TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

BRAVE	Beaconless Routing Algorithm for Vehicular Environment
CH	Cluster Head
CM	Cluster Member
CBLTR	Cluster-Based Life-Time Routing
CCH	Candidate Cluster Head
CORA	Control Overhead Reduction Algorithm
CBR	Cluster-Based Routing Protocol
CHADS	CH advertisement message
CBDRP	Cluster-Based Directional Routing Protocol
CHLT	Cluster Head Life-Time
CMLT	Cluster Member Life-Time
CMHELLO	CM HELLO message
CHR	CH routing protocol
DSRC	Dedicated Short Range Communication
ECORA	Enhanced Control Overhead Reduction Algorithm
FCC	Federal Communication Commission
GPCR	Greedy Perimeter Coordinator Routing
IGRP	Intersection-based Geographical Routing Protocol
GyTAR	Greedy traffic-aware routing protocol

GPS	Global Positioning System
ICH	Intersection Cluster Head
IDVR	Intersection Dynamic VANET Routing
ITS	Intelligent Transportation System
LT	Life-Time
MANET	Mobile Adhoc NETwork
MoZo	MOving-ZOne-based
PCHEA	Passive CH Election Avoidance protocol
PDR	Packet Delivery Ratio
SCSR	Set of Candidate Shortest Routes
SDN	Software Defined Network
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
V2I	Vehicle to Infrastructure
VANET	Vehicle Adhoc NETwork
VDLA	Vehicle Density and Load Aware routing protocol

INTRODUCTION

The Intelligent Transportation System (ITS) that includes all types of communications between vehicles is an important next-generation transportation system. ITS provides many facilities to the passengers, such as safety applications, assistant to the drivers, emergency warning, etc. Vehicular Ad Hoc NETwork (VANET) is a derived form of self-organized Mobile Ad Hoc NETwork (MANET). In VANET, vehicles are equipped with an On-Board Units (OBUs) that can communicate with each other (V2V communications), and/or with stationary road-side units (V2I) that are installed along the roads. Figure 0.1 presents these two kind of communication. VANETs have several characteristics that makes it different from MANETs, such as high node mobility, predictable and restricted mobility patterns, rapid network topology change, and frequent battery charging, so energy consumption is not a big issue in VANET (Tayal & Triphathi (2012)).

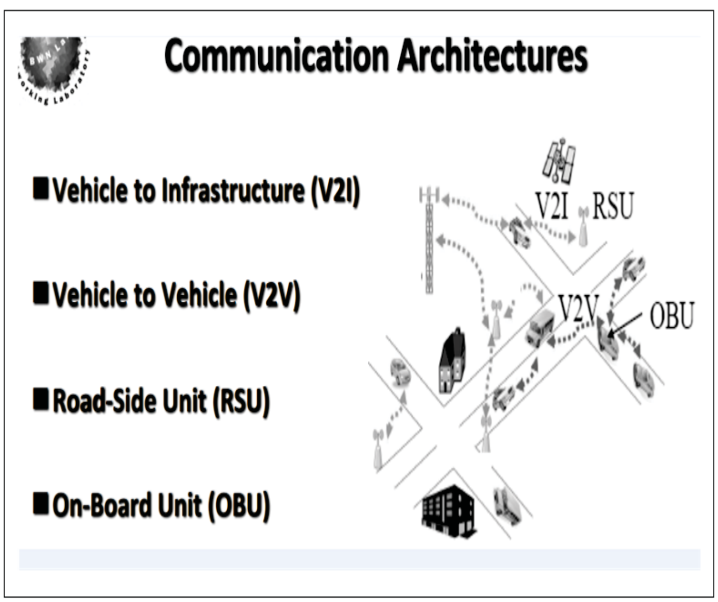

Figure 0.1 V2V and V2I communication,
https:www.studyblue.com (taken in September 11,2017)

Dedicated Short Range Communication (DSRC) technology is an emerging technology that is developed to work in very high dynamic networks, to support fast link establishment and to minimize communication latency. DSRC is designed to ensure the reliability of safety applications, taking into consideration the time constraints for this type of applications. In the United States, Federal Communication Commission (FCC) has allocated 5.9GHz for DSRC technique to support public and commercial application in V2V and V2I. The frequency takes the range of (5.850-5.925) GHz and divides it into seven non-overlapping 10MHz channels. The DSRC is developed to support the data transfer in a rapidly changing topology networks, such as VANET, where time response and the high transmission rate is required. VANETs deal with two wireless access standards: IEEE 802.11p deals with the physical and MAC layer, and IEEE 1609 deals with higher-layer protocols. According to IEEE 802.11p, vehicles are capable to share their GPS related position together with velocity and acceleration (reference:551 (2010)).

VANET is proposed and adapts different types of routing protocols, such as proactive(Spaho *et al.* (2012)), reactive(Ding *et al.* (2011); Abedi *et al.* (2009); Yu *et al.* (2011)), hybrid (Wu *et al.* (2013) Chai *et al.* (2013)), and geographic-based routing protocols (Liu *et al.* (2008); Luo & Zhou (2011)). The proactive and reactive routing protocols are classified under the topology based routing protocol category, which aims to discover the route between the source and destination before starting the data transmission. The main difference between the two is that the proactive routing protocol initiates a route discovery to all nodes located in the entire network, yielding an increase in control overhead messages and end-to-end delay. While in the reactive routing protocol, a source node initiates a discovery process to reach only the desired destination. This process reduces the control overhead messages; however, the route discovery process is required in finding a route for every new node. The hybrid routing protocol combines the features of both proactive and reactive routing protocol. The nodes in the hybrid network are grouped together in a particular area called clusters. Hybrid routing protocols, sometimes called Cluster-Based Routing (CBR) protocols, are designed to improve the network scalability by allowing the nodes within the clusters to communicate through a pre-selected Cluster Heads (CHs) using a proactive routing protocol. However, in the case of communication between clusters, a reactive routing protocol is triggered.

Geographic-based routing protocols or Location-based routing protocols combine the position information with topological knowledge of the actual road map and surroundings. In geographic-based routing protocols, the data is transmitted directly from the source to the destination without initiating any route discovery process. Therefore, each forwarding node assumes to know the following: its current location (using GPS), neighbors locations (by periodically exchanging of Hello messages), and destination location (by using location service protocol(Camp *et al.* (2002))).

Intersection-based Geographical Routing Protocol (IGRP) is a location-based routing protocol, which is suitable to urban environments. IGRP (Saleet *et al.* (2011)) is based on an effective selection of road intersections a packet must follow to reach the desired destination. This protocol is characterized by selecting the routes with high route stability. In addition it satisfies QoS constraints with tolerable delay, bandwidth usage, and error rate.

CBR protocols are widely used to improve the scalability of VANET environment and to reduce the control overhead message. Although the clustering techniques are minimizing the routing control overhead messages, frequent CH elections increase the control overhead messages associated with the re-election process. The control overhead messages are produced by: First, exchanging of HELLO messages between the CMs and the CH, and second, the CH ADvertiSement (CHADS) messages broadcasted periodically by the CH. When control overhead messages are increasing in a cluster topology, it reduces the available bandwidth resources.

Following are some advantages of clustering in VANET:

a. Improving PDR;

b. Reducing network traffic in inter-cluster network;

c. Improve the communication scalability for large number of nodes;

d. Minimal routing overhead.

The mobility in VANETs causes high topology changes and in turn leads to excessive control overhead messages and frequent link communication failures. Traditionally, clustering techniques have been used as the main solution to reduce the control overhead messages in VANET, in which the network is divided into multiple clusters and selecting one of the Cluster Members (CMs) as a Cluster Head (CH). Still, a problem occurs when the control overhead messages increase due to periodically forwarding of CM HELLO (CMHELLO) messages between the CMs and the CH, and when the CH periodically broadcasts an CH ADvertiSement (CHADS) messages to declare itself to the CMs. Hence, minimizing control overhead messages in any cluster environment is an essential goal to use the network resources efficiently.

Although the clustering techniques are minimizing the routing control overhead messages, clustering control overhead messages still consume much of VANET resources. The clustering control overhead messages are produced by : First, forwarding of control messages between the CMs and the CH named CMHELLO messages. Second, broadcasting of control messages by the CH named CHADS messages. When control overhead messages are increasing in any clustered topology, the available bandwidth resources reduce significantly. The CBR protocols aims to increase the network throughput at the cost of decreased the clustered control overhead messages.

Clustering techniques is considered an important solution to improve the network stability and performance. However, the frequent CH election mainly increases the clustered control overhead messages, which yields to consume high amount of available network resources. High clustered control overhead messages is considered the main problem that negatively impacts the network performance.

0.1 Motivation

Clustering techniques are an important solution to reduce the network complexity. The main two objectives for clustering: (1) to reduce the routing control overhead messages and (2) to increase the network stability and efficiency. In the literature, some of the researchers have investigated how the clustering techniques reduce the routing control overhead messages, but the majority of literature have ignored the impact of local clustered control overhead messages to the VANET network in general. The mechanism of selecting the stable CH directly impacts the clustered network performance in terms of throughput and end-to-end delay. The cluster is considered as stable cluster when it selects a CH with higher transmission time.

Many parameters should be considered to elect the CH in the clustered VANET topology, such as:

a. The vehicle location;

b. The vehicle velocity;

c. The current time of the vehicle.

One of the VANET characteristics is the high mobility. The high mobility in VANET forces the vehicles to periodically exchanging clustered control overhead messages. Therefore, the excessive amount of clustered control overhead messages yield to consume high amount of available bandwidth resources.

In this thesis, there are five motivation:

a. Improving the throughput in a clustered VANET topology;

b. Reducing the end-to-end delay in a clustered VANET topology;

c. Reducing the number of cluster control overhead messages in a clustered VANET topology;

d. Reducing the number of elected CH in a clustered VANET topology;

e. Reducing the number of relayed hops in a clustered VANET topology.

0.2 Problem Description

High mobility is the main characteristic that makes VANET different than MANET. Many application proposed for MANET are not applicable in VANET due to the high mobility. The challenges to solve the mobility problem started by many steps: First; clustering the network helps to reduce the complexity of flat network to be managed by small clusters, and within each cluster only a selected CH is responsible for local and external communication. Second; the mechanism to select a stable CH, the CH should be elected by considering some of the mobility parameters as velocity and acceleration. The considered parameters should provide a CH with high transmission time and less frequent status changes. In this thesis, we address many problems in clustered VANET topology, as follows:

a. Clustered VANET routing protocol suffers of low throughput and high end-to-end delay due to electing unstable CHs;

b. Clustering the VANET topology generates a high clustered control overhead messages due to exchange the CMHELLO and CHADS messages in the clusters;

c. Frequently electing CHs produce an extra control overhead messages due to not selecting the best CH;

d. Clustered VANET routing protocol produces an extra routing control overhead messages because it forwards all the time only to the adjacent CHs. When the number of relayed CHs increases the routing control overhead messages also increases.

0.3 Objectives

The main goal behind this work is to solve the problems discussed in the previous section. Therefore, The first main objective is to propose a new CH election mechanism that elects a CH with highest stability among CMs. In addition, a clustered VANET protocol is proposed in a bidirectional segment scenario and a in a grid scenario. The main objectives of these two protocols are to increase the throughput and to minimize the end-to-end delay in a clustered based topology.

The second main objective is to reduce the number of generated clustered control overhead messages in each cluster. Therefore, a clustered control overhead reduction algorithms are proposed to reduce the number of generated clustered control overhead messages in each cluster.

The third main objective is to reduce the number of CH elected in each cluster. Therefore, a proposed election avoidance protocol is proposed to reduce the number of CH elected process triggered in each cluster.

Finally, the last objective of this work is reduce the number of relayed CHs between the source and destination. Therefore, a routing protocol is proposed to reduce the number of relayed CHs in VANET topology.

This work should solve the following main issues:

a. How to increase the throughput and to reduce the end-to-end delay in a bidirectional highway topology scenario;

b. How to increase the throughput and to reduce the end-to-end delay in a grid topology scenario;

c. How to optimize the number of clustered control overhead message in a clustered topology scenario;

d. How to minimize the number of elected CH in a clustered topology scenario; and

e. How to reduce the number of relayed CHs in a clustered topology scenario.

These issues will be solved by proposing different algorithms. The evaluation of the proposed algorithms is compared with other proposed algorithms in the literature.

0.4 Thesis outline

The thesis is an article-based dissertation, which organized as follows. Chapter 1 proposes two algorithms: CBLTR protocol and IDVR protocol. The CBLTR protocol aims to increase the route stability and average throughput in a bidirectional segment scenario. The Cluster Heads (CHs) are selected based on maximum Life-Time (LT) among all vehicles that are located within each cluster. The IDVR protocol aims to increase the route stability and average throughput, and to reduce end-to-end delay in a grid topology. The elected Intersection CH (ICH) receives a Set of Candidate Shortest Routes (SCSR) closed to the desired destination from the Software Defined Network (SDN). The IDVR protocol selects the optimal route based on its current location, destination location, and the maximum of the minimum average throughput of SCSR.

Chapter 2 proposes two algorithms: First, a Control Overhead Reduction Algorithm (CORA) which aims to reduce the control overhead messages in a clustered topology, by developing a new mechanism for calculating the optimal number of CMHELLO messages. Second, an Enhanced version of CORA (ECORA) which aims to reduce the CHADS messages that broadcasted by the CHs, by proposing a CHADS prediction algorithm that enables the CH to predict the period of time for broadcasting the CHADS messages.

Chapter 3 proposes a new Passive CH election avoidance (PCHEA) protocol that aims to optimize the number of CH election process. In PCHEA protocol, each CH selects another CH based on specific information already stored in its memory, without requiring to trigger the election function. The CH sends to its CMs the next CH identification and its activation time. Also, a CH Routing (CHR) protocol is proposed, that aims to reduce the number of relayed CHs between any pair of vehicles. In CHR protocol, the CH selects the second adjacent CH among all CH located within its transmission range. The PCHEA protocol and CHR protocol significantly reduce the number of CH elected and increase the average throughput in a bidirectional highway scenario, respectively.

Last chapter presents the following: a conclusion of this work, suggested future works, and a list of related publication.

0.5 Summary of achievements and novelty

Chapter 2 proposes two routing algorithms that improves the performance of CBR protocols in any VANET environment. First; a novel Cluster-Base Life-Time Routing (CBLTR) protocol in a segment topology is introduced. The CHs are elected based on maximum LT, and the re-election process is required only when the CHs reach their corresponding threshold point. Based on the simulation results, CBLTR protocol shows a significant improvement in terms of average throughput. The enhancement in CBLTR protocol is a new mechanism to select new CHs. The selected CHs have longer LT span making the protocol more stable. Second; an Intersection Dynamic VANET Routing (IDVR) protocol in a grid topology is proposed. Each time the packet reaches the intersection, ICH recursively applies the IDVR protocol between the current intersection and the desired destination intersection, taking into account the stability of the connected route. The IDVR protocol selects the optimal route based on its current location, destination location, and a maximum of the minimum average throughput for SCSRs. IDVR increases the overall network efficiency, by increasing the route throughput, and decreasing end-to-end delay. As in our simulation, we have proved that the IDVR protocol outperforms VDLA, IRTIV, and GPCR in terms of end-to-end delay and throughput.

In Chapter 3, two algorithms are proposed: a Control Overhead Reduction Algorithm (CORA) and an Enhanced version of Control Overhead Reduction Algorithm (ECORA). These algorithms significantly reduce the number of control overhead messages generated by the CMs and the CHs in a clustered highway topology. We present a new mechanism for calculating the optimal period for forwarding or broadcasting the control overhead messages. The main contribution of these algorithm is to change the mechanism of forwarding the CM clustered control overhead messages in a clustered topology to be based on location instead of periodic of time. Also, we propose a CH prediction algorithm to enhance the CH to announce their status in specific predicted period of time instead of broadcasting within all CHLT.

Chapter 4 proposes two main protocols; first, the Passive CH Election Avoidance (PCHEA) protocol. Second, the CH Routing (CHR) protocol. The PCHEA aims to increase the network performance by reducing the number of CH election processes. The CHR protocol aims to reduce the number of relayed CHs between any pair of vehicles. As we present in the simulation results, the PCHEA protocol significantly outperforms CBLTR protocol in terms reducing the number of CH election process. Also, the CHR protocol significantly outperforms the CBLTR protocol in terms of average throughput.

In summary, the main novelty of this thesis presented in the following points:

a. Formulation the problem of the network instability in VANET and proposing a clustering model;

b. Improving the network performance by proposing a stable cluster routing protocol that applies in highway and grid topology scenarios;

c. Proposing algorithms to reduce the clustered control overhead messages in a clustered topology;

d. Proposing a CH election avoidance algorithm to avoid the control overhead messages produced due to the election processes;

e. Improving the VANET performance by reducing the number of relayed CHs in the clustered highway topology.

CHAPTER 1

PERFORMANCE IMPROVEMENT OF CLUSTER-BASED ROUTING PROTOCOL IN VANET

Ahmad Abuashour[1], Michel Kadoch[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper published in IEEE Access, July 2017

1.1 Abstract

In this article we propose three algorithms: Cluster-Based Life-Time Routing (CBLTR) protocol, Intersection Dynamic VANET Routing (IDVR) protocol, and Control Overhead Reduction Algorithm (CORA). The CBLTR protocol aims to increase the route stability and average throughput in a bidirectional segment scenario. The Cluster Heads (CHs) are selected based on maximum Life-Time (LT) among all vehicles that are located within each cluster. The IDVR protocol aims to increase the route stability and average throughput, and to reduce end-to-end delay in a grid topology. The elected Intersection CH (ICH) receives a Set of Candidate Shortest Routes (SCSR) closed to the desired destination from the Software Defined Network (SDN). The IDVR protocol selects the optimal route based on its current location, destination location, and the maximum of the minimum average throughput of SCSR. Finally, the CORA algorithm aims to reduce the control overhead messages in the clusters, by developing a new mechanism to calculate the optimal numbers of the control overhead messages between the CMs and the CH. We used SUMO traffic generator simulators and MATLAB to evaluate the performance of our proposed protocols. These protocols significantly outperform many protocols mentioned in the literature, in terms of many parameters.

1.2 Introduction

The Intelligent Transportation System (ITS) that includes all types of communications between vehicles is an important next-generation transportation system. ITS provides many facilities to the passengers, such as safety applications, assistant to the drivers, emergency warning, etc. Vehicular Ad Hoc NETwork (VANET) is a derived form of self-organized Mobile Ad Hoc NETwork (MANET). In VANET, vehicles are equipped with an On-Board Units (OBUs) that can communicate with each other (V2V communications), and/or with stationary road infrastructure units (V2I) that are installed along the roads. VANETs have several characteristics that makes it different from MANETs, such as high node mobility, predictable and restricted mobility patterns, rapid network topology change, and frequent battery charging, so energy consumption is not a big issue in VANET (Tayal & Triphathi (2012)).

Dedicated Short Range Communication (DSRC) technology is an emerging technology that is developed to work in very high dynamic networks, to support fast link establishment and to minimize communication latency. DSRC is designed to ensure the reliability of safety applications, taking into consideration the time constraints for this type of applications. In the United States, Federal Communication Commission (FCC) has allocated 5.9GHz for DSRC technique to support public and commercial application in V2V and V2I. The frequency takes the range of (5.850-5.925) GHz and divides it into seven non-overlapping 10MHz channels. The DSRC is developed to support the data transfer in a rapidly changing topology networks, such as VANET, where time response and the high transmission rate is required. VANETs deal with two wireless access standards: IEEE 802.11p deals with the physical and MAC layer, and IEEE 1609 deals with higher-layer protocols. According to IEEE 802.11p, vehicles are capable to share their GPS related position together with velocity and acceleration (551 (2010)).

VANET is proposed and adapts different types of routing protocols, such as proactive(Spaho *et al.* (2012)), reactive(Ding *et al.* (2011); Abedi *et al.* (2009); Yu *et al.* (2011)), hybrid (Wu *et al.* (2013) Chai *et al.* (2013)), and geographic-based routing protocols (Liu *et al.* (2008); Luo & Zhou (2011)). The proactive and reactive routing protocols are classified under the topology based routing protocol category, which aims to discover the route between the source and destination before starting the data transmission. The main difference between the two is that the proactive routing protocol initiates a route discovery to all nodes located in the entire network, yielding an increase in control overhead messages and end-to-end delay. While in the reactive routing protocol, a source node initiates a discovery process to reach only the desired destination. This process reduces the control overhead messages; however, the route discovery process is required in finding a route for every new node. The hybrid routing protocol combines the features of both proactive and reactive routing protocol. The nodes in the hybrid network are grouped together in a particular area called clusters. Hybrid routing protocols, sometimes called Cluster-Based Routing (CBR) protocols, are designed to improve the network scalability by allowing the nodes within the clusters to communicate through a pre-selected Cluster Heads (CHs) using a proactive routing protocol. However, in the case of communication between clusters, a reactive routing protocol is triggered.

Geographic-based routing protocols or Location-based routing protocols combine the position information with topological knowledge of the actual road map and surroundings. In geographic-based routing protocols, the data is transmitted directly from the source to the destination without initiating any route discovery process. Therefore, each forwarding node assumes to know the following: its current location (using GPS), neighbors locations (by periodically exchanging of Hello messages), and destination location (by using location service protocol(Camp *et al.* (2002))).

Intersection-based Geographical Routing Protocol (IGRP) is a location-based routing protocol, which is suitable to urban environments. IGRP (Saleet *et al.* (2011)) is based on an effective selection of road intersections a packet must follow to reach the desired destination. This protocol is characterized by selecting the routes with high route stability. In addition it satisfies QoS constraints with tolerable delay, bandwidth usage, and error rate.

CBR protocols are widely used to improve the scalability of VANET environment and to reduce the control overhead message. Although the clustering techniques are minimizing the routing control overhead messages, frequent CH elections increase the control overhead messages associated with the re-election process. The control overhead messages are produced by: First, exchanging of HELLO messages between the CMs and the CH, and second, the CH ADvertiSement (CHADS) messages broadcasted periodically by the CH. When control overhead messages are increasing in a cluster topology, it reduces the available bandwidth resources.

In this article, we define three contributions as follows:

a. We combine the characteristics of geographic-based routing protocol with cluster-based routing protocol to produce a novel CBR protocol. The proposed routing protocol is called Cluster-Based Life-Time Routing (CBLTR) protocol, which objects to eliminate the route discovery process and reduces the number of re-election process for new CHs. CBLTR protocol aims to increase the route stability and average throughput in a bidirectional segment scenario;

b. We propose a novel Intersection Dynamic VANET Routing (IDVR) protocol, which aims to increase the overall network efficiency, by increasing the routes throughput, and decreasing end-to-end delay;

c. We propose a Control Overhead Reduction Algorithm (CORA). The proposed protocol aims to minimize the number of the control overhead messages generated by CMs in a clustered segment scenario.

This article is outlined as follows; in section 1.3, we present state of the arts that related to our works. Section 1.4 presents CBLTR protocol in segments scenario. Section 1.5 explains IDVR protocol in a grid scenario. Section 1.6 explains CORA algorithm in a segment scenario. Section 1.7 explains the mathematical model. Section 1.8 shows the simulation results and analysis. Finally, section 1.9 concludes this article.

1.3 State of arts

In general, Cluster-Based Routing (CBR) protocol is a hybrid routing protocol, that divides the large network into small areas called clusters, and inside the cluster, there are a specific routing protocols called intra-cluster routing protocol. The communication between clusters is performed via pre-selected nodes called Cluster Heads (CHs). The CHs are responsible for coordinating the members of the cluster, and communicating betwen clusters using inter-cluster routing protocol (Song *et al.* (2010)). By clustering, only the CH requires to find the destination route. Therefore, the routing overhead is proportional to the number of clusters and not the number of nodes. The objectives of using clusters are to minimize the control overhead messages, and increase the scalability of the network.

A large number of algorithms have been proposed for the CH election process in VANET. There are many parameters considered to improve the CH election process, such as location, direction, and velocity. In (Abedi *et al.* (2009)), the proposed protocol elects the CHs by considering vehicle movement parameter and link quality between vehicles, forming CHs relatively more stable. The proposed protocol reduces the message overhead and MAC layer contention time at each vehicle while maintaining a high Packet Delivery Ratio (PDR).

In (Song *et al.* (2010)), Tao et al. proposed a Cluster-Based Directional Routing Protocol (CBDRP) for highway scenario, in which the CH selects another CH according to the moving direction of the vehicle. The vehicles which are closest to the center coordination of the clusters are elected as cluster heads. This protocol shows significant improvement compared with AODV and is equivalent to GPSR in terms of transmission performance.

Ahmed et al. (Louazani *et al.* (2014)) proposed Cluster-Based algorithm for connectivity maintenance in VANET (AODV-CV), the CH is elected based on the closest actual velocity to the average velocity of all nodes located inside the cluster zone. The proposed protocol outperform AODV in terms of throughput by increasing the velocity.

B. Ramakishnan et al. in(Ramakrishnan (2011)) presented a new CBR protocol called CB-VANET. This model focused on the development of clustering framework for communication among the VANET vehicles. This model decreased the latency in VANET by reducing the cluster creation time, CH election time, and cluster switching time. The vehicle with minimum velocity was chosen as the CH. The proposed protocol outperformed other protocols in terms of the creation time and switching time.

In IGRP (Saleet *et al.* (2011)), a source node needs to know the route it should use to forward data packet to the gateway, which has an up-to-date view of the local network topology. This gateway acts as an allocation service where it stores current location information about all vehicles in its vicinity. By using location management service, each vehicle reports its location information to the gateway as it moves within the transmission range of the gateway. Based on this information, the gateway constructs a set of routes between itself and the vehicles. To increase the stability, IGRP builds routes based only on the intermediate and adjacent road intersections toward the gateway.

Christian et al. (Jerbi *et al.* (2009)) proposed an intersection routing protocols called Greedy Perimeter Coordinator Routing (GPCR) protocol. When the packet is delivered to the node located at the intersection, GPCR selects the next street that has a node with the shortest route to the destination. Every time the packet is delivered to the intersection, the gateway continues forwarding to the selected path. If a local maximum problem occurred, then an alternative route should be used based on the right-hand rule.

In (Jerbi *et al.* (2009)), Moez jerbi et al. proposed an improved greedy traffic-aware routing protocol (GyTAR), which is an intersection-based geographical routing protocol. It uses clusters concept between adjacent intersection to forward the data packet, and it also considers the distance to cluster center to select the cluster head. ChunChun et al. (Zhao *et al.* (2012)) proposed a Vehicle Density and Load Aware (VDLA) routing protocol for VANETs. VDLA selects a series of intersection to construct the route to the destination. The selection is based on the real-time vehicle density, the traffic load of the corresponding road segment and the distance to the destination. VDLA outperforms GPCR in terms of average end-to-end delay and PDR.

IRTIV (Oubbati *et al.* (2014)) is a position-based routing protocols that aims to find the shortest connected route to the destination in a city scenario, by taking into consideration the real-time segment density, estimated in a completely distributed manner based on the periodic exchange of Hello messages. IRTIV periodically calculates a real-time cost value by considering traffic density. As a result, IRTIV protocol improves the PDR and reduces the end-to-end delay compared with AODV, and GyTAR.

VANETs are autonomous systems formed by connected vehicles without the need of any infrastructure. Routing in VANET is a significant challenge due to the nature of fast topology changes. The high mobility in VANET forces the vehicles to periodically exchanging control overhead messages. Therefore, the excessive amount of control overhead messages yield to consume high amount of available bandwidth resources.

Control overhead reduction techniques are an important and interesting subject in many recent researches. The main objective of minimizing the control overhead messages is improving the network efficiency by producing more bandwidth resources for data transmission.

The main solution to reduce the control overhead messages is to use the clustering technique, the concept of clustering means to transform the big network into small grouped networks called clusters. In each cluster, one of cluster members (CMs) should be elected to be responsible for all local cluster communication, and its called Cluster Head (CH). This process will significantly reduce the control overhead messages because it restricts the communication between each CM and CH instead of exchanging the control overhead messages between all the CMs in the cluster. Many researches proposed several algorithms of selecting the CH in each cluster based on specific parameters, such as: vehicle ID, vehicle location, vehicle speed, vehicle direction, and vehicle LT.

In the cluster, CMs and CH should periodically exchange the control overhead messages, the HELLO message is one of important control overhead messages used to define the vehicle identity and location in VANET network. The number of control overhead messages in the cluster is in proportion to the number of CMs. Many techniques are proposed in the literature to reduce the number of HELLO messages as follows:

In (Lo *et al.* (2013)), the authors proposed a new clustering algorithm that takes into consideration the vehicle position and speed for selecting the CH. The proposed algorithm is intended to increase the clusters stability by reducing the number of CH changes, which yields the reduction of the control overhead messages produced from frequently re-election process. In (Lo *et al.* (2013)), the authors do not mention the impact for the size of CHADS messages, and they do not consider the impact of the HELLO messages in terms of its size and its updating time period. In (Mohammad & Michele (2010)), the authors proposed a lane-based clustering algorithm to improve the network stability by reducing the CH election times. The proposed algorithm elects the CH based on the traffic flow of vehicles in the cluster. In (Rawashdeh & Mahmud (2012)),the authors enhanced a new parameter to improve the CH election. This parameter is the speed difference. By using this parameter, the cluster becomes categorized based on different speeds.

The CBDRP (Song *et al.* (2010)) concentrates on the reduction of the routing overhead packet from source to destination, without considering the control overhead messages produced by the CMs in each cluster. Pedro et al. (Ruiz *et al.* (2010)) proposed a Beacon-less Routing Algorithm for Vehicular Environment (BRAVE). The proposed protocol objective is to reduce the control overhead messages in broadcast approaches. In BRAVE, the next forwarder vehicle is reactively selected among those neighbors that have successfully received the messages. The drawback of BRAVE protocol is that each vehicle participating in the routing protocol still requires to exchange a beacon messages among them. In the simulation setting, BRAVE sets the exchanging time of the beacon message to 2 seconds to keep monitoring the vehicles location. BRAVE considered as reactive routing protocol. In general, reactive routing protocol reduce the control overhead messages compared to proactive routing protocol. However it still suffers of high control overhead messages compared to CBR protocols.

Dan et al.(Lin *et al.* (2017)) proposed a MOving-ZOne-based (MoZo) architecture. MoZo consist of multiple moving zones that group vehicles based on the movement similarity. The selected CH is responsible for managing information about CMs as well as the forwarding packets. The control overhead messages updating period for the CMs in Mozo architecture varies between moving function of 5 m/s or 4 seconds.

This article proposes a novel Cluster-Base Life-Time Routing (CBLTR) protocol in a segment topology, an Intersection Dynamic VANET Routing (IDVR) protocol in a grid topology, and Control Overhead Reduction Algorithm (CORA)in a clustered topology. The objectives of this article are to increase the route stability and average throughput in a segment topology, reduce end-to-end delay in a grid topology scenario, and reduce the control overhead messages in the clusters. In the next three sections, we analyze the methodologies we followed to achieve our objectives, respectively.

1.4 Cluster-Based Life-Time Routing (CBLTR) Protocol

In this section, we present the steps and algorithms to improve the routing stability in a bidirectional segment scenario, as follows: First, the segment is divided into multiple stationary clusters. Then, a new distributed CH election algorithm is proposed to select a CH based on specific parameters. Finally, a new routing protocol is proposed to select the most suitable candidate CH based on CH's neighbors and destination location.

1.4.1 Cluster dividing

The segment is a bidirectional road, and each segment is divided into multiple clusters that equal half of the transmission range of a standard vehicle(Song *et al.* (2010)). We assume that all vehicles have a predefined knowledge of cluster coordination and identification. Each vehicle must be assigned to one cluster at each unit of time based on its location, and with a unique ID for each vehicle and cluster. Figure 1.1 presents a segment with two clusters, it also shows the cluster edges between the clusters. At any unit of time, if each vehicle enters any cluster zone (enters the cluster edge lines between the clusters), then it becomes a member of this cluster and must send A HELLO message to the CH of the cluster (more details of CH election and sending HELLO messages are explained in Section 1.4.2 and Section 1.6, respectively).

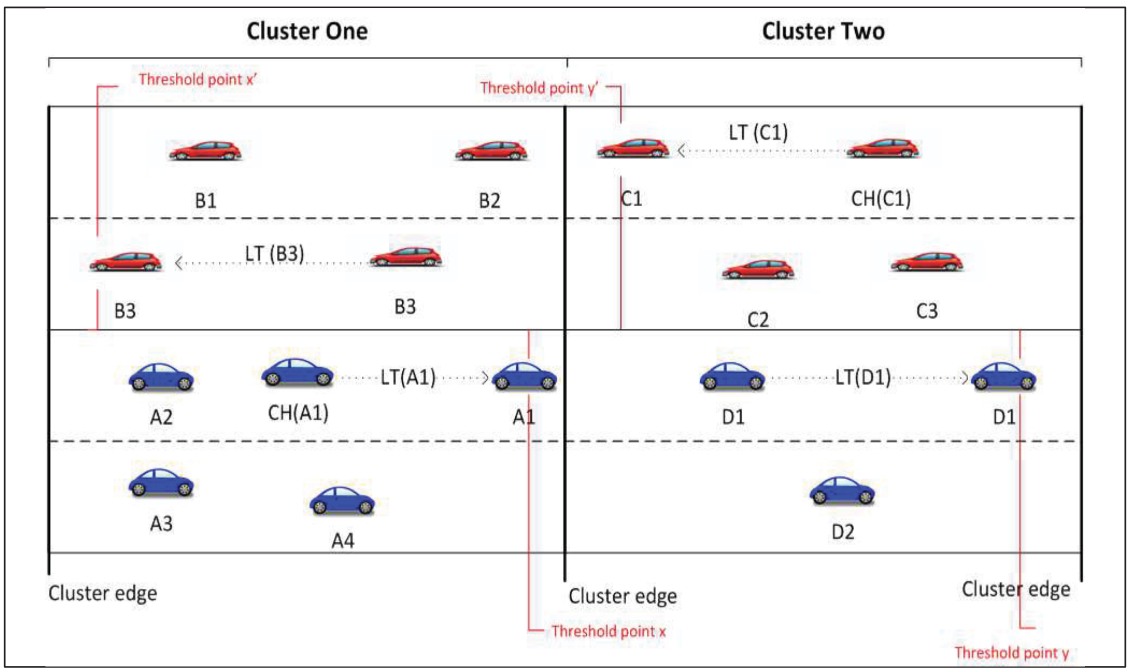

Figure 1.1 Cluster dividing and CH election

1.4.2 Cluster Head (CH) election

Each vehicle that enters a predefined stationary cluster zone should periodically calculate specific cost value, which is called Life-Time (LT). The LT of each vehicle depends on the current velocity of the vehicle as well as the distance to the predefined directional cluster edge (using an Euclidean distance equation). The vehicle with the maximum LT is elected as a CH, then it remains as the CH till it arrives at the directional threshold point; this means there are no new election until the current CH arrives at the predetermined directional threshold point. The directional threshold point is defined as a point distant from the directional edge of the cluster. The distance that separates these two points is calculated by considering the CH velocity, and the time it takes to proceed until the re-election process. The distance from the directional threshold point to the directional edge of the cluster must be enough for a CH vehicle to handover the CH function to another vehicle without losing the communication. This ensures that any vehicle in each cluster can successfully complete the re-election process. For example, if the handover time (re-election time and the time to forward the CH information to the new CH) is equal to $0.2s$, then, the threshold distance (D_{th}) is calculated dynamically based on the current CH velocity. Equation 1.1 shows and illustrates the calculations of the threshold distance in each specific cluster.

$$D_{th}(CID) = V_{CH}(CID) \times HOT \tag{1.1}$$

Where:

$D_{th}(CID)$: Threshold Distance for specific cluster.

$V_{CH}(CID)$: CH Velocity for specific cluster.

HOT: Hand-Over Time.

CID: Cluster IDentification.

Example: For a CH speed of $50Km/h$,D_{th} of cluster ID 1 equal $D_{th}(1)$ = $\frac{50\times1000m}{3600s}$ $\times\, 0.2s$= 2.7 $meter$.

Therefore, before 2.7 meters of the directional cluster edge, the Hand-Over process should be invoked, and this value varies based on the current CH velocity. In Equation 1.2, the LT is periodically calculated for each vehicle within each cluster, using the distance from the current location of the vehicle to the directional edge of the cluster and the vehicle velocity.

$$LT(i) = d_{ith}/(V_i) \tag{1.2}$$

Where:

$LT(i)$: Life-Time of vehicle i.

d_{ith} : Distance between vehicle i and directional edge of the cluster

V_i : Velocity of vehicle i.

In Figure 1.1, we present a simple process for electing the CH at specific time. In cluster 1, vehicles A1 and B3 are moving in opposite directions and each has the maximum LT in its direction, but the LT of vehicle A1 is greater than the LT of vehicle B3, therefore the vehicle A1 is elected to be as CH for cluster 1. The same election process will proceed in cluster 2, and also because the LT of the vehicle C1 is greater than the LT of the vehicle D1, then the vehicle C1 is elected as CH in cluster 2. Each elected CH (A1 and C2) keeps its status as CH until it arrives to its corresponding threshold point (x and y', respectively). When any CH arrives at its corresponding threshold point, then a new election process should start.

In Algorithm 1.1, each vehicle enters any cluster becomes member of that cluster. In (lines 1 to 9), we classified the vehicles based on its location in real time. Then the LT is calculated for all vehicles within their associated clusters at any given time. The LT is calculated based on the time that each vehicle will remain in the cluster (as in lines 10 to 23), depending mainly on the distance to the upcoming directional edge of the cluster, as well as the velocity. The vehicle that has the maximum LT at a specific time will be selected as the CH and remains as the CH till it arrives at the directional threshold point. At this time; a new election should be invoked, and a new CH must be selected. The purpose of not updating the CHs all the time is to reduce the control overhead messages produced from the re-election process, in other words, to maximize the LT for the CH.

Figure 1.2 presents a flowchart of CH election in each cluster. Each vehicle calculates the LT that it requires to reach the directional edge of the cluster. In each cluster, there is a distributed election algorithm that elects the vehicle that remains within its cluster the maximum LT time. The elected CH should announce itself periodically (every τ second), by forwarding a CH ADvertiSement (CHADS) message. At each time, if any vehicle enters a new cluster zone, its default status is CM, then it should wait (τ second) to receive a CHADS message. If the vehicle receives a CHADS message, then it keeps the status as CM, otherwise, the vehicle announces

Algorithm 1.1 LT calculation and CH election in the segment

```
for t=anytime and t<=simulationtime do
    for VID=1 to i<=Numberofveh do
        for CID=1 to CID<=Numberofclus do
            if location(VID)= Location(CID) then
                Add VID to MCID; add VID to the members of this cluster(CID)
            end
        end
    end
end
for t=anytime and t<=simulationtime do
    for CID=1 to CID<=Numberofclus do
        while VID ∈ CID and VID< MCID(CID) do
            distance(i)=abs(dirclusedge(VID)-loc(VID));
            LT (VID) = distance(VID)/velocity(VID)
        end
        CH=VID(index (max (LT)));
        if location(CH)= threshold point(CH) then
            update CH;
            else
                keep old CH
            end
        end
    end
end
```

itself as the CH. The CH should periodically (every τ second) advertise its status until it reaches its predetermined threshold point; then to avoid any communication interruption, the CH asks for early re-election process by advertising a LEAVE message. At this time, the CM with maximum LT will be elected as the new CH. If the CH arrives at the predefined threshold point and the cluster is empty of other CMs, then the CH keeps moving until it finds another CH (more details of exchanging the control overhead messages are explained in Section 1.6).

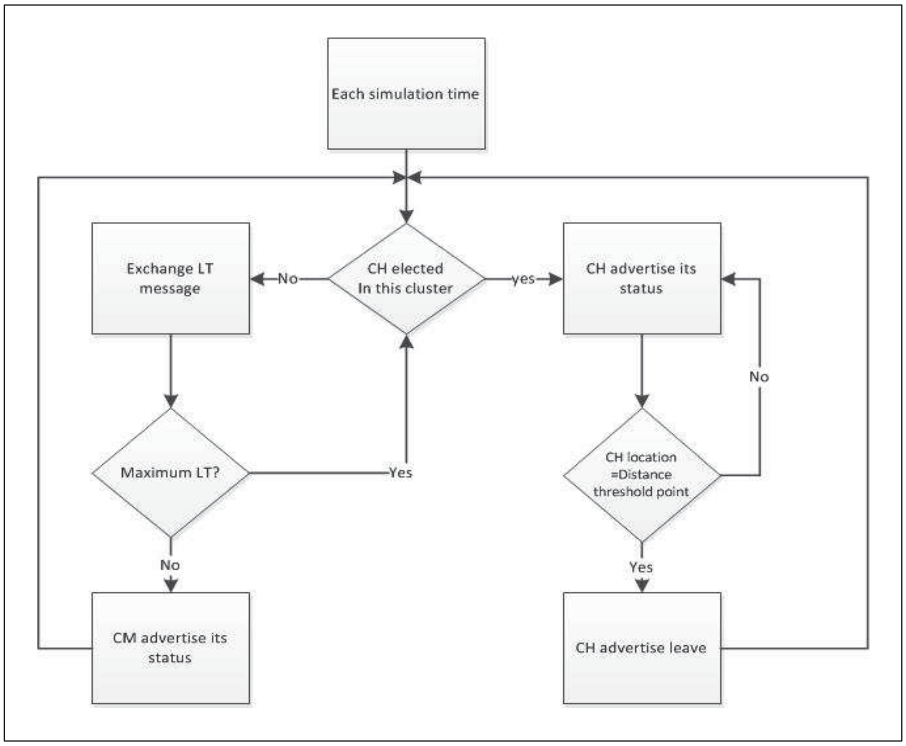

Figure 1.2   Flow chart of CH election

1.4.3 Routing procedure in the segment

The CBLTR protocol aims to propagate the packets within the segment through the selected CHs. Each CH builds its routing table and stores in it the adjacent CH IDentification (CHIDs) and its associated locations. Figure 1.3 shows the contents of the CH routing table, which contains the CH IDentification (CHID), its location, its LT, and expiry time. The expiry time is used to keep updating the routing table contents. When the local CH receives a packet, it searches in its routing table for the candidate CHs that are located close to the destination regardless of the CH's direction, then it forwards the packet to the next CH that has the maximum LT. If two candidate CHs with equal LT are available for forwarding the packets, then the CH in the same direction and closest to the destination is selected. If there are no relaying CHs to the destination, then as a recovery process the local CH follows a store-and-forward process; it stores the packet in a specific buffer and keeps moving till it finds another relaying CH. Algorithm 1.2 shows a pseudo code of the CBLTR protocol that presents the steps of propagating

Routing Table			
CHID	Location	LT	Expiry time

Figure 1.3 CH Routing table

the packets within the segment. At any time of the simulation, if the vehicle receives a packet, it first checks its CH routing table, and then selects the CHs that are closest to the destination in another table, called the candidate CH table (which has the same structure as the CH routing table in Figure 1.3). The CH with maximum LT of the candidate CH table is selected as the next forwarder vehicle. In case LT values are equal, then the one closest to the destination is selected regardless of its direction. Finally, if the CH routing table is empty, then the current CH follows a store-and-forward process. Figure 1.4 illustrates the process of calculating the

Algorithm 1.2 CBLTR protocol

```
for t=anytime and t<=simulationtime do
    if Packet received by CH at time=t then
        Check Routing table of the CH;
        if Routing table NOT empty then
            Store the closest CHs to destination in Candi CH table;
            if Candi CH table has 2 or more CH with same maximum LT then
                Next CH= CH that closest to the destination;
            else
                Next CH= CH with maximum LT
            end
        end
        else
            Store and Forward
        end
    end
end
end
```

throughput. The filled circle is the CHs and the unfilled circle is the CMs. The LT is calculated for the CHs in each cluster, and only the candidate with maximum LT is selected as the CH regardless of its direction. The throughput is the rate of successful data delivery over a communication channel. In Equation 1.2, each CH calculates the Transmit(T) time which is the same as the LT for its associated cluster. Each cluster has two directional CHs that move in opposite directions. After calculating the LT time for each CH within each cluster, we select the maximum LT in each cluster. Each CH remains as CH until it arrives to its predetermined corresponding threshold point. The throughput is then calculated by multiplying the transmission rate (S) by the fraction of T that each CH will remain in its associated cluster.

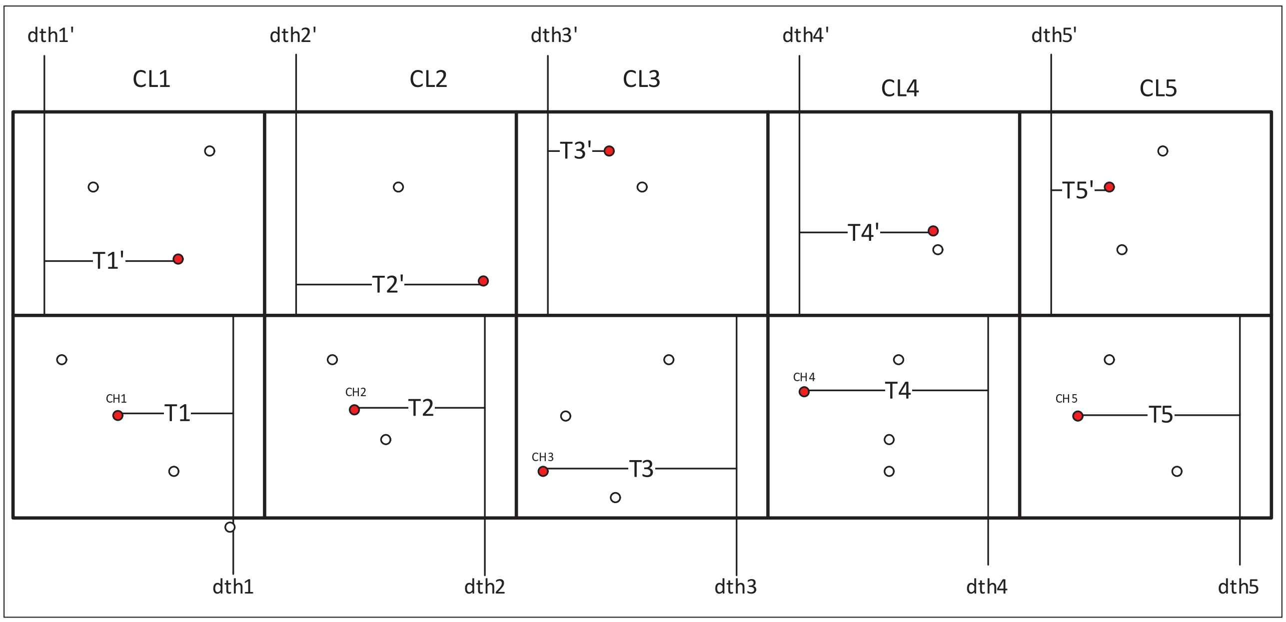

Figure 1.4 Average throughput calculation for a bidirectional segment

In Equation 1.3, we calculate the throughput for n clusters in a bidirectional segment.

$$Throughput = S_1 \times \frac{max(T_1, T_1')}{max(T_1, T_1') + H} \cdot S_2 \times \frac{max(T_2, T_2')}{max(T_2, T_2') + H} \cdot .... \cdot S_n \times \frac{max(T_n, T_n')}{max(T_n, T_n') + H} \tag{1.3}$$

Where:

n: Maximum number of clusters on the segment.

S_i: The transmission rates for the cluster CH of cluster i.

T_1, T_1': The transmit time of CH in cluster i in two directions.

H: The Hand-Over time.

In Equation 1.4, for simplicity, we assume that the Ts are the same for the entire segment. We take the average of all Ts for the segment as follows:

$$T_{avg} = \frac{\sum_{k=1}^{n} T_k}{n} \tag{1.4}$$

By substitute T_{avg} in Equation 1.3, we calculate the average throughput as in Equation 1.5:

$$AverageThroughput = S_1 \times \frac{T_{avg}}{T_{avg}+H} \cdot S_2 \times \frac{T_{avg}}{T_{avg}+H} \cdot .... \cdot S_n \times \frac{T_{avg}}{T_{avg}+H} \tag{1.5}$$

But

$$S_2 = S_1 \times \frac{T_{avg}}{T_{avg}+H}, S_3 = S_2 \times \frac{T_{avg}}{T_{avg}+H} \tag{1.6}$$

In general,

$$S_n = S_{n-1} \times \frac{T_{avg}}{T_{avg}+H}, for \quad n \geq 2 \tag{1.7}$$

By substitute Equation 1.7 in 1.5, we obtain

$$AverageThroughput = S_1 \times \prod_{i=1}^{n} (\frac{T_{avg}}{T_{avg}+H})^i \tag{1.8}$$

Where:

T_{avg}: Average transmit time for the segment.

i: cluster sequence number.

n: Maximum number of of clusters in the segment.

H: Hand-Over time.

S_1: data rate of the first cluster in the segment.

In Equation 1.8, we calculate the average throughput for any segment size. In addition, it determines the degree of stability for any segment. The segments with higher average throughput indicates higher segment stability. In Section 1.7.1 we present more theoretical analysis of LT in a cluster.

1.5 Intersection Dynamic VANET Routing (IDVR) protocol

IDVR is a new Intersection Dynamic VANET Routing protocol. There are two main contributions of this protocol. First, we use the CHs in relaying the packets from the source to the destination; then the CHs are selected based on maximum LT. By relaying the packets via CHs, we increase the segment stability and reduce the probability of link failure (Abuashour & Kadoch (2016)). Second, we propose an Intersection Dynamic VANET Routing (IDVR) protocol, which computes the optimal route to the destination taking into account the real-time traffic from source to destination, and the current source and destination intersection location. The IDVR algorithm works in real-time and recursively operates at each intersection until it arrives at the final destination. Our objectives are to increase the route stability and average throughput, and to reduce end-to-end delay in a grid topology scenario. In the next subsections, we analyze the methodology we followed to achieve our objectives.

1.5.1 Software Defined Network (SDN)

A Software Defined Network is used to provide flexibility to networks and to introduce new features and services to VANETs. Ian Ku et al. (Ku *et al.* (2014)) evaluate the performance of SDN-based VANET architecture with other traditional MANET/VANET routing protocols, including GPSR, OLSR, AODV, and DSDV. The results show that the PDR is much higher when adopting SDN in VANET environments.

We use SDN to define the candidate routes between two intersections; SDN requires creating a table that includes segment IDs, as well as average throughput (as calculated based on Equation 1.8), and this information must be updated periodically. Figure 1.5 shows the contents of the SDN table. The design of full SDN architecture is beyond the scope of this article. The SDN provides upon request the candidate routes between the source intersection and the destination intersection (the intersection closest to the destination location) using the Dijkstra algorithm. Each candidate route consists of a series of intersections and the corresponding weight.

SDN table parameters		
Segment ID	Average throughput	Expiry time

Figure 1.5 SDN parameters

1.5.2 Intersection Cluster Head (ICH)

When any vehicle enters the intersection cluster zone, it wait for τ second. If it receives any CHADS message, then it announce itself as CM and sends a HELLO message to the ICH, otherwise it announce itself as ICH. In Figure 1.6, there are 4 vehicles want to enter the cluster intersection zone. The first vehicle that enters the intersection zone will announce itself as ICH, and any vehicle enters after that will announce itself as CM. The ICH Keeps its status as ICH and periodically forward CHADS message until it arrives to its corresponding threshold point. At the moment when the ICH arrives at the threshold point a new election process should be invoked. At this time, the new ICH will be elected among the CMs that are located within the cluster intersection zone and has the maximum LT.

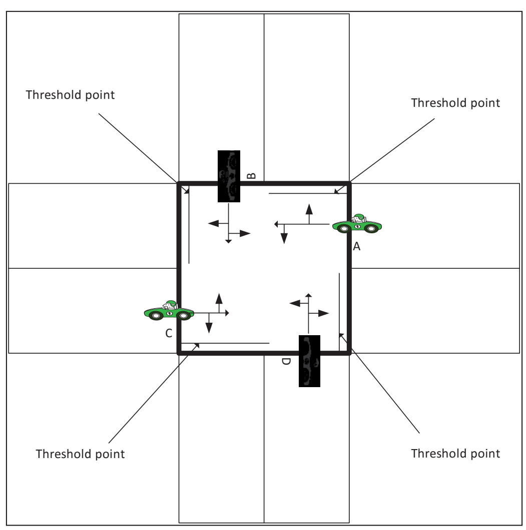

Figure 1.6 ICH election

In Algorithm 1.3, we show how the vehicles join the intersection cluster coordination within the simulation time. Furthermore, we explain how to select the Intersection CH (ICH). First, the vehicle with maximum LT is elected as the ICH and maintains its status until it reaches a predefined threshold point. When it arrives at the threshold point, a new ICH should be elected and all data should propagate to the new elected ICH. The packets are propagated via pre-selected CHs in each segment; as they arrive at the intersection, the packets are propagated via ICH.

Algorithm 1.3 ICH election in the intersection

```
for t=anytime and t<=simulationtime do
    for VID=1 to i<=Numberofveh do
        for ICID=1 to ICID<=Numberofinter do
            if location(VID)= Location(ICID) then
                Add VID to MICID; add VID to the members of this cluster(ICID)
            end
        end
    end
end
for t=anytime and t<=simulationtime do
    for ICID=1 to CID<=Numofinter do
        while VID ∈ CID and VID< MCID(CID) do
            distance(i)=abs(dirclusedge(VID)-loc(VID));
            LT(VID) = distance(VID)/velocity(VID)
        end
        ICH=VID(index (max (LT)));
        if location(ICH)= threshold point(ICH) then
            update ICH;
            else
                keep old ICH
            end
        end
    end
end
```

1.5.3 An Intersection Dynamic VANET Routing (IDVR) Protocol

When the packets arrive at the intersection cluster, the ICH determines the real-time optimal route that the packet is supposed to follow to reach the desired destination, taking into account the maximum of the minimum average throughput for all candidate routes (more details in Section 1.7.2). The SDN provides the candidate routes between the current intersection and the destination intersection. In Figure 1.7, each candidate route has unique identification (R_{ID}), which consists of a series of intersections and the corresponding weight. The weight for each route is calculated by computing the average throughput (as in Equation 1.8) for each segment, and then selecting the minimum value. When there are no vehicles at the intersection cluster zone, the current CH follows the rule of store-and-forward, by storing the packets inside the CH buffer and continuing to move until it reaches another CH within its transmission range and closer to the destination intersection than itself.

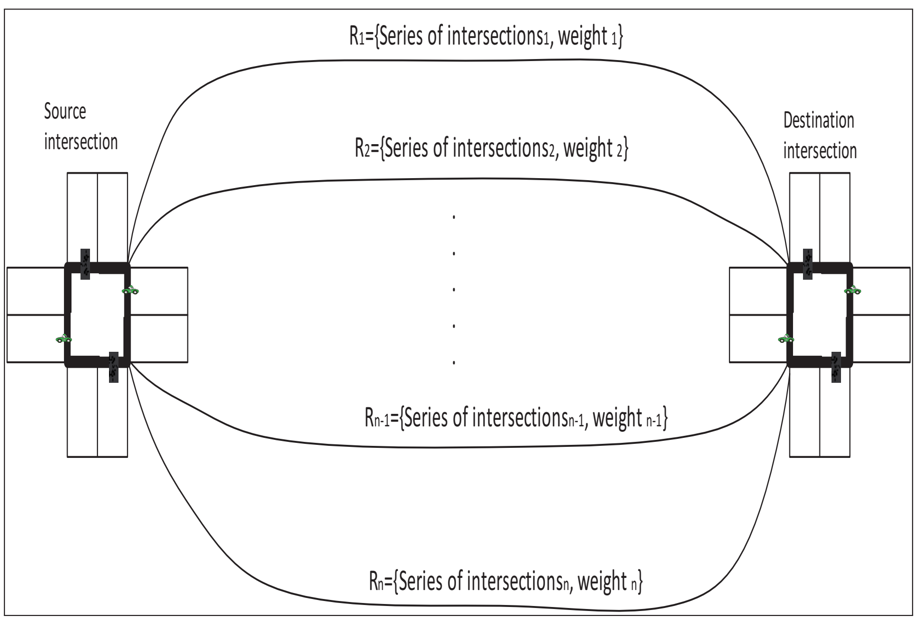

Figure 1.7 Set of candidate route

Algorithm 1.4 explains in pseudo code the IDVR protocol. In IDVR, each forwarder node (source node) obtains all possible routes to the desired destination and store them in specific buffer (*routeset*), as in line 2. Then it calculates the minimum number of intersections from itself to the desired destination and stores it in another buffer (*minseg*), as in line 3. To limit the routing search, first, we define a constraint to search only for routes located between a predefined minimum number of intersection (*minseg*) and a predefined maximum number of intersection (*maxseg*). The routes that pass successfully this constraint is stored in ($cons1valid$) buffer, as in line (5-10). Second, we check the routes validity in ($cons1valid$) buffer; all the segments for each route in ($cons1valid$) should be greater than a predefined specific threshold value. We assigned a binary value of one for each segment that has a throughput value that is greater than a predefined specific threshold value, and a binary value of zero for each segment that does not have a greater value than a predefined specific threshold value, as in lines (11-17). Finally, we multiply the binaries value for each route in ($cons1valid$). The routes that passed the previous two constraints will be stored in ($cons2valid$) buffer, as in line (18-22). To calculate the weight for each route, we calculate the average throughput for each segment within the route, then select the minimum average throughput value as the weight for the route. The route weight is stored in ($validrouteset$) buffer, as in line (23-31). The optimal route is the route that has the maximum route weight among ($validrouteset$), as in lines 32 and 33.

Each route consists of a series of segments. Let us consider that we have n routes, as follows:

$$SCSR = (R_1, R_2, ..., R_n) \tag{1.9}$$

Where:

$SCSR$: Set of Candidate Shortest Route.

R_n: Weight for route n.

n: Maximum number of routes.

To calculate the average throughput for each route, we should calculate the average throughput for each segment within the route, and then select the minimum average throughput (see Equation 1.10). Finally, a maximum of the minimum average throughput for SCSRs will be selected as the next segment of the selected route (as in Equation 1.11).

$$AVGR_{ID} = min(AvgS(1), AvgS(2), ..., AvgS(x)) \tag{1.10}$$

$$optroute = max(AVGR_1, AVGR_2, ..., AVGR_n) \tag{1.11}$$

Algorithm 1.4 IDVR Algorithm

```
for t=anytime and t<=simulationtime do
    routeset=shortestroute(S,D);
    minseg=min(shortestroute(S,D));
    maxseg=minsegement+2;
    for i=1 to maxsizeof(route set) do
        if minseg<numseg(routeset(i))<maxseg then
            cons1valid=routeset(i);
            numofsegcons1valid=sizeof(cons1valid)
        end
    end
    for j=1 to numofsegcons1valid do
        if cons1valid(j)>threshold then
            cons1valid(j)=1;
            else
                cons1valid(j)=0
            end
        end
    end
    for y=1 to sizeof(cons1valid) do
        if multiplication(cons1valid)=1 then
            cons2valid(y)=cons1valid(y)
        end
    end
    for i=1 to maxsizeof(cons2valid) do
        validrouteID=cons2valid(i);
        for j=1 to sizeof(validrouteID) do
            weight(j)=avgthr(validrouteID(j))
        end
        RouteIDweight(i)= Minimum(weight) ;
        routeID(i)=validrouteID
    end
    validrouteset=(routeID, RouteIDweight);
    selectedroute=maximum(validrouteset);
    return routeID(selectedroute)
end
```

Where:

R_{ID}: Route ID.

$AVGR_{ID}$: Average throughput of the route R_{ID}.

$AvgS(x)$: Average throughput of segment x in the R_{ID}.

x: Total number of segment in the route R_{ID}.

n: Total number of valid routes.

$optroute$: Optimal route.

1.6 Control Overhead Reduction Algorithm (CORA)

In VANET, the CBR protocols do not require every vehicle to know the entire topology information. Only the selected CH vehicles require to know the topology information and other CMs only require to periodically exchange their information with the CH. HELLO message is one kind of the control overhead messages that we discussing in this article. Any CM should inform the CH about its identity by sending a HELLO message, in addition it could combine other parameters such as current location, direction, velocity, and LT. The increasing size of HELLO messages is an important issue that degrade the performance of any mobile and limited networks resources . Furthermore, the frequently exchanging of HELLO message negatively impact the network performance.

Therefore, in this section we first propose a new algorithm called Control Overhead Reduction Algorithm (CORA), that aims to reduce the number of control overhead messages in a clustered topology. We then present a new design for HELLO message, by minimizing the number of parameters in HELLO message. CORA is based on the assumption that each vehicle in the VANET environment knows its current location and cluster ID by using a digital map and Global Positioning System (GPS). In the following section, we describe how CORA algorithm is able to minimize the HELLO messages between the CMs and the CH.

1.6.1 Exchanging of control messages

In general, each vehicle must be defined as CM or CH at any time. Algorithm 1.5 explains in pseudo code the CORA algorithm, initially, each vehicle enters any cluster coordination zone sets its status as CM by default. Then it should wait for τ second (lines 2 and 3), if it does not receive CHADS message, then it changes its status to CH and starts periodically (every τ second) to forward CHADS message (lines 9 and 10). If the CM receives the CHADS message then it stays as CM and replies with only one HELLO message. The replied message consists of the CM Identification and the remaining LT required to leave the cluster zone (lines 4 to 7). The remaining LT is varied among vehicle due to the velocity variation. The CHADS message consists of CH Identification and the remaining LT. The objective of periodically exchanging CHADS message is to inform new CMs arrival that an active CH exists. When the CH receives all replies from the CMs within its associated LT, the CH is capable to calculate the Candidate CH (CCH) before leaving the cluster. Therefore, the CMs do not require to periodically update their information with the CH. In other word, the HELLO messages produced by the CMs are proportional to the number of CH changes instead of specific period of time. This finally yields to significantly minimizing the control overhead messages in each cluster.

Algorithm 1.5 CORA protocol

```
1  for t=anytime and t<=simulationtime do
2      if any vehicle enters the cluster Zone then
3          vehicle staus = CM and wait τ sec;
4          if CM receives CH ads message then
5              vehicle staus = CM;
6              reply to CH by one message;
7              Containes <CMID, CMLT>;
8              else
9                  vehicle staus = CH;
10                 every τ sec send CH ads
11             end
12         end
13     end
14 end
```

To calculate the number of CHADS message within the simulation time, we first divide the elected CH remaining LT time by the period of exchanging time τ (τ is a constant value). The results give us the number of CHADS message for one CH, as in Equation 1.12. Then to get the overall CHADS messages, we calculate the summation for all elected CHs in the same cluster within the simulation time, as in Equation 1.13.

$$AdsCH_{ijk} = \frac{CHLT_{ijk}}{\tau} \tag{1.12}$$

Where:

$AdsCH_{ijk}$: The total number of CHADS messages produced from CH with ID i in cluster j in segment ID k.

$CHLT_{ijk}$: The remaining LT for CH with ID i in cluster ID j in segment ID k.

τ : The periodic exchanging time for CHADS message.

$$\begin{aligned} TotalAdsclus_{jk} &= \sum_{i=1}^{x} AdsCH_{ijk}, \\ where \quad 0 &< TotalAdsclus_{jk} < simulationtime \end{aligned} \tag{1.13}$$

Where:

$TotalAdsclus_{jk}$: The number of CHADS message produced from CHs in cluster ID j in segment ID k.

x : The maximum number of elected CH within the simulation time for cluster j.

To calculate the total CHADS messages generated in a segment, we do the summation of the number of CHADS messages for each cluster, as follow:

$$\begin{aligned} TotalCHAds_k &= \sum_{j=1}^{y} TotalAdsclus_{jk} \\ &= \sum_{j=1}^{y} \sum_{i=1}^{x} \frac{CHLT_{ijk}}{\tau} \end{aligned} \tag{1.14}$$

Where:

$TotalCHAds_k$: The number of CHADS message produced from CHs in segment ID k.

$CHLT_{ijk}$: The remaining LT for CH with ID i and Cluster ID j in segment ID k.

y : The maximum number of of clusters within the segment.

Since τ is constant value, then the number of CHADS messages produced by the CH are proportional to the CH LT value in each cluster.

In Figure 1.8, the CH forwards CHADS messages every τ seconds to all of its CMs until its LT expires. Each selected CH should periodically forward an CHADS messages to announce itself in the cluster zone. The vehicles A,B,C, and D are CMs that receive CHADS messages from the vehicle CH while its LT time does not expire. On the other side, when any vehicle enters the

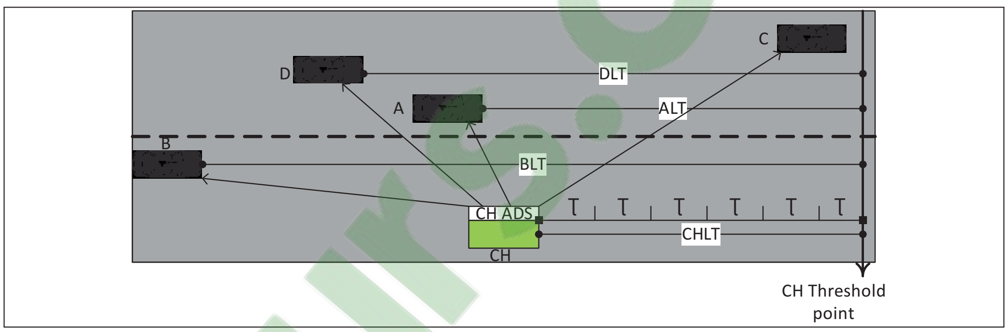

Figure 1.8 CHADS message periodically (every τ seconds)

cluster zone, then its default status is CM. It should exchange the HELLO message with the CH. In this article the main contribution is to minimize the number of CM HELLO (CMHELLO) messages by taking into consideration CHLT. When any vehicle enters the cluster zone, it sends a CMHELLO message to the CH (if it receives the CHADS after τ second), there are then two scenarios: First, if the CM Life Time (CMLT) is greater than CHLT, the number of HELLO message equals to the number of CH changes within the CMLT plus two (the mandatory two HELLO messages when the CM enters the cluster and before leaves the cluster), else the CM will generate the CMHELLO message only two times, when it enters the cluster and before it leaves the cluster, and this is because CMLT is shorter than CHLT. Figure 1.9 explains a scenarios of exchanging the CMHELLO message; first, when vehicles enters the cluster zone (as vehicle B), then it should send CMHELLO message; and when the vehicles leave the cluster zone, it sends another CMHELLO message (as vehicle C). While the vehicles (vehicle A and D) already in the cluster zone and within the CHLT do not require to send any HELLO message. Figure 1.10 explains another scenario when the CH (Old CH) arrives to the threshold point (the point that the current CH should select another CH), the CH sends an CHADS message informing the CMs of the new CH. At this time all the CMs (vehicle A and D) should send the CMHELLO message to the new CH.

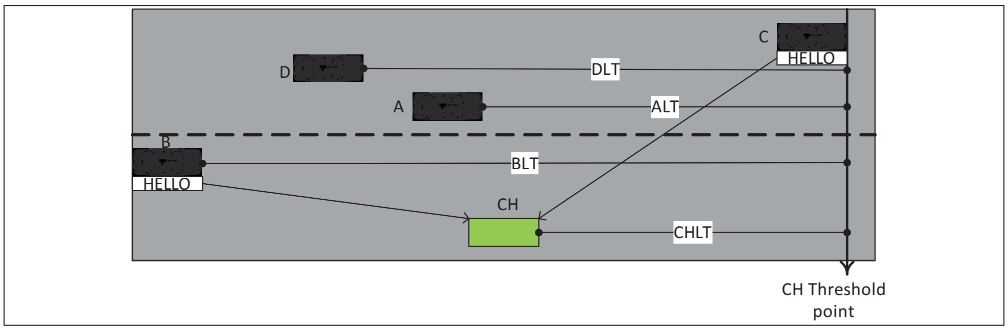

Figure 1.9 CMHELLO message when enters and leaves the cluster

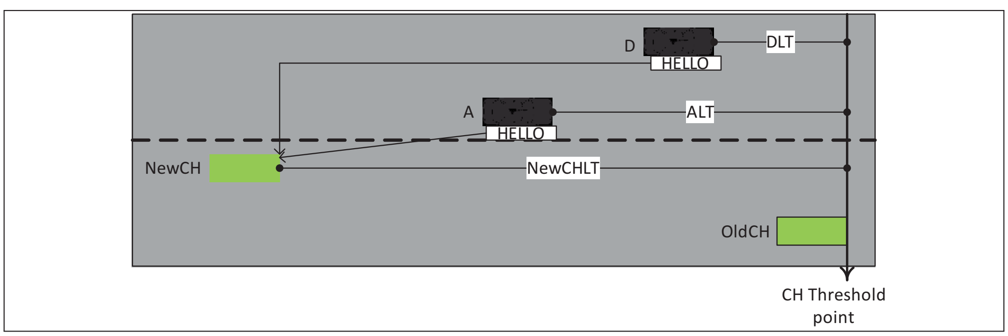

Figure 1.10 CMHELLO message when new CH selected

The following Equation describes mathematically the two scenarios in Figure 1.9 and 1.10:

$$NumCM_{ijk} = \begin{cases} numCH_{ijk} + 2, & \text{if } CMLT_{ijk} > CHLT_{jk} \\ 2, & \text{if } CMLT_{ijk} \leq CHLT_{jk} \end{cases} \tag{1.15}$$

Where:

$NumCM_{ijk}$: The number of HELLO message produced by CM with ID i in cluster ID j in segment ID k.

$numCH_{ijk}$: The number of CH changes within $CMLT_{ijk}$.

$CMLT_{ijk}$: The remaining LT for CM with ID i in cluster ID j in segment ID k.

$CHLT_{jk}$: The remaining LT for current CH with in cluster ID j in segment ID k.

Algorithm 1.6 explains how to calculate the overall CMHELLO messages in each cluster.

Algorithm 1.6 Total number of CMHELLO messages

Input : total of CM ads = 0
for *i=1 to Max number of CMs* **do**
  if $CMLT_i > CHLT$ **then**
    total of CM ads =2 + Num of CH changes Within $CMLT_i$ **else**
      total of CM ads =2 + total of CM ads
    end
  end
end
Output: return total of CM ads

We can mathematically formulate the total of CMHELLO messages for specific cluster as the following Equation:

$$TotalHELLO_k = \sum_{j=1}^{y} NumCM_{ijk} \tag{1.16}$$

Where:

$TotalHELLO_k$: Total number of CMHELLO messages produced from CMs in cluster k.

y : Total number of CM in the cluster ID k.

Also, we can mathematically formulate the total of CMHELLO messages for a specific segment as the following Equation:

$$TotalCMHELLO_m = \sum_{j=1}^{p} TotalHELLO_j \tag{1.17}$$

Where:

$TotalCMHELLO_m$: The total number of HELLO message produced from CMs in segment ID m.

p : Total number of clusters in the segment ID m.

Finally, the total control overhead messages within the simulation time equal the summation of CMHELLO messages produced from the CMs and the periodical broadcasting of the CHADS messages produced by the CHs. As the following Equation:

$$TotalAdsmesage_k = TotalCMHELLO_k + TotalCHAds_k \tag{1.18}$$

1.6.2 Designing of control overhead messages

In this section, we propose a new design for CHADS messages and CMHELLO messages. In the literature, many researchers assume different sizes of control overhead message. Mohammad et al. (Hadded *et al.* (2015)) assume that the messages generated by the CH contains highway ID, CH ID, direction, specific weight value. In contrast, the CMHELLO messages are periodically broadcasted and contains CMID, highway ID, direction, position, and speed. Dan et al. (Lin *et al.* (2017)) propose a new CBR protocol that groups the vehicle moving in the same direction in one cluster. The CMs sends periodically a CMHELLO message that contains vehicle ID, location, speed, and the direction of next intersection. Based on this information, an Algorithm is proposed to select the CH.

In Figure 1.11 and 1.12, we present the contents of CMHELLO and CHADS messages, respectively. The CMHELLO message consists of CMID and current CMLT (the time that the current CM requires till arrive at the threshold point), and the CHADS message consists of CHID, and current CHLT. An important point we have to mention is that the CH broadcasts the CHADS messages periodically (every τ second). While the CMHELLO messages are forwarded to the CH in three cases: First, when the CM enters the cluster zone. Second, when the CM leave the cluster zone. Third, when a new CH announces itself. Therefore, in the first contribution we minimize these messages to two parameter (vehicle ID, and vehicle LT) instead of many parameters mentioned in the literature, such as, location, speed, and direction. In the second contribution we optimize the number of CMHELLO messages to be forwarded only in three cases(when entering and leaving the cluster, and when CH changes), instead on exchanging in terms of time period.

CMHELLO message	
CMID	Current CMLT

Figure 1.11 CMHELLO message

CHADS message	
CHID	Current CHLT

Figure 1.12 CHADS message

1.7 Mathematical Model

In this section, we present the theoretical analysis of LT in a cluster and the grid topology mathematical model design, as follow:

1.7.1 Theoretical analysis of LT in a cluster

In this section, we explain the theoretical analysis of the LT cost value that is used in CH election.

par Each vehicle within its corresponding cluster periodically calculates the LT value. Therefore, let us assume a vehicle with ID 1 has a LT value equal to LT_1; LT_1 is the LT that the vehicle with ID 1 stays active until it reaches its corresponding threshold point (th). The LT value depends mainly on the speed and the vehicle location. If the location of vehicle ID 1 on the cluster is l_1, then the absolute distance between the vehicle and the the corresponding threshold point is denoted by $D_{l_1,th}$. $D_{l_1,th}$ is a random variable that takes values within $[0, d_{max}]$,where d_{max} is the maximum distance to the directional cluster edge. The maximum LT is calculated based on the maximum distance to the directional cluster edge and the minimum allowed speed on that cluster.

At the segment, the vehicles are moving only in two directions with one dimension (X or Y axis). Let us assume that the segment is divided into fixed size clusters (as in Figure 1.4). For simplicity, we assume that the shape of the cluster is rectangular with a length of d_{max}, and that the velocity of the vehicles follows a uniform distribution. Therefore, the probability density function (pdf) and the Cumulative Distribution Function (CDF) of the velocity (v) are determined as in the following equations respectively:

$$p(v) = \begin{cases} 0 & ,\text{if} \quad v < Min_v \\ \frac{1}{Max_v - Min_v} & ,\text{if} \quad Min_v \leq v \leq Max_v \\ 0 & ,\text{if} \quad v > Max_v \end{cases} \tag{1.19}$$

$$P(v) = \begin{cases} 0 & ,\text{if} \quad v < Min_v \\ \frac{v - Min_v}{Max_v - Min_v} & ,\text{if} \quad Min_v \leq v < Max_v \\ 1 & ,\text{if} \quad v \geq Max_v \end{cases} \tag{1.20}$$

Where: Min_v: Minimum allowed velocity in the cluster.
Max_v: Maximum allowed velocity in the cluster.

In order to transfer p_v in terms of time (t) in seconds, then we should multiply p_v by d_{max}/t^2, as in the following equation:

$$p(t) = \begin{cases} 0 & ,\text{if} \quad t < \frac{d_{max}}{Max_v} \\ \frac{d_{max}}{(Max_v - Min_v)t^2} & ,\text{if} \quad \frac{d_{max}}{Max_v} \leq t \leq \frac{d_{max}}{Min_v} \\ 0 & ,\text{if} \quad t > \frac{d_{max}}{Min_v} \end{cases} \tag{1.21}$$

By assuming that each vehicle is equipped with GPS, then each vehicle is capable of determining the distance between its location and its corresponding threshold point. Then the LT_i

equals the distance between vehicle i and the directional threshold of the cluster divided by the velocity of vehicle i (as in Equation 1.22). The Valid LT (VT) for any vehicle can be denoted as follows:

$$VT_i = \frac{d_{max} - d_{ith}}{V_i} \tag{1.22}$$

To obtain the probability value of the VT_i, we integrate the pdf of Equation 1.11 from $-\infty$ to VT_i as follows:

$$P(VT_i) = \int_{-\infty}^{VT_i} p(t)dVT = \begin{cases} 0 & ,VT_i < \frac{d_{max}}{Max_v} \\ \frac{d_{max}}{(Max_v - Min_v)}(\frac{Max_v}{d_{max}} - \frac{1}{VT_i}) & ,\frac{d_{max}}{Max_v} \leq VT_i \leq \frac{d_{max}}{Min_v} \\ 1 & ,VT_i > \frac{d_{max}}{Min_v} \end{cases} \tag{1.23}$$

The segment VT value is determined by multiplying the cluster VT from the first cluster adjacent to the intersection at the beginning of the segment to the last cluster adjacent to the intersection at the end of the segment. Therefore, the probability value of the Valid LT(VT) for the segment equals the multiplication of the CDF for all the clusters in the segment, as follows:

$$\begin{aligned} P_{seg}(VT) = P_1(VT_1) \times P_2(VT_2) .... \times P_n(VT_n), \\ where VT_n \in [\frac{d_{max}}{Max_v}, \frac{d_{max}}{Min_v}] \end{aligned} \tag{1.24}$$

Where:

$P_{seg}(VT)$: Probability that the segment has Valid LT.

VT_n: Valid LT of vehicle n.

$P_n(VT_n)$:Probability that cluster n has valid LT n (VT_n).

In Table 1.1, we present the numerical results for the probability of segment LT validity in terms of velocities and the size of clusters. Based in Equation 1.24, we calculate the probability that the segment has valid LT, when the segment is divided into different cluster sizes, different segment sizes (size of one,two,three, and four clusters), and different ranges of velocity.

Table 1.1 The probability of segment LT validity

Range of velocity (Km/h)	Cluster Size=250m				Cluster Size=500m			
	1 clus	**2 clus**	**3 clus**	**4 cluss**	**1 clus**	**2 clus**	**3 clus**	**4 clus**
(10-30)	0.6731	0.4531	0.305	0.2053	0.6746	0.4551	0.3070	0.2071
(30-50)	0.5421	0.2939	0.1593	0.0864	0.5634	0.3174	0.1788	0.1008
(50-70)	0.5327	0.2838	0.1512	0.0805	0.5381	0.2896	0.1558	0.0838
(70-90)	0.3864	0.1493	0.0577	0.0223	0.4767	0.2272	0.1083	0.0516

1.7.2 The design of grid topology

In this section we design a grid topology that consists of a series of segments and intersections. In Figure 1.13, each segment and each intersection has a unique identification. Let us assume that the grid dimensions are n horizontal intersections and m vertical intersections; thereby, we have $n \times m$ intersections and $(n-1) \times (m-1)$ segments.

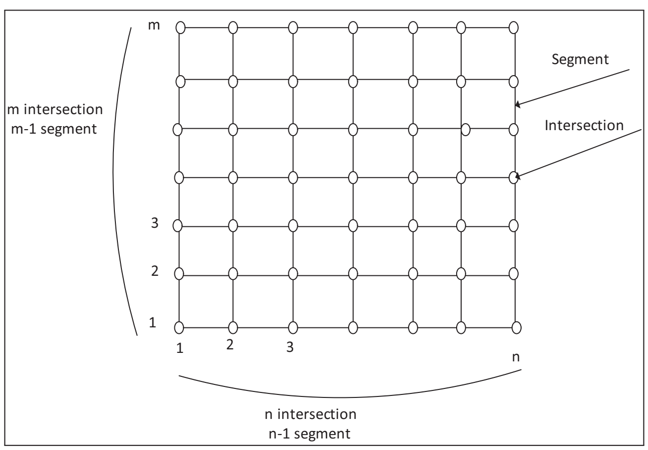

Figure 1.13 Grid topology

The Intersections Set (IS) contains all intersection IDs in the topology, as follows:

$$IS = [1, 2, 3...., ((n \times m) - 1), (n \times m)] \tag{1.25}$$

Where:

$n \times m$:Maximum number of intersections in the grid topology.

The Segments Set (SS) is defined by a new set that contains all segments in the grid topology, as follows:

$$SS = [S_1, S_2, ..., S_{max}] \tag{1.26}$$

Where:

S_i: Segment ID i

max: Maximum number of segments, which equal $m \times (n-1) + n \times (m-1)$.

The SDN defines all candidate routes from the Source Intersection (SI) to the Destination Intersection (DI). We define two constraints for route validity, as follows:

a. The Number of Segments for each Candidate Route (*NSCR*) should be varied between the minimum number of segments between SI and DI and the maximum number of segments. NSCR falls within the following range:

$$NSCR = [minsegments, maxsegments] \tag{1.27}$$

Where:

$minsegment$: Minimum number of segments in the shortest route between SI and DI.

$maxsegment$: Maximum number of Segments between SI and DI.

b. To find the route validity for each segment in the SS, we define a binary variable that finds the route connectivity of each segment within its corresponding route. In Equation 1.28, C_i is the connectivity status of segment i.

$$C_i = \begin{cases} 1 & if avgthr(i) >= Thresholdvalue \\ 0 & otherwise \end{cases} \tag{1.28}$$

When all the segments are defined as valid within the route, the route is valid. In Equation 1.29, if the product of C_i is equal to 1, then this route is valid.

$$\prod_{i=1}^{MNSCR} C_i = 1 \tag{1.29}$$

Where:

MNSCR: Maximum Number of Segments for the Candidate Route.

In Equation 1.30, each candidate route calculates the average throughput value which equals the minimum average throughput of the segment within the route.

$$MinAvgSet = \begin{cases} Avgeragethroughput(P_1) \\ Avgeragethroughput(P_2) \\ \\ \\ \\ Avgeragethroughput(P_{z-1}) \\ Avgeragethroughput(P_z)) \end{cases} \tag{1.30}$$

Where:

z: maximum number of valid candidate route.

The optimal route based on our Algorithm is the maximum of the MinAvgSet, as in Equation 31:

$$Optimalroute = max(MinAvgSet) \tag{1.31}$$

At each intersection, the ICH forwards the packets to the first segment of the optimal route. At each intersection, the ICH dynamically recalculates the optimal route to the desired destination. Therefore, the throughput for the route to the desired destination is the product of the throughput for the first segment in the optimal route. Remember here that the optimal route is determined at each intersection in real-time. In Equation 1.32, the throughput in a grid topology is calculated by multiplying the average throughput (calculated in Equation 1.8) of the first segment in the current optimal route by the throughput of the first segment of the next optimal route at the next intersection, and so on, until arriving at the desired destination, as follows:

$$GThr = AvgThr_1 \times AvgThr_2 \times ... \times AvgThrt_n \tag{1.32}$$

Where:
$GThr$: the throughput for the route in a grid topology.
$AvgThr_n$: average throughput of first segment in the optimal route number n.

An end-to-end ($E2Edelay$) delay in a grid topology is the time that the packet takes to arrive at the destination, which is the commutative delay for the first segment of the optimal route at each intersection. In Equation 1.33, we calculate the $E2Edelay$ for any two vehicles in a grid topology.

$$E2Edelay = delay_1 + delay_2 + ... + delay_n \tag{1.33}$$

Where: $E2Edelay$: end-to-end delay between any vehicles in a grid topology.
$delay_n$: end-to-end delay between two adjacent intersections of first segment in the optimal route number n.

1.8 Simulation and analysis

By using the SUMO version 0.28.0 traffic generator and Matlab version R2016b, we evaluate the performance of our proposed protocols in different scenarios and in terms of different performance metrics, as follows:

1.8.0.1 Evaluating the CBLTR protocol

To evaluate the proposed CBLTR protocol, we implemented a bidirectional segment of 1000 meter long and 20 meter wide. The segment is divided into fixed sizes of clusters of 250 meter length. We initially distributed 40 vehicles on the segment using uniform distribution, and we gave each vehicle a constant velocity randomly selected from predefined velocity ranges, as follows: 10-30km/h, 30-50km/h, and 50-70km/h. In Figure 1.14, we present the simulation results of CBLTR, CBDRP, AODV-CV, and CBVANET protocols, respectively, in terms of average throughput and speed range. We calculate the average throughput for the segment based on Equation 1.8, assuming that the transmission rate is 2 Mbps. The simulation time is 90 seconds, and we calculate the average throughput periodically (every 5 seconds). The results show that the CBLTR protocol significantly outperforms CBDRP, AODV, and CBVANET protocols in terms of throughput and link stability. CBVANET also improves the throughput compared with CBDRP and AODV-CV; however the link is not stable, since this protocol depends only on minimum velocity to elect the CH regardless of its location, thus increasing the number of CH re-election processes as well as reducing the CHLT. In addition, the CBLTR protocol maintains higher communication stability by increasing the velocity range compared to CBDRP, AODV-CV, and CBVANET protocols.

We compare the CBLTR protocol with CBVANET and CBDRP protocols. CBVANET selects the CH based on minimum node velocity, regardless of the distance to the directional cluster edge; and CBDRP selects the CHs based on the distance to the cluster center. The CBLTR protocol shows a significant improvement in average throughput compared to the CBDRP protocol. Note that CBDRP selects the CH based on location without considering mobility parameters. However, CBVANET improves the average throughput compared to CBDRP. But our proposed protocol CBLTR outperforms CBVANET in terms of average throughput and stability.

Based on Equation 1.8, we calculate the analytical solution for the optimal throughput that we can gain from the segment. Figure 1.15 shows the analytical solution of average throughput in

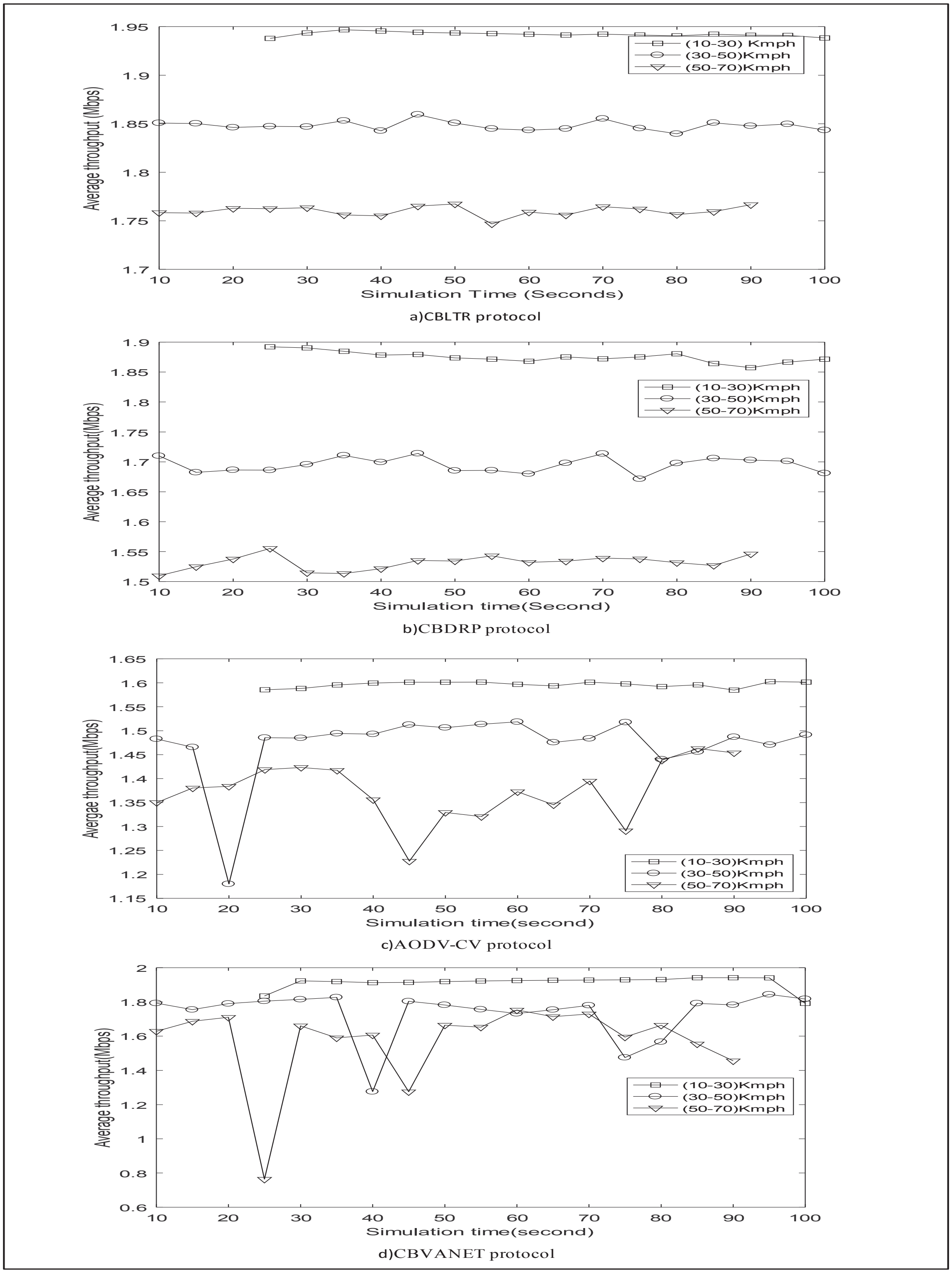

Figure 1.14 Average throughput calculation for bidirectional segment,
Figure (a) represents our proposed protocol CBLTR protocol,
Figure (b) represents CBDRP protocol,
Figure (c) represents AODV-CV protocol,
Figure (d) represents CBVANET protocol

terms of transmit time. We ran the simulation for 200 seconds. In addition we assumed speed range(10-60)km/h, and distributed 80 vehicles within the segment using poisson distribution with arrival rate 1 vehicle per second. The results start to monitor when all vehicles enter the segment. The figure clearly proves that our proposed protocol outperforms others protocols in terms of average throughput. In addition, the CBLTR protocol is closer to the optimal solution.

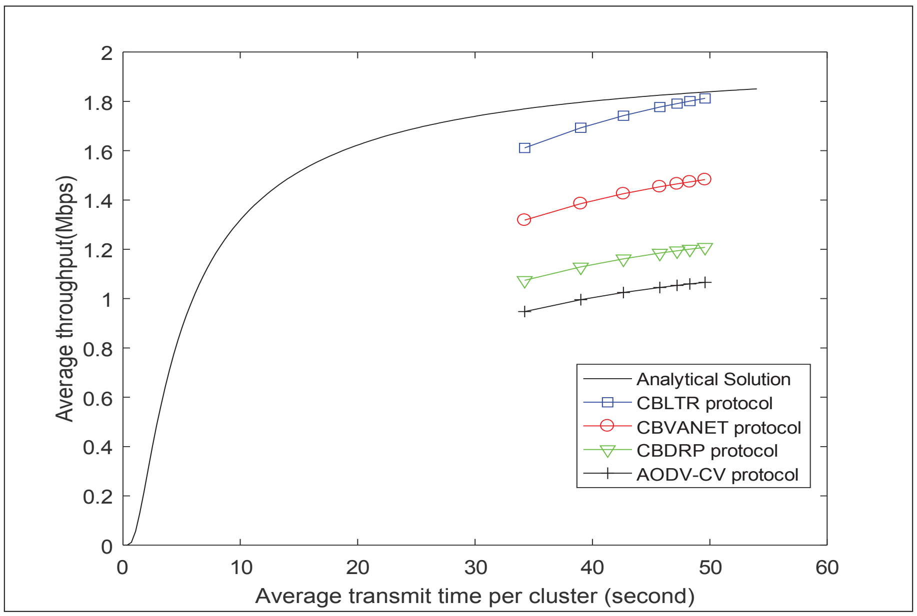

Figure 1.15 Comparison between optimal throughput with simulation results

Cluster stability can be defined through different mechanisms, but the main mechanisms are CH duration and the number of CH changes. CH duration is the period of time that the CH maintaines its status as CH; maximizing CH duration is useful to improve cluster stability, as well as minimizing the control overhead messages that yields from frequent re-election processes. The number of CH changes is the number of vehicles that change its status from CH to CM within a period of time. The analysis shows that frequently changing CH minimizes network stability (Ucar *et al.* (2013)). In contrast, the CBLTR protocol elects the CH in each cluster based on periodical calculation of the LT. The selected vehicle maintains and advertises its status as CH until it arrives at the predefined threshold point. In Figures 1.16 and 1.17, we ran the simulation for 500 seconds; then we calculated the average CH duration and the average number of CH re-election by comparing the CBLTR election algorithm with other CH election algorithms mentioned in the literature. We evaluate the performance in terms of average CH duration and average number of CH re-election processes. The results show that the CBLTR protocol outperforms other election algorithms in terms of average CH duration and the average number of CH changes.

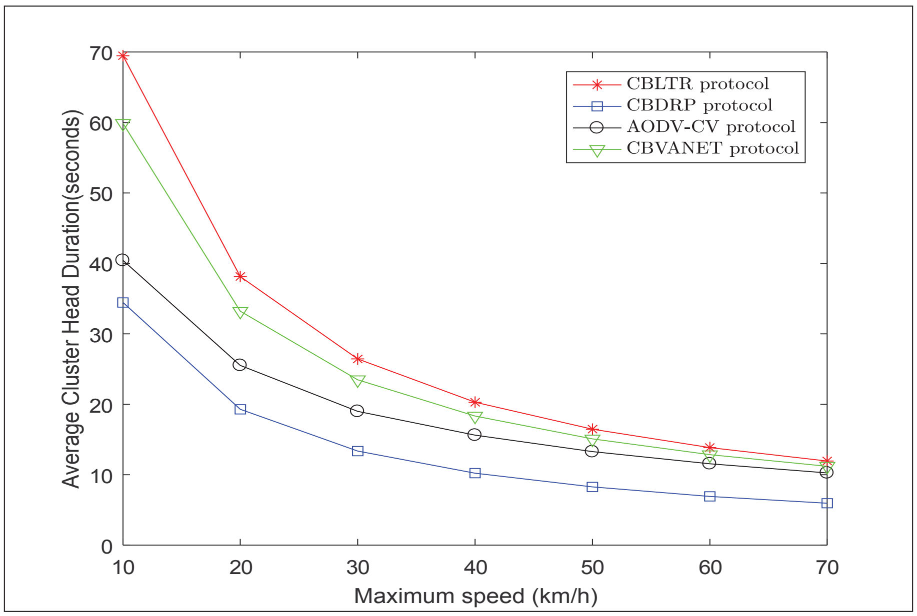

Figure 1.16 Average Cluster Head Duration vs speed

1.8.0.2 Evaluating the IDVR protocol

To evaluate the IDVR protocol, we assess the performance of the IDVR algorithm in terms of end-to-end delay and throughput for a grid topology. We compare our results with three other intersection routing protocols. The grid topology characteristic and our simulation parameters are presented in Table 1.2. The simulation scenario uses a grid topology of $4000m \times 4000m$ area that consists of 25 intersections and 40 bidirectional segments. We used SUMO to generate mobility for 500 to 1000 vehicles randomly distributed by the simulation time, and the velocities are uniformally distributed to the vehicle within a range of 10 km/h to 60km/h. The type of routing protocol used in the segment is the CBLTR protocol (Abuashour & Kadoch

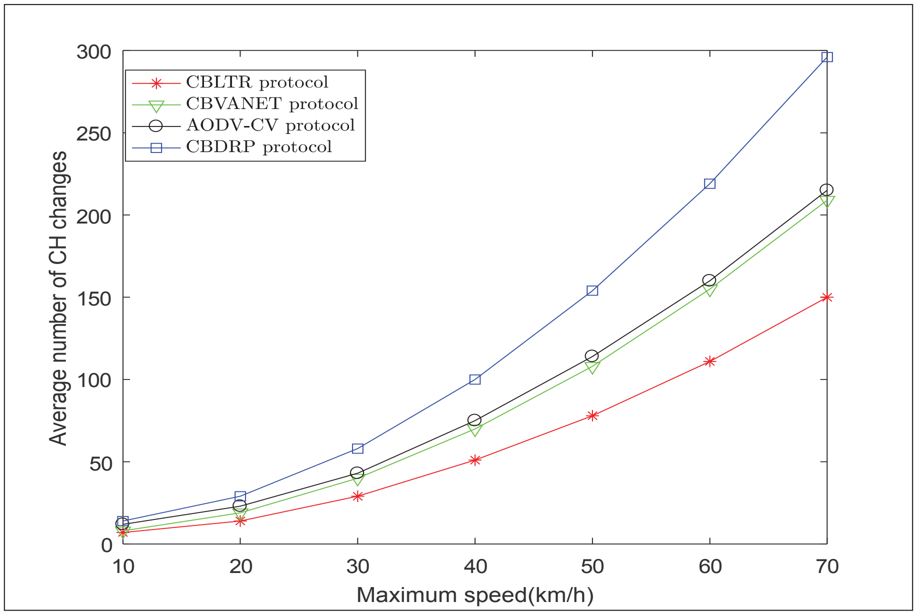

Figure 1.17 Average number of CH changes vs speed

Table 1.2 Simulation parameters for IDVR protocol

Parameter	**Value**
Simulation time	1000 second
Topology type	Grid topology
Number of intersection	25
Number of segments	40
Communication range	250
Vehicle range speed	$(10-60)kmph$
Packet size	512byte
Data sending rate	2 Mbps
Number of clusters per segment	4
Threshold value	1 Mbps
Maximum number of segments of valid route	Minimum number of segments in a valid route + 2
Type of communication in the segments	CBLTR (Abuashour & Kadoch (2016))

(2016)). CBLTR outperforms other routing protocols in terms of average throughput, by taking into account the maximum LT for selecting CH and the next forwarded nodes.

At the intersection, the IDVR protocol selects the next street based on the stability for the route between the source and destination. IDVR uses the CBLTR protocol to propagate the packet within the segment. In addition, it takes into account the route validity, so that all the segments within the selected route are connected and stable at each intersection. IDVR is a real-time dynamic protocol; each time the packet reaches the intersection, ICH recursively applies IDVR protocol between the current intersection and the desired destination intersection. In the literature, many researchers are investigating intersection routing protocols, such as VDLA, IRTIV, and GPCR. VDLA adopts sequential selection of intersections to construct the routes; the selection is based on real-time traffic density, the traffic load of the corresponding road segment, and the distance to the destination. IRTIV aims to find the shortest connected route to the destination in a city scenario, by taking into account the real-time segment density estimated by a completely distributed approach based on the periodic exchange of Hello messages. GPCR selects the next street that has a node with the shortest route to the destination.

In Figures 1.18 and 1.19, we compare the IDVR protocol, VDLA, IRTIV, and GPCR in terms of throughput and end-to-end delay based on Equations 1.32 and 1.33, respectively. As in our simulation results, we prove that the IDVR protocol significantly outperforms VDLA, IRTIV, and GPCR in terms of end-to-end delay and throughput.

1.8.0.3 Evaluating the CORA algorithm

To evaluate the performance of CORA protocol, we implemented a bidirectional highway scenario with length 10000 meters, then we divided the highway to fixed sizes of clusters of length 250 meter each. In Table 1.3, we present the simulation parameter we used. The vehicles enters the highway scenario in fixed rate which equals 1 vehicle/sec. The simulation starts to gather the results after all vehicle enter the Highway scenario. We calculate the number of CMHELLO messages in each cluster, to be more fair in our comparison we assumed that the vehicles use the same architecture of HELLO message, since in general we want to validate the CORA algorithm in terms of reducing the number of CMHELLO messages. In Figure 1.20, we compare our results with three other protocols mentioned in the literature; CBDRP, BRAVE,

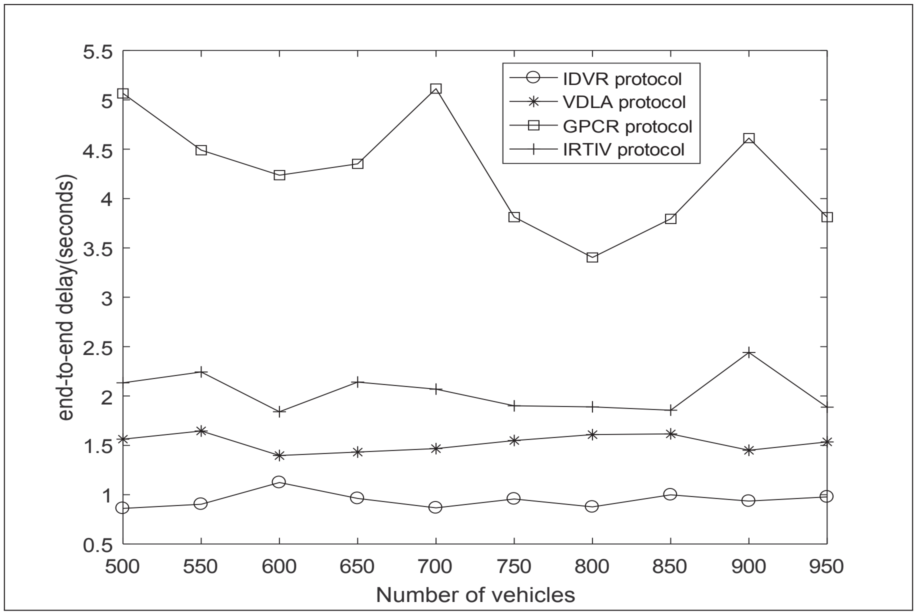

Figure 1.18   End-to-end delay comparison in a grid topology

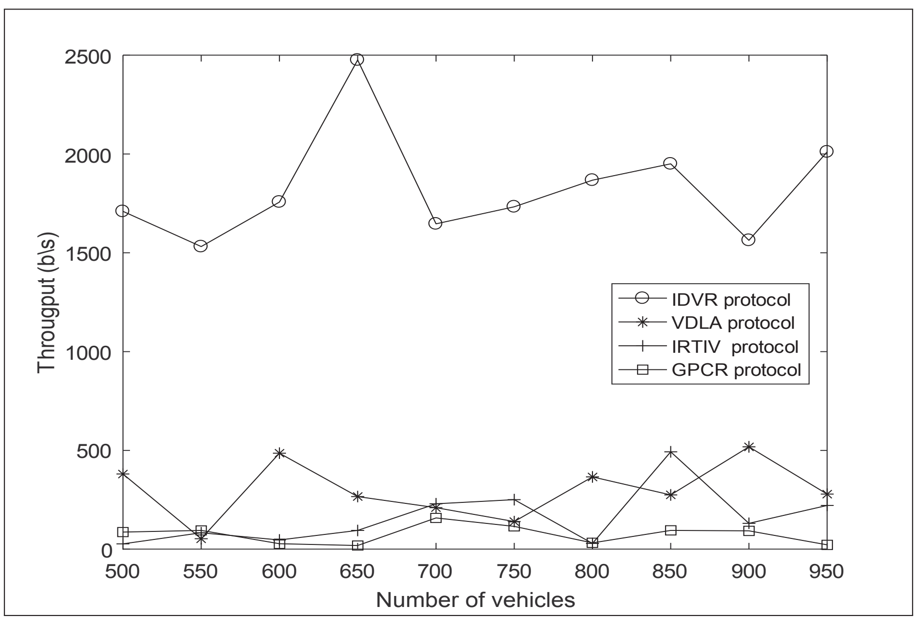

Figure 1.19   Throughput comparison in a grid topology

and MoZo protocols. In CBDRP protocol, the CMs in each cluster are updated very quickly, and this yields to produce many HELLO messages. In BRAVE protocol, the HELLO interval is 2 second. In MoZo protocol, the authors assume that the vehicles need to send HELLO

Table 1.3 Simulation parameters of CORA algorithm

Parameter	**Value**
Simulation time	500 second
Topology type	Highway scenario
Number of cluster	40
Number of vehicles in each direction	100
Vehicles arrival rate	1 vehicle/sec
Communication range	250
Vehicle range speed	$(10-60)kmph$
CH protocol used	CBLTR (Abuashour & Kadoch (2016))

updates messages when they deviate from their defined original moving function more than 5 m/s or the time from the last update which equals to 4 seconds.

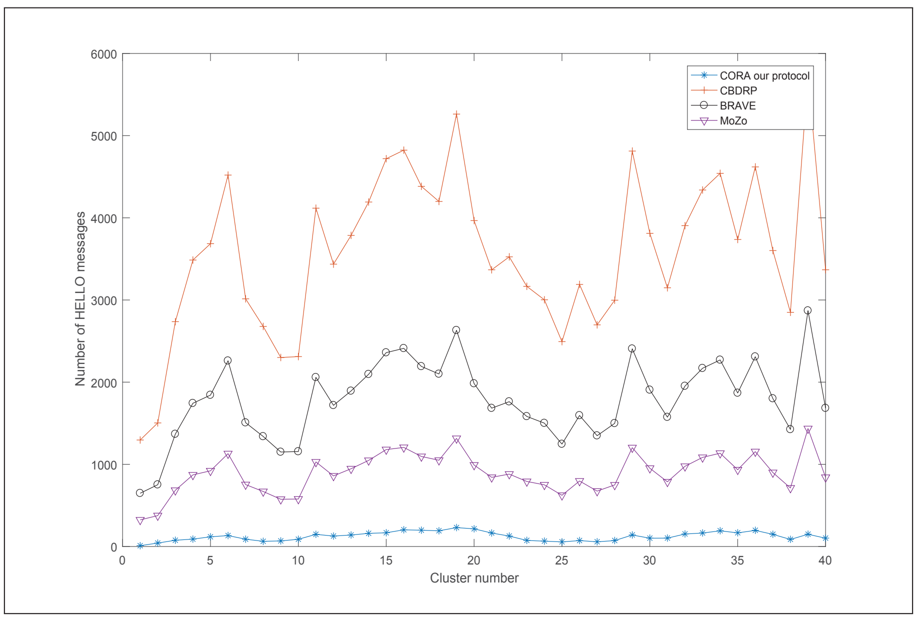

Figure 1.20 Number of HELLO message in highway scenario

CORA outperforms all previous protocols in terms of the number of CMHELLO message. CORA protocol minimizes the CMHELLO message by avoiding periodically exchanging of

HELLO messages. CORA propagate the CMHELLO messages in three scenarios which are: when the CM enters the cluster zone, second; when the CM leave the cluster zone, and when new CH announces itself. In general, CORA calculate the optimal number of CMHELLO messages in each cluster.

1.9 Conclusion

This article proposed three algorithms that improve the performance of CBR protocols in any VANET environment. First; a novel Cluster-Base Life-Time Routing (CBLTR) protocol in a segment topology is introduced. The CHs are elected based on maximum LT, and the re-election process is required only when the CHs reach their corresponding threshold point. Based on the simulation results, CBLTR protocol shows a significant improvement in terms of average throughput. The enhancement in CBLTR protocol is a new mechanism to select new CHs. The selected CHs have longer LT span making the protocol more stable.

Second; an Intersection Dynamic VANET Routing (IDVR) protocol in a grid topology is proposed. Each time the packet reaches the intersection, ICH recursively applies the IDVR protocol between the current intersection and the desired destination intersection, taking into account the stability of the connected route. The IDVR protocol selects the optimal route based on its current location, destination location, and a maximum of the minimum average throughput for SCSRs. IDVR increases the overall network efficiency, by increasing the route throughput, and decreasing end-to-end delay. As in our simulation, we have proved that the IDVR protocol outperforms VDLA, IRTIV, and GPCR in terms of end-to-end delay and throughput.

Finally; we proposed a Control Overhead Reduction Algorithm (CORA), which aims to reduce the control overhead messages in the clusters, by developing new mechanism for calculating the optimal period for updating or exchanging control messages between the CMs and the CH. CORA propagate the HELLO messages in three scenarios: when the CM enters the cluster zone, second; when the CM leave the cluster zone, and when new CH announces itself. Based in the simulation results, CORA significantly minimized the number of HELLO messages in each cluster and in the segment with multiple clusters in general.

CHAPTER 2

CONTROL OVERHEAD REDUCTION IN CLUSTER-BASED VANET HIGHWAY TOPOLOGY

Ahmad Abuashour[1], Michel Kadoch[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper submitted for publication, September 2017

2.1 Abstract

Vehicular Ad-Hoc NETwork(VANET) is a unique form of Mobile Ad-Hoc NETwork(MANET), where the nodes act as vehicles moving with relatively high mobility, and moving in a predefined routes. The mobility in VANETs causes high topology changes and in turn leads to excessive control overhead messages and frequent link communication failures. Traditionally, clustering techniques have been used as the main solution to reduce the control overhead messages in VANET, in which the network is divided into multiple clusters and selecting one of the Cluster Members (CMs) as a Cluster Head (CH). Still, a problem occurs when the control overhead messages increase due to periodically forwarding of CM HELLO (CMHELLO) messages between the CMs and the CH, and when the CH periodically broadcasts an CH ADvertiSement (CHADS) messages to declare itself to the CMs. Hence, minimizing control overhead messages in any cluster environment is an essential goal to use the network resources efficiently. In this article, we propose two algorithms: First, a Control Overhead Reduction Algorithm (CORA) which aims to reduce the control overhead messages in a clustered topology, by developing a new mechanism for calculating the optimal number of CMHELLO messages. Second, an Enhanced version of CORA (ECORA) which aims to reduce the CHADS messages that broadcasted by the CHs, by proposing a CHADS prediction algorithm that enables the CH to predict the period of time for broadcasting the CHADS messages. In addition, we present a new format of control overhead messages that reduce the number of parameters transmitted.

Finally, we evaluate the performance of our proposed works by comparing with other recent researches that published in this field.

2.2 Introduction

The Intelligent Transportation System (ITS) that includes all types of communications between vehicles is an important next-generation transportation systems. ITS provides many facilities to the passengers, such as safety applications, assistant to the drivers, emergency warning, etc. Vehicular Ad Hoc NETwork (VANET) is a derived form of self-organized Mobile Ad Hoc NETwork (MANET). In VANET, vehicles are equipped with an On-Board Units (OBUs) that can communicate with each other (V2V communications), or/and with stationary road infrastructure units (V2I) that are installed along the roads. VANETs have several characteristics that makes it different from MANETs; such as high node mobility, predictable and restricted mobility patterns, rapid network topology change, and frequent battery charging, so energy consumption is not a big issue(Tayal & Triphathi (2012)).

Dedicated Short Range Communication (DSRC) technology is an emerging technology that is developed to work in very high dynamic networks, to support fast link establishment and to minimize communication latency. DSRC is designed to ensure the reliability of safety applications, taking into consideration the time constraints for this type of applications. In the United States, Federal Communication Commission (FCC) has allocated 5.9GHz for DSRC technique to support public and commercial application in V2V and V2I. The frequency takes the range of (5.850-5.925) GHz and divides it into seven non-overlapping 10MHz channels. The DSRC is developed to support the data transfer in a rapidly changing topology networks, such as VANET, where time response and the high transmission rate is required. VANETs deal with two wireless access standards: IEEE 802.11p deals with the physical and MAC layer, and IEEE 1609 deals with higher-layer protocols. According to IEEE 802.11p, vehicles are capable to share their GPS related position together with velocity and acceleration (551 (2010)).

VANET is proposed and improved different types of routing protocols, such as proactive (Spaho *et al.* (2012)), reactive (Ding *et al.* (2011); Abedi *et al.* (2009); Yu *et al.* (2011)), and hybrid (Wu *et al.* (2013) Chai *et al.* (2013)). The proactive and reactive routing protocols are classified under the topology based routing protocol category, which aims to discover the route between the source and destination before starting the data transmission. The main difference between the two is that the proactive routing protocol initiates a route discovery to the all nodes located in the entire network, yielding an increase in control overhead messages and end-to-end delay. While in the reactive routing protocol, a source node initiates a discovery process to reach only to the desired destination. This process reduces the routing control overhead messages; however, the route discovery process is required in finding a route for every new node. The hybrid routing protocol combines the features of both proactive and reactive routing protocol. The nodes in the hybrid network are grouped together in a particular area called clusters. Hybrid routing protocols, and sometimes called Cluster-Based Routing(CBR) protocols, are designed to improve the network scalability by allowing the nodes within the clusters to communicate through a pre-selected Cluster Heads (CHs) using a proactive routing protocol. However, in the case of communication between clusters, a reactive routing protocol is triggered.

CBR protocols are widely used to improve the scalability of VANET environment and to reduce the control overhead message. Although the clustering techniques are minimizing the routing control overhead messages, clustering control overhead messages still consume much of VANET resources. The clustering control overhead messages are produced by : First, forwarding of control messages between the CMs and the CH named CMHELLO messages. Second, broadcasting of control messages by the CH named CHADS messages. When control overhead messages are increasing in any clustered topology, the available bandwidth resources reduce significantly. The CBR protocols aims to increase the network throughput at the cost of increased the clustered control overhead messages. The main objectives of this article are minimizing the number of control overhead messages in CBR protocols and increasing the available resources in a clustered highway topology.

In this article we proposed two algorithms: First; a new algorithm that reduces the number of CMHELLO messages, called CORA algorithm. Second; an Enhanced version of CORA (ECORA) algorithm that aims to reduce the number of CHADS messages. After that, we present a new message format for CMHELLO and CHADS. In addition, we proposed a new type of CH messages named CH notification messages. CORA and ECORA are based on the assumption that each vehicle in the VANET environment can know its current location and cluster ID by using a digital map and Global Positioning System (GPS). Also, in this article we used a Cluster-Based Life-Time Routing (CBLTR) protocol as the main cluster routing protocol, which outperforms many other cluster protocols in terms of increasing the average throughput and stability in clustered network. The CBLTR (Abuashour & Kadoch (2016)) protocol selects the CH based on maximum LT among the CMs.

This article is outlined as follows; in Section 2.3 we present a literature review that relates to control overhead reduction techniques in clustered topology. Section 2.4 presents CORA algorithm. Section 2.5 presents ECORA algorithm. Section 2.6 presents the new format of clustered control overhead messages. Section 2.7 presents the simulation, results and analysis. Finally, section 2.8 concludes this article.

2.3 State of arts

Control overhead reduction techniques are an important and interesting subject in many recent researches. The main objective of minimizing the control overhead messages is improving the network efficiency by producing more bandwidth resources for data transmission.

VANETs are an autonomous systems formed by connected vehicles without the need for any infrastructure. Routing in VANET is a significant challenge due to the nature of fast topology changes. The high mobility in VANET forces the vehicles to periodically exchange control overhead messages. Therefore, the excessive amount of control overhead messages yields to consume high amount of available bandwidth resources.

The main solution to reduce the control overhead messages is to use the clustering technique. The concept of clustering means to transform the big network into small grouped networks called clusters. In each cluster, one of cluster members (CMs) should be elected to be responsible for all local cluster communication, and it is called Cluster Head (CH). This process will significantly reduce the routing control overhead messages because it restricts the communication between each CM and CH instead of exchanging the control overhead messages between all the vehicles in the VANET topology. Many researches proposed several algorithms of selecting the CH in any cluster topology based on specific parameters, such as: vehicle ID, vehicle location, vehicle speed, vehicle direction, and vehicle Life-Time (LT). In this article, we assume that the CHs are elected mainly based on maximum LT (Abuashour & Kadoch (2016)). The elected CHs based on maximum LT significantly outperforms many other CBR protocols that takes into consideration other parameters. The process of electing CH is out of scope in this article. In general, dividing the network into multiple clusters reduces the communication overhead and improves the network efficiency.

In the cluster, CMs and CH should periodically forward and broadcast the control overhead messages. The CMHELLO message is an important control overhead messages used to define the vehicle identity and location in clustered VANET network. The number of CMHELLO messages is mainly in inverse proportion to the period of time for forwarding the message. So, in any cluster if the period of time increases, then the number of CMHELLO message decreases, and vice versa. In addition, the number of CMHELLO messages in any cluster is also in proportion to the number of CMs in the cluster. The CHADS message is another type of message that is broadcasted periodically by the selected CH to announce itself. Many techniques are proposed in the literature to reduce the number of the clustered control overhead messages as follows:

In (Lo *et al.* (2013)), the authors proposed a new clustering algorithm that intends to create stable clusters by reducing clustering overhead. This algorithm proposed to select the CH based on the vehicle position and speed. The proposed algorithm intended to increase the clusters stability by reducing the number of CH changes, which yields to reduce the control overhead messages produced from frequently reelection process . This algoritm assumes that each node should periodically broadcast A HELLO message that contains the node's location, motion vector, and RBM (Relative Position and Mobility). In this paper (Lo *et al.* (2013)), the authors do not mention the impact of the size of CHADS message, as well as they are not considering the impact of the CMHELLO messages in terms of its size and its updating period of time.

In (Mohammad & Michele (2010)), the authors proposed a lane-based clustering algorithm to improve the network stability by reducing the CH election times. The proposed algorithm elects the CH based on the traffic flow of vehicles in the cluster. In fact, this approch deals with dynamic clusters, which reduces the number of CH changes in the cluster. Each vehicle broadcasts a control message that contains vehicles lane location, vehicle location, and vehicle speed. Also, this paper (Mohammad & Michele (2010)) does not mention the size and the period of time for broadcasting the control overhead messages.

In (Rawashdeh & Mahmud (2012)), the authors considered a new parameter to improve the CH election, which is the speed difference metric. By using this parameter, the cluster becomes categorized based on different speeds. In this paper (Rawashdeh & Mahmud (2012)), all vehicles broadcast periodically within it's transmission range a control overhead message that contains the position, velocity, node degree, and direction.

Tao et al. (Song *et al.* (2010)) proposed a Cluster-Based Directional Routing Protocol (CBDRP) for highway scenario. The vehicles broadcasts periodically the control overhead messages that contain location of the cluster, location of the vehicle, and the velocity of the vehicle. A CH distributed algorithm is used to select one CH among CMs. The selected CH has full information about its CMs. CBDRP concentrates in reducing the routing overhead packet from source to destination, but it does not considers the control overhead messages produced by the vehicles in each cluster.

Pedro et al. (Ruiz *et al.* (2010)) proposed a Beacon-less Routing Algorithm for Vehicular Environment (BRAVE). The proposed protocol objective is to reduce the control overhead messages in broadcast approaches. In BRAVE, the next forwarder vehicle is reactively selected among those neighbors that have successfully received the messages. The drawback of BRAVE protocol is that each vehicle participates in the routing protocol and still required to exchange a beacon message among them. In the simulation setting, BRAVE sets the exchanging time of the beacon messages to 2 seconds to keep monitoring the vehicles location. In general, reactive routing protocol reduce the control overhead messages compared to proactive routing protocol. However, it still suffering of high control overhead messages compared to CBR protocols.

Dan et al. (Lin *et al.* (2017)) proposed a MOving-ZOne-based (MoZo) architecture, MoZo consist of multiple moving zones that group vehicles based on the movement similarity. the selected CH is responsible for managing information about CMs as well as the forwarding packets. The control overhead messages updating period for the CMs in MoZo architecture is varied between the changing of moving function of 5 m/s or 4 seconds.

In (Khakpour *et al.* (2017)), the authors proposed two clustered-based algorithm for target in VANET. The objective of these algorithm is reducing the clustered control overhead messages by using a prediction function to predict the CM information. If the CH can predict the CM information instead of receiving it, then the control overhead messages will be decreased significantly. The prediction function requires the current location, speed, and time to predict the future location. The drawback of these algorithms is that the CH should trigger the prediction function periodically to ensure the prediction accuracy, and this also yields to increase the clustered control overhead messages. In (Singh *et al.* (2014)), a periodically live message is broadcasted by every node for announcement of it's existence in the cluster. Also, this paper does not consider the live message size and the period of updating these messages in its evaluation.

In the literature covered up to date, the authors did not provide any guidelines to exploit the cluster resources. Though the main properties of any clustering algorithm are the high amount of the CMHELLO message and CHADS messages. In addition, most of the literature is ignoring the control overhead message size, by considering many parameters to be appended to it, such as: speed, direction, location, CM ID, and etc. To the best of our knowledge there are no researches that investigated the reduction of the clustered control overhead message by optimizing the number of generated control overhead messages in a clustered VANET topology. Therefore, in this article we propose a CORA and ECORA algorithms that optimize the number of CMHELLO and CHADS messages, respectively.

2.4 Control Overhead Reduction Algorithm

In VANET, the CBR protocols do not require that every vehicle knows the entire topology information. Only the selected CHs require to know the topology information and other CMs only require to periodically forward their information to the CH via CMHELLO messages. CM periodically forwards a CMHELLO messages to inform the CH about CM identity and other parameters; such as current location, direction, velocity, and LT. The increasing size of the control overhead messages consider an important issue that degrade the performance of any mobile and limited resources networks. Furthermore, the frequently forwarding of the CMHELLO message negatively impact the network performance which significantly reduces the available resources. In other hand, CHADS is another control overhead message broadcasted periodically by the CHs. The purpose of this kind of messages is to keep announcing the CH presence in any cluster, and to inform any new vehicle entering the cluster zone of the CH identity.

In the next two subsections, we present the proposed CORA algorithm and the mathematical calculation for the number of control overhead messages.

2.4.1 CORA algorithm

In general, each vehicle must be defined as CM or CH at any time. Algorithm 2.1 explains the CORA algorithm as follows; initially, each vehicle entering any cluster coordination zone sets its status as CM by default. Then it waits for τ second, if it does not receive any CHADS message, it changes its status to CH and starts periodically (every τ second) broadcasting CHADS message. This message consists of CH Identification information and the remaining LT that the CH spends in the cluster zone. Otherwise, it stays as CM and replies with only one CMHELLO message which consists of the CM Identification and the remaining LT that the CM expect to spend in the cluster zone. The remaining LT is varied among vehicle due to the velocity variation. The objective of periodically broadcasting CHADS message is to inform newly-arrived CMs that an active CH exist. When the CH receives all replies from the CMs within its associated LT, the CH is capable to calculate the candidate CH (CCH) before leaving the cluster. Therefore, the CMs do not require to periodically update their information with the CH while the CHLT is not expired. In other word, the CMHELLO messages produced by the CMs are proportional to the number of CH changes instead of specific period of time. Thus, that yields to minimize significantly the control overhead messages in each cluster.

Algorithm 2.1 CORA protocol

```
for t=1 to end of simulation time do
    if any vehicle enters the cluster Zone then
        vehicle staus = CM;
        waitτ sec;
        if CM receives CH ads message then
            vehicle staus = CM;
            reply to CH by one message;
            Containes <CMID, CMLT>;
            else
                vehicle staus= CH ;
                every τ sec send CH ads
            end
        end
    end
end
```

2.4.2 Mathematical calculation of control overhead in CORA algorithm

To calculate the number of CHADS message within the simulation time, first we divide the elected CH remaining LT time by the period of exchanging time τ (τ is a constant value), as in

Equation 2.1:

$$AdsCH_{ijk} = \frac{CHLT_{ijk}}{\tau} \tag{2.1}$$

Where:

$AdsCH_{ijk}$: The total number of CHADS messages produced from CH with ID i in cluster j in segment ID k.

$CHLT_{ijk}$: The remaining LT for CH with ID i in cluster ID j in segment ID k.

τ : The periodic exchanging time for CHADS message.

Next, we calculate the overall CHADS messages for all elected CHs in the same cluster within the simulation time, as in Equation 2.2:

$$\begin{aligned} TotalAdsclus_{jk} &= \sum_{i=1}^{x} AdsCH_{ijk} \\ 0 < TotalAdsclus_{jk} &< simulationtime \end{aligned} \tag{2.2}$$

Where:

$TotalAdsclus_{jk}$: The number of CHADS message produced from CHs in cluster ID j in segment ID k.

x : The maximum number of elected CH within the simulation time for cluster j.

To calculate the total CHADS messages that generated in a segment with multi-cluster, we do the summation for the number of CHADS messages for each cluster, as follow:

$$\begin{aligned} TotalCHAds_k &= \sum_{j=1}^{y} TotalAdsclus_{jk} \\ &= \sum_{j=1}^{y} \sum_{i=1}^{x} \frac{CHLT_{ijk}}{\tau} \end{aligned} \tag{2.3}$$

Where:

$TotalCHAds_k$: The number of CHADS message produced by CHs in segment ID k.

$CHLT_{ijk}$: The remaining LT for CH with ID i and Cluster ID j in segment ID k.

y : The maximum number of clusters within the segment.

Since τ is constant value, then the number of CHADS messages that produced by the CH is proportional to the CHLT value in each cluster. In Figure 2.1, the CH forwards CHADS messages every τ seconds to all of its CMs until its LT expires. Each selected CH should periodically forward the CHADS messages to announce itself in the cluster zone. The vehicles A,B,C, and D are CMs that receive CHADS from the CH while its LT time does not expire.

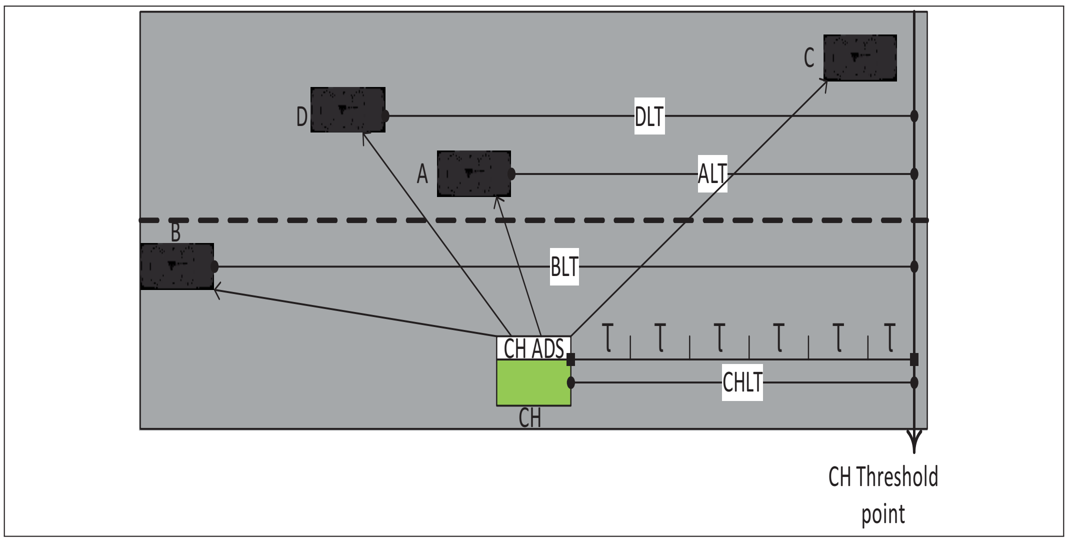

Figure 2.1 CHADS message periodically (every τ seconds)

On the other hand, when any vehicle enters the cluster zone, its default status is CM. It should exchange the CMHELLO message with the CH. So, in this article the main contribution is to minimize the number of CMHELLO messages by taking into consideration CHLT. When any vehicle enters the cluster zone, it sends a CMHELLO message to the CH (if it receives the CHADS after τ second). In this case we have two scenarios; if the CMLT is greater than CHLT, then the number of CMHELLO message equals to the number of CH changes within the CMLT plus two (the mandatory two CMHELLO messages when the CM enters the cluster and before it leaves the cluster). Otherwise, the CM generates the CMHELLO message only two times; when it enters the cluster and before leaves the cluster. Figure 2.2 explains a scenario of forwarding CMHELLO messages; first, when vehicles enter the cluster zone (as vehicle B), it should then send CMHELLO message, and when the vehicle leaves the cluster zone, it then sends another CMHELLO message(as vehicle C), whereas the vehicles (vehicle A and D) that already in the cluster zone and within the CHLT do not require to send any CMHELLO message. Figure 2.3 explains another scenario when the CH (Old CH) arrives to the threshold point (the point that the current CH should select another CH), the old CH sends an CHADS message informing the CMs for the new CH, in the meantime; all the CMs (vehicle A and D) should send the CMHELLO message to the new CH.

The following Equation describes mathematically the two scenarios in Figure 2.2 and 2.3:

$$NumCM_{ijk} = \begin{cases} numCH_{ijk} + 2, & \text{if } CMLT_{ijk} > CHLT_{jk} \\ 2, & \text{if } CMLT_{ijk} \leq CHLT_{jk} \end{cases} \tag{2.4}$$

Where:

$NumCM_{ijk}$: The number of CMHELLO message produced by CM with ID i in cluster ID j in segment ID k.

$numCH_{ijk}$: The number of CH changes within $CMLT_{ijk}$.

$CMLT_{ijk}$: The remaining LT for CM with ID i in cluster ID j in segment ID k.

$CHLT_{jk}$: The remaining LT for current CH with in cluster ID j in segment ID k.

The following algorithm explains how to calculate the overall CMHELLO messages in each cluster:

Algorithm 2.2 TOTAL NUMBER OF CMHELLO MESSAGES

```
Input  : total of CM ads = 0
for i=1 to Max number of CMs do
    if CMLTi>CHLT then
        total of CM ads =2 + Num of CH changes Within CMLTi ;
        else
            total of CM ads =2 + total of CM ads
        end
    end
end
Output: return total of CM ads
```

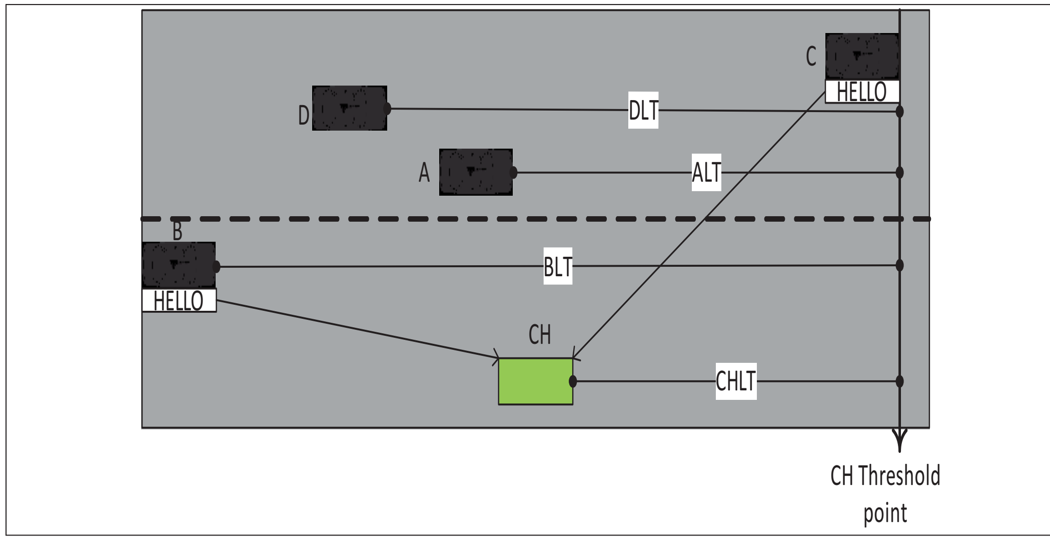

Figure 2.2 CMHELLO message when enters and leaves the cluster

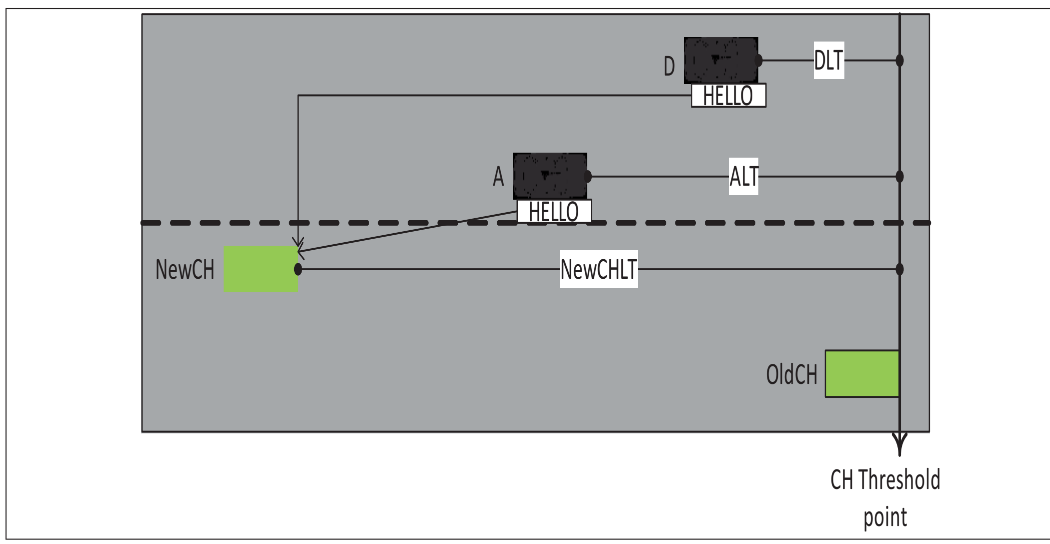

Figure 2.3 CMHELLO message when new CH selected

We can mathematically formulate the total of CMHELLO messages for a specific cluster by following expression:

$$TotalHELLO_k = \sum_{j=1}^{y} NumCM_{ijk} \tag{2.5}$$

Where:

$TotalHELLO_k$: The total number of CMHELLO messages produced from CMs in cluster k.

y : The maximum number of CM in the cluster ID k.

Also, we can mathematically formulate the total of CMHELLO messages for specific pre-divided cluster segment as the following Equation:

$$TotalCMHELLO_m = \sum_{j=1}^{p} TotalHELLO_j \tag{2.6}$$

Where:

$TotalCMHELLO_m$: The total number of CMHELLO message produced from CMs in segment ID m.

p : The maximum number of clusters in the segment ID m.

Finally, the total control overhead messages within the simulation time equals the summation of CMHELLO messages forwarded by the CMs and the periodically CHADS messages broadcasted by the CHs in segment ID k. As in the following Equation:

$$TotalAdsmesage_k = TotalCMHELLO_k + TotalCHAds_k \tag{2.7}$$

2.5 An Enhanced version of Control Overhead Reduction Algorithm

In this section, we propose an Enhanced version of CORA (ECORA) algorithm. In CORA algorithm, we eliminate the periodical updates of CMHELLO message, since the CMHELLO message are produced only in three instances; when the CM enters and leaves the cluster zone, and when the CH arrives at the predetermined threshold point. Therefore, the number of CMHELLO messages is mainly independent of time. CORA significantly reduced the number of CMHELLO messages. On the other hand, CH still depends on time for broadcasting the CHADS messages. As we explained before, the CH should announce it’s status periodically to inform any new vehicle that joins the cluster. Therefore, the number of CHADS messages generated by the CH are proportional to the CHLT divided by the CH updating period(τ seconds). CORA algorithm mainly solved the problem of frequently forwarded the CMHELLO messages; however, CHADS messages still depend on time.

Therefore, we propose in this section a ECORA. ECORA aims mainly to reduce the number of CHADS messages. To solve this problem, we build a CHADS prediction algorithm that helps the CHs to broadcast the CHADS in specific period of time instead of broadcasting within all the CHLT. Therefore, each CH should predict the expected period of time that any CM may enter to it's cluster zone.

The main characteristic of VANET that make it different than MANET is that the vehicles move in predictable and restricted mobility pattern. Which means if you know the initial location and speed of any vehicle, then you can predict any vehicle's location in the future. In other word, if the CH is capable to predict the period of time when any CM may enter its cluster zone, then the CH can broadcast the CHADS message only in this period. In the following subsections, we explain in details a novel CHADS prediction algorithms, the mathematical for calculate the number of the control overhead messages , and the CH broadcast time prediction.

2.5.1 CHADS prediction algorithms

When any CH receives a CMHELLO message associated to one of its CM, this mean that the CMID attached to this message intends to leave the cluster zone within CMLT. Then the CH informs it's CH neighbors of the prediction time of the arriving new CM/CMs to their cluster zone. As soon as the neighbor CHs receives the CH notification, they calculate a period of time to broadcast the CHADS message (more details in Subsection 2.5.3).

As we explained in CORA algorithm, if the CM did not receives the CHADS message within τ seconds, then the CMHELLO messages are forwarded after τ seconds since the CM enters the cluster zone. In ECORA, we propose a CHADS prediction algorithm that is capable to let the CH to predict a period of time for broadcasting the CHADS messages instead of broadcasting during all CHLT. Figure 2.4 explains in steps the main idea of CHADS prediction algorithm.

First, we present a snapshot of clustered highway scenario. Let us assume that these clusters are located within the highway topology (in the middle) and not located at the highway edges (in the boundaries). Also, the vehicles inside the circles A2, B5, and B8 are the CHs of the clusters n-1, n, and n+1, respectively. In addition, the rest of the vehicles are CMs. The vehicles inside the rectangle are CMs that are intending to leave their clusters zone. If any CH detects a new CMHELLO message for CMs intending to leave their cluster zones, then it should forward CH notification to the neighbor CHs. This notification informs the neighbors CHs of new arriving vehicles within a specific period of time. CH who receives these notifications should prepare itself for new arrival vehicles within the received specific period of time. For example, when the CHs A2 and B8 receive CMHELLO messages from vehicles A3 and B6, respectively, then each of the CH forwards a notification message for its neighbor CHs (as vehicle B5 in our example) to prepare itself for any new arrival vehicles. As soon as B5 receives any notification message from any neighbor CHs, then it start broadcasting CHADS message within the received specific period of time, and so on. Since this algorithm forwards the CHADS message only when the CH receives any notification from the neighbor CHs. Then, as a results of this algorithm, we are capable to optimize the number of CHADS to be proportional to the number of notification messages received by the CH instead of CHLT.

In Figure 2.5, we define a transition diagram with four modes for the CH, which are: sleep, active, notification, and broadcast modes. We explain each mode as follows: First, CH sleep mode, it is the default mode for all other modes when they are not receiving any CMHELLO message. In this mode, the CH stays in the idle state and it does not transmit any CHADS messages. The CH changes its mode in two cases; the first case it changes to the active mode at any time it receives any CMHELLO message; and the second case it changes to broadcast mode when it receives a notification message from any neighbor CHs. Second, CH active mode, in this mode the CH changes its mode in three cases; the first case is when the CH receives the CMHELLO message. If the CMHELLO message informs of new vehicle arrival (vehicles enter the cluster zone), then the CH stays in the active mode at time t; after that, if it does not receive another CMHELLO message from new entering vehicle, then it flips back

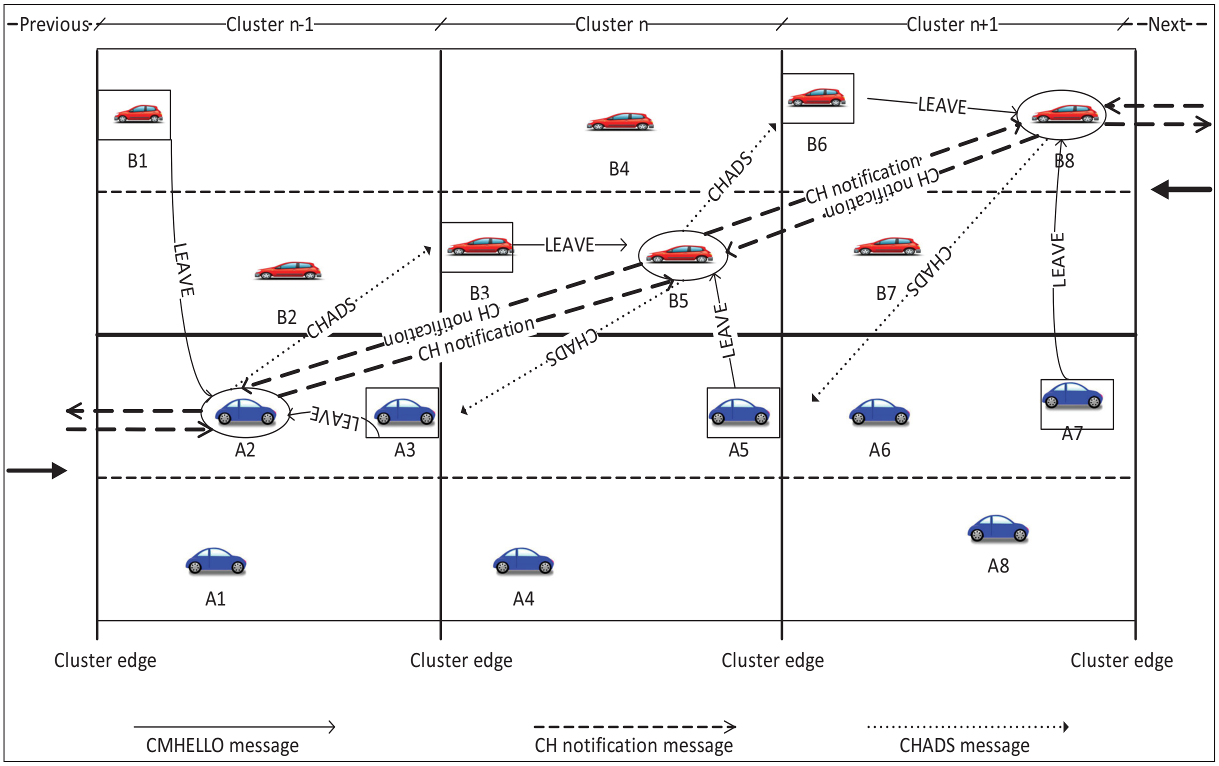

Figure 2.4 CHADS prediction algorithms

to the sleep mode. The second case when the CH receives the CMHELLO message from a leaving vehicle, it changes to the notification mode. The third case, if the CH receives at any time a CH notification message then it changes to the broadcast mode. Third, CH notification mode, in this mode the CH forwards to the neighbor CHs a list of CM/CMs that are leaving their current cluster zone and moving to another cluster. Fourth, CH broadcast mode, in this mode the CH starts to broadcast its CHADS messages for a period of τ second, and after this period the CH flips back to the sleep mode.

Algorithm 2.3 presents a pseudo code for CHADS prediction. The default mode for any CH is the sleep mode, and in this mode the CH keeps silent. At any time, the CH changes to notification mode if it receives a CMHELLO message tagged with LEAVE (lines 3-7), in this mode the CH forwards a notification messages to its CH neighbors. Each notification mode includes the CMID, and the its leaving time. Also, at any time, if any CH receives any notification message,

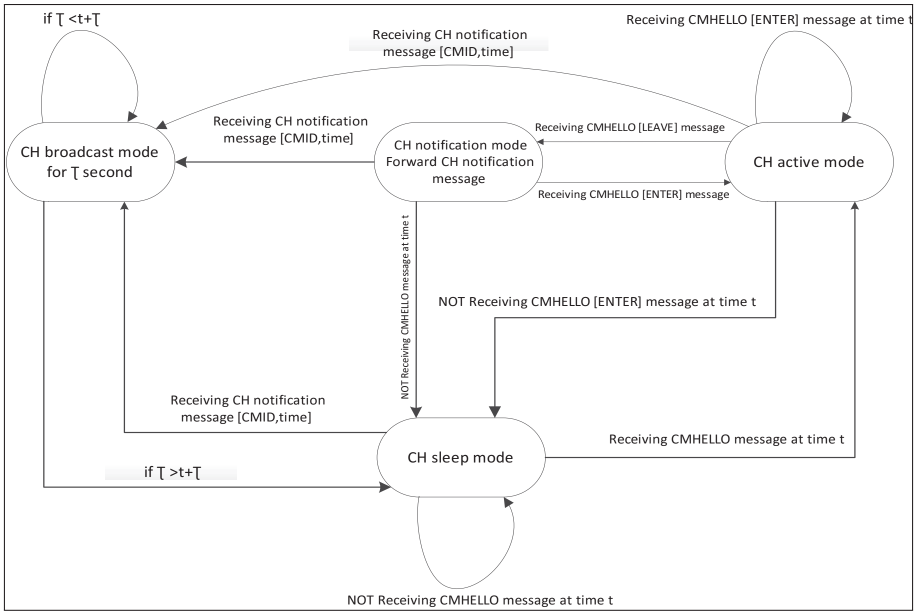

Figure 2.5 State chart of CH transition modes

then it changes its status to broadcast mode (lines 8-11), and start broadcasting the CHADS within τ from the time that the CH received in the notification message.

Algorithm 2.3 CHADS prediction algorithm

1 **for** *t=anytime to t<=simulation time* **do**
2   CH mode = sleep ;
3   **if** *CH receives CMHELLO message* **then**
4   CH mode = notification ;
5   CH forward notification message[CM, time]
6   **end**
7   **if** *CM receives CH notification message* **then**
8   CH mode = broadcast ;
9   broadcast CHADS at[time,time+τ]
10   **end**
11 **end**

2.5.2 Mathematical calculation of control overhead in ECORA algorithm

In ECORA algorithm, the CH broadcast the CHADS messages only if it receives A CMHELLO messages that tagged with LEAVE. Therefore, the number of CHADS message broadcasted by the CH is proportional to the number of CMs that leave the cluster zone in specific period of time. If we assume that the vehicles are always in moving status, then any vehicle that enters any cluster zone should also leaves the cluster zone after a period of time. Algorithm 2.4 explains in pseudocode the method for computing the total number of CHADS messages. So, the number of CHADS messages is also proportional to the number of CM that enters the cluster zone within a period of time.

Algorithm 2.4 TOTAL NUMBER OF CHADS MESSAGES

```
Input  : total of CH ads = 0
for i=1 to Max number of CMs do
    for CID=1 to Max number of clusters do
        if CM location=CID location then
            total of CH ads =total of CH ads+1;
            else
                total of CH ads = total of CH ads
            end
        end
    end
end
Output: return total of CH ads
```

In ECORA, the total number of control overhead messages are calculated in the same way as in CORA algorithm except that the number of CHADS message is proportional to the number of vehicles located in any cluster within a period of time. In addition, we add the number of CH notification messages, which is also equal to the number of CMs located in any cluster within a period of time, as follows:

$$\begin{aligned} TotalAdsmesage_k = TotalCMHELLO_k + TotalCHAds_k \\ + TotalCHnotification_k \end{aligned} \tag{2.8}$$

2.5.3 CH broadcast time prediction

The vehicles move on the highways within a predefined maximum and minimum velocity limits. In general, we assume that most of the vehicles are moving in a constant speed and within the predefined range limits. In the highway scenario, the highway is divided into stationary cluster zones. The length of each cluster zone equals half of the transmission range of the standard vehicle. Therefore, at any time each CH is capable to communicate with neighbor CHs without communication failure. The communication failure might happen only in case when there no vehicles are located inside the cluster zone.

The main problem in clustered topology is that the CH is announcing itself periodically, and this yield to produce an excessive amount of clustered control overhead messages. Therefore, by proposing a CH broadcast time prediction mechanism, then the CHs are able to predict the period of time for arriving any new CMs, after that the CHs broadcast its status only within this period of time. Successful prediction of broadcasting time can significantly reduce the number of broadcasted CHADS messages. In highway scenario, most of the vehicles are moving respectively in consistent speeds. So, when any vehicle intend to leave the cluster zone or in other words arrive to its associated predefined threshold point, then it forwards a CMHELLO message (see Figure 2.7 for the contents of the CMHELLO message). In turn, the CH calculates the predicted time that the CM intend to leave its cluster zone. In addition, the CH forward to it's neighbors CHs a notification message of upcoming CMs and the prediction time of arriving to their cluster zones. We assumed here that the clock is synchronized between all vehicles in our clustered topology.

Any vehicle enters the cluster zone can calculate the threshold point. The threshold point is the safety point where the vehicle should leave the cluster zone without losing the communication with the CH. In other words, at this point the vehicle should inform the CH of leaving the current cluster by forwarding CMHELLO message. The time needed by each vehicle to forward the CMHELLO message is denoted by T . The threshold point is calculated based on Equation 2.9. This Equation shows and illustrates the calculations of the threshold distance for vehicle i in each cluster.

$$D_{ith} = V_i \times T \tag{2.9}$$

Where:

D_{ith} : Absolute distance of the vehicle i between the its threshold point and the cluster edge.

V_i : Velocity of vehicle i.

T: Time to forward the CMHELLO message to the CH.

The absolute distance of the vehicle i between the threshold point and the cluster edge is denoted by $|D_{ith}|$ and its velocity is given by V_i, respectively. Then the life time that the vehicle requires to leave its current cluster and enter the next cluster is predicted as

$$Leavetime_i = currenttime_i + \frac{|D_{ith}|}{V_i} \tag{2.10}$$

Each vehicle arriving at the predefined threshold point should forward CMHELLO message to the CH. The expected vehicle leaving time should be appended to this message. When the CH receives the message, it forwards a notification message to the neighbors CH informing of new CMs arriving. The notification message consist of CMID and the prediction time (calculated in Equation 2.9) of arriving to neighbor cluster zones.

In Figure 2.6, we present a sample numerical results of calculating the threshold distance for any vehicle by taking into consideration the vehicle velocity and the time for forwarding the CMHELLO message.

Velocities (Km\h)	T=0.1	T=0.2	T=0.3
	D_{ith}		
20	0.56	1.11	1.67
40	1.11	2.22	3.33
60	1.67	3.33	5
80	2.22	4.44	6.67
100	2.78	5.56	8.33

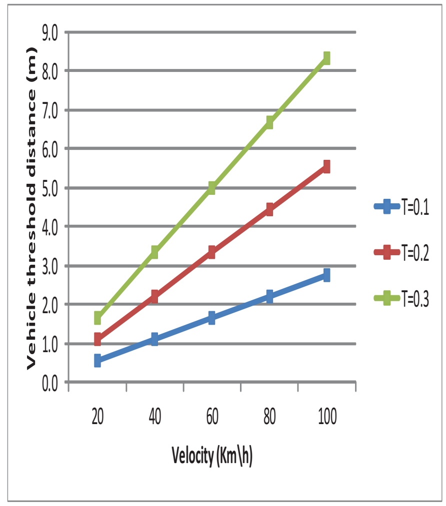

Figure 2.6 Threshold distance vs vehicle velocity and CMHELLO forwarding time

2.6 Designing of control overhead messages

In this section, we propose a new design for CHADS, CMHELLO, and CH notification messages. In the literature, many researchers assume different sizes of control overhead message. Mohammad et al. (Hadded *et al.* (2015)) assume that the messages generated by the CH consists of highway ID, CHID, direction, and specific weight value. In contrast, the CMHELLO messages are periodic broadcasting and it consists of CMID, highway ID, direction, position, and speed. Dan et al. (Lin *et al.* (2017)) propose a new CBR protocol that groups the vehicle moving in the same direction in one cluster, and the CMs send periodically a CMHELLO message that consists of CMID, location , speed and the next intersection ID.

In Figure 2.7 and 2.8, we present the contents of CMHELLO massage and CHADS message, respectively. The CMHELLO message consists of CMID and CMLT (the time that the current CM requires to arrive at the threshold point), and the CHADS message consists of CHID and CHLT. An important point we have to mention here is that in CORA algorithm the CHADS message broadcasted periodically (every τ second), while the CM forwards the CMHELLO messages in three cases: first, when the CM enters the cluster zone; second, when the CM leave the cluster zone; and third, when a new CH is elected. The new CH should be elected before the old CH arrives at the threshold point to avoid the overhead that occurs due to the reelection process. Therefore, we propose a new message format for the control overhead messages, the new messages mainly consists of two parameter only which are vehicle ID and vehicle LT.

ECORA algorithm designed another message format, which is CH notification message. In Figure 2.9, we present the format of CH notification message. This message consists of a CMID and the time he decided to leave the current cluster. As soon as any CH turns to the notification mode, it forwards the notification message to it's neighbor CHs. In ECORA, the CHADS message is broadcasted only when the CH changes to the broadcast mode, or in other word, when it receives the CH notification message.

CMHELLO message	
CMID	CMLT

Figure 2.7
CMHELLO message

CHADS message	
CHID	CHLT

Figure 2.8
CHADS message

CH notification message	
CMID	CMLT

Figure 2.9
CH notification message

2.7 Simulation, results, and analysis

By using the SUMO version 0.28.0 traffic generator (Krajzewicz *et al.* (2012)) and Matlab version R2016b, we calculate the number of CMHELLO messages based on Equation 2.6. In Table 2.1, we present the simulation parameter we used to evaluate the performance of our proposed work.

We first implemented a bidirectional highway scenario with length 10000 meters, then we divided the highway to fixed sizes of clusters of length 250 meter each. The vehicles enters the highway scenario in fixed rate which equals 1 vehicle/sec. When any vehicle arrives at edges of the highway, it makes a U turn and drives back in the opposite direction. The SUMO traffic generator keeps safety distance between the vehicles, and the distance distribution between the vehicle follow an exponential distribution. All the vehicles remain in the highway until the end of the simulation. The simulation starts to gather the results after all vehicle entering the highway scenario. To evaluate the performance of CORA protocol, we calculate

Table 2.1 Simulation parameters of CORA & ECORA algorithms

Parameter	**Value**
Simulation time	500 second
Number of cluster	40
Number of vehicles in each direction	200
Vehicles arrival rate	1 vehicle/sec
Communication range	250
Vehicle range speed	$(10-60)kmph$
Data sending rate	2 Mbps
CH protocol used	CBLTR

the number of CMHELLO, CHADS, and the total of the control overhead messages in each cluster. We assumed here that the vehicles use the same format of CMHELLO and CHADS message in terms of size. In Figure 2.10, we compare CORA algorithm with three other protocols mentioned in the literature; CBDRP, BRAVE, and MoZo protocols. In CBDRP protocol, the CMs in each cluster frequently forwards CMHELLO messages. In BRAVE protocol, the CMHELLO interval is 2 second. In MoZo protocol, the authors assume that the vehicles need to send CMHELLO updates messages when they deviate from their defined original moving function more than 5 m/s or the time from the last update which equals to 4 seconds.

In Table 2, we present a numerical results to validate the performance of the CORA protocol. Column 2 calculates the average number of CMHELLO messages generated within the simulation time by CORA, MoZo, BRAVE, and CBDRP protocols. Column 3 calculates the percentage number of HELLO messages generated by the CMs. The percentage is calculated by dividing the average number of CMHELLO messages that generated by any algorithm to the overall CMHELLO messages that generated by all algorithms. The CORA algorithm significantly reduces the CMHELLO messages, where it generates the minimum percentage of

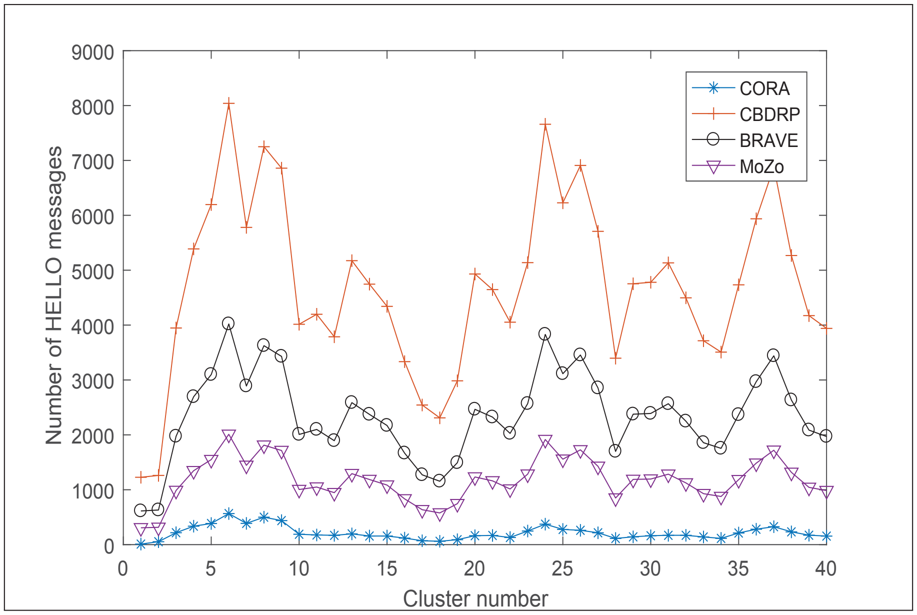

Figure 2.10 Number of CMHELLO message in highway scenario

CMHELLO messages, equal to 2.5 %, and the main reason is that to forward the CMHELLO messages, only in three instances that we explained in the previous section. In contrast, MoZo, BRAVE, and CBDRP algorithms, show high number of CMHELLO messages, and the reason of that is because all of these protocols forward periodically the CMHELLO messages.

Table 2.2 The mean and percentage of HELLO messages generated

Protocol Name	**Mean**	**Percentage of HELLO messages generated**
CORA	215.25	2.5%
MoZo	1215.8	13.9%
BRAVE	2431.6	27.8%
CBDRP	4863.3	55.8%

We evaluate the performance of CORA algorithm in terms of the total number of control overhead messages. Based in Equation 2.7, the total of number of control messages are the summation of all messages forwarded by the CMs and broadcasted by the CHs in a specific period of time. In Figure 2.11 , we present the total number of control overhead messages for the

CORA algorithm and another traditional CBR protocol(such as CBDRP). As shown in Figure 2.11a, in traditional CBR protocol all the vehicles in the clusters should forwards or broadcasts the control overhead messages periodically and depending mainly on time. Therefore, an excessive amount of generated control overhead messages are produced in a traditional CBR protocols. In contrast, Figure 2.11b shows that CORA algorithm achieves a significant reduction of CMHELLO messages, and the reason of that is because the CM forwards CMHELLO messages only in three instances; when the CMs enters or leaves the cluster zone and CH election process notification received. In other words, the CMs mainly depend on the location to forwards the CMHELLO message rather than the times.

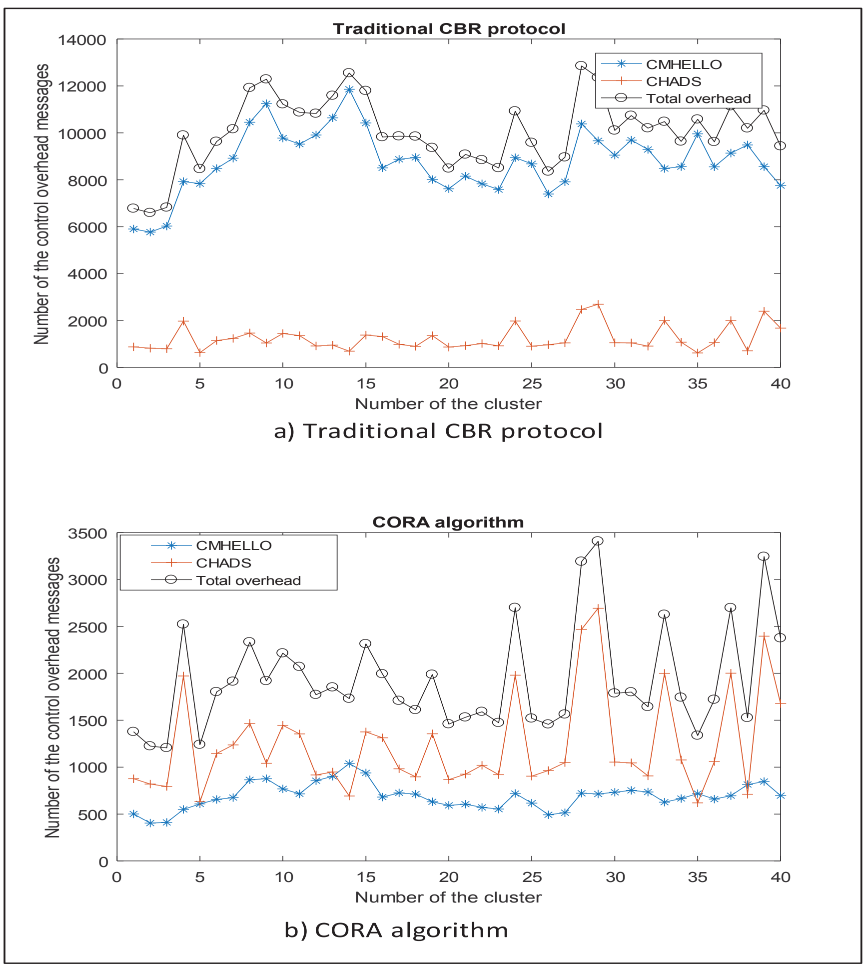

Figure 2.11 CORA vs traditional CBR protocol

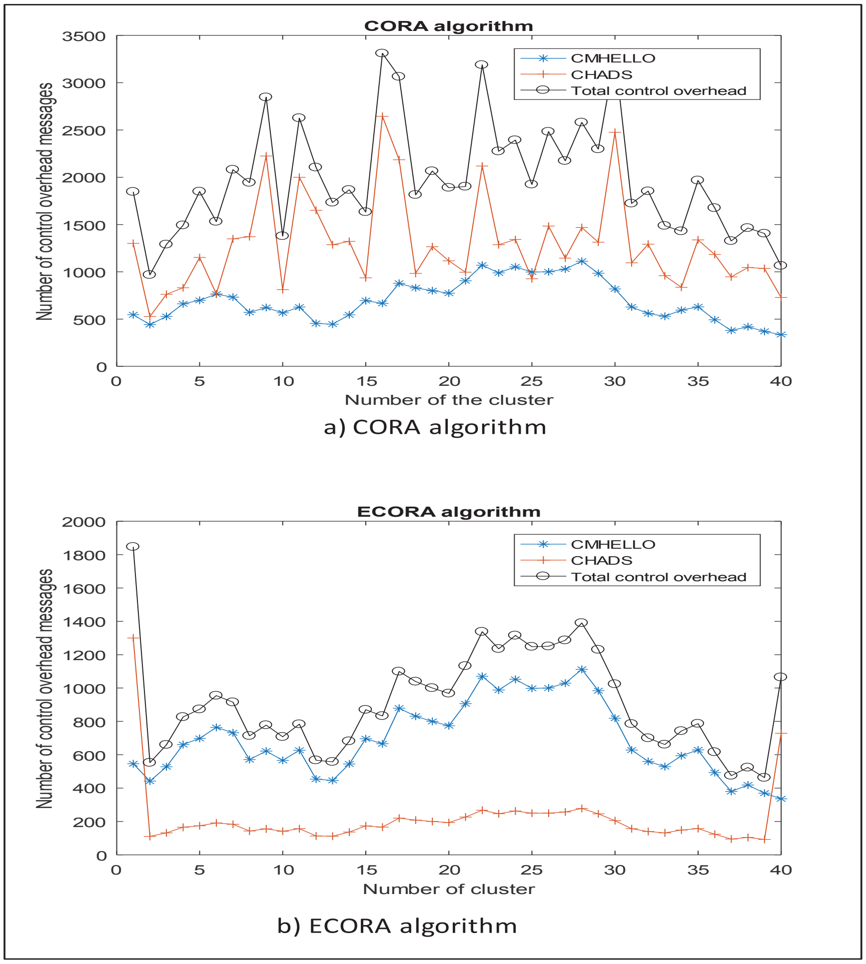

Figure 2.12 CORA vs ECORA algorithm

CORA outperforms all previous protocols in terms of the number of CMHELLO messages generated in each cluster within a period of time. The CORA protocol minimizes the number of CMHELLO messages to avoid periodically exchanging of CMHELLO message. CORA propagates the CMHELLO messages in three instances; which are: when the CM enters the cluster zone, second; when the CM leave the cluster zone, and third; when new CH announces itself. In general, CORA calculates the optimal number of CMHELLO messages in each cluster.

CORA algorithm mainly solves the problem of frequently forwarding the CMHELLO messages. On other hand, the CH still requires to broadcasts its status periodically for announcing the CH existence in the cluster. Also theses broadcasted CHADS messages negatively impact the network performance by reducing the available resources. Therefore, we also proposed another protocol, which is ECORA. ECORA is an enhanced version of CORA in which the CH is capable to predict a period of time to broadcast the CHADS instead of broadcasting within all the CHLT. In any clustered highway topology, the first and last clusters are called active clusters, since the CHs in theses two clusters should broadcast the CHADS periodically. However, other clusters called passive clusters. In these clusters the CH uses ECORA algorithm and broadcasts only when it changes its status to broadcast mode. In Figure 2.12, we compare our proposed two protocols in terms of the control overhead messages generated in a highway clustered topology. We can notice that when using ECORA, in active clusters, the number of CHADS messages are higher than that in passive clusters, and as we explained the CHs in these clusters should broadcasts the CHADS periodically. Based on simulation results, we can notice that ECORA achieves much more significant improvement in terms of the number of control overhead messages compared with CORA and traditional CBR protocols. In general, ECORA outperforms CORA and CBDRP protocols with reduction of the number of control overhead messages by about 65% and 93%, respectively.

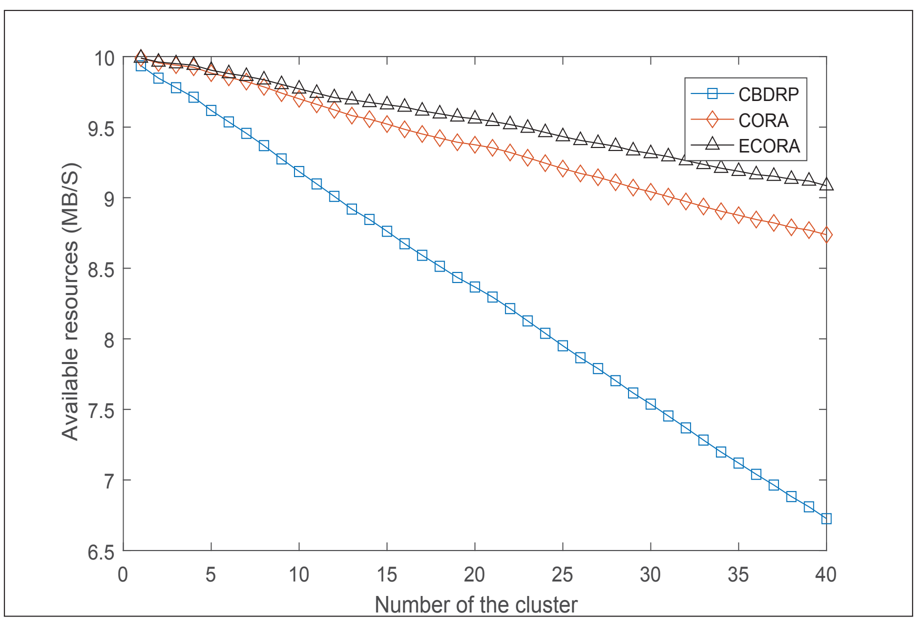

Figure 2.13 Available resources

A number of papers are evaluating the impact of clustered control overhead messages on the available network resources. In addition, all of these papers explain the contents of these messages without assigning a specific size for them. However, in this article we presented a new format for control overhead messages. The new format reduced the number of parameters to only two, which are the vehicle ID and its LT. On other hand, these articles assumed more parameters in addition to the vehicle ID, such as location, direction, speed, vehicle degree,...etc. Since there are no article mentioning the actual size of these messages, we will assumes that all of these messages have the same size, which is 10 Bytes. In Figure 2.13, we evaluate the impact of the control overhead messages on available resources of a clustered highway topology, by comparing our proposed protocols and CBDRP protocol in terms of available resources, we assume that the initial data rate is 10MB/s. The simulation results show that ECORA and CORA algorithms keep higher available resources compared with CBDRP protocol. The main reasons of that is because our proposed protocols reduce the number of generated control overhead messages compared with CBDRP protocol.

2.8 Conclusion

In this article, we proposed two algorithms: a Control Overhead Reduction Algorithm (CORA) and an Enhanced version of Control Overhead Reduction Algorithm (ECORA). These algorithms significantly reduce the number of control overhead messages generated by the CMs and the CHs in a clustered highway topology. We present a new mechanism for calculating the optimal period for forwarding or broadcasting the control overhead messages. The main contribution of these algorithms is to change the mechanism of forwarding the CM clustered control overhead messages in a clustered topology to be based on location instead of periodic of time. Also, we propose a CH prediction algorithm to enhance the CH to announce their status in specific predicted period of time instead of broadcasting within all CHLT. In addition, we present a new message format for the CMHELLO and CHADS messages, these messages consist of vehicle ID and vehicle LT. We also present a CH notification message to be forwarded to neighbor CHs when any CM intends to leave it's cluster zone. CORA propagate the CMHELLO messages in three scenarios: when the CM enters the cluster zone, second; when the CM leave the cluster zone, and when new CH elected. ECORA proposed A CH prediction algorithm to informs neighbor CHs of the prediction time of arriving new CMs to their clusters. Based in the simulation results, CORA and ECORA have significantly reduced the number of control overhead messages and maintained more resources in each cluster in a clustered highway topology.

CHAPTER 3

PASSIVE CH ELECTION AVOIDANCE AND CH ROUTING PROTOCOLS IN VANET

Ahmad Abuashour[1], Michel Kadoch[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper submitted for publication, October 2017

3.1 Abstract

In general, the high speed in Vehicular Ad-Hoc NETwork(VANET) yields to frequent links failure and reduces the network efficiency. Clustering technique is considered an important solution to improve the network stability and performance. However, the frequent CH election mainly increases the clustered control overhead messages, which yields to consume high amount of available network resources. High clustered control overhead messages is considered the main problem that negatively impacts the network performance. In this letter, we concentrate on the reduction of CH election control overhead messages. Therefore, we propose a new Passive CH election avoidance (PCHEA) protocol that aims to optimize the number of CH election process. In PCHEA protocol, each CH selects another CH based on specific information already stored in its memory, without requiring to trigger the election function. Also, we propose a CH Routing (CHR) protocol that aims to reduce the number of relayed CHs between any pair of vehicles. In CHR protocol, the CH selects the second adjacent CH (SACH) among all CHs located within its transmission range. The PCHEA and CHR protocols significantly reduce the number of CH elected and increase the average throughput in a bidirectional highway scenario, respectively.

3.2 Introduction

In general, Cluster-Based Routing (CBR) protocol (Abuashour & Kadoch (2017)) is a hybrid routing protocol, that divides the large network into small areas called clusters, and inside the cluster, there are many routing protocols called intra-cluster routing protocols. The communication between clusters is performed via pre-selected nodes called Cluster Heads (CHs). The CHs are responsible for coordinating the members of the cluster, and communicating betwen clusters using inter-cluster routing protocols (Song *et al.* (2010)).

The most important characteristic for any clustering technique is to create a stable cluster with minimum clustered control overhead messages. Clustered control overhead messages are mainly due to clustering maintenance and CH election process. In this letter, we assume the network is already stationary pre-clustered. Therefore, the control overhead messages generated by the clustering maintenance is eliminated. Furthermore, the process of election CHs still produces high control overhead messages. Any time, when the CH arrives to a predefined threshold point(the point to elect another CH), it should elect another CH. Generally, CH election process is a decentralized process in which each node within the cluster zone can participate to become the CH. Many algorithms in the literature considered one, two, or more parameters to elect the CH, such as: closer to the cluster center (Jerbi *et al.* (2009)), minimum velocity (Ramakrishnan (2011)), closer to the average velocity (Louazani *et al.* (2014)), and the CH Life Time (CHLT)(Abuashour & Kadoch (2017)). By selecting the CH based on maximum LT we can obtain a stable CHs with higher CH transmission time, thus reducing the number of elected CH in the network. Therefore, to improve the cluster stability we should elect a stable CH with maximum LT, and the CH remains as CH until it arrives at a predefined threshold point.

In this letter, we propose a novel Passive CH Election Avoidance (PCHEA) protocol that reduces the election control overhead messages produced by the frequently reelection process. The main contribution that makes PCHEA protocol different from previous proposed protocols is that it avoids triggering the CH election function when the CH intend to leave its cluster zone. Also, we propose a novel CH Routing (CHR) protocol. This protocol mainly aims to reduce the number of relayed CHs between any pair of vehicles, in order to improve the network efficiency and to reduce the end-to-end delay. The main contribution of this protocol is selecting the best CH located within its transmission range and achieving higher throughput compared to other CHs located within the forwarded CH transmission range.

3.3 Passive CH Election Avoidance (PCHEA) protocol

In general, each vehicle entering the cluster zone should be CM or CH at any time. Based on CORA algorithm (Abuashour & Kadoch (2017)), each vehicle entering any cluster coordination zone sets its status as CM by default. Then it should wait for τ second. If it does not receive any message from the CH, then it changes its status to CH and periodically (every τ second) forwards CH ADvertiSement(CHADS) message. If the CM receives the CHADS message, then it stays as CM and replies only with one HELLO message. The HELLO message consists of the CM Identification, and the remaining LT it requires to leave the cluster zone. We proposed An extinsion version of CORA (Abuashour & Kadoch (2017)). ECORA aims to reduce the broadcasted CH advertisement (CHADS) messages, by proposing a broadcast prediction algorithm that enables the CH to predict the period of time for broadcasting the CHADS messages. These two algorithms neglected the control overhead messages produced by the CH election process. Therefore, we propose PCHEA protocol to reduce the number of CH election processes.

Each elected CH should calculate specific threshold point in three steps:

a. Calculate the safety distance that the CH can elect another CH without losing the communication. This distance depends on predefined HandOver Time (HOT). The HOT is the summation of the re-election process time and the time to forward the current CH information to the new CH. The safety distance equal:

$$Safty(CH) = V_{CH} \times HOT \tag{3.1}$$

Where:

V_{CH} : CH Velocity.

HOT: Hand-Over Time.

b. Calculate the absolute distance from the CH to the predefined directional cluster edge. As follows:

$$Dist(CH) = CH_{loc} - CH_{edge} \tag{3.2}$$

Where:

CH_{loc} : Current CH location.

CH_{edge} : Predefined CH directional edge (Abuashour & Kadoch (2017)).

c. Calculate the CH threshold point. Which is equal to absolute value of subtraction step 2 and step 1. As follows:

$$Thpoint(CH) = |Dist(CH) - Safty(CH)| \tag{3.3}$$

Algorithm 3.1 explains in pseudo code the PCHEA protocol. As soon as the CH arrives to the predefined threshold point (as in line 5), it received all HELLO messages from the CMs, then it can then determine the next CH by selecting one of the CMs stored in memory. The selected CM should be the CM with the maximum LT (as in line 6). The following equation determines how the CH select the next CH:

$$NextCH = max(CM_1, CM_2, ..., CM_n) \tag{3.4}$$

Where:

CM_i: Cluster member with ID i.

The CH forwards a message to the CMs informing them of the next CH ID and its activation time (which is the time that the current CH requires to forward the cluster information to the new CH) which is equal to x (as in line 7 and 8). At this point, when the CM receives this message, it first compares with it's ID (as in line 12). If the CM ID is the same as the next CH ID, then it updates its status as CH at the received activation time which is equal to the $currenttime + x$ (as in line 13). Else the CM updates the CH ID and the activation time in its memory with the new CH received in the message (as in line 15).

Algorithm 3.1 PCHEA protocol

```
1  for t=anytime and t<=simulationtime do
2      if vehstat=CH then
3          CH calculate threshold distance;
4          CH stores CMs in CM table;
5          if CHloc= clusedge-threshdis then
6              Next CH= maxLT(CMs);
7              HOT= x;
8              CH forwards message <next CH, HOT>
9          end
10     end
11     else
12         if CM received CH message then
13             if CM ID= next CH ID then
14                 CM status= next CH ID at (t+HOT) else
15                     CHID= next CH ID at (t+HOT)
16                 end
17             end
18         end
19     end
20 end
```

3.4 CH Routing (CHR) protocol

In Figure 3.1, we explain all possible scenarios of CH routing for Adjacent CH (ACH) and Second Adjacent (SACH) as following: if we assume vehicle C is the CH of cluster n and it is the transmitted vehicle, then vehicle C can communicate with ACHs (vehicles B and D) without any probability of link failure. Also, vehicle C can communicate with the SCHs (vehicles A and E) but with some probability of link failure. However, there is a tradeoff in CH forwarding of ACH or SACH. On the positive side, relaying to the SACH leads to reduce the number of relayed CH between any pair of vehicles and thus increases the network throughput and minimizes the end-to-end delay. On the negative side, by relaying to the SACH, the probability of the communication failure increases significantly. Therefore, the CHR protocol defines some constraints to select the SACH by avoiding the probability communication failure.

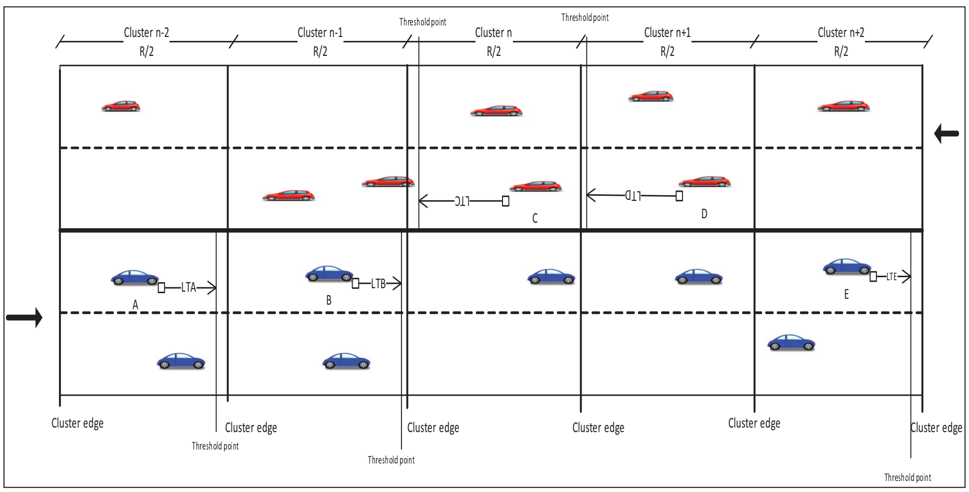

Figure 3.1 CH routing for ACH and SACH

For simplicity, we assume the data is transmitted in one direction from left to right. So, if the CH A wants to route the received data to CH E, it should first detect how many CHs are located within its transmission range. Here we have two possible choices; which are one CH (the ACH) or two CHs (ACH and SACH). If there is only one CH, then the data should mandatory transmit to it(CH B). Otherwise, the CH A should select the SACH (CH C). In Figure 3.2, the Flowchart explains the process of how CHR protocol works. The threshold time value is calculated based on the life time that the current CH predicts to remain in its cluster zone and is calculated as follows:

$$Th_t(i) = d_{ith}/(V_i) \tag{3.5}$$

Where;

$Th_t(i)$: Threshold time of CH i.

d_{ith}: Distance between the CH i and directional edge of the cluster.
V_i: Velocity of the CH i.

If the transmitted time is greater than the threshold time, then the CH compares it's LT with the SACH LT. If the SACH LT is greater than the transmitted CH LT, this means that the transmitted CH can select the SACH as next relay CH without any probability of communication failure. In other words, if the SACH LT is greater than or equal to the transmitted CH LT, then it can be selected as the next relay CH. Otherwise, the transmitted CH can transmit the received data in two parts; the first part transmitted within the absolute difference between the SACH LT and the transmitted CH LT, and in the second part it applies the CHR protocol again, and so on.

Mathematically, the probability of link failure for the SACH (p) can be presented as follows:

$$p = \begin{cases} 1 & \text{,if} \quad d_{CH,SACH} > R \\ \frac{LT_{CH}}{MLT_{CH}} & \text{,if} \quad d_{CH,SACH} \leq R \,\&\, T \leq Th_t \\ \frac{\sum_{i=1}^{T} LT_{CH}}{\sum_{i=1}^{T} MLT_{CH}} & \text{,if} \quad d_{CH,SACH} \leq R \,\&\, T > Th_t \end{cases} \tag{3.6}$$

Where;
$d_{CH,SACH}$: The absolute distance between the CH and the SACH.
R: CH transmission range.
T: Transmission time.
LT_{CH}: current CH LT.
MLT_{CH}: Maximum CH LT.

3.5 Simulation, results, and analysis

By using the SUMO version 0.28.0 traffic generator and Matlab version R2016b, we evaluate the performance of our proposed protocols in different scenarios and in terms of different performance metrics, as follows:

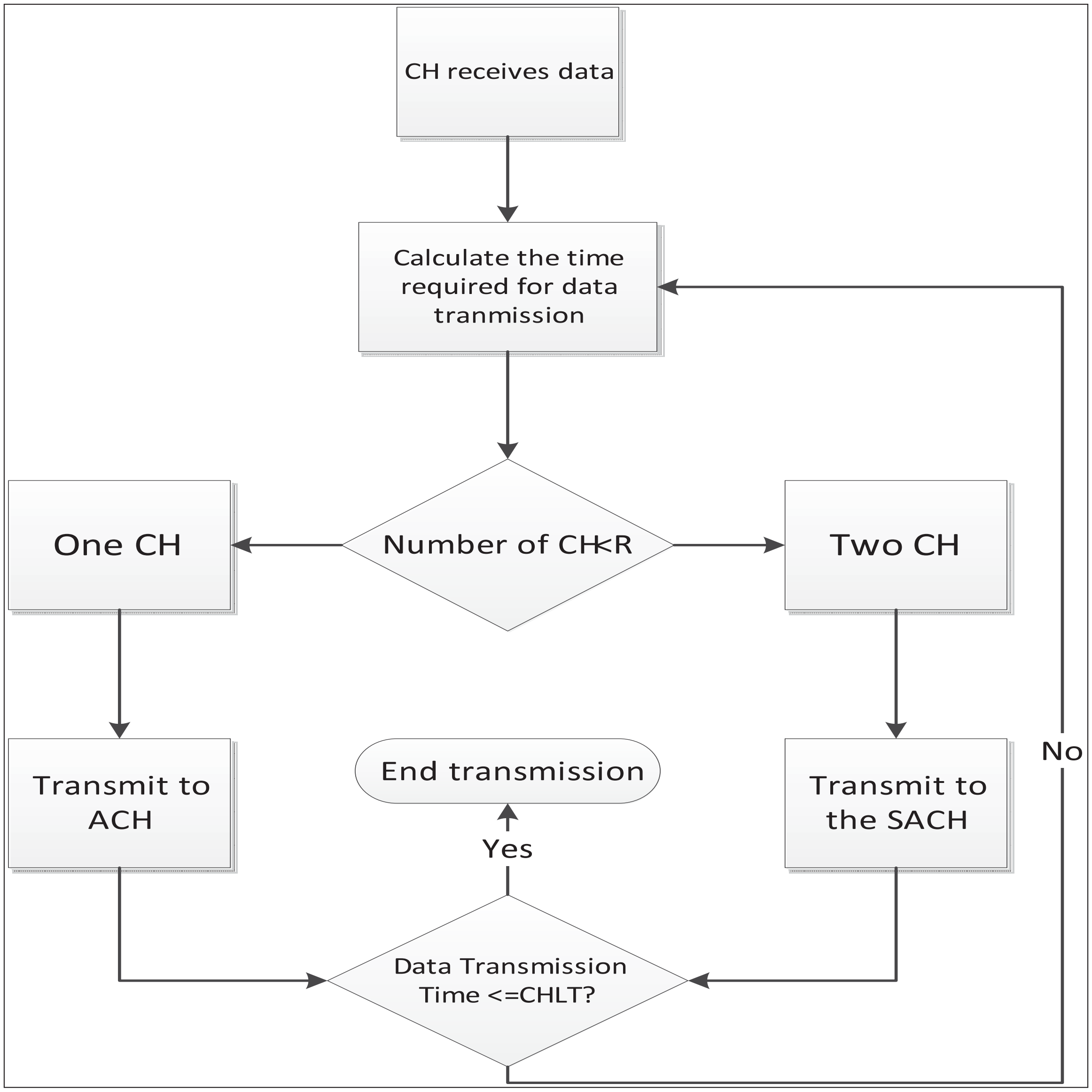

Figure 3.2 Flowchart of CHR protocol

To evaluate the performance of PCHEA protocol, we compare PCHEA protocol with CBLTR protocol (Abuashour & Kadoch (2016)) in terms of the average number of CH elected. We select the CBLTR protocol because it elects the CH based on LT and it outperforms other protocols that consider other parameters in terms of reducing the number of CH election processes. Therefore, we implemented a bidirectional highway scenario with length of 10000 meters, after that we divided the highway into fixed sizes of clusters of length 250 meter each. The vehicles enters the highway scenario in fixed rate which equals 1 vehicle/sec. The simulation starts to gather the results after all vehicle enter the Highway scenario.

we applied the CBLTR protocol and the PCHEA protocol on the bidirectional highway topology, respectively. The results in Figure 3.3A show that, based on the CBLTR protocol, the number of elected CH increases when the average velocity increases, and this increase is due to triggering the CH election process every time the CH arrives at the threshold point. In addition, Figure 3.3B shows that, based on the PCHEA protocol the number of CH elected is lightly increased as the velocity increases. This proposed protocol significantly reduces the number of elected CH compared to CBLTR protocol because it avoids the CH election process at the threshold point. In PCHEA protocol, the election of the CH occurs only at the beginning of the simulation or when any vehicle enters an empty cluster. In addition, when any CH arrives at the threshold point, it only informs other CMs about the next CH without requiring to make new CH election. In Figure 3.3C, we present a comparison between CBLTR protocol and PCHEA protocol in terms of the average number of elected CH. We clearly notice that the PCHEA protocol significantly outperforms CBLTR protocol in terms of reducing the number of CH election process.

In Figure 3.4, we evaluate the impact of the control overhead messages on available resources of a clustered highway topology, by comparing the PCHEA protocol and ECORA protocol in terms of available resources. We assume that the initial data rate is 10MB/s. The simulation results show that PCHEA protocol keeps higher available resources compared to ECORA protocol. The main reasons of that is because the PCHEA protocol reduces the number of CH elected in each cluster and reduces the CH election control overhead messages generated due to frequent election processes. In general, PCHEA protocol outperforms ECORA algorithm by reduction of the number of control overhead messages by about 90%.

To evaluate the performance of CHR protocol, we compare CHR protocol with CBLTR protocol (Abuashour & Kadoch (2016)) in terms of the average throughput. In CBLTR protocol, when any CH receives a packet, it forwards the packet to the next CH that are located towards

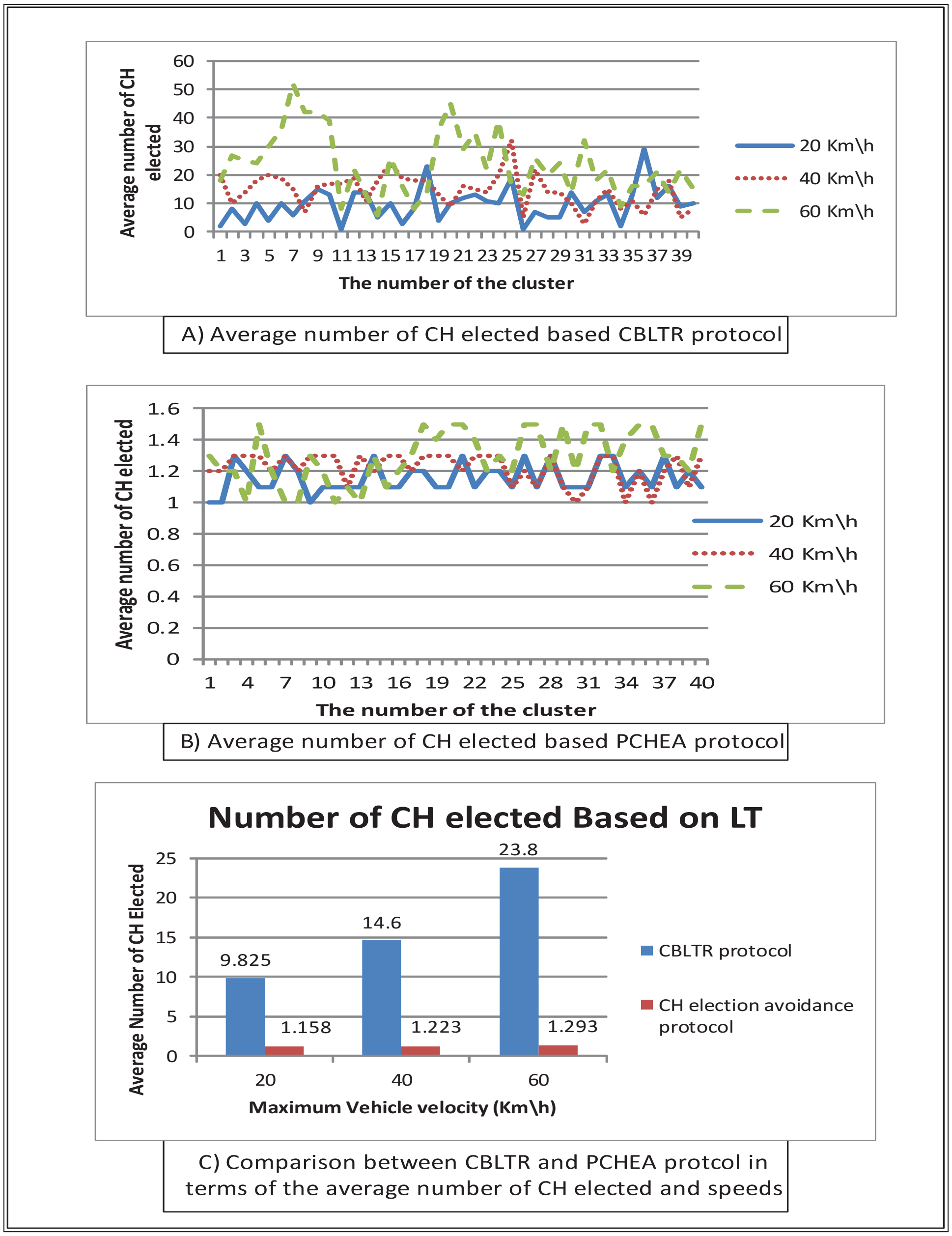

Figure 3.3 comparison between PCHEA and CBLTR protocols

the destination regardless of its moving direction. In addition, it always forwards to ACHs. In contrast, CHR protocol proposed a novel protocol that enables the CH to forward the packet to the SACH if it existed within its transmission range. This contribution significantly reduces the number of relayed CH between any pair of CHs. In addition it also increases the throughput and reduces the end-to-end delay. In Figure 3.5, we compare CHR protocol with CBLTR pro-

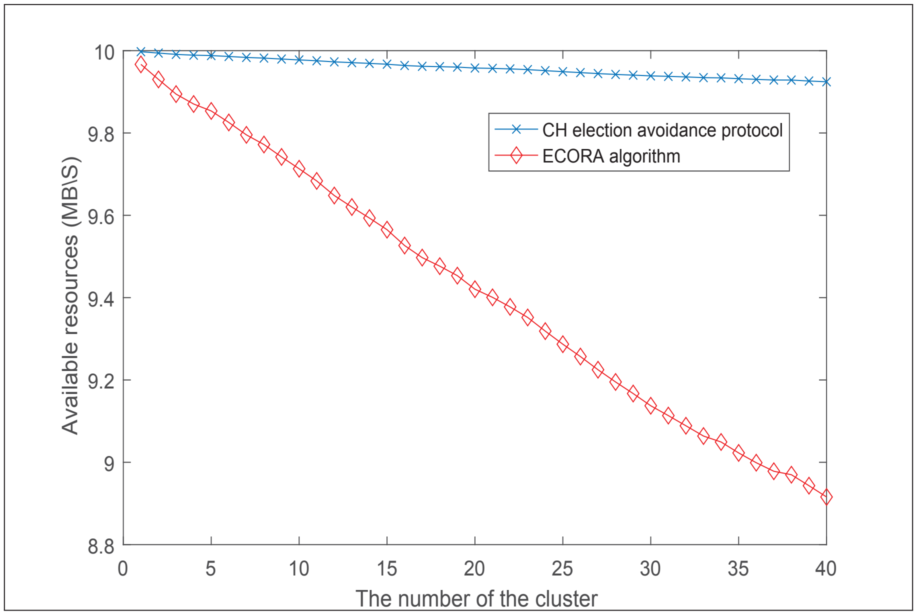

Figure 3.4 comparison between PCHEA and ECORA protocols in terms of resources availability in highway scenario

tocol and another protocol that elects the CH based on minimum velocity (Song *et al.* (2010)). The results show that the CHR protocol significantly outperforms other two protocols in terms of average throughput.

3.6 Conclusion

In this letter we proposed two main protocols; first, the Passive CH Election Avoidance (PCHEA) protocol. Second, the CH Routing (CHR) protocol. The PCHEA aims to increase the network performance by reducing the number of CH election processes. The CHR protocol aims to reduce the number of relayed CHs between any pair of vehicles. As we present in the simulation results, the PCHEA protocol significantly outperforms CBLTR protocol in terms reducing the number of CH election process. Also, the CHR protocol significantly outperforms the CBLTR protocol in terms of average throughput.

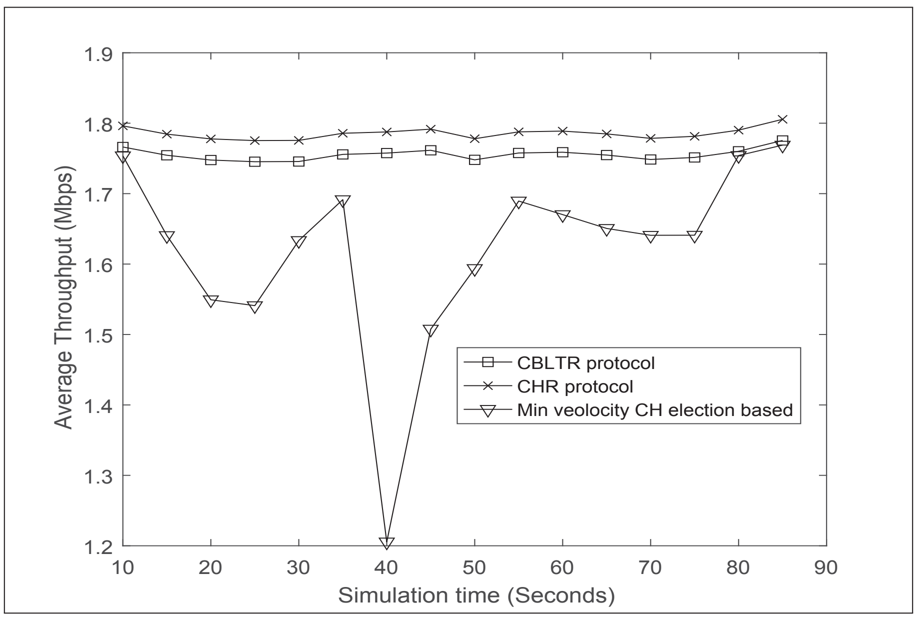

Figure 3.5 Comparison between CHR, CBLTR, and CH min velocity election based protocols in terms of average throughput

CHAPTER 4

CONCLUSION, FUTURE WORKS, AND LIST OF PUBLICATIONS

4.1 Conclusion

In VANET, the main drawback is the network instability, which yields to reduce the network efficiency. Clustering techniques is mainly used to create a stable clustered topology with minimum clustered control overhead messages. This thesis mainly concentrate to achieve this objective. As explained in Chapter 1, two main algorithms are proposed to improve the performance in a clustered highway and grid VANET scenarios. The CHs are elected based on maximum LT, and the re-election process is required only when the CHs reach their corresponding threshold point. Based on the simulation results, CBLTR protocol shows a significant improvement in terms of average throughput. The enhancement in CBLTR protocol is a new mechanism to select new CHs. The selected CHs have longer LT span making the protocol more stable. Also, an Intersection Dynamic VANET Routing (IDVR) protocol in a grid topology is proposed. The IDVR protocol selects the optimal route based on its current location, destination location, and a maximum of the minimum average throughput for SCSRs. IDVR increases the overall network efficiency, by increasing the route throughput, and decreasing end-to-end delay.

In Chapter 2, clustering techniques have been used as the main solution to reduce the control overhead messages in VANET, in which the network is divided into multiple stationary clusters and selecting one of the Cluster Members (CMs) as a Cluster Head (CH). Still, a problem occurs when the control overhead messages increase due to periodically forwarding of CM HELLO (CMHELLO) messages between the CMs and the CH, and when the CH periodically broadcasts an CH ADvertiSement (CHADS) messages to declare itself to the CMs. Hence, minimizing control overhead messages in any cluster environment is an essential goal to use the network resources efficiently. Therefore, we proposed two algorithms: a Control Overhead Reduction Algorithm (CORA) and an Enhanced version of Control Overhead Reduction

Algorithm (ECORA). These algorithms significantly reduce the number of control overhead messages generated by the CMs and the CHs in a clustered highway topology. We present a new mechanism for calculating the optimal period for forwarding or broadcasting the control overhead messages. The main contribution of these algorithm is to change the mechanism of forwarding the CM clustered control overhead messages in a clustered topology to be based on location instead of periodic of time. Based in the simulation results, CORA and ECORA have significantly reduced the number of control overhead messages and maintained more resources in each cluster in a clustered highway topology.

Chapter 3 addresses the problem of control overhead messages that generated due to frequent CH election. This generated control overhead messages yields to consume high amount of available network resources, which at the end negatively impacts on the overall clustered VANET performance. Therefore, a Passive CH Election Avoidance (PCHEA) protocol is proposed that aims to reduce the number of elected CH in Clustered highway VANET scenario. In addition, in this Chapter also a CH routing protocol is proposed to reduce the number of relayed CHs between any pair of vehicles. As we present in the simulation results, the PCHEA protocol significantly outperforms CBLTR protocol in terms reducing the number of CH election process. Also, the CHR protocol significantly outperforms the CBLTR protocol in terms of average throughput.

4.2 Future works

As a future plan, We suggest more research issues to be conducted on clustered VANET routing protocol, and the following should be considered:

a. Considering other mobility parameter, such as acceleration rate. Evaluate this parameter in terms of selection a stable CH;

b. Evaluate the performance of CBR protocols in a dynamic clustered topology and compare the results with stationary clustered topology;

c. Evaluate the performance of our proposed protocols on a real-map topology (real experimental test). Furthermore, validate out simulation results with real experimental results.

4.3 List of publications

Below is the list of publications delivered from the work related to this thesis: Journals (Published and submitted):

a. A. Abuashour; M. Kadoch, "Performance Improvement of Cluster-Based Routing Protocol in VANET," in IEEE Access , vol.PP, no.99, pp.1-1 doi: 10.1109/ACCESS.2017.2733380 ACCEPTED;

b. A. Abuashour; M. Kadoch, "Control Overhead Reduction In Cluster-Based VANET Highway Topology" SUBMITTED to Electronics in September 25 2017;

c. A. Abuashour; M. Kadoch, "Passive CH Election Avoidance Protocol In VANET" SUBMITTED to IEEE wireless communication letters.

Conferences (Published):

a. A. Abuashour and M. Kadoch, "A Cluster-Based Life-Time Routing Protocol in VANET," 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, 2016, pp. 213-219; doi: 10.1109/FiCloud.2016.38

b. A. Abuashour and M. Kadoch, " Intersection dynamic VANET routing (IDVR) protocol " 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic. August 21-23,2017;

c. A. Abuashour and M. Kadoch,"Control Overhead Reduction In Cluster-Based VANET Routing Protocol". 9th EAI International Conference on Ad Hoc Networks. Niagra Falls, Canada. September 28–29, 2017.

BIBLIOGRAPHY

(2010). Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. *Ieee std 802.11p-2010 (amendment to ieee std 802.11-2007 as amended by ieee std 802.11k-2008, ieee std 802.11r-2008, ieee std 802.11y-2008, ieee std 802.11n-2009, and ieee std 802.11w-2009)*, 1-51. doi: 10.1109/IEEESTD.2010.5514475.

Abedi, O., Berangi, R. & Azgomi, M. A. (2009, June). Improving route stability and overhead on aodv;routing protocol and make it usable for vanet. *2009 29th ieee international conference on distributed computing systems workshops*, pp. 464-467. doi: 10.1109/ICDCSW.2009.88.

Abuashour, A. & Kadoch, M. (2016, Aug). A cluster-based life-time routing protocol in vanet. *2016 ieee 4th international conference on future internet of things and cloud (ficloud)*, pp. 213-219. doi: 10.1109/FiCloud.2016.38.

Abuashour, A. & Kadoch, M. (2017). Performance improvement of cluster-based routing protocol in vanet. *Ieee access*, PP(99), 1-1. doi: 10.1109/ACCESS.2017.2733380.

AlMheiri, S. M. & AlQamzi, H. S. (2015, Feb). Manets and vanets clustering algorithms: A survey. *2015 ieee 8th gcc conference exhibition*, pp. 1-6. doi: 10.1109/IEEEGCC.2015.7060048.

Camp, T., Boleng, J. & Wilcox, L. (2002). Location information services in mobile ad hoc networks. *2002 ieee international conference on communications. conference proceedings. icc 2002 (cat. no.02ch37333)*, 5, 3318-3324 vol.5. doi: 10.1109/ICC.2002.997446.

Chai, R., Yang, B., Li, L., Sun, X. & Chen, Q. (2013, Oct). Clustering-based data transmission algorithms for vanet. *2013 international conference on wireless communications and signal processing*, pp. 1-6. doi: 10.1109/WCSP.2013.6677258.

Ding, B., Chen, Z., Wang, Y. & Yu, H. (2011, Nov). An improved aodv routing protocol for vanets. *2011 international conference on wireless communications and signal processing (wcsp)*, pp. 1-5. doi: 10.1109/WCSP.2011.6096736.

Garbiso, J., Diaconescu, A., Coupechoux, M. & Leroy, B. (2016, Nov). Dynamic cluster size optimization in hybrid cellular-vehicular networks. *2016 ieee 19th international conference on intelligent transportation systems (itsc)*, pp. 557-563. doi: 10.1109/ITSC.2016.7795609.

Hadded, M., Zagrouba, R., Laouiti, A., Muhlethaler, P. & Saidane, L. A. (2015, May). A multi-objective genetic algorithm-based adaptive weighted clustering protocol in vanet. *2015 ieee congress on evolutionary computation (cec)*, pp. 994-1002. doi: 10.1109/CEC.2015.7256998.

Jerbi, M., Senouci, S. M., Rasheed, T. & Ghamri-Doudane, Y. (2009). Towards efficient geographic routing in urban vehicular networks. *Ieee transactions on vehicular technology*, 58(9), 5048-5059. doi: 10.1109/TVT.2009.2024341.

Khakpour, S., Pazzi, R. W. & El-Khatib, K. (2017). Using clustering for target tracking in vehicular ad hoc networks. *Vehicular communications*, 9, 83 - 96. doi: https://doi.org/10.1016/j.vehcom.2017.02.002.

Krajzewicz, D., Erdmann, J., Behrisch, M. & Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International journal on advances in systems and measurements*, 5(3&4), 128–138.

Ku, I., Lu, Y., Gerla, M., Gomes, R. L., Ongaro, F. & Cerqueira, E. (2014, June). Towards software-defined vanet: Architecture and services. *2014 13th annual mediterranean ad hoc networking workshop (med-hoc-net)*, pp. 103-110. doi: 10.1109/Med-HocNet.2014.6849111.

Lee, K. C., Haerri, J., Lee, U. & Gerla, M. (2007, Nov). Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios. *2007 ieee globecom workshops*, pp. 1-10. doi: 10.1109/GLOCOMW.2007.4437832.

Lin, D., Kang, J., Squicciarini, A., Wu, Y., Gurung, S. & Tonguz, O. (2017). Mozo: A moving zone based routing protocol using pure v2v communication in vanets. *Ieee transactions on mobile computing*, 16(5), 1357-1370. doi: 10.1109/TMC.2016.2592915.

Liu, L., Wang, Z. & Jehng, W. K. (2008, Aug). A geographic source routing protocol for traffic sensing in urban environment. *2008 ieee international conference on automation science and engineering*, pp. 347-352. doi: 10.1109/COASE.2008.4626520.

Lo, S.-C., Lin, Y.-J. & Gao, J.-S. (2013). A multi-head clustering algorithm in vehicular ad hoc networks. *International journal of computer theory and engineering*, 5(2), 242.

Louazani, A., Senouci, S. M. & Bendaoud, M. A. (2014, June). Clustering-based algorithm for connectivity maintenance in vehicular ad-hoc networks. *2014 14th international conference on innovations for community services (i4cs)*, pp. 34-38. doi: 10.1109/I4CS.2014.6860550.

Luo, D. & Zhou, J. (2011, Sept). An improved hybrid location-based routing protocol for ad hoc networks. *2011 7th international conference on wireless communications, networking and mobile computing*, pp. 1-4. doi: 10.1109/wicom.2011.6040306.

Luo, Y., Zhang, W. & Hu, Y. (2010, April). A new cluster based routing protocol for vanet. *2010 second international conference on networks security, wireless communications and trusted computing*, 1, 176-180. doi: 10.1109/NSWCTC.2010.48.

Maslekar, N., Boussedjra, M., Mouzna, J. & Labiod, H. (2011, July). A stable clustering algorithm for efficiency applications in vanets. *2011 7th international wireless* *communications and mobile computing conference*, pp. 1188-1193. doi: 10.1109/I-WCMC.2011.5982709.

Mohammad, S. A. & Michele, C. W. (2010, Oct). Using traffic flow for cluster formation in vehicular ad-hoc networks. *Ieee local computer network conference*, pp. 631-636. doi: 10.1109/LCN.2010.5735785.

Mohammed Nasr, M. M., Abdelgader, A. M. S., Wang, Z.-G. & Shen, L.-F. (2016). Vanet clustering based routing protocol suitable for deserts. *Sensors*, 16(4).

Morales, M. M. C., Hong, C. S. & Bang, Y. C. (2011, Sept). An adaptable mobility-aware clustering algorithm in vehicular networks. *2011 13th asia-pacific network operations and management symposium*, pp. 1-6. doi: 10.1109/APNOMS.2011.6077004.

Oubbati, O. S., Lagraa, N., Lakas, A. & Yagoubi, M. B. (2014, March). Irtiv: Intelligent routing protocol using real time traffic information in urban vehicular environment. *2014 6th international conference on new technologies, mobility and security (ntms)*, pp. 1-4. doi: 10.1109/NTMS.2014.6814028.

Oubbati, O. S., Lakas, A., Lagraa, N. & Yagoubi, M. B. (2016, April). Uvar: An intersection uav-assisted vanet routing protocol. *2016 ieee wireless communications and networking conference*, pp. 1-6. doi: 10.1109/WCNC.2016.7564747.

Paul, A. & G, P. K. (2013, Dec). A cluster based leader election algorithm for manets. *2013 international conference on control communication and computing (iccc)*, pp. 496-499. doi: 10.1109/ICCC.2013.6731705.

Ramakrishnan, B. (2011). Cbvanet: A cluster based vehicular adhoc network model for simple highway communication.

Rawashdeh, Z. Y. & Mahmud, S. M. (2012). A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. *Eurasip journal on wireless communications and networking*, 2012(1), 15. doi: 10.1186/1687-1499-2012-15.

Rossi, G. V., Fan, Z., Chin, W. H. & Leung, K. K. (2017, March). Stable clustering for ad-hoc vehicle networking. *2017 ieee wireless communications and networking conference (wcnc)*, pp. 1-6. doi: 10.1109/WCNC.2017.7925786.

Ruiz, P. M., Cabrera, V., Martinez, J. A. & Ros, F. J. (2010, Nov). Brave: Beacon-less routing algorithm for vehicular environments. *The 7th ieee international conference on mobile ad-hoc and sensor systems (ieee mass 2010)*, pp. 709-714. doi: 10.1109/MASS.2010.5663798.

Saleet, H., Langar, R., Naik, K., Boutaba, R., Nayak, A. & Goel, N. (2011). Intersection-based geographical routing protocol for vanets: A proposal and analysis. *Ieee transactions on vehicular technology*, 60(9), 4560-4574. doi: 10.1109/TVT.2011.2173510.

Singh, S., Rajpal, N. & Sharma, A. (2014). Address allocation for manet merge and partition using cluster based routing. *Springerplus*, 3(1), 605. doi: 10.1186/2193-1801-3-605.

Song, T., Xia, W., Song, T. & Shen, L. (2010, Nov). A cluster-based directional routing protocol in vanet. *2010 ieee 12th international conference on communication technology*, pp. 1172-1175. doi: 10.1109/ICCT.2010.5689132.

Spaho, E., Ikeda, M., Barolli, L., Xhafa, F., Younas, M. & Takizawa, M. (2012, Nov). Performance of olsr and dsdv protocols in a vanet scenario: Evaluation using cavenet and ns3. *2012 seventh international conference on broadband, wireless computing, communication and applications*, pp. 108-113. doi: 10.1109/BWCCA.2012.28.

Tayal, S. & Triphathi, M. R. (2012, Jan). Vanet-challenges in selection of vehicular mobility model. *2012 second international conference on advanced computing communication technologies*, pp. 231-235. doi: 10.1109/ACCT.2012.119.

Ucar, S., Ergen, S. C. & Ozkasap, O. (2013, April). Vmasc: Vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks. *2013 ieee wireless communications and networking conference (wcnc)*, pp. 2381-2386. doi: 10.1109/WCNC.2013.6554933.

Wu, C., Ohzahata, S. & Kato, T. (2013, Sept). Can we generate efficient routes by using only beacons? backbone routing in vanets. *2013 ieee 24th annual international symposium on personal, indoor, and mobile radio communications (pimrc)*, pp. 2929-2934. doi: 10.1109/PIMRC.2013.6666648.

Yu, X., Guo, H. & Wong, W. C. (2011, July). A reliable routing protocol for vanet communications. *2011 7th international wireless communications and mobile computing conference*, pp. 1748-1753. doi: 10.1109/IWCMC.2011.5982800.

Zhao, C., Li, C., Zhu, L., Lin, H. & Li, J. (2012, Oct). A vehicle density and load aware routing protocol for vanets in city scenarios. *2012 international conference on wireless communications and signal processing (wcsp)*, pp. 1-6. doi: 10.1109/WCSP.2012.6542825.