is frequent when there is a high dynamism in the vehicular network, that is, constant changes in vehicular densities. [43]

2.5 Advantages of vehicle-to-vehicle (V2V) communications

V2V-type vehicular communications have many advantages over normal wireless connections, especially in broadcasting emergency messages. These communications have a low cost of maintenance and implementation compared to other technologies such as WiMax, universal mobile telecommunications system (UMTS) and long-term evolution (LTE). The main advantage is that V2V connections do not require a service provider or infrastructure. According to Table 2.2, vehicles can communicate directly with single hop messages, without the need for intermediate bases. V2V is efficient for emergency notifications, data distribution over time thanks to its high transmission speed and low delay. Even V2V could work in places where another type of wireless connection would not have access and would fail in its service [44]. However, there are factors that can affect these vehicular connections (such as topology that can change dynamically, or intermittent connectivity). V2V communications can be a complement to other types of network architectures in local areas of interest.

Properties		Approaches		
		V2V	**V2I**	**UMTS-LTE**
Communication latency		Low	Medium	High
Link availability		Medium to High	Low	Low to Medium
Data rate		High	Medium to High	Low to Medium
System availability	Local	High	High	High
	Global	High	Low	Medium
Cost issue	Initial	Medium to High	High	High
	Operational	Null to low	Medium	Medium to High
Communication service area		Small to medium	Medium	Large
Exploit geographic relevance of data		Yes	No	No
Support for traffic safety applications		High	Low to Medium	Low

Table 2.2: Different applications for wireless communications [2]

2.6 Assumptions and requirements

In this work, we mainly consider V2V communications. In this context, we assume that each vehicle is equipped with on-board wireless devices in compliance with the available IEEE 802.11p standard. Besides, the proposed scheme assumes that every vehicle is able to permanently determine its current geographical position using Global Positioning System (GPS) or any other localization service. Furthermore, we assume either the presence of a local application running on the source vehicle or the presence of a fixed infrastructure (Access point, Road-side Unit, etc.) responsible for data message generation [45]. All generated messages need to be disseminated within an area of interest.

Chapter 3

Theoretical Framework

In this chapter, we present the simulation framework used for the design of the dissemination protocol. In addition, each component of the framework used is detailed to achieve a simulation of the vehicular network as realistic as possible.

3.1 VEINS

Veins [46] (Vehicular In Network Simulation Framework) is a simulation framework focused on vehicular network communication developed as open source software. This program is composed of 2 simulators: OMNeT ++, a network simulator that works with events and SUMO, simulates constantly changing and dynamic traffic. Both simulators are connected via a TCP socket. These programs work in co-simulation to recreate the environment of a real vehicular network without sacrificing speed. For this reason, it is possible to model the influence of vehicle networks on established traffic and examine the complex interactions between both domains of the simulators. Figure 3.1 shows Veins program on OMNeT++ IDE running on Linux. The SUMO and OMNeT ++ GUI and IDE can be used to quickly configure and interactively run simulations. Veins offers a complete set of models for the simulation of inter-vehicular networks. This base allows other vehicular networks to be implemented with its structure, being friendly with the user and easy to learn. In this work, Veins was used as the basis for modeling our protocol and the other protocols that were used for performance analysis. The veins version used for this project is 5.0

3.2 OMNeT++

OMNeT++ [47] is a C++-based discrete event simulator for modeling communication networks, parallel systems, or multiprocessors. OMNeT++ is open-source simulation tool, and can be used under Academic Public License for research-oriented and educational proposes. Figure 3.1 shows OMNeT++ IDE. OMNeT ++ application areas are specific and supported by simulation models such as mobility or INET. These models are developed completely independently of OMNeT ++ and follow their own release cycles. OMNeT ++ has a command line and graphical and interactive user interfaces which allows analyzing the events in real time. Simulation results are written into output vector and output scalar files. Result files are text-based, so it can be processed by R. The OMNeT++ version used for this project is 5.6.1.

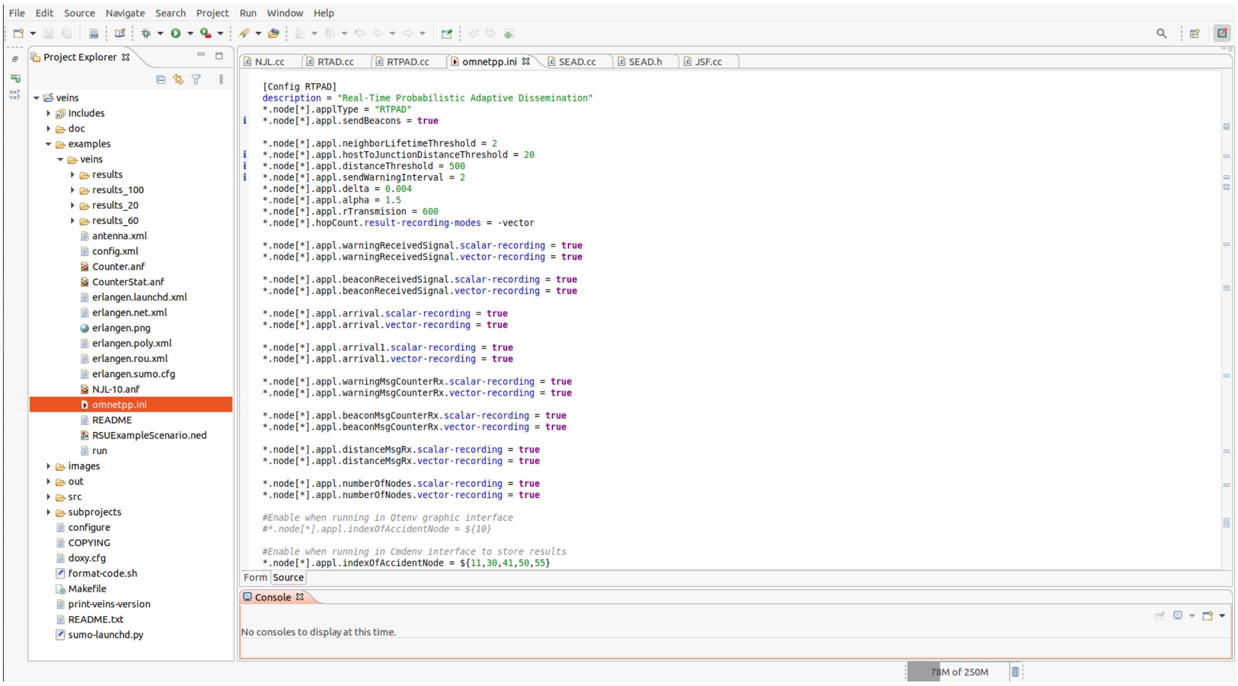

Figure 3.1: Veins simulation framework on OMNeT++ IDE simulator

3.3 Simulation of Urban Mobility-SUMO

SUMO [48] is a portable microscope traffic simulator developed to manage vehicular networks. SUMO contains a wide variety of tools and functions for simulating large networks. It simulates with a realistic environment, in which moving vehicles, parked and also pedestrians can be nodes of the network. Each vehicle is explicitly based on its model, has its own route, and moves individually through the specified network. Figure 3.2 shows an example of SUMO, VEINS, and OMNeT++ running a simulation of RTPAD diffusion behavior.

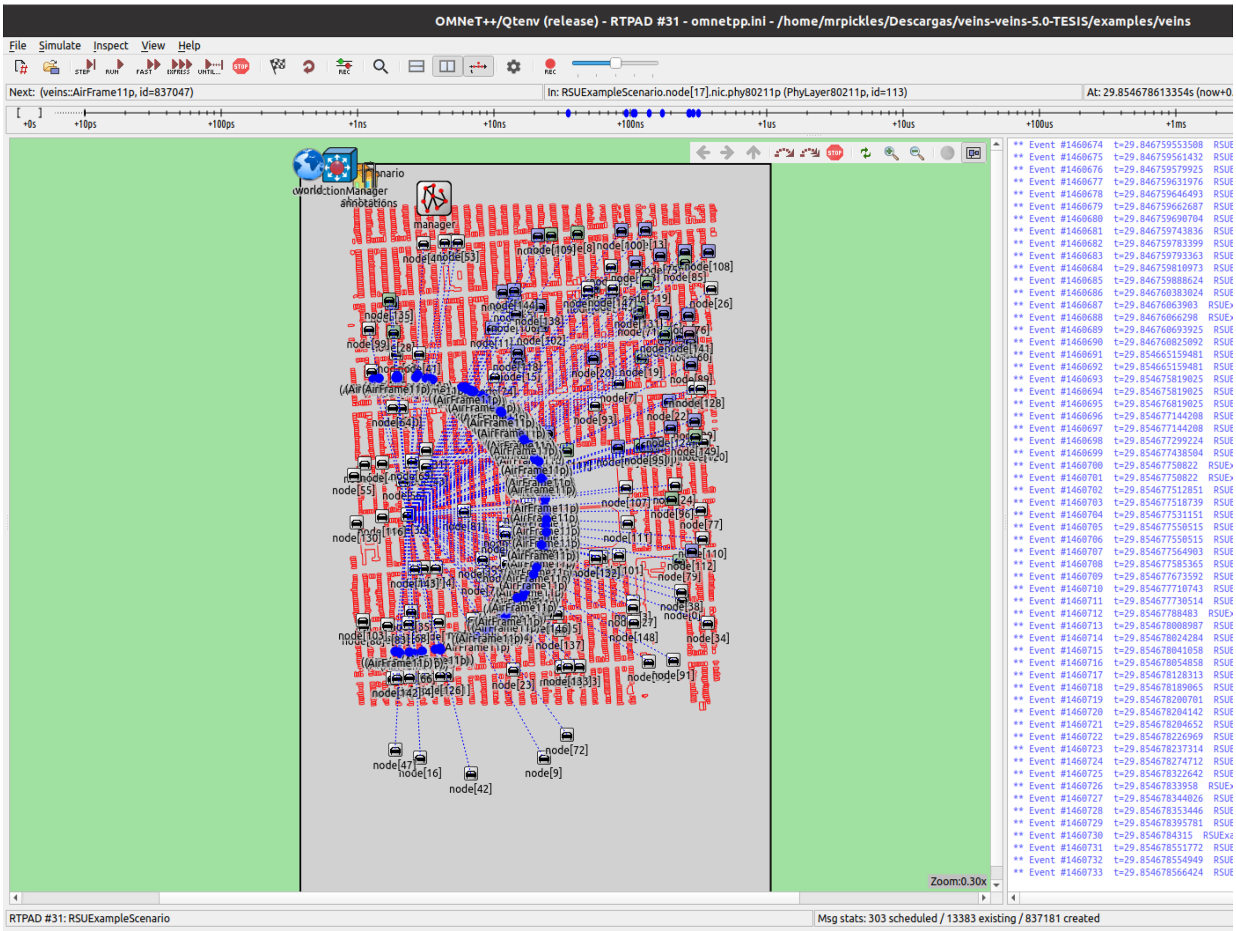

Figure 3.2: SUMO Interface running our protocol RTPAD, with OMNeT++ and VEINS

3.4 OpenStreetMap

Also known as OSM, OpenStreetMap [49] generates a certain network of roads and obstacles. This system offers a graphical interface, which allows you to select a specific area and obtain all that information in a file with an .osm extension. Figure 3.3 shows the OpenStreetMap online platform. With the help of SUMO, we convert this file containing all the necessary information into the files that VEINS needs for the simulation. With this tool we obtained the map in which the simulations of all the protocols used in this investigation were made. OpenStreetMap is free, and has its online platform.

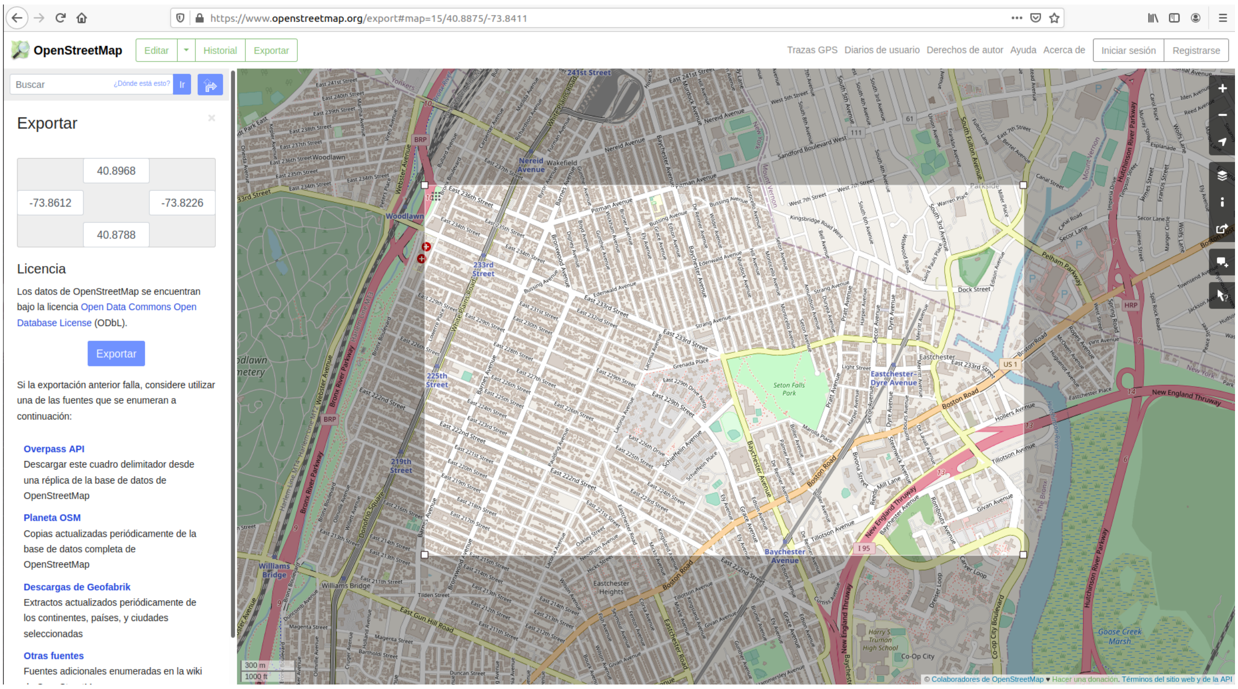

Figure 3.3: OpenStreetMap online platform

3.5 R

R [50] is a statistical computing language and environment and also a complete set of software installations for data analysis and manipulation, calculation and graphical display. It is designed as a programming language and allows its users to add functionality and operations using the definition of new functions. Advanced users can write C code to manipulate R objects directly. We used R in this project to process the data obtained as results of the simulations, and to graph all the metrics used for analysis. Figure 3.4 shows R-Studio IDE running on Linux. R version used for this project is 4.0.0.

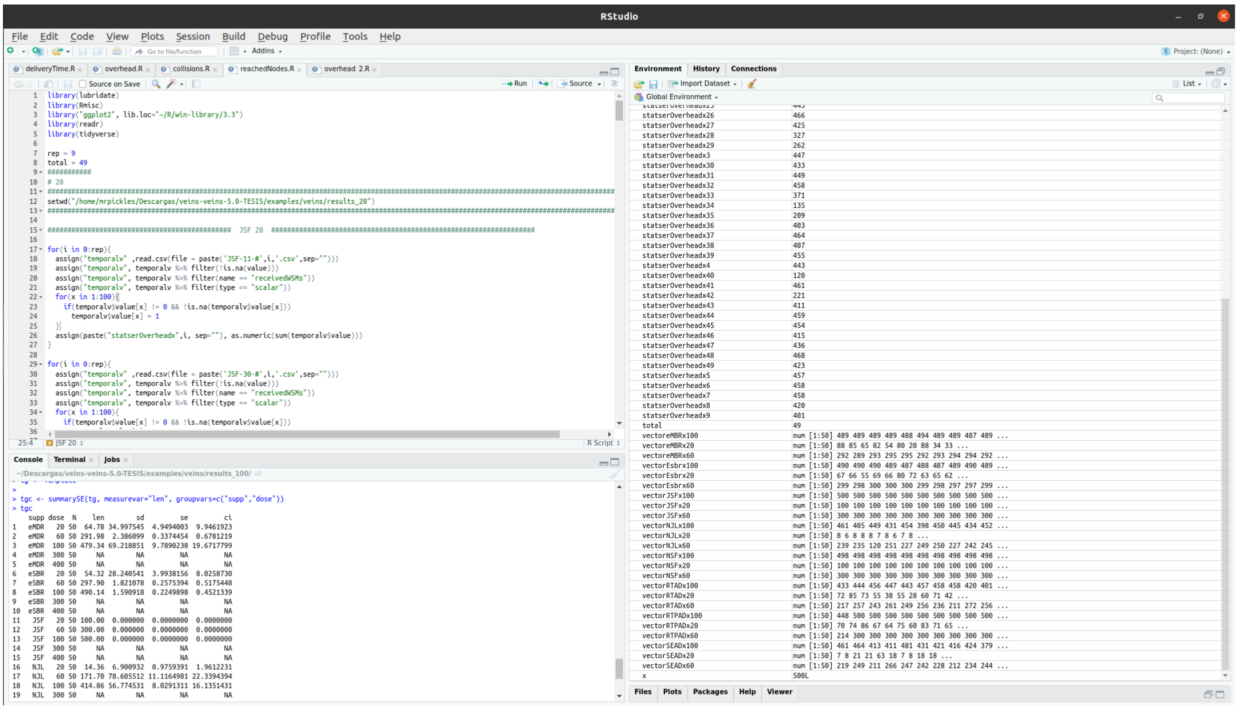

Figure 3.4: R-Studio IDE

Chapter 4

State of the Art

This chapter presents a review of the main data dissemination protocols in vehicle networks. In vehicle networks, alert messages are used to immediately disseminate information on the network. Each vehicle can transmit a message to its neighbors when it is in an alert situation. When this message is received by another vehicle, it will have the possibility of being re-transmitted in the form of multiple hops to inform as many vehicles as possible. This broadcast service is offered by the broadcast mechanisms of the IEEE 802.11p protocol. In this work, we focus on schemes based on multiple hops. These schemes use message re-transmission probabilities or metrics to schedule a message forwarding, and techniques such as Store and Forward to resolve disconnected network problems. We have selected some data dissemination protocols that have presented the best performance in the different evaluations made in the current literature.

4.1 Junction Store and Forward Protocol (JSF)

This data dissemination protocol for vehicle networks decides whether to relay the message based on the topology of the map in which the vehicle is located. The message received by the vehicle is stored, and forwarded in an optimal situation. The Junction Store and Forward (JSF) [51] scheme is based on the fact that vehicles located near intersections are more likely to reach other new vehicles that are in the re-transmission range, since at that point the connection with other streets is achieved. The methodology with which the JSF scheme works is summarized in the flow diagram shown in Figure 4.1. This dissemination protocol requires that each vehicle generate and update its list of neighbors periodically. This list is generated thanks to one-hop messages called beacons, which contain the information of the neighboring vehicle such as its position, speed, etc. Upon receiving a new message, the receiving vehicle ensures that there are new neighbors in the area and that the vehicle is also located at an intersection in order to relay the message. Thanks to the GPS system, it can be determined if the vehicle is at a junction. A timer is also used to discard obsolete messages. This process can be repeated an infinite number of times (N). That is why the value N can be determined according to a counter, which allows each vehicle to re-transmit that number of times a message. In this project we allow JSF to re-transmit any times as possible in the simulation time.

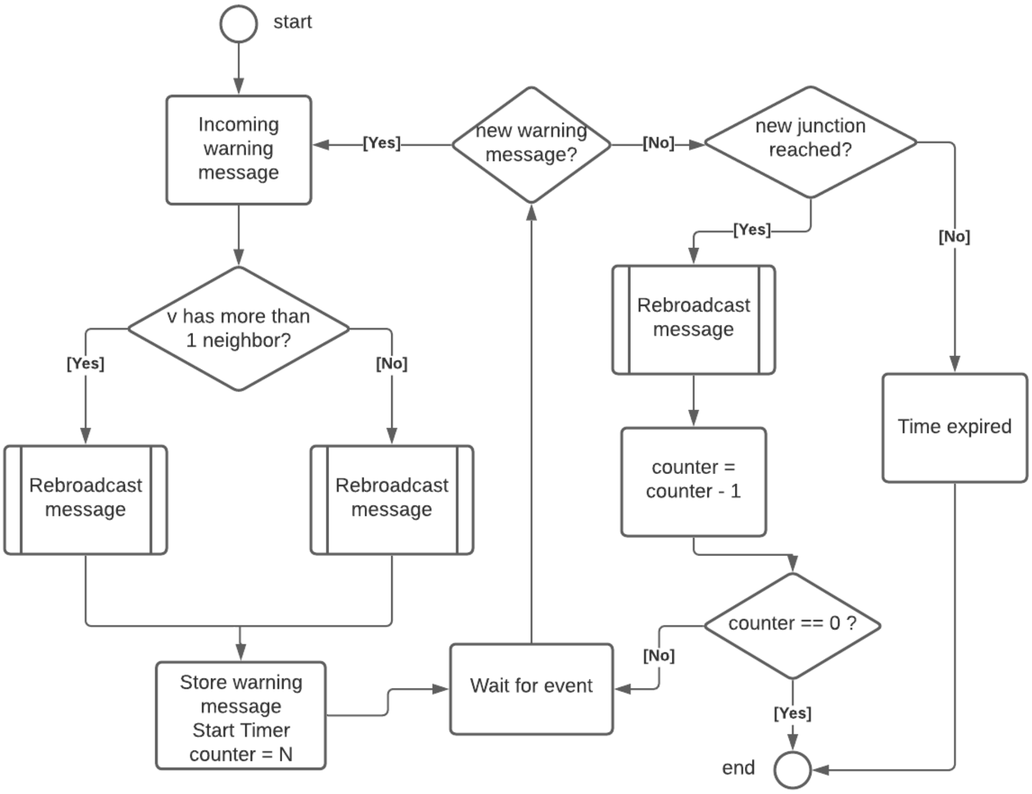

Figure 4.1: JSF dissemination scheme working flowchart

4.2 Neighbor Store and Forward Protocol (NSF)

This protocol attempts to improve the performance of the Store Carry and Forward technique to troubleshoot fragmented networks. The Neighbor Store and Forward (NSF) scheme [51] is very similar to JSF, with the minimal difference that instead of all vehicles at an intersection re-transmitting, only the vehicle closest to the intersection re-transmits. This scheme requires a list of neighbors that is updated periodically through one-hop messages. Once a vehicle is found that is not in the updated neighbor list, the stored message is resent. In this way the protocol focuses on informing new vehicles as soon as they arrive in the area, and not informing new areas of the topology as JSF does. Figure 4.2 shows the flow chart of the NSF protocol.

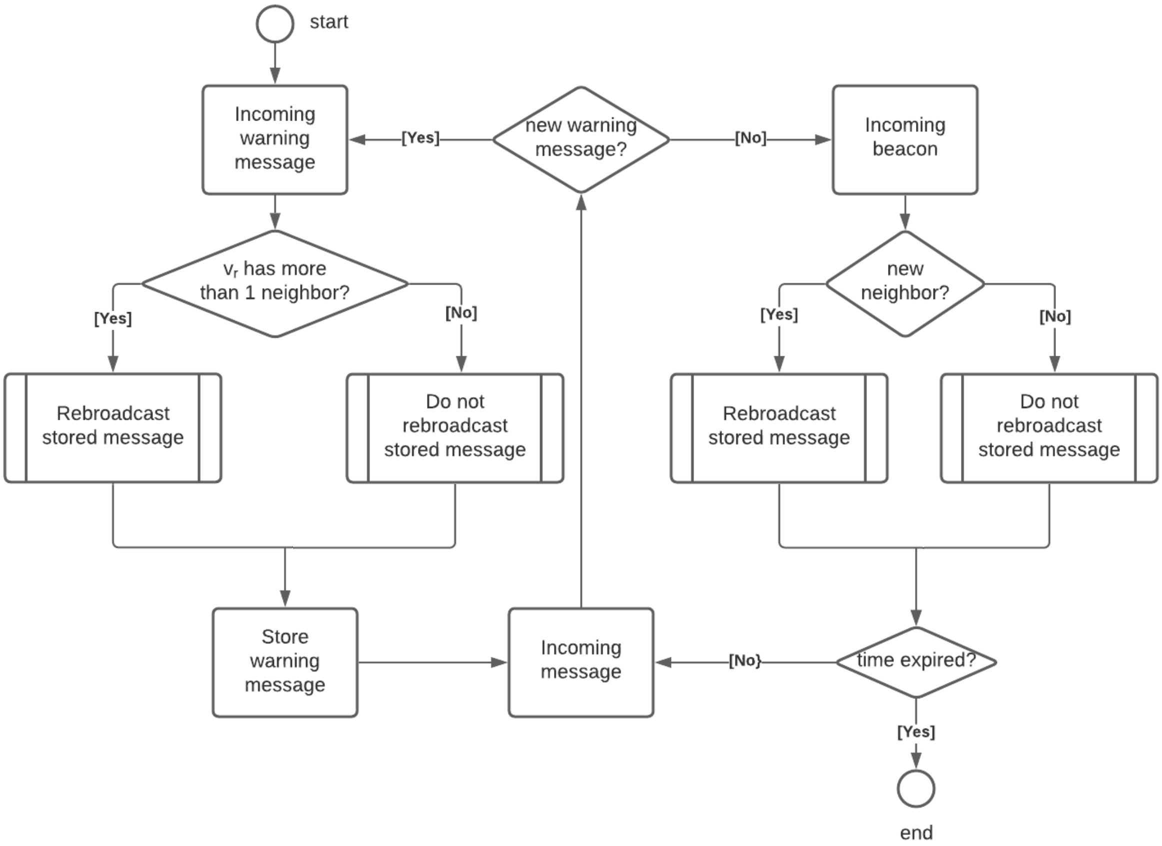

Figure 4.2: NSF dissemination scheme working flowchart

4.3 The enhanced Street Broadcast Reduction (eSBR)

The data dissemination protocol named Enhanced Street Broadcast Reduction (eSBR) [52] obtains information from the map on which it is applied. With the help of GPS it also provides an optimal dissemination of warning messages from vehicle networks. There are two conditions that the receiving vehicle can meet to relay a message: the vehicle is on another street to the sending vehicle, or the vehicle is at a distance or greater than a predetermined value. Thanks to these conditions, the message can travel most of the area of the zone of interest. The information of the map is very important for this protocol since to overcome the blind areas generated by obstacles.

4.4 The enhanced Message Dissemination for Road-maps (eMDR)

The data dissemination protocol Enhanced Message Diffusion for Road-maps (eMDR) [53] is very similar to the eSBR protocol. The main difference between these protocols is that eMDR seeks to reduce the number of messages produced by re transmitting the same alert message multiple times. This protocol also uses the information of the map in which it is located, such as streets and junctions. To fulfill its objective, the vehicle receiving the message only re-transmit if it is in a junction and is the closest to the center of the same. Figure 4.3 shows the flow chart of the eMDR protocol. This protocol has shown that it is capable of reducing the number of re-transmissions without reducing the rate of vehicles that receive the message.

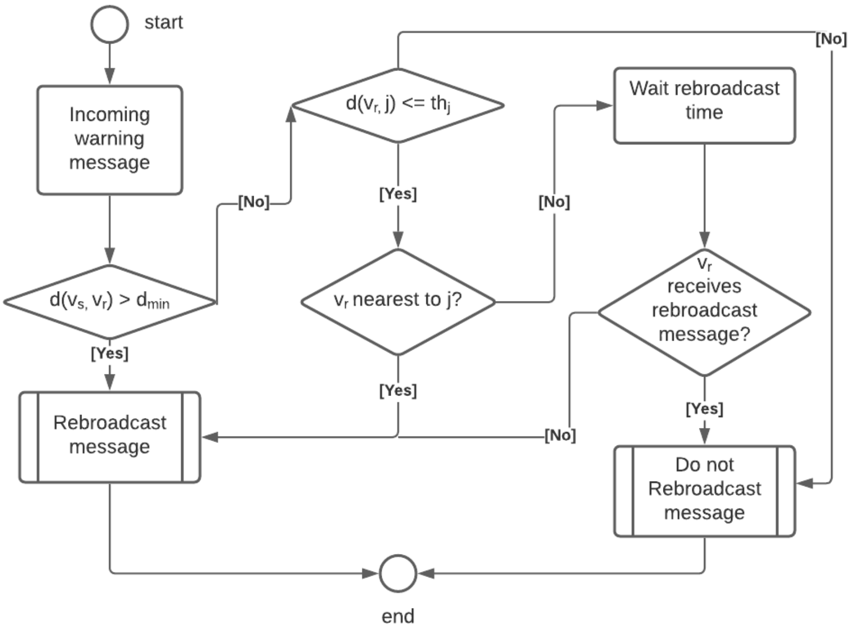

Figure 4.3: eMDR dissemination scheme working flowchart

4.5 Nearest Junction Located (NJL)

The diffusion scheme called Nearest Junction Located (NJL) [3] is based on the topology of the map where the vehicles are located, using the integrated positioning system to define the vehicles closest to a junction. This scheme also needs a list of neighbors that is updated periodically to know the position of the other neighbors at the intersection. Its operation is described below. When a vehicle receives an alert message, it checks if it is the closest receiver to any road junction. This protocol is not optimal for low-density or sparse network scenarios, because it is very restrictive. NJL performs efficiently in high traffic density scenarios. Figure 4.4 shows the flow chart of the NJL dissemination protocol.

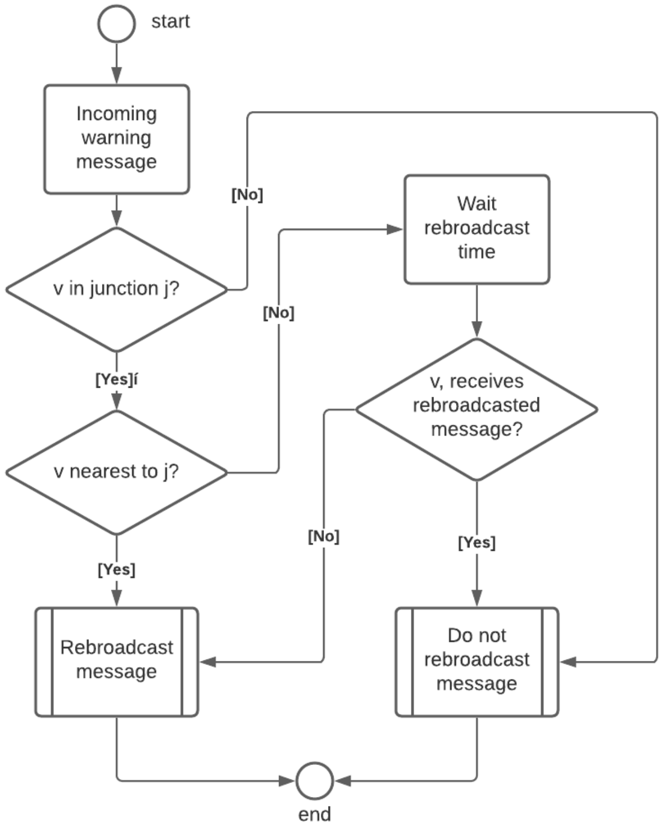

Figure 4.4: NJL dissemination scheme working flowchart

4.6 Simple and Efficient Adaptive Dissemination Protocol (SEAD)

The data dissemination protocol for vehicular networks SEAD [54] is a hybrid protocol that combines delay and forwarding probability. The main objective of this scheme is to be as simple as it is efficient for data dissemination. It is simple because the design of this protocol does not use single hop messages, that is beacons, which takes into account the complexity of the scheme. It is also efficient, since it reduces broadcast storms with the probability of message forwarding. The proportion of messages forwarded is considered according to a value of the ”alpha” protocol, which is a variable that directly affects the probability of message forwarding. Figure 4.5 represents the flow chart of the SEAD protocol. This protocol has 2 verification, the first is when the vehicle receives the message and it is checked if the message is new; When the novelty of the message is fulfilled, it is copied into the vehicle's buffer. If this is not new, it should be discarded after updating the redundancy radius. If the receiving vehicle is on another street or behind the sending vehicle, this vehicle is assigned for scheduled for rebroadcast. Each message assigned for rebroadcast has a waiting time which is calculated in the process, taking into account the distance of the sending and receiving vehicle. When the timeout expires, the message is resent if the forwarding probability is met. This probability value is permanently updated, depending on how many messages each particular vehicle receives.

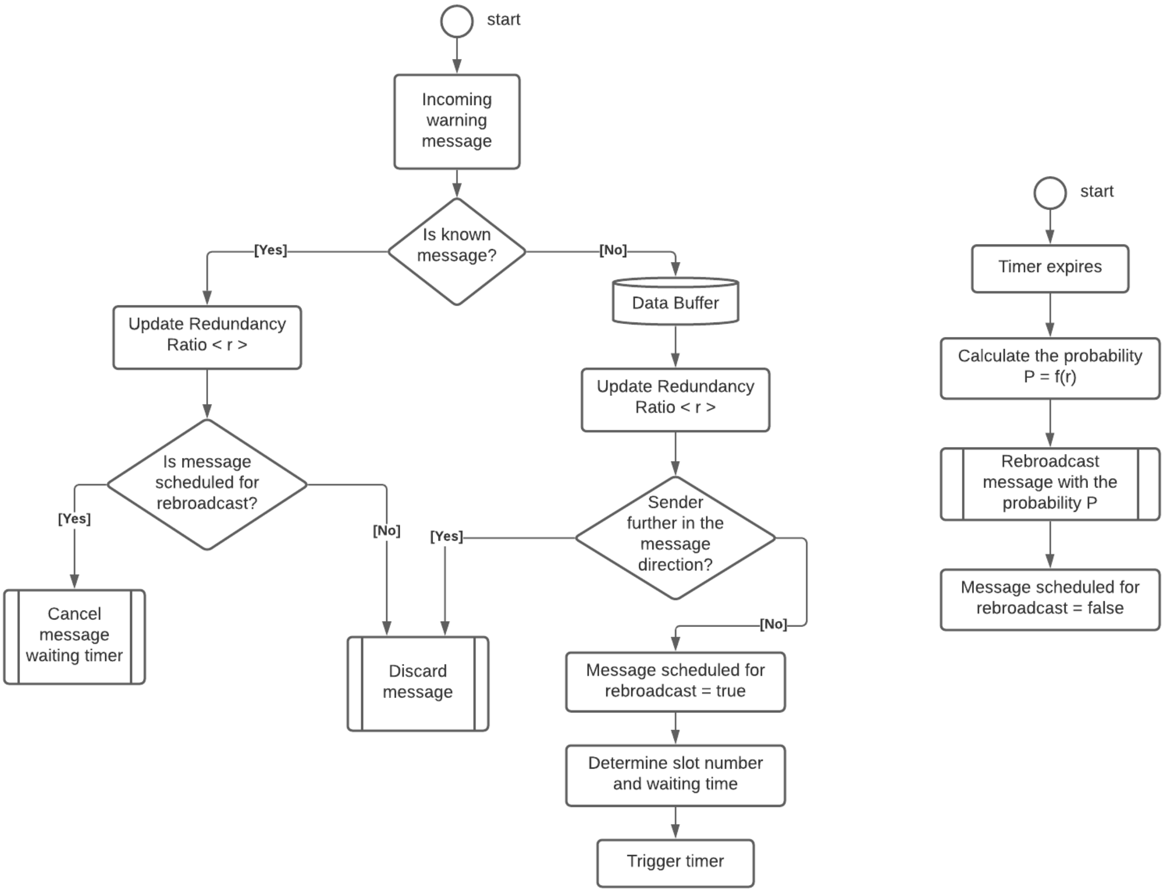

Figure 4.5: SEAD dissemination scheme working flowchart

4.7 Real-Time Adaptive Dissemination Protocol (RTAD)

The main feature of RTAD [3] protocol is for each vehicle to dynamically adopt a specific dissemination scheme according to the scenario in which it is moving. Adaptive protocols have been a very important advance in the dissemination of data in vehicle networks, and it is the fundamental reason why our proposal has been inspired by this dissemination scheme. Figure 4.6 shows the general process of how RTAD has been developed. It begins with an analysis of the performance of a group of dissemination protocols to obtain those with the best performance and then the process of choosing one by one the most appropriate scheme for each specific situation. RTAD decides which scheme to use in a particular way based on two parameters: its SJ ratio and the estimated density of the vehicle. The SJ ratio is the number of streets divided by the number of junctions on the map of interest. Vehicle density is calculated using a calculated linear regression. According to the authors, this regression is reliable for any type of map and vehicle density [3].

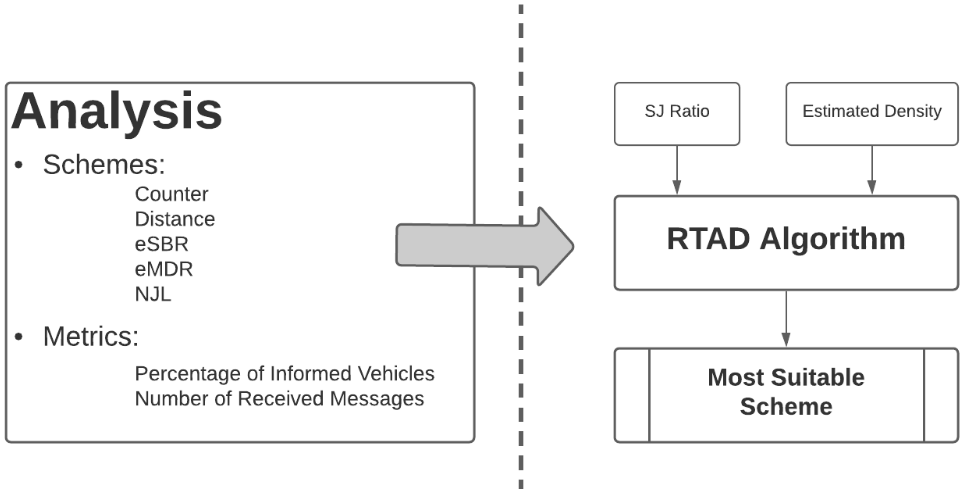

Figure 4.6: RTAD dissemination scheme working flowchart

Chapter 5

Methodology

This chapter explains in detail the proposed method for developing our proposal. It begins with a overview of our dissemination protocol. Furthermore, this chapter describes the performance metrics used to evaluate the performance of our protocol in comparison with other protocols presented in the literature. Finally, the description of the conducted experiments of this research is presented; each description contains the objective of the experiment and its general explanation.

5.1 Real-Time Probabilistic Adaptive Dissemination Protocol (RTPAD)

In the current literature there are several dissemination schemes that perform well in high, medium or low traffic flow scenarios. We have made an exhaustive selection of these protocols and have designed an adaptive mechanism that chooses the scheme that best adapts to the scenario in real time. Our proposal is a new adaptive system that allows each vehicle to take its appropriate diffusion scheme to optimize the use of the network and reach more informed vehicles in the shortest possible time. This scheme uses input parameters such as the topology of the map in which it is located and the list of neighbors that is updated periodically, this to approximate a vehicular density.

Real-Time Probabilistic Adaptive Dissemination Protocol-RTPAD is a real-time data dissemination system for vehicle networks, which indicates to each vehicle which dissemination scheme appropriate to use for its specific situation. Its main advantage is that thanks to its real-time operation it obtains optimal results by mitigating the main problems in the dissemination of data in VANETs, which are broadcast storms and disconnected networks. Our proposal is an extension of the RTAD [3] data dissemination protocol. Our dissemination system uses a neighbor table that is periodically updated thanks to single hop beacon messages. This table of neighbors allows to approximate a partial density in each vehicle, using a linear regression formula. Also, our protocol uses the topology characteristics of the map in which it is located, that is, the number of streets and junctions that exist in the area of interest. Unlike RTAD, our proposal includes the implementation of a store and carry mechanism to increase the message delivery rate in the case of low densities and a forwarding probability value to decrease the amount of messages re-transmitted at high densities.

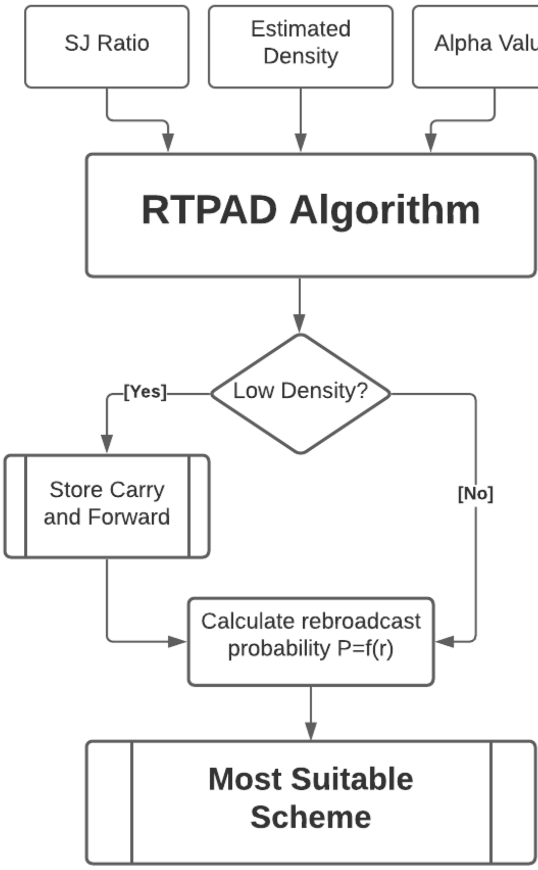

Figure 5.1: RTPAD dissemination scheme working flowchart

Figure 5.1 presents the flow chart of RTPAD protocol. It takes as parameter the SJ Ratio, traffic estimated density, α value, and uses them to do the RTPAD algorithm. In this algorithm the protocol chooses the most suitable scheme based on their particular environment. Then it sees if the estimated density is low to assume the use of the store carry and forward. After, the protocol calculates the rebroadcast probability to take the decision of rebroadcast or no the current message. As an output we have the most appropriate scheme and the protocol will make the decisions already defined before. These decisions are whether to do the store carry and forward mechanism or whether or not to forward the message.

5.1.1 Broadcasting Technique

To fully explain the algorithm of our proposal, it is necessary to clarify the metrics it uses.

- **Density estimation function:**
  Our algorithm needs to estimate the vehicular density of each particular vehicle in

order to make the decision as to which dissemination scheme to use. Therefore, it is necessary to make an accurate estimate of the density to obtain optimal results. Sanguesa, et al. [3] proposes a method that estimates vehicle density based on a linear regression formula at any instant of time, that is, it will be calculated periodically.

$$f(\rho, s) = a_1 + a_2\rho + a_3 s + a_4\rho^2 + a_5 s^2 + a_6\rho^3 + a_7 s^3 + a_8\rho s + a_9\rho^2 s + a_{10}\rho s^2 \quad (5.1)$$

Eq. 5.1 shows the polynomial formula that estimates the density in number of vehicles per km^2 according to how many neighbors it has and the value of SJ Ratio which will be explained next. The equation receives as parameters a ρ that the number of neighbors it has and s the value of SJ Ratio. Table 5.1 presents the polynomial coefficients values of the equation.

Coefficient	**Value**
a_1	$-7.9174488563075374E+02$
a_2	$-6.5986335423139231E-01$
a_3	$2.2717144481059472E+03$
a_4	$1.1989021509173876E+00$
a_5	$-2.1019985100197723E+03$
a_6	$-1.7509026843351649E-02$
a_7	$6.3096785945693671E+02$
a_8	$-4.8107314165096247E+00$
a_9	$-7.6438506962308739E-01$
a_{10}	$1.4601116345732333E+01$

Table 5.1: Density estimation equation coefficients [3] .

- **SJ Ratio:**
  SJ Ratio is based on the topology of the map, it is the relation between number of streets and junctions in a specific zone. An SJ Ratio greater than 1 indicates that the topology of the zone is complex, which means that there are more streets than junctions.

  $$SJ(s, j) = \frac{s}{j} \quad (5.2)$$

  Eq. 5.2 shows the equation of the relation SJ Ratio proposed by RTAD [3]. This equation takes as parameters s the number of streets and j the number of junctions on the road map.

- **Redundancy Metric:**
  Redundancy Ratio "r" is the relation between the number of received messages per original messages. This metric is proposed by SEAD [54].

  $$r = \frac{Total\ Received\ Messages(original + duplicated)}{Total\ New\ Messages(original)} \quad (5.3)$$

Eq. 5.3 shows the equation of the Redundancy Ratio. RTPAD allows to update this metric periodically, when the vehicle is receiving messages.

- **α value:**
This parameter is proposed by SEAD [54] and it allows us to fix a definite amount of redundancy, which is enough to guaranty a certain ratio of packet delivery. In this paper we set $\alpha = 2$.

- **Re-broadcast Probability P:**
This metric is used to define if an received warning message would be re-transmitted, and it was proposed by SEAD [54].

$$P_i = \frac{2\alpha}{r_i} * P_{prev} = \frac{2\alpha}{r_0} * \frac{2\alpha}{r_{prev}} = \frac{2\alpha}{r_0} * \frac{2\alpha}{r_1} ... \frac{2\alpha}{r_{i-1}} * \frac{2\alpha}{r_i} = \frac{(2\alpha)^{i+1}}{\prod_{k=0}^{i} r_k} \tag{5.4}$$

As inputs needs the current redundancy ratio value (r_i) of the vehicle at time i and the previous one (r_{prev}) incorporated in the precious calculated probability (P_{prev}) for the last sent packet. This means that the probability is the product of all previous probabilities used. In this way, the possibility of carrying out a forwarding is inversely proportional to the redundancy radius, and also, inversely proportional to the vehicular density.

Algorithm 1 details how RTPAD works. First, it uses the updated neighbor table of the vehicle receiving the message and uses it depending on the SJ ratio to calculate the estimated vehicle density. SJ ratio is a parameter calculated in the same way in each vehicle thanks to its GPS system to obtain the characteristics of the map on which it is mobilized in real time.

Thanks to parameters defined in the RTAD protocol, it has been estimated that a density greater than 125 vehicles per square kilometer is considered high and that is why we must use a protocol assigned for said density. In the same way, a map with an SJ ratio greater than 0.9 is considered a medium complexity map, therefore, it will have more obstacles to broadcast efficiently because it does not have so many junctions compared to its streets. Every time a vehicle receives a beacon or a hop message, it performs this process again, so that it updates its convenience protocol periodically. Finally, the result of this algorithm is the most suitable protocol for dissemination according to its position and the position of its neighbors in real time. The computational process of this system is very low, so that the processing system of each vehicle will have no problems in carrying it out.

5.2 Performance evaluation

In this section, we explain how this dissemination protocol is evaluated through simulations. Also, we detail how the simulations were configured, how the protocols were implemented in the different scenarios. Finally, a review of how the results were obtained.

Algorithm 1: RTPAD implementation

Require: $D - density(vehicles/km^2)$.

Require: $SJR - SJRatio$

Require: $\alpha - AlphaValue$

Ensure: $Suitable_{bcast}$ - most suitable scheme

α value is used to calculate rebroadcast probability

if (D > 175) OR (D > 125 AND SJR < 1.1) OR (D > 75 AND SJ < 1.05) **then**

  return NJL;

else if (SJR > 1.3 AND D < 175) **then**

  return eSBR;

else

  return eMDR;

end if

5.2.1 Simulation setup

A simulation of vehicular networks requires that the protocol under study be evaluated in a scenario that is as realistic as possible. So we have chosen VEINS as the simulation framework. With this framework, we fulfill the requirement that the simulation be as realistic. Furthermore, to guarantee the reliability of the results presented, a total of 50 simulations have been carried out, 10 simulations for each point for each broadcast protocol and assigning 5 different vehicles that emit the alert message. With this, we reach results with a standard error of 5 % in a 95 % confidence interval. The Media Access Control (MAC) layer model uses a data rate of 6Mb/s, and is based on IEEE 802.11p with a transmit power of 20mW and the receiver has a sensitivity of -89dB. In the table 5.2, we can see the simulation parameters in the scenarios used. Beacon messages have a frequency of 1Hz in all simulation scenarios. This frequency is the highest that is expected in the transmission of these messages, having a freshness of beacon messages of a hop. SUMO helps us to make all the vehicles move randomly on the map. Annex 1 in the appendices you can find the codes developed to simulate the RTPAD protocol. There are four codes programmed in c, such as RTPAD.cc, RTPAD.h, RTPAD.ned, and the omnetpp.ini file.

Parameter	**Specifications**
Network simulator	Veins
Mobility trace generator	SUMO
Simulation duration	100 s
Vehicles' Density	20 to 100 Vehicles/km
Data packet frequency	1 Hz
Data packet size	500 bytes
Number of source vehicles	5
Propagation model	Nakagami
Phy/Mac protocol	IEEE 802.11p
Bit rate	6 Mbit/s
Transmission range	$\sim$ 700 m
δ	4 ms
Max speed	20 m/s
Number of run	10
α	2

Table 5.2: Simulation Parameters

5.2.2 Scenario description

We simulate on a map of the city of San Francisco, with an area of interest of 5 km^2. Most streets on the map are two-way, and there are a variety of obstacles. In Figure 5.2 you can see the map that we use in the SUMO graphical interface. This map extracted from OpenStreetMap has a Manhattan-style and the buildings are represented as red rectangles. We rely on an accident generated by a random vehicle at 10 seconds of the simulation to see the behavior of the vehicle network, and how this message is disseminated throughout the area of interest. The information that this alert message contains is the location in coordinates of the source vehicle, as well as the speed at which it is going, and other variables of interest.

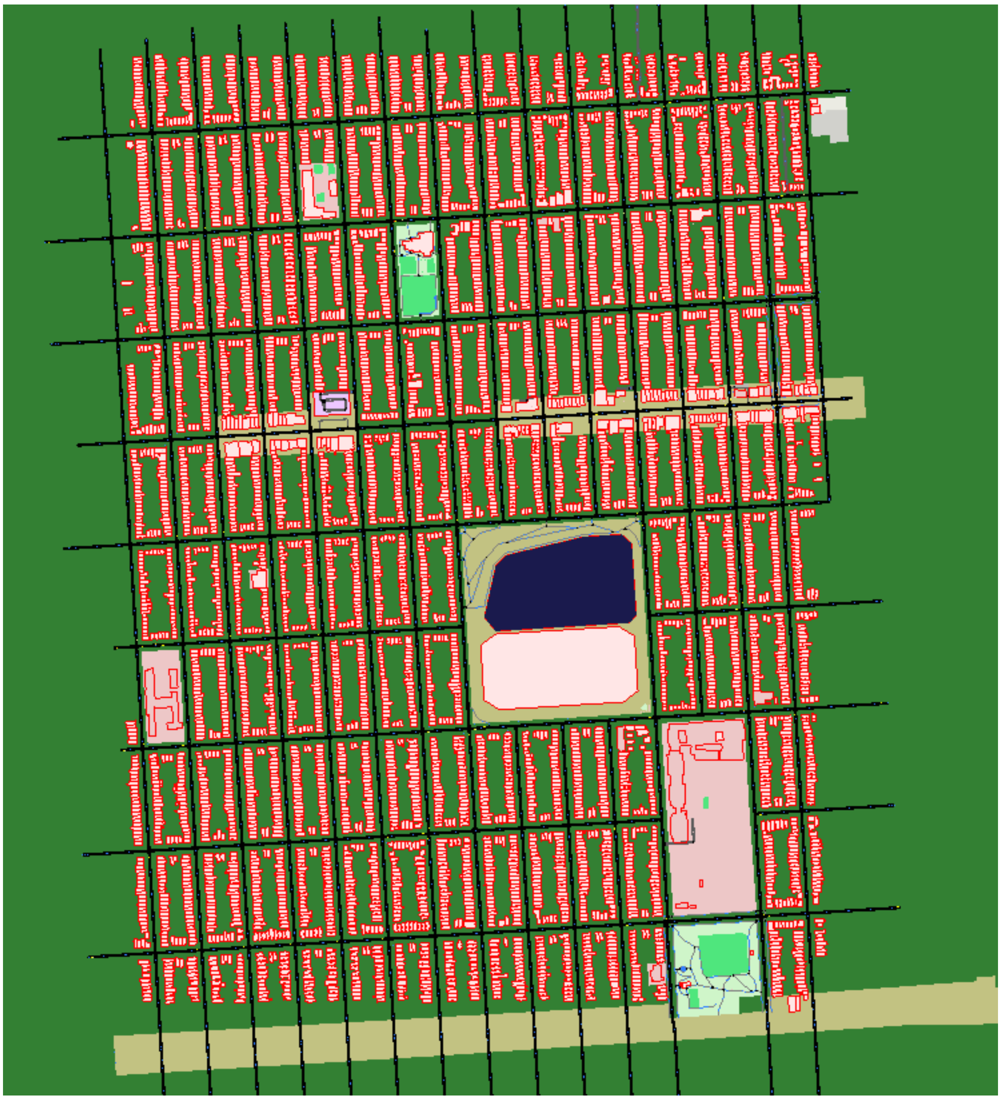

Figure 5.2: Vehicular network scenario in SUMO-gui. San Francisco city

Chapter 6

Results and Discussion

This chapter shows the results obtained from our simulations. The evaluation aims to measure the performance the diffusion capacity of RTPAD in a realistic vehicular network scenary using QoS metrics. To evaluate our proposal, in each repetition of the simulation we use different scenarios.

6.1 Evaluation Metrics

The assessment of the performances of our protocol is carried out through three metrics, namely Percent of Informed Vehicles, Messages Received per Vehicle and Collisions at the mac layer, which are detailed below.

6.1.1 Percent of Informed Vehicles

The percent of informed vehicles is the ratio between the number of vehicles that received the message and the total number of vehicles on the network at the time of the evaluation. The most important objective in the dissemination of data in vehicle networks is to achieve the greatest number of informed vehicles in a short time. This metric can be affected by different factors such as message size, node group size, coverage range and node mobility. A robust dissemination of messages occurs when the percentage of vehicles reported reaches a value of 100%.

6.1.2 Messages received per vehicle

Another important metric in the data dissemination process in vehicle networks is the number of messages generated and re-transmitted during the dissemination process. A high re-transmission of messages is the main cause of collisions and contention of communication channels. For this reason, it is important to reduce the number of message re-transmissions without affecting the percentage of reported vehicles so that the communication channel works properly and other applications can use it.

6.1.3 Collisions

Collisions are the total number of packet collisions on the MAC layer. They are generated by the great number of messages generated in the network. This can be due to high densities in vehicle networks or due to the characteristics of the dissemination protocol [19].

6.2 Simulation Results

To analyze the performance of our proposal with other dissemination protocols, we have segmented into two groups, named as static and adaptive dissemination protocols. Due to their design characteristics, the JSF, NSF, NJL, eMDR and eSBR protocols are static protocols. Likewise, the RTAD, RTPAD and SEAD protocols, due to their characteristics of adaptation to the environment, are identified as adaptive protocols [54].

6.2.1 RTPAD vs static dissemination schemes

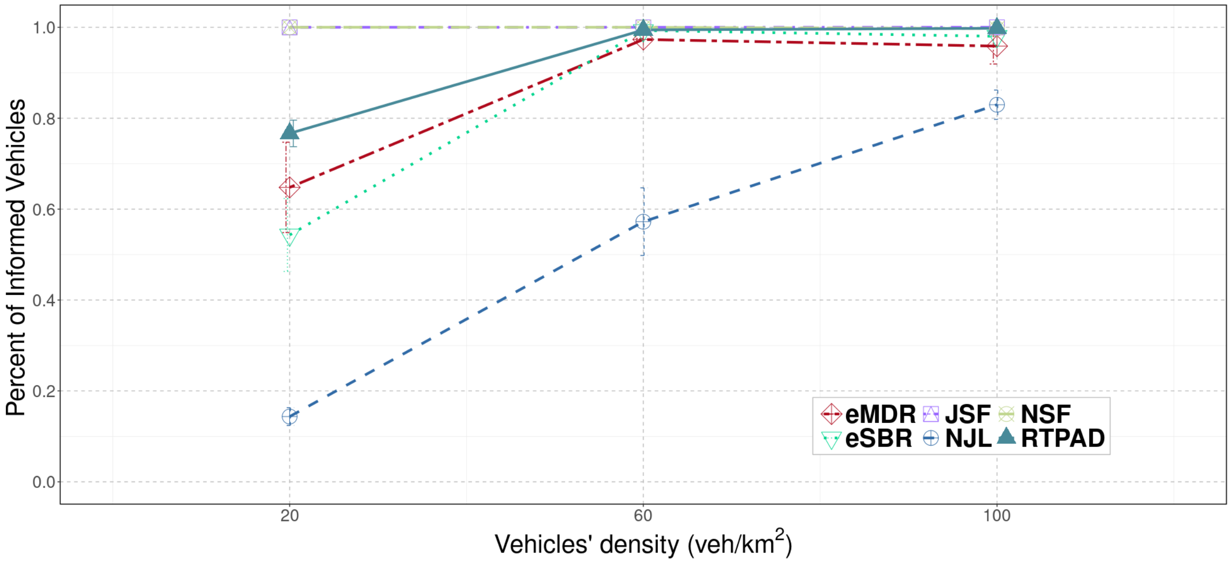

Figure 6.1: Percentage of informed vehicles in San Francisco for 20, 60 and 100 $vehicles/km^2$

Figure 6.1 shows the percentage of vehicles informed. In particular, we include the percentage of informed vehicles for the map of San Francisco when simulating three different densities: 20, 60 and 100$vehicles/km^2$. Analyzing these data, we can affirm that at low densities, the protocols that reach the maximum percentage of diffusion are JSF and NSF protocols, followed by our RTPAD protocol with an approximate of 78% of informed vehicles. Our protocol performs better than eMDR, eSBR and NJL protocols in low densities due to the implementation of the store carry and forward mechanism, which works only in low densities and solves the problem of disconnected and partially disconnected networks. At an intermediate density (that is, 60 $vehicles/km^2$), we observe that the trend of reported vehicles remains at the maximum for the NSF and JSF protocols. Our RTPAD protocol reaches the maximum percentage of reported vehicles in the area of interest, while the eSBR and eMDR protocols already reach an average delivery of 95% of reported vehicles. On the other hand, NJL protocol is reported with 60%, which is very low. Finally, when reaching the $100 vehicles/km^2$, we observe that the eMDR and eSBR protocols have reached, on average, 100% of reported vehicles. NJL protocol reaches a 82% of reported vehicles in the area of interest. As shown, the RTPAD dissemination scheme achieves the highest percentage of vehicles informed under low, medium and high vehicle density conditions.

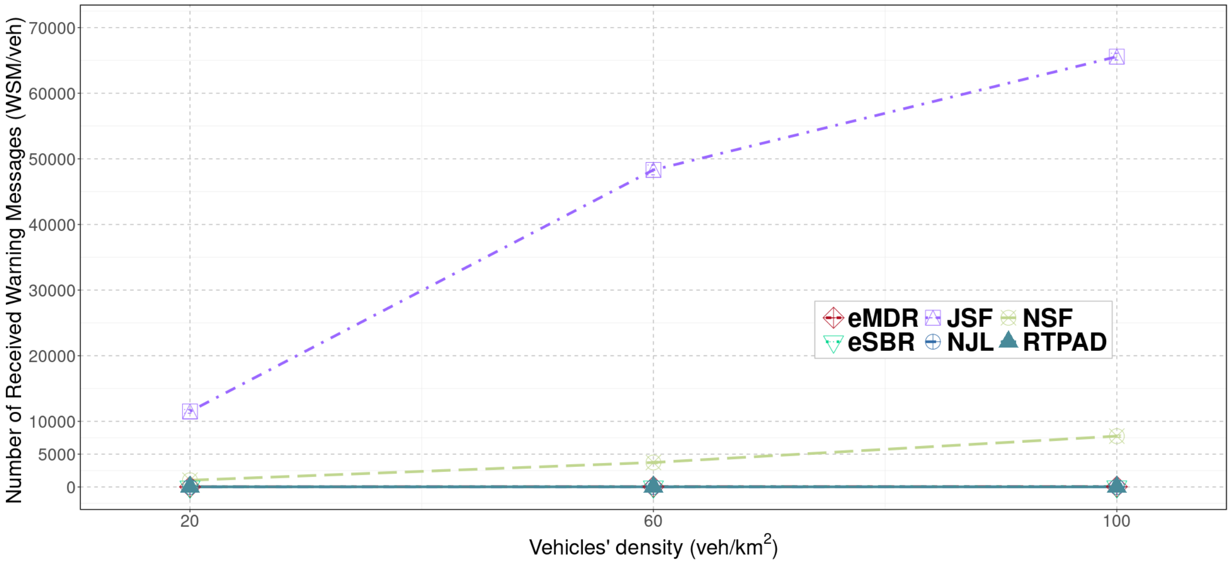

Figure 6.2: Number of messages received per vehicle for static dissemination schemes in big scale

Regarding the messages received per vehicle (see Figures 6.2 and 6.3), we could interpret that there is a correlation between the number of messages received and the percentage of vehicles reported. However, in Figure 6.2, we can see how the messages received per vehicle grow exponentially as the vehicle density increases for JSF and NSF protocols, reaching a maximum of 65k and 7k messages respectively. This is an excessive use of the communication channel, generating the well-known broadcast storms caused by multiple collisions at the mac layer level. In Figure 6.3, we observe a optimal use of the communication channel by the eSBR, eMDR, NJL and RTPAD protocols. In general, it is observed that in all protocols grow steadily but moderately. However, RTPAD has an almost constant level of received messages without affecting the number of vehicles reported. Specifically, it is observed that the maximum average number of messages received per vehicle at low densities is 22, which is acceptable for the capacity of the communication channel. This is thanks to the forwarding probability system that was implemented, which allows reducing the re-transmission rate and solving the problem of broadcast storms allowing other applications to make use of the channel while transmitting information. On the other hand, eMDR, eSBR and NJL protocols have a very low re-transmission rate which is consistent with the results of the number of vehicles reported since they did not reach an optimal percentage of diffusion.

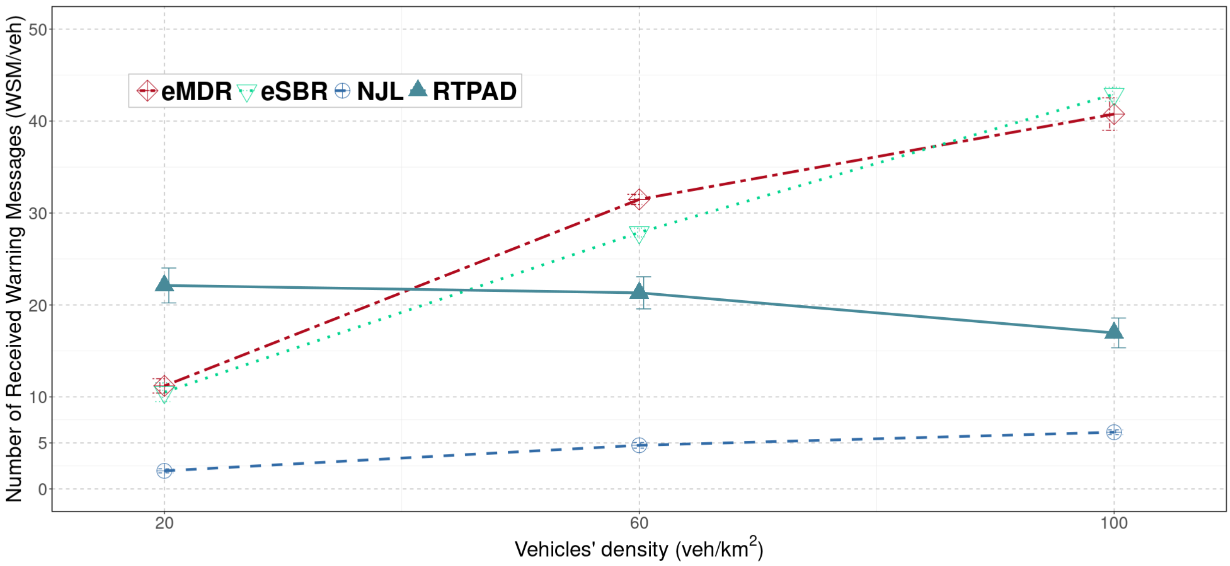

Figure 6.3: Number of messages received per vehicle for static dissemination schemes in small scale

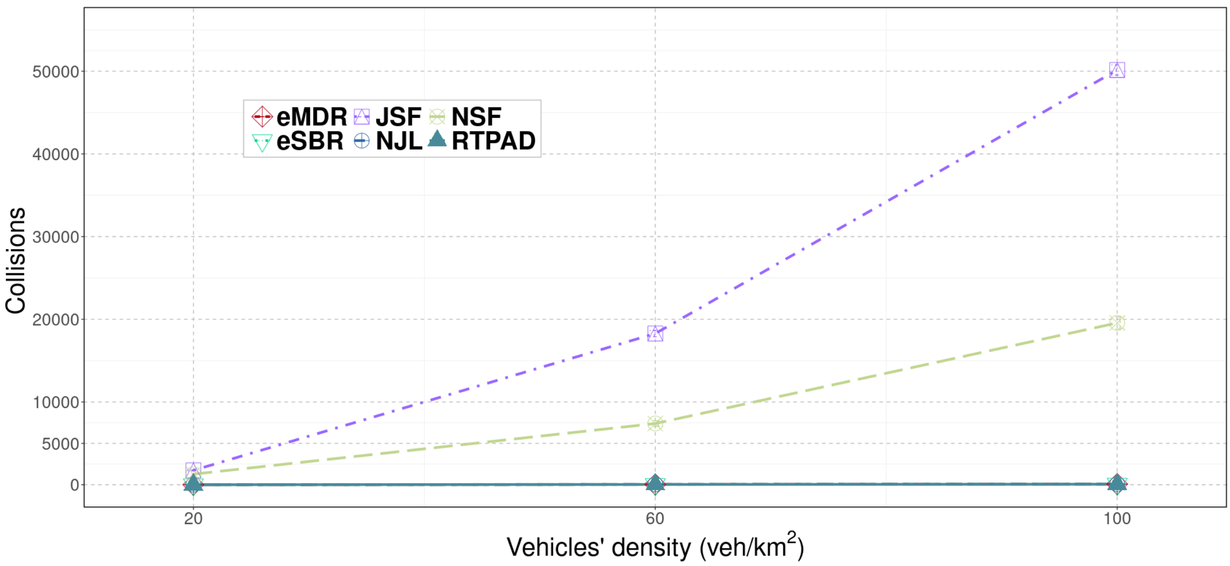

Figure 6.4: Number of collisions in MAC layer for static dissemination schemes in big scale

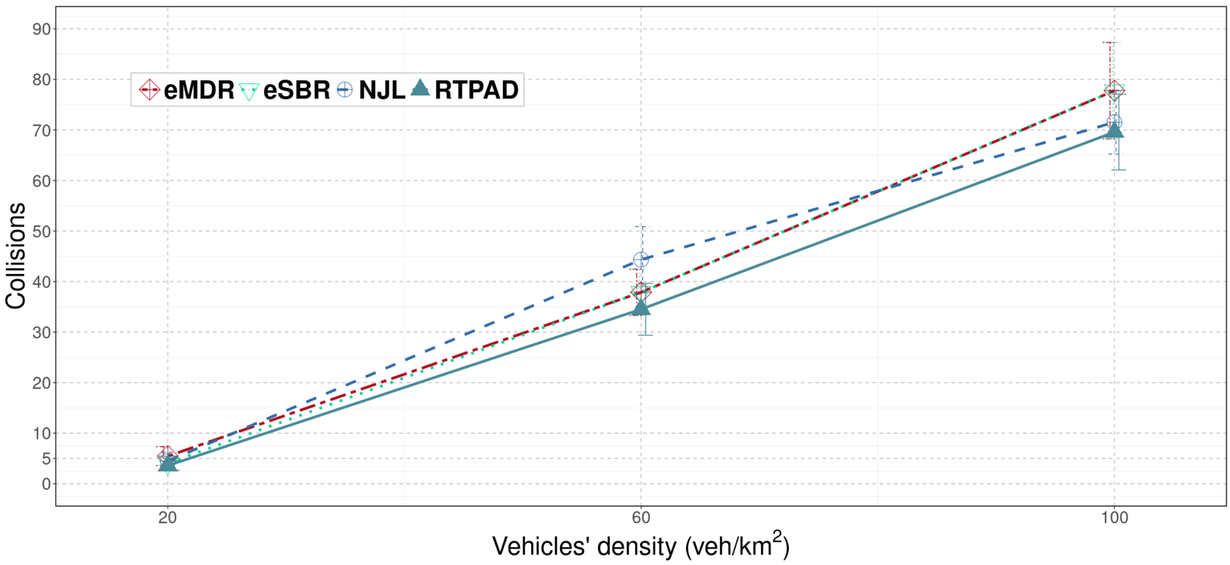

Figure 6.5: Number of collisions in MAC layer for static dissemination schemes in small scale

Finally, it is important to analyze the collisions in the mac layer since one of the main objectives of a dissemination protocol is to optimize the use of the channel so that the service does not stop, and other applications can make use of the channel. In Figure 6.4, we can see the number of collisions generated in the mac layer while varying the vehicular density. It is notable that JSF and NSF protocols have a huge difference with the other static dissemination schemes. In fact, it is observed that while the vehicular density increases, the collisions also grow exponentially for the JSF and NSF protocols with a very high value of 50k and 20k respectively. This excessive amount of collisions have as a consequence, the collapse of the channel and the suspension of the communication service. So, these protocols are not recommended for data dissemination at intermediate and high vehicle densities. In Figure 6.5, we can find collisions in the mac layer but on a smaller scale to be able to analyze them in a better way. In Figure 6.5, it is observed that all the protocols have a similar growth in collisions as vehicle density increases. Reaching a maximum of 80 collisions in the mac layer at the highest density. It is important to emphasize that RTPAD has the lowest collision average. Thus, RTPAD is an optimal protocol for the dissemination of data without the danger of the collapse of the communications channel.

6.2.2 RTPAD vs dynamic dissemination schemes

SEAD and RTAD are dynamic dissemination protocols for vehicular networks. This kind of protocols can change their mode of operation, understanding their environment and responding optimally to a change in real time. In other words, these protocols can improve the transmission process or reduce channel contention. Our proposal, RTPAD protocol is an example of this kind of schemes.

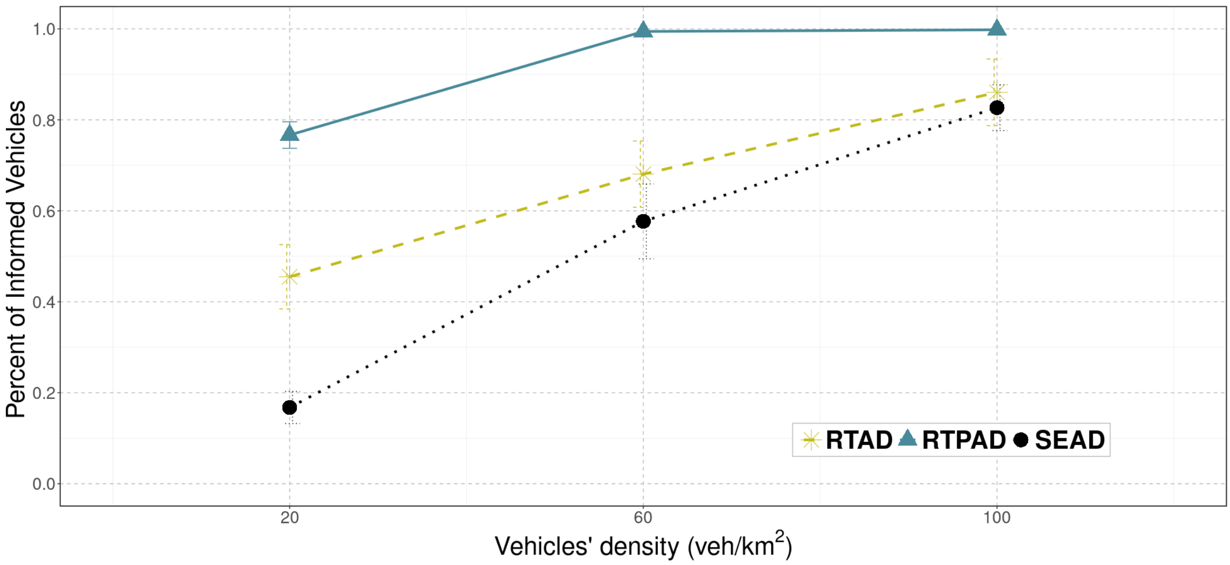

Figure 6.6: Percentage of informed vehicles for dynamic dissemination schemes

Figure 6.6 shows the results of the percentage of vehicles reported while varying the vehicle density. We can observe that at low densities, RTPAD protocol has a much higher performance in the diffusion of the message. Specifically, RTPAD reports 78% of vehicles reported in the area of interest at 20 vehicles per km^2. This result is thanks to the implementation of the store carry and forward mechanism which solves the problem of disconnected networks. On the other hand, the RTAD and SEAD protocols have a low rate of reported vehicles with values of 46% and 19%, respectively. These results are very poor for an information dissemination protocol. As the vehicle density increases, the rate of reported vehicles also grows. RTPAD protocol already reaches the 100% of vehicles reported in the network. However, the other protocols get a maximum of 68% on average of reported vehicles. In the highest vehicular density ($100 vehicles/km^2$), it is observed that the trend of vehicles reported with RTPAD protocol remains at the maximum, but the other protocols get a maximum of 82% on average of vehicles reported. In short, we can say that RTPAD has better results when optimizing the diffusion process and reaching a greater number of informed vehicles in the area of interest. It achieves this thanks to its store carry and forward system in low densities and its characteristic of adapting to the best dissemination system according to its environment.

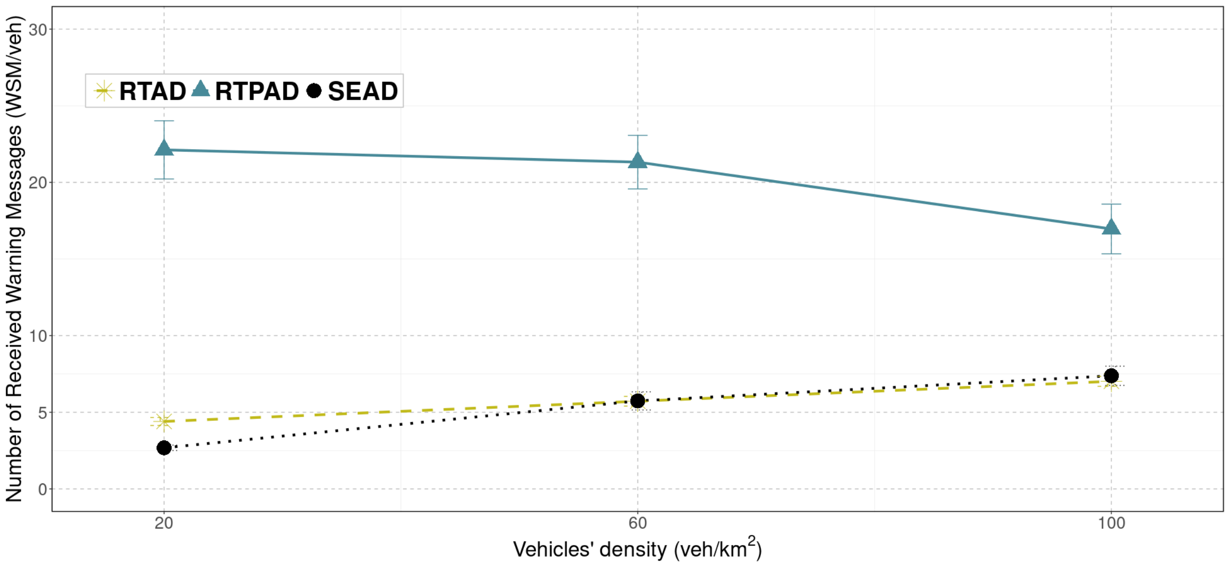

Figure 6.7: Number of messages received per vehicle for dynamic dissemination schemes

Figure 6.7 shows results of the number of messages received per vehicle while varying the traffic density. Here, it can be seen in a general way that these adaptive protocols are more optimal for the correct use of the communication channel. The number of received warning messages do not pass of 22 messages received per vehicle. This amount is very low compared to the static protocols that we already reviewed in the previous section. Specifically, we can observe that at low densities RTAD and SEAD protocols have 5 messages received per vehicle. This value is consistent with their rate of informed vehicles at low densities. RTPAD protocol presents 22 messages received per vehicle. However, this value is in the optimal range of channel use. As traffic density increases, so does the number of messages received per vehicle for the SEAD and RTAD protocols. Notice how RTPAD decreases the number of messages received per vehicle as its density increases. This is mainly due to the system forwarding probability that optimizes the use of the communication channel, mitigating the presence of a broadcast storm. At the highest densities, RTPAD decreases the number of messages received as vehicle density increases, thus showing it as an optimal protocol for its use in very high vehicle densities, without affecting the number of vehicles reported.

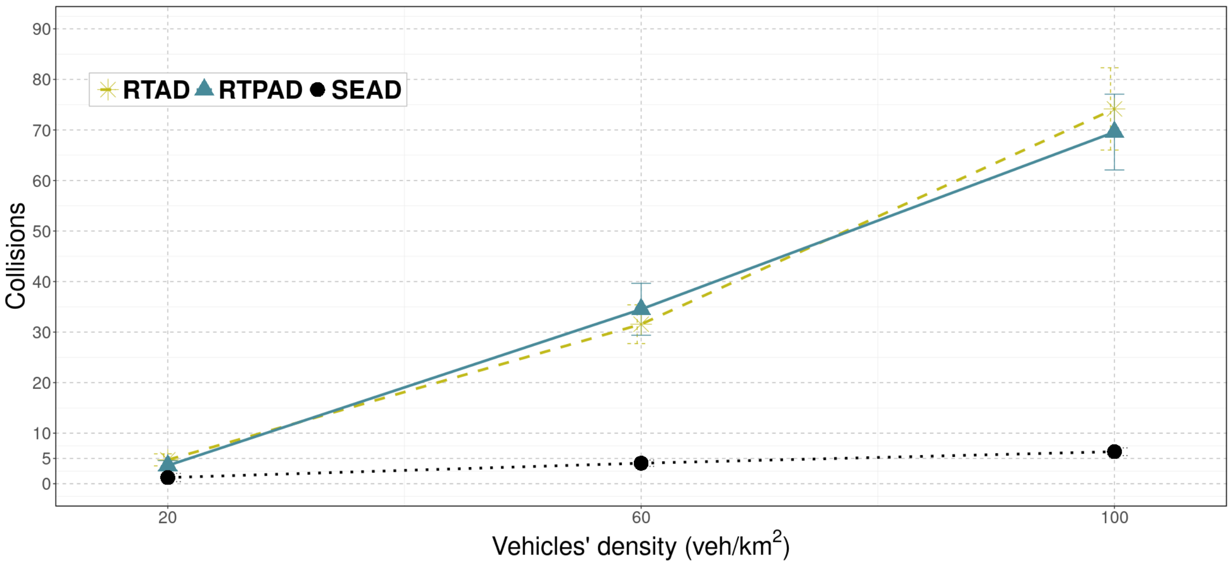

Figure 6.8: Number of collisions in MAC layer for dynamic dissemination schemes

Finally, the behavior of collisions in dynamic dissemination schemes can be observed while the vehicular density is varied. As the vehicle density increases, the number of collisions in the mac layer also increases simultaneously. This behavior is noticeable in all protocols, but in the SEAD protocol we observe that the number of collisions is very low despite the increase in vehicle density. This is due to the longer waiting time that is calculated to re-transmit a message, and the channel is optimized thanks to this waiting time. We note that RTPAD in high densities reaches a maximum of 70 collisions on average, which is an acceptable number for the amount of information that can be transmitted on the channel that is handled. In addition, it is clear that RTPAD is an efficient and optimal protocol for any situation that may arise in vehicle networks. In this analysis that we carry out of the results obtained, our protocol has fulfilled all the expectations that were raised.

Chapter 7

Conclusions

7.1 Conclusions

Throughout this thesis, four main conclusions have been made:

1. We reviewed and implemented seven data dissemination protocols that were developed based on their topology or in their geographical location. The selected schemes were JSF[55], NSF[55], eSBR[52], eMDR[53], SEAD[54], NJL[55], and RTAD[3]. All these dissemination schemes were evaluated based on the metrics such as informed vehicles and collisions in the MAC-layer. According to the evaluation, we selected eSBR, eMDR, and NJL as the best schemes for low, medium, and high vehicular density respectively.

2. We have proposed our real-time probabilistic adaptive diffusion protocol (RTPAD). RTPAD is a data dissemination protocol that takes as inputs the traffic estimated density, number of streets and junctions, and the α value to choose the most suitable scheme based on the RTAD algorithm [3]. Also, RTPAD presents the store carry and forward mechanisms and the probability of re-transmission.

3. Our RTPAD proposal has excellent results in the message diffusion process in all situations compared to the different static and dynamic dissemination schemes. RTPAD is efficient due to the comparison of the informed vehicles with other protocols while the vehicular density is growing. Also, our dissemination scheme is optimal because it maintains low use of the communication channel without affecting the number of informed vehicles.

4. In this research we have studied existing protocols for dissemination of emergence messages in vehicular ad-hoc networks. Also, we have examined several protocols for dissemination of emergence messages, and we have designed a novel protocol to improve the overall performance in different scenarios. Moreover, we have provided a simulation framework to simulate realistic vehicular scenarios.

Bibliography

[1] M. Faezipour, M. Nourani, A. Saeed, and S. Addepalli, "Progress and challenges in intelligent vehicle area networks," *Communications of the ACM*, vol. 55, pp. 90–100, 02 2012.

[2] F. Rezaei, K. Naik, A. Nayak, and s. Yousefi, "Effective warning data dissemination scheme in vehicular networks for intelligent transportation system applications," 10 2013, pp. 1071–1076.

[3] J. Sanguesa, M. Fogue, P. Garrido, F. Martinez, J.-C. Cano, C. Calafate, and P. Manzoni, "Rtad: a real-time adaptive dissemination system for vanets," *Computer Communications*, vol. 60, 02 2015.

[4] B. Tian, K. M. Hou, and J. Li, "Trad: Traffic adaptive data dissemination protocol for both urban and highway vanets," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016, pp. 724–731.

[5] M. D. Rakesh Kumar, "A Review of Various VANET Data Dissemination Protocols," *International Journal of u- and e- Service, Science and Technology*, vol. 5, pp. 27–44, 2012. [Online]. Available: https://www.earticle.net/Article/A208750

[6] T. Kosch, C. Adler, S. Eichler, C. Schroth, and M. Strassberger, "The scalability problem of vehicular ad hoc networks and how to solve it," *Wireless Communications, IEEE*, vol. 13, pp. 22 – 28, 11 2006.

[7] M. Chaqfeh, A. Lakas, and I. Jawhar, "A survey on data dissemination in vehicular ad hoc networks," *Vehicular Communications*, vol. 1, no. 4, pp. 214–225, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209614000448

[8] W. Chen, R. K. Guha, T. J. Kwon, J. Lee, and Y.-Y. Hsu, "A survey and challenges in routing and data dissemination in vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 787–795, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.862

[9] R. Oliveira, C. Montez, A. Boukerche, and M. S. Wangham, "Reliable data dissemination protocol for vanet traffic safety applications," *Ad Hoc Networks*, vol. 63, pp. 30–44, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870517300835

[10] N. Cenerario, T. Delot, and S. Ilarri, “A content-based dissemination protocol for vanets: Exploiting the encounter probability,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 771–782, 2011.

[11] C. Iza-Paredes, A. M. Mezher, M. Aguilar Igartua, and J. Forné, “Game-theoretical design of an adaptive distributed dissemination protocol for vanets,” *Sensors*, vol. 18, no. 1, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/1/294

[12] L. A. Villas, A. Boukerche, G. Maia, R. W. Pazzi, and A. A. Loureiro, “Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks,” *Computer Networks*, vol. 75, pp. 381–394, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128614003673

[13] F. Naeimipoor and A. Boukerche, “A hybrid video dissemination protocol for vanets,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 112–117.

[14] R. S. Schwartz, R. R. R. Barbosa, N. Meratnia, G. Heijenk, and H. Scholten, “A directional data dissemination protocol for vehicular environments,” *Computer Communications*, vol. 34, no. 17, pp. 2057–2071, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366411001010

[15] A. Soua and H. Afifi, “Adaptive data collection protocol using reinforcement learning for vanets,” in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2013, pp. 1040–1045.

[16] A. Mahajan, N. Potnis, K. Gopalan, and A.-I. Wang, “Urban mobility models for vanets,” 07 2010.

[17] GPS, http://www.gps.gov/systems/gps.

[18] E. Schoch, F. Kargl, M. Weber, and T. Leinmuller, “Communication patterns in vanets,” *IEEE Communications Magazine*, vol. 46, no. 11, pp. 119–125, 2008.

[19] B. K. Chaurasia, R. S. Tomar, S. Verma, and G. S. Tomar, “Suitability of manet routing protocols for vehicular ad hoc networks,” in *2012 International Conference on Communication Systems and Network Technologies*, 2012, pp. 334–338.

[20] E. Spaho, M. Ikeda, L. Barolli, F. Xhafa, M. Younas, and M. Takizawa, “Performance evaluation of olsr and aodv protocols in a vanet crossroad scenario,” in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, 2013, pp. 577–582.

[21] W. Viriyasitavat, O. K. Tonguz, and F. Bai, “Network connectivity of vanets in urban areas,” in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1–9.

[22] S. Yousefi, E. Altman, and R. El-Azouzi, “Study of connectivity in vehicular ad hoc networks,” in *2007 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops*, 2007, pp. 1–6.

[23] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.1a: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpan)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–700, 2005.

[24] I. C. S. L. S. Committee, I. M. Theory, T. Society, I. of Electrical, E. Engineers, and I.-S. S. Board, *IEEE Standard for WirelessMAN-Advanced Air Interface for Broadband Wireless Access Systems: Higher reliability networks. Amendment 2*, ser. IEEE Std. IEEE, 2013. [Online]. Available: https://books.google.com.ec/books?id=jqsWngEACAAJ

[25] "Approved ieee draft amendment to ieee standard for information technology-telecommunications and information exchange between systems-part 15.4:wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans): Amendment to add alternate phy (amendment of ieee std 802.15.4)," *IEEE Approved Std P802.15.4a/D7, Jan 2007*, 2007.

[26] M. S. Al-kahtani, "Survey on security attacks in vehicular ad hoc networks (vanets)," in *2012 6th International Conference on Signal Processing and Communication Systems*, 2012, pp. 1–9.

[27] K. Mershad and H. Artail, "A framework for secure and efficient data acquisition in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 536–551, 2013.

[28] CVIS, https://www.cvisproject.org/.

[29] R. Baldessari, B. Bödekker, M. Deegener, A. Festag, W. Franz, C. Kellum, T. Kosch, A. Kovacs, M. Lenardi, C. Menig, T. Peichl, M. Röckl, D. Seeberger, M. Straßberger, H. Stratil, H.-J. Vögel, B. Weyl, and W. Zhang, "Car-2-car communication consortium - manifesto," 01 2007.

[30] AHS, https://www.www.fhwa.dot.gov/publications/publicroads/94summer/p94su1.cfm.

[31] C. Little, "The intelligent vehicle initiative: advancing" human-centered" smart vehicles," *Public Roads*, vol. 61, no. 2, pp. 18–25, 1997.

[32] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: Taxonomy and design," *Wireless Networks*, vol. 16, pp. 2349–2370, 11 2010.

[33] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehicular mobility simulation for vanets," in *40th Annual Simulation Symposium (ANSS'07)*, 2007, pp. 301–309.

[34] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "Vanetmobisim: Generating realistic mobility patterns for vanets." New York, NY, USA: Association for Computing Machinery, 2006. [Online]. Available: https://doi.org/10.1145/1161064.1161084

[35] C. D. Wang and J. P. Thompson, *Apparatus and method for motion detection and tracking of objects in a region for collision avoidance utilizing a real-time adaptive probabilistic neural network*, 1997, u.S patent no. 5.613.039.

[36] European project REACT, http://www.react-project.org.

[37] F. Soldo, C. Casetti, C. Chiasserini, and P. A. Chaparro, "Video streaming distribution in vanets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1085–1091, 2011.

[38] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee, "Routing protocols in vehicular ad hoc networks: A survey and future perspectives," *J. Inf. Sci. Eng.*, vol. 26, pp. 913–932, 05 2010.

[39] J. Bernsen and D. Manivannan, "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification," *Pervasive and Mobile Computing*, vol. 5, no. 1, pp. 1–18, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574119208000758

[40] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *A Taxonomy of Data Communication Protocols for Vehicular Ad Hoc Networks*, 2013, pp. 515–544.

[41] M. Artimy, W. Robertson, and W. Phillips, *Vehicular Ad Hoc Networks: An Emerging Technology Toward Safe and Efficient Transportation*, 03 2008, pp. 405 – 432.

[42] S. Najafzadeh, N. Ithnin, and S. Razak, "Broadcasting in connected and fragmented vehicular ad hoc networks," *International Journal of Vehicular Technology*, vol. 2014, pp. 1–15, 03 2014.

[43] L. Villas, A. Boukerche, R. Araujo, A. Loureiro, and J. Ueyama, "Network partition-aware geographical data dissemination," 06 2013, pp. 1439–1443.

[44] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1556, 2007.

[45] M. Benamar, N. Benamar, and D. El Ouadghiri, "The effect of cooperation of nodes on vdtn routing protocols," 10 2015.

[46] VEINS, http://veins.car2x.org.

[47] OMNET Simulator, http://www.omnetpp.org.

[48] SUMO Simulator, http://sumo.sourceforge.net.

[49] OpenStreeMap, https://www.openstreetmap.org/.

[50] R, https://www.r-project.org/.

[51] J. Sanguesa, M. Fogue, P. Garrido, F. Martinez, J.-C. Cano, and C. Calafate, "Using topology and neighbor information to overcome adverse vehicle density conditions," *Transportation Research Part C Emerging Technologies*, vol. 42, 02 2014.

[52] F. J. Martinez, M. Fogue, M. Coll, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Evaluating the impact of a novel warning message dissemination scheme for vanets using real city maps," in *NETWORKING 2010*, M. Crovella, L. M. Feeney, D. Rubenstein, and S. V. Raghavan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 265–276.

[53] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Evaluating the impact of a novel message dissemination scheme for vehicular networks using real maps," *Transportation Research Part C: Emerging Technologies*, vol. 25, pp. 61–80, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X12000678

[54] I. Achour, T. Bejaoui, A. Busson, and S. Tabbane, " SEAD: A simple and efficient adaptive data dissemination protocol in vehicular ad-hoc networks ," *Wireless Networks*, p. 11, Sep. 2015. [Online]. Available: https://hal.inria.fr/hal-01242289

[55] J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J. Cano, and C. T. Calafate, "Topology-based broadcast schemes for urban scenarios targeting adverse density conditions," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, pp. 2528–2533.

Appendices

.1 Appendix 1.

omnetpp.ini

```
[Config RTPAD]
description = "Real-Time Probabilistic Adaptive Dissemination"
*.node[*].applType = "RTPAD"
*.node[*].appl.sendBeacons = true

*.node[*].appl.neighborLifetimeThreshold = 2
*.node[*].appl.hostToJunctionDistanceThreshold = 20
*.node[*].appl.distanceThreshold = 500
*.node[*].appl.sendWarningInterval = 2
*.node[*].appl.delta = 0.004
*.node[*].appl.alpha = 1.5
*.node[*].appl.rTransmision = 600
*.node[*].hopCount.result-recording-modes = -vector

*.node[*].appl.warningReceivedSignal.scalar-recording = true
*.node[*].appl.warningReceivedSignal.vector-recording = true

*.node[*].appl.beaconReceivedSignal.scalar-recording = true
*.node[*].appl.beaconReceivedSignal.vector-recording = true

*.node[*].appl.arrival.scalar-recording = true
*.node[*].appl.arrival.vector-recording = true

*.node[*].appl.arrival1.scalar-recording = true
*.node[*].appl.arrival1.vector-recording = true

*.node[*].appl.warningMsgCounterRx.scalar-recording = true
*.node[*].appl.warningMsgCounterRx.vector-recording = true

*.node[*].appl.beaconMsgCounterRx.scalar-recording = true
*.node[*].appl.beaconMsgCounterRx.vector-recording = true

*.node[*].appl.distanceMsgRx.scalar-recording = true
*.node[*].appl.distanceMsgRx.vector-recording = true

*.node[*].appl.numberOfNodes.scalar-recording = true
*.node[*].appl.numberOfNodes.vector-recording = true

#Enable when running in Qtenv graphic interface
#*.node[*].appl.indexOfAccidentNode = ${10}

```

```
#Enable when running in Cmdenv interface to store results
*.node[*].appl.indexOfAccidentNode = ${11,30,41,50,55}
```

RTPAD.cc

```
#include "veins/modules/application/traci/RTPAD.h"

using namespace veins;
Define_Module(veins::RTPAD);
const double timeSlot = 0.000013;

void RTPAD::initialize(int stage)
{
    timeoutEventBeacon = new cMessage("timeoutEventBeacon");
    timeoutEventWarning = new cMessage("timeoutEventWarning");
    DemoBaseApplLayer::initialize(stage);
    if (stage == 0) {
        // Initializing members and pointers of your application goes
            here
        sentMessage = false;
        senderReceiverDistanceThreshold = par("
            senderReceiverDistanceThreshold").doubleValue();
        neighborLifetimeThreshold = par("neighborLifetimeThreshold").
            doubleValue();
        randomWaitingTime= par("randomWaitingTime").doubleValue();
        distanceThreshold = par("distanceThreshold").doubleValue();
        indexOfAccidentNode = par("indexOfAccidentNode").intValue();
        hostToJunctionDistanceThreshold = par("
            hostToJunctionDistanceThreshold").doubleValue();
        junctionIds = mobility->getCommandInterface()->getJunctionIds
            ();
        lanes = mobility->getCommandInterface()->getLaneIds();

        arrivalSignal = registerSignal("arrival");
        arrivalSignal1 = registerSignal("arrival1");

        warningMsgCounterSignalRx = registerSignal("
            warningMsgCounterRx");
        beaconMsgCounterSignalRx = registerSignal("beaconMsgCounterRx"
            );
        distanceMsgSignalRx = registerSignal("distanceMsgRx");
        numberOfNodesSignal = registerSignal("numberOfNodes");

        receivedData=receivedBeacons=0;
        lastDroveAt = simTime();
```

```
        contador = 0;

        rRedundant = 0.0;
        rTransmision = par("rTransmision").doubleValue();
        rTransmisionV = rTransmision / 100;

        delta = par("delta").doubleValue();
        alpha = par("alpha").doubleValue();
        Pt=1;

        //Map junctionsID with its position
        for (list<string>::iterator i = junctionIds.begin(); i != junctionIds.end(); ++i) {
            string jId = *i;
            Coord jPos = mobility->getCommandInterface()->junction(jId).getPosition();
            junctionMap[jId] = jPos;
        }

        //Taking the number of streets
        for(list<string>::iterator i = lanes.begin(); i!= lanes.end(); ++i){
            string jId = *i;
            jId = jId.substr(0, jId.find('_'));
            if(jId[0] != ':' && !(find(streets.begin(), streets.end(), jId) != streets.end()))
                streets.push_back(jId);
        }

        //Calculate SJ Ratio
        SJ = (float)streets.size() / (float)junctionMap.size();
    }
    else if (stage == 1) {
        //Nothing
    }
}

//Function to estimate traffic density
float RTPAD::density(float SJ, int p)
{
    return abs(a1 + a2*p + a3*SJ + a4*pow(p,2) + a5*pow(SJ,2) + a6*pow(p,3) + a7*pow(SJ,3) + a8*(p*SJ) + a9*(pow(p,2)*SJ) + a10*(p*pow(SJ,2)));
}
```

```

//Check which protocol to Use
string RTPAD::checkProtocol(float SJ, int density){
    if( (density > 175) || (density > 125 && SJ < 1.1) || (density>75
        && SJ<1.05 ))
        return "NJL";
    else if( SJ > 0.8 && density < 175 )
        return "eSBR";
    else
        return "eMBR";
}

//Check if the vehicles is the closest to Junction
bool RTPAD::hostIsClosestToJunction(string junctionId)
{
    Coord jPos = junctionMap[junctionId];
    double hDist = jPos.distance(mobility->getPositionAt(simTime()));
    for (uint i = 0; i < neighbors.size(); ++i) {
        BaseFrame1609_4* neighbor = neighbors[i];
        if (jPos.distance(neighbor->getSenderPos()) < hDist) {
            return false;
        }
    }
    return true;
}

//Check if the vehicle is on Junction and return de ID of the junction
string RTPAD::hostIsInJunction()
{
    for (map<string,Coord>::iterator i = junctionMap.begin(); i !=
        junctionMap.end(); ++i) {
        string jId = i->first;
        Coord jPos = i->second;
        Coord hPos = mobility->getPositionAt(simTime());
        if (jPos.distance(hPos) < hostToJunctionDistanceThreshold) {
            return jId;
        }
    }
    return string();
}

//Check if the vehicle is on Junction
bool RTPAD::vehicleOnJunction()
{
```

```
    bool onJunction=false;
    for (map<string,Coord>::iterator i = junctionMap.begin(); i !=
        junctionMap.end(); ++i) {
        string jId = i->first;
        Coord jPos = i->second;
        Coord hPos = mobility->getPositionAt(simTime());
        if (jPos.distance(hPos) < hostToJunctionDistanceThreshold) {
            onJunction=true;
            return onJunction;
        }
    }
    return onJunction;
}

//Calculate Distance to Junction
double RTPAD::calcDistJoin(BaseFrame1609_4* wsm)
{
     for (list<string>::iterator i = junctionIds.begin(); i !=
         junctionIds.end(); ++i) {
         string jId = *i;
         Coord jPos = mobility->getCommandInterface()->junction(jId).
             getPosition();
         Coord hPos = mobility->getPositionAt(simTime());
         junctionDistance[jId] = jPos.distance(hPos);
     }
     double distJ=getMin(junctionDistance);
     return distJ;

}

//Get the minimum value
double RTPAD::getMin(map<string, double> mymap)
{
    pair<string, double> min = *min_element(mymap.begin(), mymap.end()
        , CompareSecond());
    return min.second;
}

//Forward a Message
void RTPAD::forwardMessage(double p)
{
    // Duplicate message and send the copy.
    BaseFrame1609_4 *copy = receivedWarningMessageMap[idMsg][0]->dup()
        ;
```

```
    // Increment hop count.
    copy->setHopCount(copy->getHopCount()+1);
    // Update Position
    copy->setSenderPos(mobility->getPositionAt(simTime()));
    copy->setCarId(getParentModule()->getIndex());
    sendDown(copy);
    //std::cerr << "[INFO] RE-BROADCAST  STARTED @simTime: " <<
        simTime().str() << " from node: " << getParentModule()->
        getIndex() << endl;
    return;
}

double RTPAD::distanceFactorCalc(double d_sr, double d_ri, double
    r_max, bool  r_vehicleOnIntersection)
{
    // Dsr is the relative distance between source s and receptor r
        vehicles
    // Dri is the relative distance between vehicle r and the next
        nearest intersection
    // Rmax is the maximum transmission range
    double d;
    if (!r_vehicleOnIntersection)
        if(d_sr>r_max)
            d=1;
        else
            d=d_sr/r_max;
    else if(d_ri<10)
        d=1;
    else
        d=1-(d_ri/(d_ri+1));
    return d;
}

void RTPAD::finish()
{
    DemoBaseApplLayer::finish();
}

void RTPAD::onBSM(DemoSafetyMessage* bsm)
{
    BaseFrame1609_4* wsm = dynamic_cast<BaseFrame1609_4*>(bsm);

    receivedBeacons++;
    emit(beaconMsgCounterSignalRx, receivedBeacons);
```

```

    isNewNeighbor = true;
    vector<uint> indices;

    for (uint i = 0; i < neighbors.size(); ++i) {
        BaseFrame1609_4* neighbor = neighbors[i];
        if (neighbor->getPsid() == wsm->getPsid()) {
            isNewNeighbor = false;
            neighbor->setSenderPos(bsm->getSenderPos());
        }
        else {
            // Check for removal
            if (simTime() - neighbor->getArrivalTime() >
               neighborLifetimeThreshold)
                indices.push_back(i);
        }
    }

    if(isNewNeighbor)
        neighbors.push_back(wsm->dup());

    WaveShortMessages newNeighborList;
    for (uint i = 0; i < neighbors.size(); ++i) {
        bool keepNeighbor = true;
        for (uint j = 0; j < indices.size(); ++j) {
            if (i == indices[j])
                keepNeighbor = false;
        }
        if (keepNeighbor)
            newNeighborList.push_back(neighbors[i]);
    }

    neighbors = newNeighborList;
    densityV = density(SJ,neighbors.size());
    protocol = checkProtocol(SJ,(int)densityV);

    //Store Carry and Forward Mechanism
    if(isNewNeighbor && receivedWarningMessageMap.size() > 0 && !
       timeoutEventWarning->isScheduled() && neighbors.size()<3)
        scheduleAt(simTime(), timeoutEventWarning);

    //Debug
    //EV<<"density in "<<getParentModule()->getIndex()<<"    "<<
       densityV<<endl;
```

```
    //EV<< streets.size() << "  " << junctionIds.size() << endl;
    //EV<<"SJ : "<<SJ<<endl;
    //EV<<"neighbors: "<<neighbors.size()<<endl;
    //EV<<"protocol :"<< protocol<<endl;
}

void RTPAD::onBSM_NSF(DemoSafetyMessage* bsm){
    if (isNewNeighbor) {
        if (receivedWarningMessageMap.size() > 1){
            Pt*=((2*alpha)/rRedundant);
            for (map<long,WaveShortMessages>::iterator i =
                receivedWarningMessageMap.begin(); i !=
                receivedWarningMessageMap.end(); ++i) {
                BaseFrame1609_4* msg = i->second[0];
                ASSERT(msg);
                // disseminate warning message
                if((rand()%100 + 1) <= (Pt*10)){
                    sendDown(msg->dup());
                    std::cerr << "[INFO] RE-BROADCAST  STARTED FROM
                        BSM_NSF @simTime: " << simTime().str() << "
                        from node: " << getParentModule()->getIndex()
                        << endl;
                }
            }
        }
    }
}

void RTPAD::onBSM_JSF(DemoSafetyMessage* bsm){

    BaseFrame1609_4* wsm = dynamic_cast<BaseFrame1609_4*>(bsm);

    IntVector indices1 = wsm->getIndices();

    if (isNewNeighbor) {
         for (map<long,WaveShortMessages>::iterator i =
             receivedWarningMessageMap.begin(); i !=
             receivedWarningMessageMap.end(); ++i)
         {
              BaseFrame1609_4* msg = i->second[0];
              findHost()->getDisplayString().setTagArg("i", 1, "purple"
                  );
              // disseminate warning message
              if (timeoutEventBeacon->isScheduled()){
```

```
                    cancelEvent(timeoutEventBeacon);
                }
                else{
                    double dsr = (mobility->getPositionAt(simTime()).distance(wsm->getSourcePos()));
                    double distanceFromPreviousNode=mobility->getPositionAt(simTime()).distance(wsm->getSenderPos());
                    double distanceToIntersection=calcDistJoin(wsm);
                    timeoutBeacon=(simtime_t)(timeSlot*distanceFromPreviousNode)+SimTime(randomWaitingTime, SIMTIME_US);
                    scheduleAt(simTime()+timeoutBeacon , timeoutEventBeacon);
                }
            }
        }

    if(neighbors.size()>0 && receivedWarningMessageMap.size()>0 && vehicleOnJunction())
        if((rand()%100 + 1) <= (Pt*10))
            forwardMessage(1);
}


void RTPAD::onWSM(BaseFrame1609_4* wsm)
{
    if (sentMessage)
        return;

    receivedData++;
    emit(warningMsgCounterSignalRx, receivedData);

    idMsg = wsm->getTreeId();
    receivedWarningMessageMap[wsm->getTreeId()].push_back(wsm->dup());
    isNewWarning = true;
    if (receivedWarningMessageMap[wsm->getTreeId()].size() == 1) {
        std::cerr << "[DEBUG] node: " << getParentModule()->getIndex() << " wsm->getTreeId(): " << wsm->getTreeId() << std::endl;
        findHost()->getDisplayString().setTagArg("i", 1, "green");
        simtime_t delayFirstNewMessage = wsm->getArrivalTime()-wsm->getCreationTime();
        emit(arrivalSignal, delayFirstNewMessage);
    }
```

```
    else
        isNewWarning = false;

    rRedundant = receivedData;

    //Goes to the most suitable scheme
    if(protocol == "NJL"){
        onWSM_NJL(wsm);
        return;
    }
    else if(protocol == "eSBR"){
        onWSM_eSBR(wsm);
        return;
    }
    else{
        onWSM_eMBR(wsm);
        return;
    }
}

void RTPAD::onWSM_NSF(BaseFrame1609_4* wsm){
    if(isNewWarning){
        if (neighbors.size() > 1) {
            Pt*=((2*alpha)/rRedundant);
            if((rand()%100 + 1) <= (Pt*10)){
                sendDown(wsm->dup());
                //std::cerr << "[INFO] RE-BROADCAST  STARTED FROM
                    WSM_NSF @simTime: " << simTime().str() << " from
                    node: " << getParentModule()->getIndex() << endl;
            }
        }
    }
}

void RTPAD::onWSM_JSF(BaseFrame1609_4* wsm){

    if (isNewWarning && vehicleOnJunction()) {
        //EV<<"ENVIANDO SELFMSG EN : "<<timeoutWarning<<endl;

        if (uniform(0,1) < 1){
            forwardMessage(1);
            return;
        }
        else{
```

```
            if (timeoutEventWarning->isScheduled())
                cancelEvent(timeoutEventWarning);
            else{
                double dsr=(mobility->getPositionAt(simTime()).distance(wsm->getSourcePos()));
                double d=distanceFactorCalc(dsr,calcDistJoin(wsm),distanceThreshold,vehicleOnJunction());
                double distanceFromPreviousNode=mobility->getPositionAt(simTime()).distance(wsm->getSenderPos());
                double difDistance=1;
                (distanceThreshold-distanceFromPreviousNode)>0 ? difDistance=abs(distanceThreshold-distanceFromPreviousNode):difDistance=1;
                timeoutWarning = (simtime_t)(timeSlot * difDistance) + SimTime(randomWaitingTime, SIMTIME_US);
                scheduleAt(simTime()+timeoutWarning , timeoutEventWarning);
            }
        }
    }
    else{
        if (timeoutEventWarning->isScheduled()) {
            //std::cerr << "[INFO] RE-BROADCAST  CANCELED @simTime: " << simTime().str() << " in node: " << getParentModule()->getIndex() << endl;
            cancelEvent(timeoutEventWarning);
        }
    }
}

void RTPAD::onWSM_NJL(BaseFrame1609_4* wsm){

    string jId = hostIsInJunction();

    if (jId.empty()){
        //EV<<"Not in Junction"<<endl;
    }
    else{
        if (!hostIsClosestToJunction(jId)) {
            findHost()->getDisplayString().setTagArg("i", 1, "purple");
            if(!timeoutEventWarning->isScheduled()){
                scheduleAt(simTime() + sendWarningInterval,
```

```
                    timeoutEventWarning);
                }
            }
        else{
            findHost()->getDisplayString().setTagArg("i", 1, "blue");
            if(!timeoutEventWarning->isScheduled()){
                scheduleAt(simTime(), timeoutEventWarning);
            }
            else{
                cancelEvent(timeoutEventWarning);
                scheduleAt(simTime(), timeoutEventWarning);
            }
        }
    }
    return;
}

void RTPAD::onWSM_eSBR(BaseFrame1609_4* wsm){

    if(isNewWarning){
        if ((wsm->getSenderPos().distance(curPosition) >
            senderReceiverDistanceThreshold)
                || (mobility->getRoadId() != wsm->getWsmData())
                || ((mobility->getRoadId() == wsm->getWsmData()) &&
                    vehicleOnJunction())) {
            scheduleAt(simTime() + 2 + uniform(0.01, 0.2),
                timeoutEventWarning);
        }
        simtime_t delayFirstNewMessage = wsm->getArrivalTime()-wsm->
            getCreationTime();
        emit(arrivalSignal, delayFirstNewMessage);
    }
}

void RTPAD::onWSM_eMBR(BaseFrame1609_4* wsm){

    if(isNewWarning){
        if ((wsm->getSenderPos().distance(curPosition) >
            senderReceiverDistanceThreshold)
                || (mobility->getRoadId() != wsm->getWsmData())
                || ((mobility->getRoadId() == wsm->getWsmData()) &&
                    hostIsClosestToJunction(hostIsInJunction()))) {
            scheduleAt(simTime() + 2 + uniform(0.01, 0.2),
                timeoutEventWarning);
```

```
        }
        simtime_t delayFirstNewMessage = wsm->getArrivalTime()-wsm->
            getCreationTime();
        emit(arrivalSignal, delayFirstNewMessage);
    }
}


void RTPAD::onWSA(DemoServiceAdvertisment* wsa)
{
    //Nothing to do here.
}

void RTPAD::handleSelfMsg(cMessage* msg)
{
    if ((!strcmp(msg->getName(), "timeoutEventWarning"))) {
        WaveShortMessages ms = receivedWarningMessageMap[idMsg];

        if (ms.empty() || ms.size() > 1){
            std::cerr << "[INFO] RE-BROADCAST  CANCELED @simTime: " <<
                 simTime().str() << " from node: " << getParentModule()
                ->getIndex() << endl;
            return;
        }

        Pt*=((2*alpha)/rRedundant);
        if((rand()%100 + 1) <= (Pt*10)){
            sendDown(ms[0]->dup());
            findHost()->getDisplayString().setTagArg("i", 1, "blue");
            //std::cerr << "[INFO] RE-BROADCAST  STARTED @simTime: "
                << simTime().str() << " from node: " << getParentModule
                ()->getIndex() << endl;
            return;
        }
    }
    else if((!strcmp(msg->getName(), "timeoutEventBeacon"))){
        Pt*=((2*alpha)/rRedundant);
        if((rand()%100 + 1) <= (Pt*10))
            forwardMessage(1);
        return;
    }
    else
        DemoBaseApplLayer::handleSelfMsg(msg);

```

```
    return;
}

void RTPAD::handlePositionUpdate(cObject* obj)
{
    DemoBaseApplLayer::handlePositionUpdate(obj);

    if (mobility->getSpeed() < 1) {
        if ((simTime() - lastDroveAt >= 10)
            && (!sentMessage)
            && (indexOfAccidentNode == getParentModule()->getIndex()))
            {

            //std::cerr << "[DEBUG] ACCIDENT STARTED @simTime: " << simTime().str() << " for node: " << getParentModule()->getIndex() << endl;

            findHost()->getDisplayString().setTagArg("i", 1, "red");
            if (!sentMessage){
                BaseFrame1609_4* wsm = new BaseFrame1609_4();
                populateWSM(wsm);
                wsm->setWsmData(mobility->getRoadId().c_str());
                wsm->setPsid(getParentModule()->getIndex());
                wsm->setUserPriority(0);
                wsm->setWsmVersion(1);
                wsm->setTimestamp(simTime());
                wsm->setSenderAddress(myId);
                wsm->setSpeed(mobility->getSpeed());
                wsm->setCarId(getParentModule()->getIndex());
                wsm->setAngleRad(mobility->getHeading().getRad());
                wsm->setVecX(curPosition.x);
                wsm->setVecY(curPosition.y);
                wsm->setVecZ(curPosition.z);
                wsm->setRecipientAddress(-1);
                wsm->setSenderPos(curPosition);
                wsm->setSourcePos(curPosition);
                wsm->setSerial(0);
                wsm->setIndices(indicesWarningMsgsApp);
                sendDown(wsm->dup());
                sentMessage = true;
            }
        }
    }
    else {
```

```
            lastDroveAt = simTime();
        }
}
}
```

RTPAD.h

```
#pragma once

#include "veins/veins.h"
#include "veins/modules/application/ieee80211p/DemoBaseApplLayer.h"
#include "veins/modules/application/traci/beacon.h"

using namespace omnetpp;
using namespace veins;
using namespace omnetpp;
using namespace std;

typedef std::vector<BaseFrame1609_4*> WaveShortMessages;
typedef std::pair<std::string, double> MyPairType;

namespace veins {
class VEINS_API RTPAD : public DemoBaseApplLayer {
public:
    void initialize(int stage) override;
    void finish() override;
    long long a1 = -791.74488563075374;
    long long a2 = -0.65986335423139231;
    long long a3 = 2271.7144481059472;
    long long a4 = 1.1989021509173876;
    long long a5 = -2101.9985100197723;
    long long a6 = -0.017509026843351649;
    long long a7 = 630.96785945693671;
    long long a8 = -4.8107314165096247;
    long long a9 = -0.76438506962308739;
    long long a10 = 14.601116345732333;
    float densityV = 0;
    float SJ = 0;
    long idMsg;
    bool sentMessage;
    int indexOfAccidentNode;
    double randomWaitingTime;
    list<string> lanes;
    string protocol;
    vector<string> streets;
```

```
    list<string> junctionIds;
    WaveShortMessages neighbors;
    bool isNewNeighbor;
    bool isNewWarning;
    vector<int> indicesWarningMsgsApp;
    double distanceThreshold;
    double sendWarningInterval = 2;
    int contador;

    double neighborLifetimeThreshold;
    std::map<std::string,Coord> junctionMap;
    double hostToJunctionDistanceThreshold;
    uint32_t receivedBeacons;
    uint32_t receivedData;
    simtime_t lastDroveAt;

    simtime_t timeoutBeacon;
    cMessage *timeoutEventBeacon;

    simtime_t timeoutWarning;
    cMessage *timeoutEventWarning;

    simsignal_t arrivalSignal;
    simsignal_t arrivalSignal1;
    simsignal_t warningMsgCounterSignalRx;
    simsignal_t beaconMsgCounterSignalRx;
    //simsignal_t warningMsgCounterSignalTx;
    //simsignal_t beaconMsgCounterSignalTx;
    simsignal_t distanceMsgSignalRx;
    simsignal_t numberOfNodesSignal;

    map<string,double> junctionDistance;
    map<long,WaveShortMessages> receivedWarningMessageMap;

    double senderReceiverDistanceThreshold;
    double rRedundant;
    double rTransmisionV;
    double rTransmision;
    double nSlots;
    double wTime;
    double delta;
    double Pt=1.0;
    double alpha;

```

```
    struct CompareSecond
    {
        bool operator()(const MyPairType& left, const MyPairType&
            right) const
        {
        return left.second < right.second;
        }
    };

protected:
    void onBSM(DemoSafetyMessage* bsm) override;
    void onWSM(BaseFrame1609_4* wsm) override;
    void onWSA(DemoServiceAdvertisment* wsa) override;
    void handleSelfMsg(cMessage* msg) override;
    void handlePositionUpdate(cObject* obj) override;

    void onBSM_NSF(DemoSafetyMessage* bsm);
    void onBSM_JSF(DemoSafetyMessage* bsm);

    void onWSM_NSF(BaseFrame1609_4* wsm);
    void onWSM_JSF(BaseFrame1609_4* wsm);
    void onWSM_NJL(BaseFrame1609_4* wsm);
    void onWSM_eSBR(BaseFrame1609_4* wsm);
    void onWSM_eMBR(BaseFrame1609_4* wsm);

    float density(float SJ, int p);
    string checkProtocol(float SJ, int density);
    bool hostIsClosestToJunction(std::string junctionId);
    string hostIsInJunction();
    double  getMin(std::map<std::string, double> mymap);
    bool vehicleOnJunction();
    string getIdJunction();
    double calcDistJoin(BaseFrame1609_4* wsm);
    void forwardMessage(double p);
    double distanceFactorCalc(double dsr, double dri, double rmax,
        bool inIntersection);
};

}
```

RTPAD.ned

```
package org.car2x.veins.modules.application.traci;
import org.car2x.veins.modules.application.ieee80211p.
    DemoBaseApplLayer;
```

```

simple RTPAD extends DemoBaseApplLayer
{
    parameters:
    @signal[arrival](type="simtime_t");
        @statistic[delayFirstNewMessage](title="delayFirstNewMessage"; source="arrival"; record=vector,stats; );

        @signal[arrival1](type="long");
        @statistic[hopCount](title="hop count"; source="arrival1"; record=vector,stats;);

        @signal[warningMsgCounterRx](type="long");
        @statistic[warningMsgCounterRx](title="warningMsgCounterRx"; source="warningMsgCounterRx"; record=vector,stats; );

        @signal[beaconMsgCounterRx](type="long");
        @statistic[beaconMsgCounterRx](title="beaconMsgCounterRx"; source="beaconMsgCounterRx"; record=vector,stats; );

        @signal[distanceMsgRx](type="double");
        @statistic[distanceMsgRx](title="distanceMsgRx"; source="distanceMsgRx"; record=vector,stats; );

        @signal[numberOfNodes](type="long");
        @statistic[numberOfNodes](title="numberOfNodes"; source="numberOfNodes"; record=vector,stats;);

    @class(veins::RTPAD);
    @display("i=block/app2");
    double neighborLifetimeThreshold = default(2.0);
    volatile double randomWaitingTime = default(uniform(1,11));
    double hostToJunctionDistanceThreshold = default(20);
    double senderReceiverDistanceThreshold = default(20);
    int indexOfAccidentNode = default(10);
    double distanceThreshold = default(500);
    double sendWarningInterval = default(2);
    double rTransmision;
    double delta;
    double alpha;
}
```