Electric and non-electric vehicle clustering in LTE and 802.11p wireless networks

University of Thessaly

School of Engineering

Department of Electrical and Computer Engineering

Tserpe Paraskevi

Supervisor: Assoc.Prof. Katsaros Dimitrios

July 2020

Ομαδοποίηση μεικτών στόλων οχημάτων ηλεκτρικών και μη ηλεκτρικών σε δίκτυα LTE και 802.11p

Πανεπιστήμιο Θεσσαλίας

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τσερπέ Παρασκευή

Επιβλέπων καθηγητής: Κατσαρός Δημήτριος

Ιούλιος 2020

The committee members, hereby, certify that have read the thesis presented by Tserpe Paraskevi and that it is fully adequate in scope and quality as a partial requirement for the degree of Master of Electrical and Computer Engineering.

Supervisor: Assoc. Prof. Katsaros Dimitrios

2nd committee member: Assoc. Prof. Korakis Athanasios

3rd committee member:Tousidou Eleni

Dedicated to my family and friends.

Declaration

The content of this work does not necessarily reflect the views of the Department, the Supervisor, or the committee that approved it.

The author of this work certifies that any help he/she had for its preparation is fully recognized and refers to the work. He/she also states that he/she has cited any sources from which it used data, ideas or words, whether they are accurate or paraphrased.

Περίληψη

Τα ad-hoc δίκτυα οχημάτων έχουν κεντρίσει σημαντικά το ενδιαφέρον τις τελευταίες δεκαετίες. Το περιβάλλον ενός δικτύου οχημάτων παρουσιάζει πολλές προκλήσεις, καθώς συνδιάζει μία σταθερή υποδομή, όπως οι οδικές μονάδες, και ad-hoc επικοινωνίες μεταξύ των οχημάτων. Επιπρόσθετα, στα κυψελοειδή δίκτυα η απόδοση του φάσματος μειώνεται δραστικά καθώς ο αριθμός των χρηστών αυξάνεται. Αυτή η ιδιαιτερότητα έφερε την ανάγκη για δημιουργία αποτελεσματικών αλγορίθμων, που θα διευκόλυναν την επικοινωνία τέτοιων κόμβων. Υπάρχει μεγάλος αριθμός εφαρμογών, όπως προειδοποιήσεις ασφαλείας σε περίπτωση ατυχήματος, διανομή περιεχομένου, μετάδοση πληροφοριών σχετικά με κυκλοφοριακή συμφόρηση και διαχείριση της κυκλοφορίας, οι οποίες απαιτούν σταθερή ομαδοποίηση των οχημάτων. Επιπλέον, η χρήση ηλεκτρικών οχημάτων είναι πολύ διαδεδομένη τα τελευταία χρόνια, λόγω της συνεισφοράς τους στο περιβάλλον. Οπότε, δημιουργείται η ανάγκη να συμπεριλάβουμε και αυτόν τον τύπο οχημάτων στο μοντέλο μας.

Στα πλαίσια της παρούσας διπλωματικής εργασίας προτείνουμε τον EAVC(Energy Aware VANET Clustering), ένα πρωτόκολλο ομαδοπίησης μεικτών στόλων οχημάτων ηλεκτρικών και μη ηλεκτρικών που συνδυάζει LTE και 802.11p τεχνολογίες. Πιο συγκεκριμένα ένας αλγόριθμος με ασαφή λογική χρησιμοποιείται στο πρώτο επίπεδο για την επιλογή των clusterhead κόμβων και ένας Q-learning αλγόριθμος χρησιμοποιείται στο δεύτερο επιπέδο για να συντονίσει τον αριθμό των gateway κόμβων,καθώς και και για τον σχηματισμό και τη συντήρηση των cluster. Πραγματοποιήσαμε αρκετές προσομοιώσεις για να αξιολογήσουμε την απόδοση του προτεινόμενου αλγορίθμου, εφαρμόζοντας διάφορα σενάρια για διάφορες συνθήκες του δικτύου αντίστιχα.

Abstract

Vehicular Ad-hoc networks (VANETs) have attracted tremendous attention in recent years. A vehicular network is a challenging environment since it combines a fixed infrastructure such as roadside units(RSUs), and ad-hoc communications among vehicles. However, in cellular networks, the spectrum efficiency drops drastically along with the increase of the user density. This peculiarity brought the necessity for efficient algorithms that would facilitate the communication of such nodes. There does exist a large number of applications such as safety warnings in case of an accident, content distribution, broadcast of information concerning traffic jams and traffic management, which demand stable clustering among the vehicles. Furthermore, the use of electric vehicles(EV) has become more widespread in past years, due to their contribution to the environment. So we need to include such vehicles in our model.

In this disertation we propose the Energy Aware VANET Clustering(EAVC), a two-level clustering approach for electric and non-electric vehicles, integrating LTE(Long Term Evolution) with 802.11p. More specifically, a fuzzy logic-based algorithm is used in the first-level clustering for the cluster head(CH) selection, and a Q-learning algorithm is employed in the second-level clustering to tune the number of gateway nodes, as well as for formation and maintenance of clusters. We conducted several simulations to evaluate the performance of the proposed algorithm, by applying various scenarios for different network conditions respectively.

Acknowledgements

First and foremost, I would like to express my thanks and appreciation to my supervisor Prof. Dimitrios Katsaros, for his constant support and guidance throughout the research and writing of this thesis.

I am also grateful to Prof. Korakis Athanasios and Tousidou Eleni for their valuable help. Additionally, I would like to thank Pavlos Basaras and Foteini Katsarou for their time and all the useful tips concerning the programs SUMO and NS3.

Furthermore, I wish to thank my friends for their support and encouragement. Altogether, we shared both beautiful moments and concerns too.

Finally, I couldn't omit my family for their unconditional love and patience. They taught me, above all, to fight for my dreams and never give up.

Thank you all!

Contents

List of Figures

List of Tables

Abbrevations

VANET	Vehicular Ad hoc NETwork
RSU	RoadSide Unit
EAVC	Energy Aware VANET Clustering
LTE	Long Term Evolution
MANET	Mobile Ad hoc NETwork
SUMO	Simulation of Urban MObility
WANET	Wireless Ad hoc NETwork
WMN	Wireless Mesh Networks
WSN	Wireless Sensor Network
EV	Electric Vehicle
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
ITS	Intelligent Transportation System
GPSR	Greedy Perimeter Stateless Routing
GPS	Global Positioning System
RSS	Received Signal Strength
CH	Cluster Head
GW	GateWay
CM	Cluster Member
CDS-SVB	Connected Dominating Set based Stable Virtual Backbone
WAVE	Wireless Access in Vehicular Environments
VF	Velocity Factor
LF	Leadership Factor
SQF	Signal Quality Factor
EF	Energy Factor
CV	Competency Value
BS	Base Station
HRR	Hello Reception Ratio

Chapter 1

Introduction

In the last decades, clustering in Vehicular Ad hoc Networks(VANETS) is an interesting field due to he challenges that characterize them. One of the most crucial limitations in this area is the limited bandwidth that is available for each node in cellular networks. The exponential increase in vehicles population is a big challenge to contents distribution. Moreover, such networks are characterized by frequent topology change due to the high mobility and average speed of vehicles. The need for solutions in this area becomes apparent. In addition, VANETs are widespread for their condribution in a range fo applications. They can be used for exchanging information about the current driving situation, whether conditions and hazard areas[17]. The key to routing and data dissemination in VANETs seems to be the clustering technique.

The topic of this thesis is the clustering in VANETs, consisting of both electric and non-electric vehicles. The clustering application essentially aims to the as efficient as possible data distribution among nodes, overcoming the limitations in VANETs. In more detail, we propose the Energy Efficient VANET Clustering(EEVC), a two-level clustering approach in VANETs, integrating LTE and 802.11p interface. This approach includes velocity, vehicle density, channel condition, and the remaining vehicles' energy in the selection of cluster heads. We perform extensive simulations to evaluate the proposed algorithm by comparing it with other standards. Our protocol can quickly adapt to various scenarios with different vehicle densities and velocities.

1.1 Thesis structure

Once we have presented the main topic of this dissertation, let's give a brief description of what's next. More specifically, the remainder of this thesis is organized as follows:

- **Chapter 02** gives a brief outline of the theory concerning ad-hoc networks. MANETs and VANETs, their features, and comparison between them are also presented in this chapter. Furthermore, clustering techniques and their benefits to such networks are described too.
- In **Chapter 03**, the simulation tools(Simulation of Urban MObility and NS-3), as well as their installation procedure are described in detail. Specifically, we used SUMO to produce some traffic mobility and C++ code for the simulation. The files that are produced by this method are also conceivable by NS-3.
- The proposed protocol overview is presented in **Chapter 04**. In specific there is the definition of the problem, our protocol explanation, and some code parts of the algorithm.
- **Chapter 05**, contains the simulation network and the results that we obtained, which are mainly focused on the number of clusters concerning time , the stability of clusters in different vehicle densities, velocities, transmission ranges and environments. There are detailed charts and tables that represent them.
- Finally, we draw our conclusions about the proposed protocol in **Chapter 06**.

Chapter 2

Fundamentals of VANETs

First and foremost we have to present the basic terminology around VANETs before we move on to the implementation of our algorithm and techniques for clustering. The presented concepts are those of ad hoc networks, the differences between VANETs and MANETs, and their possible applications.

2.1 Ad-Hoc Networks

A wireless ad hoc network(WANET) is a decentralized type of wireless network. Their basic characteristic is that there is no fixed infrastructure to sustain the network. An ad-hoc network can be considered as a set of individual devices, communicating with each other directly if they are in the transmission range of each other, and such nodes are called neighbors. However, an ad hoc network must also ensure communication between nodes that are only indirectly connected. Any pair of non-neighbor nodes communicate through intermediate nodes forming the backbone of the network. Figure 2.1 demonstrates an example of an ad hoc network. There are many techniques for backbone formation, including clustering and flooding. Since the last one is achieved through the exchange of broadcast messages, many of them may be redundant and lead to collision and contention problems. Before analyzing the benefits of clustering we present the decentralized concept and its contribution to ad hoc networks[18].

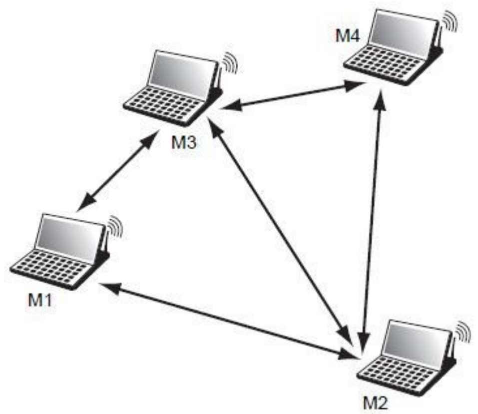

Figure 2.1: An example of ad hoc network

There do exist two major categories of networks based on their organizational structure. They are classified into centralized networks and decentralized ones. Centralized network architecture is built around a single node or group of nodes that handle all the major processing and operate as an agent for all communications. Of course, it is a requirement for such nodes to be aware of the exact topology of the network. On the other hand, in decentralized network architecture, the workload is distributed among several machines, instead of being dependent on a single point. Activities regarding planning and decision-making, are distributed among multiple nodes of the network. So, it arises that decentralization is more appropriate for ad hoc networks because there is no pre-existing infrastructure and the nodes are not able to know the exact topology of the network. Some key advantages to decentralized networks are the increased system reliability, as there is no single point of failure, scalability, and privacy.

Moreover, wireless ad hoc networks can be divided by their application into the following categories:

1. Mobile Ad hoc NETworks(MANETs)
2. Vehicular Ad hoc NETworks(VANETs)
3. Wireless Mesh Networks(WMN)
4. Wireless Sensor Networks(WSN)

A ***mobile ad hoc network*** (MANET) is a collection of mobile nodes that act as both routers and hosts in an ad hoc wireless network and that dynamically self-organize in a wireless network without the support of any pre-established infrastructure[13]. This type of network is described extensively in the following sections.

A ***vehicular ad hoc network*** (VANET) is a subcategory of MANETs. In this case of network nodes are exclusively vehicles. More details for VANETs are given below.

A ***Wireless Mesh Networks***(WMNs) consist of mesh routers and mesh clients that communicate with each other to share the network connection across a large area. The mesh clients are often laptops, cell phones, and other wireless devices.

A ***wireless sensor network***(WSN) can be considered as a finite set of spatially distributed sensors for monitoring and recording the physical phenomena and organizing the collected data at a central point. WSNs provide measurements about environmental conditions such as temperature, pollution levels, humidity, and wind[10].

2.2 MANETs and VANETs

As it was mentioned above, the acronym MANET refers to a set of wireless devices, equipped with transceivers which forward and receive packets, forming a spontaneous self-arranged network with no central management. The main characteristic of such nodes is their high mobility. They are allowed to move any time in any direction, a fact that leads to the frequent change of the links between them.

There is a wide range of algorithms for MANETs to increase the network efficiency and stability of the system. Many of them aim to minimize the number of exchanged messages between the nodes to avoid collision and contention problems.

VANET is a particular type of MANET. In specific, the term VANETs stands for Vehicular Ad hoc Networks and share common characteristics with MANETs as the nodes are mobile and the network is self-organized in this case too. It consists of vehicles and Road Side Units (RSUs) equipped with radios. The infrastructure of a VANET is displayed in Figure 2.2. The main difference between MANETs and VANETS is that the last one has the constraint of fast topology changes due to high speeds of vehicles, and high node mobility, but we have the extra limitation of movement within predefined roads[30].In addition, the number of nodes is much larger in such networks. In order to manage these special characteristics that distinguish VANETs of MANETs, it is required to design different algorithms and protocols.

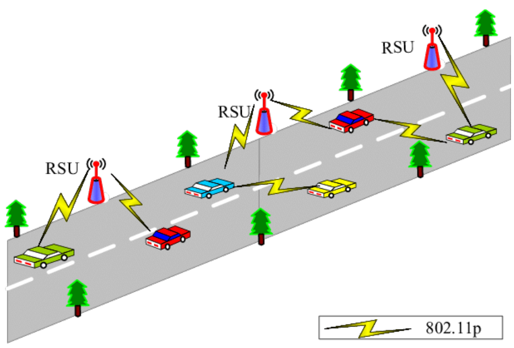

Figure 2.2: An example of Vehicular Ad Hoc Network

2.2.1 Comparison between MANETs and VANETs

MANETs and VANETs have a lot in common, but there are several differences that distinguish both environments. In Table 2.1 below, it is presented the comparative main characteristics of both networks which were originally cited on the paper.[7]

Parameters	**MANET**	**VANET**
Cost of production	Cheap	Expensive
Change in network topology	Slow	Frequent and very fast
Mobility	Low	High
Node density	Sparse	Dense and frequently variable
Bandwidth	Hundred kps	Thousands kps
Transmission range	Up to 100m	Up to 500m
Node lifetime	Depends on power resource	depends on lifetime of vehicle
Multihop routing	Available	Weakly available
Reliability	Medium	High
Moving pattern of nodes	Random	Regular
Addressing scheme	Attribute based	Location based
Position acquisition	Using ultra sonic	Using GPS,RADAR

Table 2.1: Comparison between MANETs and VANETs regarding with specific parameters, as presented on "A comparative study of MANET and VANET Environment".

2.2.2 Types of vehicles on a VANET

Depending on the vehicular environment(City or highway) the vehicular network consists of different types of vehicles.

For instance, in an urban environment, the network usually consists of cars, buses, motorcycles, taxis and ambulances, and so forth. On the other hand in the highway, we observe vehicles such as cars, motorcycles, buses, and trucks. In the following section, we present some characteristics of electric cars, a special type of vehicle.

2.3 Electric Vehicles

Electric Vehicles(EVs) are a particular category of vehicles because of their special characteristics. Electric cars have one or more electric motors instead of internal combustion engines. They also have a battery and must be parked at a charging station to provide power. It is worth noting that as it uses only electricity instead of fuel to operate, they are an environmentally friendly solution.

2.3.1 Contribution to the environment

Generally, the amount of infection that EVs produce depends on how electricity is produced, as there are no fuel emissions but heat emissions. Electric vehicles powered by renewable energy sources(sun, wind) are emission-free. Energy conservation is important, and CO_2 emission reduction is also essential to reduce the pollution of environment[27]. But even the worst way of producing electricity causes less infection than conventional vehicles.

2.3.2 Driving range

Current electric vehicles have a smaller driving range (per charge) compared to conventional vehicles (per fueling). First of all, efficiency depends on driving conditions. Excessive outside temperatures tend to reduce this range as more energy is needed to heat or cool the vehicle. Another important factor that affects the high driving speeds, due to the energy required to overcome the increased reverse(force opposing the movement of a body). In terms of vehicle acceleration, fast acceleration reduces this range. Finally, the loading of heavy loads in combination with the weight of the vehicle and driving on important slopes also reduces the range[5].

2.4 Energy consumption

2.4.1 External conditions that affect the energy consumption

Generally, energy consumption is affected by driving patterns(vehicle speed profile), the external driving environment, and both vehicle-to-vehicle(V2V) and vehicle-to-infrastructure (V2I) communication.

The largest amount (up to 62%) of energy, is consumed in engine failures due to friction, pumping, air circulation, and heat waste.17% is lost due to the inactivity of the car in the urban areas. In such cases, there is no steady flow in the movement because of road signs, traffic lights, and junctions that cause the switch of the car(turned on-turned off). Furthermore accessories like air-conditioning, power steering and use up to 2% of vehicle energy. Additional factors of energy consumption in EVs are driveline, brakes, aerodynamic, and slip resistance[12].

2.4.2 Energy related to V2V communication

In vehicle-to-vehicle communication, the energy consumption is attributed to the wireless node and it performs in a similar way to the energy consumption of a mobile node. As the amount of exchanged messages becomes larger, the remained power reduces.

2.5 Applications of VANETs

In vehicular ad hoc networks, vehicles are equipped with short-range radios in order to communicate with each other and possibly with roadside infrastructure to enable a range of applications from Internet access and driver assistance to transportation safely an emergency response. They designed to enhance safety, driving efficiency, and make the driving experience and navigational decisions more comfortable. They are necessary for security, safety, rescue, exploration, military and communication redundancy systems in non-populated areas, besides its ordinary use in urban environments as an essential part of intelligent transportation systems (ITS). A large number of car accidents could be avoided with sufficient warnings. Additional important applications on VANETs are the prevention of collisions and real-time traffic conditions and monitoring[29].

2.6 Limitations on VANETs

Vehicular ad hoc networks pose many challenging research issues, due to the limitations that characterize them. Some of these are presented below[4].

1. High mobility of nodes
2. Security and Privacy concerns
3. Signal loss
4. Decreased latency
5. rapidly changing network topology
6. Dynamic topology, where it is hard to find out malignant nodes
7. Bandwidth constraints in cellular environments
8. Energy constraints
9. Transmission errors
10. Pinpoint the location of vehicle
11. Unreliable channel conditions

2.7 Routing strategies in VANETs

The main requirement of routing protocols is to achieve minimal communication time with minimum consumption of network resources. Many routing protocols have been developed for Mobile Ad Hoc Networks (MANETs), and some of them can be applied directly to VANETs. However, simulation results showed that they suffer from poor performances because of the characteristics that distinguish VANETs from MANETs. So tracing and maintaining routes is a very challenging task in VANETs.[11]

Routing protocols that use geographical location information obtained from street maps, seem like a promising solution for VANETs. Comparisons between topology-based routing and **position-based** routing identify the last one more suitable strategy for VANETs. GPSR (Greedy Perimeter Stateless Routing) is one of the best known position-based protocols in the literature[22].

Broadcast-based routing : Broadcasting is an extremely useful routing technique in vehicular ad hoc networks. Nodes share data about road conditions,traffic, and safety messages. This method is achieved by implementing flooding. In a network, flooding is the forwarding by a router of a packet from any node to every other node attached to the router except the node from which the packet arrived. Flooding ensures that the packet will eventually reach all nodes in the network. However, this method overloads the network with redundant messages. As a result, when the number of nodes in the network increases, the performance drops dramatically[17].

The most efficient routing strategy in VANETs seems to be the cluster-based routing which will be discussed extensively in the sections below.

2.7.1 Methods for retrieving position in VANETs

The techniques motioned above require to receive some position information in order to function properly. Each vehicle must be able to decide either their exact location or its position related to other vehicles. The most widespread methods for this are :

- GPS (Global Positioning System)
- Doppler effect from where we export the Doppler Value
- RSS (Received Signal Strength)

2.8 Clustering

As already mentioned, VANETs are a special class of MANETs, which provide a distinctive approach for intelligent transport systems (ITS)[28], [25]. So they inherit all their characteristics, including the infrastructure-less function of network and inability of knowing the exact position among nodes. In order to improve communication of nodes for the exchange of such information, we form stable clusters by grouping the vehicles.

Clustering is the process of associating a set of objects in such a way that objects in the same group, called a cluster, act as a single system and operate similar tasks than those in other clusters. Nodes that belong to a cluster are allocated geographically adjacent and communicate via direct links[11].

All the nodes are divided into one of the following 3 categories according to certain rules, which different from algorithm to algorithm.

1. **Cluster-head(CH)**: each cluster can have a cluster-head, which is responsible for intra-and inter-cluster coordination in the network management functions.
2. **Gateways(GW)**: they disseminate the message to their neighboring cluster and they are used for the communication between clusters.
3. **Ordinary nodes(or cluster members)(CM)**: they just join a cluster and receive the message within it without rebroadcasting it to other nodes.

Worth mentioning is the fact that during the formation of clustering in a highly dynamic network, we may have some nodes in the undecided state which are mainly ex-members of a cluster that currently don't belong to any cluster. Figure 2.3 presents the architecture of a cluster network[26].

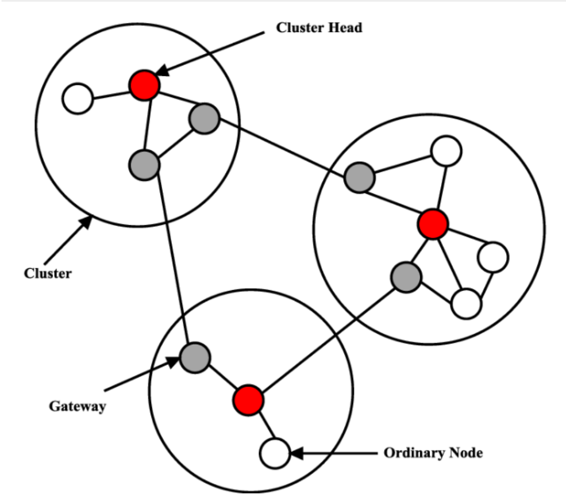

Figure 2.3: Cluster architecture. Red nodes represent cluster-heads, grey nodes are gateway nodes and blank nodes are cluster members respectively

2.8.1 Clustering techniques

The specific functions of the cluster head differ depending on the application, as does the method by which they are selected[16],[6].

Each node within the network owns a unique identifier(ID). Algorithms for the formation and maintenance of clusters are classified into 3 categories.

a)Lowest-ID algorithm: This scheme favors nodes with lower identifiers to become CHs without taking in mind the mobility patterns of the nodes. Nodes periodically broadcast the list with nodes that can hear including its own id. The node with the minimum id becomes the cluster head in its own neighborhood.

b)Highest-Degree algorithm: The degree assignment to each node results from the distance from the other nodes of the network. The node with the maximum degree selected to be cluster head.

c)Beacon-Based algorithm: Based on the periodical transmission of hello messages that contains information related to the state of a node. Based on the state of the neighbors, a node can select its own state. A cluster head will consider a change of its state if it receives a message from another cluster head. A cluster head receiving a hello message from another CH will remain in the same state if it has more cluster members of its own than the sender.

The last method is the only applicable in real vehicular environments.

2.8.2 Benefits of clustering in VANETs

There are many advantages of applying clustering techniques in VANETs. These benefits are related to spatial reuse of resources, which solves the problem with limited bandwidth which suffer MANETs/VANETs. Cluster heads and gateways form the backbone of the network, in order to avoid flooding which improves the total performance of the network. Also if one node fails, the workload is redistributed among the other vehicles for uninterrupted operation. Clustering is also beneficial for stability and scalability. Theoretically, there is no limit on the number of machines that can belong to the cluster. However, at this point, the tradeoff of cluster size and network performance is introduced. Another tradeoff is the period for re clustering and stability of the network.

In chapter 4 the concept of our proposed clustering algorithm for vehicular ad hoc networks is analyzed extensively.

Chapter 3

Simulation tools

To obtain the results of our algorithm we had to create a realistic environment where the vehicles could move with some velocity towards a random direction but into predefined roads.

The simulation of the road networks was made with tool SUMO. Its usage and the installation steps are discussed in more detail in the following pages. Furthermore, there is a brief demonstration of the simulation network used about receiving the results of our algorithm made with SUMO.

3.1 Simulation of Urban MObility (SUMO)

”Simulation of Urban MObility” (SUMO) is an open-source, highly portable, microscopic, and continuous traffic simulation package designed to handle large networks. It provides the ability of intermodal simulation including pedestrians and comes with a large set of tools for scenario creation. It allows simulating how a given traffic demand which consists of vehicles moves through a given road network. It is purely microscopic: each vehicle is modeled explicitly, has an own route, and moves individually through the network. Simulations are deterministic by default but there are various options for introducing randomness. SUMO is licensed under the EPL 2.0. [1].

Features
Space-continuous and time-discrete vehicle movement
Representation of different vehicle types.
Support of multi-lane streets with lane changing
Implementation of different right-of-way rules, traffic lights
A fast openGL graphical user interface is available

Table 3.1: Features of SUMO.

SUMO requires configuration and data files for proper execution. These files describe the traffic-related part of a map, the roads and intersections the simulated vehicles run along or across. SUMO network can be considered as a directed graph. Nodes, usually named ”junctions” in SUMO-context, represent intersections, and ”edges” represent roads or streets. Note that edges are unidirectional. Specifically, the SUMO network contains the following information:

1. every street (edge) as a collection of lanes, including the position, shape and speed limit of every lane
2. traffic light logics referenced by junctions
3. junctions, including their right of way regulation
4. connections between lanes at junctions (nodes)

Although being readable (XML) by human beings, a SUMO network file is not meant to be edited by hand. SUMO provides additional command-line applications like NETCONVERT that can also convert an existing map from various formats or generate geometrically simple abstract road maps with NETGENERATE or NETEDIT. Furthermore, SUMO is usually coupled to a communication network simulator using middleware which is necessary for our simulation.

3.1.1 Installation Steps

In this section the basic procedure for installing SUMO is briefly presented.

If the operational system(OS) is debian or ubuntu, SUMO is part of the regular distribution and can be installed like this:

- sudo apt-get install sumo sumo-tools sumo-doc

If you need a more up-to-date ubuntu version, it may be found in a separate ppa, which is added like this:

- sudo add-apt-repository ppa:sumo/stable
- sudo apt-get update
- sudo apt-get install sumo sumo-tools sumo-doc

To be able to run SUMO on Linux, just follow these steps:

1. Install all of the required tools and libraries
2. Get the source code
3. Build the SUMO binarie

For ubuntu this boils down to:

- sudo apt-get install cmake python g++ libxerces-c-dev libfox-1.6-dev libgdal-dev libproj-dev libgl2ps-dev swig git clone –recursive https://github.com/eclipse/sumo
- export SUMO_HOME="$PWD/sumo"
- mkdir sumo/build/cmake-build && cd sumo/build/cmake-build cmake ../..
- make -j$(nproc)

The nproc command gives you the number of logical cores on your computer, so that make will start parallel build jobs which makes the build a lot faster. If nproc is not available on your system, insert a fixed number here or leave the option out. You may also try:

- make -j $(grep -c p̂rocessor /proc/cpuinfo)

For the correct setting of SUMO_HOME you have to remember the correct path, where you build your SUMO, the SUMO build path. This path is shown with pwd at the end of getting the source code. If you want to develop actively on sumo we strongly recommend to use the git repository.

Repository checkout (recommended)

- git clone –recursive https://github.com/eclipse/sumo

- cd sumo
- git fetch origin refs/replace/*:refs/replace/*
- pwd

The additional fetch of the replacements is necessary to get a full local project history.

Release version or nightly tarball

We used sumo 1.4.0 version for our simulation. Download sumo-src-1.4.0.tar.gz or other versions and do the following:

- tar xzf sumo-src-version.tar.gz
- cd sumo-version/
- pwd

Definition of SUMO_HOME

- export SUMO_HOME="/home/user/sumo-version"

You can check that SUMO_HOME was successfully set if you type

- echo $SUMO_HOME

Installing the SUMO binaries(optional step) If you want to install the SUMO binaries into your system, run

- sudo make install

You have to adjust your SUMO_HOME variable to the install dir (usually /usr/local/share/sumo)

- export SUMO_HOME=/usr/local/share/sumo

We can also implement and evaluate V2X communication technologies by coupling to a communication network simulator (OMNeT++ or ns-3). In our case ns-3 is used and is extensively described to the next section.

3.2 NS-3

NS-3 is a discrete-event network simulator, targeted primarily for research and educational use. NS comes from Network Simulator, it is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use [2].

3.2.1 Simulation workflow

The general process of creating a simulation can be divided into several steps:

1. Topology definition: To ease the creation of basic facilities and define their interrelationships, ns-3 has a system of containers and helpers that facilitates this process.
2. Model development: Models are added to simulation (for example, UDP, IPv4, point-to-point devices and links, applications); most of the time this is done using helpers.
3. Node and link configuration: models set their default values (for example, the size of packets sent by an application or MTU of a point-to-point link); most of the time this is done using the attribute system.
4. Execution: Simulation facilities generate events, data requested by the user is logged.
5. Performance analysis: After the simulation is finished and data is available as a time-stamped event trace. This data can then be statistically analysed with tools like R to draw conclusions.
6. Graphical Visualization: Raw or processed data collected in a simulation can be graphed using tools like Gnuplot, matplotlib or XGRAPH.

3.2.2 Mobility Generation

Producing traffic can be achieved by using SUMO and then use the TraceExporter, a SUMO's extension, to export SUMO traces. TraceExporter is written in Java and by its application there are produced three files: config, activity and mobility. The procedure about this this is given below:

To define a network in SUMO at least two files are needed:

- nod.xml for nodes
- edg.xml for the streets between them

Let's name these files ”mynetwork.nod.xml” and mynetwork.edg.xml respectively. These files are used to export the ”mynetwork.net.xml” file. We might need to add the “mynetwork.typ.xml” file in the creation of the “mynetwork.net.xml” to include different types of vehicles. Then we can combine the “mynetwork.net.xml” file with a “mynetwork.rou.xml”(which contains the routes of our vehicles) by calling the sumo commands below:

- netconvert –node-files=mynetwork.nod.xml –edge-files=mynetwork.edg.xml –type-files=hello.typ.xml –output-file=mynetwork.net.xml
- sumo-gui -n mynetwork.net.xml -r mynetwork.rou.xml –netstate-dump netstate.xml

In the path “tools/traceExporter” of SUMO, there the file “traceExporter.jar”. We copy this file into the same folder as the pre-mentioned sumo files and we do the following:

- java -jar traceExporter.jar ns2 -n mynetwork.net.xml -t netstate.xml -a activity.tcl -m mobility.tcl -c config.tcl -p 1 -b 0 -e 150

-n net_file
-t trace_file
-a activity.tcl, describes the time that each vehicle does its first and last movement.
-m mobility.tcl, contains the actual movements of the nodes.
-c config_file is used to set some optional variables which describe the simulation scenario
-p penetration_level
-b begin_time
-e end_time

Penetration level means the ratio of selecting vehicles that will be traced for output and is a float in [0, 1]. For instance, penetration level of 1 equals to no filtering and 0 means no vehicles will be selected.

The mobility.tcl file is necessary for ns-3 and for our C++ code and we rename it to “mynetwork.ns_movements” for using it.

Chapter 4

Implementation of EAVC

The efficient data dissemination is a big challenge. In recent years, with such many applications supported by VANETs, there is an increasing demand for downloading a large amount of content to the vehicles. However, when the concept of cellular networks is introduced, things become more complicated. A cellular network is not capable of supporting a large number of user terminals, due to its limited bandwidth in a dense vehicle environment.

In chapter 2 we extensively presented the benefits of clustering in VANETs. Content distribution becomes more efficient by grouping vehicles into clusters. Cluster heads and gateways, forming the backbone of the network, own the primary role for data dissemination among vehicles, so they should be selected carefully. Clustering is also essential for decreasing the delay of data propagation.

4.1 Related work

In literature, there have been many algorithms applied on VANETs for efficient content distribution. The vehicle to cloud communication is done by combining V2V with vehicle-to-infrastructure (V2I) communication, where roadside units (RSUs) provide Internet access to the vehicles. Due to the high node mobility, the performance of data dissemination degrades because of the limited link lifetime between the vehicle and RSU[19]. A recent survey on content downloading can be found in [21]. Li et al. have discussed the use of broadcast communications for content distribution in VANETs. Although broadcast communications can distribute the same content to multiple receivers, it overloads the network with redundant messages. There is a proposal, specifically GESC protocol[24], which takes into consideration the topological relations of nodes to detect who is more suitable in operating communication tasks between clusters, but it is unappropriated for highly mobile nodes, such as vehicles.

Concerning the route creation and backbone formation Togou et al.[20] have proposed CDS-SVB, a connected dominating set (CDS) based stable virtual backbone creation algorithm which creates routes by considering vehicle speed and their distribution in the whole network. This undoubtedly ensures the stability of the system. Since backbones are generated one-by-one, the backbone formation algorithm of CDS-SVB is not fully distributed.

The proposal that we relied on for the larger part of our implementation was the[30] which achieves content distribution through a two-level clustering approach integrating LTE and IEEE 802.11p for V2V communications. The extra factor that we introduce is the energy factor, which is calculated provided that we know the remaining energy of vehicles. This makes our protocol innovative as there is not such a combination in literature so far.

4.2 Proposed algorithm

RSUs are responsible for providing Internet to the vehicles. However, RSU deployments are expected to be costly and also V2R based VANETs suffer form fast fading and short connectivity time, which are caused by the high relative-speed difference between fast-moving vehicles and the stationary RSUs. Besides, V2V-based VANETs are independent of road-side conditions, a fact that makes the network more flexible particularly in rural areas where roadside infrastructures are not always applicable [15]. We hence consider a protocol that does not count on RSUs. An alternative solution is to use exclusively Long Term Evolution(LTE) interface installed on vehicles for providing Internet access to the VANETs. However, once again this is not practical due to the cost of cellular transmissions and limitations of the spectrum in cellular networks.

After examining an adequate number of clustering protocols in VANETs we concluded to introduce a hybrid architecture of **LTE** and **IEEE 802.11p**, based on ”Cluster-Based Content Distribution Integrating LTE and IEEE 802.11p with Fuzzy Logic and Q-Learning” from Celimuge Wu and Tsutomu Yoshinaga, Xianfu Chen, Lin Zhang, and Yusheng Ji paper[30].

IEEE 802.11p is an approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments (WAVE), a vehicular communication system. It defines enhancements to 802.11, required to support Intelligent Transportation Systems (ITS) applications. This includes data exchange between high-speed vehicles and between the vehicles and the RSUs.

There are two main technical obstacles to integrating LTE with IEEE 802.11p. On the one hand, the election of cluster heads and gateway nodes should bear in mind the overall network performance. On the other hand, the route creation from a vehicle to a gateway is demanding due to high vehicle mobility and varying node density. We need an approach that takes into account jointly the vehicle mobility, vehicle distribution, and the link qualities between vehicles.

Many parameters affect the node distribution. More specifically, for certain hours or road segments, vehicles are densely deployed, and therefore number of vehicles is expected to be huge[17]. As in IEEE 802.11 so in the 802.11p standard, the larger the number of concurrent sending nodes, the lower the efficiency of the network due to the exponential backoff based contention scheme at the MAC layer[9].

The implementation of our algorithm divided into two parts. We propose the EAVC (Energy Aware VANET Clustering), a two-level clustering approach where in the first level a fuzzy-logic algorithm is introduced to solve the MAC layer contention in IEEE 802.11p-based V2V communications under a high-density condition, and in the second level a Q-learning algorithm is used for selecting gateway nodes which bridge V2V and LTE . Firstly, we have to mention that in order to form and maintain clustering structures, nodes in some pairs have to exchange hello messages periodically. The period time for these messages to be exchanged depends on the mobility patterns of the nodes. In our case this this predefined interval is 1 sec by default. Furthermore, there is no need for joining/leaving messages. So the network does not overloaded with redundant messages.

This Chapter is dedicated to the description of our EAVC. In terms of structure, firstly we present the problem. Then we continue in basic equations and figures and finally the pseudocode of our algorithm is presented.

4.3 Assumptions

On the following algorithm, the technique that was used to retrieve position information is GPS. Each node is also equipped with two wireless interfaces, namely, LTE interface and IEEE 802.11p interface respectively. All nodes are aware of the road map information and the average transmission range for V2V communications(R), based on euclidean distance.

Each node sends its own position , the number of vehicles driving toward the same direction ,its remaining energy, and velocity information using hello messages. We achieve unicast communications for V2V communications which are easier to conduct retransmissions as compared to broadcast communications and make our model more reliable.

4.4 EAVC overview

We have to come up with the problem of forwarding packets from the LTE Base Station(BS) to the vehicles.

In our model we include two types of cluster heads:

1. **Edge Cluster Head**: responsible for providing the IEEE 802.11p communications between its cluster members and the gateway cluster heads

2. **Gateway Cluster Head**: provide the LTE communications to other vehicles.

As shown in Fig 4.1 , the contents are transmitted from an LTE BS to a gateway node, and then distributed to multiple vehicles. In our case only the gateway nodes utilize LTE interface and contact with other vehicles through IEEE 802.11p interface for V2V communications.

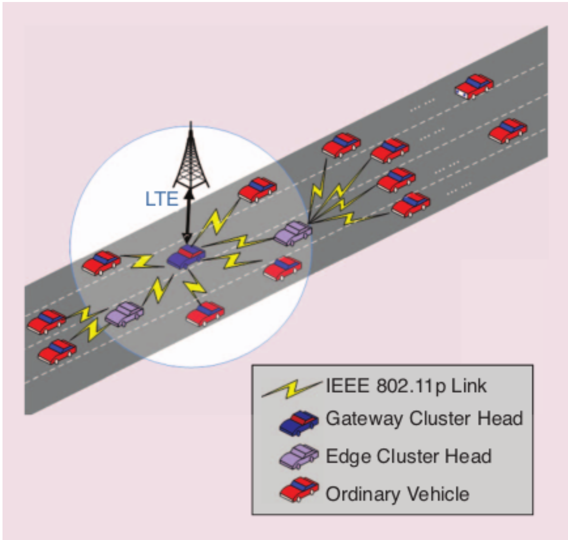

Figure 4.1: Cluster-based content distribution integrating LTE and 802.11p interface

The fuzzy-logic algorithm in the first level is responsible for selection of the Edge Cluster Heads and ensures that these CHs are stable due to Velocity,Leadership,Signal Quality and Energy factors that take into consideration for this choice. Each edge cluster head node applies a Q-learning algorithm to decide whether to act as a gateway node or not. This is accomplished through the reward that is allocated from BS according to the number of connected vehicles. More details for this reward are given in the following pages.

4.5 First level Fuzzy Logic Algorithm

As previously mentioned, we do not use any cluster joining/leaving messages for the maintenance of cluster member information. After cluster heads are determined, each cluster head announces the number of cluster members using the hello messages. The selection of Edge Cluster Head nodes is managed through hello messages exchange. Each vehicle inform their neighbors for :

1. its velocity .
2. the number of vehicles moving toward the same direction as the current vehicle
3. the link quality between the vehicle and the other nodes

4. its remaining energy

The first two factors are used to ensure that the generated cluster heads are stable. The third factor gives a higher priority for vehicles which could provide better channel conditions to their cluster members to selected as CHs. For instance, vehicles such us buses or trucks higher antenna can provide longer line-of-sight distance[14].The forth factor is also used in for priority reasons in order to none CH run out of energy and lead to the reclustering. Upon reception of a hello message, each node calculates a competency value for itself and its one-hop neighbors(in range$\frac{R}{2}$). This value determines which nodes are more suitable for being CHs. More specifically, the node with the largest value in its $\frac{R}{4}$ region claimed itself as CH by using hello messages. This ensures that at least 2 Cluster Heads would exist in R region, fact that makes the communication between 2 clusters reliable.

4.6 Why velocity is an important factor in VANETs clustering?

We have to well-considering of the factors which we include in the design of a clustering scheme. The degree of the velocity difference among neighboring vehicles is the key issue for constructing relatively stable clustering topology. A lower relative velocity simply means that the neighbors of a certain node have spent a longer time in its transmission range. Consequently, we can conclude that the mentioned node as CH ensures a more stable system. [23]

In the cluster creation procedure, nodes that driver intentions can be predicted like truck drivers that keep an almost constant velocity must be favored to become clusterheads, due to being more stable in terms of mobility.[17]

4.6.1 Velocity Factor(VF)

In order to include velocity of vehicles in the proposed protocol, Celimuge Wu and Tsutomu Yoshinaga, Xianfu Chen, Lin Zhang and Yusheng Ji formed the velocity factor(VF) in the competency value calculation. Upon the reception of a hello message from node m, node s calculates :

$$VF(s,m) = \frac{|v(m)| - \min_{\gamma \in N_s} |v(\gamma)|}{\max_{\gamma \in N_s} |v(\gamma)|} \tag{4.1}$$

where N_s is the neighbor set of node s , and v($\cdot$) denotes the velocity. Thus, the lower the factor, the lower the velocity of node m.

We update this factor periodically with the interval of one second based on a weighted exponential moving average:

$$VF_i(s,m) \leftarrow (1-\alpha) \times VF_{i-1}(s,m) + a \times VF_i(s,m) \tag{4.2}$$

where i is the interval index, $VFi-1(s,m)$ and $VF_i(s,m)$ indicate the previous value and current value of VF respectively.

We initialize $VF_0 = 1$, and the updating factor α is constantly set to 0.7 .

4.7 Why is it important to include only nodes that are moving towards the same direction in the same cluster?

As in the case of velocity, so the same direction is a factor that enhances the stability in clustering formation. Considering only vehicles that moving towards same direction helps maintaining the existing clusters for a longer period. Communication links are more durable between such nodes. Vehicle distribution is a crucial factor in clusterhead selection.

4.7.1 Leadership Factor(LF)

So, we introduce the leadership factor(LF) based on [30] in our model. Actually, each node in vicinity sends the density of nodes that are moving towards the current vehicle in the hello message. This parameter is taken into consideration to the formula of this factor:

$$LF(s,m) = \frac{c(s)}{\max_{\gamma \in N_s} c(\gamma)} \tag{4.3}$$

where c(s) denotes the number of vehicles moving toward the same direction with the node s. A higher LF leads to better CH selection. We initialize LF to 0. In every hello message reception(1 sec) LF is updated in the same way as VF:

$$LF_i(s,m) \leftarrow (1-\alpha) \times LF_{i-1}(s,m) + a \times LF_i(s,m) \tag{4.4}$$

4.8 Signal Quality Factor(SQF)

Link qualities in V2V communications can be very bad due to multipath fading, shadowing and Doppler shifts cased by the high mobility of nodes[15]. Thus, signal qualities between vehicles should be also taken into consideration in the first-level algorithm to increase efficiency of our system. Hence, we introduce the Signal Quality Factor (SQF) for channel evaluation. In its calculation procedure we use the hello packet reception ratio.

Each node maintains a counter for the number of hello messages received from all neighbors within R distance. In this way vehicles can calculate its own reception ratio of the hello messages. This ratio is used to estimate the channel condition among vehicles.

SQF is defined as follows:

$$SQF(s) = \frac{cntReceived}{totalSent} \tag{4.5}$$

where the cntReceived indicates the number of hellos received by the neighbors and the totalSent is the number of hellos sent by the neighbors. The initial value of SQF is 0 and with the updating interval 1 sec :

$$SQF_i(s,m) \leftarrow (1-\alpha) \times SQF_{i-1}(s,m) + a \times SQF_i(s,m) \tag{4.6}$$

4.9 Energy Factor

We previously refered to the features that MANETs and VANETs have in common. A CH selection procedure in MANETs includes above all the power consumption of nodes. This motivates us to introduce the parameter of remaining energy of electric and non-electric vehicles to the clustering formulation. This extension leads to stability and efficiency enhancement.

This parameter is initialized to the maximum energy that a vehicle is able to store and it changes according to the factors that we described in chapter 2, including motion of the vehicle and V2V communication.

Since the energy consumption is affected by the transmission and reception of a message, our model is considered as energy efficient as it contains no join/leaving the cluster messages and the overhead is low.

In every hello message exchange the energy of both electric and non-electric vehicles is exponentially decreasing.

4.10 Neighbor list

Each node will allocate a neighbor list, which has a neighbor list entryN_i^j for every neighbor in $\frac{R}{2}$ range. The elements that list contains are given below:

$$N_i^j = \begin{cases} ID_j : \text{node id(j)} \\ (x,y)_j : \text{position of node j} \\ (v_x, v_y) : \text{velocity of node j} \\ vf_j : \text{velocity factor of j} \\ lf_j : \text{leadership factor of j} \\ sqf_j : \text{signal quality factor of j} \\ ch_j : \text{cluster head of j} \\ ef_j : \text{energy factor of j} \end{cases}$$

Each node will periodically broadcast a HELLO beacon at a predefined interval (1sec by default), which will contain its ID, position, velocity, number of vehicles that moving toward the same direction as the current vehicle,and its remaining energy. Let's name this interval HELLO_T. Each time that a node transmits a HELLO message we increase by 1 the counter for sent messages, in order to utilize it to the hello reception ratio calculation. Similarly when node i receives a HELLO message message from node j, received messages' counter is increased by 1 and node i calculates its current factors with the corresponding formulas, given by equations 4.2,4.3 and 4.4 and then updates these factors and its neighbor list with the received data. The basic procedure is :

Algorithm 1 Factors' Calculation and Update

```
cntSent ← 0
cntReceived ← 0
while True do
    if HELLO_T then
        for each node in R/2 range do
            node[i].send(helloMsg)
            cntSent++
        end for
    end if
    if node_i.receive(helloMsgfromnode_j) then
        cntReceived++
        Calculate VFij, LFij, SQFi,EFij
        Node i adds/updates its neighbor list entry N_i^j
        Node i updates VFijj,LFij, SQFi,EFij
    end if
end while
```

4.11 Fuzzification and fuzzy rules

Since we calculated the three factors, in this step we use predefined membership functions to convert these factors to fuzzy values, and predefined fuzzy rules to calculate the final fuzzy value for each neighbor. The fuzzy membership functions are defined as shown in Figure 4.2. The linguistic variables of the VF are defined as Slow, Medium, Fast. Similarly, the linguistic variables for the LF and SQF are defined as Good, Fair, Poor and Good, Medium, Bad respectively. The energy factor follows different calculation procedure. We first calculate the rank and we include it to the competency value calculation.

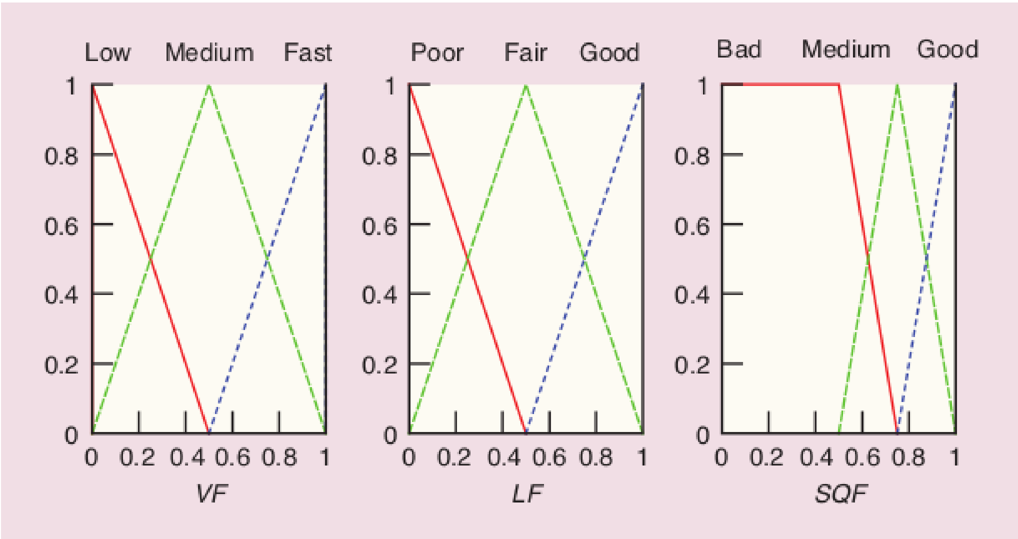

Figure 4.2: Fuzzy membership functions. To the left VF is represented , LF in the middle and SQF to the right respectively

Each node calculates the rank (a competency value for selected as CH) of each neighbor based on the IF/THEN rules as described in Table 4.1. The linguistic variables for the rank are defined as Perfect, Good, Acceptable, Unpreferable, Bad, Very Bad.

Algorithm 2 Competency Value Calculation

```
for each node in VANET do
    for each node in R/2 range do
        Calculate competency value
    end for
end for
```

We use the Min-Max method to calculate the evaluation results. More specifically, we use the minimal value of the antecedent as the final degree for each rule. For combing the degrees of multiple rules, we take the maximal value of multiple rules as the final degree.

Rule index	**VF**	**LF**	**SQF**	**Rank**
1	SLOW	GOOD	GOOD	PERFECT
2	SLOW	GOOD	MEDIUM	GOOD
3	SLOW	GOOD	BAD	UNPREFERABLE
4	SLOW	FAIR	GOOD	GOOD
5	SLOW	FAIR	MEDIUM	ACCEPTABLE
6	SLOW	FAIR	BAD	BAD
7	SLOW	POOR	GOOD	UNPREFERABLE
8	SLOW	POOR	MEDIUM	BAD
9	SLOW	POOR	BAD	VERYBAD
10	MEDIUM	GOOD	GOOD	GOOD
11	MEDIUM	GOOD	MEDIUM	ACCEPTABLE
12	MEDIUM	GOOD	BAD	BAD
13	MEDIUM	FAIR	GOOD	ACCEPTABLE
14	MEDIUM	FAIR	MEDIUM	UNPREFERABLE
15	MEDIUM	FAIR	BAD	BAD
16	MEDIUM	POOR	GOOD	BAD
17	MEDIUM	POOR	MEDIUM	BAD
18	MEDIUM	POOR	BAD	VERYBAD
19	FAST	GOOD	GOOD	UNPREFERABLE
20	FAST	GOOD	MEDIUM	BAD
21	FAST	GOOD	BAD	VERYBAD
22	FAST	FAIR	GOOD	BAD
23	FAST	FAIR	MEDIUM	BAD
24	FAST	FAIR	BAD	VERYBAD
25	FAST	POOR	GOOD	BAD
26	FAST	POOR	MEDIUM	VERYBAD
27	FAST	POOR	BAD	VERYBAD

Table 4.1: Rule base.

4.11.1 Defuzzification

After the computation of the fuzzy competency value we have to convert it to a numerical value, based on fuzzy output membership function. The output membership function is defined as in Figure 4.3. The Center of Gravity (COG) method [8] is used for the defuzzification.

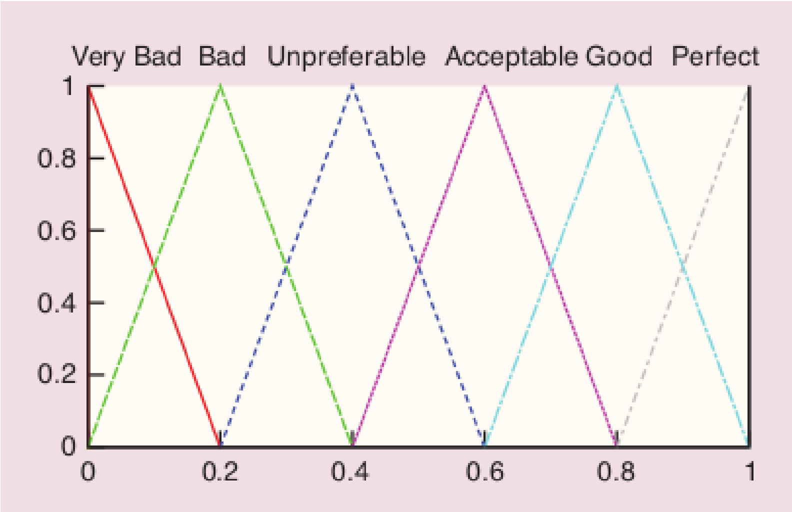

Figure 4.3: Output membership function.

After calculating the rank, we should combine it with the Energy Factor of each vehicle for better performance. The reason that we do not fuzzify the EF is that we desire to to give more weight to this factor than the other ones. The node with the largest Competency Value(CV) and maximum Energy Factor declares itself as CH by broadcasting hello messages in vicinity.

Algorithm 3 CH Declaration

for each node in VANET **do**
    for each node in vicinity **do**
        node_ch = nodes.competency_value.max() and nodes.energy_factor.max()
    end for
    hello_msg = [id, isCH = True]
    node_ch.send(hello_msg)
end for

4.12 Second level Q-learning algorithm

In the second level edge cluster head uses a Q-learning algorithm to decide whether to remain its own role or to act as a gateway node. The model for the Q-learning algorithm is defined as follows[30].

The edge cluster heads are the learning agents. Each agent learns the environment by exchanging hello messages with other agents. The action at each node is to choose the next node for the data dissemination. This next hop could be either an LTE BS or a neighbor edge cluster node. Each node stores a Q-value $[Q(s_t, m)]$ for each state and action.

More specifically, the state(s_t) identifies number of neighbor nodes of the current vehicle. For simplicity, as shown in Figure 4.4, we map the number of nodes to a discrete set 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, and use the corresponding value as the state. Moreover, C_{min} and C_{max} are defined as 5 and 45 by default. Allowed action for each agent is to choose next hop for transmission.

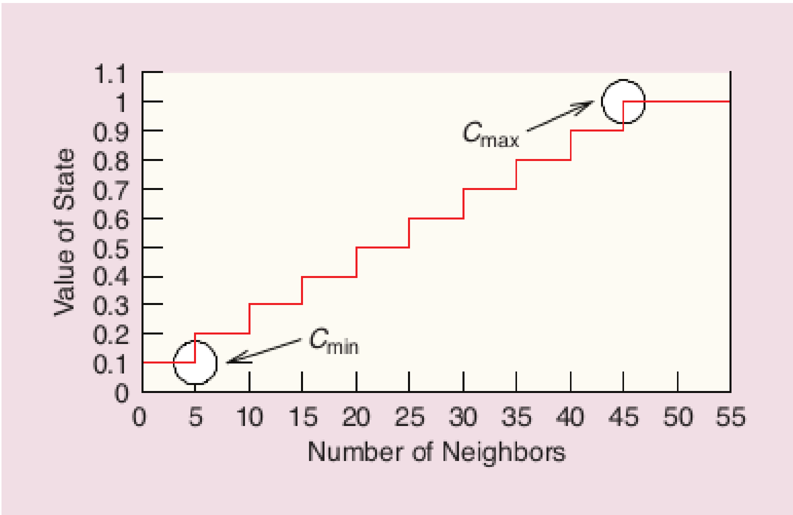

Figure 4.4: State mapping according to the number of neighboring vehicles

4.12.1 CH route selection

Upon reception of a hello message, each non-gateway cluster head node updates its Q -Table. In this step of algorithm a hello message contains the maximal Q -value and it is transmitted through IEEE 802.11 interface, as well as in the first-level algorithm. In contrast, since each gateway cluster head is directly connected to the BS, the corresponding Q-Table is updated independent of hello messages , but with the same periodic predefined interval(1 sec by default). We initialize each Q -value to 0. After receiving a hello message from node m , node l updates its own Q -value to the RSU as follows:

$$Q_l(s_t, m) \leftarrow \alpha \times HRR(l, m) \times Rwd + \gamma \times \max_{\gamma in N_m} Q_m(s_{t+1}, \gamma) + (1 - \alpha) \times Q_l(s_t, m) \quad (4.7)$$

where $HRR(l,m)$ is the reception ratio of hello messages sent from node m to node l. $Q_l(s_t,m)$ to the left denotes the Q-value of node l for the current state(s_t), while the $Q_l(s_t,m)$ on the right side of arrow shows the previous one. In addition N_m identifies the vicinity of node m and s_{t+1} is the next state.

The dumping factor γ equals to 0.9 and the learning rate α is 0.7 in this step too.The reward(Rwd) in equation 4.7 is defined as follows:

$$Rwd = \begin{cases} \min\left(\frac{BW_{LTE}}{BW_{11p}\times|N_{BS}|},1\right) & ,if l \in N_{BS} \\ 0 & ,otherwise \end{cases} \tag{4.8}$$

where BW_{LTE} is the available downlink bandwidth of LTE(the bandwidth through which node transmit data to BS), and BW_{11p} is the maximum achievable throughput of IEEE 802.11p link. BW_{LTE} equals to 300 Mbps and BW_{11p} is 27 Mbps respectively.

N_{BS} shows the neighbors of BS. Actually, the neighbor set includes the nodes that are connected to the BS. From equation 4.8 we obtain that if node m is a neighbor of the BS, the reward is positive,but in any other case this reward is 0.

In Rwd calculation we observe that the number of neighbors of nodes discount this factor. Combining equations 4.7 and 4.8 we conclude that a smaller hop count results in a larger reward and therefore in a larger Q -value. Hence, when the number of connected devices is large a vehicle can increase its reward by connecting to a gateway node instead of directly connecting to the BS. Additional discount factor for this Rwd reward is the packet loss probability of each link which constitutes the communication route.

The proposed protocol is willing to form the route which enhances the network performance in terms of integration of LTE and IEEE 802.11p.

4.12.2 Non-CH route formation.

A noncluster-head node selects a neighboring cluster head node (edge cluster head or gateway cluster head) as the next hop for data exchange instead of directly connecting with an LTE BS. The route selection is based on a Q-learning algorithm where the reward is distributed by the cluster head nodes. Each non-cluster-head node maintains a Q-Table in which each entry is corresponding to the value of using a neighboring CH as the next packet forwarding node.

Upon reception of hello messages(which contains the maximal Q-value and the node id) a non CH node updates its own Q-Table and then chooses the next hop which owns the largest Q-value. The direct Rwd in this case is 0 and so the equation 4.7 is defined for the Q-Table update at a non CH node :

$$Q_c(s_t,m) \leftarrow \alpha \times HRR(c,m) \times \gamma \times \max_{\gamma in N_m} Q_m(s_{t+1},\gamma) + (1-\alpha) \times Q_c(s_t,m) \tag{4.9}$$

Here c is the non CH node and m represents the neighboring CH node.

Chapter 5

Simulation Results

Given the fact that we desire to obtain some results as much as possible closer to real conditions, we had to create a realistic network where the vehicles could freely move in any direction with some velocity but in predefined roads. To be more accurate, we had to perform extensive simulations in various network topologies. As previously mentioned SUMO and ns-3 were used for the creation and simulation of such networks. In more detail, the traffic simulations are conducted with SUMO and the trace files are used in ns-3 to perform clustering. The description of the environment of the 2 networks used for the simulation, is given in the next section.

5.1 Simulation network

On both networks, we define 6 types of vehicles to take part in our simulation by using the SUMO tool. The types are differentiated in the maximum speed of vehicles, their height, their class. A type, for example, refers to electric vehicles, another one corresponds to trucks and buses or conventional cars, and so forth. The minimum distance between 2 vehicles is corresponding to the minimum allowed distance that they have to keep from the car that moves in the front. Vehicles moving with different random velocities among roads and lanes. Furthermore, the Nakagami propagation model was used to simulate channel fading[3]. We calculate the average transmission range for V2V communication, based on the euclidean distance. Although the transmission range can be up to 1000 m in IEEE 802.11p, this setting is more suitable for evaluating our protocol, as a longer distance could be difficult to use an efficient modulation and coding scheme. Moreover, the whole network is covered by one LTE Base Station(BS). The LTE bandwidth equals to 300 Mbps and the bandwidth of 802.11p is set to 27 Mbps.

5.1.1 Freeway model

The first network demonstrates the movement of nodes on a highway. The topology consists of 2 roads that are moving in a parallel way and towards the same direction. Hence we have two types of roads related to the number of lanes the allowed direction and maximum speed limit. The first road consists of 3 lanes and the second one consists of 2 lanes. Figure 5.1 represents the topology.

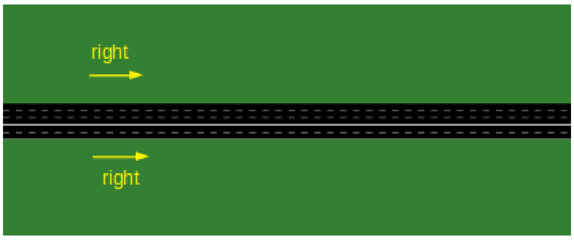

Figure 5.1: SUMO instance for the first network that consists of 2 parallel right-way roads. The first road includes 3 lanes while the second one includes 2 lanes.

5.1.2 City model

In the second case, the network includes 12x12 one-way edges that form squares. For instance, if a road permits right-way motion then the exactly previous and next one permit only left-way direction. Up and down directions operate in a similar way. The 12 horizontal edges are intersected with the 12 vertical ones, forming 144 possible intersections. In an intersection, a vehicle is able to change its direction or keeping its current one. Once again 100 nodes take part during the simulation. The topology is demonstrated in Figure 5.2.

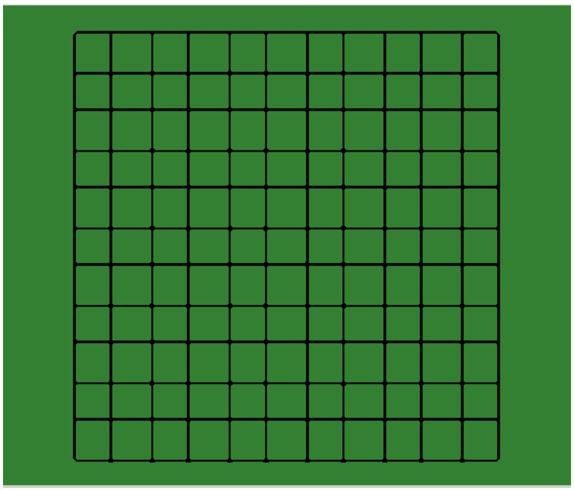

Figure 5.2: SUMO instance for the second topology. 12X12 roads are represented, 12 vertical and 12 horizontal that form squares. Their direction is alternately change.

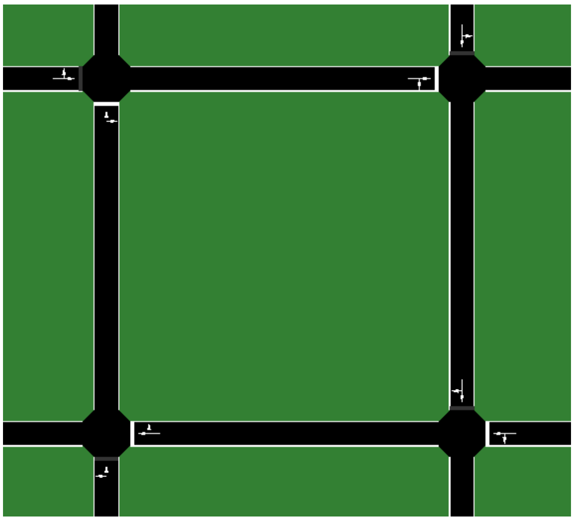

Figure 5.3: SUMO instance for the second network that demonstrates the direction of the roads.

Provided that we simulate a city environment, some roads were constantly preferred than others. Similarly, there are some rush hours during the day, that there is more traffic in the roads.

Not only did we create 2 different network topologies, but also we evaluated the proposed algorithm for different number of vehicles, transmission ranges, and various average vehicle velocities. The results are illustrated in the following charts and tables.

5.2 Performance for different number of vehicles

We collected results about the number of clusters per minute, cluster size per minute, number of messages, and the number of gateway nodes. We conducted simulations about 50, 100, 150, and 200 nodes for each network topology respectively.

5.2.1 Number of clusters versus Number of vehicles

The number of clusters is an important metric of the clustering procedure. Here is introduced the trade-off number of clusters and thus cluster size and the performance of the clustering technique. The existence of many small clusters means that too many CHs and too gateway nodes take part in the forwarding and thus flooding occurs and benefits from clustering are diminished. On the other hand, very few large clusters are not efficient as the channel is shared among too many nodes of the same cluster and the spectrum efficiency drops drastically.

In this section, we present our measurements related to the number of formed clusters(per minute) for different numbers of vehicles both in the highway topology and in an urban environment. The transmission range (R) is approximately 50 m and in the urban environment 200 m, after euclidean-based calculations. The obtained results are demonstrated in figure 5.4.

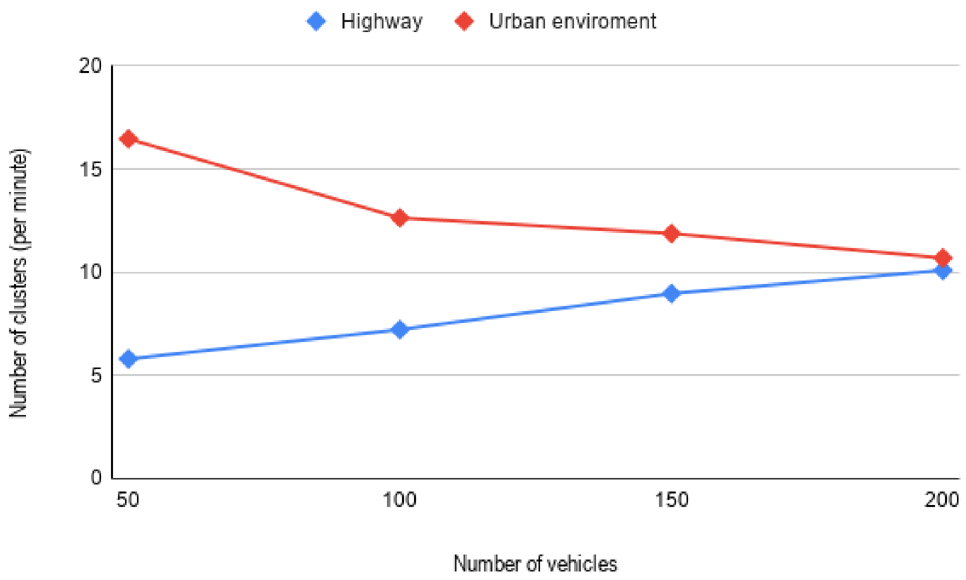

Figure 5.4: Number of clusters (per minute) for various number of vehicles. Blue line: Highway model and red line: Urban environment, respectively

Regarding the highway network, we observe that the more the number of nodes increases, the more the number of clusters produced. As the transmission range is constant in these experiments, the more vehicles there are, the more the number of vicinities that are created and hence the formed clusters.

On the other hand, in the urban environment, we perceive that the number of clusters generated is inversely proportional to the total number of nodes. The reason for this behavior is that when there are a few vehicles in a large environment, they are more scattered on the roads. Thus, more neighborhoods are created and the number of clusters increases, while when more vehicles are driving on the roads, they are more densely located.

In both cases we observe that the rate at which clusters are created is decreasing over a period of time.

5.2.2 Cluster size versus Number of vehicles

Maintaining the pre-mentioned parameters, we now calculate the average cluster-size per minute for both environments and various numbers of vehicles. Our measurements can be shown in figure 5.5.

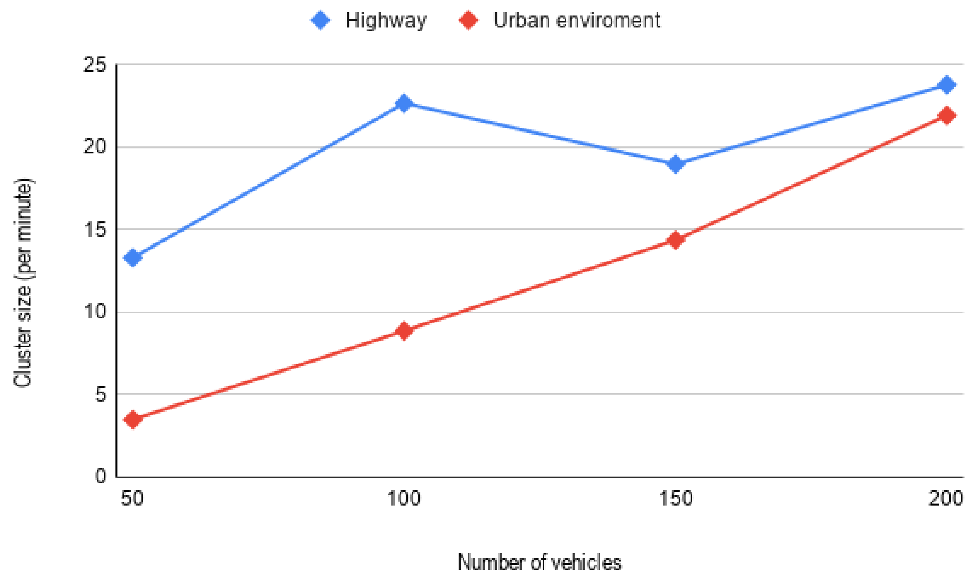

Figure 5.5: Cluster size (per minute) for various number of vehicles. Blue line: Highway model and red line: Urban environment.

From the illustrated chart in figure 5.5 we can realize that in the city model(red line), a smaller total number of vehicles on the network creates clusters including fewer nodes. The reason was already said in the previous section. In this case, vehicles are sparely positioned in the network.

In the case of the freeway simulation network(blue line), we observe that the $cluster-size \times numberofclusters$ is larger than the total number of vehicles that participate in the simulation. This is because as the vehicles are densely deployed in the network they are included in more than one cluster at the same time. However, in the previous topology, the product($cluster-size \times numberofclusters$) is at least equal to the total number of vehicles. This leads us to the conclusion that all nodes are covered by some cluster.

5.2.3 Average number of exchanged messages versus Number of vehicles

We evaluated the EAVC protocol by capturing measurements about the average number of exchanged messages per minute. In figure 5.6 we represent our results.

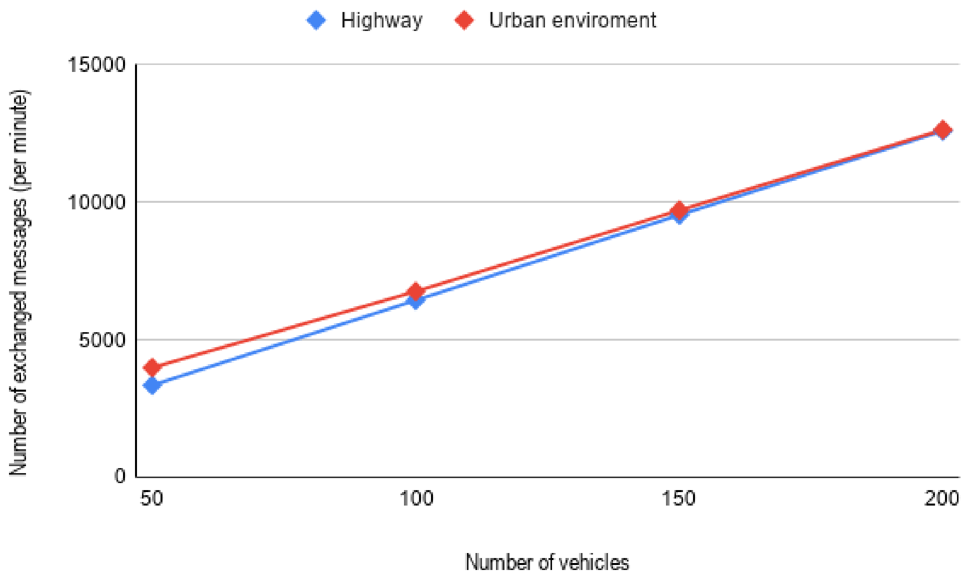

Figure 5.6: Average number of exchanged messages (per minute) for various number of vehicles. Blue line: Highway model and red line: Urban environment.

In both scenarios, we realize that the larger is the number of vehicles the number of messages is increasing. The reason is that each node advertises its position, velocity, density, channel condition, and its remaining energy by using hello messages. After the cluster head selection, CHs also send hello messages that include their cluster members. Thus, the total number of messages sent in the network is proportional to the total number of vehicles that take part in the simulation. The benefit of the proposed algorithm in the total overhead is that no joining/leaving the cluster messages are broadcasted.

5.2.4 Number of gateway nodes versus Number of vehicles.

The number of gateway nodes is a significant parameter for our protocol evaluation. With the increase of the number of gateway nodes, the available bandwidth for each gateway decreases while the route quality between a non-gateway vehicle and a gateway node improves. Table 5.1 below shows the number of gateway nodes per second, which arise from the simulations, for both topologies and various numbers of vehicles.

Number of gateway nodes		
Freeway topology	Urban environment	**Number of vehicles**
10	12	50
10	12	100
10	12	150
10	12	200

Table 5.1: Number of gateway nodes versus various numbers of vehicles for both networks(freeway and city topology respectively).

The number of gateway nodes results from the Q-learning algorithm, mentioned in chapter 04 from the equations 4.7 and 4.8 according to the number of receivers, the quality of V2V links, and the available LTE bandwidth. Thus it is independent of the number of nodes that participate in the network.

For the specific algorithm, the number of gateways is the same for various numbers of vehicles for both networks. Thus, the algorithm could be characterized as efficient concerning this metric, as the number of gateway nodes is not becoming larger with the increase of the total number of nodes.

5.3 Protocol evaluation for various transmission ranges

This section is dedicated to the presentation of the results obtained by the simulation in similar metrics as the previous one but in this case, what is different each time is the transmission range (R). The participated nodes are set to 200 and they covered by one LTE BS.

5.3.1 Number of Clusters versus Transmission Range

Figure 5.7 displays the average number of generated clusters per minute versus different transmission ranges in meters for both environments.

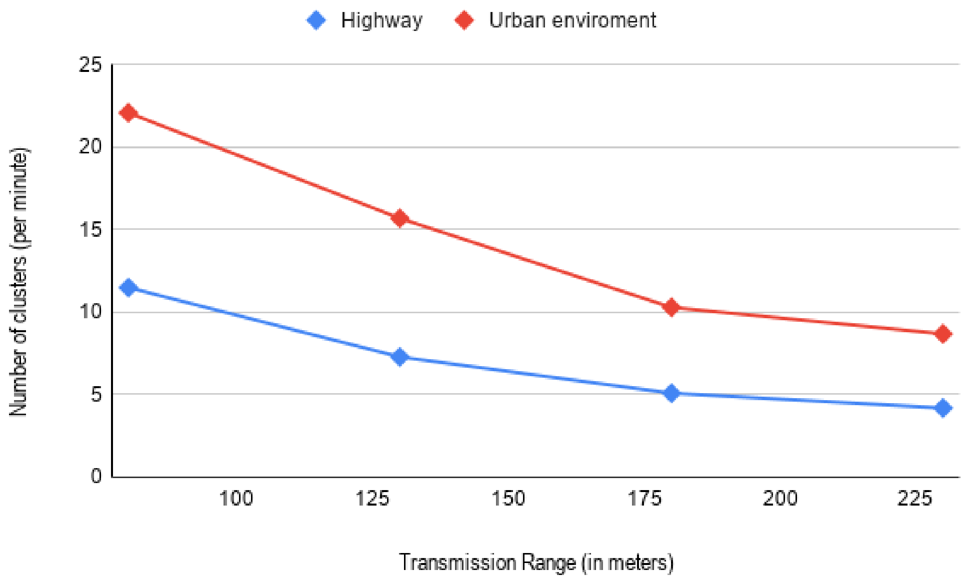

Figure 5.7: Number of clusters (per minute) versus transmission range(R) in meters. Blue line: Highway topology, Red line : Urban environment.

Cluster formation is resulting after the competency value calculation. In specific each node calculates the competency value for each neighbor in R/2 area. The node owned the maximum competency value within the range of R/4 declares itself as CH. Hence, the number of generated clusters is affected by the transmission range. In more detail, in both scenarios we observe that this number decreases as the transmission range increases.

Concerning the cluster lifetime and therefore their stability, in the highway network increasing the transmission range increases the probability that a vehicle maintains connectivity with its cluster. On the other hand, in the urban environment, we realize that the increase in communication range does not have a big positive impact as in cluster stability. This is due to the behavior of vehicles in such topologies. In an urban network, vehicles change their direction more frequently than in highways. They also accelerate and decelerate more often due to turns and congestion. This combination causes reclustering.

5.3.2 Cluster size versus Transmission range

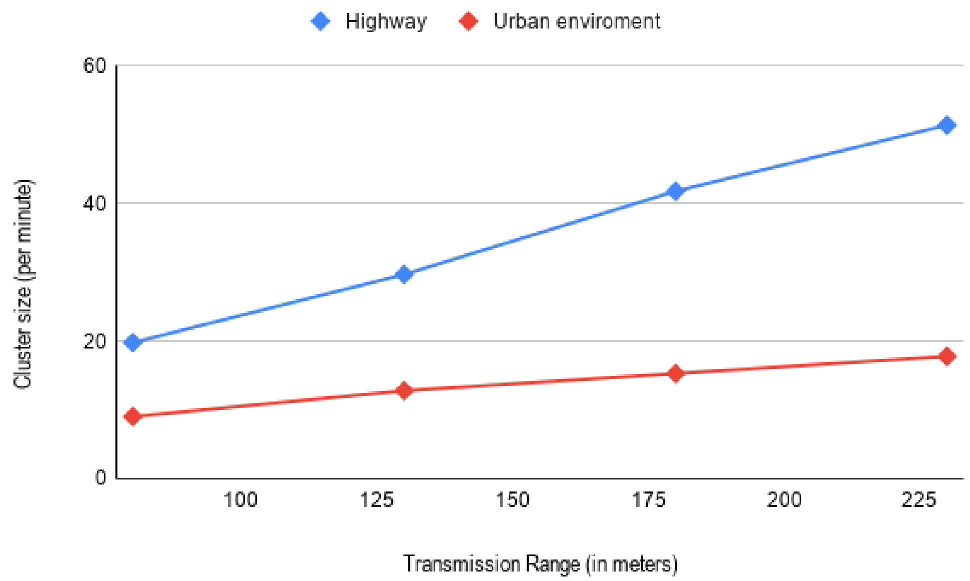

Figure 5.8: Cluster size (per minute) versus transmission range(R) in meters. Blue line: Highway topology, Red line : Urban environment.

In figure 5.8 we can see the cluster size per minute for different transmission range. Once again in the second network, vehicles are more sparely located so smaller clusters but more in number are created compared to the highway model. As we measure the average cluster size in one minute we observe that at some time clusters including only 4 vehicles are generated. Concerning the transmission range in both topologies, cluster size is becoming larger with the increase of this range.

5.4 Performance for various velocities

Velocity has a great influence on the formation of the clusters. During this procedure, we take into account the velocity factor, the contribution of which is described in chapter 04 (section 4.6). Hence, we conducted simulations for various mean velocities in kilometers per hour(km/h) in both highway and urban environments. In specific, we set the mean velocity of vehicles to 20, 40, 60, 80 km/h at each time respectively. The number of nodes is 200 and the transmission range here is also 200 m.

A significant factor for characterizing our algorithm efficient or not is the lifetime and stability of clusters. Figure 5.9 illustrates the mean lifetime of clusters in seconds versus mean velocity in km/h. The blue line represents the highway topology, while the red line shows the urban one.

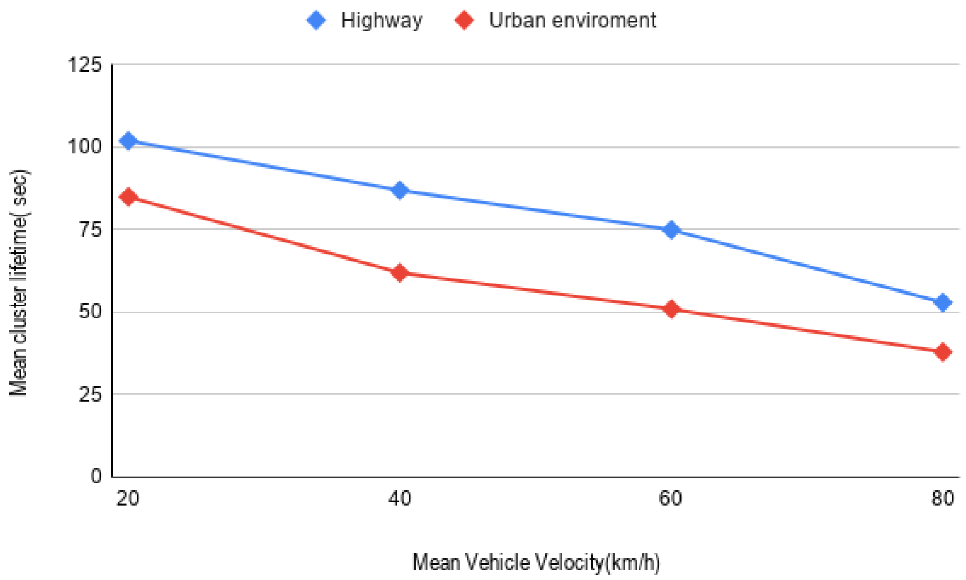

Figure 5.9: Mean lifetime of clusters in seconds versus the mean speed(km/h). Blue line: Highway topology, Red line : Urban environment.

In both scenarios, we observe that the mean lifetime of clusters decreases, as the vehicle speed increases. It is also obvious that in the urban environment the mean lifetime is shorter in every case of vehicle speed than the highway. This is because in such areas cluster formation is not mostly affected by the velocity of vehicles as the nodes cannot reach the maximum velocity as they always have to turn on and turn off due to the traffic and in order to obey in the road signs.

Chapter 6

Conclusion

Vehicular Ad hoc Networks(VANETs) own an essential role in a range of applications, due to the great benefits that they bring to them. So efficient content distribution in such a network becomes apparent. In this dissertation, we introduce the EAVC(Energy Aware VANET Clustering), a two-level clustering algorithm, integrating IEE 802.11p and LTE interface, for data dissemination in vehicular networks including both electric vehicles and conventional cars. In more detail, in the first level, we implement a fuzzy-logic based algorithm to solve the MAC contention problems and in the second level, a Q-learning algorithm is used to tune which nodes will act as gateways, which bridge V2V and LTE communications. Vehicle velocity, direction, link qualities as long as the vehicle's remaining energy have been taken into account int the formation of clusters.

For our protocol evaluation, we performed extensive simulations for various network conditions and various parameters respectively. From the results, it is obvious that the EAVC performs better on the highway than the urban environment, due to the special characteristics of such areas. Furthermore, electric vehicles are more preferable as cluster heads as they consume energy at a lower rate than conventional cars. Next, the tall vehicles as buses and trucks are also selected as CHs because the reach a lower relative velocity than the other vehicles and they provide better link qualities. After the receiving measurements, the algorithm could be characterized as energy efficient and stable as in both scenarios it performs well for a various number of nodes, transmission ranges, and velocities. It can be shown that the number of generated clusters is not very small, and in such a way to lose the clustering benefits but also is not very big and the inter-cluster communication channel to be collapsed. It also becomes apparent that it does not overload the network with redundant messages ending up to flooding, as only hello messages are exchanged among nodes. Stability appears through the mean lifetime of clusters.

Future work could include the social driving patterns of both electric and non-electric vehicles besides the pre-mentioned factors to the IEE 802.11p and LTE integration, as it seems to improve the formation and maintenance of cluster in such networks.

Bibliography

[1] URL: `https://sumo.dlr.de/docs/`.

[2] URL: `https://www.nsnam.org/`.

[3] M. Yeleswarapu A. Khan S. Sadhu. “A comparative analysis of DSRC and 802.11 over vehicular ad hoc networks”. In: *Univ. California, Santa Barbara, CA, Project Rep* (2009), pp. 1–8.

[4] Zeadally S.; Hunt R.; Chen Y; Irwin A.; Hassan A. “Vehicular ad hoc networks (VANETS): status, results, and challenges”. In: *Telecommunication Systems* 50 (Aug. 2012), pp. 217–241.

[5] Xinkai Wu; David Freese; Alfredo Cabrera; William A.Kitch. “Electric vehicles’ energy consumption measurement and estimation”. In: *Transportation Research Part D: and Environment* 34.1 (Jan. 2015). doi:https://doi.org/10.1016/j.trd.2014.10.007, pp. 52–67.

[6] Craig Cooper; Daniel Franklin; Montserrat Ros; Farzad Safaei; Mehran Abolhasan. “A Comparative Survey of VANET Clustering Techniques”. In: *IEEE Communications Surveys Tutorials* 19 (2017).

[7] Dr.R.K.Chauhan Arzoo Dahiya. “A Comparative study of MANET and VANET Environment”. In: *Journal of computing* 2 (7 July 2010).

[8] S. Ohzahata C. Wu and T. Kato. “VANET broadcast protocol based on fuzzy logic and lightweight retransmission mechanism”. In: *IEICE Trans. Commun.* E95-B.2 (Feb. 2012), pp. 415–425.

[9] Zahid Khan; Pingzhi Fan; Fakhar Abbas; Hongyang Chen. “Two-Level Cluster Based Routing Scheme for 5G V2X Communication”. In: *Emerging technologies on vehicle to everything (V2X)* 7 (Jan. 2019). doi:10.1109/ACCESS.2019.2892180, pp. 16194–16205.

[10] I.F.Akyildiz; W.Su; Y.Sankarasubtamaniam; E.Cayrci. “Wireless sensor networks: a survey”. In: *Computer Networks* 38 (4 Mar. 2002), pp. 393–422.

[11] Yu Wang Fan Li. “Routing in vehicular ad hoc networks: A survey”. In: *IEEE Vehicular Technology Magazine* 2 (2 June 2007). doi:10.1109/MVT.2007.912927, pp. 12–22.

[12] Minggao Ouyang Hewu Wang Xiabin Zhang. “Energy consumption of electric vehicles based on real-world driving patterns: A case study of Beijing”. In: *Applied Energy* 157 (Nov. 2015). doi:https://doi.org/10.1016/j.apenergy.2015.05.057, pp. 710–719.

[13] Jenifer J.-N.Liu Imrich Chlamtac Macro Conti. “Mobile ad hoc networking: imperatives and challenges”. In: *Egyptian Informatics Journal 19* 1 (1 July 2003). doi:https://doi.org/10.1016/S1570-8705(03)00013-1, pp. 13–64.

[14] Leandros A. Malgaras; Dimitrios Katsaros. “Enhanced Spring Clustering in VANETs with Obstruction Considerations”. In: (2013).

[15] Leandros A. Malgaras; Dimitrios Katsaros. “Social Clustering of Vehicles Based on Semi-Markov Processes”. In: *IEEE Transactions On Vehicular Terminology* 65.1 (Jan. 2016).

[16] Dimitrios Katsaros Leandros A. Maglaras. “Clustering in Urban environments: Virtual forces applied to vehicles”. In: *IEEE International Conference on Communications* (2013).

[17] Dimitrios Katsaros Leandros A. Maglaras. “Distributed Clustering in Vehicular Networks”. In: *Second International Workshop on Vehicular Communications and Networking* (2012).

[18] Z.J. Haas Lidong Zhou. “Securing ad hoc networks”. In: *IEEE Network* 13.6447131 (6 Dec. 1999), pp. 24–30.

[19] JiuJang Cheng; JanLu Cheng; MengChu Zhou; FouQiang Liu; SangCe Gao; Cong Liu. “Routing in Internet of Vehicles: A Review”. In: *IEEE Transactions on Intelligent Transportation Systems* 16 (5 Oct. 2015), pp. 2339–2352.

[20] A. Hafid M. A. Togou and P. K. Sahu. “A stable minimum velocity CDS-based virtual backbone for VANET in city environment”. In: *Proc. IEEE Local Computer Network Conf.* 25.1 (2014), pp. 510–513.

[21] Z. Yang M. Li and W. Lou. “CodeOn: Cooperative popular content distribution for vehicular networks using symbol level network coding”. In: *IEEE J. Select. Areas Commun.* 29.1 (Sept. 2013), pp. 538–547.

[22] Lina Zhu; Changle Li; Bingbing Li; Xinbin Wang; Guoqiag Mao. “Geographic Routing in Multilevel Scenarios of Vehicular Ad Hoc Networks”. In: *IEEE Transactions on Vehicular Technology* 65 (9 Nov. 2015). doi:10.1109/TVT.2015.2499205, pp. 7740–7753.

[23] Badia Bouhdid Mohamed Aissa Abdelfettah Belghith. “Cluster Connectivity Assurance Metrics in Vehicular ad hoc Networks”. In: *Egyptian Informatics Journal 19* 52 (June 2015). doi:https://doi.org/10.1016/j.procs.2015.05.088, pp. 294–301.

[24] D. Katsaros N. Dimokas and Y. Manolopoulos. “Energy-efficient distributed clustering in wireless sensor networks”. In: *Elsevier-Computer Networks* 70.4 (Apr. 2010), pp. 371–383.

[25] Rajasekhara BabuEzendu Ariwa P. Venkata KrishnaM, ed. *Global Trends in Computing and Communication Systems*. Springer, Berlin, Heidelberg, December 9-11, 2011.

[26] P. Rajeswari and T.N. Ravi. “He-SERIeS: An inventive communication model for data offloading in MANET”. In: *Egyptian Informatics Journal 19* (June 2017). doi:10.1016/j.eij.2017.06.001.

[27] Steven E. Shladovers Sadayuki Tsugawa Sabina Jeschke. “A Review of Truck Platooning Projects for Energy Savings”. In: *IEEE Transactions of Inteligent Vehicles* 1.1 (Mar. 2016).

[28] Doraisami JothiSiva Santhosh JayachandranJohnson and Rama Krishnan. “A Case Study on Various Routing Strategies of VANETs”. In: *Communications in Computer and Information Science* 269 (2011), pp. 353–362.

[29] Narottam Chand Vishal Kumar Shailendra Mishra. "Applications of VANETs: Present Future". In: *Communications and Network* 5 (Feb. 2013), pp. 12–15.

[30] Celimuge Wu and Tsutomu Yoshnaga; Xianfu Chen; Lin Zhang; Yusheng Ji. "Cluster-Based Content Distribution Integrating LTE and IEEE 802.11p with Fuzzy Logic and Q-learning". In: *IEEE Computational intelligence magazine* (Feb. 2018).