UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

STABLE DYNAMIC FEEDBACK-BASED PREDICTIVE CLUSTERING PROTOCOL FOR VEHICULAR AD HOC NETWORKS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

MOHAMMAD MUKHTARUZZAMAN
Norman, Oklahoma
2023

STABLE DYNAMIC FEEDBACK-BASED PREDICTIVE CLUSTERING PROTOCOL FOR VEHICULAR AD HOC NETWORKS

A DISSERTATION APPROVED FOR THE

SCHOOL OF COMPUTER SCIENCE

BY THE COMMITTEE CONSISTING OF

Dr. Mohammed Atiquzzaman, Chair

Dr. Dean Hougen

Dr. Changwook Kim

Dr. Ronald Barnes

Dedication

This dissertation is dedicated to my parents:

Abdus Salam Mridha

&

Monoara Begum

Acknowledgments

I want to acknowledge the contributions of a few individuals. I wish to express my enormous gratitude to my advisor, Dr. Mohammed Atiquzzaman, for his expert advice, continuous supervision throughout my PhD study and providing an excellent research atmosphere.

I greatly appreciate the guidance and time given by my committee members: Dr. Dean Hougen, Dr. Changwook Kim and Dr. Ronald Barnes. Special thanks to Dr. Rafal Jabrzemski for his continuous support.

I am thankful to my parents; brothers and sisters and their families. They have always been supporting me and encouraging me with their best wishes.

Table of Contents

List of Tables

List of Figures

Abstract

Scalability presents a significant challenge in vehicular communication, particularly when there is no hierarchical structure in place to manage the increasing number of vehicles. As the number of vehicles increases, they may encounter the broadcast storm problem, which can cause network congestion and reduce communication efficiency. Clustering can solve these issues, but due to high vehicle mobility, clustering in vehicular ad hoc networks (VANET) suffers from stability issues. Existing clustering algorithms are optimized for either cluster head or member, and for highways or intersections. The lack of intelligent use of mobility parameters like velocity, acceleration, direction, position, distance, degree of vehicles, and movement at intersections, also contributes to cluster stability problems. A dynamic clustering algorithm that efficiently utilizes all mobility parameters can resolve these issues in VANETs.

To provide higher stability in VANET clustering, a novel robust and dynamic mobility-based clustering algorithm called junction-based clustering protocol for VANET (JCV) is proposed in this dissertation. Unlike previous studies, JCV takes into account position, distance, movement at the junction, degree of a vehicle, and time spent on the road to select the cluster head (CH). JCV considers transmission range, the moving direction of the vehicle at the next junction, and vehicle density in the creation of a cluster. JCV's performance is compared with two existing VANET clustering protocols in terms of the average cluster head duration, the average cluster member (CM) duration, the average number of cluster head changes, and the percentage of vehicles

participating in the clustering process, etc. To evaluate the performance of JCV, we developed a new cloud-based VANET simulator (CVANETSIM). The simulation results show that JCV outperforms the existing algorithms and achieves better stability in terms of the average CH duration (4%), the average CM duration (8%), the number of CM (6%), the ratio of CM (22%), the average CH change rate (14%), the number of CH (10%), the number of non-cluster vehicles (7%), and clustering overhead (35%).

The dissertation also introduced a stable dynamic feedback-based predictive clustering (SDPC) protocol for VANET, which ensures cluster stability in both highway and intersection scenarios, irrespective of the road topology. SDPC considers vehicle relative velocity, acceleration, position, distance, transmission range, moving direction at the intersection, and vehicle density to create a cluster. The cluster head is selected based on the future construction of the road, considering relative distance, movement at the intersection, degree of vehicles, majority-vehicle, and probable cluster head duration. The performance of SDPC is compared with four existing VANET clustering algorithms in various road topologies, in terms of the average cluster head change rate, duration of the cluster head, duration of the cluster member, and the clustering overhead. The simulation results show that SDPC outperforms existing algorithms, achieving better clustering stability in terms of the average CH change rate (50%), the average CH duration (15%), the average CM duration (6%), and the clustering overhead (35%).

Chapter 1

Introduction

We are all aware of the high number of crashes and vehicle pileups that occur on roads each year, resulting in the loss of more than a million lives annually, as well as numerous injuries that can lead to long-term disabilities. The cost of medical care is also significant, and the suffering of the victims' families cannot be measured.

Enabling vehicles to communicate with each other presents a solution to reduce the high number of accidents on roads. Through vehicular ad hoc networks (VANET), vehicles can transmit crucial information, which can be instantly received and acted upon by other vehicles to prevent accidents. However, the challenge lies in scalability, as the transmission of messages can become overwhelming when hundreds of vehicles communicate with each other, resulting in the loss of critical messages during crucial moments. Clustering offers a viable solution to this challenge.

Vehicular clustering involves creating a group of vehicles that communicate only with other vehicles in their cluster, ultimately reducing the number of messages that need to be transmitted simultaneously. However, the challenge in clustering is stability since clusters often break down due to the high speed of vehicles, continuous changes in position, varying acceleration, and different moving directions, among other factors. Current vehicular clustering protocols can provide only partial stability, making it necessary to address this issue to enhance communication among vehicles.

In this dissertation, we proposed two clustering protocols to provide cluster stability in VANET. We also presented a cloud-based VANET simulator to simulate clustering protocols in VANET. Stable clustering protocols can enable smooth communication, ultimately reducing the number of road crashes and saving millions of lives.

1.1 Background

Vehicular communication has become a rapidly growing area of research, particularly in the realm of intelligent transportation systems (ITS). IEEE 802.11P and IEEE 1609 have defined wireless access in vehicular environments (WAVE) using dedicated short-range communications (DSRC) frequency bands for wireless communication in VANET. DSRC/WAVE is used to fulfill the low latency requirement for safety and control messages in vehicle-to-vehicle (V2V) communication, while long-term evolution (LTE) is employed for vehicle-to-infrastructure (V2I) communication.

An on-board unit (OBU) is installed in each vehicle to communicate with the roadside unit (RSU) and other vehicles. The RSU is a fixed infrastructure like the base station of a cellular network. Global positioning system (GPS) is used to access vehicle location. When vehicles communicate with each other, it is referred to as V2V communication. Conversely, when a vehicle communicates with an RSU, it is called V2I communication. Together, V2V and V2I communications are known as vehicle-to-everything (V2X) communications. For the purposes of this dissertation, the terms vehicle, car, and node are used interchangeably to refer to a vehicle. Figure 1.1 depicts a vehicular communication scenario.

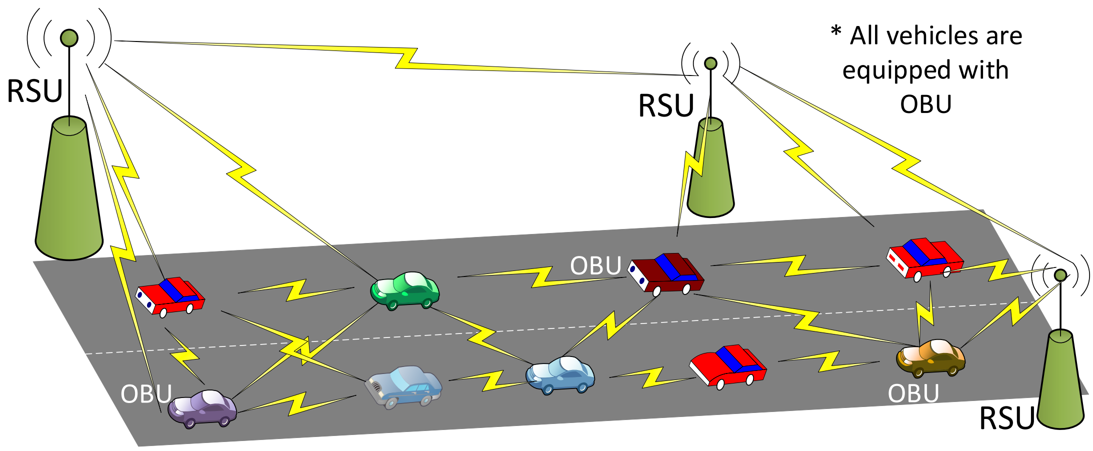

Figure 1.1: Vehicular communication in VANET.

V2V communication is critical in VANET due to the limited number of RSUs; however, VANET presents unique challenges due to vehicles' high mobility [1]. While VANETs do not suffer from energy deficiencies, they experience significant message loss when the number of vehicles increases, as messages are disseminated among the vehicles without a central infrastructure. Scalability is thus a significant issue in VANETs, which also face problems with the hidden terminal, broadcast storm, message security, packet routing, congestion control, and resource management [2-5]. To address these issues, the literature has explored hierarchical structures where nearby vehicles with shared characteristics form clusters. Clustering considers a large network of vehicles as a collection of smaller networks or clusters.

1.2 Cluster Formation in VANET

In VANET clustering, the cluster head (CH) plays a key role as the leader of the cluster. The formation of clusters is shown in Figure 1.2. The clustering process can be initiated in various ways based on input parameters, and the resulting cluster includes CH, cluster members (CM), and sometimes cluster gateway-members (CG). A cluster can have one CH, two CGs, and any number of CMs. In this structure, CH functions as a mobile router and CMs function as mobile nodes. CGs act as intermediaries between CH and CMs by facilitating communication with other clusters.

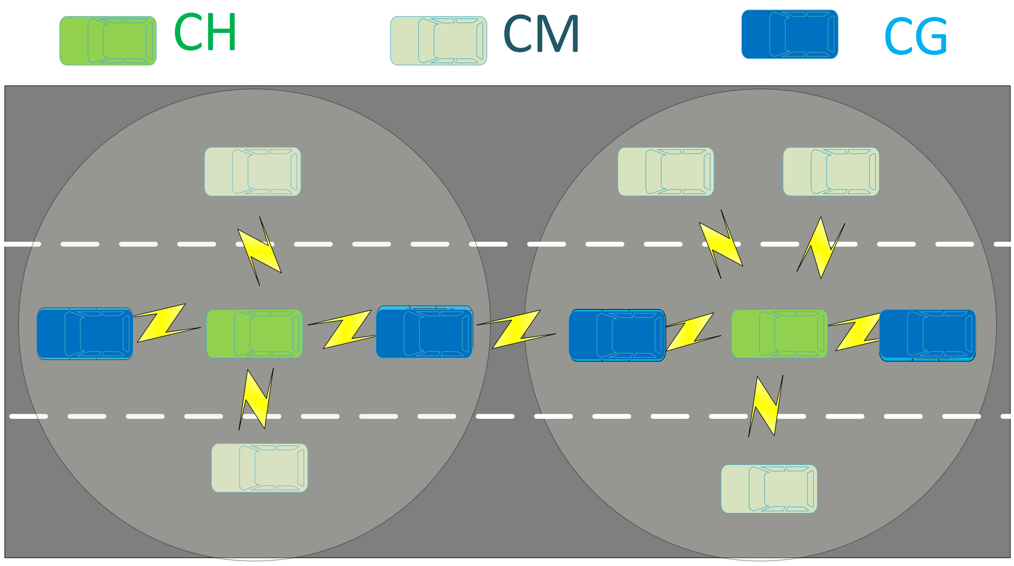

Figure 1.2: Cluster formation in VANET.

1.3 Problem Statement

In VANET clustering, vehicles are grouped based on similar features such as velocity, distance, or degree of vehicles. However, due to the highly mobile nature of vehicles with random velocity and frequent direction changes, clusters often break down, which undermines stability of the clusters. The primary challenge in VANET clustering is to minimize the breakdown of clusters and create stable ones.

Clustering is introduced in VANET to address issues such as scalability. In a city, scalability is more critical at junctions or intersections where many vehicles converge. However, clusters often break down at intersections due to vehicles' different route after the intersection, resulting in re-clustering and diminishing the effectiveness of clustering. Breakdown of clusters, especially at intersections, is critical for cluster stability.

Additionally, the research on VANET clustering is advancing rapidly; however, simulation platforms for VANET are not advancing at a steady rate to meet the growing demand for a hierarchical structure of high-mobility vehicles. A simulation platform for VANET clustering requires the use of a large database, real-time data processing, complex multi-level calculations, and accessibility over the internet. A new VANET simulation platform with clustering module is necessary.

1.4 Objectives of this Research

The *objectives* of this dissertation are summarized as follows:

- Our first objective is to develop a VANET clustering protocol that can enhance cluster stability by considering appropriate mobility parameters. Stability is measured by CH-related metrics and CM-related metrics. Our aim is to improve the performance for both CH- and CM-related metrics taking account of junctions in the city environment and minimizing the number of clusters.
- Roads are constructed as a combination of highways and intersections. We want to develop a second clustering protocol that can provide better stability in both highway and intersection, regardless of the road topology. Instead of

number of metrics, we want to focus on achieving the optimum result for the metrics which are more significant for cluster stability.

- We also want to develop a fully functional VANET simulator that is easily accessible over the internet, machine-independent, and includes a clustering module.

1.5 Previous Works

Many clustering protocols have been proposed in the literature to provide cluster stability minimizing cluster breakdown. Several VANET clustering protocols [6-9] use the relative velocity to select the CH. Vehicular multi-hop algorithm for stable clustering (VMaSC) [6] falls in this category which selects the vehicle with the lowest average relative velocity as the CH, so that member vehicles require more time to exit the CH's transmission range (TR). Once the CH is selected, a cluster is formed including vehicles within the CH's TR.

The position of vehicles is also taken into consideration in selecting the CH in some protocols [10, 11]. Dynamic clustering in VANET (DCV) [10] is in this category which uses the position of vehicles as the basis for clustering. The distances between the vehicles within the clusters are measured, and the relatively central vehicle is selected as the CH. This ensures that the member vehicles need to cover a greater distance to exit the cluster. Once the CH is selected, a cluster is formed including the vehicles within the TR of the CH.

Efficient CH selection (ECHS) protocol [12] used the degree of vehicles as the basis of clustering. The degree of each vehicle is counted, and the vehicle with the highest degree is selected as the CH. This approach also used the centrality of the CH like DCV; however, ECHS choose the vehicle with even degree in both sides as the CH. They also proposed multiple CGs at both sides. After selecting the CH, a cluster is formed, including the vehicles within the TR of the CH.

The selection of CH and the formation process of a cluster are crucial for achieving cluster stability in VANET. In the past, VANET clustering protocols have mainly relied on either the relative speed or the relative position of the vehicles to form clusters. While some existing protocols have addressed certain problems associated with VANET clustering stability, such as VMaSC being optimized for CH-related metrics but not for CM-related metrics, or DCV being optimized for only CM-related metrics but not for CH-related metrics. Therefore, an effective and stable clustering protocol is *absent* in the literature that can optimize both CH-related and CM-related metrics [13].

Secondly, the existing protocols are mainly effective on highways or straight roads without any intersections. They have been simulated on straight road only. Intersections are not considered at all in these algorithms which is not realistic in a city environment, where intersections are very common. On the other hand, the algorithms that considered intersections [14, 15] are not effective. Intersection-based clustering (IBC) [14] considered intersection but forced the clusters breakdown at the twice of transmission range from the intersection which increases the rate of CH change and reduces the average duration of CH. The authors thought that any accident may not happen in this region and breakdown at a single point will not cause scalability problem or security problem in that specific area which is not realistic. Consequently, both types of existing protocols suffer from frequent breakdown of clusters. Therefore, there is a need for new

VANET clustering protocol that takes into account road intersections properly during the CH selection and cluster formation process, and effective for highways also to address the stability issue in VANET.

Finally, it should be noted that while VANET research is advancing rapidly, simulation platforms for VANET clustering are not keeping pace with the demand for more complex and hierarchical structures of highly mobile vehicles. Existing simulators, such as NS-2 [16], lack the necessary features for VANET clustering simulation. While specialized VANET simulators, such as TraNS, Veins [17], and NetSim [18], have been developed, they often require proprietary software or specific operating systems and can be difficult to learn and use. Furthermore, they do not include a clustering module or offer easy internet accessibility. Therefore, the development of a new VANET clustering simulator [19] is a necessity to meet the growing demand for simulation tools that can accurately model the behavior of highly mobile vehicles in complex, real-world scenarios.

1.6 Proposed Methods

We proposed two VANET clustering protocols in this dissertation: junction-based clustering protocol for VANET (JCV) and stable dynamic feedback-based predictive clustering (SDPC). We also developed a VANET simulator, cloud-based VANET simulator (CVANETSIM). The architecture of JCV and SDPC is shown in Figure 1.3.

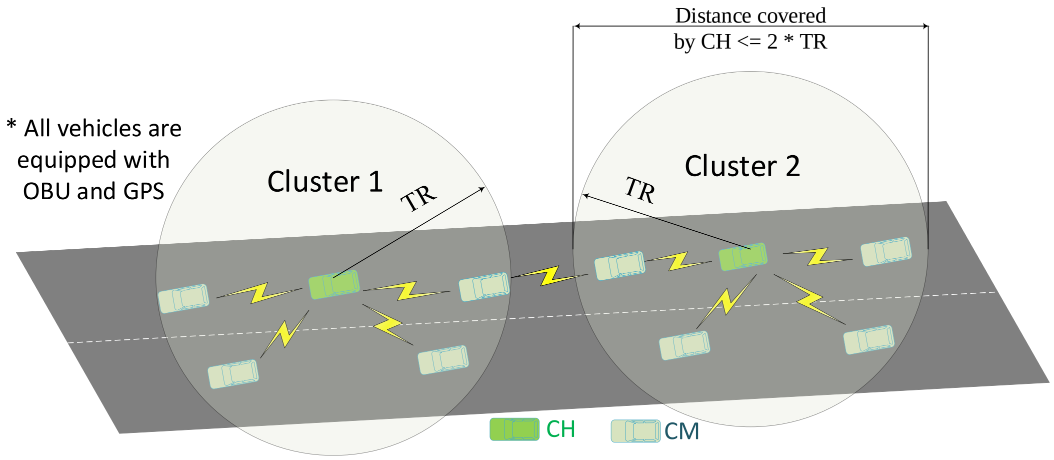

Figure 1.3: Architecture of JCV and SDPC.

1.6.1 JCV

JCV is a single-hop VANET clustering protocol that aims to achieve better clustering stability for both CH and CM. We have simplified the state transitions and reduced the number of states for vehicles and used a greater number of parameters in JCV. JCV considers the vehicles' movement at the junctions during the CH selection process to reduce the number of CH change. It uses the position of vehicles to calculate the distances of the vehicles. JCV also considers the degree of vehicles to reduce the number of clusters. By fulfilling other conditions, the vehicle with the lowest relative distance is selected as the CH. The parameters include the position of the vehicles, distance, moving direction, the movement of the vehicles at the intersection, transmission range, and degree. In JCV, vehicles are not forced to join in any cluster. They have freedom not to join in the clusters if the vehicle considers its joining is not beneficial. By considering these parameters, we aim to improve the stability of JCV for both CH and CM.

JCV is compared with VMaSC and DCV based on all kinds of stability-metrics regarding CH, CM and un-clustered vehicles. The results demonstrate that JCV outperforms both VMaSC and DCV in terms of CH-related, CM-related, and EN-related metrics. It also shows that JCV can reduce the number of clusters, the number of non-clustered vehicles and clustering overhead. A higher CH duration (4%), CM duration (8%), number of CM (6%), the ratio of CM (22%), and a lower number of CH change rate (14%), number of CH (10%), number of non-cluster vehicles (7%), clustering overhead (35%) have been achieved by JCV. Overall, JCV provides better clustering stability in VANET in terms of different type stability metrics.

1.6.2 SDPC

In SDPC, we want to achieve stability for both highway and intersection. SDPC, the all-weather clustering protocol, considers various parameters such as velocity, position, distance, acceleration, moving direction, and future movement to ensure cluster stability even at intersections. The CH is selected from the majority-vehicles. Majority-vehicles are defined by the largest number of vehicles that will run in the same direction after the next intersection. SDPC predicts the movement of every vehicle in the future and calculates the probable CH-lifetime for the probable clusters to maximize the CH lifetime and minimize the rate of CH change. SDPC aims to select the best stable vehicle as the CH. This helps in improving the overall performance of VANET clustering protocols, especially in situations where road topology is complex, and vehicles are moving in different directions.

The stability of VANET clustering protocols can be measured by various metrics such as the average CH duration, the average number of CH changes, the average CM duration, and the average clustering overhead, etc. However, the average duration of CH, the average number of CH change, and the average clustering overhead are considered the most critical metrics to evaluate the stability of VANET clustering protocols. Achieving a higher average CH duration and minimizing the number of CH changes are the most significant challenges in achieving stable clustering in VANETs. Additionally, reducing the clustering overhead is also necessary for stability. The average CM duration is also essential for VANET clustering stability if it is combined with the average CH duration, the average CH change rate, and the average clustering overhead.

SDPC was compared with four existing algorithms, and it demonstrated superior cluster stability on various road topologies. The results show that SDPC outperforms the four existing clustering algorithms in terms of cluster stability metrics. Specifically, SDPC achieves a 50% reduction in the average CH change rate, a 15% increase in the average CH duration, a 6% increase in the average CM duration, and a 35% decrease in clustering overhead. These improvements demonstrate the effectiveness of SDPC in achieving better stability and performance in VANET clustering.

1.6.3 CVANETSIM

CVANETSIM is a fully functional VANET simulator that includes a clustering module and can be accessed over the internet without requiring specialized installation or new skills. The discrete-event simulator, CVANETSIM, is developed using Java Server

Page and MySQL database server in a Tomcat web server. We have also used a PHP server to access databases over the internet. Since CVANETSIM is accessible from the internet, it does not serve as a simulator only, any kind of real-life VANET application can be developed using this simulator without storing or processing any data on a local machine. CVANETSIM used MySQL database to access SUMO data and then process them to using multi-level matrix to create cluster. CVANETSIM analyzes various features of vehicles such as degree, transmission range, velocity, relative velocity, distance, position, relative distance, angle, etc. After simulation, it generates various results that can be used to create charts.

1.7 Comparison with previous works

The dissertation presents a comparison of JCV, SDPC, and CVANETSIM in Sections 1.7.1, 1.7.2, and 1.7.3, respectively.

1.7.1 JCV

JCV offers the advantage of reducing the number of states required for vehicle classification. In comparison to VMaSC, which utilizes five states, and DCV, which uses four states for vehicle classification, JCV uses only three states.

JCV also simplifies state transitions. The number of state transitions required in VMaSC, DCV, and JCV are nine, eight, and six, respectively.

VMaSC selects the vehicle with the lowest average relative velocity as the CH and DCV selects the vehicle that is in geographically center position as the CH. JCV gives precedence to the distance of the vehicles over the velocity; however, JCV considers none of these parameters are sufficient for cluster stability. JCV considers these parameters along with the movement of the vehicles at the intersection and the degree of vehicles to increase stability.

In JCV, a vehicle has freedom to make its own decision whether it will join in a cluster or not by considering its future route. The difference of JCV with the previous works is that a vehicle will join into a cluster only when joining a cluster is beneficial; otherwise, a vehicle will remain in the un-clustered state.

DCV used a lower TR than the actual transmission range which creates extra number of clusters while VMaSC and JCV used the actual transmission range as the TR.

VMaSC is a multi-hop clustering protocol which allows vehicles to join in a cluster through another CM, whereas DCV and JCV are single-hop clustering protocol which allows the vehicles to join in a cluster that are within the TR of the CH.

VMaSC focuses on optimization of CH-related metrics and DCV focuses on CM-related metrics, whereas JCV optimized both CH and CM-related metrics.

JCV used degree of vehicles as a parameter during the CH selection process and cluster formation which reduce the number of clusters compared to VMaSC and DCV, who did not consider degree of vehicles during cluster creation.

JCV requires you to know the movement of the vehicles at the next intersection. This minimum information sharing gives JCV a huge advantage. In destination-based clustering, the destination of the vehicle is shared with other vehicles that is a security threat and violation of privacy. JCV achieves the same performance but shares only the movement of the vehicles after the current road segment, i.e., for the next segment only. When two vehicles are running towards the same destination, they only share their movement of one road segment at a time and continue as a cluster up to the destination. Hence, JCV used the intersection wisely and efficiently.

1.7.2 SDPC

VMaSC selects the vehicle with the lowest average relative velocity as the CH and DCV selects the vehicle in geographically center position as the CH. ECHS selects the vehicle with even degree as the CH and divides vehicles in three groups depending on velocity. IBC forces the clusters to break down at the twice of the TR from the intersection. SDPC selects the vehicle with the highest predicted stability as the CH. SDPC uses majority-vehicles and the movement of the vehicles at the next intersection along with a predictive approach of a probable CH with the best lifetime during the CH selection process to maximize the cluster stability.

The advantage of SDPC is a lower number of states and a lower number of state transitions. While VMaSC used five states of vehicles and nine states transitions and DCV used four states of vehicles and six state transitions, ECHS and IBC did not mention anything about state or state transition optimization. However, SDPC used three states of vehicles and reduced the number of state transitions to four. It results in a lower number of clustering overhead for SDPC.

While VMaSC, IBC and DCV did not consider the degree of vehicles at all during the CH selection process, ECHS selects the CH based on the degree. SDPC used degree in a limited sense during the CH selection process to break the tie among multiple candidates for the CH to reduce unnecessary clusters.

SDPC is a single-hop clustering protocol like DCV, ECHS and IBC, whereas VMaSC is a multi-hop clustering protocol. However, single-hop VMaSC has been used in the simulation to maintain uniformity among the protocols.

SDPC and IBC requires to know the movement of the vehicles at the next intersection, which is not needed in VMaSC, ECHS and DCV.

In SDPC, the vehicles who do not fall in the majority-vehicle category, they can either choose to form their own cluster or can join in the existing cluster without affecting the CH selection process of the majority-vehicles, which is unique for SDPC.

1.7.3 CVANETSIM

CVANETSIM is the first of its kind and offers several advantages. It includes a database server, real-time data processing, is machine-independent, accessible from the internet, and features a clustering module. Users do not require any specialized skills, installations, or extensive training to use it. The following features are unique to CVANETSIM:

- Fully functional clustering module.
- Using a database to provide efficient access and update on data.
- Cloud-based platform to access over the internet.
- No installation or specialized skill needed.
- Easy to use graphical user interface (GUI).

1.8 Contributions

The dissertation makes significant contributions to the field of VANET clustering and provides new insights into the development and evaluation of clustering protocols. The key contributions of the dissertation are summarized as follows:

- **JCV VANET clustering protocol**: We proposed a new VANET clustering protocol - junction-based clustering protocol for VANET (JCV). JCV achieves better stability for both CH and CM. The proposed clustering protocol is evaluated using various metrics including average CH duration, number of CH changes, average CM duration, average number of clusters, and clustering overhead. The evaluation showed that JCV outperformed existing clustering protocols in terms of stability.
- **SDPC VANET clustering protocol**: To provide a VANET clustering protocol which can provide stability to the cluster regardless of road topology, we proposed the second protocol - stable dynamic feedback-based predictive clustering (SDPC). SDPC provides stability for both highways and intersections, regardless of road topology. SDPC is evaluated using various metrics such as average CH duration, number of CH changes, average CM duration, and clustering overhead. The evaluation results showed that SDPC outperformed existing clustering protocols in terms of stability regardless of road topology.
- **Development of a VANET simulator**: A cloud-based VANET simulator (CVANETSIM) is presented with a clustering module which can be accessed

over the internet. The simulator provides a platform for researchers to develop and evaluate different VANET clustering protocols.

1.9 Summary

The motivations for the dissertation, background of the dissertation, and the stability challenge in VANET clustering have been presented in this chapter. The problem statement, our objectives, the previous works, and the proposed solutions are also presented. Then, we presented a comparison of the proposed works with the previous works and our contributions in this dissertation. In the next chapter, we presented the literature review with a detailed classification of VANET clustering protocols.

1.10 Organization of the Dissertation

The rest of the dissertation is organized as follows. Chapter 2 presents a literature review with a detailed classification of VANET clustering protocols. In Chapter 3, we proposed our first clustering protocol to optimize both the CH-related metrics and the CM-related metrics using multiple parameters. In Chapter 4, we proposed our second clustering protocol to provide stability regardless of topology. A cloud-based simulator for VANET is presented and the simulator setup is described in Chapter 5. In Chapter 6, we presented the simulation results and analysis of the results comparing JCV and SDPC with other existing clustering protocols. Finally, Chapter 7 has the concluding remarks and the future works.

Chapter 2

Literature Review

A comprehensive analysis of existing VANET clustering protocols, their classification, and the challenges associated with them have been presented in this chapter. By presenting a detailed analysis, the chapter provides directions for future research in the field of clustering protocols for VANET. Researchers can use this information to identify the gaps and limitations of the existing protocols and design new protocols that address the challenges and provide better performance [20]. Overall, the literature review chapter can play an important role in advancing the research in VANET clustering protocols.

2.1 Taxonomy of VANET Clustering

Many clustering schemes have been published in the literature. The classification of clustering protocols in this dissertation provides a comprehensive overview of the existing literature in the field of VANET clustering protocols which was missing in the existing literature reviews [1] [21-29]. The taxonomy of VANET clustering protocols has been shown in Figure 2.1. Clustering parameters, evaluation metrics, and their relationship with stability are thoroughly analyzed. The schemes have been classified into three primary categories: intelligence-based strategies, mobility-based strategies, and multi-hop-based strategies. Intelligence-based strategies are further classified into machine learning, fuzzy logic, and hybrid algorithms. Mobility-based strategies have been classified into vehicle mobility and network mobility algorithms while multi-hop-based strategies are classified into 2-hop and *N*-hop algorithms. It can be noted that mobility-based strategies and multi-hop-based strategies focus on the stability of the clusters while most of the intelligence strategies do not focus on cluster stability in cluster creation. Each group of algorithms have been classified further which is not shown in the figure but discussed in the respective section. By categorizing the protocols into these groups, it becomes easier for researchers to identify the most suitable approach for their research and to compare different approaches based on their characteristics. Additionally, the classification also highlights the challenges in each group and provides insights for future research.

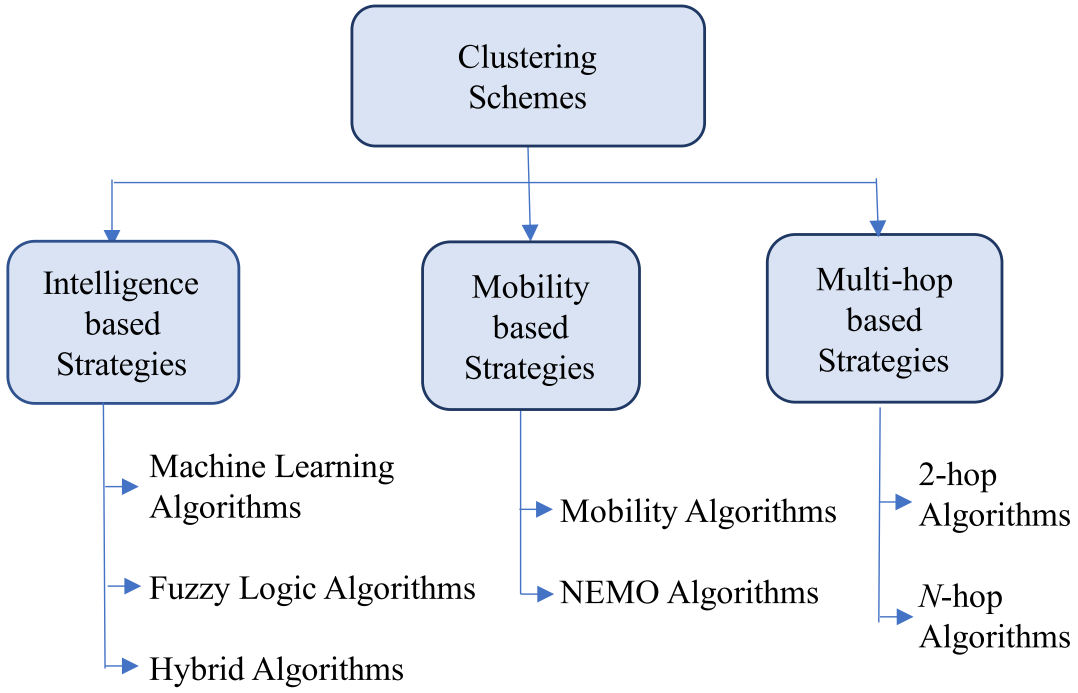

Figure 2.1: Taxonomy of VANET clustering schemes.

2.2 Intelligence-based Strategies

Clustering is an important concept in machine learning and data mining. Many clustering algorithms have been developed such as *k*-means and hierarchical clustering. Some of the clustering algorithms from machine learning and data mining have been used by some authors in VANET for vehicle clustering. Fuzzy logic is also used in some protocols in VANET clustering. We have classified and analyzed machine learning algorithms and fuzzy logic algorithms separately. We have also analyzed the hybrid architecture of machine learning and fuzzy logic in a separate section. It should be noted that most of the intelligence strategies in VANET do not focus on cluster stability. Classification of intelligence strategies is shown in Figure 2.2.

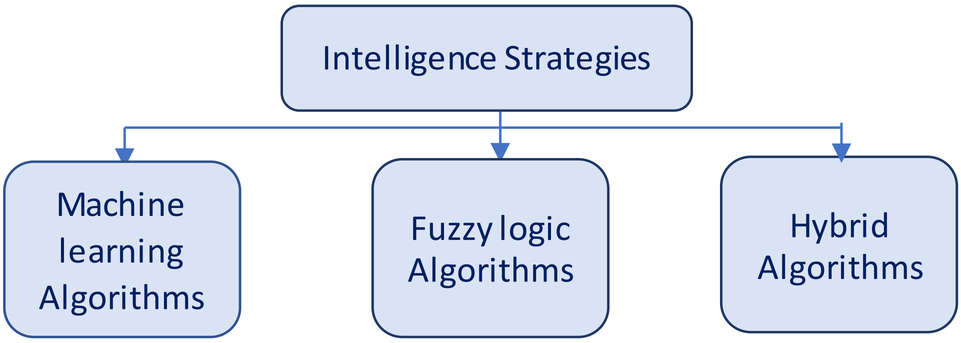

Figure 2.2: Classification of intelligence VANET clustering strategies.

2.2.1 Machine Learning Algorithms

Clustering algorithms are used in data mining and machine learning to cluster similar types of objects. The *k*-means algorithm is utilized in multiple VANET clustering protocols to create *k* number of clusters. Initial centroids are assumed and coordinates of the vehicles are given as input. Then Euclidean distance is calculated to determine the new centroid and the centroids are elected as the CH. Whenever a CM joins into a cluster or leave a cluster, the mean of the cluster is susceptible to change and need to re-calculate the new mean of the cluster to reflect the change that can lead to elect a new vehicle as the CH. In hierarchical clustering, Euclidean distances of all the vehicles are calculated and the vehicles connect with each other sequentially starting from the minimum distance. Table 2.1 presents the schemes which are based on machine learning clustering algorithm.

Table 2.1: Evaluation of machine learning strategies.

Ref.	**Algorithm**	**Evaluation**
[30]	*K*-means	Certificate
[31]	*K*-means	PDR, throughput, overhead
[32]	*K*-means	PDR, throughput, delay
[33]	*K*-means	CH duration, signal quality, TR
[34]	*K*-means	Sum Rate
[35]	*K*-means	Delay, overhead
[36]	*K*-Harmonic means	Coverage
[37]	Hierarchical	PDR, throughput, delay, degree
[38]	MFO	No. of Clusters, TR
[39]	MFO	No. of Clusters, grid size

2.2.1.1 *K*-means-based Clustering Protocols

To enhance the security of communicating vehicles, *k*-means-based clustering protocol is proposed by Zhang et al. [30], which is the first attempt to use a data mining or machine learning algorithm for VANET clustering protocol. It is improved by Almulla et al. [35]. Instead of mobility parameters, a machine learning algorithm is used for faster certificate validation. Since the security is the main concerned here. Clustering is applied to the list of certificates received from the vehicles to extract security measurements of the vehicles taking assistance from RSU. However, the number of clusters is an input of this algorithm but density and number of vehicles can vary in different cases. Hence, the number of clusters should be an independent variable that can increase or decrease depending on the number of vehicles and density, otherwise, the number of CM in a cluster can be very high or very low.

Another *k*-means-based clustering algorithm is proposed by Bansal et al. [31] to divide the vehicles into clusters. Three parameters are given as input. Both *x* dimension and *y* dimension, i.e., the position of the vehicles is considered to form the clusters. The number of clusters is given as input, then a modified *k*-means algorithm is applied to divide the vehicles into clusters. To choose the CH, the centroid of the cluster is selected along with some security issues. To increase security, a hashing technique is used to encrypt or decrypt the packets. After selecting the centroid as the CH, the rest of the vehicles join the cluster as the CMs of the cluster. No separate maintenance phase is required since the clusters cannot overlap in intelligent clustering including *k*-means algorithm. Simulation results show that packet delivery ratio (PDR) and throughput can be improved in the proposed algorithm while routing overhead increases compare to base *k*-means algorithm; however, the number of clusters is an input of this algorithm, but density and number of vehicles can vary in a different scenario. Hence, the number of clusters should be an independent variable that can increase or decrease depending on the number of vehicles and density. Otherwise, the number of CMs in a cluster can be very high or very low. Additionally, if any vehicle joins or leaves the cluster, the mean of the entire cluster can be changed with the change of the CH itself that reduces the cluster lifetime and cluster stability.

Instead of enhancing security measure [30] [31], *k*-means is used to solve data congestion problem [32] to decrease packet loss and end-to-end delay. Clustering is performed using *k*-means algorithm based on distance and direction along with message size, the validity of messages, and type of messages. Two types of control strategies have been used: open-loop and closed-loop solutions. The open-loop solutions prevent congestion before it happens while closed-loop solutions control the congestion after detection. Instead of vehicles, clustering of messages is performed at RSU where features, number of clusters and number of iterations are given as input. However, the number of clusters is fixed, and initial centroids are set based on a first come first serve basis which is inefficient for stability and cluster lifetime.

A modified *k*-means algorithm [40] is proposed for straight-road or highway environment where random selection of the number of clusters and the initial CH is prevented. Vehicles are assigned into clusters based on the link reliability depending on vehicle density, relative speed, and distance. The CH is selected in this routing protocol based on velocity, degree of node, and the buffer size; however, no major cluster stability metrics are evaluated.

K-means is also used to increase stability [33] of the clusters. The distance of the vehicles is calculated to find the minimum average distance from a given vehicle to be selected as the cluster head. Distance is measured by Euclidean distance and all pair shortest path is calculated within a cluster to choose the CH. However, the limitation of the number of clusters persists like [30], [31] and [32]. A cellular-based clustering strategy is proposed [34] to enable multiple users to establish connections simultaneously using *k*-means clustering. The strategy is not proposed particularly for vehicular communication.

One limitation is common for all the above strategies. The limitation is *k*-means clustering algorithm is sensitive to the initial centroids. To overcome this drawback adaptive *k*-harmonic means [36] is proposed where a vehicle must meet the minimum bandwidth requirement to be elected as the CH. Traditional *k*-harmonic means, where relative distance and centroids are measured, is modified to make compatible with the mobility of vehicles. The velocity of the vehicles is considered along with their position to form the clusters. However, the limitation of the fixed number of clusters is continued that can cause a problem in V2V communication with many vehicles.

2.2.1.2 Hierarchical Clustering Algorithms

To overcome the limitations of *k*-means algorithm, an agglomerative hierarchical clustering approach [37] is proposed where the direction and speed of the vehicles are considered to form a cluster along with some quality of service (QoS) parameters. The past duration of the node acting as a CH, PDR, and TR are considered for CH. In hierarchical clustering, Euclidean distances of all the vehicles are calculated and the vehicles connect with each other sequentially starting from the minimum distance and do not require the number of clusters as the input. The vehicles are considered as two clusters based on their direction and CH is selected based on the duration of acting as a CH in the past. However, while the implementation of *k*-means is simple, the hierarchical approach requires proximity matrix calculation. Moreover, once a CM joins to a cluster, it cannot be undone, but topology can change any time in VANET and requires a change in the cluster also.

2.2.1.3 Optimization Algorithms

Nature-inspired algorithm is proposed [38] [39] for VANET clustering, based on moth-flame optimization (MFO) [41]. The MFO algorithm depends on the navigation method of moths, which follow a spiral flying path, called transverse orientation. Moth can fly maintaining a fixed angle with the moon that can be considered as a straight path for a long distance. The same concept is used to cluster the vehicles in a highway environment to optimize cluster considering speed, direction, grid size, the degree of a node, and transmission range of the vehicles; however, no result is provided to prove the stability of the clusters in terms of the average CH duration or the average CH change rate.

2.2.2 Fuzzy Logic Algorithms

Many clustering strategies in VANET are based on fuzzy logic as shown in Table 2.2. Instead of the binary value of true or false, the degree of certainty is considered in the fuzzy logic system (FLS).

Table 2.2: Evaluation of fuzzy logic strategies.

Ref.	Fuzzy Input	Evaluation
[42]	Speed, distance, acceleration	CH duration, cluster size, CM duration, delay, reliability, cluster size, PDR
[43]	Relative Speed, Distance	CH duration, cluster size, CM duration
[44]	Speed, acceleration, direction	CH candidate values
[45]	Speed, acceleration, brake frequency, coordinate, time	Message credibility
[46]	Position, velocity	Delay, packet loss, throughput
[47]	Speed, degree of node, link quality	CH duration, service delay
[48]	Speed, centrality, security, trustworthiness	Remain or leave the cluster

Five steps of FLS can be considered in terms of VANET. In the first step, the input parameters such as relative speed, vehicle distance, moving direction, and acceleration are defined. In the second step, fuzzification is performed where a fuzzifier transforms the input parameters into a fuzzy set. The third step is performed by an inference engine where the fuzzy rules are defined based on the knowledge base and applied on the fuzzy set to produce the output fuzzy sets. Defuzzification process is performed by a defuzzifier in the next phase to generate crisp output values from the output fuzzy sets. In the last step, tuning of the system is performed reviewing the range of the inputs and outputs, revising the fuzzy sets, and tuning the rules. The steps of FLS is shown in Figure 2.3.

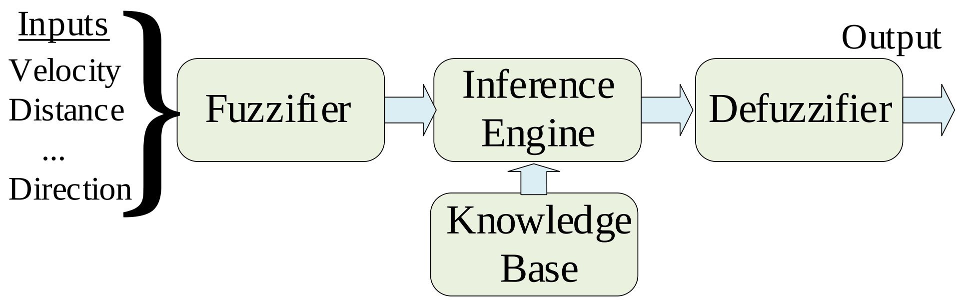

Figure 2.3: Structure of FLS in VANET.

2.2.2.1 Stability Algorithms

A fuzzy logic-based CH selection algorithm is proposed by Hafeez et al. [43], the first instance of introducing fuzzy logic system in VANET clustering. In this work, a fuzzy system is developed where two metrics such as relative speed and distance are given as input to the fuzzifier to start the clustering formation process. The fuzzy logic inference system is used to predict the speed and position. Then the CH is selected by the defuzzifier. If the stability factor of the CH falls below a predefined threshold value, a new member is selected as the CH. Merging two clusters is allowed in this scheme. If the second CH reaches to half of the TR of the first CH, the second CH will merge with the first CH; however, considering only two input parameters in the fuzzy input sets affects the performance of the selection of the CH. The average CH change rate, which is the most important stability metric, is not evaluated.

The protocol of Hafeez et al. [43] is improved [42] by adding acceleration as the input parameter along with speed and distance to create more stable clusters. Fuzzy logic inference system is integrated with an adaptive learning mechanism to provide a more stable cluster. Like [43], the stability of the vehicle comparing to the neighbor vehicles is given preference to be selected as the CH. Similarly, merging of two clusters is allowed and follow the same process; however, three input parameters are also considered insufficient for the highly dynamic nature in VANET. Along with vehicle speed and vehicle centrality, trust related parameters are also considered to form the clusters to decide a vehicle will remain with the cluster or leave [48]; however, acceleration and direction of the vehicles are not considered in this approach. To further improve the performance of [42], the direction of the vehicles is also considered [44] along with the speed, distance, and acceleration. However, the major mobility metrics are not evaluated to justify the performance of the protocol.

2.2.2.2 Congestion Control Algorithms

Instead of stability of the cluster, congestion control is highlighted in a multilevel cluster-based message fusion approach [45]. They used feature level fusion at the low level and decision level fusion at the high level. To minimize information redundancy fuzzy logic-based approach is used at the low feature level. To detect congestion, probability-based information fusion is proposed at the high decision level. However, no simulation results are provided to establish their claim.

2.2.2.3 QoS Algorithms

Stability is given priority in [42], [43] and [44] without considering QoS. A hybrid network architecture of V2V and cellular network is proposed [46] where QoS is improved using a fuzzy logic-based gateway selection technique. Cluster is formed considering traffic type of the vehicle and the CH is selected using received signal strength and load. The CH is the leader of the cluster but may not work as the gateway to communicate with the cellular network [46] which is a unique concept for RSU assisted VANET clustering strategies while CH or CG is generally selected as the gateway in VANET; however, dynamic clustering at higher speed will cause frequent CH change as well as frequent change of CG candidates and will increase the complexity selecting the gateway node that can increase packet loss and end-to-end delay.

To solve the limitation of resources in dynamic vehicular cloud architecture, a fuzzy-based CH selection process is proposed [47]. To improve reliability and quality-of-service, a CH works as a cloud controller who can create, delete and update the vehicular cloud. CH is selected based on a fit factor. However, reliability and QoS of such strategy is questionable, since the increase in degree of node can decrease the performance of the CH allocating resources to a large number of members. Moreover, performance of the scheme degraded in the absence of RSU.

2.2.3 Hybrid Strategies

Machine learning algorithms are integrated with fuzzy logic system to make the cluster formation process and CH selection process more efficient in a hybrid manner. Classification of hybrid algorithms is presented in Table 2.3.

Table 2.3: Evaluation of hybrid strategies.

Ref.	Algorithm	Evaluation
[49]	Fuzzy, Q-learning	CH duration, percentage of stability, service delay
[50]	fuzzy, Q- learning	PDR, throughput, no. of handoffs, delay
[51]	Fuzzy, Q- learning	PDR, no. of collided frame, delay, throughput
[52]	Fuzzy, Q-learning	PDR, no. of collided frame, delay, throughput
[53]	K-means, fuzzy	Congestion
[54]	Fuzzy, Dolphin Swarm	Detection rate, detection time, false positive rate

2.2.3.1 Reinforcement Learning Algorithms with Fuzzy Logic

To improve efficiency and reliability of cloud services in a vehicular environment, new architecture is proposed [47] [49] using a reinforcement learning algorithm, Q-learning, along with the fuzzy logic algorithm. The CH is selected based on fuzzy logic and resource management is improved using Q-learning-based service provider selection technique. CMs are limited to the communication range of the CH. CH is selected based on the fit factor, where the cluster is formed depending on speed, degree, and RSU link quality. Every vehicle broadcasts its fit factor to be selected as the CH. Resource management is improved by deploying the Q-learning technique to select the service provider from the neighborhood vehicles that improve the efficiency of the CH selection process. Three different queuing methods such as first in first out, bandwidth aware, and resource-aware are used. However, among the mobility parameters, the relative speed is only considered to select the CH, other parameters such as acceleration and direction are ignored. Moreover, the algorithm is RSU dependent.

A data storage scheme is proposed [50] that store the data employing a fuzzy logic-based protocol considering multiple metrics such as throughput, stability, and bandwidth efficiency. To increase the stability of the fuzzy decision, Q-learning is used. A slow vehicle is selected as the CH to avoid frequent change of cluster heads to make the cluster more stable considering vehicle velocity, degree of node, and channel condition. However, slow vehicles cannot be the most suitable candidate to become a CH because the faster cars will cross the slow vehicles in a relatively short period of time that will further destabilize the clusters.

Specifically, for vehicle to road-side units communication, a reinforcement learning algorithm is used to create clusters [51] and fuzzy logic is used to make the cluster more stable considering vehicle mobility, vehicle distribution, and channel condition. The work in [51] is extended [52] to improve the performance using a multi-hop data virtualization. Instead of end-to-end feedback, hop-by-hop acknowledgment is used to increase end-to-end PDR in a multi-hop transmission. However, the QoS is an issue when vehicle density grows faster.

2.2.3.2 K-means with Fuzzy Logic

Authors of [49] [50] [51] and [52] used Q-learning algorithm where Bhanja et al. [53] addresses the issue of traffic congestion in a dynamic vehicle environment using *k*-means clustering algorithm integrated with Arduino controller and a PHP web server. A fuzzy rule-based inference system is proposed considering four attributes: vehicle speed, rain, fog, and brake frequency. For all the vehicles, the fuzzy congestion output is sent to a PHP cloud server through an ESP8266 wi-fi module. This module also generates a two-dimensional position of the vehicles as an alternate of GPS. The PHP server uses K-means clustering algorithm to form the clusters without any assistance from RSUs. However, the time to connect to the PHP server and to send or receive information from an external server is required that can cause additional delay. Moreover, k-means always choose the centroid as the CH and for any change in the cluster may cause to change the CH every time.

2.2.3.3 Dolphin Swarm Algorithms with Fuzzy Logic

All clustering strategies discussed here use a single CH for a cluster where the CH acts as the leader of the cluster. To reduce the overload of the CH in the cluster, a multiple CH scheme is proposed in the hybrid fuzzy multi-criteria decision making [54] where fuzzy analytic hierarchy process is used to make the fuzzy decision optimal. The load of the leader of a cluster is distributed among the CHs. To secure the communication, intuition detection system has been proposed utilizing the Dolphin Swarm behavior instead of rule-based system to detect newer attacks which are not present in the database and to differentiate between the malicious and the normal nodes. The CH is selected based on velocity, social contact, integrity, availability, etc. Each CH will appoint another CH based on security and trustiness, the new CH will appoint another new CH, hence, a clustered swarm of dolphins are created. However, clustering efficiency or clustering stability issues are not described, and no simulation result provided to measure the clustering efficiency or stability of the clusters based on multiple CHs.

2.2.4 Summary of Intelligence-based Strategies

The most important parts in VANET clustering process are the CH selection and cluster formation. Efficiency of clusters largely depends on the cluster formation process where stability of the clusters depends mainly on the CH selection process. The efficiency of the clusters is evaluated in terms of packet loss, end-to-end delay, and throughput more frequently while the stability of the clusters is evaluated based on the average number of CH change, the average CH duration, the average CM duration, and clustering overhead.

Intelligence-based protocols concentrate on cluster formation process and do not consider stability of the clusters in the most cases. As a result, the clustering protocols developed based on the intelligence algorithms may break frequently and cannot create stable clusters.

2.3 Mobility-based Strategies

The most common clustering strategies in VANET are mobility-based strategies. The mobility parameters of vehicles, such as relative speed, moving direction, acceleration, position etc., are the basic parameters used for mobility strategies. Creating efficient clusters is not the objective of mobility strategies. Because clusters break down frequently due to high mobility of the vehicles. As a result, stability of the clusters is the main concern for the mobility strategies.

The concept of network mobility (NEMO) was introduced in NEMO Basic Support Protocol (NBSP) [55], where mobile router (MR) can move from one access router to another access router along with its network retaining its internet protocol (IP) address. Many efficient algorithms have been presented in the literature [56-58] for efficient routing for NEMO. VANET clustering concept has a similarity with the concept of NEMO and some VANET clustering techniques are proposed in literature based on NEMO protocols. We presented NEMO-based clustering techniques in a separate section. This is a *difference* of our work with the existing works that we have provided a distinct classification for the NEMO-based strategies.

2.3.1 Vehicle Mobility Protocols

The most popular clustering techniques developed in VANET are based on vehicle mobility. Table 2.4 summarizes the mobility-based clustering protocols. Main purpose of the mobility-based clustering strategies is to provide more stability to the clusters; however, some mobility-based clustering strategies are proposed to facilitate data dissemination, medium access control (MAC) management, and QoS. Mobility metrics such as the average relative velocity of the vehicle, acceleration, position, direction, etc. are considered to select the CH and form the clusters. Stability-based metrics such as average CH duration, average CM duration, number of state change, etc. are evaluated in mobility-based clustering strategies.

Table 2.4: Evaluation of mobility strategies.

Ref.	Purpose	Evaluation
[7]	Stability	Cluster lifetime, overhead
[59]	Stability	CH duration, connection duration, re-association rate/time
[60]	Stability	Cluster lifetime, percentage of CH, state change
[61]	Stability	Cluster lifetime, no. of cluster change, no. of cluster
[62]	Stability	CH lifetime, no. of cluster change
[63, 64]	Stability	CH lifetime, CH change, throughput, cluster lifetime, end-to-end delay
[65]	Stability	CM duration, re-clustering time
[66]	Stability	Number of packets, collision ratio, overhead
[67]	Stability	No. of CH, delay estimation, overhead
[68]	QoS	CH duration, no. of clusters, PDR, overhead
[10]	Stability	No. of clusters, CH/CM duration, CH/CM change rate, state change, clustering efficiency
[69]	Stability	CH/CM duration, CH change, no. of clusters
[70]	Stability	CH/CM duration, CH change, no. of clusters

[71]	Stability	Cluster lifetime, average vehicle state transition
[72]	Stability, security	CH duration, CM duration, no. of state change, packet loss ratio
[73]	Stability	CH change
[8]	Data dissemination	Throughput, energy consumption, reliability, delay
[74]	Routing	PDR, delay, and overhead.
[75]	Routing	CH duration, number of Ch change, PDR, delay, overhead
[76]	Selective routing	PDR, delay
[77]	Data dissemination	Overhead
[78]	Congestion	CH change rate, PDR
[79]	Data dissemination	Success rate, data dissemination efficiency
[80]	QoS	Throughput, delay, PDR, packet loss
[81]	Congestion, QoS	CH/CM duration, number of clusters, PDR, delay
[82]	MAC, safety messages	Throughput, PDR, delay
[83]	Resource management	CH duration, sum rate, cumulative distribution function, throughput
[84]	Co-operative clustering	Act as CH, participation
[85]	MAC	Throughput, delay, PDR
[86]	MAC for safety message	Throughput, latency, overhead, packet loss

[87]	Weighted	PDR, number of clusters, overhead
[88]	General purpose	CH duration, CM duration, no. of state change. Overhead, single CH, single vehicle, no. of clusters

2.3.1.1 Stability Protocols

Dynamic clustering algorithm [62] is proposed to increase the stability of the clusters in a highly dynamic environment. The cluster is formed considering the similarity of the vehicles in terms of relative speed. The CH is selected based on the average velocity and acceleration of the vehicles without considering direction or future movement of the vehicle. Performance is evaluated based on two parameters only; average CH duration and average number of cluster change; however, average CM duration, average clustering overhead, etc. are not evaluated to measure the stability of the clusters. Similar cluster formation process and CH selection criteria are followed by Souza et al. [60]. Additionally, authors [60] prevent frequent merging of the clusters to increase the stability of the clusters. To accomplish this, several CHs are allowed to be present within the communication range for a certain amount of time. Hence, the lifetime of the clusters increases and increases stability. Like [62], CH lifetime is evaluated [60]. Moreover, percentage of CH in relation to total number of vehicles is also evaluated along with number of state change; however, CM related parameters are neglected. Hadded et al. [71] considered angular position and direction of the vehicles to create clusters; however, cluster lifetime is given priority without considering cluster members-related metrics.

Goonewardene et al. [65] proposed a robust mobility-adaptive clustering (RMAC) where the CH is selected based on relative speed, location, and direction of the vehicle. Unlike other clustering strategies, each vehicle maintains a routing table for neighbor vehicles which are beyond its communication range. A vehicle can operate in a dual state, that means, a vehicle can simultaneously act as a CH and as a CM. In a dual state, the vehicle will be the CH for its own cluster and a CM for one or more other clusters. CMs are 1-hop clusters where all the CMs are within the communication range of the CH, however, not all the vehicles within the range of a CH are CM. Therefore, overlapping of the clusters is possible, and multiple CHs can operate in close proximity without merging. However, stability is measured based on two metrics only: average CM duration and re-clustering time which is not sufficient to measure the performance of stability in a dynamic manner.

A mobility prediction-based clustering is proposed by Ni et al. [59] using Doppler effect during the movement of the cars. To predict the relative speed, vehicles exchange *Hello* packets periodically and calculate Doppler shifts to initiate clustering process. The vehicle with the lowest relative speed is selected as the CH. Once the cluster is formed, vehicles exchange message to predict the future movement; however, an analytical model is presented comparing with two MANET clustering algorithms, no simulation result is presented to compare the strategy with a VANET clustering algorithm. Similarly, software-defined networking enabled social-aware clustering algorithm is proposed [7] to improve the cluster stability based on a social pattern. The moving pattern and sojourn time are considered to get the social pattern. Vehicles are grouped in a cluster who follow the same route based on the historical movement pattern.

Relative speed and inter-vehicle distance are considered to select the CH. Even though simulation results presented [7] to shows that it can improve the performance from [59], the strategy is evaluated based on cluster lifetime along with clustering overhead only. The lifetime of a cluster is an important parameter but cannot be the only measurement to measure the stability of the clusters. A cluster may have a longer lifetime, but frequent CM disconnection can decrease its stability, hence, CM related metrics should also be considered to measure the stability of the clusters.

Direction-based clustering algorithm is presented by Maslekar et al. [66] [67] considering the direction of the vehicles along with its location during the formation process of the clusters. Direction at the intersection is determined by the destination which can be a security and privacy violation. Moreover, no simulation results have been presented comparing other VANET clustering algorithms. A similar approach is used by Zhou et al. [89] using the intersection and the help of the base stations. A comparison with [67] is presented for only packet loss ratio, overhead and CH lifetime. Along with the location and direction, the speed difference is also considered in Rawashdeh et al. [61] to form a stable cluster, specifically for highway environment. The vehicles that show similar mobility patterns are clustered in a single cluster where the vehicles with high mobility are in a single cluster and the vehicle with low mobility form a different cluster. Simulation is performed in C++ evaluating average number of cluster change per vehicle, average cluster lifetime, and total number of clusters; however, CM related parameters such as average CM duration and average state change are not evaluated. Along with the position and speed of the vehicles, acceleration is also considered [72] to provide more stability and security, however, the scheme is optimized for highway only. Additionally, the average CH duration achieved in this scheme is not significant which decreases cluster lifetime.

Traffic pattern of buses is used to improve stability [73] by decreasing the number of CH change. Velocity, position, and direction of the vehicles are used as the mobility metric along with fixed route pattern of buses in urban area. Number of CH change is compared with one-hop VMaSC [6], however, stability of the clusters does not depend on a single parameter of the CH. Moreover, CM related parameters are ignored.

The lowest neighbor vehicle mobility is considered to be the CH to increase cluster stability in neighbor-stability-based VANET clustering [63, 64]. The car changes their moving direction frequently, therefore, to reduce cluster formation time and to minimize the number of CH change, the vehicle with the lowest relative speed is selected as the CH in this scheme. However, performances are evaluated in terms of two MANET strategies, no simulation results are provided comparing with any VANET strategy. Also, the method is optimized for only the urban scenario. To provide stability to the clusters, mobility-based clustering scheme is presented [8] [9], where stability of the clusters is evaluated considering the average CM duration along with three parameters used [61]: average CH duration, average rate of CH change, and average number of clusters. However, the average clustering overhead is not evaluated. Moreover, a lower number of clusters does not always increase the stability of the clusters, because there is a possibility that a few numbers of clusters can hold many CMs at a time that can further destabilize the clusters.

Bersali et al. [9] proposed a new collaborative clustering algorithm based on node score where a vehicle with high node score is selected as the CH. The node score is calculated using degree of node, distance, link stability, average relative speed, average relative acceleration etc. They also introduced a backup CH for better cluster stability; however, how a backup CH improves cluster stability is not described properly. Moreover, only three metrics have been evaluated that is not sufficient to prove the performance of the algorithm.

Moving direction, relative vehicle position, and link lifetime are considered to form a cluster in DCV [10] for straight road. A temporary state for the CHs and a safe distance threshold have been introduced to increase the stability of the clusters. CH is chosen from the vehicles which are nearest to the center of a cluster so that its neighbor can spend more travel time to leave the cluster. Temporary cluster head (CHt) is used to begin the cluster formation process and it becomes CM if it has no member, otherwise, it changes to CH. If two clusters come closer than a predefined safe distance threshold, then they merge to become a single cluster. Along with the four metrics used in [69] and [70], four more metrics have been evaluated for clustering stability: average CM duration, average state change rate per node, number of vehicles in clustered state, and CM disconnection frequency.

DCV is proposed to improve the stability of the clusters. To elect as the CH, preference is given to a node with a geographical central position among the vehicles while average relative mobility is given preference in VMaSC. Since the node at the edge of a cluster is vulnerable to be detached from the CH, the transmission range in DCV is lower than the actual TR.

CH selection and cluster creation in DCV:

1. Set cluster length less than twice of *TR*.
2. Select the vehicle as the CH which is the nearest to the center.
3. The vehicles who are running in the same direction as the CH and one-hop distances between the CH and the vehicles are the CMs of the cluster.
4. Create a cluster based on the center position of the CH.
5. For the rest of the vehicles, REPEAT STEP 1 to 4.

However, this scheme is optimized for urban scenarios without considering the reliability issues, therefore, in a sparse environment this scheme creates a greater number of clusters that will decrease the average CH duration. Consequently, cluster stability will decrease, and cluster lifetime will become low. Most of the stability metrics have been evaluated; however, this scheme is optimized for CM metrics. Hence, CH-related metrics such as CH duration, number of CH change, etc., are not optimized. Moreover, many times it creates a large number of single clusters.

Intersection-based Clustering (IBC) [14] selects the vehicle with the highest selection metric as the CH when there is no intersection within the twice of transmission range. The most important feature of IBC is consideration of intersection; however, it forces clusters to break down at the distance of twice of transmission range from the intersection. As a result, the number of CH increases, and the CH duration decreases even though forced bread down decreases CH duration at the cost of higher CH rate. The average percentage of CH, the number of isolated vehicles, and the average cluster lifetime evaluated is inadequate to measure stability. If the road segment is close to or less than twice that of TR, which is common in a city, this protocol will suffer heavily. Moreover, the performance is not compared with any stability-based clustering protocols. We cannot say that accident will not happen at the twice distance from the intersection and we do not need optimized cluster in this place. Also, concentrating breakdown at a single point is always vulnerable.

While VMaSC and DCV are not considering the degree of vehicles at all during the CH selection process, ECHS [12] selects the vehicle with even density as the CH. ECHS calculates the degree of vehicles in front and back and the vehicle with the equal number of front and back vehicle is selected as the CH. While both DCV and ECHS want to select the vehicle in the center as the CH, their approach is different. DCV chooses the CH based on the distance of the vehicles where ECHS selects the vehicle based on even density counting the degree of vehicles. ECHS also divides the vehicles into three groups based on the speed of the vehicles. ECHS evaluates the average CH duration, the average CM duration, the number of clusters, and clustering overhead; however, the results are not compared with stability-based protocol and the vehicles are not considering their movement direction at the intersection.

2.3.1.2 Data Dissemination/ Routing Algorithms

Clustering-based data dissemination protocol is proposed [8] by improving a non-cluster-based routing protocol. The most reliable vehicle is selected as the CH based on the average relative velocity of the vehicles. The relative velocity, position, and direction are considered during cluster formation to reduce the disconnected problem during low density in highway and the broadcast storm problem during high density in urban area; clustering stability metrics such as CH duration, CM duration, etc. are not evaluated. Therefore, the lifetime of the clusters in this algorithm is questionable. Yang et al. [74] also tries to improve the stability of an existing routing protocol using clustering. They divide the vehicles into multiple zones, the zones are replaced by the clusters and a CH is assigned for each cluster.

Clustering-based routing algorithm is proposed [75] by reducing control overhead considering location, direction, velocity, destination, interest list, and lifetime, however, it suffers frequent number of cluster changes that reduces the stability of the clusters. Prediction-based routing protocol has been proposed [76] for medical vehicle in case of emergency to increase reliability and stability of the clusters. Metrics such as medical vehicle attributes, road condition, and driving environment are considered to predict a route for the emergency vehicles to avoid high traffic.

A passive approach for efficient data dissemination scheme is proposed [77] to reduce overhead and increase stability considering vehicle position and velocity. However, the stability performance is evaluated based on overhead only which is not even a significant parameter for stability. The authors of [77] also proposed a clustering-based data collection scheme [78] to control congestion where only the CH can communicate with RSU. Density, velocity, and direction are considered to select the CH. However, comparison results provided in terms of a mobile network algorithm which is not developed to solve the congestion in VANET, hence, the provided results are not sufficient to evaluate the performance of the scheme.

2.3.1.3 QoS Algorithms

To provide quality of service during RSU failure, a concept of intelligent CH is introduced in density-based scheme [80] where density of the vehicles is considered along with distance and speed to select the intelligent CH. This concept could be used in any RSU-based clustering algorithms during RSU failure; however, experimental result is not presented to justify the improvement of QoS parameters. Regin et al. [81] also proposed a density-based scheme to reduce congestion and increase QoS using trained dataset. Like Taherkhani et al. [10], four states of vehicle are considered where a supplementary CH state is used during cluster formation process which can be compared with the temporary CH state of [10]. If the node density crosses a predefined threshold, the cluster is formed. The most stable and reliable node is chosen as the CH; however, the performance is compared with a very old strategy, rather comparison with some of the new clustering techniques is required to establish the competency of the algorithm. A resource management scheme proposed [83] for cellular-vehicle-to-vehicle and cellular-vehicle-to-infrastructure to improve the performance of cellular users in terms of sum rate, latency and throughput, however, purpose of this scheme is to show that the clustering can improve the performance of a non-clustering algorithm.

Mobility metrics with the QoS metrics such as bandwidth, the degree of the neighborhood, and link quality are considered [68] and the CH is selected based on the suitability of these values. Clusters are divided into two layers: static clustering for V2I communication and dynamic clustering for V2V communication. When the CHs are in the communication range of the RSUs, all vehicles act as the CMs. When no RSU is reachable, a CH acts like a router. Merging of clusters is allowed if they reside within

TR for a certain amount of time. Four parameters are evaluated: CH duration, number of clusters, PDR, and clustering overhead; however, PDR or overhead can be better metrics to measure clustering efficiency rather than the stability of the clusters. Moreover, CM related parameters are not evaluated. Besides, the simulation results presented are only for highway scenario.

Ahmad et al. [84, 90, 91] proposed co-operative clustering for driving assistance for heterogenous LTE network where each vehicle shares the cost of LTE network. Vehicles are inspired to join into cluster and inspired to act as a CH providing reward for participation. The number of vehicles as the CH and participation of vehicles in clusters are evaluated and compared with Ucar et al. [6]; however, not only the increasing number of CH can reduce the share of LTE cost but also decrease the stability of the clusters. Moreover, reliability issues are not considered to select the CH, hence, less reliable vehicles are also inspired to be the CH which can cause security threat to the member vehicles.

2.3.1.4 MAC Algorithms

Clustering-based MAC protocol is proposed [82] [86] for faster delivery of safety messages and for efficient resource management [85]. Shat et al. [82] introduces three new control packets instead of RTS/CTS (request to send/clear to send) packet where Haq et al. [85] tries to reduce the packet conflict due to hidden terminal problem, and Chaurasia [86] tries to solve hidden terminal problem using a reserve channel that can be used by safety messages even during the congestion. The clustering-based MAC protocols are working as evidence that clustering does not solve only the scalability problem but also can be utilized for various purposes in VANET.

2.3.1.5 General-purpose Algorithms

General-purpose clustering algorithm is proposed [88, 92] to provide a more stable and efficient cluster considering velocity, position, direction, and link quality. Double-head cluster is used, so that a CM does not get disconnected from its cluster even it loses connection with the primary CH. Four states of vehicles are used where one vehicle acts as a mirror of the CH and works as the backup CH when a CM loses connection with its primary CH. Four states of vehicles are used where one vehicle acts as the backup CH when a CM loses connection with its primary CH. The relative position in the cluster, relative speed, average signal-to-noise ratio, and average link expiration time are considered to become the CH; however, performance comparison with the recent clustering algorithms need to be presented. Also, how the two CHs handle PDR, delay, and throughput is not evaluated.

A weighted clustering protocol is proposed [87] where CH is selected based on the reputation of the vehicles along with direction, position, velocity, number of nearby vehicles, and lane ID. Reputation of a vehicle is calculated as a number, the vehicle worked as a CH. PDR, number of clusters, and control overhead are evaluated, however, the results are compared with two MANET protocols, no comparison with VANET protocols have been presented.

2.3.2 NEMO Algorithms

Some of the clustering schemes adopted NEMO or mobile-IP (MIP) concept in VANET environment as shown in Table 2.5. NEMO-based clustering techniques mainly developed for faster handoff, i.e., to reduce the total number of handoffs and handoff latency. In these strategies, some authors explicitly mentioned the use of NMEO concept while few authors used the concept of NEMO for clustering without mentioning explicitly.

Table 2.5: Evaluation of NEMO strategies.

Ref.	Protocol	Evaluation
[93]	NEMO	Packet loss, delay
[94]	NEMO	Cost, Timer selection
[95, 96]	NEMO	Handoff latency, PDR, overhead
[97]	NEMO	Cost
[98]	MIP	IP address management
[99]	MIP	PDR, delay
[100]	NEMO	Power, delay

2.3.2.1 Handoff Algorithms

To reduce the number of handoffs as well as handoff latency, clustering strategy is applied [93] for NEMO-based VANET. The MR is considered as the CH of the cluster. The MR and mobile nodes connected with the MR are treated as a cluster. Since CH handles the routing procedure in clustering strategies, vehicles are divided into clusters to minimize the number of handoffs. The vehicles acquire their care-of-address from the CMs of the new cluster prior to actual handoff, hence, latency can be reduced. However, no simulation result is provided comparing the result of the scheme to prove the effectiveness of the scheme in VANET scenario.

To solve the handoff and packet loss problem in high-speed VANET, NEMO based protocols [95, 96] are proposed for highway. The MR acts as the CH and the network is treated as the cluster. In this protocol, the car can acquire IP address from the VANET through a V2V communications to achieve network mobility. To execute the pre-handoff procedure, the vehicle relies on the assistance of the front vehicle to acquire its care-of address, or it may acquire its new IP address through multi-hop relays from the vehicle on the lanes of the same or opposite direction. Hence, it reduces the handoff delay and maintains the connectivity to the Internet; however, comparisons with other clustering protocols are absent.

An IPv6-based mobility management solution has been proposed for vehicular networks [99]. To reduce the mobility handover frequency and the delay, a distributed address configuration algorithm is proposed. A vehicle can establish a routing path to reach the nearest access point and achieve multi-hop communication with the internet through this routing path. Mobility and fixed routing structure for transportation are also considered [97] where mobility is classified into intra mobility and global mobility. Similar concept is used by Ko [94] where a pre-defined timer is used to reduce the number of messages for communication between two vehicles; however, no simulation result is provided that can show the effectiveness of the scheme.

2.3.2.2 Auto-addressing Algorithms

Mobile IPv6-based dynamic auto-addressing protocols have been investigated in the VANET scenario in cluster based addressing scheme (CBAS) [98]. In this MIP-based scheme, incoming vehicles are assigned unique IP address and clustering is used to overcome the problem of maintaining the unique IP address since vehicle communicate through its CH. Vehicles are clustered based on their relative speed and the CH assigns the IP address to its member and ensures that the assigned IP addresses in its vicinity are unique. However, no simulation result is provided that can show the effectiveness of the scheme.

2.3.2.3 Security Algorithms

NEMO-based solution for VANET clustering discussed above are mainly to solve the handoff problem while clustering for NEMO to increase the security for vehicular communication has been reported [100]. The network is treated as the cluster while the MR is called the CH. In this scheme, vehicles are grouped in different clusters to reduce the probability of attack and different clusters can be accessible through their corresponding CH only that works as extra layer protection. However, like other NEMO-based schemes, performances are compared with NEMO-based solutions, no results are provided comparing the scheme with other VANET clustering algorithms.

2.3.3 Summary of Mobility-based Strategies

Most of the vehicle mobility-based solutions are proposed to increase the stability of the clusters, because even an efficient clustering algorithm can perform worse in the high mobile environment. Maintaining stability of the clusters is given priority in mobility-based strategies. Some mobility-based strategies also serve data dissemination, MAC, QoS, etc. Mobility-based strategies have been evaluated mostly using the average CH duration, the average CM duration, and the average number of CH change. NEMO-based solutions either refrained from providing simulation result or presented results comparing with NEMO basic support protocol or other NEMO-based protocol; however, none of them compared their performances with other clustering algorithms. Hence, the suitability of NEMO-based clustering protocol for VANET environment is still uninvestigated.

2.4 Multi-hop-based Strategies

Multi-hop strategies are also mobility-based strategies but work in a multi-hop manner. Generally, a cluster of vehicles means one-hop cluster where a CH can reach all its CMs directly, because CMs are within the range of the CH; however, some clustering algorithms are based on multi-hop strategy. When a vehicle cannot reach the CH of a cluster directly but can reach a member of the cluster, then the new vehicle joins to the cluster through a CM. Hence, a CH can cover CMs in a multi-hop manner which is termed multi-hop clustering, or *N*-hop clustering, or *k*-hop clustering. The value of *N* or *k* depends on the number of hops. In the Figure 2.4, 2-hop CM cannot reach the CH but can reach a CM of the CH. As a result, a 2-hop CM joins the cluster through a CM creating a multi-hop clustering.

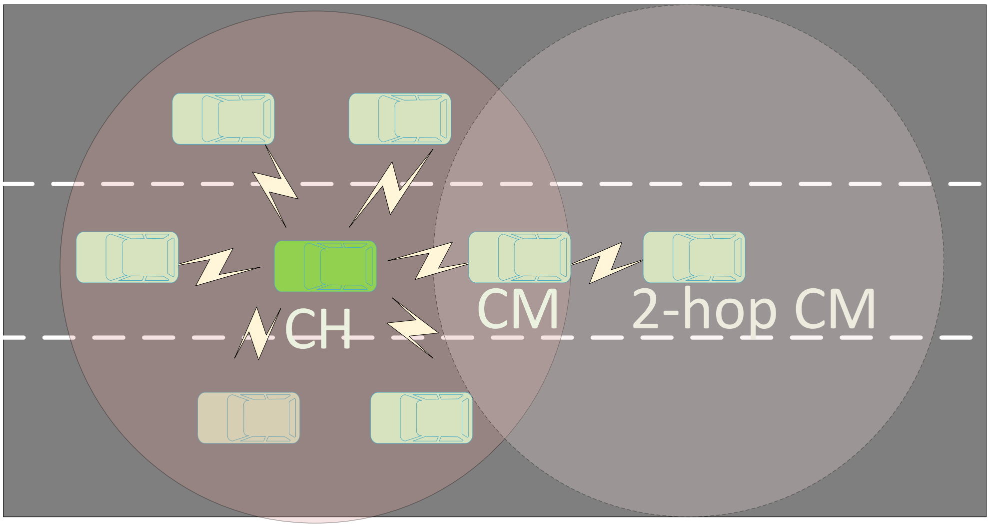

Figure 2.4: Multi-hop clustering concept in VANET.

Since scalability is the main challenge for VANET, some clustering concepts are published in the literature based on multi-hop transmission of the packet to reduce the number of clusters. Hence, a CH can cover a larger area. Our work is clearly *different* in that we have evaluated multi-hop protocols in detail. In subsection 2.4.1, 2-hop-based protocols are presented and in subsection 2.4.2, the protocols which are evaluated for more than two hops are presented.

2.4.1 2-hop Protocols

In 2-hop communication, the CH can reach up to 2-hops of vehicle for its coverage. In Table 2.6, the column 'Number of hops' means the number of hops evaluated in the algorithm.

Table 2.6: Evaluation of 2-hop strategies.

Ref.	**Number of hops**	**Evaluation**
[101]	2-hop	PDR, routing overhead
[102]	2-hop	No. of clusters, average lifetime, cluster size, CH/CM changing time
[103]	2-hop	No. of clusters, cluster change
[104]	2-hop	No. of CH, cluster change, time spent in cluster
[79]	2-hop	Success rate, dissemination efficiency

Stability is the main concern for Tian et al. [101]. They consider the position information as well as the moving direction of the vehicles to divide the vehicles into clusters. Each vehicle broadcasts *Hello* message including its latitude, longitude, direction and time to the entire network. The receiving vehicle will first check the *Hello*'s hop count value and if the number of hops is larger than the threshold value, it will discard the *Hello* message. The sender vehicle updates its routing table by calculating the distance with the vehicle. The cluster heads are chosen based on the stability of the vehicles; however, simulation results are provided comparing with AODV in terms of routing overhead which is a routing protocol.

In the absence of GPS or without knowing the car's location, multi-hop hierarchical clustering algorithm is proposed [103] [104] to connect the vehicle into a two-hop cluster in the shortest possible time considering time and space complexity; compromising the quality of the cluster. The center vehicle is chosen as the CH. In these multi-hop schemes, if a CH loses its members and is within the range of a CG of another CH, the single CH merges with the CH of the larger cluster. Additionally, if a CH arrives in the TR of another CH and the first CH has a shorter distance to the CMs compared to the second CH, both clusters will merge with the first CH as the new CH and the second CH as the CG. If necessary, the cluster can be optimized in the maintenance phase after creating the initial cluster; however, the simulation results are provided comparing the data with a mobile ad hoc network clustering technique in terms of number of clusters and number of cluster change, where time spent in cluster is also evaluated. No significant clustering stability or clustering efficiency-related parameters are evaluated.

A clustering-based routing protocol is proposed [105] considering vehicle position and moving direction. Each vehicle broadcasts beacon message including its latitude, longitude, and direction. The receiving vehicle will first check the number of hop count value of the received message, if the number of hops is larger than a threshold value, it will discard the message. Upon receiving the acknowledgment, sender vehicle updates its routing table by calculating the Euclidean distance of the vehicles and the vehicles belong to its closest CH. Simulation results are provided for PDR, routing overhead etc.; however, end-to-end delay, throughput etc. are not evaluated which were necessary to measure clustering efficiency. In multi-hop communications, packets need to travel longer distances compared to single-hop clustering algorithms; hence, in multi-hop communications end-to-end delay increases while throughput decreases.

Network criticality is used as the metric in a robust multi-hop-based algorithm, presented [102]. In this criticality-based clustering technique, the robustness of an undirected network graph to the change of the environment, such as the destination change or topology change, is termed as network criticality and interpreted as an electrical circuit where vehicles show resistant to any change in the environment. Some parameters like number of clusters, average lifetime, cluster size etc. have been evaluated; however, the important parameters like duration of vehicles spent as CH, duration of vehicles spent as CM, and number of state change per vehicle are not evaluated. To solve the congestion problem, a multi-hop-based data dissemination protocol is proposed [79] by minimizing the overhead messages; however, sufficient results are not provided that can prove the stability of the protocol.

2.4.2 *N*-hop Protocols

Many multi-hop-based protocols provided results where CH can reach two or more hops, such as 3-hop, 4-hop, or 5-hop coverage. Table 2.7 represents *N*-hop protocols where the column 'Number of hops' means the number of hops evaluated by the protocols.

Vehicles are allowed to broadcast regular beacon message periodically and calculate the relative mobility based upon two consecutive beacon messages received from the same node [106]. Each vehicle calculates the aggregate mobility value, which is the sum of relative mobility values and the weight value for all the neighboring nodes in *N*-hops. The vehicle then broadcast their aggregate mobility value in the *N*-hop neighborhood and the vehicle with the smallest aggregate mobility value is selected as the CH. If a vehicle receives multiple beacon messages, it selects the CH which is the closest in terms of hop count. The vehicle with the lowest relative mobility is selected as the CH when more than one CH candidates have the same hop count. Average CH duration, average CM duration, and the number of CH change have been evaluated, however, the number of state change or number of vehicles in the clustered state are not evaluated which are also important parameters for clustering stability.

Table 2.7: Evaluation of *N*-hop strategies.

Ref.	**Number of hops**	**Evaluation**
[106]	2-hop, 3-hop, 5-hop	CH/CM duration, no. of CH change
[107]	3-hop	CH/CM duration, no. of CH change, no. of clusters, overhead
[108]	3-hop	CH/CM duration, CH change, overhead
[109]	2+ hop	Overhead, no. of packet/time for cluster selection
[110]	2-hop, 3-hop, 4-hop	Overhead, cost, hit probability
[111]	2-hop, 3-hop	CH/CM duration, CH change
[6]	2+ hop	CH/CM duration, CH change rate, no. of unclustered vehicle, overhead

A distributed multi-hop clustering algorithm for VANETs based on neighborhood follow is proposed by Chen et al. [107], where relative mobility is given preference. It considers the relationship among the vehicles within the neighborhood to choose the CH. Due to high mobility vehicles cannot identify the vehicles in its multi-hop neighbors, therefore, they consider the vehicle in one hop as a single cluster. CM chooses its CH based on the stability of the vehicles and their history of the movement which is denoted as neighborhood follow relationship. All the CMs follow the CH. They do not use location service, rather depend on the topology. Performance is measured in terms of CH duration, CM duration, number of CH change, number of clusters, and overhead. The movement of the movement of the vehicles at the intersection is not considered.

When the relative speed of the CH changes, it causes frequent CH change. To increase the routing performance by reducing the number of CH change, a novel passive multi-hop clustering algorithm is proposed [108] based on a previous work [101]. The number of candidates to become the CH in a multi-hop scenario is more than a single-hop clustering and the most stable vehicle becomes the CH. In a multi-hop clustering, the CH can have *N*-hop coverage compared to single-hop clustering and can achieve more stability and reliability. Simulation results are compared with VMaSC and two other muti-hop clustering protocols for CH duration, CM duration, number of CH change, and overhead. An RSU assisted multi-hop scheme is proposed [109] based on a previous work [101]. A new vehicle broadcasts *Hello* packets to all its neighbors with its position, speed, and direction. The neighbors reply with another *Hello* packet that increases the number of packet dissemination and number of packet loss. To solve this problem, the new node communicates with the RSU to receive information about the stability of the clusters and can join into the cluster in relatively faster time, however, important metrics are not evaluated.

A novel *k*-hop clustering approach is presented by Zhang et al. [110]. To select CH, it measures the highest connectivity, vehicle mobility, and host ID. Max–min k-hop heuristic approach is modified for cluster formation by considering highest connectivity in terms of signal strength and vehicle mobility. The scheme can dynamically adjust the period of announcement of location information according to vehicle velocity to reduce overhead. A distance-based converge-cast is deployed to collect all memberships within the cluster, including the members located on the cluster border. To enhance clustering stability, vehicle activation and deactivation are used by considering the radio link expiration time and the number of vehicles connected to a CH. To increase stability, it considers the inter-cluster link expiration time, however, no important clustering metrics are evaluated.

Vehicular multi-hop algorithm for stable clustering was proposed [6] [111] by Ucar et al. Cellular technologies have been used in conjunction with IEEE 802.11p to reduce the cost of communications between vehicles and base stations as well as the number of handoffs. Average relative speed is measured among the neighbors of the *N*-hop to create the clusters and the vehicle with the lowest mobility wins to become the CH. A new vehicle adds to the neighbor CH or CM in a multi-hop manner instead of connecting with the CH directly. Merging is allowed in this scheme when two CHs overlap for a certain amount of time. In a multi-hop communication, the CH acts as a dual-interface node where CH communicates with CMs via IEEE 802.11p interface and connects the cluster to the cellular network via the LTE interface; however, simulation results are not impressive for V2V communication.

VMaSC used the relative speed as the basis for clustering. Average relative speed (*AVG_REL_SPEED*) calculated for each vehicle (only the same direction vehicles are considered) as following:

$$AVG\_REL\_SPEED = \frac{\sum_{j=1}^{N(i)} \left| Speed_i - Speed_{ij} \right|}{N(i)} \tag{2.1}$$

Here, N is the number of cars within the coverage of TR of k-hop distance, $Speed_{ij}$ is the speed of jth number of vehicles. The vehicle with minimum AVG_REL_SPEED wins the race to become the CH for the cluster.

CH selection and cluster creation in VMaSC:

1. Calculate the average relative velocity for all the vehicles.
2. Select the vehicle with the lowest average relative velocity as the CH.
3. Create a cluster based on the TR of the CH with the vehicles who are running in the same direction of the CH.
4. For the rest of the vehicles, REPEAT STEP 1 to 3.

In the absence of RSU, a part of the proposed scheme would not work. The results are specific to a particular set of parameters under which the simulation is carried out. Also, CM-related metrics are not optimized, and intersections are not considered.

2.4.3 Analysis on Multi-hop-based Strategies

Multi-hop-based strategies cover a larger area compared to single hop clustering protocols and may create a lower number of clusters for the equal number of vehicles. On the other hand, clustering overhead increases in the multi-hop-based protocols because of increased number of message transmission which can affect the stability of the clusters.

2.5 Challenges in VANET Clustering

The most important parts in VANET clustering are the CH selection and cluster formation. Efficiency of clusters largely depends on the cluster formation process where stability of the clusters mainly depends on the CH selection process and then on the cluster formation process. The efficiency of the clusters is evaluated in terms of packet loss and end-to-end delay more frequently while the stability of the clusters is evaluated based on the average CH change rate, average CH duration, average CM duration, and clustering overhead.

Machine learning and fuzzy logic algorithms have been evaluated in terms of packet loss and end-to-end delay more frequently while mobility-based clustering protocols, including multi-hop-based protocols, are evaluated based on the average CH duration, the average CM duration, and the number of CH change. It can be concluded that intelligent based algorithms are concentrating more on efficient packet delivery while mobility strategies emphasize stability of the clusters.

To create a lower number of clusters, multi-hop-based protocols have been proposed. CH can get a larger coverage and can reduce the number of clusters and number of CHs. Some multi-hop protocols are evaluated for only 2-hop distance, some of the approaches evaluate for 3-hop, 4-hop, and 5-hop also. The challenge for the multi-hop approaches is extra overhead and complex routing due to its larger cluster size. CH change process is also complex which affects cluster stability.

Mobility-based clustering approaches are the most common technique for clustering in VANET where mobility parameters are important. Although vehicle mobility is mainly used in mobility-based schemes, some research works performed clustering based on NEMO, considering the similarities between clustering concept in VANET with NEMO. However, NEMO can be more suitable for cellular architecture than V2V communication.

CH selection and cluster formation process are important for cluster stability in VANET. The existing VANET clustering protocols can partially provide stability. For example, VMaSC is optimized for CH-related metrics but CM-related metrics are not optimized. On the other hand, DCV is optimized for only CM-related metrics but not optimized for CH-related metrics. Therefore, a stable clustering protocol is *absent* in the literature that can optimize both CH-related and CM-related metrics.

Moreover, the existing algorithms are kind of effective in straight road without any intersection. As a result, clusters of the existing algorithms break down at the intersection of the roads. Therefore, a clustering protocol for VANET is also *absent* in the literature which considers the intersections of the road properly during the CH selection and effective for highway and can provide stability regardless of the topology of the roads.

2.6 Summary

Detailed analysis on VANET clustering strategies is presented in this chapter with an intensive discussion on intelligent-based strategies, mobility-based strategies, and multi-hop-based strategies. Mobility clustering protocols are more frequently used for VANET clustering and the most important issue in VANET clustering is stability.

In Chapter 3, we presented our first VANET clustering protocol that considers multiple mobility parameters to provide stability to the clusters in terms of both CH and CM. Our second clustering protocol is presented in Chapter 4, which can provide stability in both highway and intersection, regardless of topology of the roads.

Chapter 3

Junction-based Clustering Protocol

The existing clustering protocols for VANET are optimized for either cluster head or cluster member. To achieve optimized performance for both CH-related and CM-related metrics, a new robust and dynamic mobility-based clustering protocol is presented in this chapter. In the proposed junction-based clustering for VANET (JCV), transmission range, moving direction of the vehicle at the next junction, and vehicle density are considered in the creation of a cluster, whereas relative position, distance, movement at the junction, degree of the vehicles, and time spent on the road are considered to select the cluster head.

3.1 Introduction

Previous VANET clustering protocols used the relative speed or the relative position of the vehicle as the basis of the clustering process. The existing algorithms can partially provide cluster stability, for instance, VMaSC is optimized for CH-related metrics but the metrics related to CM are not optimized. On the other hand, DCV is optimized for only CM-related metrics but not optimized for CH-related metrics. Therefore, an effective stable clustering protocol is absent in the literature that can optimize both CH-related and CM-related metrics.

The stability of the clusters does not depend on a single parameter. To increase cluster stability, the stability of the CH, and the stability of the CMs are both important.

Mobility parameters, such as relative position, relative velocity, moving direction, and vehicle density, etc., are used to form clusters. Stability metrics, such as the average CH duration, the average number of CH change, the average CM duration, and clustering overhead, etc., are used to evaluate cluster performance. In VMaSC, CH-related metrics such as the average duration of CH, the average number of CH change, etc., performed well but CM-related metrics are not optimized and need improvement. On the other hand, CM-related metrics such as the average CM duration, the average number of CMs, etc., are improved in DCV at the expense of the performance of CH-related metrics. Therefore, an intelligent approach of VANET clustering is needed where both CH and CM-related metrics are improved to optimize the stability of the clusters.

Our objective in this chapter is to present a VANET clustering protocol to achieve better cluster stability through consideration of appropriate mobility parameters. Our aim is to improve the performance of both the CH and the CM-related metrics. In JCV, transmission range, vehicle position, and degree of a node are considered along with the route of the vehicle at the next intersection. The advantage of JCV is the fewer number of vehicle states, which will result in simplified state transition. The difference of JCV with the previous works is, a vehicle will join into a cluster only when joining a cluster is beneficial; otherwise, a vehicle will remain in the un-clustered state. Hence, JCV will have a longer cluster-life.

CH-related parameters, such as the average CH duration, the average number of CH change, the average number of CH as well as CM-related parameters such as the average CM duration, the average number of CM, the ratio of CMs, etc., are evaluated.

The number and duration of the non-clustered vehicles are also evaluated. The evaluation setup and simulation results are provided in Chapter 5 and 6.

3.2 Architecture of JCV

Figure 3.1 shows the architecture of JCV. JCV uses OBU for wireless communication and a GPS to access the geographical location. Three states of the cars are used: Entry state (EN), Cluster Head (CH), and Cluster Member (CM).

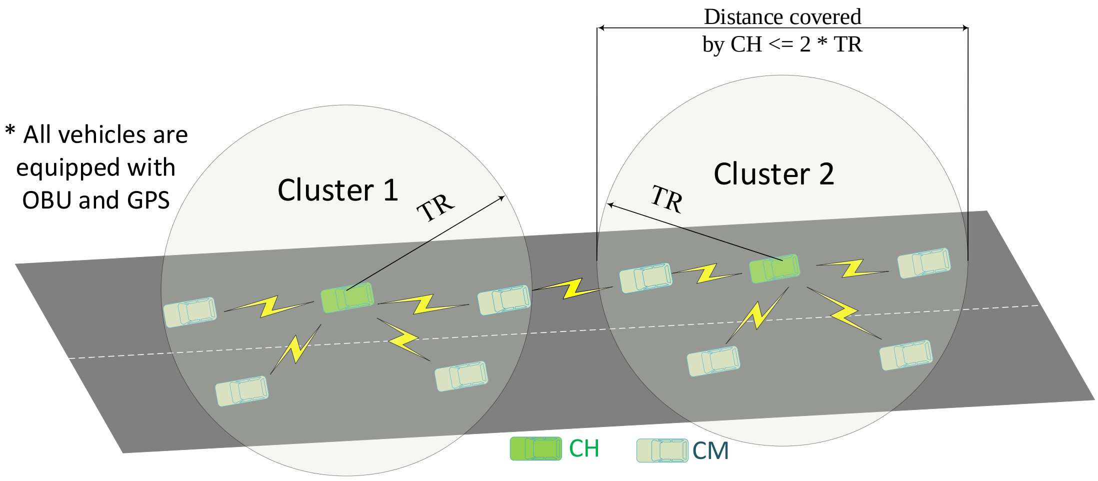

Figure 3.1: Architecture of JCV.

CH is the head of a cluster, CM is a member of a cluster, and EN is the initial entry state of a vehicle. A vehicle can access its neighbor vehicles within its transmission range (TR). Hence the TR of the CH of a cluster limits the number of vehicles.

Table 3.1 represents the notations used in JCV.

Table 3.1: Notation used in JCV with description.

NOTATION	DESCRIPTION
CH	CLUSTER HEAD
CM	CLUSTER MEMBER
EN	ENTRY STATE
TR	TRANSMISSION RANGE
CH_{IJ}	CH OF NEWLY CREATED CLUSTER AFTER MERGING
CH_TIMER	CLUSTER HEAD STATE TIMER
CM_TIMER	CLUSTER MEMBER STATE TIMER
EN_TIMER	ENTRY STATE TIMER
VEH_INFO	VEHICLE INFO
EN_REQ	MESSAGE FROM EN
CH_RESP	RESPONSE FROM A CH
VEH_ADV	PERIODIC VEHICLE ADVERTISE MESSAGE
REL_DIST	RELATIVE DISTANCE
AVG_REL_DIST	AVERAGE RELATIVE DISTANCE
ACK_MERGE	ACKNOWLEDGEMENT MESSAGE AFTER MERGING
VEH_RANK	RANKING OF A VEHICLE
STATUS	STATUS OF A VEHICLE

Now we will describe the basic process of clustering. Let G (V, E) be the graph representing the VANET, where V is the set of vehicles and E is the set of V2V communication links. To form clusters, we can define a set of rules based on the distance between vehicles:

- Each vehicle $v \in$ V broadcasts regular *Hello* message to its neighbors within its TR.
- Each vehicle $v \in$ V selects the CH based on CH selection criteria.
- A vehicle $v \in$ V joins the cluster of its CH if it is within a distance of TR.
- The CH periodically broadcasts a message to its cluster members to keep them alive.
- If a vehicle $v \in$ V does not receive a message from its CH within a certain time, it leaves the cluster and repeats the process.

JCV is a single-hop clustering protocol that aims to achieve better clustering stability for both CH and CM. We have simplified the state transitions and reduced the number of states for vehicles and used a greater number of parameters in JCV. JCV considers the vehicles' movement at the junctions during the CH selection process to reduce the number of CH change. It uses the position of vehicles to calculate the distances of the vehicles. JCV also considers the degree of vehicles to reduce the number of clusters. By fulfilling other conditions, the vehicle with the lowest relative distance is selected as the CH. The parameters include the position of the vehicles, distance, moving direction, the movement of the vehicles at the intersection, transmission range, and degree. In JCV, vehicles are not forced to join in any cluster. They have freedom not to join in the clusters if the vehicle considers its joining is not beneficial. By considering these parameters, we aim to improve the stability of JCV for both CH and CM.

3.3 State of vehicles in JCV

The number of states used in VMaSC is five and the number of state transitions is nine. In DCV, the number of states is four and the number of state transitions is eight. JCV simplifies the number of states to three and the number of state transitions to six. Figure 3.2 shows the number of states and state transitions for VMaSC, and Figure 3.3 shows the number of state and state transitions for DCV. Figure 3.4 illustrates the number of state and state transitions of a vehicle in JCV.

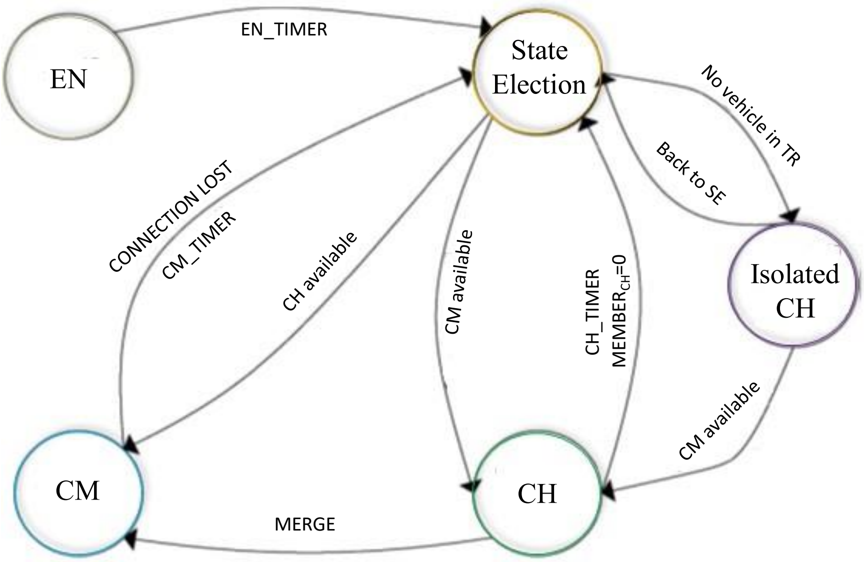

Figure 3.2: State transition diagram of VMaSC.

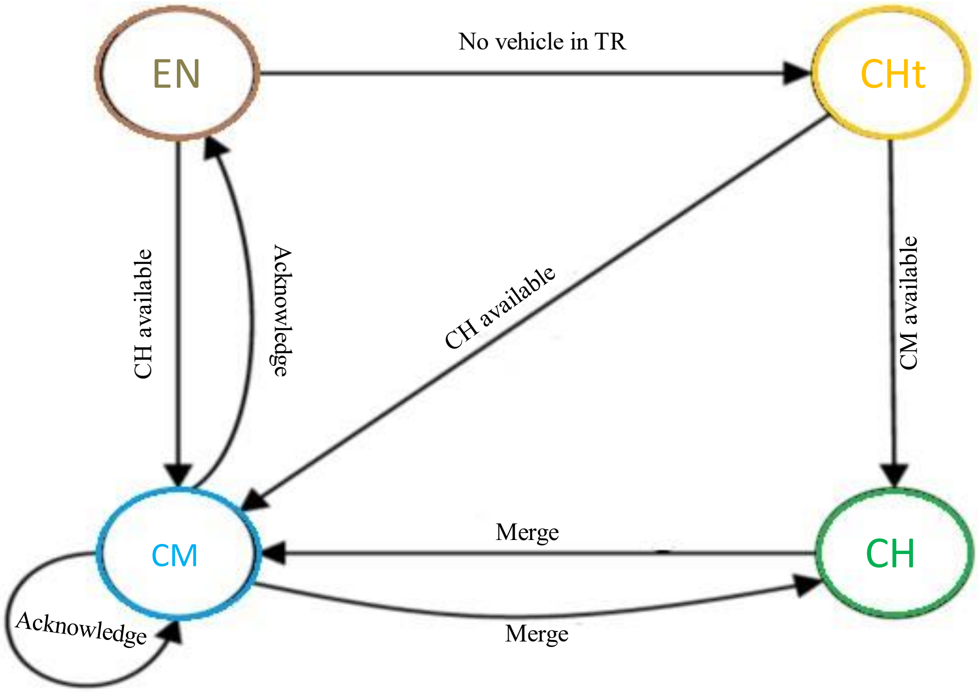

Figure 3.3: State transition diagram of DCV.

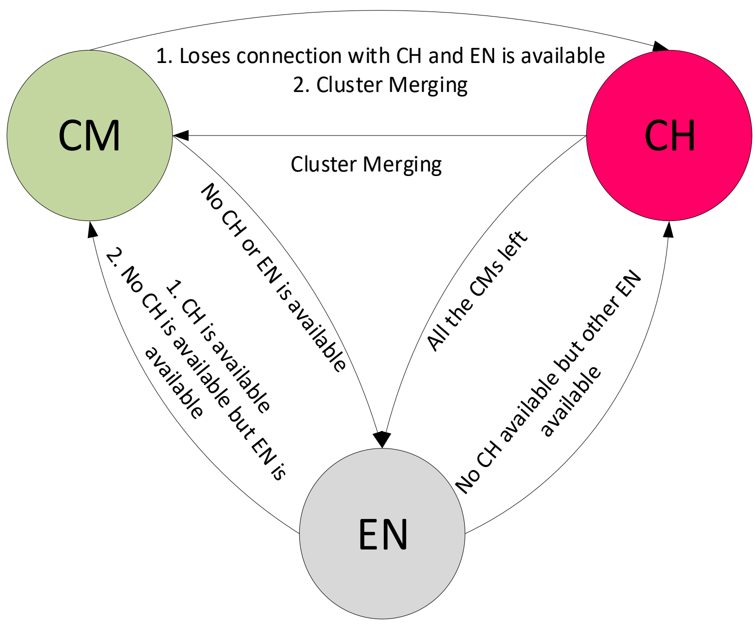

Figure 3.4: State transition diagram of JCV.

3.4 State transition in JCV

JCV uses three states and six state transitions while VMaSC uses five states and DCV uses four states. In JCV, the vehicle remains in one of the three states: CH, CM and EN.

- EN: EN is the initial state when it does not belong to any cluster. When a vehicle enters the road for the first time it will collect information of other vehicles within its *TR*. At this state, it waits to join an existing cluster. If there is no CH available, it will remain as EN. If a vehicle leaves a cluster, it will be in the state of EN again. At any given time, a vehicle in EN state signifies the absence of any vehicle or CH within its *TR*. It can also be thought of as belonging to a cluster of one vehicle.
- CH: CH is the leader of a cluster. Based on the CH selection criteria, a vehicle is selected as the CH of a cluster.
- CM: If a vehicle becomes a member of any existing cluster, the state of the vehicle is CM.

Vehicles advertise a *VEH_ADV* message periodically within its *TR* to inform the presence on the road. *VEH_INFO* is used to store information about the vehicle itself and its neighbor vehicles. Algorithm to join a cluster for a newly arrived vehicle in JCV is given below:

==

FORALL *Vehicle* Є *VEH_INFO* DO

IF *REL_DIST* < *TR* THEN

Send *EN_REQ*;

IF *CH_RESP* received THEN

STATUS == CM ;

Exit;

ENDIF

ENDFOR

IF *Timer* = = 0 && *STATUS* = = *EN* THEN

Broadcast *VEH_ADV*;

Exit;

ENDIF

==

EN: EN can convert to a CH or CM.

- EN → CH: If there is no CH available but at least one EN is available within the *TR* of the EN and the EN is a better candidate for CH, then a cluster will be created and the EN will be the CH of the new cluster.
- EN → CM: The vehicle enters from EN to CM state if it finds a CH within the *TR* (*REL_DIST* < *TR*) and receives an acknowledgment message (*CH_RESP*) from a CH of an existing cluster in response to *EN_REQ*. Secondly, if there is no CH available but at least one EN is available within the *TR* of the EN, and

the other EN is a better candidate to become the CH, then the EN will become a CM of the new cluster.

CH: A CH can transform to CM or EN.

- CH → CM: If two CHs come within the *TR* of each other and decide to merge into one cluster, CH starts *CH_TIMER* and starts communication, then one or both CH changes their *STATUS* into CM depending on the selection of the new CH. The same is applied if any EN comes within the *TR*. Due to the dynamicity of the vehicles, if a CM becomes a better candidate to be the CH, the present CH downgrades its *STATUS* to CM.
- CH → EN: If all the CMs leave a cluster, the CH will become an EN.

CM: A CM can transform to CH or EN.

- CM → CH: A CM can become a CH if the CM loses connection with its CH and some vehicles are present within TR and this CM is the best candidate to become the CH. Secondly, during the cluster merging process, a CM can be elected as the CH of the new cluster if the CM is the best candidate compared to existing CHs and the rest of the vehicles. CM starts *CM_TIMER* and starts the process. Thirdly, a CM can become a CH of its cluster if it becomes the best candidate than the rest of the vehicles due to vehicles' movement.
- CM → EN: A CM changes its *STATUS* to EN if the CM loses connection with its CH and no other vehicle is available within its *TR*.

3.5 Cluster formation in JCV

Figure 3.5 illustrates the cluster formation process in JCV. Upon entering the road, the state of a vehicle is EN, and the EN checks for neighboring vehicles within its *TR*. If any CH is available, the new vehicle will consider joining the cluster. The EN sets *EN_TIMER* and sends a join request (*EN_REQ*) to the CH. Upon receiving the *EN_REQ* message from EN, the CH sends a *CH_RESP* message to EN. Upon receiving a *CH_RESP* message, the EN joins the cluster and changes its state to CM. If no response is received from a CH before *EN_TIMER* expires, the EN remains as EN, and the process is continued.

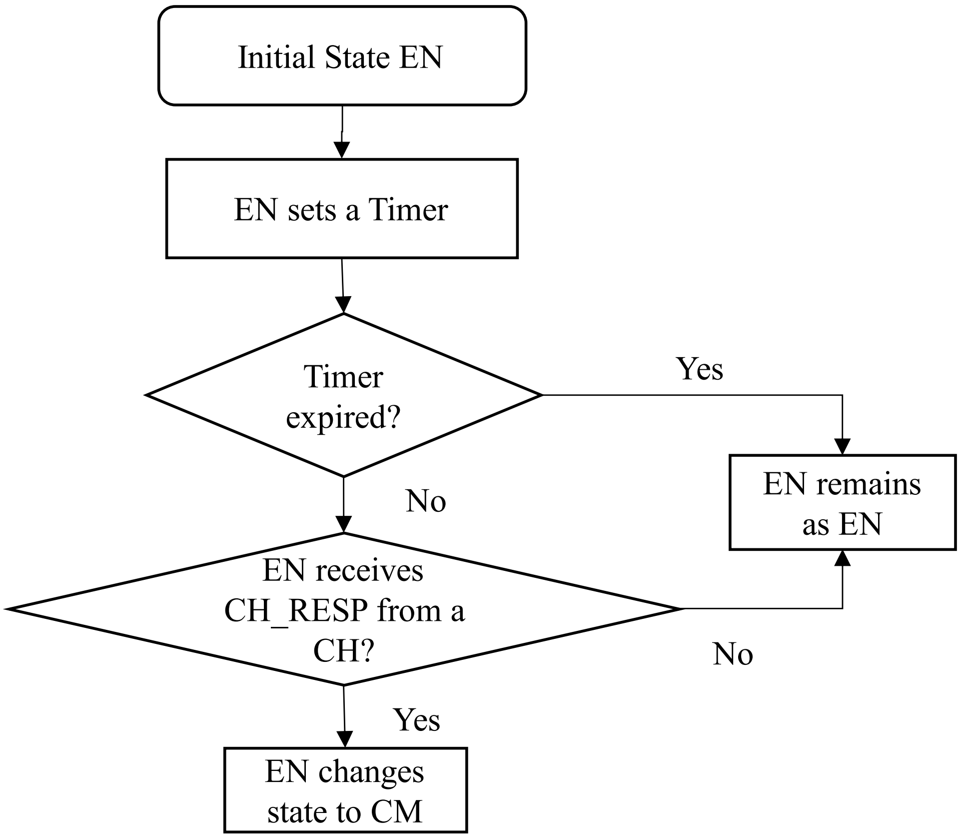

Figure 3.5: Cluster formation process in JCV.

JCV uses an intelligent approach to create clusters. Traditionally, the vehicles which move in the same direction are considered to form a cluster to prevent the joining of the vehicles coming from the opposite direction; however, in a city environment where junctions and intersections are quite common, the clusters are broken when the vehicles change their direction in a junction or intersection. To increase the stability of the cluster, JCV considers the movement of the vehicles at the next junction in the formation process of a cluster. The vehicles that will move in the same direction after the next junction join the same cluster. The algorithm prevents the vehicle which will move in a different direction in the next junction, hence, the vehicles joining in a cluster are ensured to move together at least two junctions. In this way, the lifetime of the clusters increases, and the stability of the cluster increases significantly.

When a vehicle joins in a cluster, the CH updates its member list adding the newly joined CM. A similar approach is applied when two EN comes within *TR* of each other. After exchanging request and response messages, two EN create a cluster where one EN becomes the CH, and the other becomes a CM.

3.6 CH selection

Each vehicle exchanges *VEH_ADV* message with all the vehicles within its *TR*. Each vehicle within the *TR* who will move in the same direction at the next junction becomes ready to join in a single cluster. Each vehicle is given a score based on their relative distance to assign a rank (*VEH_RANK*) to each vehicle, i.e., when *REL_DIST* < *TR*, the relative distance of each vehicle is measured. From the relative distances, the average relative distance (*AVG_REL_DIST*) is calculated for each vehicle. The vehicle with the lowest *AVG_REL_DIST* gets the highest score among the vehicles. Based on the score, the vehicles are ranked and the vehicle with the highest rank is selected as the CH. If two vehicles obtain the same rank, then the degree of a node is considered. If the degree of the vehicles is also the same, then the vehicle which enters the road first and spends more time on the road becomes the CH.

$AVG\_REL\_DIST_i$ for each vehicle is calculated as below:

$$AVG\_REL\_DIST_i = \frac{\sum_{j=1}^{N} \left| REL\_DIST_{ij} \right|}{N} \tag{3.1}$$

Here, N is the number of vehicles within the coverage of *TR*. REL_DIST_{ij} is the distance between $Vehicle_i$ and $Vehicle_j$. The vehicle with minimum *AVG_REL_DIST* wins the race to become the CH for the cluster.

3.7 Cluster merging in JCV

Cluster merging is possible in JCV; however, the mechanism of JCV keeps the requirement of merging of two clusters at a minimum level. Since the vehicles which will move in the same direction after the next junction are joining in the same cluster and when a new vehicle comes within the *TR* of the CH is joining in the cluster, as a result, the possibility of coming two CHs within the *TR* of each other in a city scenario where junctions are very common is rare. However, if two CHs come within the *TR* of each other, cluster merging process will be triggered. This process is like the joining of EN in an existing cluster; hence the algorithm avoids extra message transmission and makes the transition simple and efficient. The algorithm represents the clustering merging process in JCV is given below:

==

IF CH_i receives request for merge from CH_j

CH_i estimates the length of the merged cluster

IF the *length* $\leq 2 * TR$ THEN

CH_i and CH_j selects a *Vehicle* as the new CH, CH_{ij}

CH_i and CH_j sends *ACK_merge* to CH_{ij}

$CH_i \rightarrow CM_{ij}$

$CH_j \rightarrow CM_{ij}$

ENDIF

ENDIF

IF CM_i or CM_j is selected as the new CH (CH_{ij}) and receives *ACK_merge*

$CM_i \rightarrow CH_{ij}$ or $CM_j \rightarrow CH_{ij}$

$CM_{ij} \leftarrow CM_i$, $CM_{ij} \leftarrow CM_j$

CH_{ij} broadcasts *HELLO* message to all CM_{ij}

ENDIF

==

3.8 Leaving a cluster in JCV

If a vehicle leaves a cluster, the CH updates its member list accordingly. If a CM wants to leave a cluster, it sends a message to the CH. Upon receiving the message, the CH deletes its entry for the CM and sends an acknowledge message to the CM. The algorithm for cluster leaving is given below:

==

IF *REL_DIST* $\leq$ *TR* AND *CH* receives regular message from *CM* THEN

CH updates the information of *CM*

ENDIF

IF message gap exceeds waiting period THEN

CH deletes *CM*

ENDIF

==

In most cases the leaving of a CM happens when a CM reaches out of the *TR* of the CH. If a CH does not hear anything from a CM in a certain period, CH considers that the CM has left the cluster and removes the CM entry from its member list. The CH also advertises this information to all of its CMs.

3.9 Routing Structure in JCV

Clustering in VANET has its unique routing structure. Without clustering, vehicles do not have any hierarchy in VANET. Clustering creates a hierarchical structure. The routing structure in JCV is shown in Table 3.2.

Table 3.2: Routing table structure in JCV.

Vehicle ID (V_i)	CH	CM

V_i is the vehicle identification number. CH is the vehicle ID of the cluster head and CM is the list of CMs in the cluster.

3.10 Comparison of JCV with VMaSC and DCV

VMaSC selects the vehicle with the lowest average relative velocity as the CH. DCV selects the vehicle that is in geographically center position as the CH. JCV gives precedence to the distance of the vehicles over the velocity; however, JCV considers none of these parameters are sufficient for clustering stability. JCV considers these parameters along with the movement of the vehicles at the intersection and the degree of vehicles to increase stability.

DCV has a lower *TR* than the actual transmission range while VMaSC and JCV used the actual transmission range as the *TR*. All three algorithms are considering the vehicles are running at the same direction, i.e., the vehicles are running at the opposite direction are not considered to join in the same cluster. If we allow the vehicles from the opposite direction to join in a cluster, then we may allow to create lower number of clusters at a given time at the expense of frequent cluster-break which we want to avoid.

VMaSC and DCV are not considering the degree of vehicles as the parameter to select the CH or creating the cluster. JCV is considering the degree of vehicle to minimize the number of clusters.

VMaSC is a multi-hop clustering protocol which allows vehicles to join in a cluster through another CM where DCV and JCV are single-hop clustering protocol which allows the vehicles to join in a cluster that are within the transmission range of the CH. However, we have implemented only single-hop VMaSC for simulation purposes.

VMaSC focuses on optimization of CH-related metrics and DCV focuses on CM-related metrics where JCV optimized both CH and CM-related metrics.

The number of states in VMaSC, DCV and JCV are five, four and three, respectively. The number of state transitions are nine, eight and six in VMaSC, DCV and JCV, respectively. A comparison of VMaSC, DCV and JCV is presented in Table 3.3.

Table 3.3: Comparison of JCV with VMaSC and DCV.

Criteria	**VMaSC**	**DCV**	**JCV**
Hop Count	Multi-hop	Single hop	Single hop
Transmission Range used for the CH	Same as the Transmission Range (TR = Actual TR)	Lower than the actual Transmission Range (TR < Actual TR)	Same as the Transmission Range (TR = Actual TR)

Known to each vehicle of clusters	Position Velocity Acceleration Direction TR Degree of vehicle	Position Velocity Acceleration Direction TR	Position Velocity Acceleration Direction TR Degree of vehicle Direction at the next intersection
Parameters used for CH Selection	Position Direction TR Velocity	Position Direction TR (< TR) Distance	Position Direction TR Distance Direction at the next intersection Degree of vehicle
Emphasized on performance parameters	CH-related metrics	CM-related metrics	Both CH and CM-related metrics

Number of states of the vehicles	5	4	3
Number of State transitions	9	8	6
Direction of the vehicles	CH and CMs should be running at the same direction	CH and CMs should be running at the same direction	CH and CMs should be running at the same direction
Degree of vehicle for CH selection	Not used	Not used	Used

3.11 Summary

In this chapter, our first clustering protocol (JCV) has been presented. JCV uses a greater number of parameters in its clustering formation process and the CH selection process. JCV simplifies the number of states and state transition process. JCV aimed to achieve optimum results for all kinds of metrics. The evaluation and the results are presented in Chapter 5 and 6. In the next chapter, we will present our next VANET clustering protocol.

Chapter 4

Stable Dynamic feedback-based Predictive Clustering (SDPC) Protocol

In this chapter, we present our second clustering protocol, SDPC. We have already discussed that scalability is a major issue for vehicular communication, especially when the number of vehicles increases at any given point. VANET clustering can solve the scalability issue but suffer stability issue. Previously proposed clustering algorithms for VANET are optimized for either straight road or for intersection. Most of them are optimized for only straight-roads, i.e., highways. Moreover, the absence of the intelligent use of a combination of the mobility parameters, such as direction, position, velocity, acceleration, distance, degree of vehicle, and movement at the intersection results in cluster stability issues. A dynamic clustering algorithm considering the efficient use of all the mobility parameters can solve the stability problem in VANET regardless of road topology. To achieve higher stability for VANET, a novel stable dynamic feedback-based predictive clustering protocol (SDFPC) for VANET is proposed in this chapter. From the mobility parameters the future road scenario is constructed. The cluster is created, and the cluster head is selected based on the future construction of the road. The evaluation setup and comparison results of SDPC have been presented in Chapter 5 and 6.

4.1 Introduction

In VANET clustering, vehicles are divided into several groups based on the similarity features of the vehicles such as velocity, position, distance, degree of vehicles etc. The vehicles are highly moving objects and change their direction frequently while transmission range is limited. As a result, clusters break down frequently.

In the literature, many clustering algorithms have been proposed to minimize breakdown of the clusters, i.e., to increase cluster stability. Some algorithms, e.g. VMaSC [6], select the vehicle with the lowest relative velocity as the CH so that the member vehicles need more time to go out of the TR of the CH. After selecting the CH, a cluster is formed including the vehicles within the TR of the CH.

DCV [10] and some algorithms used position of the vehicles as the basis of clustering. The position of the vehicles among the clusters is measured and relatively center vehicle is selected as the CH so that the member vehicles need to cross more distance to go out of the cluster. After selecting the CH, a cluster is formed including the vehicles within the TR of the CH.

Some algorithms, e.g. ECHS [12], used degree of vehicles as the basis of clustering. The degree of vehicles is counted and the vehicle with the highest degree of vehicle is selected as the CH so that a greater number of vehicles can be accommodated within the cluster with a balance at both sides. After selecting the CH, a cluster is formed including the vehicles within the TR of the CH.

Some algorithms, e.g. IBC [14] used the intersection of the roads as the basis for clustering. Vehicles break down into multiple clusters based on their movement at the intersection. Developing a protocol based on the intersection only may not be effective at other places. Forced breakdown of the clusters increases the number of clusters in a single place.

Most of the existing algorithms are kind of effective in straight road without any intersection. Intersections are not given proper attention in the existing algorithms. A very few numbers of algorithms considered intersections; however, their algorithms work only in intersection and create many extra clusters. As a result, clusters of the existing algorithms break down frequently and achieve lower stability.

Clustering was proposed to solve the scalability issue and scalability becomes more significant at the intersection where the existing algorithms do not work properly at the intersection. The protocols that developed keeping mind intersection do not show better performance at highways. Therefore, we need a new clustering protocol for VANET which would consider the intersections of the road properly during the CH selection and cluster formation process to tackle the scalability and the stability issue in VANET regardless of road topology.

Most of the existing algorithms did not consider the intersections during creating clusters. As a result, the algorithms do not work at the intersection. For example, five vehicles are running on a road. We can create a cluster here. Even without creating clusters for five vehicles, we can set up effective communication among five vehicles. Moreover, creating cluster among five vehicles requires lower clustering overhead. In this case, frequent cluster breaking will not cause any effect on other vehicles on the roads that are not within the TR of these vehicles.

Now many vehicles come within the TR at the intersection and many vehicles run out of the TR. Instead of five vehicles, e.g., twenty vehicles are within the TR of the vehicles. Therefore, any vehicle is now transmitting and receiving the message with twenty vehicles. To create cluster at this point will create more clustering overhead. Besides, any change in the cluster will cause a substantial number of message transmissions. Consequently, due to lack of stability, the scalability problem will be severe.

To overcome the stability issue at the intersection without degrading the performance in other part of roads, we need to consider the movement of the vehicles at the intersection during the CH selection and formation of cluster so that a lower number of vehicles joining and leaving happens at the intersection.

If we force cluster breakdown at any given point, which is done in IBC, we are creating a high number of clustering overhead at a single point. IBC is considering this single point does not require smooth communication and there is no possibility of accident in those single points. We want to avoid any single point whether it is an intersection or any other point.

We have utilized numerous mobility parameters, such as velocity, acceleration, position, distance, direction, movement at the next intersection, transmission range, degree of vehicles, majority-vehicles, considered intersection, and constructed a future position of the vehicles to make cluster stable. We considered velocity, acceleration, movement of the vehicle, movement of the vehicle after the next intersection, and degree of vehicle etc. We considered the degree of vehicles in a limited sense, among the similarly suitable candidates only. We have also used the concept of majority-vehicles during the CH selection and the cluster formation. As a result, our algorithm can create more stable cluster regardless of topology. SDPC is an all-weather VANET clustering protocol.

Instead of all types of metrics, we want to get more optimized results for more important metrics only. In SDPC, we have evaluated the major metrics used for measuring stability and achieved better results for the average CH change rate, the average CH duration, the average CM duration, and the clustering overhead which reflects that SDPC can provide better clustering stability regardless of road topology.

4.2 State Diagram of Proposed Clustering Protocol

In the stable dynamic feedback-based predictive clustering (SDPC) protocol, vehicles can remain in one of three states: Temporal (TM), Cluster Head (CH), and Cluster Member (CM) as shown in Figure 4.1. TM is a temporary state when a vehicle enters the road, collects information about other vehicles within the transmission range, and tries to join in any existing cluster. CH is the head of a cluster and CMs are the members of the cluster. Vehicles communicate with their neighbor vehicles within their transmission range using OBU and all the vehicles are equipped with GPS. To express the difference with EN in JCV, TM is used in this section only. However, EN has been used after this section to avoid any confusion between TM and single-cluster.

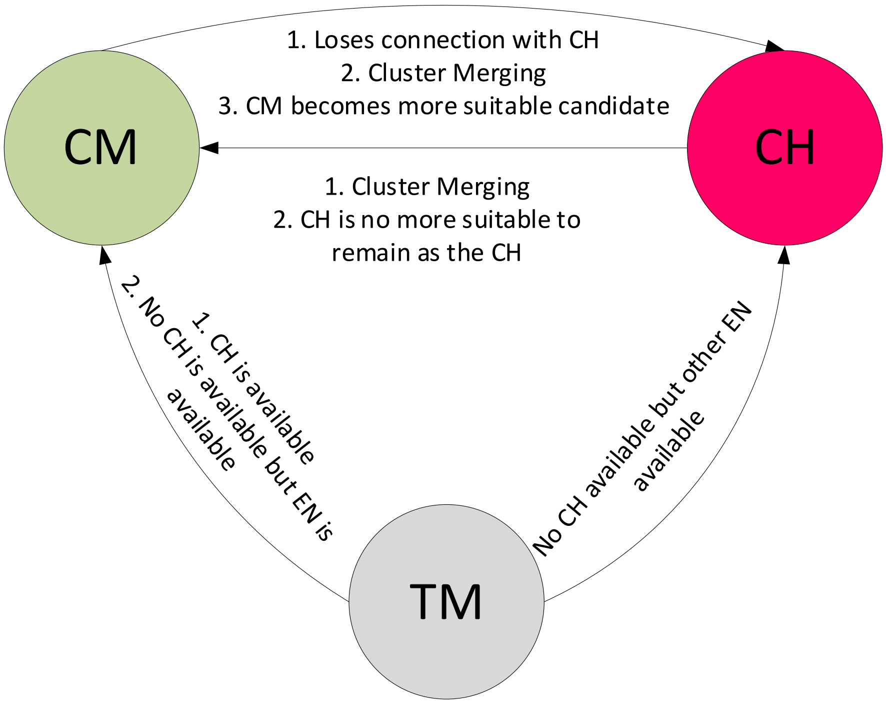

Figure 4.1: State diagram in SDPC.

SDPC protocol is a mobility-based clustering protocol like VMaSC and DCV protocols. VMaSC, DCV and SDPC are developed to provide stability to the clusters in VANET. VMaSC considers the relative velocity and DCV uses the geographical center position to select the CH. While VMaSC uses five states and DCV uses four states, SDPC uses only three states. Also, VMaSC, DCV and SDPC used nine, eight and four state transitions, respectively. Table 4.1 shows the notation used in SDPC.

Table 4.1: Notation used in SDPC with description.

NOTATION	DESCRIPTION
CH	CLUSTER HEAD
CM	CLUSTER MEMBER
TM	TEMPORAL STATE
EN	TM OR SINGLE-CLUSTER
TR	TRANSMISSION RANGE
CH_TIMER	CLUSTER HEAD STATE TIMER
CM_TIMER	CLUSTER MEMBER STATE TIMER
TM_TIMER	ENTRY STATE TIMER
CH_{IJ}	CH OF NEWLY CREATED CLUSTER AFTER MERGING
VEH_INFO	VEHICLE INFO
TM_REQ	MESSAGE FROM TM
CH_RESP	RESPONSE FROM A CH
VEH_ADV	PERIODIC VEHICLE ADVERTISE MESSAGE
ACK_MERGE	ACKNOWLEDGEMENT MESSAGE AFTER MERGING
VEH_RANK	RANKING OF A VEHICLE
STATUS	STATUS OF A VEHICLE

When a vehicle enters the road, it enters TM state. The vehicle starts *TM_TIMER*. It is a temporary state when the vehicle collects information from other vehicles withing its transmission range. If any vehicle is available within the transmission range, the vehicle tries to form a cluster.

If a vehicle is the most suitable candidate to be selected as the CH, then the vehicle changes its status to CH and advertises it to all the vehicles in its transmission range. If there is no vehicle in its transmission range, then the vehicle remains as a single cluster, i.e., a CH without any member.

If a vehicle finds any vehicle in its transmission range which is more suitable to become the CH, then the vehicle requests the CH to join the cluster. After receiving an acknowledge message from the CH, the vehicle changes its status to CM.

4.3 State Transition in SDPC

SDPC uses two states along with a temporary state and four state transitions. The number of state transitions is four which are:

- TM → CH
- TM → CM
- CH → CM
- CM → CH

Vehicles enter as TM, set a timer, and broadcast *VEH_ADV* message periodically to inform its presence on the road. From TM state, a vehicle can convert to a CH or CM. If there is no CH available but at least one TM is available within the TR of the TM and the TM is a better candidate for CH, then a cluster will be created, and the TM will be the CH of the new cluster.

The vehicle enters from TM to CM state if it receives an acknowledgment message from a CH of an existing cluster. If there is no CH available but at least one TM is available within the TR of the TM, and the other TM is a better candidate to become a CH, then the TM will become a CM of the new cluster.

A CH can transform to CM if certain condition satisfies. If two CHs come within the TR of each other and decide to merge into one cluster, start *CH_TIMER* and starts communication, then one or both CH changes their *STATUS* into CM. The same is applied if an TM comes within the TR. Moreover, due to the dynamicity of the vehicles, if a CM becomes a better candidate to be the CH, the present CH downgrades its *STATUS* to CM.

A CM can transform to CH satisfying certain criteria. A CM can become a CH if the CM loses connection with its CH and some vehicles are present within TR and this CM is the best candidate to become the CH. Secondly, during the cluster merging process, a CM can be elected as the CH of the new cluster if the CM is the best candidate compared to existing CHs and the rest of the vehicles. In this case, CM starts *CM_TIMER* and initiates the process. Thirdly, a CM can become a CH of its cluster if it becomes the best candidate than the rest of the vehicles due to vehicles' movement.

4.4 Cluster Formation

Upon entering a road, the state of a vehicle is TM, and the vehicle checks for neighboring vehicles within its transmission range. If any CH is available, the new vehicle will consider joining the cluster. The TM sets *TM_TIMER* and sends a join request (*TM_REQ*) to the CH. Upon receiving the *TM_REQ* message from a TM, the CH checks the feasibility of the TM becoming its member. If the check is successful, the CH sends a *CH_RESP* message to the TM. Upon receiving a *CH_RESP* message, the TM joins the cluster and changes its state to CM. If no response is received from a CH before *TM_TIMER* expires, the TM remains as TM in nature but changes its status to CH even though it has no member. In this case, it will be called single-cluster, i.e., cluster consisting of CH without any member. In the future works, we want to refrain from assigning any state for the temporary entry, rather we will count the number of single-cluster for un-clustered vehicles. For the rest of the dissertation, EN has been used to mean both TM and CH of single-cluster to avoid confusion. Figure 4.2 illustrates the cluster formation process in SDPC.

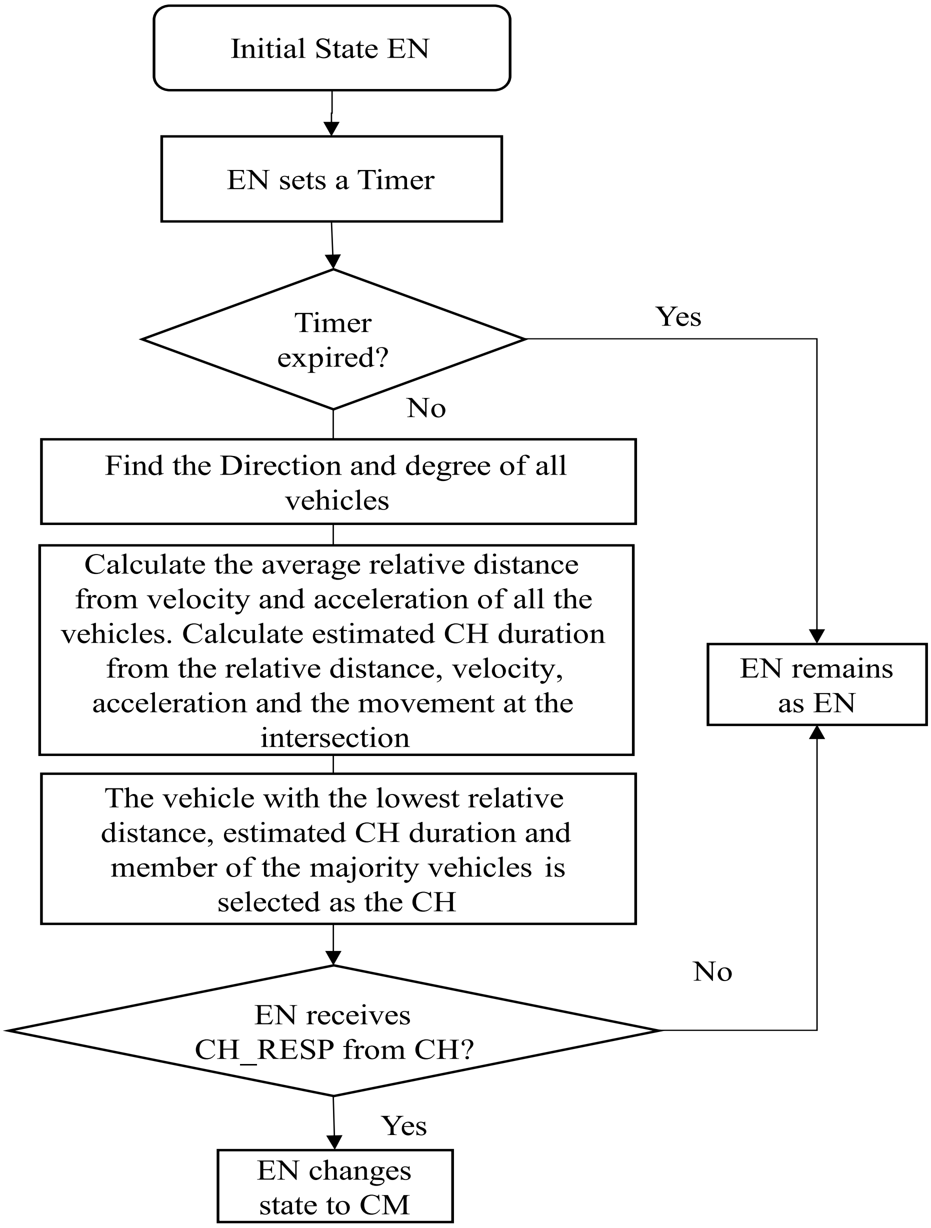

Figure 4.2: Cluster formation process in SDPC.

The most stable vehicles (vehicle with highest *VEH_RANK*) will be determined based on the mobility parameters and will be selected as the CH and cluster will be formed starting from the front end. Along with the other parameters, the movement of the vehicles after the next intersection is also considered and the CH will be selected from the majority-vehicles. The concept of majority-vehicles and the movement of the vehicles after the immediate next intersection have been illustrated in Figure 4.3.

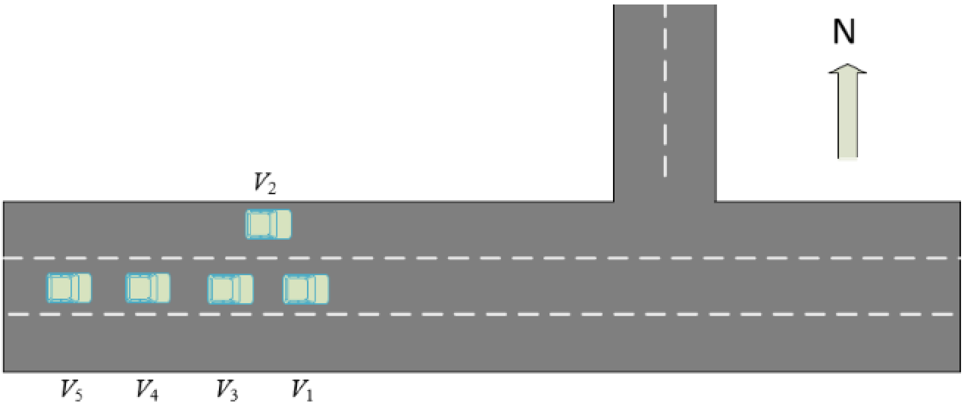

Figure 4.3: Illustration of movement of the vehicles after the next intersection.

At the next intersection, V_1, V_3, V_4, V_5 will run towards the east direction while V_2 will be running into the north direction. The number of vehicles in the east direction is four which is greater than the number of vehicles in the north direction which is one. Therefore, the CH will be selected from the vehicles who will be running into the east direction. V_2 is not a minority- vehicle and will not be considered for the candidacy of CH; however, it can join the cluster as a CM.

The degree of vehicle is the number of vehicles that can be reached directly in one-hop distance as shown in Figure 4.4. The degree of vehicle is applied in SDPC only when more than one vehicle becomes equally qualified candidates to become the CH. Hence, the calculation of the degree of vehicle is different and limited in SDPC.

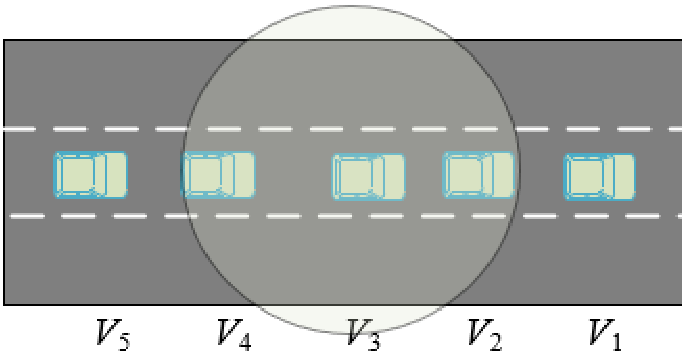

Figure 4.4: Degree of node in VANET.

The degree of node will be calculated for those vehicles who have the coverage of the first vehicle of the probable cluster. In this way, the number of stranded vehicles, i.e., the number of single clusters, can be reduced. In Figure 4.5, degree of vehicle for V_2, V_3 and V_4 is 2; however, if V_3 is selected as the CH, then V_1 and V_5 will not have any vehicle to form a cluster.

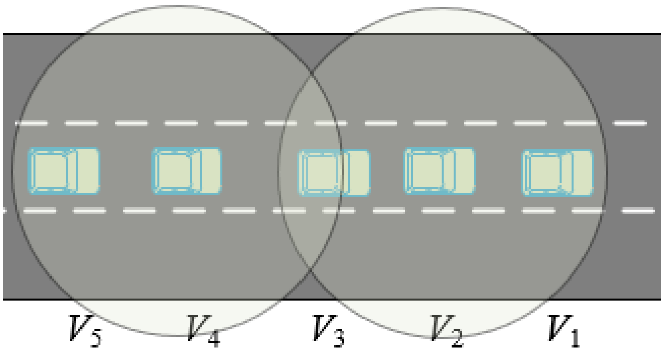

Figure 4.5: Degree of node consideration in SDPC.

On the other hand, if V_2 is selected as the CH who has the same degree of node as V_3 and V_4, V_4 and V_5 will get the opportunity to form a new cluster. Hence, a greater number of vehicles will be in clustered state. All the vehicles, including any vehicle who have a different moving direction after the next intersection, will be considered to count the degree of vehicle of a vehicle; however, a vehicle with different moving direction after the next intersection will not be a candidate to be a CH and will be considered to break the tie for more than one candidate for the CH. The minority vehicle will be considered to become the CM of the new cluster.

4.5 CH Selection

The most stable vehicles will be determined and will be selected as the CH and the clusters will be created starting from the front end.

In this protocol, geographical front vehicle from a group of vehicles where clustering algorithms will be applied is termed as the first-vehicle. The following notations have been used.

n = Total number of vehicles on the road.

V_i = i^{th} vehicles ($1 \leq i \leq n$).

V_j = j^{th} vehicles ($1 \leq j \leq n$).

$a_i(t)$ = Acceleration/deceleration of V_j at time t.

$s_{ij}(t)$ = Distance between the vehicles V_i and V_j at time t.

$\overline{s}_i(t)$= Average distance of vehicles V_i with respect to $n - 1$ vehicles.

$u_i(t)$ = Velocity of V_i at time t.

$u_{i,j}(t)$ = Relative velocity between V_i and V_j at time $t = u_i(t) - u_j(t)$.

$\overline{u}_i(t)$ = Average relative velocity of V_i with respect to other vehicles at time t

$$= \frac{\sum_{j=1}^{n} (u_{i,j}(t))}{n-1} \text{ where } i \neq j.$$

Velocity, acceleration, position, and distance are considered to calculate $\overline{s}_{ij}(t)$ for all the vehicles. Each vehicle advertises its expected or probable CH lifetime to the other vehicles. probable CH lifetime is calculated based on the mobility of the vehicles and the neighbor vehicles. A vehicle from the vehicles with the highest probable CH lifetime will be selected as the CH. If more than one vehicle has the highest probable CH lifetime, a vehicle among the vehicles with $\min(\overline{s}_i(t))$ will be selected as the CH of the cluster. If we get more than one vehicle, the vehicle with the higher degree of vehicles will get preference.

For all the vehicles, $s_{ij}(t)$ will be calculated, at any time, t,

$$s_{ij}(t) = s_{ij}(0) + \int_0^t u_{i,j}(t)\, dt$$

Similarly, $u_{i,j}(t) = u_{ij}(0) + \int_0^t (a_i(t) - a_j(t))\, dt$

$$s_{ij}(t) = s_{ij}(0) + \left(u_i(0) - u_j(0)\right) t + \int_0^t \int_0^t \left(a_i(t) - a_j(t)\right) dt\, dt$$

Based on $s_{ij}(t)$, we can predict the future position of each vehicle. Based on $s_{ij}(t)$ and *TR* of the vehicles, we can calculate the probable CH-lifetime for each vehicle at time *t*. Average distance of V_i with respect to $n - 1$ vehicles,

$$\overline{s}_i(t) = \frac{\sum_{j=1}^{n} \left(s_{i,j}(t)\right)}{n - 1}$$

Based on the probable CH-lifetime and $\overline{s}_i(t)$, we can narrow down our list of probable CH. Among the highest probable CH, the vehicle with the lowest $\overline{s}_i(t)$ will be go to the next level. Therefore, the CH will be from,

$$\min(\overline{s}_i(t)), \text{ where } (1 \leq i \leq n)$$

Among these set of probable CH, the degree of vehicle will be applied, i.e., the vehicle with $\min(\overline{s}_i(t))$ and $\max(degree\ of\ vehicle)$ will be selected as the CH to compact the length of the cluster and maximizing the number of vehicles within the same length. If we have multiple candidates, then the vehicle with the lowest relative velocity (*AVG_REL_SPEED*) will be selected as the CH. In case of multiple vehicles remain at this stage, the vehicle that spends more time on road will be selected as the CH.

4.6 Cluster formation Algorithm

For any group of vehicles DO the following:

1. Find the direction for all the vehicles.
2. Find the degree of vehicle for all the vehicles.
3. Find the directions for all the vehicles at the next intersection.
4. Calculate $\overline{s}_i(t)$ for all the vehicles (where $t = 0$).

5. CH Selection: Select the CH based on the CH selection algorithm described next.
6. Create a cluster based on the *TR* of the CH selected at STEP 5.
7. Select a new first vehicle from the rest of the vehicles excluding the vehicles who are already in a cluster.
8. Repeat STEP 5 to 7.

4.7 CH selection Algorithm

Select the CH if the vehicle satisfies the following conditions at $t = 0$:

a) The first vehicles should be in the range of the CH.
b) Majority-vehicles should be in the same direction as the CH after the next intersection.
c) Find the CMs of the potential CHs.
d) Find the max (t) until the CMs remain in the potential clusters and the number of potential clusters < 2 [for each CH, $s_{ij} < TR$ to remain in the potential cluster; s_{ij} = Distance between vehicle V_i and Vj where V_i = potential CHs and V_j = potential CMs]
e) Select the vehicle as the CH for which $\bar{s}_i(max(t))$ is the minimum, i.e., the vehicle with $\min(\bar{s}_i(max(t)))$.
f) The CH should be from $\min(\bar{s}_i(max(t)))$ for which degree of vehicles is the maximum.
g) If multiple candidates are found, the vehicle with the lowest relative velocity $\min(\bar{u}_i(t))$ among the vehicle with $\min(\bar{s}_i(max(t)))$ and $\max(degree\ of\ vehicle)$ will be selected as the CH.

4.8 Gateway Selection

Cluster gateways are two CMs who maintains communication with other clusters. Two vehicles from the front and the back will be selected as the CGs. After creating the cluster, CH will appoint two CMs as the CGs as shown in Figure 4.6.

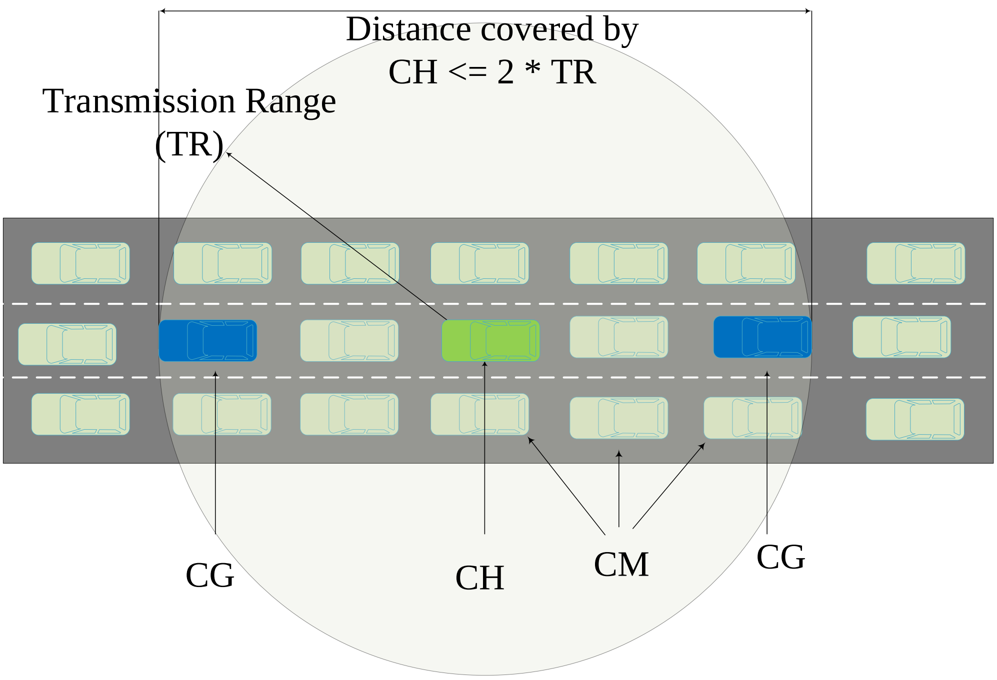

Figure 4.6: Gateway nodes in SDPC.

4.9 Cluster Merging

If two clusters overlap for a predefined period, two CHs will initiate cluster merging process. Two CHs will transmit information of all the member vehicles. If a combined cluster is possible, two CH will start the merging process. Both the CH will transmit information of the CMs to the newly selected CH. Upon receiving the request from the CHs, the new CH will send an acknowledgement message to the old CHs. The old CHs will send the response messages to the new CH as well as to all the CMs about its status as the CH. The cluster merging algorithm is given below:

==

IF CH_i receives request for merge from CH_j

CH_i estimates the length of the merged cluster

IF the $length \leq 2 * TR$ THEN

CH_i and CH_j selects a *Vehicle* as the new CH, CH_{ij}

CH_i and CH_j sends *ACK_merge* to CH_{ij}

$CH_i \rightarrow CM_{ij}$

$CH_j \rightarrow CM_{ij}$

ENDIF

ENDIF

IF CM_i or CM_j is selected as the new CH (CH_{ij}) and receives *ACK_merge*

$CM_i \rightarrow CH_{ij}$ or $CM_j \rightarrow CH_{ij}$

$CM_{ij} \leftarrow CM_{i,}$ $CM_{ij} \leftarrow CM_j$

CH_{ij} broadcasts *HELLO* message to all CM_{ij}

ENDIF

==

4.10 Cluster Leaving

When a CM wants to leave a cluster, the CM will send a request to the CH. Upon receiving an acknowledgement message from the CH, the CM will leave the cluster. Besides, if the CH does not receive any regular *HELLO* message from any CM for a specified time, the vehicle will be no longer considered as the member of the cluster. The CH will delete the entry for the CM and will advertise to the CMs of the cluster. The cluster leaving algorithm is SDPC is given below:

==

IF $s_{ij} < TR$ AND *CH* receives *HELLO* message from *CM* THEN

CH updates the information of *CM*

ENDIF

IF message gap exceeds waiting period THEN

CH deletes *CM*

ENDIF

==

4.11 Routing Structure in SDPC

In VANET, vehicles do not have any hierarchy; however, clustering protocol creates a hierarchical structure. The routing table entries for the vehicles in SDPC are shown in in Table 4.2.

Table 4.2: Routing table structure in SDPC.

Vehicle ID (V_i)	IsCH	$IsCG_{front}$	$IsCG_{back}$	CH	CG_{front}	CG_{back}	CML

V_i is the vehicle identification number. IsCH denotes whether the vehicle is the CH of the cluster. The value is 'Yes', if V_i is the CH of the cluster, otherwise the value is 'No'. Similarly, $IsCG_{front}$ denotes whether the vehicle is the front CG of the cluster and $IsCG_{back}$ denotes whether the vehicle is the back CG of the cluster. IsCH, $IsCG_{front}$, and $IsCG_{back}$ are derived from the next columns but added in column 2, 3, and 4 to retrieve information faster. CH is the vehicle ID of the cluster head. CG_{front} is the vehicle ID of the gateway vehicle of the cluster on the front and CG_{front} is the vehicle ID of the gateway vehicle of cluster at the back. CML is the list of CMs in the cluster.

Some algorithms such as VMaSC did not mention any CG selection algorithm since it is a multi-hop protocol; however, at least two CGs are required to route the packets in single-hop clustering. For simulation, we also use single-hop version of VMaSC. Therefore, we are assuming that VMaSC will also have two CGs, one in the front and one in the back.

As an example, routing table structure in SDPC for V_3 in the Figure 4.7 is shown in Table 4.3.

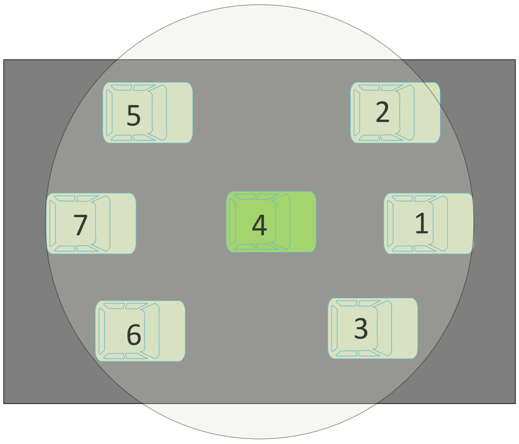

Figure 4.7: Routing table example in SDPC.

Even though CGs can be excluded from the CML, in that case the member of a cluster will be formed combining CG_{front} list, CG_{back} list, and CML; however, CGs are also CMs. Therefore, to remove any confusion, we added them in CML.

Table 4.3: Routing table entry in SDPC.

Vehicle ID	IsCH	$IsCG_{front}$	Is CG_{back}	CH	CG_{front}	CG_{back}	CML
V_3				V_4	V_1	V_7	V_1 V_2 V_5 V_6 V_7

4.12 Comparison of SDPC with VMaSC, DCV, ECHS and IBC

VMaSC selects the vehicle with the lowest average relative velocity ($\overline{u}_i(t)$) as the CH. VMaSC uses five state of vehicles and nine state of transitions. VMaSC does not consider the degree of node in the CH selection process but consider to creation of cluster.

DCV selects the vehicle that geographically middle or the lowest $\overline{s}_{ij}(t)$ as the CH. DCV uses four state of vehicles and eight state of transitions. Moreover, DCV has a lower TR and IBC re-clusters at 2*TR distance from the intersection. DCV does not consider degree of node in the CH selection or cluster formation process.

While VMaSC and DCV are not considering the degree of vehicles at all during the CH selection process, ECHS selects the vehicle with the even density as the CH. ECHS calculates the degree of vehicles in front and back and the vehicle with the equal number of front and back vehicle is selected as the CH. ECHS also divides vehicle based on their velocity.

IBC selects the CH based on velocity and distance when there is no intersection within the twice of transmission range; otherwise, it creates multiple clusters for the vehicles based on the direction at the intersection. The most important feature of IBC is consideration of intersection at a specific place of the road. It forces clusters to break down at the distance of twice of transmission range (2*TR) from the intersection.

SDPC selects the vehicle with the highest predicted stability as the CH. SDPC uses three states of the vehicles and four state transitions. SDPC consider the degree of vehicles partially to minimize the number of clusters when multiple vehicles are qualified to become the CH without affecting the CH duration or the number of CH change. SDPC uses majority-vehicles and the movement of the vehicles along with a predictive approach of a probable CH with the best clustering stability during the CH selection process to maximize the cluster stability.

SDPC used a lower number of states and a lower number of state transitions. While VMaSC used five states of vehicles and nine states transitions and DCV used four states of vehicles and six state transitions, ECHS and IBC did not mention anything about state or state transition optimization. However, SDPC used three states of vehicles and reduced the number of state transitions to four. It results in a lower number of clustering overhead for SDPC.

While VMaSC, IBC and DCV did not consider the degree of vehicles at all during the CH selection process, ECHS selects the CH based on the degree. SDPC used degree in a limited sense during the CH selection process to break the tie among multiple candidates for the CH to reduce unnecessary clusters.

SDPC and IBC requires to know the movement of the vehicles at the next intersection, which is not needed in VMaSC, ECHS and DCV.

SDPC is a single-hop clustering protocol like DCV, ECHS and IBC, whereas VMaSC is a multi-hop clustering protocol. However, single-hop VMaSC has been used in the simulation to maintain uniformity among the protocols.

In SDPC, the vehicles who do not fall in the majority-vehicle category, they can either choose to form their own cluster or can join in the existing cluster without affecting the CH selection process of the majority-vehicles, which is unique for SDPC. A comparison of SDPC with VMaSC, DCV, ECHS and IBC is presented in Table 4.4.

Table 4.4: Comparison of SDPC with VMaSC, DCV and ECHS.

Criteria	**VMaSC**	**DCV**	**ECHS**	**IBC**	**SDPC**
Hop Count	Multi-hop	Single hop	Single hop	Single hop	Single hop
Transmission Range used for the CH	Same as the TR (TR = Actual TR)	Lower than the actual TR (TR < Actual TR)	Same as the TR (TR = Actual TR)	Same as the TR (TR = Actual TR)	Same as the TR (TR = Actual TR)
Known to each vehicle of clusters	Position Velocity Acc. Direction TR	Position Velocity Acc. Direction TR Distance	Position Velocity Direction TR Degree of vehicle	Position Velocity Acc. Direction TR Distance	Position Velocity Acc. Direction TR

	Degree of vehicle			Direction at the next intersection	Degree of vehicle Distance First vehicle Majority-vehicles Direction at the next intersection
Parameters used for CH Selection	Position Direction TR Velocity	Position Direction TR (< TR) Distance	Position Direction TR Degree of vehicles	Position Direction TR Distance Direction at the next intersection	Position Direction TR Velocity Acceleration

					Direction at the next intersection Degree of vehicles Majority-vehicles
Emphasized	Straight road	Straight road	Straight road	Intersection	Straight road and intersection
Total number of states of the vehicles	5	4	Not mentioned	Not mentioned	3
Number of state transitions	9	8	Not mentioned	Not mentioned	4
Direction of the vehicles	CH and CMs should be running at the same direction	CH and CMs should be running at the same direction	CH and CMs should be running at the same direction	CH and CMs should be running at the same direction	CH and CMs should be running at the same direction

Degree of vehicle for CH selection	Not used	Not used	Used	Not used	Used with condition
Inter-cluster communication	LTE	Gateway node	Gateway node	Not mentioned	Gateway node
Gateway selection	N/A	Two sides	Two sides	N/A	Two sides
Single or multiple gateways	N/A	Single	Multiple	N/A	Single

4.13 Summary

In this chapter, SDPC clustering protocol is presented that is an all-weather VANET clustering protocol and can provide stability for both highways and intersections regardless of the road topology. A comparison with previous algorithms such as VMaSC, DCV, and ECHS has also been presented. Chapter 5 describes the evaluation setup for simulation with the description of our simulation platform CVAENTSIM.

Chapter 5

Experimental Evaluation

VANET clustering is different than any other kind of clustering due to the high mobility of the vehicles. Likewise, VANET simulator requires some unique features such as internet-based real-time data processing, huge data analysis, the complex calculation to maintain hierarchy among the vehicles; however, neither web-based VANET simulator nor clustering module available in the existing simulators. Therefore, a simulator that will be able to simulate any feature of VANET equipped with a clustering module and accessible via the internet is a growing need in vehicular communication research. A fully functional discrete-event VANET simulator has been developed that includes all the features to simulate VANET clustering. Moreover, the simulator provides an easy and interactive web interface. It is the *first of its kind* which integrates features of the VANET simulator, built-in VANET clustering module, uses database server for data storing and processing, and accessible over internet. The simulator has been described in Section 5.1. The simulation setup of JCV is discussed in Section 5.2 and the setup for SDPC is presented in Section 5.3.

5.1 Cloud-based VANET Simulator (CVANETSIM)

Internet of vehicles (IoV) and connected vehicles require vehicular communication which is facilitated by VANET. VANET has some unique features such as high mobility that requires specialized software to simulate. Moreover, VANET clustering requires complex calculation for a hierarchical structure in real-time. To simulate a VANET clustering scenario, two parts are simulated. The first part is to generate traffic that is provided by traffic simulator like simulation for urban mobility (SUMO) [112], MOVE, VanetMobiSim, etc. The second part is the main part of the VANET simulation what is simulated by a network simulator.

Internet-of-things (IoT), internet-of-vehicles (IoV), etc. are advancing rapidly. At the same time, simulation platforms for VANET are not advancing at a steady rate that can meet the growing demand for advanced simulation tools such as VANET clustering which requires the use of a large database, real-time data processing, complex multi-level hierarchical calculation, etc. Moreover, the platform should be accessible through the internet, preferably, a cloud-based platform with a user-friendly environment.

Some simulation platforms have been developed over the years; however, while VANET research is advancing rapidly, the simulation platforms are not advancing concurrently. Some attempts have been made to develop specialized VANET simulators such as TraNS, veins, Netsim, etc.; however, these platforms are either proprietary-based or need a specific operating system and specialized installation, and do not have accessibility over internet. Development of any protocol on these platforms requires a long time to acquire necessary skills. More importantly, they do not have any clustering module.

As part of our research, we need a VANET simulation platform that is equipped with a fully functional clustering module, easy access of data through a database server, machine-independent, and easily accessible over internet. Moreover, a cloud-based

VANET simulation platform will facilitate a way for an application that can be used to implement any real-life vehicular communication.

Our objective is to develop a fully functional VANET simulator with a complete clustering module. Our aim is to develop a simulator which will be using a database server for large data and will be accessible over internet. CVANETSIM is the first of its kind. The following features are available in CVANETSIM:

- All necessary features of a VANET simulator.
- Fully functional clustering module.
- Using a database to provide efficient access and update on data.
- Cloud-based platform to access over the internet.
- No installation or specialized skill needed.
- Easy to use graphical user interface (GUI).
- Password protected user authentication.

5.2 Architecture of CVANETSIM

The discrete-event CVANET simulator is developed using Java Server Page and MySQL database server in a Tomcat web server. We have also used an additional PHP server to access database over internet. A traditional network simulator was not developed keeping in mind the VANET scenario. VANET requires analysis of big data as well as clustering protocol demands complex multi-level calculation. Due to this combination, clustering protocols are not readily available in the traditional network simulator. Moreover, to build a real-life application for VANET, we need a web-based VANET simulator that is capable of handling big data using a database. The simulator will have a clustering protocol and can be accessed over the internet. CVANETSIM is developed especially for VANET scenarios and fulfilled all these requirements including a VANET clustering module. Since CVANETSIM is accessible from the internet, CVANETSIM does not serve as a simulator only, any kind of real-life VANET application can be developed using this simulator without storing or processing any data on a local machine. CVANETSIM is a discrete-event simulator which uses MySQL database to import SUMO traffic data and then process traffic data to calculate necessary calculation to create cluster. CVANETSIM processes SUMO data and analyzes various features of vehicles such as degree, transmission range, velocity, relative velocity, distance, position, relative distance, angle, etc. After extracting all information, the cluster is formed, and the CH is selected based on individual algorithm. The architecture of CVANETSIM is shown in Figure 5.1.

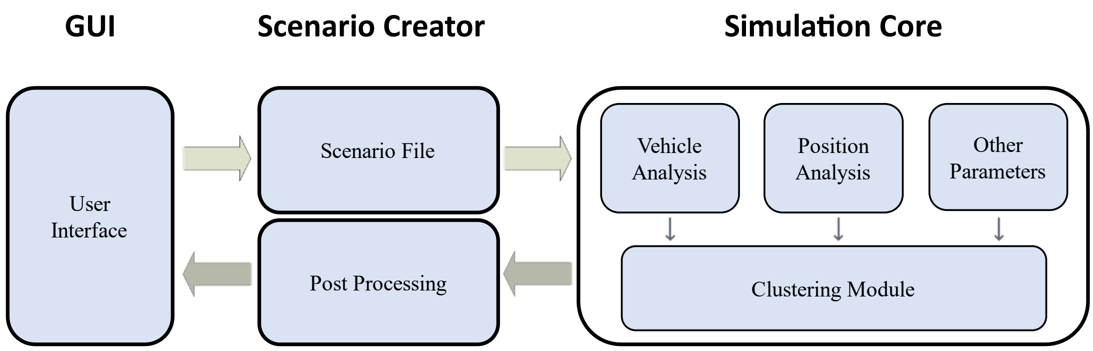

Figure 5.1: Architecture of CVANETSIM.

CVANET has four main components: GUI, Scenario file, Simulation core, and post-processing engine. GUI is the graphical user interface of CVANETSIM. The scenario file is the traffic data given as the input. An analysis is performed on features of vehicles, the position of the vehicles, and the features of DSRC, etc. If clustering protocol is needed to be implemented, the clustering module receives all analyzed data and process further to create cluster and cluster head (CH) depending on the algorithm and transmission range. In the next processing phase, all other vehicles are assigned a cluster depending on the clustering protocol and transmission range.

CVANETSIM is password protected and the user can find menu items after login. A screenshot of the developed CVANETSIM GUI is shown in Figure 5.2.

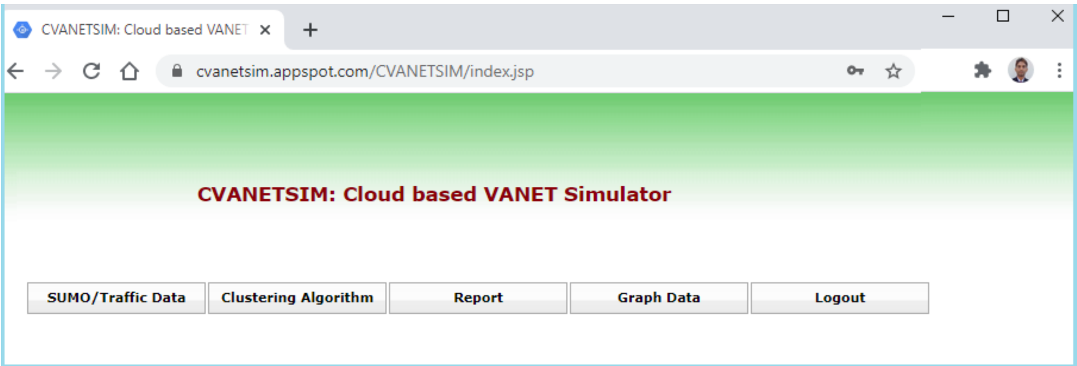

Figure 5.2: GUI of CVANETSIM.

Four main items in the menu are:

- SUMO/Traffic Data: Data from SUMO or any traffic simulator is given as the input to be processed for the next phase. Vehicle features, location, other parameters are analyzed for every discrete event.

- Clustering Algorithm: Here is the code for a specific VANET clustering algorithm. Based on the algorithm, CH is assigned, and cluster is formed at this stage.
- Report: A detailed report is generated at this stage after analyzing each discrete event and mentioned time interval.
- Graph Data: From the processed data, varieties of reports are generated which can be used to generate graphs. CVANETSIM can generate the average CH change rate, the average CH duration, the average CM duration, the average number of clusters, the average number of CM, the average number of non-clustered vehicles, etc.

5.3 Simulation Setup for JCV

We have used SUMO to generate traffic data and CVANETSIM to simulate the network part to generate cluster. The road consists of a two-lane and two-way road of 25 km in length. A total of 100 vehicles entered the road with a velocity range from 10 m/s to 35 m/s. After all the vehicles have entered the road, the simulation runs for 300 seconds.

To achieve real-life environment, different classes of vehicles and different road topologies have been used. The first class of vehicles always runs at 10 m/s and the maximum speed of the second class of vehicles ranges from 10 m/s to 35 m/s. The overtaking decision of the vehicles is calculated using the distance, velocity of the two vehicles, and acceleration-deceleration of the vehicles.

Three topologies are created, where 25 to 45 vehicles pass through each topology. The first topology is a highway-like straight road and the second topology contains junctions occasionally where the third topology consists of multiple junctions. For each scenario, 100 seconds are calculated which accumulates a total of 100 vehicles of 300 seconds. Table 5.1 presents the simulation setup of JCV.

Table 5.1: Simulation parameters for JCV.

NOTATION	**DESCRIPTION**
SIMULATION TIME	300 S
MAXIMUM VELOCITY	10 – 35 M/S
NUMBER OF VEHICLES	100
TRAMISSION RANGE	200 M
ROAD LENGTH	25 KM
ACCELERATION RATE	1~10 M/S^2
DECELARATION RATE	3~8 M/S^2
TRAFFIC FLOW RATE	1200 VEHICLES/HOUR
MAC PROTOCOL	IEEE 802.11P
VEH_ADV PERIOD	200 MS
VEH_ADV SIZE	64 BYTES
EN_TIMER	1 S
CM_TIMER	1 S
CH_TIMER	1 S
MERGE_TIMER	1 S
MOBILITY MODEL	CAR-FOLLOWING MODEL

The simulation result received using one hundred vehicles will reflect the real scenario for any number of vehicles in a scalable context, because some metrics such as the number of clusters will increase proportionately while some metrics such as the average CH duration and the average CM duration will remain the same. The maximum speed range of 10 m/s to 35 m/s is equivalent to 36 kilometer/hour to 126 kilometer/hour that covers both the city and highway environment suitably.

5.4 Performance Metrics in JCV

The major performance metrics used to evaluate VANET clustering stability protocols are the average CH change rate, the average CH duration, the average CM duration, and the clustering overhead. Some other minority metrics are also used in the evaluation process along with these major metrics. To evaluate JCV, we wanted to evaluate all kinds of metrics that have a relation with stability of the clusters. The metrics are described below.

- Average CH change rate (per second): This is the number of state transitions from CH to another state per unit time. It reflects the longevity of a cluster. A lower CH change rate is preferable for the stability of VANET clustering.
- Average CH duration: This is the time between a vehicle becoming a CH and subsequently changing to another state. The stability of a cluster highly depends on this metric. If the CH has a long lifetime, the necessity of creating a new cluster will be low.
- Average CM duration: This is the time a vehicle spends as a CM of a cluster. Longer CM duration will reduce the number of vehicle join/disjoin.

- Average clustering overhead: Clustering overhead is the ratio of the number of clustering packets to the total number of packets transmitted by the network. Minimizing the clustering overhead is highly favorable.
- Average number of clusters: This is the number of CH at any given time.
- Cluster participation: It is the ratio of the number of vehicles in CH or CM state versus the total number of vehicles. A higher percentage means the clustering algorithm is stable.
- Ratio of CM: It is the number of CM compared to other types of vehicles.
- Average packet transmission delay: It is the average time needed to deliver a packet from source to the receivers.
- Average number of un-clustered vehicles: It is the number of vehicles that do not belong to any cluster. This number reflects the number of unclustered vehicles during the simulation.

5.5 Simulation Setup for SDPC

Simulation setup for SDPC such as topology, entry speed, mobility model, mobility model, entry process, etc. are described in this section. Simulation parameters for SDPC are shown in Table 5.2.

5.5.1 Topology

The road topology used for SDPC is shown in Figure 5.3. Roads contain multiple intersections. Total road length is 60 km; however, no vehicle runs for the entire 60 km. The lower four nodes are the starting points for the vehicles and the upper four nodes are the end points.

Table 5.2: Simulation parameters for SDPC.

NOTATION	**DESCRIPTION**
SIMULATION TIME	300S
MAXIMUM VELOCITY	10 – 35 M/S
NUMBER OF VEHICLES	100
TRAMISSION RANGE	200 M
ACCELERATION RATE	10 M/S^2
DECELARATION RATE	5 M/S^2
TRAFFIC FLOW RATE	2 VEHICLES/SECOND
MAC PROTOCOL	IEEE 802.11P
VEH_ADV PERIOD	200 MS
VEH_ADV SIZE	64 BYTES
EN_TIMER	2 S
CM_TIMER	2 S
CH_TIMER	2 S
MERGE_TIMER	2 S
MOBILITY MODEL	CAR-FOLLOWING MODEL

Vehicles start from one of the four starting points and end at one of the four end points. At the intersection, some vehicles run straight, some vehicles run to the left direction and some vehicles run to the right direction, i.e., vehicles running from the west to the east direction can run towards the east, north, or east direction at the intersection. No vehicle runs in the west direction. A vehicle running from south to north direction can run towards the north or east direction at the intersection. Similarly, a vehicle running from the north to the south direction can run towards the south or east direction at the intersection.

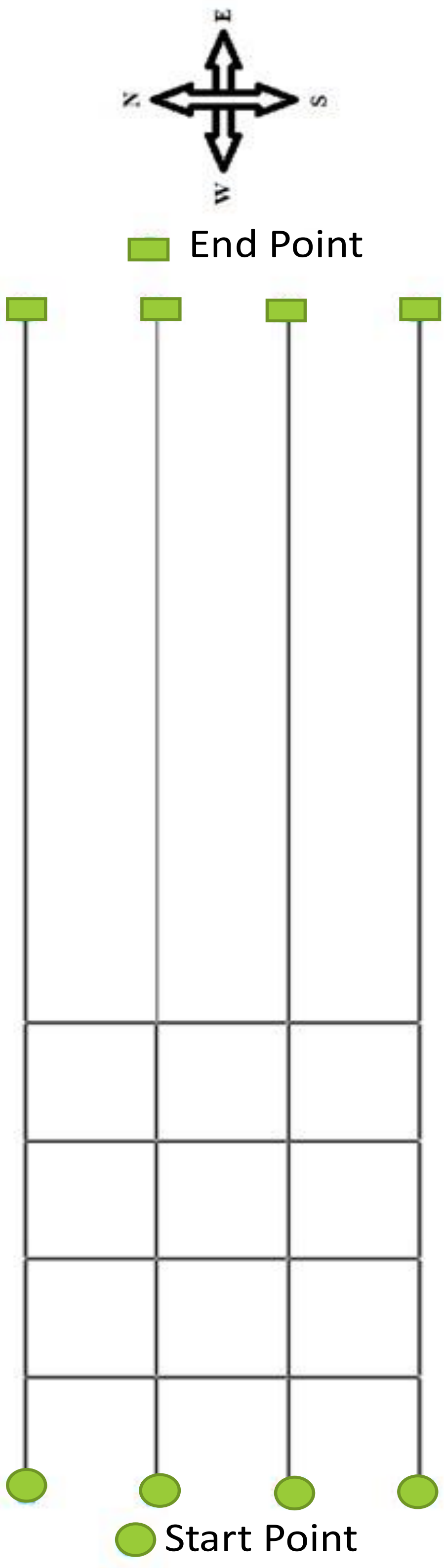

Figure 5.3: Road topology used in SDPC.

5.5.2 Number of Lanes

The road used in the simulation consists of two-lane two-way roads.

5.5.3 Overtaking

Vehicles can overtake other vehicles. The overtaking decision of the vehicles is calculated using the distance, velocity of the two vehicles, and acceleration-deceleration of the vehicles.

5.5.4 Entry speed

Entry speed varies from 10 m/s to 30 m/s depending on the maximum velocity. Entry speed does not exceed the maximum velocity.

5.5.5 Maximum velocity

Vehicles run at different maximum velocities from 10 m/s to 35 m/s. Some vehicles achieve exact maximum speed while some vehicles achieve some random speed close to the maximum speed.

5.5.6 Mobility model

Car-following models have been used where vehicles maintain a two-second gap between two vehicles.

5.5.7 Direction

Vehicles can change their direction at the intersection. If the direction is changed, vehicles slow down at the intersection at a random speed and accelerate to the maximum speed after the intersection.

5.5.8 Entry process

Vehicles enter the road following a Poisson process at the rate of 2 vehicles/second using the car-following model.

5.5.9 Result calculation

After all the vehicles have entered the road, the simulation results are calculated. Total simulation time is three hundred seconds. 100 vehicles are divided into three groups: 34, 33, and 33 in each group. The first group of vehicles ran from the east to the west direction only. The second group of vehicles occasionally change direction, and the third group of vehicles frequently change their direction at the intersection. The first group of vehicles enter the road at the rate of 2 vehicles/seconds and after all vehicles reach at the maximum speed, the results are calculated from 51 seconds to 250 seconds, i.e., 200 seconds. These results are converted into one hundred seconds. Then the second group of vehicles enter the road, and the results are calculated for 200 seconds, and the results are converted into 100 seconds. Similarly, the third group of vehicles enter the road, results are calculated for 200 seconds, and converted into 100 seconds. Finally, the results from each group are used to construct the final results for 300 seconds.

The performance of SDPC has been compared with four other VANET clustering protocols: VMaSC, DCV, ECHS and IBC. We have used SUMO for traffic generation and CVANETSIM to simulate clustering protocols.

5.6 Performance Metrics in SDPC

We focused on the major metrics for SDPC. We did not want to improve minor metrics sacrificing the optimization of major metrics. We evaluated the average CH change rate, the average CH duration, the average CM duration, and the clustering overhead. We wanted to maximize the performance of SDPC in terms of the major metrics. The performance metrics used to evaluate SDPC protocol are described below.

- Average CH change rate per second: This is the number of state transitions from CH to another state per unit time. It reflects the longevity of a cluster. A lower CH change rate is preferable for the stability of VANET clustering. For cluster stability, this is the most important metric. New cluster formation requires a lot of message transmission among the vehicles. The main metric that is used to measure stability of clustering in VANET is the average CH change rate.
- Average CH duration: This is the time between a vehicle becoming a CH and subsequently changing to another state. A longer average CH duration is highly expected.
- Average CM duration: This is the average time a vehicle spends as a CM. Average CM duration alone does not have significant effect on cluster stability. It can be related along other metrics to measure the stability of the clusters.

- Clustering overhead: This is the ratio of the number of clustering related packets to the total number of packets. This is useful for stability to know how many extra packets are generated by the vehicles due to clustering compared to the total number of packet transmission.

Along with these four metrics, we have generated some other graphs to analyze the performance of SDPC.

5.7 Summary

In this chapter, the details of CVANETSIM simulator and simulation setup for JCV and SDPC have been presented. CVANETSIM comes with many unique features. Chapter 6 presents the results of JCV and SDPC and compares the results with the existing protocols.

Chapter 6

Results and Analysis

We have compared the performance of JCV with VMaSC and DCV in Section 6.1. Performance comparison of SDPC with VMaSC, ECHS, IBC and DCV presented in Section 6.2.

6.1 Results of JCV

We presented the comparison of stability metrics in Section 6.1.1 to Section 6.1.10. Before that, we presented results generated by JCV for CH, CM and EN.

The average duration of CH, CM and EN for JCV are shown in Figure 6.1 for different maximum velocities. The average CH duration remains remarkably high for JCV for different velocities. It maintains a stable position even though it slightly decreases at the higher velocities. The average CM duration is also very high and gradually decreases at the higher velocities. On the other hand, the average EN duration increases at higher speeds. Even then, the average CH duration and the average CM duration remains very high compared to the average EN duration in every case.

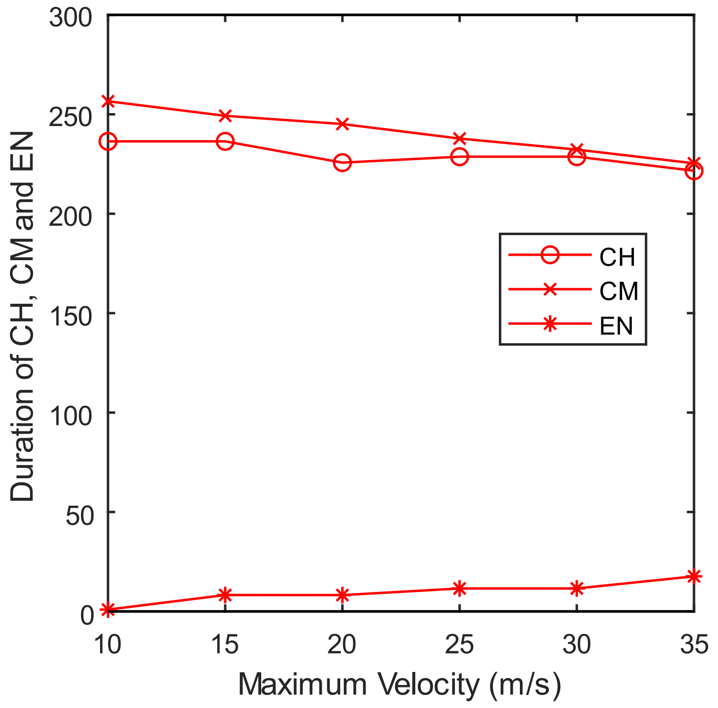

Figure 6.1: Impact of vehicle velocity for JCV on duration of CH, CM, and EN.

Figure 6.2 shows the average number of CH, CM and EN at the different maximum velocities. It shows that the average number of CH remains always higher than the average number of EN. A lower average number of EN indicates that very few vehicles remain as unclustered. At the same time, a large number of vehicles remain as the CM which also represents the stability of the cluster.

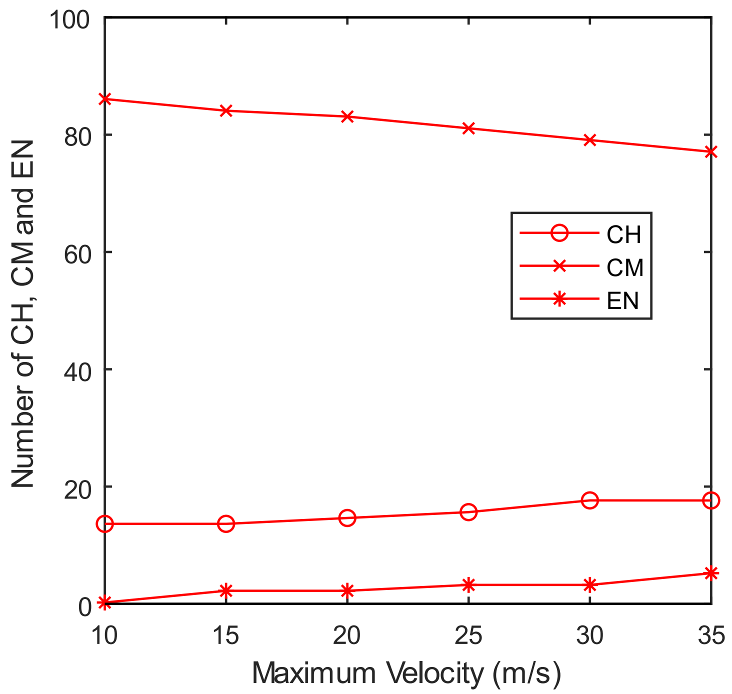

Figure 6.2: Impact of vehicle velocity for JCV on number of CH, CM, and EN.

Figure 6.3 shows the average number of vehicles in clustered and unclustered states at different velocities. At higher speeds, vehicles easily get out of the transmission range of other vehicles since vehicles move longer distances every second. We have used car-following models where vehicles maintain two-second distance from other vehicles. Even then, more than 90% of vehicles remain in a clustered state, which shows the efficiency of JCV.

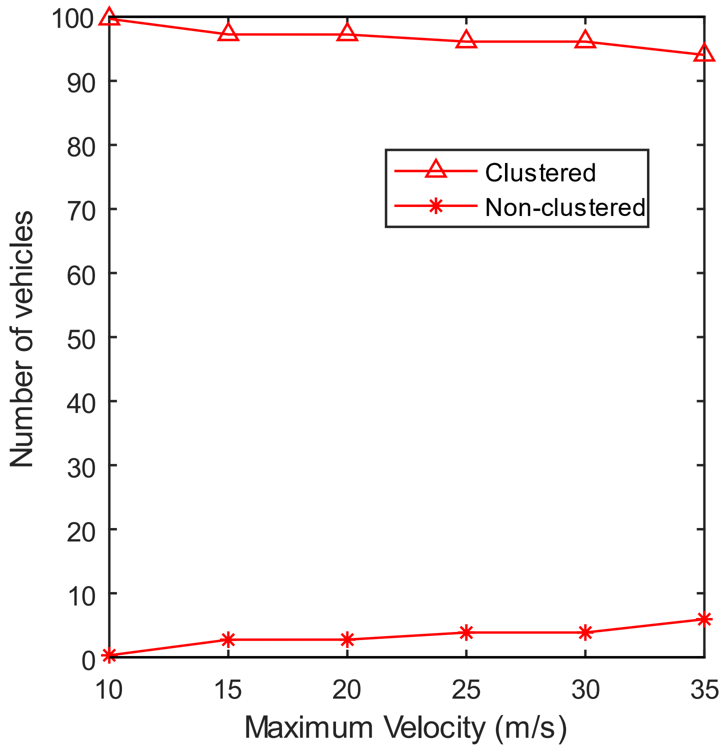

Figure 6.3: Impact of vehicle velocity on Vehicles in cluster.

We will now present the pattern for all three protocols for CH, CM and EN. Figure 6.4 shows the average number of CH, CM, and EN in terms of velocity. Similar pattern or tendency is shown by all three algorithms. More number of CM are created compared to the average number of CH and EN. The average number of EN is the lowest compared to the number of CM and CH. The average number of CM decreases and the average number of CH and the average number of EN increase when velocity increases because vehicle dynamicity is changed frequently at the higher speeds.

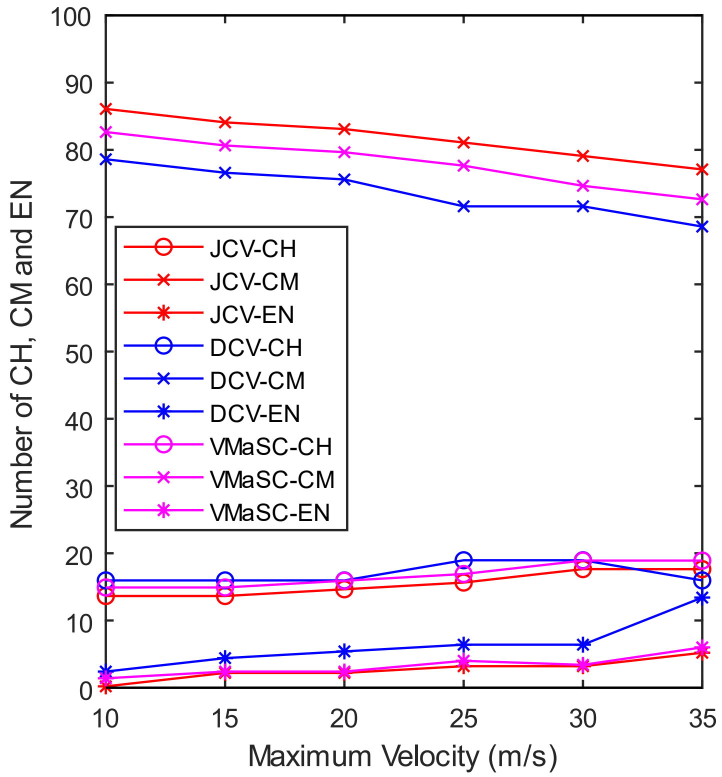

Figure 6.4: Impact of vehicle velocity on number of CH, CM, and EN.

Figure 6.5 shows the average duration of CH, CM and EN in terms of vehicle velocity. The average duration for CH and CM for all three algorithms is relatively closer and much higher than the average duration of EN. This figure also shows that the tendency of the protocols is similar in nature. The average CH duration and the average CM duration gradually decrease whereas the average EN duration gradually increases.

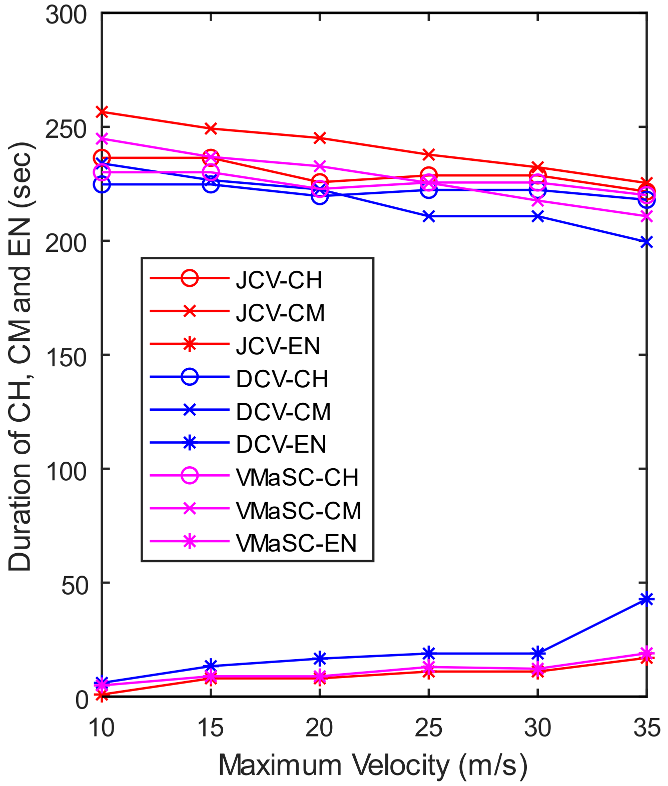

Figure 6.5: Impact of vehicle velocity on duration of CH, CM, and EN.

Figure 6.6 shows the average number of vehicles in clustered states versus the average number of vehicles in the non-clustered states; our target is to get more vehicles in the clustered state. All three algorithms show a similar pattern. The average number of vehicles in clustered state slightly decreases at a higher speed. However, all three algorithms achieve a higher number of vehicles in the clustered state compared to the number of vehicles in EN state.

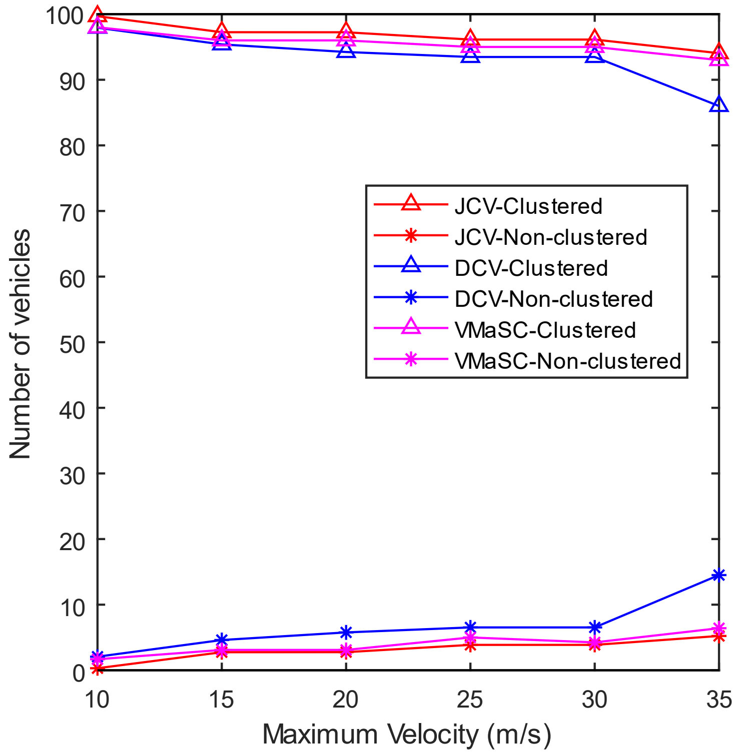

Figure 6.6: Impact of vehicle velocity on Vehicles in cluster.

We see that the graphs generated by evaluating the performance of JCV are similar pattern of the graphs generated by evaluating the performances of VMaSC and DCV. We will now present the detailed comparison results for JCV with VMaSC and DCV for different stability metrics.

6.1.1 Average CH Duration

Figure 6.7 shows the average duration of CH for different maximum velocities. JCV achieves a higher average duration for all different maximum velocities compared to VMaSC and DCV.

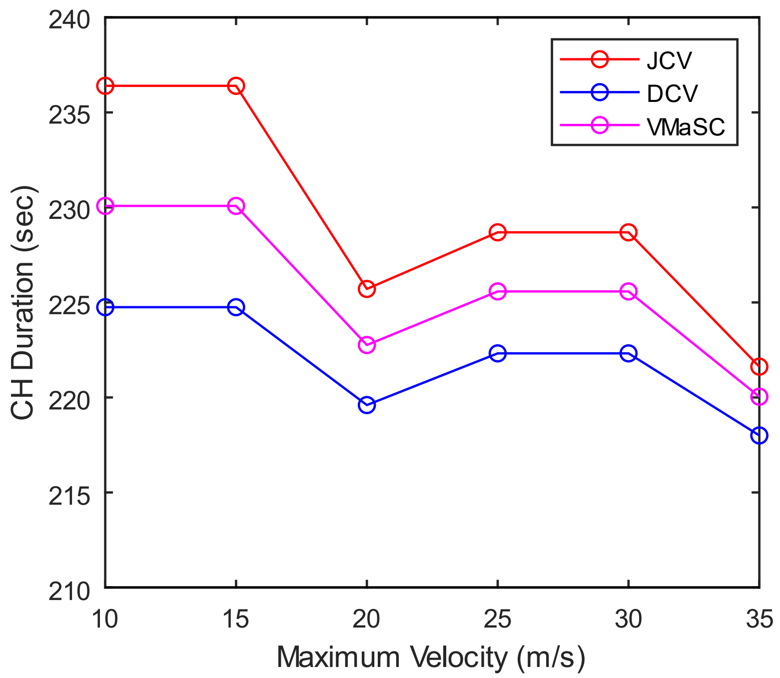

Figure 6.7: Average CH duration vs maximum velocity.

The difference of the average duration compared to the other two algorithms decreases when the velocity of the vehicles increases; however, the value of JCV remains always higher than VMaSC and DCV. This is achieved because of the intelligent approach of CH selection. Firstly, the vehicle in a stable position gets preference to become CH. The lowest relative distant vehicle is chosen as the CH. Secondly, even different direction vehicle can join in a cluster that increases a vehicle to get cluster members for a higher duration. Finally, a vehicle which will take the same route at the next junction is considered to become the CH, hence, after becoming CH, a vehicle remains in the CH state for a longer time.

6.1.2 Average CM Duration

Figure 6.8 shows the average duration of CM for different maximum velocities. JCV achieves a higher average duration for CM for varying velocity compared to both VMaSC and DCV. Even though the average duration decreases at the higher velocities, JCV maintains consistency and achieves higher values.

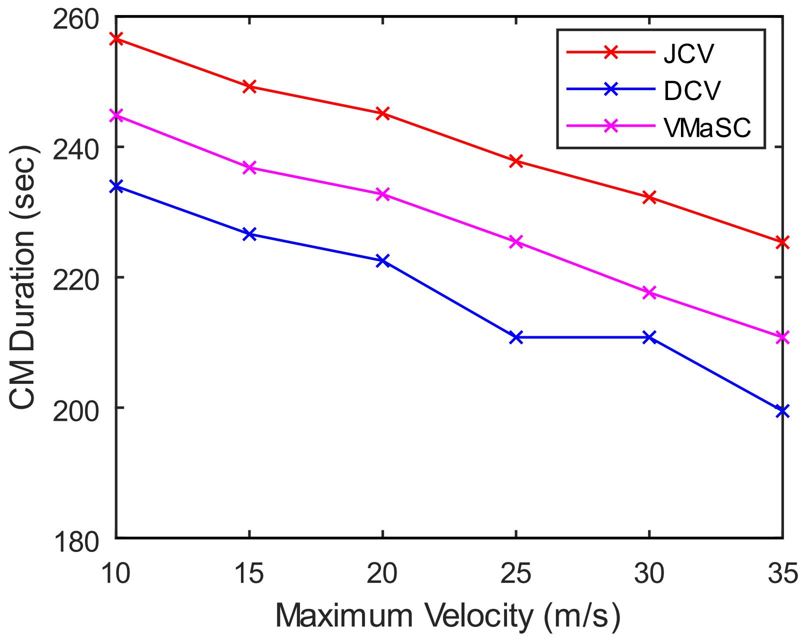

Figure 6.8: Average CM duration vs maximum velocity.

The higher average duration of CM is achieved because of the robust approach of CM joining a cluster. Firstly, a vehicle from the same direction can join the cluster. Secondly, a vehicle from a different direction at a junction can also join a cluster if their paths are the same up to the next junction. This increases the probability of a vehicle to get a CH to join and therefore increases the average duration for the CM.

6.1.3 Average CH Change Rate

Figure 6.9 shows the average CH change rate per second of JCV compared to VMaSC and DCV. A lower average CH change rate means the cluster is more stable. JCV achieves only 7% change rate in lower velocity and increases up to 11% CH change rate where VMaSC and DCV achieve up to 13% change depending on the velocity. Consideration of movement at the intersections reduces CH change in JCV. A lower average change rate is more beneficial for the stability of the clusters if the average CH duration remains high.

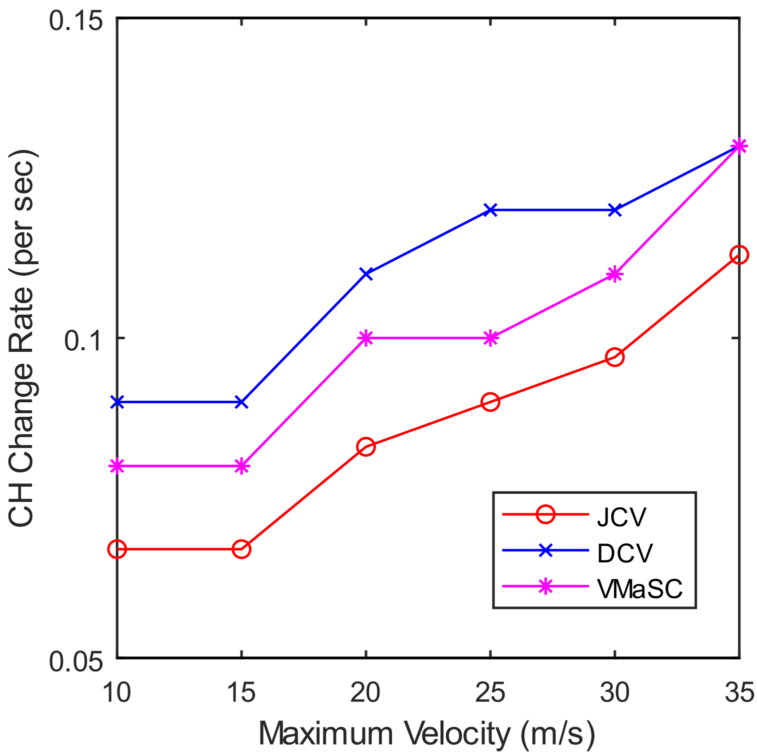

Figure 6.9: Average CH change rate vs maximum velocity.

6.1.4 Average Clustering Overhead

Figure 6.10 shows that JCV results in much lower average clustering overhead. This is achieved because of the lower number of states and simpler state transition in JCV compared to DCV and VMaSC. While VMaSC used five states and nine state transitions, and DCV used four states and eight state transitions, JCV minimized the number of states into three and simplified state transitions to six. This reduction and simplification reflect in the clustering overhead.

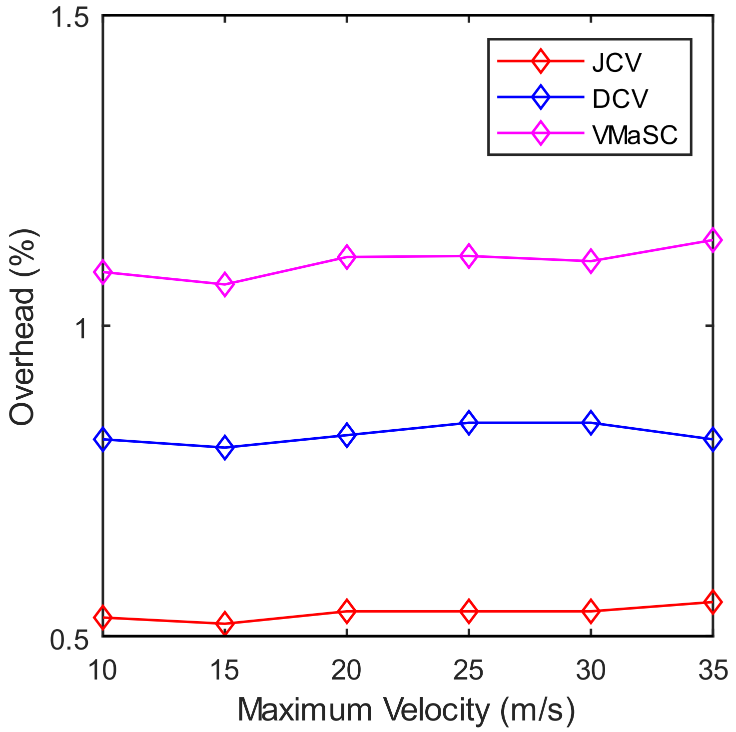

Figure 6.10: Overhead vs maximum velocity.

6.1.5 Average Number of Clusters

Figure 6.11 shows that JCV creates a lower average number of clusters compared to VMaSC and DCV in most of the cases. When the velocity is 35 m/s, DCV creates a lower average number of clusters because DCV creates a greater number of un-clustered vehicles at this stage since it uses a safe distance threshold which is smaller than the TR. The lower number of clusters is coming from the result of the lower number of cluster participation which is shown in the next figure. Therefore, even though DCV shows a lower number of clusters at one stage, it cannot be considered as the strength of DCV.

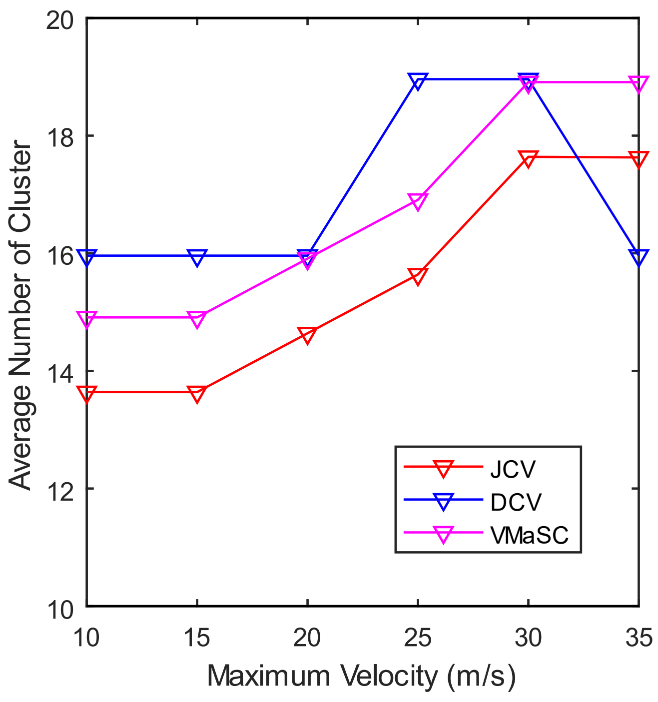

Figure 6.11: Average number of cluster vs maximum velocity.

6.1.6 Percentage of Cluster Participation

Figure 6.12 compares the number of cluster participation in percentage. We can see that JCV includes a greater number of vehicles in clusters. In higher velocity, VMaSC shows similar performance but JCV remains on higher side; however, DCV creates clusters with a greater number of unclustered vehicles, especially, when the velocity increases to 35 m/s. We told about this in the previous figure that DCV creates many single-cluster at different point.

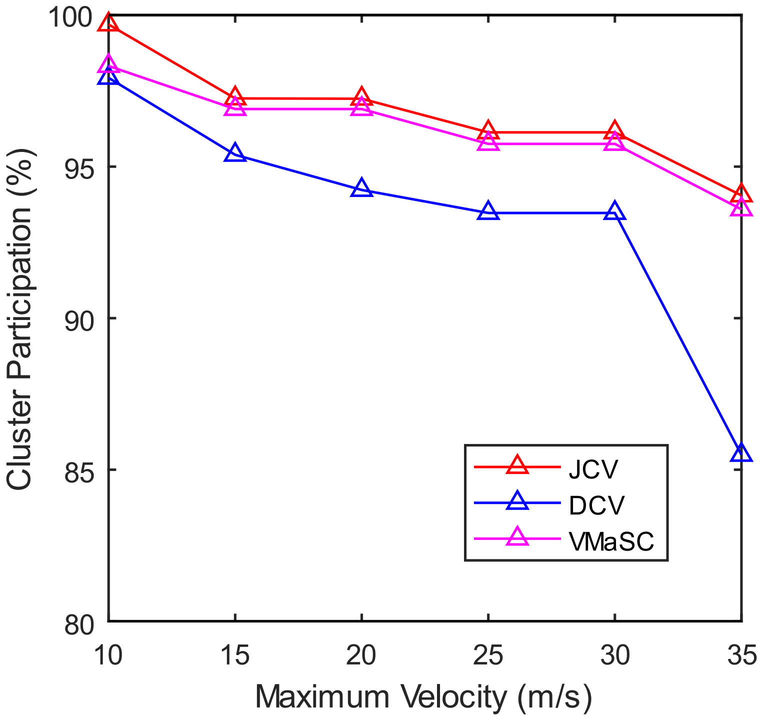

Figure 6.12: Percentage of cluster participation vs maximum velocity.

6.1.7 Average Number of CM

Figure 6.13 shows the average number of CM for all three algorithms. JCV achieves a higher number compared to VMaSC and DCV. A greater number of vehicles in CM state reduce the number of contenders for the resources when traffic is high.

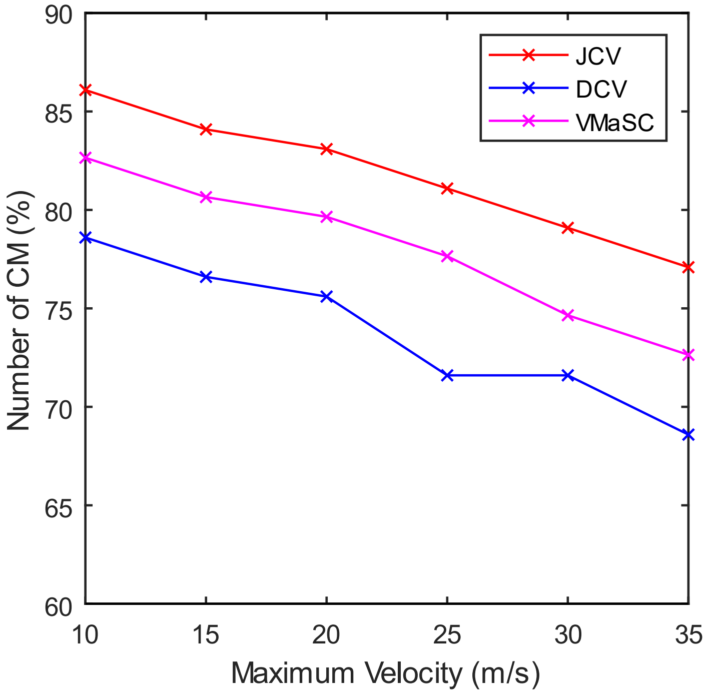

Figure 6.13: Number of CM vs maximum velocity.

6.1.8 Average Ratio of CM

Figure 6.14 shows the average ratio of CM in percentage. It also shows that JCV creates a greater number of CM compared to the number of CH and EN jointly. This is achieved because the probability of a vehicle being in a cluster is higher in JCV.

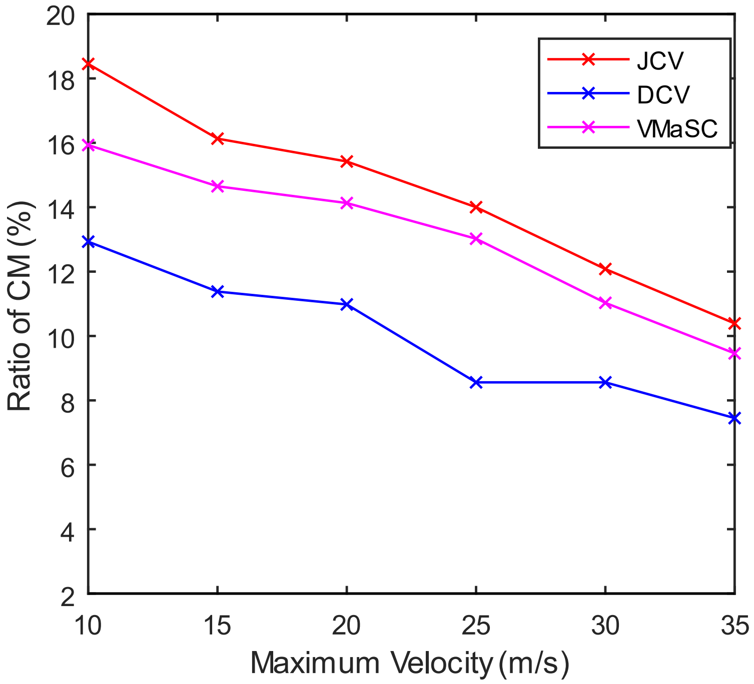

Figure 6.14: Ratio of CM vs maximum velocity.

6.1.9 Average Delay

Figure 6.15 shows the average packet transmission delay of JCV compared to DCV and VMaSC. Since JCV creates lower clustering overhead, therefore, the transmission channel remains open for a longer time for packet transmission. Hence, JCV achieves lower delay for transmitting a packet from a source to a destination. JCV requires 10% and 30% less time for packet transmission compared to DCV and VMaSC, respectively.

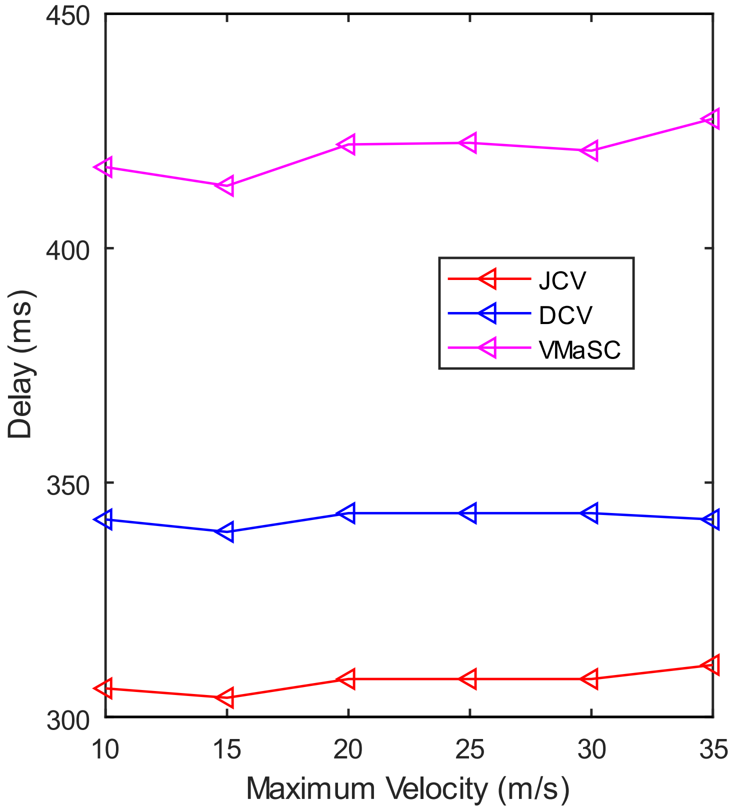

Figure 6.15: Average Delay vs maximum velocity.

6.1.10 The Average Number of Un-clustered Vehicle (EN)

Figure 6.16 shows the comparison of the average number of non-clustered vehicles in percentage for different maximum velocities for JCV, VMaSC, and DCV. Figure 6.17 shows the average duration of EN for JCV compared to VMaSC and DCV.

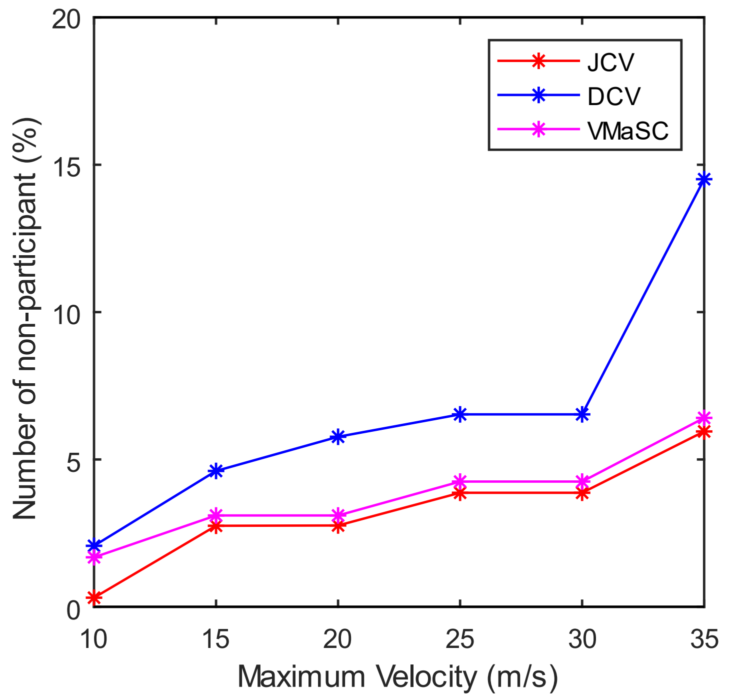

Figure 6.16: Percentage of non-participant vehicle vs maximum velocity.

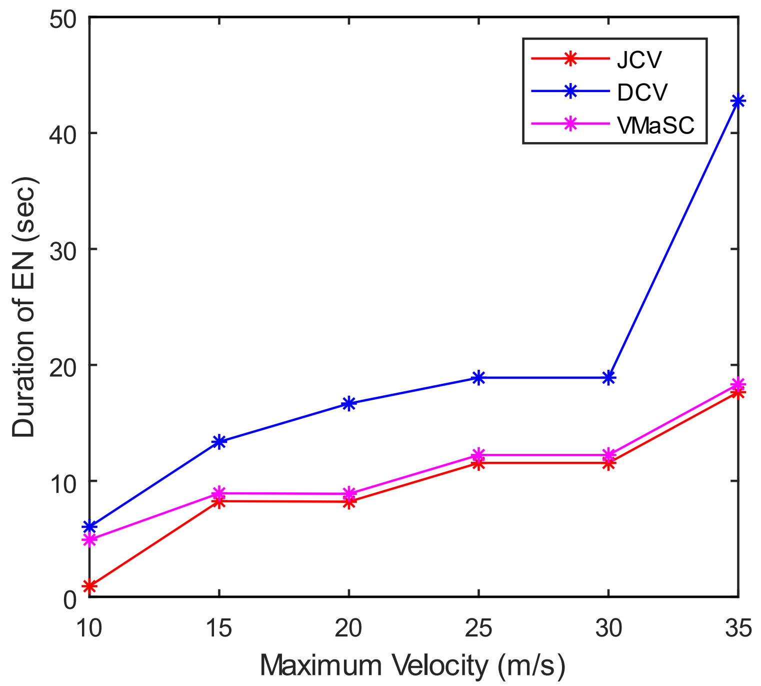

Figure 6.17: Duration of EN vs maximum velocity.

Figure 6.16 and Figure 6.17 show a similar trend and the numbers only differ. JCV creates the lowest average number of EN and the lowest average duration of EN. VMaSC shows a similar pattern to JCV at the higher velocities while achieves a similar value of DCV at a lower velocity. On the other hand, DCV creates a similar pattern with VMaSC for a lower velocity while creates a much worse values at the higher velocities. However, JCV creates the lowest number and duration of EN because of a more intelligent, robust, and inclusive approach where more vehicles are given the opportunity to join in a cluster.

6.2 Results of SDPC

The results of SDPC are compared with VMaSC, DCV, IBC and ECHS. At first, we presented some case studies on SDPC, VMaSC, DCV, IBC and ECHS with their flowcharts. Then, the simulation results are presented where the performance of SDPC is compared with VMaSC, DCV, ECHS and IBC.

6.2.1 Case Study

The flowchart of clustering process in VMaSC, DCV, ECHS, IBC, and SDPC have been presented in Figure 6.18, Figure 6.19, Figure 6.20, Figure 6.21, and Figure 6.22, respectively.

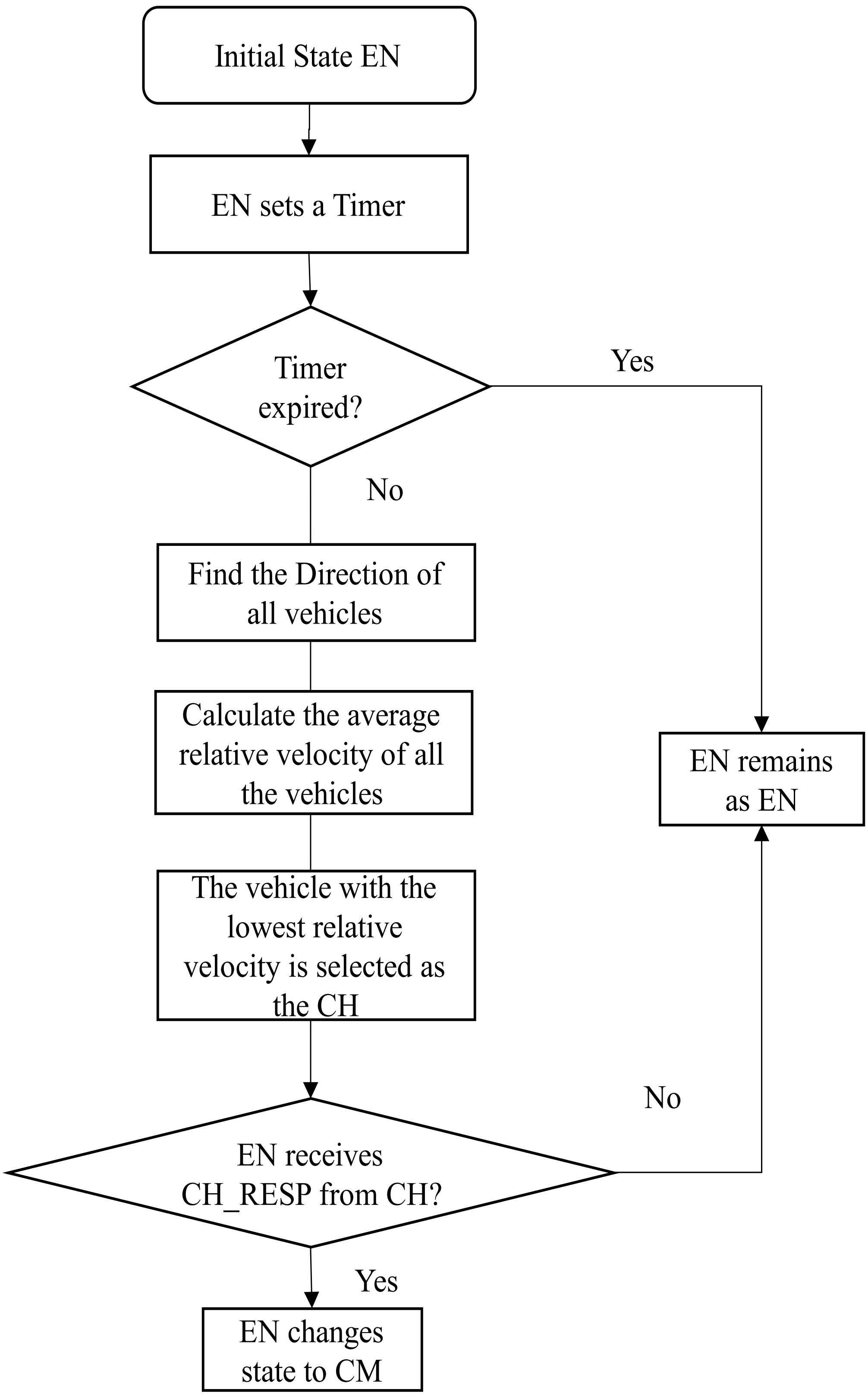

Figure 6.18: Flowchart of VMaSC.

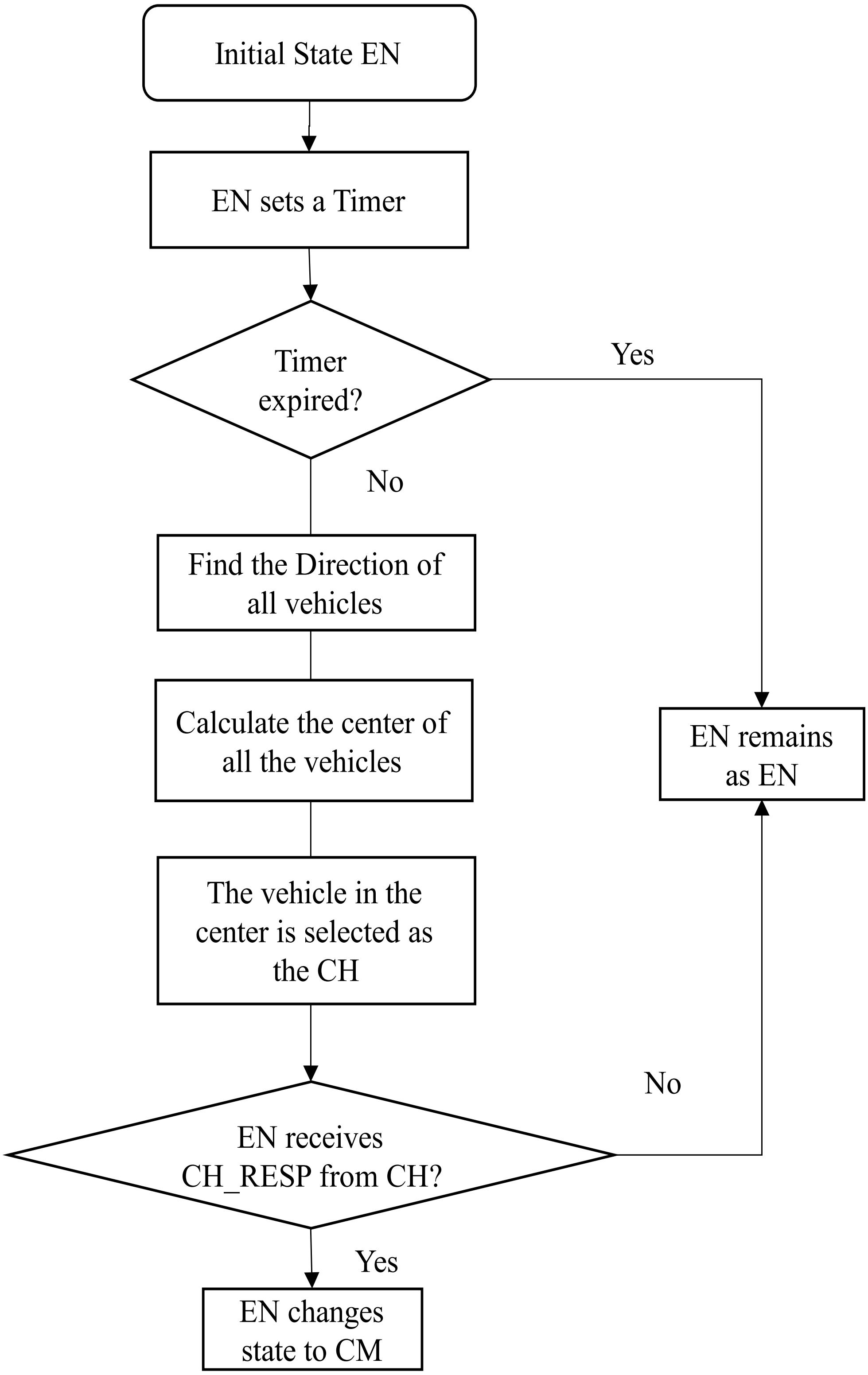

Figure 6.19: Flowchart of DCV.

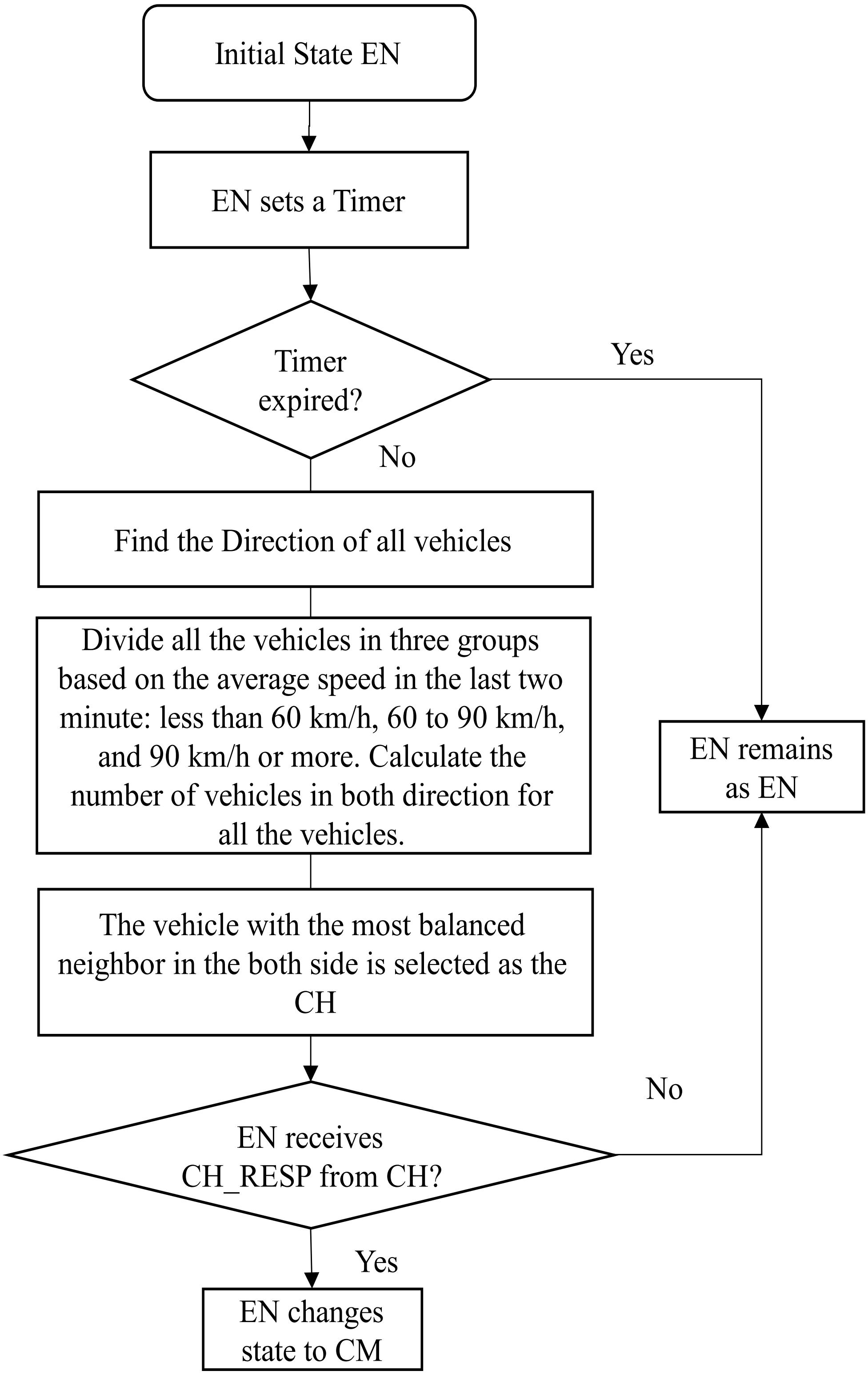

Figure 6.20: Flowchart of ECHS.

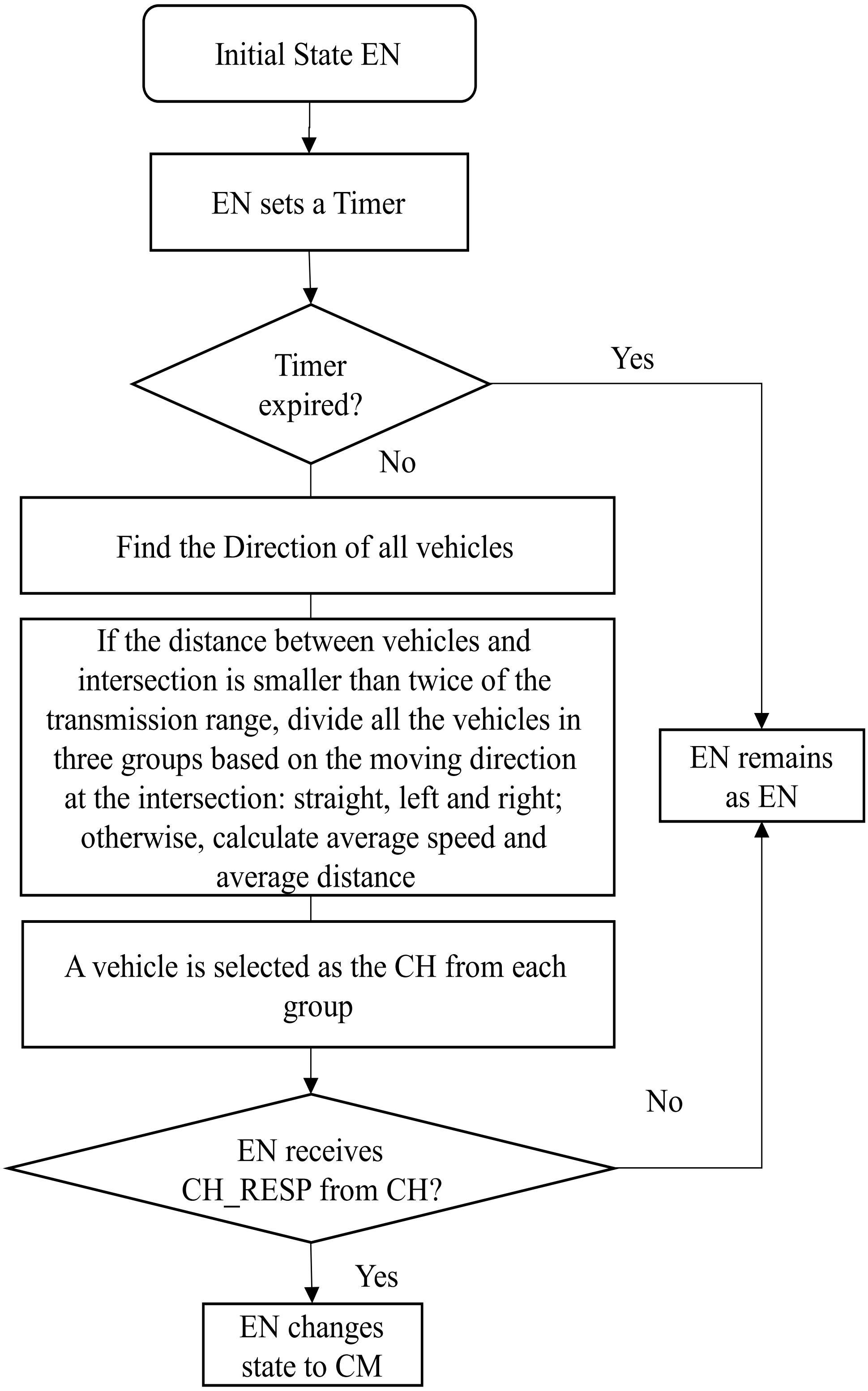

Figure 6.21: Flowchart of IBC.

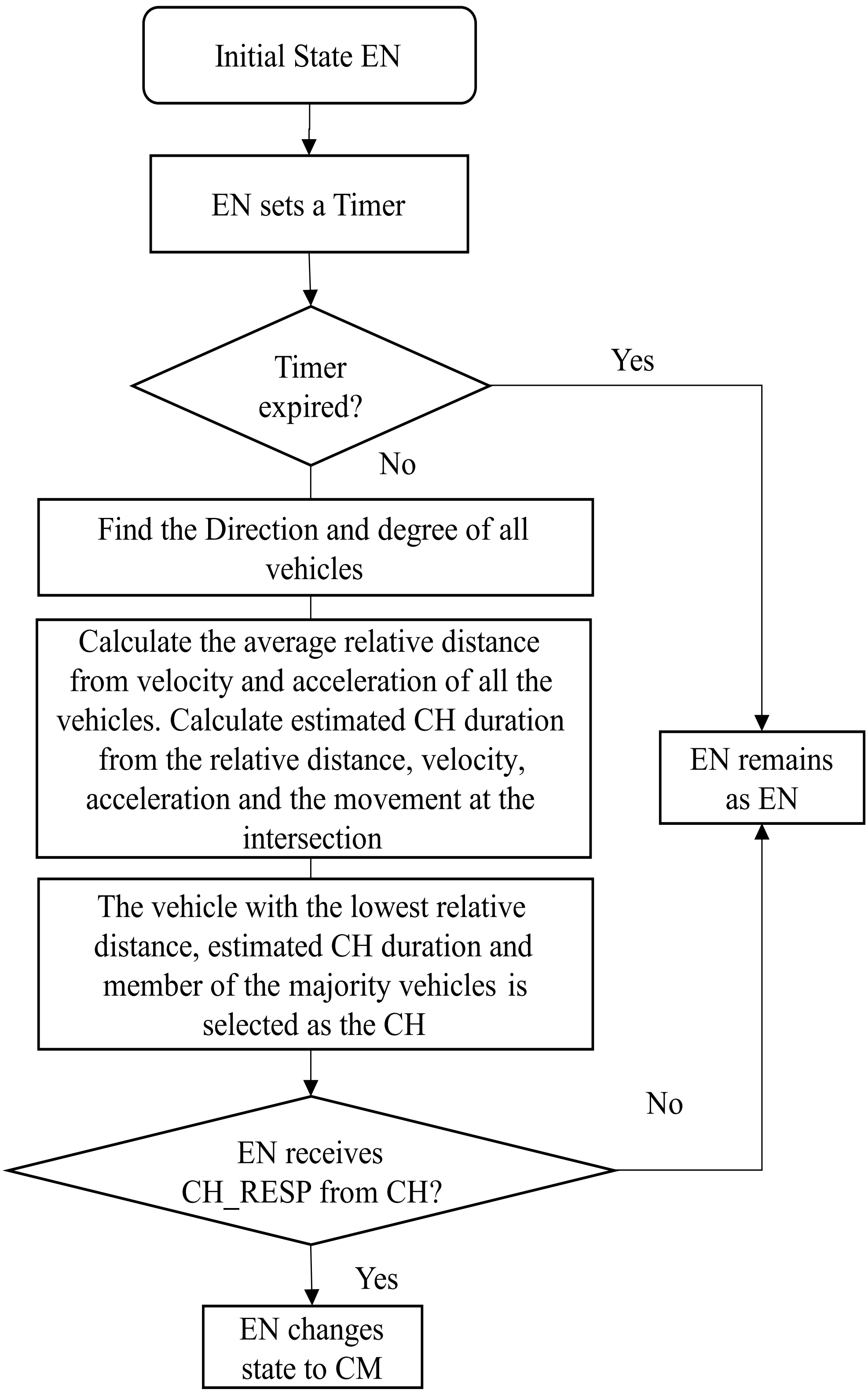

Figure 6.22: Flowchart of SDPC.

We will now describe multiple cases to analyze the performance of five clustering protocols. We will choose the CH and will create the clusters based on the clustering formation process and the CH selection criteria for different scenarios to evaluate the performance of the protocols.

Case 1:

Figure 6.23 and Figure 6.24 show the scenario for case 1. Seven vehicles are travelling from the left to the right at the same velocity, and all the vehicles are within the transmission range of the other vehicles, $u_i(0) = 10$, $a_i(0) = 0$, and $s_{ij}(0) < 200$ m, where $0 < i < 8$ and $0 < j < 8$.

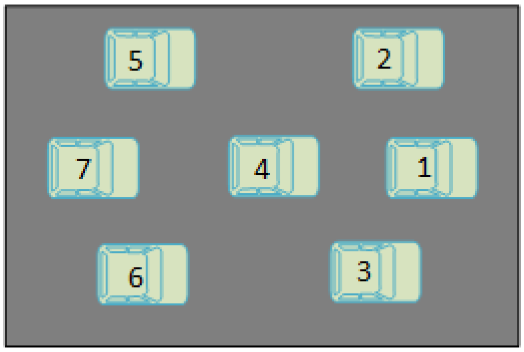

Figure 6.23: Vehicles running at the same velocity.

Therefore, $u_i(t) = 10$, and $a_i(t) = 0$, and $s_{ij}(t) < 200$ m.

- VMaSC selects the vehicle with the lowest $\overline{u}_i(t)$ as the CH, therefore, a possible $CH_{VMaSC} = V_4$
- DCV selects the vehicle that geographically middle or the lowest $\overline{s}_{ij}(t)$ as the CH, therefore, $CH_{DCV} = V_4$

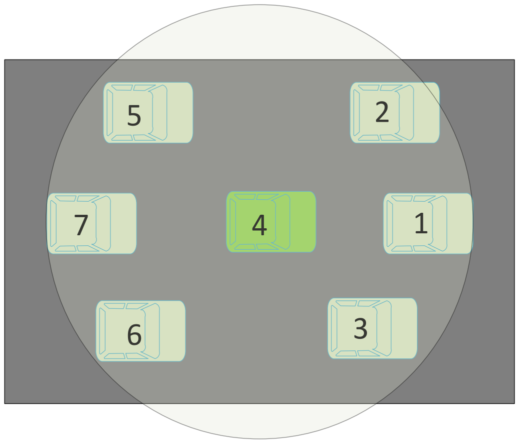

Figure 6.24: Cluster formation and CH selection for Case 1.

- ECHS selects the vehicle with even density or even degree of vehicles as the CH, therefore, $CH_{ECHS} = V_4$
- IBC selects the vehicle with the highest selection metric as the CH, therefore, $CH_{IBC} = V_4$
- SDPC selects the vehicle with the highest predicted stability as the CH, then considers the relative velocity and position, and the degree of vehicle, therefore, $CH_{SDPC} = V_4$

In this case, the CH for DCV, ECHS, IBC, and SDPC is the same, i.e., V_4. For VMaSC, it can be V_4 or any vehicle from the seven vehicles since their relative velocities are the same. For all the algorithms, CH change rate, CH duration and clustering overhead are the same because here the positions are not changing further.

Case 2:

Vehicles are travelling from the left to the right at the same velocity, but all the vehicles are not within the transmission range of all other vehicles, i.e., $s_{ij}(t) > 200$ m for some vehicles. The distance between V_1 and V_5, V_1 and V_6 and V_1 and V_7 are greater than 200 m as shows in Figure 6.25.

$u_i(0) = 10$, $a_i(0) = 0$, and $s_{1j}(0) > 200$ m, where $4 < j < 8$.

Also, $u_i(t) = 10$, and $a_i(t) = 0$, and $s_{1j}(0) > 200$ m, where $4 < j < 8$.

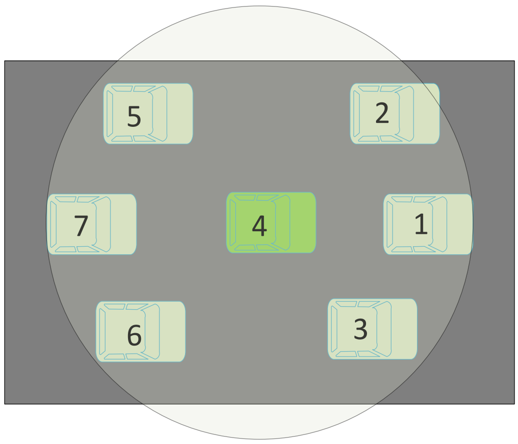

Figure 6.25: Cluster formation and CH selection for DCV, ECHS, IBC and SDPC for Case 2.

In this case, the CH for DCV, ECHS, IBC, and SDPC is the same, i.e., V_4. Since $s_{4j}(0) < 200$ m where $0 < j < 8$, these four algorithms create only one cluster.

- VMaSC selects the vehicle with the lowest $\overline{u}_i(t)$ as the CH, therefore, VMaSC can choose any vehicle but there is a high chance that the first vehicle is selected as the CH, i.e., $CH_{VMaSC} = V_1$
- DCV selects the vehicle that geographically middle or the lowest $\overline{s}_{ij}(t)$ as the CH, therefore, $CH_{DCV} = V_4$
- ECHS selects the vehicle with even density or even degree of vehicles as the CH, therefore, $CH_{ECHS} = V_4$

- IBC selects the vehicle with the highest selection metric as the CH, therefore, $CH_{IBC} = V_4$
- SDPC selects the vehicle with the highest predicted stability as the CH, then considers the relative velocity and position, and then the degree of vehicle, therefore, $CH_{SDPC} = V_4$

In VMaSC, there is a high chance that V_1 is selected as the CH as it enters the road before the other vehicles, otherwise VMaSC will create some random clusters as shown in Figure 6.26.

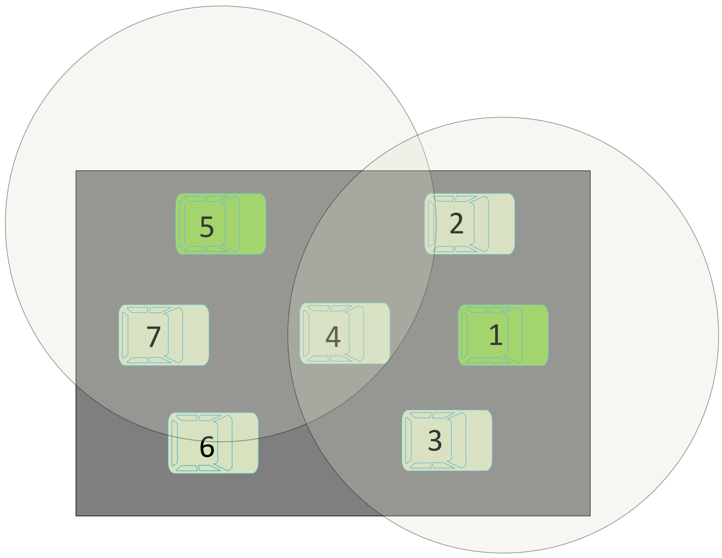

Figure 6.26: Cluster formation and CH selection for VMaSC for case 2.

For all five algorithms, CH change rate, and CH duration are the same because the relative positions of the vehicles are not changing further. VMaSC creates multiple clusters, as a result, clustering overhead for VMaSC is increased than other four algorithms.

Case 3:

Figure 6.27, Figure 6.28 and Figure 6.29 present the scenarios for Case 3. The vehicles are travelling from the left to the right at the same velocity, but some vehicles are not in the transmission range of the other vehicles, i.e., $s_{ij}(t)$ > 200 m for some vehicles. V_1 to V_6 are in the transmission range of each other; however, V_7 is not in the transmission range of V_1 to V_3 and the distance between V_4 and V_7 is close to 200 m.

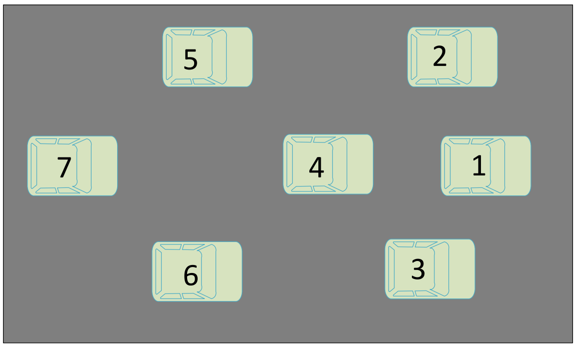

Figure 6.27: Vehicles with 200 m distance among some vehicles.

$u_i(0) = 10$, $a_i(0) = 0$, and $s_{47}(0) \approx 200$ m.

Also, $u_i(t) = 10$, and $a_i(t) = 0$, and $s_{47}(t) \approx 200$ m.

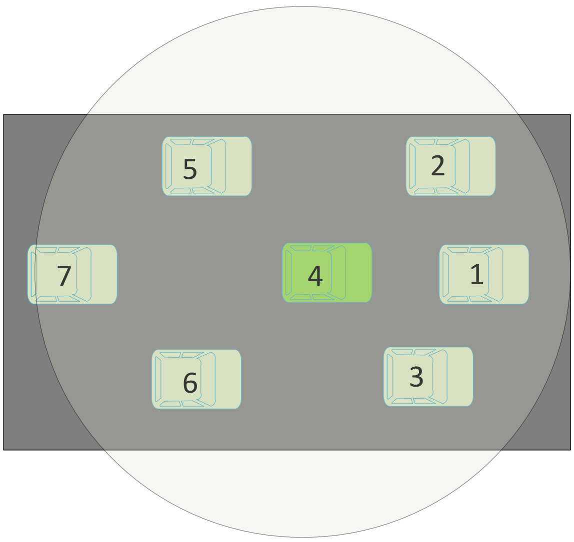

Figure 6.28: Clustering for VMaSC, ECHS, IBC, and SDPC.

- VMaSC selects the vehicle with the lowest $\overline{u}_i(t)$ as the CH, therefore, a possible $CH_{VMaSC} = V_4$
- DCV selects the vehicle that geographically middle or the lowest $\overline{s}_{ij}(t)$ as the CH, therefore, $CH_{DCV} = V_4$
- ECHS selects the vehicle with even density or even degree of vehicles as the CH, therefore, $CH_{ECHS} = V_4$
- IBC selects the vehicle with the highest selection metric as the CH, therefore, $CH_{IBC} = V_4$
- SDPC selects the vehicle with the highest predicted stability as the CH, then considers the relative velocity and position, and then the degree of vehicle, therefore, $CH_{SDPC} = V_4$

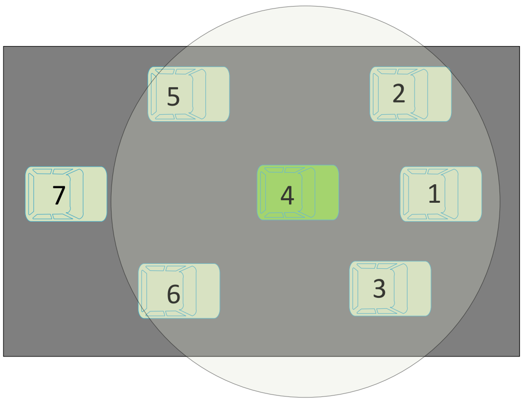

Figure 6.29: Cluster coverage of CH in DCV.

In this case, ECHS, IBC, and SDPC will create one cluster with V_4 as the CH. If VMaSC chooses V_4 as the CH, then it will also create one cluster; however, V_4 in DCV cannot cover V_7, since DCV uses a *TR* which is less than the actual *TR*. i.e., $TR_{DCV} < TR$.

Therefore, DCV creates one cluster and abandons a vehicle, i.e., one cluster plus one single-cluster.

Case 4:

The vehicles are travelling from the left to the right and the positions of the vehicles are V_1 (200, 12), V_2 (180, 4), V_3 (160, 8), V_4 (140, 12), and V_5 (120, 4) and all vehicles are within the *TR* of the other vehicles at $t = 0$ as shown in Figure 6.30.

$s_{ij}(0) < TR$ at $t = 0$ and $TR = 200$ m.

$u_i(0) = (10, 6, 10, 2, 2)$

$a_i(0) = (1, 2, 1, 2, 1)$

Now, $\overline{s}_i(0) = (50.53, 35.68, 30.3, 35.68, 50.53)$.

Maximum velocity = 30 m/s and the length of the lane is 1000 m between the two intersections. We select the CH according to the five protocols and create clusters at $t = 0$. Then we evaluate the performance of the protocols based on the mobility parameters. We have taken into consideration that DCV has a lower *TR* than the actual transmission range, and IBC re-clusters at 2**TR* distance from the intersection. We evaluate the CH duration before the first change in the cluster, i.e., in this case, when the first vehicle leaves the cluster.

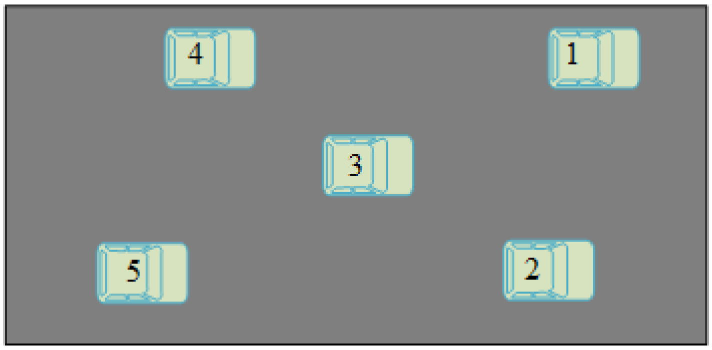

Figure 6.30: Vehicles with different velocity and acceleration for Case 4.

If we calculate the time for all five vehicles, ignoring the difference in *y*-axis since the lane-width is negligible compared to the long distance in *x*-axis and vehicles can change lane and can run in the same lane, we will get the CH duration for five algorithms. If we analyze the performance of each individual vehicle based on the velocity, acceleration, position, and distance of the vehicles, we will get the CH duration for vehicles V_1 to V_5.

If V_1 is selected as the CH of this cluster, then the CH duration is 15 sec. Because at t = 16, the position of V_1 and V_5 becomes (496, 12) and (288, 4), i.e., the distance between V_1 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_2 is selected as the CH of this cluster, then the CH duration is 12 sec. Because at t = 13, the position of V_2 and V_5 becomes (438, 4) and (237, 4), i.e., the distance between V_2 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_3 is selected as the CH of this cluster, then the CH duration is 20 sec. Because at t = 21, the position of V_3 and V_5 becomes (600, 8) and (393, 4), i.e., the distance between V_3 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_4 is selected as the CH of this cluster, then the CH duration is 21 sec. Because at t = 22, the position of V_4 and V_5 becomes (618, 12) and (417, 4), i.e., the distance between V_4 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_5 is selected as the CH of this cluster, then the CH duration is 21 sec. Because at t = 22, the position of V_4 and V_5 becomes (618, 12) and (417, 4), i.e., the distance between V_4 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

Now we will investigate the CH duration for VMaSC, DCV, ECHS, IBC, and SDPC. Based on the CH selection algorithm, five clustering protocols choose a set of different vehicles as the CH.

- VMaSC selects the vehicle with the lowest $\overline{u}_i(t)$ as the CH, therefore, $CH_{VMaSC} = V_2$
- DCV selects the vehicle that geographically middle or the lowest $\overline{s}_{ij}(t)$ as the CH, therefore, $CH_{DCV} = V_3$
- ECHS selects the vehicle with even density or even degree of vehicles as the CH, therefore, $CH_{ECHS} = V_3$

- IBC selects the vehicle with the highest selection metric as the CH, therefore, $CH_{IBC} = V_4$ or V_5
- SDPC selects the vehicle with the highest predicted stability as the CH, then considers the relative velocity and position, and then the degree of vehicle, therefore, $CH_{SDPC} = V_4$

Now, based on the CH selected by VMaSC, DCV, ECHS, IBC, and SDPC, we get the following.

1. The CH duration for VMaSC is 12 sec.
2. The CH duration for DCV is 17 sec and the CH duration for ECHS is 20 sec. Even though DCV and ECHS choose the same vehicle as the CH in this case, the transmission range of DCV is less than the transmission range of ECHS and the cluster of DCV breaks earlier. As a result, two CH duration differ.
3. The CH duration for IBC is 18 sec and the CH duration for SDPC is 21 sec. Again, IBC and SDPC choose the same vehicle as the CH in this case; however, IBC triggers re-clustering when a vehicle reaches 2*TR distance from the intersection. In this case, V_2 reaches at (618, 4) at $t = 19$, i.e., it reaches at the point where the distance from the intersection becomes less than 2*200 m and cluster breaks while SDPC continues up to 21 sec.

Therefore, SDPC is achieving higher CH duration compared to VMaSC, DCV, ECHS, and IBC algorithms.

Case 5:

We made a small modification from case 4 as shown in Figure 6.31; however, the differences in results are significant. We changed the value of $u_5(0)$ from 2 to 3. The positions of the vehicles are V_1 (200, 12), V_2 (180, 4), V_3 (160, 8), V_4 (140, 12), and V_5 (120, 4) and all vehicles are within the *TR* of the other vehicles for all five algorithms.

$s_{ij}(0) < TR$ at $t = 0$ and $TR = 200$ m.

Now, $\bar{s}_i(0)$ = (50.53, 35.68, 30.3, 35.68, 50.53).

$u_i(0)$ = (10, 6, 10, 2, 3)

$a_i(0)$ = (1, 2, 1, 2, 1)

Maximum velocity = 30 m/s and the length of the lane is 1000 m between the two intersections. We select the CH according to the five protocols and create clusters at $t = 0$. We evaluate the performance of the protocols based on the mobility parameters. We have taken into consideration that DCV has a lower *TR* than the actual transmission range, and IBC re-clusters at 2**TR* distance from the intersection. We evaluate the CH duration before the first change in the cluster, i.e., in this case, when the first vehicle leaves the cluster.

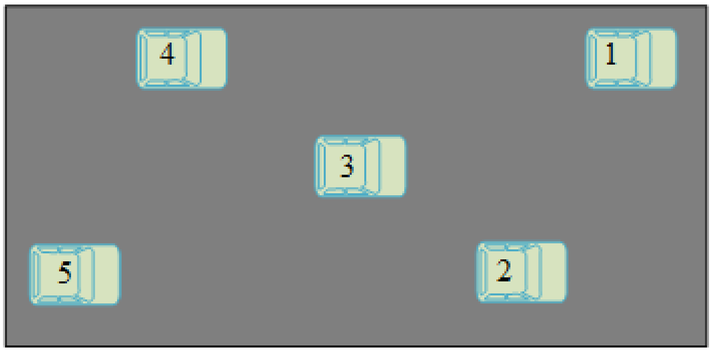

Figure 6.31: Vehicles with different velocity and acceleration for Case 5.

If we calculate the time for all five vehicles, ignoring the difference in *y*-axis since the lane-width is negligible compared to the long distance and vehicles can change lane any time and can run in the same lane, we will get the CH duration for five protocols. If we analyze the performance of each individual vehicle based on the velocity, acceleration, distance, and position of the vehicles, we will find the CH duration for vehicles V_1 to V_5.

If V_1 is selected as the CH of this cluster, then the CH duration is 17 sec. Because at t = 18, the position of V_1 and V_5 becomes (551, 12) and (345, 4), i.e., the distance between V_1 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_2 is selected as the CH of this cluster, then the CH duration is 13 sec. Because at t = 14, the position of V_2 and V_5 becomes (468, 4) and (267, 4), i.e., the distance between V_2 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_3 is selected as the CH of this cluster, then the CH duration is 25 sec. Because at t = 26, the position of V_3 and V_5 becomes (750, 8) and (549, 4), i.e., the distance between V_3 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_4 is selected as the CH of this cluster, then the CH duration is 149 sec. Because at t = 150, the position of V_4 and V_5 becomes (4458, 12) and (4269, 4), i.e., the distance between V_4 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

If V_5 is selected as the CH of this cluster, then the CH duration is 149 sec. Because at t = 150, the position of V_4 and V_5 becomes (4458, 12) and (4269, 4), i.e., the distance between V_4 and V_5 exceeds 200 m of the transmission range and cluster breaks or at least re-clustering triggered.

Now we will investigate the CH duration for VMaSC, DCV, ECHS, IBC, and SDPC. Based on the CH selection algorithm, five clustering protocols choose different vehicles as the CH.

- VMaSC selects the vehicle with the lowest $\overline{u}_i(t)$ as the CH, therefore, $CH_{VMaSC} = V_2$
- DCV selects the vehicle that geographically middle or the lowest $\overline{s}_{ij}(t)$ as the CH, therefore, $CH_{DCV} = V_3$
- ECHS selects the vehicle with even density as the CH, therefore, $CH_{ECHS} = V_3$
- IBC selects the vehicle with the highest selection metric as the CH, therefore, $CH_{IBC} = V_4$ or V_5

- SDPC selects the vehicle with the highest predicted stability as the CH, therefore, $CH_{SDPC} = V_4$

Moreover, DCV has a lower TR and IBC re-clusters at 2**TR* distance from the intersection.

If we calculate the time for all five vehicles ignoring the difference in *y*-axis since the lane-width is negligible compared to the long distance and vehicles can change lanes and can run in the same lane, we will get the CH duration for five algorithms.

1. The CH duration for VMaSC is 13 sec.
2. The CH duration for DCV is 20 sec and the CH duration for ECHS is 25 sec. Even though DCV and ECHS choose the same vehicle as the CH in this case, the transmission range of DCV is less than the transmission range of ECHS and the cluster of DCV breaks earlier. As a result, two CH duration differ.
3. The CH duration for IBC is 18 sec and the CH duration for SDPC is 149 sec Again, IBC and SDPC choose the same vehicle as the CH in this case; however, IBC triggers re-clustering when a vehicle reaches 2**TR* distance from the intersection. In this case, V_2 reaches at (618, 4) at $t = 19$, i.e., it reaches at the point where the distance from the intersection becomes less than 2*200 m and cluster breaks while SDPC continues up to 149 sec.

SDPC is achieving higher CH duration compared to VMaSC, DCV, ECHS, and IBC algorithms; however, the difference between the duration of SDPC and other four algorithms are significantly different. VMaSC, DCV, ECHS, and IBC achieve 13 sec, 20 sec, 25 sec, and 18 sec while SDPC achieves 149 sec.

6.2.2 Impact of Velocity

We presented the comparison of stability metrics in Section 6.2.3 to Section 6.2.6 on different topologies of roads including highways and intersections. At first, we presented results generated by SDPC for CH, CM and EN.

The average duration of CH, CM and EN for SDPC are shown in Figure 6.32 for different maximum velocities. The average CH duration remains very high for SDPC for different velocities even though it slightly decreases at the higher velocities. The duration bounced back at 25 m/s and 30 m/s before going down again at 35 m/s but never crossed the previous high at 10 m/s. Therefore, the graph is going down. Ultimately, if we increase the velocity over 35 m/s, at one stage this graph will become flat. The average CM duration is also very high and gradually decreases for higher velocity. On the other hand, the average EN duration remains at the same range. Even then, the average CH duration and the average CM duration remains very high compared to the average EN duration in every case.

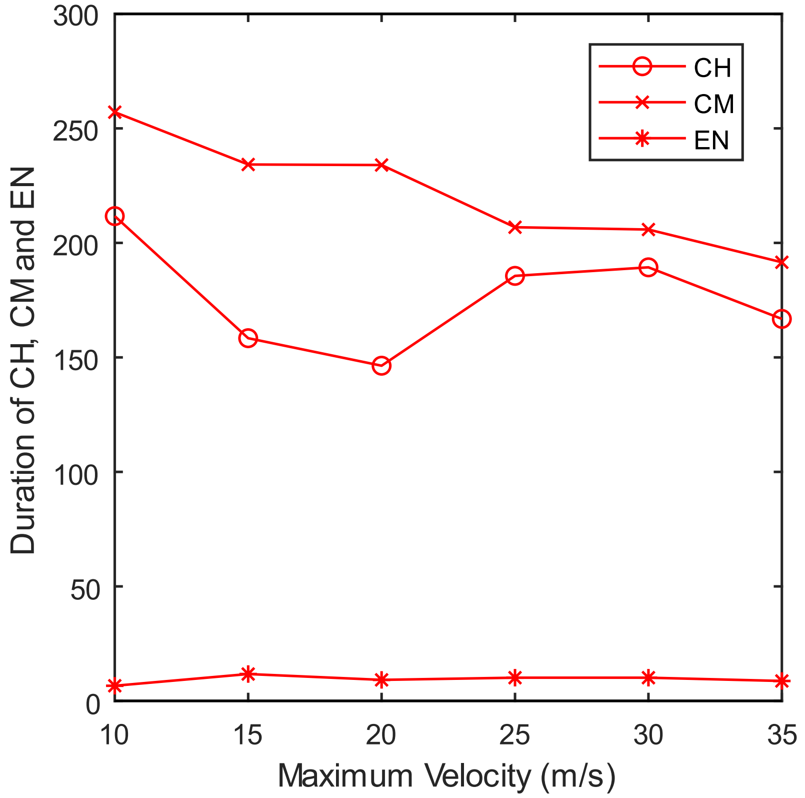

Figure 6.32: Impact of vehicle velocity on SDPC for duration of CH, CM, and EN.

Figure 6.33 shows the average number of CH, CM and EN at the different maximum velocities. It shows that the average number of CH remains always higher than the average number of EN. A lower average number of EN indicates that a few vehicles only remain as EN. At the same time, a large number of vehicles remain as the CM which also represents the stability of the cluster.

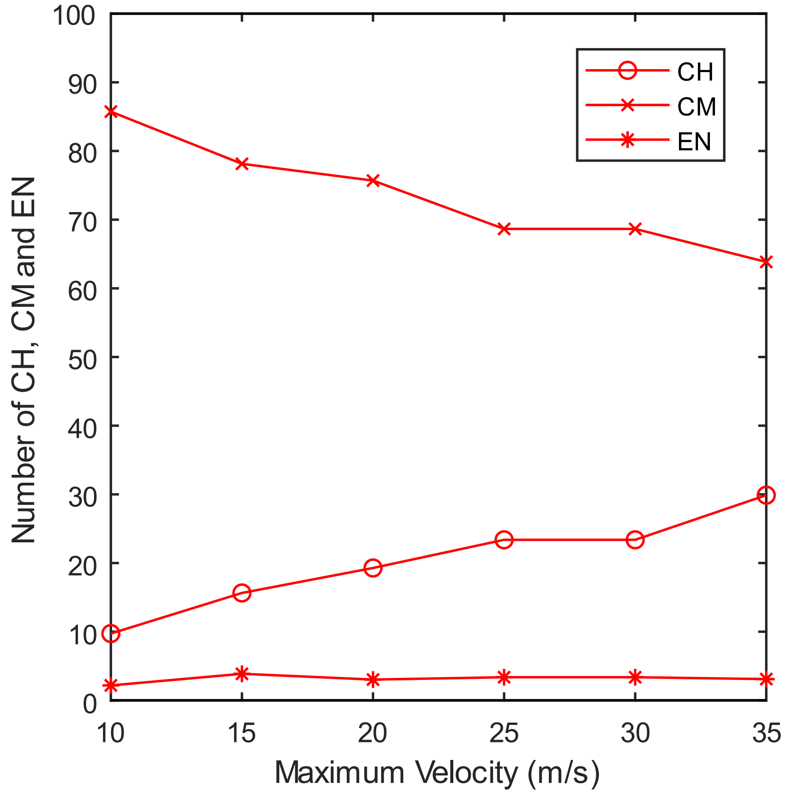

Figure 6.33: Impact of vehicle velocity for SDPC on number of CH, CM, and EN.

Figure 6.34 shows the average number of vehicles in clustered and unclustered states at the different maximum velocities. At higher speeds, vehicles more frequently get out of the transmission range of other vehicles since now vehicles move longer distances every second. We have used car-following models where vehicles maintain two-second distance from other vehicles. Even then, more than 90% of vehicles belong to a cluster, which shows the efficiency of SDPC.

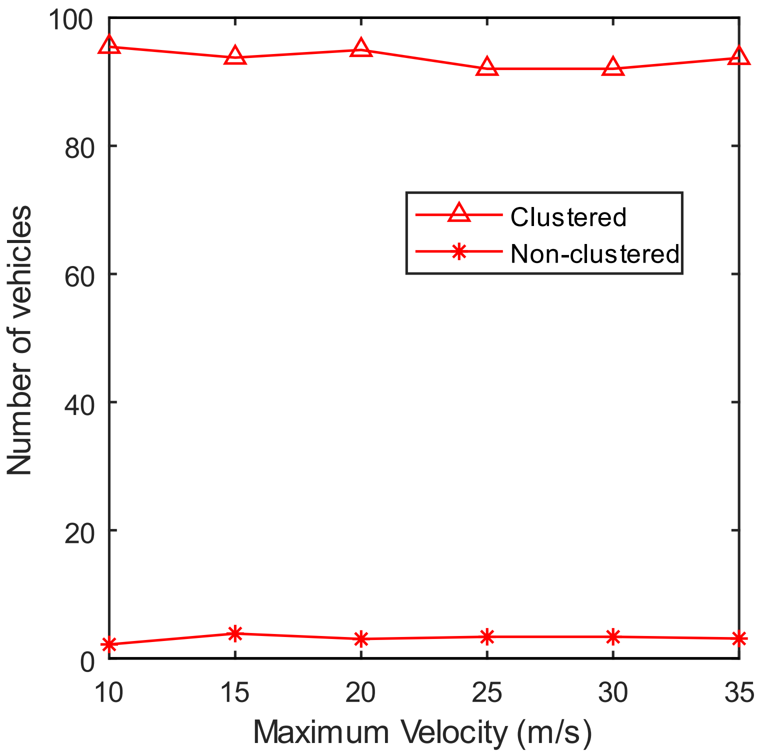

Figure 6.34: Impact of velocity for SDPC on clustering.

Figure 6.35 shows the average duration of CH, CM and EN at the different maximum velocities. We see the pattern of SDPC is like the pattern of other algorithms. The CH durations are going up and down while not crossing the previous high. The durations of CM are continuously going down and the durations of EN remains unchanged.

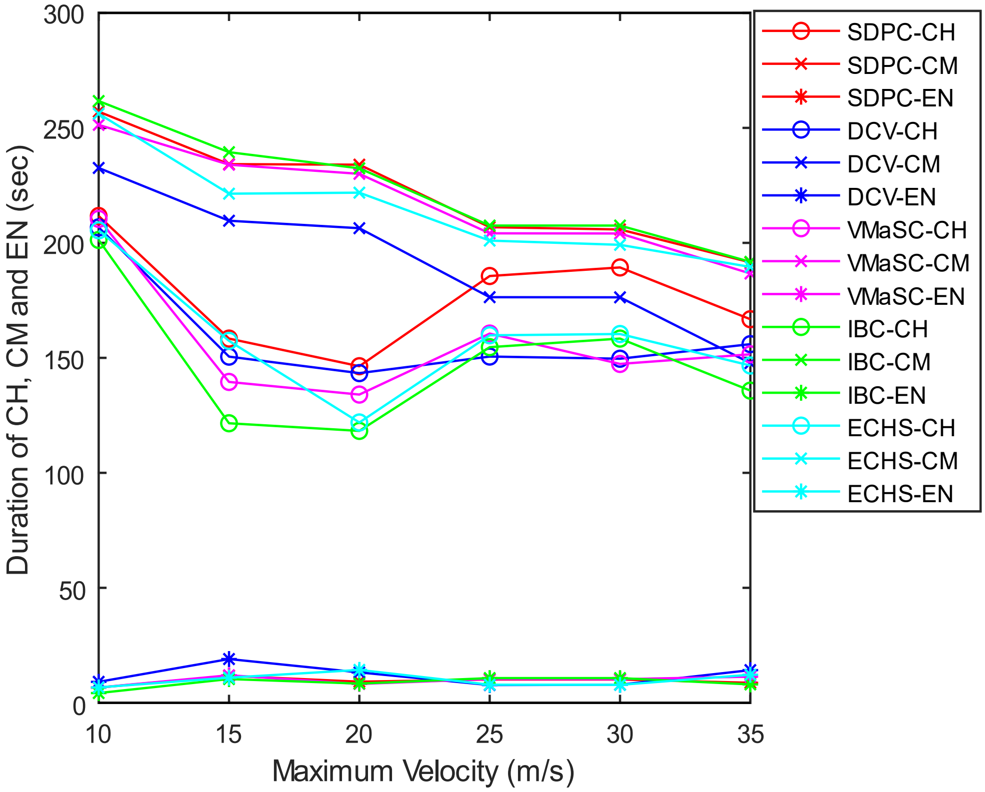

Figure 6.35: Impact of vehicle velocity on duration of CH, CM, and EN.

We will now present the pattern for all five protocols for CH, CM and EN. Figure 6.36 shows the average number of CH, CM, and EN in terms of velocity. The average number of CH increases and number of CM decreases at a higher speed where the number of EN remains at the same level.

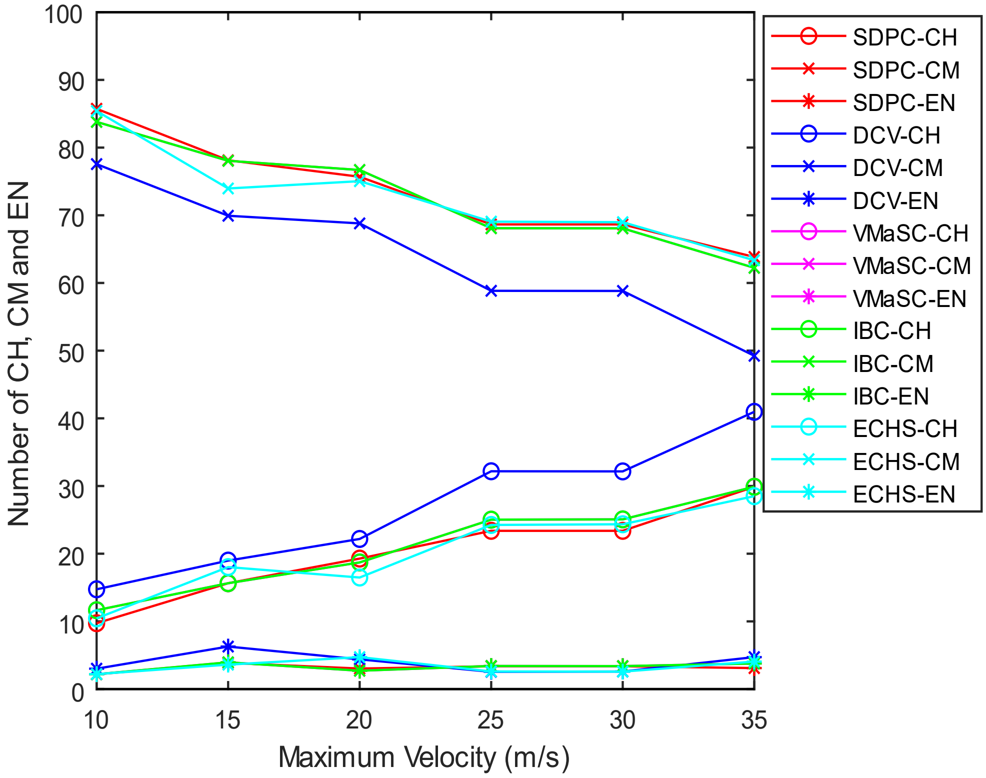

Figure 6.36: Impact of vehicle velocity on number of CH, CM, and EN.

Figure 6.37 shows the average number of vehicles in clustered states versus the average number of vehicles in the non-clustered states since our target is to get more vehicles in the clustered state. We see that protocols show similar patterns and create clusters with a greater number of vehicles compared to the number of EN vehicles.

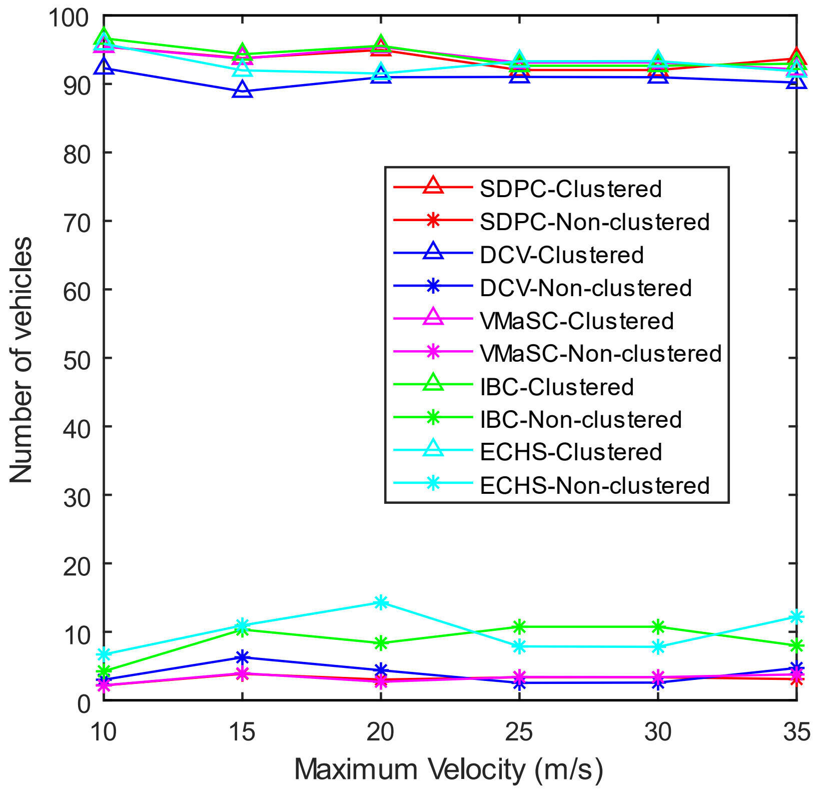

Figure 6.37: Impact of velocity on Vehicles' clustering.

We see that the graphs generated by evaluating SDPC are similar pattern of the graphs generated by evaluating VMaSC, IBC, ECHS and DCV. We will now present the detailed comparison results for SDPC with four other clustering algorithms: VMaSC, ECHS, IBC and DCV using major stability metrics.

6.2.3 Aaverage CH Change Rate (per second)

Figure 6.38 shows the average CH change rate per second of SDPC compared to VMaSC, DCV, IBC and ECHS at different maximum velocities. The differences between SDPC and other algorithms vary for different maximum velocity; however, SDPC always shows lower average CH change rate compared to other four algorithms. A lower average CH change rate means the cluster is more stable. The differences between SDPC and other algorithms are nominal at 10 m/s and 15 m/s, i.e., at the lower velocity of the vehicles, whereas the differences are significant at 20 m/s to 35 m/s, i.e., at the higher velocity. SDPC achieves 0.10 at 10 m/s to 0.20 at 35 m/s average CH change rate per second where VMaSC and ECHS achieve 0.11 to 0.27, and DCV achieves 0.20 to 0.32 average CH change rate per second for different maximum vehicle velocity.

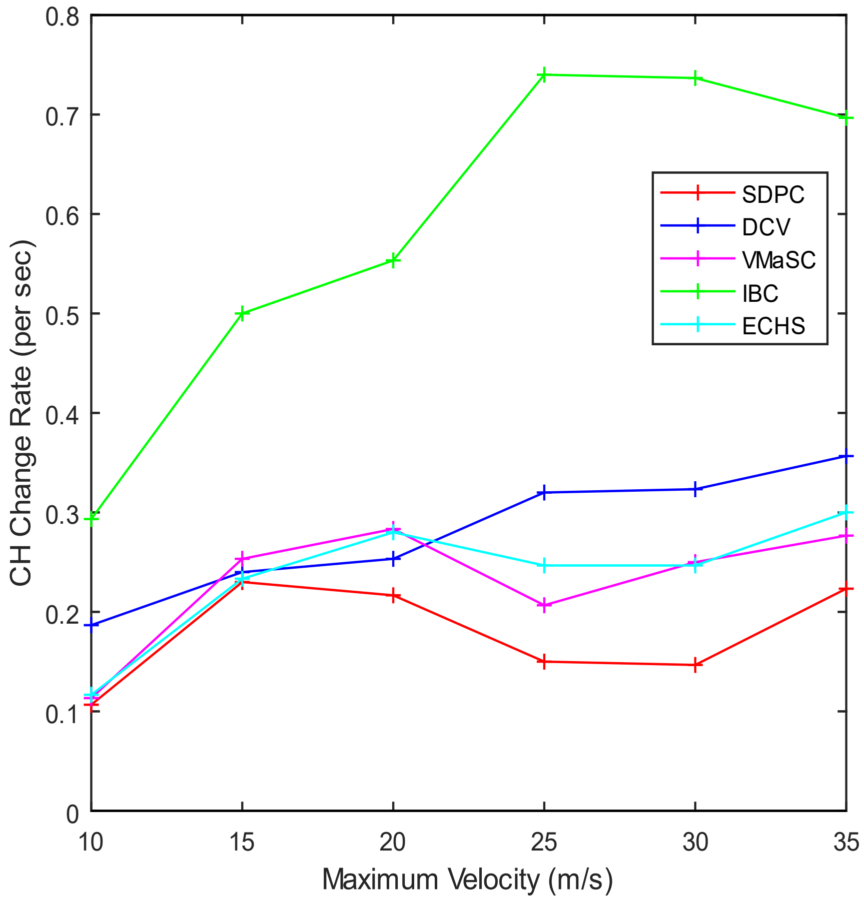

Figure 6.38: Average CH change per second vs maximum vehicle velocity.

Among VMaSC, ECHS and DCV protocols, ECHS shows lower average CH change rate when the maximum vehicle velocity is from 10 m/s to 20 m/s and VMaSC achieves lower CH change rate when the maximum vehicle velocity is from 25 m/s to 35 m/s.

The graph shows that only relative velocity or relative position or degree of node cannot always give the optimum stability for different maximum velocity. Rather a combination of these parameters can provide optimized value.

IBC achieves a range of values from 0.30 to 0.73 as the average CH change rate. The curve for IBC shows much worse performance than SDPC, VMaSC, DCV, and ECHS, because of their forced cluster breakdown at the 2**TR* distance from the intersection. In IBC, when a vehicle enters at the distance of twice of the transmission range from the intersection, vehicle breaks out from the existing cluster and forms a new cluster based on their direction at the intersection. The number of new clusters can be one, two, or three since IBC creates three clusters within the 2**TR* distance: one cluster for vehicles that will run straight direction after the intersection, one cluster for the vehicles that will run at the left direction after the intersection and the third cluster for the vehicles that will run at the right direction after the intersection. As a result, IBC can never provide an optimum average CH change rate compared to the other four protocols.

6.2.4 Average Duration of CH

Figure 6.39 shows the average duration of CH for different maximum vehicle velocities. SDPC achieves a higher average CH duration compared to VMaSC, DCV, IBC and ECHS. SDPC achieves a slightly better average CH duration when the maximum vehicle velocity is low, i.e., from 10 m/s to 20 m/s, and achieves a much higher average CH duration when the maximum vehicle velocities are higher ranges, i.e., from 25 m/s to 35 m/s. When the velocity of the vehicles are 25 m/s and 30 m/s, SDPC achieves 30, 32, 38 and 41 seconds higher average CH duration compared to ECHS, IBC, VMaSC and DCV, respectively.

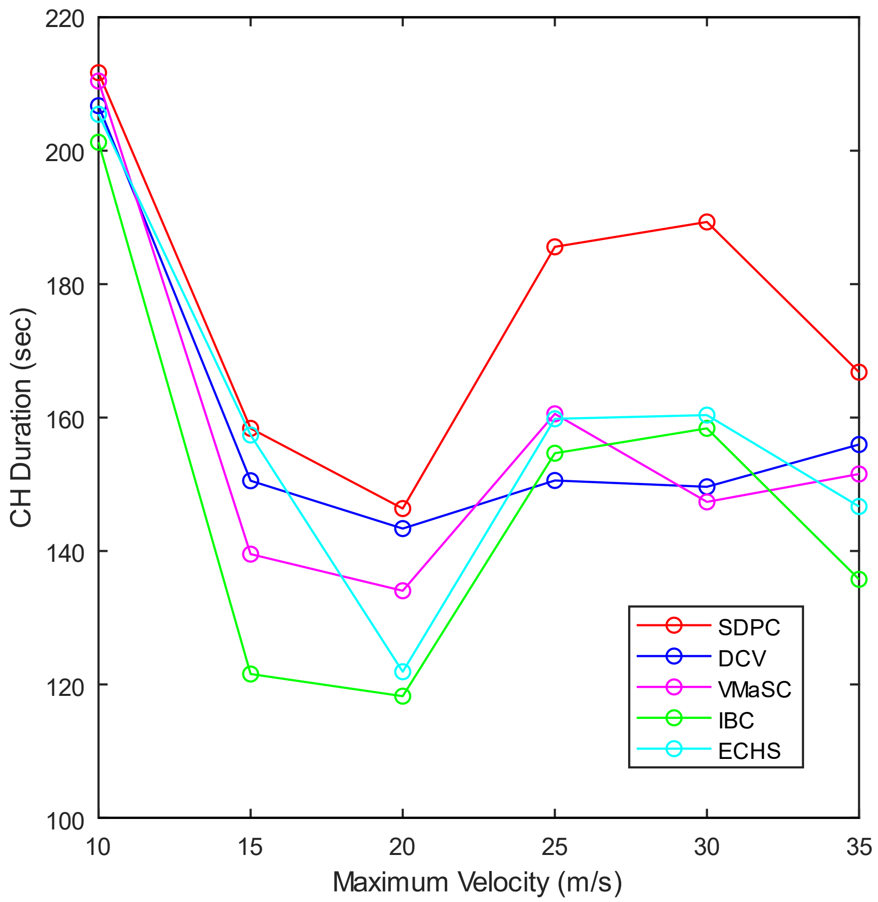

Figure 6.39: Average CH duration vs maximum vehicle velocity.

The graph shows that instead of fewer parameters if we use multiple parameters and consider movement at the intersection, we can achieve a higher average CH duration. Moreover, we are selecting the CH from the majority-vehicles which contributed to increase the average duration of CH. From a vehicle cluster, some vehicles will run straight after the intersection, some vehicles will run at the left direction, and some vehicles will run at the right direction at the direction. The direction in which the greatest number of vehicles will be running after the intersection, we are calling this group of vehicles in the cluster as the majority-vehicles and choosing the CH from this group that helps SDPC to achieve a higher average CH duration compared to VMaSC, DCV, ECHS, and IBC.

6.2.5 Average Duration of CM

Figure 6.40 shows the average duration of CM for different maximum vehicle velocities. From 10 m/s to 35 m/s, the average CM duration is continuously decreased because at the higher speed, the mobility of vehicles increases. The average duration of CM can reflect the stability of a protocol if the average duration of CM is read with other stability metrics, because an inefficient protocol can achieve a higher CM duration by creating clusters with a small number of member vehicles. In this case, the CM will reside with the CH for longer time. Therefore, we need to read the average duration of CM with other important metrics.

Except DCV, the performance of the other four algorithms is relatively in the same range while SDPC and IBC achieves slightly higher average CM duration compared to VMaSC and ECHS. Since DCV uses a smaller transmission range than the actual transmission range, its CH can cover a small number of vehicles resulting in a higher number of unattended vehicles which reflects in the average CM duration.

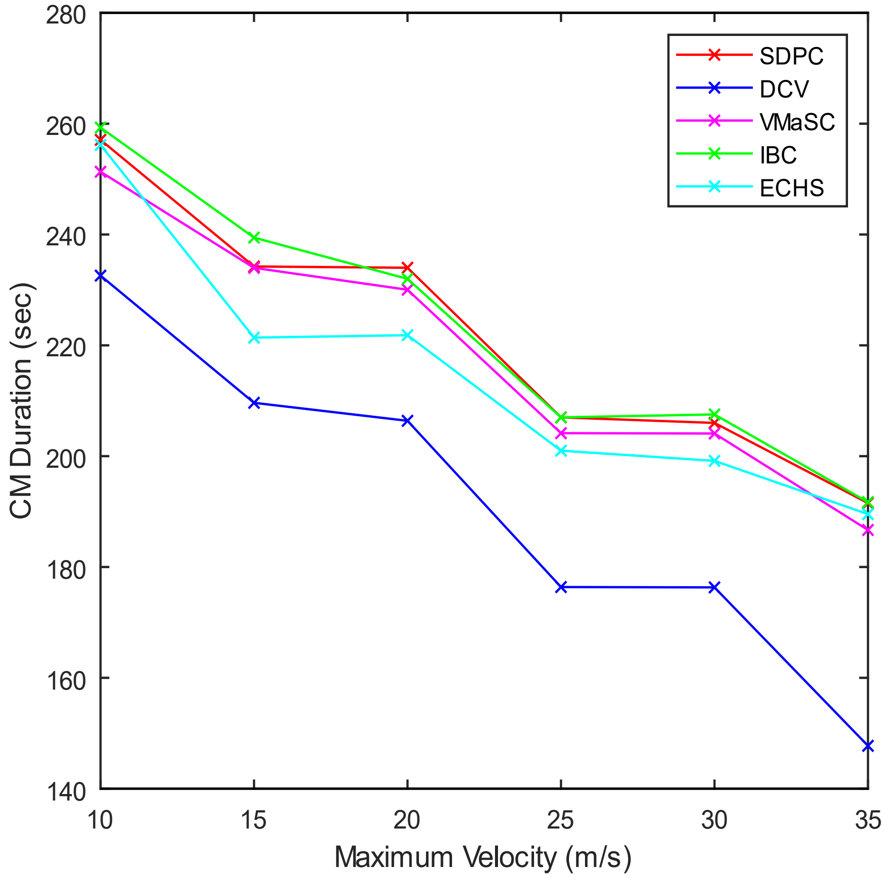

Figure 6.40: Average CM duration vs maximum vehicle velocity.

Even though the averages for SDPC and IBC are almost overlapping each other, IBC's performance for the average CM duration is not due to the strength of IBC algorithm, rather IBC forces vehicles to break from the existing cluster at a distance of 2**TR* from the intersection. As a result, new vehicle as EN is becoming available for vehicle with EN status to form a new cluster which was not able to join in a cluster previously for some reason, e.g., due to out of the range of the CH.

Even though we do not consider the average CM duration alone as important performance criteria for cluster stability, a slightly higher CM average achieved by SDPC compared to VMaSC, ECHS and DCV shows that while we improved the performance of the average CH change rate, the average CH duration, and the average clustering overhead, we did not compromise with the average CM duration also.

6.2.6 Average Clustering Overhead

Figure 6.41 shows the percentage of clustering overhead of SDPC, DCV, VMaSC, IBC and ECHS for different maximum vehicle velocities. From 10 m/s to 35 m/s, SDPC shows a significantly lower number of clustering overhead compared to the other four algorithms. SDPC achieves 11% at 10 m/s to 26% at 35 m/s. ECHS ranges from 16% to 35%, VMaSC ranges from 17% to 35% and DCV ranges from 21% to 44% at the speed of 10 m/s to 35 m/s. IBC rises from 30% at 10 m/s to 68% at 25 m/s, then remains nearly unchanged.

SDPC had achieved a lower average number of CH change rate compared to VMaSC, DCV, ECHS, and IBC. At the same time, SDPC uses a lower number of states for the vehicles, and a lower number of state transitions for the vehicles. SDPC minimized the number of states to only three and the number of state transitions to four. A lower number of state transition and a lower number of CH change contributed to a lower clustering overhead for SDPC.

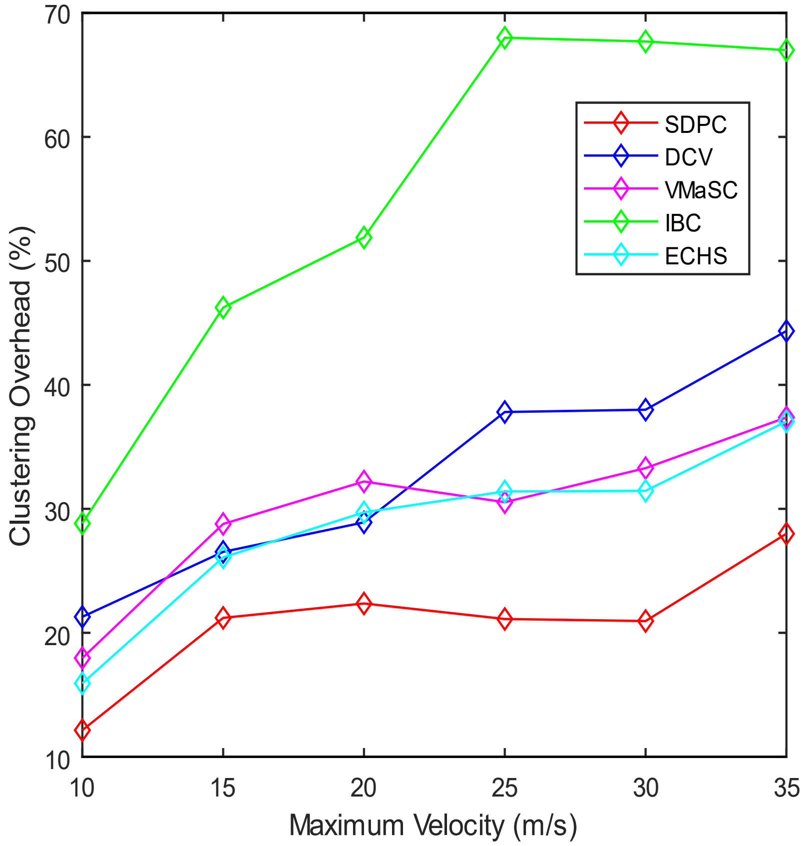

Figure 6.41: Clustering overhead vs maximum vehicle velocity.

6.3 Summary

In this chapter, the detailed results and analysis of the proposed protocols have been presented comparing with the existing protocols. JCV is evaluated to show that it can optimize both CH and CM-related metrics to provide stability. SDPC is evaluated to show that it can provide optimum stability for all the major metrics in highways and intersections, regardless of the road topology. In Chapter 7, we presented the conclusions, and the future works to conclude the dissertation.

Chapter 7

Conclusions and Future Works

We have presented two VANET clustering protocols and one VANET simulation platform in this dissertation. We will conclude this dissertation presenting the summary of the dissertation in Section 7.1 and the future works in Section 7.2.

7.1 Summary

We presented JCV clustering protocol which utilizes several parameters such as distance among the vehicles, vehicle position, transmission range, degree of a node and the route of the vehicle at the next intersection for clustering. JCV uses a fewer number of vehicle states which results in simplified state transition. A vehicle can make its own decision whether it will join in a cluster or not by considering its movement. This next route consideration improves the performance of CH and CM. CH-related and CM-related metrics have been improved in JCV. The average CH duration, the average number of CH change etc. are improved. CM-related metrics such as the average CM duration, the average number of CM, the average ratio of CMs, etc. are also improved. The average number and the average duration of the non-clustered vehicles are also improved.

Higher average CH duration, average CM duration, average number of CM, the average ratio of CM, and a lower average number of CH change, average number of CH, average number of non-cluster vehicles have been achieved by JCV. JCV achieves better stability in terms of the average CH duration (4%), the average CM duration (8%), the number of CM (6%), the ratio of CM (22%), the average CH change rate (14%), the number of CH (10%), the number of non-cluster vehicles (7%), and clustering overhead (35%).

We have also presented an all-weather stable dynamic feedback based predictive VANET clustering protocol that has achieved stability regardless of topology. Instead of creating cluster based on the current road scenario, the future road scenario is constructed based on the continuous feedback from the vehicles and cluster is formed based on the constructed future road scenario. Also, an efficient approach of using the moving direction at the next intersection is considered in the cluster formation process. The relative position and movement of the vehicles at the next intersection are important in selecting the CH. When multiple vehicles' candidacy for the CH are equal, degree of vehicles is considered in a limited sense. Hence, the presented protocol achieves higher stability, preventing frequent breaking of the clusters at the intersection while maintaining optimum performance for highways.

SDPC shows superior performance in terms of the major metrics such as the average CH change rate, the average CH duration, the average CM duration, and the average clustering overhead. SDPC achieves better clustering stability in terms of the average CH change rate (50%), the average CH duration (15%), the average CM duration (6%), and the clustering overhead (35%). Generally, previous clustering algorithms made a trade-off between the performance at the straight road and at the intersection. On the contrary, the presented clustering algorithm achieves optimized performance at both highways and intersections.

We have also presented a new VANET simulator, CVANETSIM. CVANETSIM comes as the first of its kind with a database server, real-time data processing, machine-independent platform, accessibility from internet, and a clustering module.

7.2 Future Works

In the future, we want to investigate the possibility of 6G in VANET when device-to-device communication will be available for vehicular environment. We also want to expand from V2V to V2X as a part of expansion from IoV to IoT and want to solve the newly arrived problem. All the vehicles on the road are not trustworthy and they can disrupt the communication among the vehicles. Therefore, we want to investigate all the security concerns also.

- JCV and SDPC are independent of communication technology. They only require V2V communication. Any communication technology who can provide device-to-device communication can be used in JCV and SDPC. For simulation purposes, we have used the transmission range of DSRC. However, 6G is under research, which is expected to be able to provide device-to-device communication. In the future, we want to investigate both DSRC and 6G to compare their performance in VANET perspective so that we can choose a more suitable technology for VANET.
- Now we are using V2V which is termed as internet of vehicles (IoV). V2X will be a more realistic part of IoT. Using more frequent communication and availability of 6G will require more measures for pedestrian safety and fuel consumption by the vehicles. Fuel consumption is not only important for the

shortage of fuel but also the climate change issue needs to be considered. To save our environment and our future generations, we need to investigate ways of reducing the consumption of fuel.

- Network security will become a crucial issue for reliable communication in VANET. Clustering can solve the issue partially and passively. However, the trustworthiness of the vehicles is very important because if a vehicle spreads misinformation, it will damage the communication in the network and can cause accidents and fatalities. For example, a vehicle can transmit false positive or false negative messages and can disrupt the communication among the vehicles. We want to investigate all the possible security concerns in VANET and their optimum solutions.

Bibliography

[1] M. Mukhtaruzzaman and M. Atiquzzaman, "Clustering in vehicular ad hoc network: Algorithms and challenges," *Computers & Electrical Engineering,* vol. 88, no. 106851, pp. 1-15, 2020.

[2] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of Network and Computer Applications,* vol. 37, pp. 380-392, 2014.

[3] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, "A review of clustering algorithms in VANETs," *Annals of Telecommunications,* pp. 1-23, 2021.

[4] R. S. Bali, N. Kumar, and J. J. Rodrigues, "Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions," *Vehicular communications,* vol. 1, no. 3, pp. 134-152, 2014.

[5] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of VANET clustering techniques," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 1, pp. 657-681, 2017.

[6] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-cluster-based IEEE 802.11 p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 4, pp. 2621-2636, 2016.

[7] W. Qi, Q. Song, X. Wang, L. Guo, and Z. Ning, "SDN-Enabled Social-Aware Clustering in 5G-VANET Systems," *IEEE Access,* vol. 6, pp. 28213-28224, 2018.

[8] S. Shafi and D. V. Ratnam, "A Cross Layer Cluster Based Routing Approach for Efficient Multimedia Data Dissemination with Improved Reliability in VANETs," *Wireless Personal Communications,* pp. 1-18, 2019.

[9] M. Bersali, A. Rachedi, and H. Bouarfa, "A New Collaborative Clustering Approach for the Internet of Vehicles (CCA-IoV)," in *2020 Second International Conference on Embedded & Distributed Systems (EDiS)*, pp. 58-63, 2020.

[10] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Vèque, "A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs)," *Vehicular Communications,* vol. 9, pp. 233-241, 2017.

[11] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Vèque, "A new mobility-based clustering algorithm for vehicular ad hoc networks (VANETs),"

presented at the IEEE/IFIP Network Operations and Management Symposium, 2016.

[12] M. Banikhalaf and M. A. Khder, "A Simple and Robust Clustering Scheme for Large-Scale and Dynamic VANETs," *IEEE Access*, vol. 8, pp. 103565-103575, 2020.

[13] M. Mukhtaruzzaman and M. Atiquzzaman, "Junction-based stable clustering algorithm for vehicular ad hoc network," *Annals of Telecommunications*, vol. 76, pp. 777–786, 2021.

[14] H. Zhao, J. Liu, J. Wu, and W. Liu, "An intersection-based clustering algorithm for vehicular ad hoc networks," presented at the 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP), 2016.

[15] M. Mukhtaruzzaman and M. Atiquzzaman, "Stable Dynamic Predictive Clustering (SDPC) Protocol for Vehicular Ad hoc Networks (VANETs)," *arXiv*.

[16] *NS-2*. Available: https://www.isi.edu/nsnam/ns/

[17] *Veins*. Available: https://veins.car2x.org

[18] *NetSim*. Available: https://www.tetcos.com/

[19] M. Mukhtaruzzaman and M. Atiquzzaman, "Cloud based VANET Simulator (CVANETSIM)," *arXiv*.

[20] M. Mukhtaruzzaman and M. Atiquzzaman, "Clustering in VANET: Algorithms and challenges," *arXiv*.

[21] I. Tal and G.-M. Muntean, "Towards smarter cities and roads: A survey of clustering algorithms in vanets," in *Convergence of Broadband, Broadcast, and Cellular Network Technologies*, pp. 16-50, 2014.

[22] E. Ahmed and H. Gharavi, "Cooperative vehicular networking: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 996-1014, 2018.

[23] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, "Machine Learning for Vehicular Networks: Recent Advances and Application Examples," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 94-101, 2018.

[24] S. M. AlMheiri and H. S. AlQamzi, "MANETs and VANETs clustering algorithms: A survey," in *IEEE 8th GCC Conference & Exhibition*, pp. 1-6, 2015.

[25] S. Bitam, A. Mellouk, and S. Zeadally, "Bio-inspired routing algorithms survey for vehicular ad hoc networks," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 2, pp. 843-867, 2015.

[26] P. Yang, J. Wang, Y. Zhang, Z. Tang, and S. Song, "Clustering algorithm in VANETs: A survey," in *IEEE 9th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pp. 166-170, 2015.

[27] R. S. Hande and A. Muddana, "Comprehensive survey on clustering-based efficient data dissemination algorithms for VANET," in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 629-632, 2016.

[28] A. Muddasar, O. Alma, H. Rasheed, A. K. Ammara, and Z. Zhongshan, "A comprehensive survey on clustering in vehicular networks: Current solutions and future challenges," *Ad Hoc Networks,* vol. 124, 2022.

[29] M. K. J. Alsabah, H. Trabelsi, and W. Jerbi, "Survey on Clustering in VANET Networks," presented at the 18th IEEE International Multi-Conference on Systems, Signals and Devices, 2021.

[30] Q. Zhang, M. Almulla, Y. Ren, and A. Boukerche, "An efficient certificate revocation validation scheme with k-means clustering for vehicular ad hoc networks," in *IEEE Symposium on Computers and Communications (ISCC),* pp. 000862-000867, 2012.

[31] S. K. Bansal, A. S. Bisen, and R. Gupta, "A secure hashing technique for k-means based cluster approach in VANET," in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 2037-2041, 2016.

[32] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, no. 11, pp. 3275-3285, 2016.

[33] I. Hussain and C. Bingcai, "Cluster Formation and Cluster Head Selection approach for Vehicle Ad-Hoc Network (VANETs) using K-Means and Floyd-Warshall Technique," *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 12, pp. 11-15, 2017.

[34] J. Cui, Z. Ding, P. Fan, and N. Al-Dhahir, "Unsupervised Machine Learning-Based User Clustering in Millimeter-Wave-NOMA Systems," *IEEE Transactions on Wireless Communications,* vol. 17, no. 11, pp. 7425-7440, 2018.

[35] M. Almulla, Q. Zhang, A. Boukerche, and Y. Ren, "An efficient k‐Means authentication scheme for digital certificates revocation validation in vehicular ad hoc networks," *Wireless Communications and Mobile Computing,* vol. 14, no. 16, pp. 1546-1563, 2014.

[36] R. Chai, X. Ge, and Q. Chen, "Adaptive K-Harmonic Means clustering algorithm for VANETs," in *International Symposium on Communications and Information Technologies (ISCIT),* pp. 233-237, 2014.

[37] P. Bhosale and A. Vidhate, "An agglomerative approach to elect the cluster head in VANET," in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES),* pp. 1340-1344, 2016.

[38] Y. A. Shah, H. A. Habib, F. Aadil, M. F. Khan, M. Maqsood, and T. Nawaz, "CAMONET: Moth-flame optimization (MFO) based clustering algorithm for VANETs," *IEEE Access,* vol. 6, pp. 48611-48624, 2018.

[39] M. F. Khan, F. Aadil, M. Maqsood, S. H. R. Bukhari, M. Hussain, and Y. Nam, "Moth Flame Clustering Algorithm for Internet of Vehicle (MFCA-IoV)," *IEEE Access*, vol. 7, pp. 11613-11629, 2019.

[40] K. Kandali, L. Bennis, and H. Bennis, "A New Hybrid Routing Protocol Using a Modified K-Means Clustering Algorithm and Continuous Hopfield Network for VANET," *IEEE Access,* vol. 9, pp. 47169-47183, 2021.

[41] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228-249, 2015.

[42] K. A. Hafeez, L. Zhao, J. W. Mark, X. Shen, and Z. Niu, "Distributed multichannel and mobility-aware cluster-based MAC protocol for vehicular ad hoc networks," *IEEE Transactions on vehicular Technology,* vol. 62, no. 8, pp. 3886-3902, 2013.

[43] K. A. Hafeez, L. Zhao, Z. Liao, and B. N.-W. Ma, "A fuzzy-logic-based cluster head selection algorithm in VANETs," in *IEEE International Conference on Communications (ICC),* pp. 203-207, 2012.

[44] A. Çalhan, "A fuzzy logic based clustering strategy for improving vehicular ad-hoc network performance," *Sadhana-Academy Proceedings in Engineering Sciences,* vol. 40, no. 2, pp. 351-367, 2015.

[45] W. Zhao, L. Zhang, W. Zhu, and Y. Zhao, "Multilevel cluster-based information fusion in vehicle ad hoc networks," in *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing,* pp. 1357-1362, 2013.

[46] G. El Mouna Zhioua, N. Tabbane, H. Labiod, and S. Tabbane, "A fuzzy multi-metric QoS-balancing gateway selection algorithm in a clustered VANET to LTE advanced hybrid cellular network," *IEEE Transactions on Vehicular Technology,* vol. 64, no. 2, pp. 804-817, 2015.

[47] H. R. Arkian, R. E. Atani, and S. Kamali, "FcVcA: A fuzzy clustering-based vehicular cloud architecture," in *7th International Workshop on Communication Technologies for Vehicles (Nets4Cars-Fall),* pp. 24-28, 2014.

[48] K. Bylykbashi, Y. Liu, D. Elmazi, K. Matsuo, M. Ikeda, and L. Barolli, "A Secure and Trustworthy Intelligent System for Clustering in VANETs Using Fuzzy Logic," in *International Conference on Advanced Information Networking and Applications*, pp. 156-165, 2019.

[49] H. R. Arkian, R. E. Atani, A. Diyanat, and A. Pourkhalili, "A cluster-based vehicular cloud architecture with learning-based resource management," *The Journal of Supercomputing,* vol. 71, no. 4, pp. 1401-1426, 2015.

[50] C. Wu, T. Yoshinaga, Y. Ji, T. Murase, and Y. Zhang, "A reinforcement learning-based data storage scheme for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology,* vol. 66, no. 7, pp. 6336-6348, 2017.

[51] C. Wu, T. Yoshinaga, and Y. Ji, "V2R Communication Protocol Based on Game Theory Inspired Clustering," in *IEEE 86th Vehicular Technology Conference (VTC-Fall),* pp. 1-6, 2017.

[52] C. Wu, T. Yoshinaga, Y. Ji, and Y. Zhang, "Computational Intelligence Inspired Data Delivery for Vehicle-to-Roadside Communications," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 12, pp. 12038-12048, 2018.

[53] U. Bhanja, R. Kumar, A. Routroy, S. Behura, and S. Mahapatra, "Dynamic trafile congestion detection in VANETS using a Fuzzy rule-based system and K-means clustering," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS),* pp. 1-6, 2017.

[54] S. Sharma and A. Kaul, "Hybrid fuzzy multi-criteria decision making based multi cluster head dolphin swarm optimized IDS for VANET," *Vehicular Communications,* vol. 12, pp. 23-38, 2018.

[55] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol," 2070-1721, 2004.

[56] H.-J. Lim, M. Kim, J.-H. Lee, and T. M. Chung, "Route optimization in nested NEMO: classification, evaluation, and analysis from NEMO fringe stub perspective," *IEEE Transactions on Mobile Computing,* vol. 8, no. 11, pp. 1554-1572, 2009.

[57] M. H. Kabir, M. Mukhtaruzzaman, and M. Atiquzzaman, "Efficient route optimization scheme for nested-NEMO," *Journal of Network and Computer Applications,* vol. 36, no. 3, pp. 1039-1049, 2013.

[58] A. Z. M. Shahriar, M. Atiquzzaman, and W. Ivancic, "Route optimization in network mobility: Solutions, classification, comparison, and future research directions," *IEEE Communications Surveys & Tutorials,* vol. 12, no. 1, 2010.

[59] M. Ni, Z. Zhong, and D. Zhao, "MPBC: A mobility prediction-based clustering scheme for ad hoc networks," *IEEE Transactions on Vehicular Technology,* vol. 60, no. 9, pp. 4549-4559, 2011.

[60] E. Souza, I. Nikolaidis, and P. Gburzynski, "A new aggregate local mobility (ALM) clustering algorithm for VANETs," in *IEEE International Conference on Communications (ICC),* pp. 1-5, 2010.

[61] Z. Y. Rawashdeh and S. M. Mahmud, "A novel algorithm to form stable clusters in vehicular ad hoc networks on highways," *EURASIP Journal on Wireless Communications and Networking,* vol. 2012, no. 1, p. 15, 2012.

[62] W. Fan, Y. Shi, S. Chen, and L. Zou, "A mobility metrics based dynamic clustering algorithm for VANETs," 2011.

[63] J.-H. Kwon, C. Kwon, and E.-J. Kim, "Neighbor mobility-based clustering scheme for vehicular ad hoc networks," in *International Conference on Platform Technology and Service (PlatCon),* pp. 31-32, 2015.

[64] J.-H. Kwon, H. S. Chang, T. Shon, J.-J. Jung, and E.-J. Kim, "Neighbor stability-based VANET clustering for urban vehicular environments," *The Journal of Supercomputing,* vol. 72, no. 1, pp. 161-176, 2016.

[65] R. Goonewardene, F. Ali, and E. Stipidis, "Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks," *IET Intelligent Transport Systems,* vol. 3, no. 2, p. 148, 2009.

[66] N. Maslekar, M. Boussedjra, J. Mouzna, and H. Labiod, "C-DRIVE: clustering based on direction in vehicular environment," in *4th IFIP International Conference on New Technologies, Mobility and Security (NTMS),* pp. 1-5, 2011.

[67] N. Maslekar, J. Mouzna, H. Labiod, M. Devisetty, and M. Pai, "Modified C-DRIVE: Clustering based on direction in vehicular environment," in *IEEE Intelligent Vehicles Symposium (IV),* pp. 845-850, 2011.

[68] H. R. Arkian, R. E. Atani, A. Pourkhalili, and S. Kamali, "A Stable Clustering Scheme Based on Adaptive Multiple Metric in Vehicular Ad-hoc Networks," *Journal of Information Science and Engineering,* vol. 31, no. 2, pp. 361-386, 2015.

[69] C. Shea, B. Hassanabadi, and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *IEEE Global Telecommunications Conference*, pp. 1-6, 2009.

[70] B. Hassanabadi, C. Shea, L. Zhang, and S. Valaee, "Clustering in vehicular ad hoc networks using affinity propagation," *Ad Hoc Networks,* vol. 13, pp. 535-548, 2014.

[71] M. Hadded, P. Muhlethaler, A. Laouiti, and L. A. Saidane, "A novel angle-based clustering algorithm for vehicular ad hoc networks," in *Vehicular Ad-Hoc Networks for Smart Cities*, pp. 27-38, 2017.

[72] X. Cheng and B. Huang, "A center-based secure and stable clustering algorithm for VANETs on highways," *Wireless Communications and Mobile Computing,* vol. 2019, 2019.

[73] H.-W. Tseng, R.-Y. Wu, and C.-W. Lo, "A stable clustering algorithm using the traffic regularity of buses in urban VANET scenarios," *Wireless Networks,* pp. 1-15, 2019.

[74] X. Yang, Q. Chen, C. Chen, and J. Zhao, "Improved ZRP Routing Protocol Based on Clustering," *Procedia computer science,* vol. 131, pp. 992-1000, 2018.

[75] K. Aravindhan and C. S. G. Dhas, "Destination-aware context-based routing protocol with hybrid soft computing cluster algorithm for VANET," *Soft Computing,* vol. 23, no. 8, pp. 2499-2507, 2019.

[76] J. Cheng *et al.*, "Research on the Prediction-Based Clustering Method in the Community of Medical Vehicles for Connected Health," *IEEE Access,* vol. 7, pp. 71884-71896, 2019.

[77] A. Touil and F. Ghadi, "Efficient dissemination based on passive approach and dynamic clustering for VANET," *Procedia Computer Science,* vol. 127, pp. 369-378, 2018.

[78] A. Touil, A. Sbai, and F. Ghadi, "Cluster-based Data Collection Scheme for Vehicular Ad-hoc Networks," *Procedia computer science,* vol. 148, pp. 62-69, 2019.

[79] B. Dwivedy, A. K. Bhola, and S. Yadav, "Cluster Based Multi Hop Data Dissemination Protocol in V2V Networks using Whale Optimization Technique," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM),* pp. 228-231, 2019.

[80] C. Raut and S. Devane, "QoS Enhancement using Intelligent Cluster Head Selection Routing Technique in Vehicular Ad Hoc Networks in case of RSU

failure," in *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1-6, 2018.

[81] R. Regin and T. Menakadevi, "Dynamic Clustering Mechanism to Avoid Congestion Control in Vehicular Ad Hoc Networks Based on Node Density," *Wireless Personal Communications*, pp. 1-21, 2019.

[82] A. S. Shah, H. Ilhan, and U. Tureli, "qCB-MAC: QoS aware cluster-based MAC protocol for VANETs," in *Science and Information Conference*, pp. 685-695, 2018.

[83] F. Abbas, G. Liu, Z. Khan, K. Zheng, and P. Fan, "Clustering Based Resource Management Scheme for Latency and Sum Rate Optimization in V2X Networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1-6, 2019.

[84] I. Ahmad, R. M. Noor, M. R. Zaba, M. A. Qureshi, M. Imran, and M. Shoaib, "A Cooperative Heterogeneous Vehicular Clustering Mechanism for Road Traffic Management," *International Journal of Parallel Programming*, pp. 1-20, 2019.

[85] A. U. Haq, K. Liu, and M. B. Latif, "A Location-and Mobility-Aware Clustering-Based TDMA MAC Protocol for Vehicular Ad-hoc Networks," in *2019 28th Wireless and Optical Communications Conference (WOCC)*, pp. 1-5, 2019.

[86] B. K. Chaurasia, M. I. Alam, A. Prakash, R. S. Tomar, and S. Verma, "MPMAC: Clustering Based MAC Protocol for VANETs," *Wireless Personal Communications*, pp. 1-28, 2019.

[87] C. J. Joshua, R. Duraisamy, and V. Varadarajan, "A Reputation based Weighted Clustering Protocol in VANET: A Multi-objective Firefly Approach," *Mobile Networks and Applications*, pp. 1-11, 2019.

[88] G. H. Alsuhli, A. Khattab, and Y. A. Fahmy, "Double-head clustering for resilient VANETs," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[89] Y. Zhou, X. Wu, and P. Wang, "A novel vehicle clustering algorithm in intersection scenario," in *9th International Conference on Advanced Infocomm Technology (ICAIT)*, pp. 431-437, 2017.

[90] I. Ahmad, R. Md Noor, and M. Reza Z'aba, "LTE efficiency when used in traffic information systems: A stable interest aware clustering," *International Journal of Communication Systems*, vol. 32, no. 2, 2019.

[91] I. Ahmad, R. M. Noor, I. Ahmedy, S. A. A. Shah, E. A. I. Yaqoob, and M. Imran, "VANET–LTE based heterogeneous vehicular clustering for driving assistance and route planning applications," *Computer Networks*, vol. 145, pp. 128-140, 2018.

[92] G. H. Alsuhli, A. Khattab, and Y. A. Fahmy, "A Mobility-Based Double-Head Clustering Algorithm for Dynamic VANET," in *2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC)*, pp. 91-94, 2018.

[93] A. Boukerche, Z. Zhang, and X. Fei, "Reducing handoff latency for NEMO-based vehicular ad hoc networks," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pp. 1-5, 2011.

[94] H. Ko, S. Pack, J.-H. Lee, and A. Petrescu, "DLM: Delayed location management in network mobility (NEMO)-based public transportation systems," *Journal of Network and Computer Applications*, vol. 85, pp. 127-133, 2017.

[95] Y. S. Chen, C. S. Hsu, and C. H. Cheng, "Network mobility protocol for vehicular ad hoc networks," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3042-3063, 2014.

[96] Y.-S. Chen, C.-H. Cheng, C.-S. Hsu, and G.-M. Chiu, "Network Mobility Protocol for Vehicular Ad Hoc Networks," in *IEEE Wireless Communications and Networking Conference*, pp. 1-6, 2009.

[97] H. Lee, D. Jung, and D. Lee, "Proxy Based Seamless Connection Management Scheme for Smart Transportation Service," *Wireless Personal Communications*, vol. 91, no. 4, pp. 1669-1679, 2016.

[98] S. A. Bugti, X. Chunhe, L. Wie, and E. Hussain, "Cluster based addressing scheme in vanet (canvet stateful addressing approach)," in *IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pp. 450-454, 2011.

[99] W. Xiaonan and Q. Huanyan, "Mobility management solution for IPv6-based vehicular networks," *Computer Standards & Interfaces*, vol. 36, no. 1, pp. 66-75, 2013.

[100] S. Taha and X. Shen, "A physical-layer location privacy-preserving scheme for mobile public hotspots in NEMO-based VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1665-1680, 2013.

[101] D. Tian, Y. Wang, G. Lu, and G. Yu, "A VANETs routing algorithm based on Euclidean distance clustering," in *2nd International Conference on Future Computer and Communication (ICFCC)*, vol. 1, pp. V1-183-V1-187, 2010.

[102] W. Li, A. Tizghadam, and A. Leon-Garcia, "Robust clustering for connected vehicles using local network criticality," in *IEEE International Conference on Communications (ICC)*, pp. 7157-7161, 2012.

[103] E. Dror, C. Avin, and Z. Lotker, "Fast randomized algorithm for hierarchical clustering in vehicular ad-hoc networks," in *10th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 1-8, 2011.

[104] E. Dror, C. Avin, and Z. Lotker, "Fast randomized algorithm for 2-hops clustering in vehicular ad-hoc networks," *Ad Hoc Networks*, vol. 11, no. 7, pp. 2002-2015, 2013.

[105] P. Zhao, K. Liu, Y. Zhang, T. Zhang, and F. Liu, "A clustering-based fast and stable routing protocol for vehicular ad hoc networks," in *Journal of Physics: Conference Series*, vol. 1060, no. 1, p. 012050, 2018.

[106] Z. Zhang, A. Boukerche, and R. Pazzi, "A novel multi-hop clustering scheme for vehicular ad-hoc networks," in *Proceedings of the 9th ACM international symposium on Mobility management and wireless access*, pp. 19-26, 2011.

[107] Y. Chen, M. Fang, S. Shi, W. Guo, and X. Zheng, "Distributed multi-hop clustering algorithm for VANETs based on neighborhood follow," *Eurasip journal on Wireless communications and networking*, vol. 2015, no. 1, p. 98, 2015.

[108] D. Zhang, H. Ge, T. Zhang, Y.-Y. Cui, X. Liu, and G. Mao, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1517-1530, 2019.

[109] J. R. Dawande, S. Silakari, and A. J. Deen, "Enhanced Distributed Multi-Hop Clustering Algorithm for VANETs Based on Neighborhood Follow (EDMCNF) Collaborated with Road Side Units," in *International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 106-113, 2015.

[110] L. Zhang and H. El-Sayed, "A novel cluster-based protocol for topology discovery in vehicular ad hoc network," *Procedia Computer Science*, vol. 10, pp. 525-534, 2012.

[111] S. Ucar, S. C. Ergen, and O. Ozkasap, "VMaSC: Vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2381-2386, 2013.

[112] *SUMO*. Available: https://www.eclipse.org/sumo/

Appendix A

Author's List of Publications

PEER-REVIEWED JOURNAL

Mohammad Mukhtaruzzaman and Mohammed Atiquzzaman, "Junction-based stable clustering algorithm for vehicular ad hoc network", *Annal of Telecommunications*, Vol. 76, no. 9, pp. 777-786, August 2021.

https://doi.org/10.1007/s12243-021-00881-9

Mohammad Mukhtaruzzaman and Mohammed Atiquzzaman, "Clustering in vehicular ad hoc network: Algorithms and challenges", *Computers & Electrical Engineering*, Vol. 88, no. 106851, pp. 1-15, December 2020.

https://doi.org/10.1016/j.compeleceng.2020.106851

Md. Humayun Kabir, Mohammad Mukhtaruzzaman, Mohammed Atiquzzaman, "Efficient route optimization scheme for nested-NEMO", *Journal of Network and Computer Applications*, Vol. 36, no. 3, pp. 959-1090, May 2013.

https://doi.org/10.1016/j.jnca.2012.12.026

Mohammad Mukhtaruzzaman and Mohammed Atiquzzaman, "Stable Dynamic Predictive Clustering (SDPC) Protocol for Vehicular Ad hoc Networks (VANETs)", https://doi.org/10.48550/arXiv.2209.08075, Under review, Computer Networks, 2023.

OTHER ARTICLES (AVAILABLE AS PREPRINT)

Mohammad Mukhtaruzzaman and Mohammed Atiquzzaman, "Cloud based VANET Simulator (CVANETSIM)". https://doi.org/10.48550/arXiv.2101.11147

Mohammad Mukhtaruzzaman and Mohammed Atiquzzaman, "Clustering in VANET: Algorithms and challenges". https://doi.org/10.48550/arXiv.2009.01964

Appendix B

Acronyms

AVG_REL_DIST	Average Relative Distance
AVG_REL_SPEED	Average Relative Speed
CG	Cluster Gateway-Member
CH	Cluster Head
CH_RESP	Response From a CH
CH_TIMER	Cluster Head State Timer
CHt	Temporary Cluster Head
CM	Cluster Member
CM_TIMER	Cluster Member State Timer
CVANETSIM	Cloud-Based VANET Simulator
DCV	Dynamic Clustering in VANET
DSRC	Dedicated Short-Range Communication
ECHS	Efficient Cluster Head Selection
EN	Entry State
EN_REQ	Message From EN
EN_TIMER	Entry State Timer
FLS	Fuzzy Logic System
GPS	Global Positioning System
GUI	Graphical User Interface
IBC	Intersection-based Clustering
IoT	Internet of Things
IoV	Internet of Vehicle
IP	Internet Protocol
ITS	Intelligent Transportation System
JCV	Junction-based Clustering Protocol for VANET

LTE	Long-Term Evolution
MAC	Medium Access Control
MFO	Moth-Flame Optimization
MIP	Mobile-IP
MR	Mobile Router
NBSP	Nemo Basic Support Protocol
NEMO	Network Mobility
OBU	On-board Unit
PDR	Packet Delivery Ratio
QoS	Quality of Service
REL_DIST	Relative Distance
RSU	Road-side Unit
SDPC	Stable Dynamic Feedback-based Predictive Clustering
STATUS	Status of a Vehicle
SUMO	Simulation for Urban Mobility
TM	Temporal State
TR	Transmission Range
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular Ad hoc Network
VEH_ADV	Periodic Vehicle Advertise Message
VEH_INFO	Vehicle Info
VEH_RANK	Ranking of a Vehicle
VMaSC	Vehicular Multi-Hop Algorithm for Stable Clustering
WAVE	Wireless Access in Vehicular Environment