



HAL
open science

Cluster-based Data Dissemination in Vehicular Ad-hoc Networks (VANETs)

Mengying Ren

► **To cite this version:**

Mengying Ren. Cluster-based Data Dissemination in Vehicular Ad-hoc Networks (VANETs). Networking and Internet Architecture [cs.NI]. Université de Technologie de Troyes, 2018. English. NNT : 2018TROY0007 . tel-02974809

HAL Id: tel-02974809

<https://theses.hal.science/tel-02974809>

Submitted on 22 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse
de doctorat
de l'UTT

Mengying REN

Cluster-based Data Dissemination in Vehicular Ad-hoc Networks (VANETs)

Spécialité :

**Ingénierie Sociotechnique des Connaissances, des Réseaux
et du Développement Durable**

2018TROY0007

Année 2018

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

**Spécialité : INGENIERIE SOCIOTECHNIQUE DES CONNAISSANCES
DES RESEAUX ET DU DEVELOPPEMENT DURABLE**

présentée et soutenue par

Mengying REN

le 6 avril 2018

Cluster-based Data Dissemination in Vehicular Ad-hoc Networks (VANETs)

JURY

M. J. HÄRRI	PROFESSEUR EURECOM - HDR	Président
M. B. BENSAOU	ASSOCIATE PROFESSOR	Examineur
M. A.-L. BEYLOT	PROFESSEUR DES UNIVERSITES	Rapporteur
M. L. KHOUKHI	PROFESSEUR ASSOCIE UTT - HDR	Directeur de thèse
Mme H. LABIOD	PROFESSEURE DES UNIVERSITES	Directrice de thèse
Mme A. MOLINARO	PROFESSORE ASSOCIATO	Rapporteur
Mme V. VÈQUE	PROFESSEURE DES UNIVERSITES	Examineur
M. Jun ZHANG	DOCTEUR	Examineur

Personnalité invitée

M. A. SERVEL	INGENIEUR EXPERT
--------------	------------------

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors Prof. Lyes Khoukhi and Prof. Houda Labiod. I am truly grateful for their continuous guidance, motivation and support, without which this work would not have been possible. Thanks to their guidance in both my master's internship and this thesis, I have been introduced to the domain of vehicular networking and become a problem solver. Meanwhile, I would also like to thank Prof. Véronique Vèque for her guidance in both internship and the thesis. During my research work, all of them kept immense interest on my topic of research and have always been available to offer valuable comments and advises.

A special thank goes to Dr. Jun Zhang. Working with him helped me with a deeper understanding on the topic of vehicular networking. The work on clustering and data dissemination algorithms in VANETs would not have been realized without the numerous discussions we have had. Thanks to his guidance in both my master's internship and this thesis. I would also like to thank Prof. Ahmed Serhrouchni for his help during my research life.

I gratefully acknowledge the members of my Ph.D defense committee. Sincere thanks to Prof. André-Luc Beylot and Prof. Antonella Molinaro for devoting their time to reading this manuscript and providing insightful comments. I thank also Prof. Jérôme Härri, Prof. Brahim Bensaou, and Sir Alain Serval, who have kindly accepted to serve on the committee.

I gratefully acknowledge all the members of Environnement de Réseaux Autonomes (ERA) of Université de Technologie de Troyes (UTT) and all the members of Informatique et Réseaux (INFRES) of Télécom ParisTech, for their enormous help and friendship from my master period to Ph.D period. I thank every lecturer, who have given excellent courses and guided my experimental works. I thank all the colleagues, for sharing a comfortable research environment with me. Although I cannot name everyone, I have felt the kindness and friendship from them over coffee-talks, discussions and even smiles we exchanged when crossed in the corridor.

I would like to express my gratitude to all the secretaries who have kindly helped me in

UTT and Télécom ParisTech, Mesdames Pascale Denis, Isabelle Leclercq, Thérèse Kazarian, Lidia Forero, and Céline Bizart. Thank for their availability and help during my study at UTT and Télécom ParisTech.

My time at UTT and Télécom ParisTech was made enjoyable and memorable in large part due to my dear friends. Particular thanks go to Chen Guo, Yi Huang, Yanyan Yang, Lingxue Yang, Xi Jia, Xufei Zhang, Hao Cai, You Wang, and Pengwenlong Gu for accompanying me all the way and letting me not give up in front of stress and pressure. Words cannot express my gratitude to them.

I would like to thank China Scholarship Council (CSC) for its generous financial support during the 60 months, which made possible my study in France.

Last but not least, I owe my deepest gratitude to my parents for their endless understanding, support and love. They have always been the source of courage in my life.

Dedicated to my beloved parents

Abstract

Due to the rapid development of wireless communication technologies and the growing demand of services in Vehicular Ad-hoc Networks (VANETs), efficient clustering algorithms are mandatory to solve the network scalability problem and to support more applications in VANETs. However, due to the dynamic nature of VANETs network topology, clustering algorithms are required to guarantee the cluster's stability. Therefore, this thesis aims to propose a framework for clustering algorithms for VANETs, to improve cluster's stability and to increase the efficiency of data transmission. In this thesis, we firstly investigate a new mobility-based scheme for dynamic clustering (MoDyC) in VANETs using the mobility-based clustering metric to construct clusters. Then, we propose our framework for clustering algorithm named UFC. Based on this framework, we evaluate the impacts of different clustering metrics and cluster merging schemes on cluster's stability. Meanwhile, a leadership-based cluster merging scheme (LCM) is presented to increase the cluster stability. To analyze the cluster stability, we propose a Markov chain model to model the clustering process and to investigate its performance. Finally, this thesis studies a scenario, aiming to disseminate vehicular emergency messages through cluster-based data dissemination scheme.

Keywords: Intelligent Transportation Systems (ITS), Vehicular Ad-hoc Networks (VANETs), clustering, cluster merging, data dissemination, Markov chain.

Résumé

En raison de l'essor rapide des technologies de communication sans fil et de la demande croissante de services dans les réseaux ad hoc véhiculaires (VANETs), le besoin d'algorithmes de clustering efficaces est nécessaire pour résoudre le problème de mise à l'échelle et le besoin croissant d'applications dans les VANETs. Cependant, à cause de la nature dynamique des VANETs, les algorithmes de clustering doivent assurer la stabilité du cluster. Pour cela, l'objectif de la thèse est de proposer un cadre générique servant à définir des algorithmes de clustering afin d'améliorer la stabilité du cluster et d'augmenter l'efficacité de la transmission de données. Dans cette thèse, nous introduisons un algorithme de clustering (MoDyC) qui se base sur la métrique de mobilité pour former des clusters. Ensuite, nous proposons un cadre générique servant à définir des algorithmes de clustering appelé UFC. Nous évaluons l'impact des métriques de clustering et des méthodes de fusion de clusters sur le critère de stabilité. Nous proposons une nouvelle méthode de fusion de cluster appelée LCM. Cette méthode basée sur la métrique Leadership permet d'augmenter la stabilité du cluster. Nous analysons les performances des algorithmes de clustering proposés. Dans le but de modéliser le processus de clustering, nous proposons un modèle basé sur les chaînes de Markov. A partir de ce modèle, nous analysons la stabilité du cluster. Cette thèse se termine par l'étude d'un scénario de diffusion de messages d'urgence en utilisant un schéma de diffusion de données en cluster.

Mots-clés : Réseaux ad-hoc véhiculaires (VANETs), Systèmes de Transport Intelligents, clustering, dissémination des données, chaînes de Markov.

Contents

Acknowledgements	i
Abstract	v
Résumé	vii
List of Tables	xiv
List of Figures	xvii
List of Acronyms	xxi
1 Introduction	1
1.1 Overview of Vehicular Ad-hoc Network	1
1.1.1 Characteristics and challenges	1
1.1.2 Technologies	3
1.1.2.1 Dedicated Short Range Communication (DSRC)	3
1.1.2.2 Cellular-V2X (C-V2X) communication	4
1.1.3 Applications	4
1.2 Motivations and Objectives	6
1.3 Thesis Contributions and organization	7
2 State-of-the-art	11
2.1 Introduction	11
2.2 Overview on clustering algorithms in VANETs	12
2.3 History of VANETs Clustering algorithms	14
2.4 Clustering mechanisms in VANETs	16
2.4.1 Cluster head selection approaches	17

2.4.1.1	First Declaration Wins	17
2.4.1.2	First vehicle in the moving direction	18
2.4.1.3	Weighted sum	18
2.4.1.4	Aggregate relative mobility	21
2.4.1.5	Dominating set based	22
2.4.1.6	Other cluster head selection approaches	23
2.4.2	Cluster construction approaches	23
2.4.2.1	Single-hop cluster	24
2.4.2.2	Multi-hop cluster	25
2.4.3	Cluster maintenance approaches	26
2.4.3.1	Vehicle leaving and joining process	26
2.4.3.2	Cluster merging process	27
2.5	Classification of clustering algorithms	28
2.5.1	Clustering for Context-based Applications	28
2.5.1.1	Cluster-based information dissemination	29
2.5.1.2	Routing	30
2.5.1.3	Traffic density estimation	30
2.5.1.4	Traffic safety	31
2.5.1.5	Traffic prediction	31
2.5.1.6	Data aggregation	31
2.5.1.7	MAC clustering	33
2.5.2	Hybrid clustering	33
2.6	Performance Evaluation	36
2.6.1	Performance metrics	37
2.6.1.1	Cluster performance metrics	38
2.6.1.2	Network performance metrics	38
2.6.2	Simulation scenarios	39
2.7	Conclusion	41
3	MoDyC: A Mobility-based Scheme for Dynamic Clustering in VANETs	43
3.1	Introduction	43
3.2	Mobility-based Clustering Algorithm	43
3.2.1	Cluster Definition	44
3.2.2	Cluster State Transition	45
3.2.3	Cluster Formation	47

3.2.4	Cluster Head Selection	48
3.2.5	Gateway Node Selection	49
3.2.6	Cluster Maintenance	49
3.2.6.1	Cluster merging	50
3.2.6.2	Leaving a cluster	51
3.2.6.3	CML and GW updating	51
3.2.7	Important Messages	52
3.3	Performance Evaluation	53
3.3.1	Performance Metrics	54
3.3.2	Results Analysis	55
3.3.2.1	Impact of "Safe Distance Threshold" D_t	55
3.3.2.2	Impact of Beacon Interval (BI)	55
3.3.2.3	Impact of maximum lane speed	56
3.4	Conclusion	60
4	A Unified Framework of Clustering Approach (UFC) in VANETs	63
4.1	Introduction	63
4.2	Description of The Framework	64
4.2.1	Vehicle states	66
4.2.2	Neighbor Sampling (NS)	66
4.2.3	Backoff-based CH Selection (BCS)	68
4.2.3.1	Metric-based CH selection method	68
4.2.3.2	Random-based CH selection method	69
4.2.4	Cluster Formation	70
4.2.5	Backup CH based vehicle re-clustering	70
4.2.5.1	Backup CH List (BCHL) creation	71
4.2.5.2	Backup CH List (BCHL) ranking	72
4.2.6	Cluster maintenance	72
4.2.6.1	CML updating	72
4.2.6.2	BCHL updating	73
4.2.6.3	Leaving a cluster	73
4.2.6.4	Cluster merging	73
4.2.7	Main messages	74
4.3	Performance Evaluation	74
4.3.1	Testing scenarios	75

4.3.2	Performance metrics	76
4.3.2.1	Macroscopic performance	77
4.3.2.2	Microscopic performance	78
4.3.3	Performance optimization of UFC	79
4.3.3.1	Backoff timer T_W calculation	79
4.3.3.2	Cluster capacity MAX_CM	82
4.3.3.3	Backup CH number MAX_BCH	83
4.3.4	Performance comparison with VMaSC and Lowest-ID	84
4.3.5	Summary of observations	90
4.4	UFC Analysis	90
4.4.1	Overhead analysis	90
4.4.2	UFC performance analysis	91
4.5	Conclusion	92
5	Impact of Merging Schemes on VANET Clusters Stability	93
5.1	Introduction	93
5.2	Related Work	94
5.3	Description of Cluster Merging Scheme	95
5.3.1	Merging condition check	95
5.3.2	Leadership-based CH_{merge} selection	96
5.4	Scheme Analysis	98
5.5	Performance Evaluation	100
5.5.1	Testing scenarios	100
5.5.2	Performance metrics	101
5.5.2.1	Macroscopic performance	102
5.5.2.2	Microscopic performance	102
5.5.3	Performance analysis	103
5.6	Conclusion	105
6	A Stochastic Model for Vehicle Clustering Performance Analysis	107
6.1	Introduction	107
6.2	Analytic Model	108
6.3	Analysis of Clustering Performance	110
6.3.1	Cluster head lifetime	112
6.3.2	Cluster member lifetime	112

6.3.3	Candidate cluster member lifetime	112
6.3.4	Time duration in initial state	113
6.3.5	Prediction of state distribution during the clustering process	113
6.4	Results and Discussion	114
6.4.1	Vehicle state lifetime	114
6.4.2	Prediction of future state distribution	116
6.5	Conclusion	118
7	Cluster-based Emergency Message Dissemination in VANETs	121
7.1	Introduction	121
7.2	Description of the scheme	121
7.3	Performance Evaluation	123
7.3.1	Testing scenarios	124
7.3.2	Performance metrics	125
7.3.3	Performance analysis	126
7.4	Conclusion	128
8	Conclusions and Perspectives	133
8.1	Conclusions	133
8.2	Perspectives	135
8.2.1	Extension of UFC	135
8.2.2	Extension of stochastic model for clustering	136
8.2.3	Perspectives of clustering algorithms	136
	Bibliography	138
A	List of publications	153
B	Résumé Français	155

List of Tables

1.1	List of use cases	5
2.1	Comparison of previous surveys	14
2.2	Number of citations	16
2.3	Classification of clustering algorithms	32
2.4	Hybrid cluster architecture	34
2.5	Existing performance metrics	37
2.6	Comparison of urban simulation scenarios	39
2.7	Comparison of highway simulation scenarios	40
3.1	Notations	45
3.2	List of important messages	52
3.3	Simulation parameters	54
4.1	Notations	65
4.2	List of important messages	74
4.3	Message exchanging	75
4.4	Vehicle setting for scenario A.1	75
4.5	Vehicle setting for scenario A.2	75
4.6	Vehicle setting for scenario B.1 and B.2	76
4.7	Testing scenario settings	76
4.8	Default simulation parameters	78
4.9	Settings of PHY and MAC layer	79
4.10	Average CH lifetime (s)	86
4.11	Average number of clusters	86
4.12	Average CM lifetime (s)	88

4.13	Average number of CMs	88
5.1	Default simulation parameters	101
6.1	Simulation results of the averaged CH and CM lifetime on NS2	115
6.2	Stationary distribution Q^*	117
7.1	Vehicle setting for scenario A.1	123
7.2	Vehicle setting for scenario A.2	123
7.3	Vehicle setting for scenario B.1 and B.2	123
7.4	Testing scenario settings	124
7.5	Default simulation parameters	124
7.6	Settings of PHY and MAC layer	125
B.1	Classification des algorithmes de clustering	164
B.2	Architectures de cluster hybrides	165
B.3	Métriques de performance	166

List of Figures

1.1	ITS-Intelligent Transportation Systems	2
1.2	V2X communications	3
1.3	Overview of the thesis	8
2.1	Cluster topology	13
2.2	The development of clustering algorithms in VANETs. (Color blue: without comparison; Color black: with comparison.)	15
3.1	Clusters (TR : Transmission Range; L : cluster length; D_t : Safe Distance threshold; GW_i, GW_b : Gateway node.)	44
3.2	State transition machine	46
3.3	Cluster formation	47
3.4	Cluster merging (L_{merge} : length of the merged cluster; D_t : Safe Distance threshold.)	50
3.5	Impacts of D_t on cluster performance	56
3.6	Impact of BI on cluster performance with the increased maximum lane speed (MLS)	57
3.7	Average duration of each vehicle state of MoDyC	58
3.8	Comparison of vehicle states' lifetime under the impact of vehicle's MLS (Confidential Interval: 95%)	58
3.9	Vehicle state number comparison under the impact of vehicle's MLS (Confidential Interval: 95%)	59
3.10	Cluster stability comparison under the impact of vehicle's MLS (Confidential Interval: 95%)	60
4.1	Clusters C_i and C_j	64

4.2	State transition	66
4.3	Vehicles' speed distribution	77
4.4	Impact of T_W calculation method on the macroscopic performance of UFC .	80
4.5	Impact of T_W calculation method on the microscopic performance of UFC .	81
4.6	Impacts of cluster capacity (MAX_CM) on cluster performance of UFC . .	82
4.7	Impacts of MAX_BCH on CM re-clustering performance	84
4.8	CH lifetime comparison between UCF, LID, and VMaSC	85
4.9	CM lifetime comparison between UFC, LID, and VMaSC	87
4.10	Cluster performance comparison between UFC and LID scheme	89
5.1	Cluster C_f and C_b	95
5.2	Impacts of MI on CH lifetime.	102
5.3	Impacts of MI on CM lifetime.	103
5.4	Impacts of MI on the average role change rate.	104
5.5	Impacts of MI on CM disconnection rate.	105
6.1	Markov chain model of vehicle clustering state transition	108
6.2	State transition probabilities $p_{ij}(t)$	109
6.3	2-state Markov chain model	111
6.4	Expected state duration E_{s_i} ($t = 700s$)	115
6.5	Expected state duration E_{s_i} ($t = 3000s$)	116
6.6	Expected state duration E_{s_i} ($300 - 330s$)	117
6.7	State distribution $\mathbf{Q}(t)$ ($t = 3000s$)	118
7.1	Emergency message dissemination	122
7.2	Packet Delivery Ratio (%)	126
7.3	Average Packet Delivery Delay (s)	127
7.4	Packet Delivery Ratio (%) under Scenario A.1	129
7.5	Packet Delivery Ratio (%) under Scenario A.2	130
7.6	Packet Delivery Ratio (%) under Scenario B.1	131
7.7	Packet Delivery Ratio (%) under Scenario B.2	132
B.1	STI-Systèmes de Transport Intelligents	156
B.2	Topologies de cluster	160
B.3	Le developement des algorithms de clustering dans les VANETs. (Couleur bleu: sans comparaison; Couleur noir: avec comparaison.)	161

B.4 Clusters (TR : Transmission Range; L : cluster length; D_t : Safe Distance threshold; GW_i, GW_b : Gateway node.)	167
B.5 Machine d'états finis	168
B.6 Impacts de D_t sur les performances du cluster	169
B.7 Comparaison de nombre d'état de véhicule sous l'impact de MLS de véhicule (Intervalle de confiance : 95%)	169
B.8 Comparaison de la stabilité du cluster sous l'impact de MLS (Intervalle de confiance : 95%)	170
B.9 Machine d'états finis	171
B.10 Comparaison de la durée de vie de CH entre UCF, LID, et VMaSC	172
B.11 Comparaison de la durée de vie de CM entre UFC, LID, et VMaSC	173
B.12 Comparaison des performances de clustering entre l'UFC et LID	174
B.13 La durée de vie de CH sous l'impact de MI	176
B.14 Le taux de déconnexions de CM sous l'impact de MI	177
B.15 La chaîne de Markov de 2-état	178
B.16 La durée d'état prévue E_{s_i} ($t = 3000s$)	179
B.17 La distribution des états $\mathbf{Q}(t)$ ($t = 3000s$)	180

List of Acronyms

3GPP Third Generation Partnership Project

5G Fifth Generation

CAM Cooperative Awareness Message

C-ITS Cooperative-Intelligent Transportation Systems

D2D Device-to-Device

DENM Decentralized Environmental Notification Message

DSRC Dedicated Short Range Communication

ETSI European Telecommunications Standards Institute

FCC Federal Communications Commission

GPS Global Positioning System

HetVNET Heterogeneous Vehicular NETwork

IEEE Institute of Electrical and Electronics Engineers

LTE Long Term Evolution

MANET Mobile Ad-hoc NETwork

OBU On Board Unit

RSS Received Signal Strength

RSU Roadside Unit

SUMO Simulation of Urban MObility

TR Transmission Range

UMTS Universal Mobile Telecommunications System

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

V2R Vehicle-to-Roadside Unit

V2X Vehicle-to-Everything

VANET Vehicular Ad-hoc NETWORK

WAVE Wireless Access in Vehicular Environment

WiFi Wireless Fidelity

Introduction

1.1 Overview of Vehicular Ad-hoc Network

Vehicular Ad-hoc Network, known as VANET, is a self-organizing network formed by a collection of moving vehicles [1]. With the rapid development of automotive manufacturing, vehicles are becoming more and more intelligent and powerful. Vehicles now can communicate with other vehicles directly through wireless technologies in a V2V (Vehicle-to-Vehicle) manner or indirectly through roadside infrastructures in a V2I (Vehicle-to-Infrastructure) or I2V (Infrastructure-to-Vehicle) manner [2]. Moreover, vehicles can also connect with motorcycles via V2M (Vehicle-to-Motorcycle) communication and pedestrians via V2P (Vehicle-to-Pedestrians) communication. Meanwhile, the development of cellular technologies, such as Long-Term Evolution (LTE), enables vehicles to communicate directly with the cellular network. Collectively, these wireless connections are referred to as V2X (vehicle-to-everything) communication, which can support numerous use-cases of Cooperative Intelligent Transportation Systems (C-ITS), including both safety and non-safety services. Figure 1.1 presents the future Intelligent Transportation Systems depicted by the European Telecommunications Standards Institute (ETSI) [3].

In the rest of this chapter, we overview the characteristics of VANET and introduce the associated technologies, as well as the applications of C-ITS. Furthermore, we identify the challenges in VANET and address the motivations and objectives of this thesis. At last, we present the contributions and organization of this thesis.

1.1.1 Characteristics and challenges

Compared with the traditional Mobile Ad-hoc Network (MANET), the specific characteristics and challenges of VANET have attracted researchers and industry from different

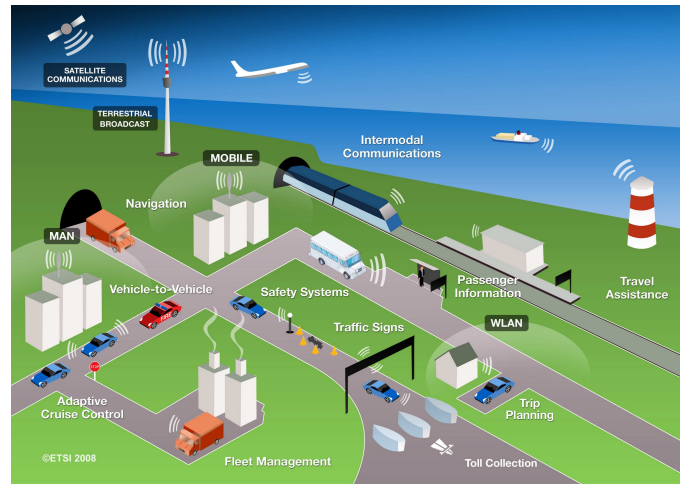


Figure 1.1: ITS-Intelligent Transportation Systems

fields to investigate VANET applications, technologies, protocols, and standards. VANET inherits similar features from MANET; however, shows some unique characteristics, leading to some new challenges.

- Predictable vehicle mobility: Instead of moving randomly in MANET, vehicles' mobility is constrained by road topologies, as well as traffic conditions, traffic lights, and road signs. Thus, a vehicle's trajectory is predictable which can assist some ITS applications.
- Various vehicle mobility patterns: VANET is characterized by varying mobility patterns. The velocities of trucks, cars, and motorcycles are usually different. Meanwhile, the velocity limitations are different according to traffic conditions.
- Highly dynamic network topology: Due to the fast movement of vehicles on the road, especially under highway scenario, the network topology changes rapidly. A vehicle with higher speed may pass another vehicle quickly, causing unstable inter-vehicle connections. Therefore, the information transmission between these vehicles is not reliable, which may further cause traffic safety problems. A large amount of research works, such as [4][5], focus on enhancing the information transmission reliability under highly dynamic network topology, especially when transmitting emergency messages.
- Unlimited network scale: The network scale of VANET may be huge, such as in big dense urban scenarios. Without a central controller in VANET, the limited transmission range can only support short-range vehicle communications, which is not sufficient to support some VANET services. Instead of this flat network architec-

ture, the hierarchical network architecture can solve the network scalability problem. Moreover, Heterogeneous Vehicular NETwork (HetVNET), which integrates cellular networks with V2V communication technologies, is also a potential solution [6] [7].

1.1.2 Technologies

As shown in Figure 1.2 [8], communications between vehicles, Road Side Units (RSUs), and infrastructures are summarized as Vehicle-to-Everything (V2X) communication. V2X communication can be supported by numerous wireless access technologies. Some of these communication technologies support medium range and short range communications in a distributed manner (e.g., DSRC). In contrast, other technologies support communications in long-range, relying on a centralized infrastructure (e.g., Cellular-V2X). The wireless access technologies are discussed in the following.

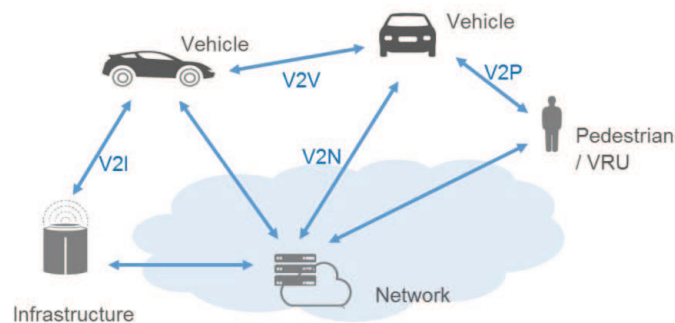


Figure 1.2: V2X communications

1.1.2.1 Dedicated Short Range Communication (DSRC)

In 1999, the U.S. Federal Communications Commission (FCC) allocated 75 MHz (from 5.850 GHz to 5.925 GHz) bandwidth for Dedicated Short Range Communication (DSRC) in vehicular environments. DSRC is based on the IEEE 802.11p/1609 Wireless Access in Vehicular Environment (WAVE) protocols [9], which are particularly designed to meet the requirements of vehicular communications. The term "Short Range" in DSRC means that the communication takes place over hundreds of meters (100m-1000m). In Europe, such as this "Short Range" communication is standardized as ETSI ITS-G5 [10], which is also based on the IEEE 802.11p. Both these IEEE 802.11p based solutions (DSRC/WAVE and ETSI ITS-G5) can support the direct V2V communication and Vehicle-to-RSU (V2R) communication with out infrastructure. In this case, emergency messages can be delivered with lower latency. However, there are some limitations without a central controller (infrastructure), especially in the case of traffic congestion due to the low throughput and

unbounded delay of CSMA/CA under high load.

1.1.2.2 Cellular-V2X (C-V2X) communication

C-V2X communication is the connection with cellular networks. With the wide deployment of cellular networks, numerous services can be realized with the support of a cellular network that can provide larger coverage and higher data rate. In recent years, the most popular cellular technologies that researchers focus on are LTE and Fifth Generation (5G).

LTE-V2X: In general, LTE networks can provide high capacity with wide coverage and support more services for vehicles. In [11], the researchers have analyzed the advantages and challenges of LTE for vehicular networking and compared the main candidate wireless technologies for V2X communications. Moreover, in [12], the authors compared the performance of LTE and Universal Mobile Telecommunications System (UMTS) for vehicular safety applications at intersections. The analysis shows that LTE system can support a demand of transmitting around 1500 Cooperative Awareness Messages (CAM)s [13] per second per cell.

LTE-D2D: Device-to-Device (D2D) communications is one of the techniques that support V2V communication [14]. In D2D, neighboring User Equipment (UE)s can directly communicate with each other. However, devices in LTE system need to discover the existence of their peer before any direct communication. Therefore, the discover process may cause long latency, which is not acceptable for transmitting emergency messages. In this case, DSRC is more suitable for V2V communication.

5G-V2X: Even though LTE can support more V2X use cases, it is expected that more stringent requirements for latency, range, speed, reliability, location accuracy, privacy, and message payloads are needed to support more advanced use cases for future intelligent and on-connected vehicles. In [15], the researchers state that the network slicing technology of 5G, which logically isolates network functions and resources, can effectively cope with a wide variety of use cases with divergent demands provided over the 5G infrastructure by multiple tenants. Meanwhile, they also propose some set of slices for 5G-V2X system.

1.1.3 Applications

C-ITS applications can be categorized into three types, including road safety, traffic efficiency, and entertainment applications. The road safety applications aim to avoid traffic accidents and protect road users' lives. The traffic efficiency applications focus on improving the traffic efficiency, such as avoiding traffic jams. Finally, the entertainment

Table 1.1: List of use cases

Group		Use cases
3GPP		Vehicle Platooning, sensor and state map sharing, automated cooperative driving for short distance grouping, cooperative collision avoidance (CoCA) of connected automated vehicles, information sharing for partial/conditional automated driving, video data sharing for assisted and improved automated driving (VaD), intersection safety information provisioning for urban driving, dynamic ride sharing, emergency trajectory alignment, cooperative lane change (CLC) of automated vehicles.
C-ITS	Day 1	Hazardous location notifications: Slow or stationary vehicle(s) & Traffic ahead warning, Road works warning, Weather conditions, Emergency brake light, Emergency vehicle approaching; Signage applications: In-vehicle signage, In-vehicle speed limits, Signal violation/Intersection Safety, Traffic signal priority request by designated vehicles, Green Light Optimal Speed Advisory (GLOSA), Probe vehicle data, Shockwave Damping (falls under ETSI Category “local hazard warning”).
	Day 1.5	Information on fuelling & charging stations for alternative fuel vehicles, Vulnerable Road user protection, On street parking management & information, Off street parking information, Park & Ride information, Connected & Cooperative navigation into and out of the city, Traffic information & Smart routing.

applications aim at providing more comfortable and convenient journeys for road users. The detailed description of each application type is introduced as follows.

- Road safety applications: These applications require vehicles to detect or gather emergency information immediately by vehicle sensors. Furthermore, the emergency messages should be delivered or disseminated to the target vehicles or areas with short latency and high accuracy. The affected range of these applications are usually small, and the safety messages should not be very large. Among all applications, the road safety application is the most investigated and most deployed. A classification of road safety applications is addressed in the ETSI standard [16], as well as the relative technical requirements.

- Traffic efficiency applications: These applications focus on the overall efficiency of the transportation system. An example is to avoid traffic jams by informing vehicles in the distance to change to the road with lower traffic density. This kind of application usually requires a large effected range. Compared with safety applications, the transmission latency and accuracy are not stringent.
- Entertainment applications: These applications are supported by various communication technologies. The applications, such as Internet access, content downloading, information sharing, always require broad bandwidth and large capacity of storage.

With the development of autonomous driving and V2X communications, more challenging use cases are considered as a basis to derive service requirements. Both recent the Third Generation Partnership Project (3GPP) technical report [17] and C-ITS Platform final report [18] have defined new V2X-based use cases. Table 1.1 lists some of the use cases.

1.2 Motivations and Objectives

Among the ITS applications mentioned above, the most urgent and challenging one is the safety-related application. For this kind of application, information dissemination usually requires low latency, high accuracy, and high reliability. At the same time, VANET has a scalability problem that cannot be solved by a flat network architecture without a central controller. Thus, researchers have proposed a hierarchical network architecture to solve this problem. In such hierarchical network, vehicles are virtually organized into different groups, called **clusters**.

In each vehicle group, there exists one leader and several member vehicles. The leader is responsible for inter-cluster communication, and group members can communicate directly with their group leader [1]. However, the dynamic network topology may cause vehicle groups to be unstable, which makes clusters meaningless, and even leads to unnecessary overhead. In this case, finding an effective method to construct stable and resistant vehicle clusters is indispensable. To this end, our first objective in this thesis is to *design a clustering algorithm for VANET to guarantee cluster's stability*.

Researchers have proposed numerous metrics to construct clusters in their clustering algorithms and have proved the effectiveness of their proposed metrics. However, it is still hard to tell which clustering metric works better in cluster's stability. Moreover, since clustering algorithms are proposed under various scenarios, it is hard to fairly evaluate

these algorithms. Thus, the second objective of this thesis is to *design a framework of the clustering algorithm in VANETs*, in order to fairly evaluate different clustering metrics and analyze their impacts.

During the research, we observed that cluster maintenance process plays an important role to ensure stability. As an indispensable part of cluster maintenance process, cluster merging method may have some influence on clustering performance. However, the method of cluster merging is always overlooked when compared to the cluster head selection and cluster formation process. Based on our proposed clustering framework, this thesis also *evaluates the impacts of cluster merging schemes on clustering performance and designs a new cluster merging scheme*.

To deeply understand clustering algorithm, modeling the entire clustering process mathematically is required. Therefore, another objective of this thesis is to *propose a stochastic model for clustering performance analysis*.

As mentioned above, road safety applications for VANETs depend mainly on disseminating warning messages to deliver information to the concerned vehicles. However, disseminating information suffers from the broadcast storm problem and the interference from the existing periodic single-hop beacon messages. The hierarchical network architecture, such as cluster, can effectively solve these problems by avoiding redundant message transmissions. Meanwhile, content caching methods enable temporary storage of less urgent information so that emergency messages can be sent as quickly as possible. The last objective of this thesis is to *propose a cluster-based data dissemination scheme for emergency message transmission*.

In summary, the main purpose of this thesis is to design effective clustering algorithms to support reliable data transmission in VANETs.

1.3 Thesis Contributions and organization

This thesis brings original contributions, essentially to clustering algorithms and cluster-based data dissemination mechanisms in VANETs. Firstly, we overview the existing clustering algorithms and cluster-based data dissemination mechanisms. Secondly, we propose a mobility-based clustering algorithm MoDyC to improve the cluster stability. Thirdly, to analyze the impacts of different clustering metrics on clustering performance, we introduce a clustering framework UFC. Based on this framework, we discuss the impacts of cluster merging schemes and improve the cluster stability via a new cluster merging method L-CM. Additionally, we model the clustering process as a discrete-time finite-state Markov

chain, which enables us to analyze and predict the clustering performance. Finally, we propose a cluster-based data dissemination scheme with data caching in VANETs. Figure 1.3 illustrates our contributions. The organization of the manuscript is listed in what follows.

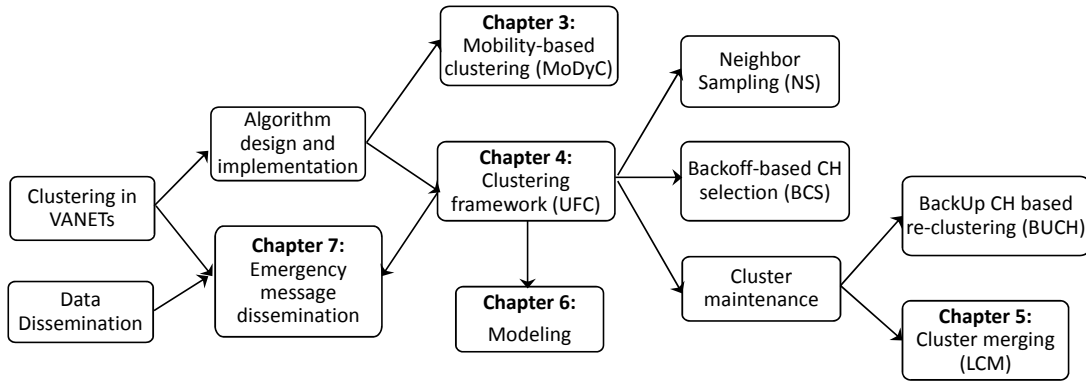


Figure 1.3: Overview of the thesis

Chapter 2 reviews the clustering algorithms in VANETs. Clustering algorithms are divided into three essential procedures: cluster head selection, cluster construction, and cluster maintenance. For each process, we summarize the methods and the challenges. Moreover, we provide a classification of the existing clustering algorithms based on the application contexts. Among them, we focus on data dissemination algorithms based on clustering. This work has been submitted as a survey paper which is under review.

Chapter 3 introduces a mobility-based clustering algorithm for VANETs, named MoDyC, which can improve the cluster's stability. Clusters are established by adding vehicles one by one which is moving towards in the same direction, and the cluster size is restricted by the distance between the cluster head and its potential cluster member. MoDyC establishes one-hop clusters, and the cluster head is the central node of the cluster. To evaluate the cluster's stability, we compare MoDyC with two benchmark clustering algorithms under the highway scenarios, and MoDyC shows higher cluster stability. This work has been published in [19] and [20].

Chapter 4 is dedicated to proposing a unified framework of clustering algorithms, named UFC, which enables a comprehensive analysis of the impacts of different clustering metrics. To analyze the framework, we generate various traffic scenarios via the traffic mobility simulator SUMO. The performance of the framework is validated through self-parameter optimization and the comparison with benchmark algorithms. This work has

been published as a journal paper [21].

Chapter 5 discusses the impact of cluster merging schemes on clustering performance. We overview the cluster merging schemes in the existing clustering algorithms and introduces a new cluster merging scheme, the leadership-based cluster merging scheme, named LCM. LCM scheme is compared with two other cluster merging schemes, based on the proposed clustering framework UFC. The results illustrate that LCM scheme can improve the cluster stability. This work has been presented and published as a conference paper [22].

Chapter 6 proposes a stochastic model to analyze the clustering performance in VANETS. The vehicle state transitions during clustering process are modeled as a discrete-time finite-state Markov chain. Every vehicle acts as one of the predefined clustering states at each time step. From the previous experimental results, we can calculate the state transition probability matrix, which enables the prediction of clustering performance without the limitation of simulation time in the network simulator. This work has been accepted as a conference paper [23].

Chapter 7 proposes an emergency message dissemination scheme combined with UFC clustering algorithm. This work is evaluated by the comparison of combining with another clustering algorithm.

Finally, Chapter 8 concludes this thesis and points out some limitations of the work. Meanwhile, the perspectives of the future work are discussed.

State-of-the-art

2.1 Introduction

For all of the ITS applications that we have mentioned in Chapter 1, the most important and investigated one is the safety-related application. ETSI has provided a detailed classification for road safety applications in the standard [16]. To support these services, information should usually be transmitted with low latency and high accuracy. However, the large number of mobile vehicles and the dynamic network topology cause scalability and unstable connection problems. One efficient solution to these challenges is clustering algorithms, which virtually organize vehicles into groups, named clusters [1].

In past years, many clustering algorithms were proposed in Mobile Ad Hoc Networks (MANETs) (e.g., [24], [25], and etc.), aiming to save network resources and to increase network efficiency. According to the predictable mobility and predefined road topology of VANETs, researchers found that previous clustering algorithms in MANETs were no longer suitable for VANETs. Due to the time of achieving clustering procedure, additional control overheads may be added. Thus, a good clustering algorithm should not only form minimum number of clusters but should also dynamically maintain the cluster structure without large overhead over the network. However, frequent dynamic cluster maintenance may cause unnecessary cluster re-formation. Therefore, cluster topology should not be sensitive to vehicles' movements in order to avoid frequent cluster re-formation.

This chapter surveys the existing clustering algorithms and cluster-based data dissemination algorithms in VANETs from the following aspects: clustering methods, context-based clustering algorithms, and cluster performance evaluation. A complete description of mechanisms in each clustering step is provided in this chapter. Moreover, a detailed comparison and discussion of various approaches and their performance metrics are pre-

sented. The main contributions in this chapter are listed in what follows:

- We summarize and compare the existing surveys by highlighting their advantages and pointing out the limitations. As far as we know, this is the first work of clustering algorithms that compares the existing related surveys;
- We provide an overview of the development of clustering algorithms in VANETs from the year 1999 to the year 2017, which has never been summarized in previous research work;
- We explore more recent research works than previous survey articles, provide a new classification according the context, and highlight the hybrid clustering algorithm, since it plays a more important role with the development of network access technologies;
- We summarize and classify the existing clustering techniques in terms of clustering procedure: cluster head selection, cluster formation, and cluster maintenance;
- We observe various context of the existing clustering algorithms, including hybrid clustering architecture;
- We present a comprehensive analysis of the performance evaluation methods of clustering algorithms, including performance metrics, simulation tools, traffic scenarios, etc.. A new classification of clustering performance metrics is proposed;
- We conclude the challenges and the future development of clustering algorithms in VANETs.

2.2 Overview on clustering algorithms in VANETs

Clusters in VANETs indicate virtual groups of vehicles. Each cluster has one cluster head (CH) that is the leader of the group. A CH is followed by several vehicles, named as cluster members (CMs). Cluster topology can be classified into two categories: one-hop clusters and multi-hop ones, as shown in Figure 2.1 . One-hop clusters are usually constructed based on CH's transmission range (TR). CHs add their one-hop neighbors into clusters. Each CM can communicate directly with its CH via V2V communication; two different CMs can communicate with each other directly or via their cluster head. In recent years, multi-hop clusters (e.g., [26], [27], and so on) are proposed in order to construct more stable clusters, in which not all CMs could communicate with their CH directly.

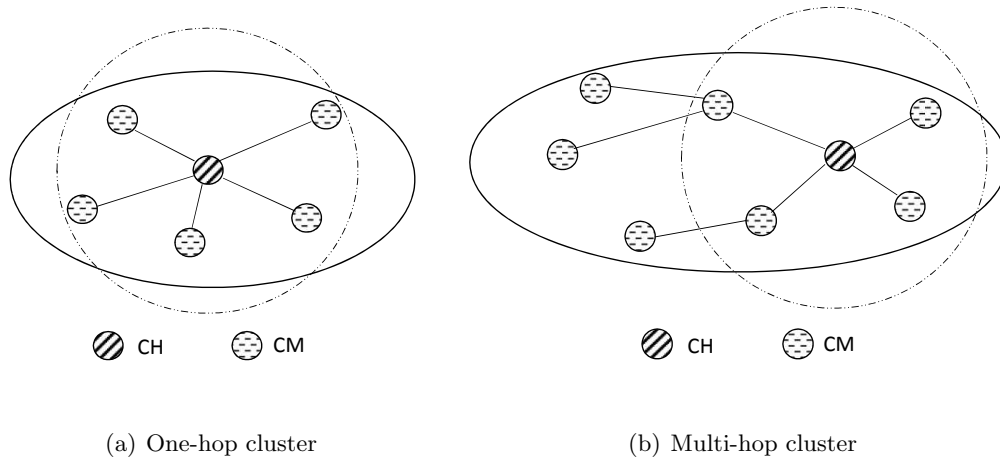


Figure 2.1: Cluster topology

In recent years, several surveys were published, which summarize the existing clustering algorithms in VANETs. In 2012, the authors of [29] have synthesized the clustering algorithms for VANETs from the year 2000 to 2012. They briefly described the clustering algorithms independently, and compared the following contents: clustering metrics, radius (cluster topology), location services, cluster density, and simulators. The authors addressed that there is a lack of a fair comparison of different clustering methods, since the simulation scenarios, network simulators, and performance metrics change substantially in different clustering algorithms. However, this survey is considered incomplete without providing a classification and detailed algorithms comparison.

In the year 2014, another survey [1] has been published, covering more research works. The authors presented a detailed classification of clustering in VANETs based upon various key parameters. The existing algorithms were divided as follows: predictive clustering, backbone based clustering, MAC based clustering, traditional clustering, hybrid clustering, and secure clustering. The authors described the algorithms precisely in each category; meanwhile, a comparison and short discussion have been given. Compared to the previous survey, this work is much more complete and meaningful; however, the parameter comparisons among different algorithms are not persuasive enough only with the fuzzy words “low”, “medium”, and “high”.

The latest survey of VANETs clustering techniques has been published in the year 2016 [31]. Compared with other existing surveys, it applied a more complete taxonomy of VANETs clustering techniques, especially from the aspects of applications of clustering and the problem of evaluating and comparing the performance of clustering algorithms, which have not been discussed in previous surveys. Moreover, the authors precisely

summarized the strategies for each part of clustering procedure, including cluster head selection, cluster formation, and cluster maintenance. It provides comprehensive analysis and comparisons; however, the authors have not analyzed the context and simulation scenarios of the algorithms, and there is a lack of comparison of performance metrics which is important in comparing clustering algorithms. In Table 2.1, a comparison of the existing surveys is summarized.

Table 2.1: Comparison of previous surveys

Comparison aspects		2012 [29]	2014 [1]	2016 [31]	Our work
Overview	Number of algorithms	19	43	45	66
General	Algorithm classification		✓	✓	✓
	Application			✓	✓
	Architecture				✓
Clustering techniques	Cluster head selection	✓	✓	✓	✓
	Cluster formation			✓	✓
	Cluster maintenance			✓	✓
Performance evaluation	Performance metrics				✓
	Simulator			✓	✓
	Traffic scenario				✓
	channel model			✓	✓

2.3 History of VANETs Clustering algorithms

The VANETs clustering techniques started to be developed in the early 1990s, and have increased in recent years. Figure 2.2 provides a blueprint for the development of clustering algorithms proposed from the early 1990s to the year 2017. The X-axis indicates the year, and the Y-axis shows the proposed algorithms. In this figure, a relation among these existing clustering algorithms is presented. The black arrow indicates that the following algorithm is compared with the original one in the simulation, and the blue arrow shows that the following algorithm follows the framework of the original one; however, it did not make a comparison.

In Figure 2.2, about 51 existing clustering algorithms have been observed from the early 1990s to the year 2017. Among these clustering algorithms, around 30% algorithms are derived from the idea of the earliest Lowest-ID (LID) algorithm [32], which was originally proposed for MANETs to increase the communication efficiency among mobile nodes. In

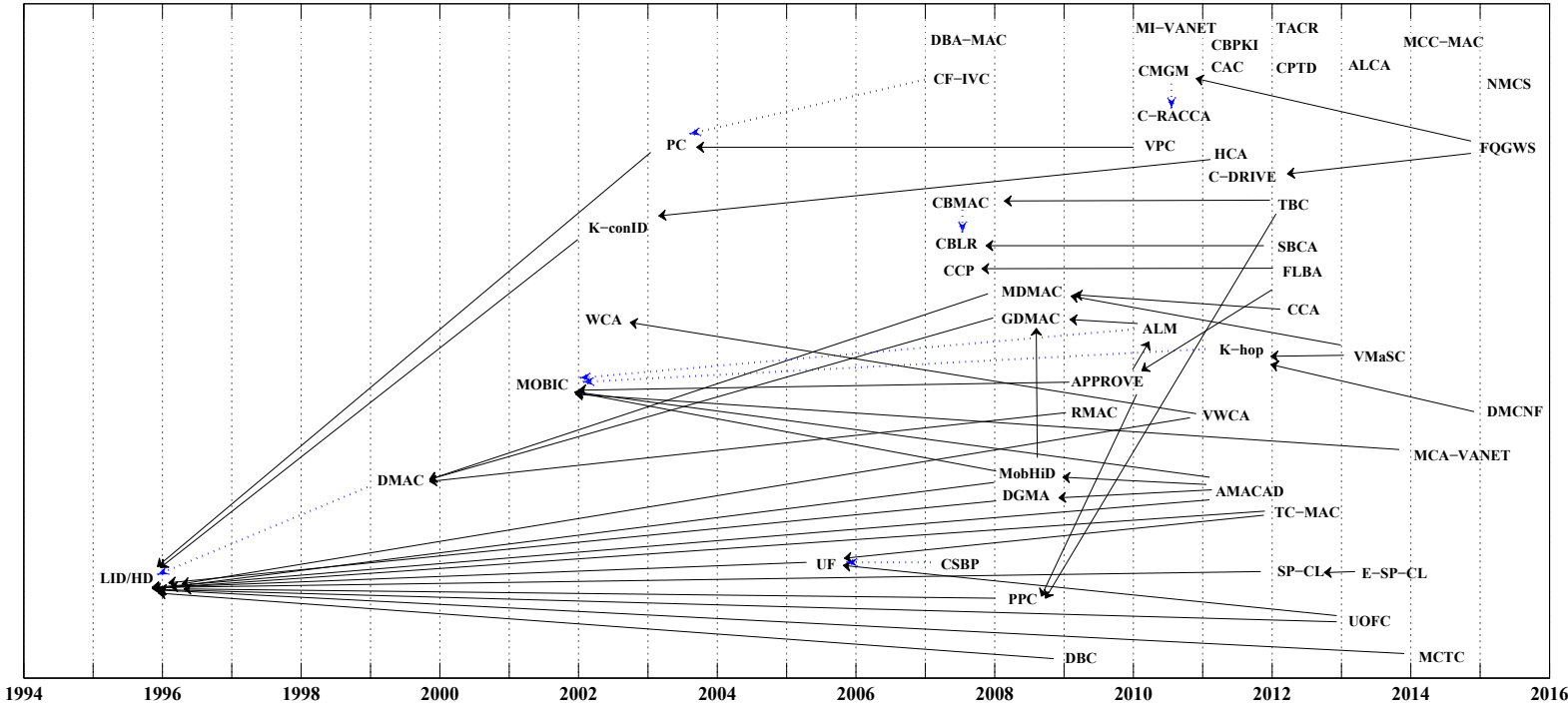


Figure 2.2: The development of clustering algorithms in VANETs. (Color blue: without comparison; Color black: with comparison.)

the year 2005, the authors summarized the existing clustering algorithms in MANETs in [34]. In this article, MANETs clustering algorithms have been classified based upon different metrics, including DS-based (Dominating Set based), low-maintenance, mobility-aware, energy-efficient, low-balancing, and combined-metrics-based clustering.

With the increased popularity of VANETs, some MANETs clustering algorithms were deployed to adapt to the particular characteristics of vehicular communications. It is easy to observe that after the year 2005, plenty of clustering algorithms were published for VANETs. Furthermore, most of these were derived from the previous MANETs clustering algorithms, including DMAC [35], MOBIC [36], [24], K-ConID [37], and PC [38].

Since 2010, numerous applications in VANETs were explored, and clustering algorithms were designed based on the requirements of solving specific problems, instead of purely to increase the **cluster stability**. Besides one-hop cluster topology, multi-hop cluster was widely accepted (e.g., K-hop [26], HCA [27], etc.) because of the increased application requirements and the limited range of wireless transmission. Meanwhile, the development of cellular technologies, including UMTS and LTE, expanded the deployment of clustering algorithms in vehicular networks. Hybrid clustering architecture, including both V2V and V2I communication manners, is becoming a new trend (e.g., [39], [40]), enabling more applications in vehicular networks. More detailed discussions are given in

the rest of this chapter.

Table 2.2: Number of citations

Algorithm	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	Total	Mean
APPROVE-2 [41]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8	18	25	12	63	15.75
K-hop [26]	N/A	N/A	N/A	N/A	N/A	2	8	8	14	23	12	67	9.57
ALM [42]	N/A	N/A	N/A	N/A	5	12	6	11	18	16	19	87	10.88
C-RACCA [43]	N/A	N/A	N/A	3	8	8	22	19	14	14	14	102	12.75
APPROVE-1 [44]	N/A	N/A	1	4	9	29	19	22	20	30	10	144	16
PPC [45]	N/A	2	10	15	9	12	15	22	21	21	19	146	14.6
CBMAC [46]	N/A	N/A	5	7	5	9	15	11	6	12	4	74	6.73
CCP-2 [47]	N/A	1	15	16	29	26	34	39	27	31	24	242	22
CCP-1 [48]	5	8	6	7	14	13	8	17	18	9	4	109	9.08

Based on the clustering algorithms shown in Fig. 2.2, the most popular algorithms that were particularly designed for VANETs are selected: PPC [45], ALM [42], k-hop [26], APPROVE [41], C-DRIVE [49][50][50], CBMAC [46], CCP [47], and C-RACCA [43]. The criteria of popularity are based on the number of citations and the frequency of this algorithm been compared with other algorithms. We observe the frequency of the article being cited from the year 2006 to the year 2017 and analyze the development trend of clustering algorithms. In Table 2.2, it is easy to observe that multi-hop cluster topology [26] is becoming popular in recent years.

2.4 Clustering mechanisms in VANETs

Vehicular clusters' construction is a dynamic procedure due to the high mobility of vehicles and intermittent communications. Vehicles have to obtain the necessary information from their neighboring vehicles, including their identities, positions, velocities, etc.. The potential cluster head will be selected based upon the criteria, such as relative mobility, received signal strength, link lifetime, etc.. Clusters will be established by adding potential vehicle members. In [31], a basic flow of clustering algorithm was introduced, indicating the general steps of clustering algorithms.

This section presents a summary of the approaches and the criteria of each of the clustering steps, including cluster head selection, cluster formation, and cluster maintenance. For each method, a discussion of the advantages and disadvantages is given.

2.4.1 Cluster head selection approaches

In clustering algorithm, the most vital part is to select a reliable leader, which can achieve the highest stability among its vicinities. In most clustering algorithms, CHs are selected at the beginning of the clustering process. Then, the cluster formation process is controlled by the selected CHs. In this subsection, the CH selection approaches are classified and analyzed.

2.4.1.1 First Declaration Wins

The First Declaration Wins (FDW) mechanism was firstly proposed in the Passive Clustering algorithm (PC) [38] in the year 2002. FDW is a cluster head election rule which does not require any metric information. FDW is based on the idea of contention, in which a vehicle, firstly claiming to be a cluster head, dominates the other vehicles within its transmission range.

In CF-IVC [53], the authors proposed a clustering formation protocol for inter-vehicle communication, based on the Passive Clustering (PC) model proposed in [38]. FDW rule has been applied in this mechanism. The first node, relaying the received packet with its CH claim, wins the competition. Any neighboring node that receives the cluster head information would change its state to be CMs.

Another protocol that applied FDW rule for CH election is PassCAR [54]. It proposed a passive clustering aided routing protocol for VANETs. If two nodes are in the CH_{READY} state, the FDW mechanism ensures that only one node will become the CH. Another node in the transmission range of CH will become the CM.

A contention based CH selection algorithm has been introduced in a Unified Framework of Clustering algorithm (UFC) [21]. In the initial state, every vehicle calculates a contention timer based on its neighbors' information. All vehicles will start to count down the contention timer at the same time, and the first one finish the countdown will become the CH and broadcast a CH announcement message.

Discussion: The FDW CH selection method can reduce the exchanged packets. Initial clusters can be established very fast without any additional restrictions. However, the stability of the formed clusters cannot be guaranteed since there are no criteria to evaluate the link connections between CMs and their CH. The method proposed in UFC [21], which is also a part of this thesis, solved this problem by adding a velocity criteria before FDW.

2.4.1.2 First vehicle in the moving direction

C-DRIVE [49] [55] was firstly proposed in the year 2009, aiming at estimating the density of vehicles on a given road segment. The authors supposed that each vehicle is equipped with a digital map to determine the direction that it will travel. The first vehicle entering the region in a particular direction is considered as CH. Each CH is responsible for computing the density of vehicles in its cluster; this information will help the traffic signal to adjust its signal timings and thus to manage the traffic effectively.

MC-DRIVE [50] is a modified C-DRIVE [55] method, to estimate vehicles' density at the intersection. The first vehicle entering the road is considered as a temporary CH, which is designed for leading the cluster formation process.

In [56], the authors proposed a multi-hop cluster-based routing protocol, named CONVOY, for highway scenario. The CH is simply the vehicle moving at the first place of the convoy. The CH leads cluster formation and controls the size of vehicle convoy.

Another typical vehicle cluster method is called platoon. Vehicle platooning is a technique where highway traffic is organized into groups of close-following vehicles called platoon or convoy. The idea of organizing traffic in platoons to dramatically increase road capacity is originally proposed in [57] by PATH for Intelligent Vehicle Highway System (IVHS). In 2015, a platoon management protocol for Cooperative Adaptive Cruise Control (CACC) was proposed in [58]. A platoon leader is the first vehicle in the platoon with at least one follower. With CACC technique, the headway can be reduced to further improve the platooning efficiency.

Discussion: The approach to select the first vehicle in the moving direction as the CH is easy to realize. This method is more suitable for the intersection area. Once the CH is chosen, it will directly add the running vehicles behind it as the CMs. However, the selected CH could frequently be overtaken on the road. How to update the CH information and reduce the control overhead is a big challenge.

2.4.1.3 Weighted sum

In VANETs, CH selection is usually based on multiple mobility metrics, such as speed, position, node degree (number of neighbors), and link lifetime. Each protocol has its cluster stability definitions. The cluster head selection is based on the following two conditions: vehicle's mobility pattern and the relative mobility respect to vehicle's neighbors. Mobility based metrics are widely used in recent VANETs CH selection, such as speed, position, link lifetime and node degree (usually the number of one-hop neighbors).

According to the research, it is easy to observe that most of the CH selection methods are based on the weighted sum of various parameters. Weight based clustering method was originally proposed for MANETs in [24]. The CH is selected based upon the combined weight value, including different metrics values, and the weighting factors are distributed to each of these metrics.

In [59], the authors proposed a compound clustering scheme based on two groups of parameters: Lowest-ID or Highest Degree, and the traffic specific position or Closest Velocity to Average algorithms, to find a more stable clustering method. A utility function was proposed, presented as the weighted sum (utility) for each vehicle. The vehicle with the highest utility is selected to be the CH. The implementation uses a weighting factor of 85% to the Lowest-ID or Highest-Degree method and 15% to the traffic-specific information of position or velocity. However, the authors did not specify how to define the weighting factor.

DBC [60] proposed a new multilevel clustering algorithm for VANETs based on vehicle's connection level estimation, link quality level estimation, and traffic condition estimation. Connectivity level estimation is to estimate the density. Link quality is presented by signal-to-noise (SNR). The notion Group Membership Lifetime (GML) counter is proposed to indicate the duration of the node links to other group members. GML is used to check the reliability of node. The vehicle with the most stable communication with other group members gains the highest weight and becomes CH. However, the calculation of the weight value has not been discussed in the paper.

In [62], the authors proposed a lane-based clustering algorithm (TC-MAC-1). The main idea is to select a CH based on the lane where most of the traffic appears. It is based on the assumption that each vehicle knows its exact lane on the road via a lane detection system and in-depth digital street map. With the knowledge of lane information, vehicles moving on the road can be easily separated by three types: Left Turn, Right Turn, and No Turn. Like previous weighted sum method, each vehicle computes and broadcasts its CH level, based on the information of network connectivity level, average distance level, and average velocity level. Here, the lane weighting factor is determined based on the ratio of the number of lanes for each traffic flow to the total number of lanes on the roadway.

The authors proposed VWCA clustering algorithm in [63], based on the original WCA [24] and entropy-based clustering algorithm [64]. The authors defined the entropy value of local networks, where the "local networks" denotes a vehicle's neighborhood list. Entropy value has been proved to reduce the frequency of cluster re-affiliation [63] effec-

tively. Besides, the authors proposed three novel parameters for the weight calculation: vehicle's distrust value, relative moving direction, and the number of neighbors. Each vehicle calculates its combined weight value in a distributed manner. The one with the minimum combined weight value is selected to be CH. The authors evaluated the influence of weighting factors on the cluster performance. The weighting factors can be dynamically chosen according to their purpose, which is considered as a great improvement.

The authors of [65] proposed a destination-based clustering protocol (AMACAD). Each vehicle calculates a weighted value between one of its neighbors, $F_{v,z}$, based on relative distance, relative speed, and relative destination. Cluster head is selected according to the sum of $F_{v,z}$. The effects of weighting factors on the number of cluster and CH lifetime were tested. Similar to VWCA [63], the weighting factors are dynamically assigned according to the different scenario.

In [66], the authors proposed a fuzzy-logic based CH selection method (FLBA). Each vehicle calculates a stability factor SF based on its average speed difference respect to all of its neighbors. CH selection is based on a weighted stability factor SF_w , represented by the weighted sum of the vehicle's current SF_w and the previous SF_{w-1} . The vehicle with the highest SF_w is selected as CH. The weighting factor, called smoothing factor, is predetermined as 0.5; however, the authors did not explain the reason for this value.

In [67], the authors proposed a multi-agent based clustering protocol (MDDC). Stability metric was introduced to determine the CH. The elected CH should have higher connectivity degree (number of neighbors), less average speed, and longer travel time on the lane. The leading vehicle computes stability factor for all the cluster members. The stability metric was represented by a weighted sum of these parameters. The CM with the highest stability metric is chosen as the CH. The weighting factors are determined by the initiator vehicle. However, the selection of weighting factors has not been described in detail.

The authors proposed an Affinity Propagation (AP) [68] based clustering algorithm (APROVE) for VANETs in [41]. Every vehicle calculates the responsibility and availability value in a distributed manner. The proposed method is based on position, velocity, and the prediction of vehicles near future position. For each clustering interval, vehicle calculates its CH convergence value, represented as the sum of the responsibility array and availability array. The vehicle with a positive convergence value will become a CH.

Discussion: Weighted sum based CH selection method was widely applied during the development of clustering algorithms in VANETs. The selected elements vary under dif-

ferent context with the assigned weighting factors. The researchers could introduce any metrics if they have an adequate reason. For example, the trust value proposed in VW-CA [63]. The approach guaranteed a comprehensive consideration of various parameters that may influence the clustering performance; however, the weighting factors assignment is still an open issue, which may considerably affect the clustering performance. Only a small part of the research works provided a reasonable approach in weighting factor selection.

2.4.1.4 Aggregate relative mobility

The aggregate relative mobility based CH selection approach was firstly proposed for MANETs in [36], named MOBIC. The authors proposed a relative mobility metric to select the CH. The ratio of the received and transmitted power between two successive packet transmissions from a neighboring node can represent the relative mobility between two nodes, indicated as a log value. Each node aggregates the variance of relative mobility value of all its neighbors, and the one that has the lowest aggregate relative mobility value with respect to its neighbors should become CH.

The idea of the aggregate relative mobility in MOBIC [36] is impressive; however, calculating two nodes' relative mobility through the Received Signal Strength (RSS) is considered unreliable. In [42], the researchers proposed a proposed an Aggregate Local Mobility (ALM) clustering algorithm to prolong the cluster lifetime for VANETs. Similar to MOBIC [36], ALM was also based on the aggregate relative mobility; however, it replaced the Received Signal Strength (RSS) by the relative distance between two nodes.

In [26], the authors proposed another aggregated relative mobility based clustering algorithm for VANETs, named K-hop. Different from ALM [42], the authors introduced the ratio of packet deliver delay of two consecutive packets to calculate vehicle's N-hop relative mobility, instead of the ratio of relative distance. Moreover, the previous one-hop cluster structure was extended to k-hop cluster structure. The aggregate relative mobility value is the sum of the relative mobility times a weight value for all neighbor nodes in N-hop. The vehicle node which has the smallest aggregate relative mobility is selected as the CH.

Discussion: Comparing with previous CH selection approaches, the aggregate relative mobility based approach is dynamic. The relative mobility between two nodes depends on the information of both the last and the current time interval. Theoretically speaking, this method is more precise with vehicles' historical data; however, the appropriation of

selecting the time interval is very important to ensure the accuracy of vehicles' relative mobilities.

2.4.1.5 Dominating set based

Dominating Set (DS) is a typical technique in the clustering algorithm. A survey about clustering for MANETs [34] has summarized DS-based clustering. The idea of finding Connected Dominating Set (CDS) for MANETs comes from the fact that any vehicle can communicate with another vehicle in the same CDS. The researchers adjusted the DS-based clustering algorithms to VANETs, and the dominating set nodes are considered as CHs, ensuring the communication among CHs.

In [45], a position based clustering algorithm (PPC) was proposed to form stable backbone in a highly dynamic vehicular environment. The authors applied the idea of Minimum Dominating Set (MDS). The proposed cluster is a 2-hop cluster, and the cluster size is controlled by the predefined maximum cluster radius. If a vehicle has long travel time and small speed deviation on the road segment, it will have higher priority to be the CH. To ensure that only one vehicle is selected in the cluster, a hash function has been introduced.

A Hierarchical Clustering Algorithm (HCA) algorithm was proposed in [69], which also selected the CHs based upon the concept of DS. The authors formulated the problem of creating 2-hop clusters and scheduling them as distributed G^2 dominating set problem. HCA aimed to find a randomized approach which can benefit both from a small dominating set and fast convergence. A CH is a member of Dominating Set G^2 , and it manages and synchronizes the shared channel access for all other nodes in the formed cluster.

In [70], a routing protocol for VANETs (SCRIP) was proposed. The objective is to choose routing paths with minimum end-to-end delay (E2ED) for non-safety applications in urban VANETs. The backbone vehicles are selected as the CDS. These nodes are selected according to vehicles' Stability Factor, considering the average distance and the speed relationship between a vehicle and its neighbors. The Stability Factor is also represented as a weighted sum value. The vehicle with lowest Stability Factor is added to the backbone.

Discussion: Formulating the clustering algorithm to a selection of dominating set is not a new solution both in MANETs and VANETs. The CHs are the vehicles in the dominating set to ensure the connection among CHs; meanwhile, to reduce the redundant clusters.

2.4.1.6 Other cluster head selection approaches

Apart from the mentioned CH selection approaches, there exist some other interesting clustering algorithm. Ant Colony System (ACS), initially proposed in [71], has been extended to solve the clustering problem. In the year 2012 and 2015, researchers have proposed two ACS-based clustering algorithm respectively: TACR [72] and ASVANET [73]. TACR used ant colony routing technique based on the trust, and the CH selection was based on vehicle's direction, position, and relative speed. Different in ASVANET algorithm, the CH selection happens after the construction. The node closest to the center of the cluster is selected to be CH. However, the performance of ASVANET did not show any improvement compared with the Lowest-ID [32] algorithm.

Similar to ASVANET, another center based CH selection method was proposed in [19]. The authors introduced the concept of temporary CH, which is the first vehicle moving forward, to assist the cluster construction. The temporary CH will deliver the CH role to the vehicle which is the closest to the center of the cluster according to the predefined cluster's radius. However, this method requires a long time for the cluster construction, and the cluster information may be incorrect.

Discussion: The CH selection process is the most crucial part of the clustering algorithm. Instead of using a single metric at the beginning, such as the node identification in Lowest-ID [32], more and more CH selection methods are essentially depending on multiple mobility metrics, such as relative speed, relative distance, number of neighbors, and link duration between two vehicles. The difference is the approach of combining these primary metrics. Besides, the strategy of CH selection also depends on the context. For instance, at the intersection scenario, like C-DRIVE [50], it is more efficient to choose CHs according to their moving directions.

2.4.2 Cluster construction approaches

This section discuss the cluster construction from the aspect of the cluster topology. After the CH selection process, clusters are formed according to multiple predefined criteria, including transmission range, cluster radius, the maximum number of member vehicles, etc.. Although the cluster formation criteria vary in each algorithm, there are only two cluster topologies of the formed cluster: single-hop cluster and multi-hop cluster.

2.4.2.1 Single-hop cluster

From the analysis, it is easy to observe that the majority of clustering algorithms establish single-hop clusters. Cluster formation depends on the information of vehicle's one-hop vicinities. For example, in PPC [45], cluster volume is limited by the cluster radius, which is determined by the average velocity of vehicles, the number of nodes in a cluster, and the number of lanes on the road. In MC-DRIVE [50], a predetermined distance threshold was designed to limit the cluster construction. The distance threshold is calculated with speed and transmission range of the vehicle approaching the intersection.

In VWCA [63], Adaptive Allocation of Transmission Range (AART) algorithm was proposed. Different from other existing clustering methods in which transmission range is constant, the AART algorithm allows vehicles to adjust the transmission range dynamically based on current vehicle density. The adaptive transmission range (100-1000m) is based on the intra-cluster communication standard, Dedicated Short-range Communication (DSRC) standard [9]. The proposed approach not only reduces the number of clusters, especially the single node clusters but also uses bandwidth resource efficiently. AART algorithm has also been applied in another clustering method proposed in [74], a Data Envelope Analysis (DEA) based clustering algorithm. In DEA, vehicle's transmission range can dynamically change based on vehicle density, between 100m to 1000m.

In [19], the authors proposed a Mobility-based scheme for dynamic clustering in VANETs (MoDyC). The cluster formation depends on CH's transmission range. Hence, the established clusters are single-hop clusters.

Discussion: Single-hop cluster topology can reduce the cluster formation time and decrease cluster management overhead since fewer information exchanges are required. Many existing clustering algorithms construct single-hop clusters directly according to the CH's transmission range or the limited cluster radius. Therefore, the number of members in a cluster will be determined by local traffic density. When vehicle density is very high, collisions could happen in the cluster and would cause low packet delivery ratio. When vehicle density is very low, a vehicle may not find any neighbors to form a cluster and stays single. Both of these two situations will cause worse cluster performance and should be avoided. The proposed AART algorithm can adjust the transmission range to local vehicle density. Moreover, limiting the maximum and the minimum number of vehicles in a cluster can also solve this problem.

2.4.2.2 Multi-hop cluster

In recent years, it is easy to observe that the multi-hop cluster topology is becoming the trend in the cluster design. Researchers believe that the structure of multi-hop clusters are more stable with fewer vehicle disconnections and re-affiliation. Moreover, some applications require information transmission in large scale, and multi-hop cluster topology can increase the information transmission efficiency.

The first multi-hop clustering scheme designed for VANETs was K-hop [26]. Similar to MOBIC [36] and ALM [42], the main idea of cluster construction is based upon aggregate relative mobility. The authors introduced an N-hop relative mobility between two nodes. The cluster size is limited by the number of hops between the CH and its farthest member vehicle.

The proposed HCA algorithm [27] is also a multi-hop clustering algorithm. It aims to establish more stable communications among vehicles for time-sensitive messages delivery. The maximum distance between CH and CM is two hops. HCA is proposed for fast topology control, which is suitable for VANETs. Meanwhile, it also introduced a channel access method using synchronization. Different from other clustering algorithms, HCA does not rely on locations services. The proposed algorithm was compared with K-ConID [37]. Although HCA showed fewer cluster switches than K-ConID, the message delivery delay was not evaluated, which was inconsistent with its objective.

CCA algorithm [75] is a multi-hop clustering algorithm which is derived from the concept of network criticality. It claimed as the first work to apply a localized robust graph metric (network criticality) to guide the cluster formation process. Cluster formation relies on local N-hop neighbors' information and link expiration time (LET). The authors evaluated the performance of CCA by setting the number of hop to 1 and two respectively. The simulation results showed a better cluster stability when the number of hops was 2.

The previous mentioned CONVOY [56] is also a multi-hop cluster. The CH controls the cluster size by the predetermined cluster length. The authors have evaluated the effect of cluster length on cluster performance and defined the cluster length as 2km.

In [76], the authors designed a multi-hop clustering scheme (VMaSC), which was simulated under realistic traffic scenario, generated by Simulation of Urban MObility (SUMO) [77]. The scheme aims to provide more stable clusters and to reduce the number of CH in the network. The CH election relies on the calculated relative mobility concerning its neighbors. The cluster size is controlled by the predetermined hops. Compared with previous multi-hop clustering algorithms, VMaSC provided a more complete and impressive

performance evaluation. It was compared with K-hop [26] for 1-hop, 2-hop, and 3-hop respectively, and showed a better performance regarding CH duration, CM duration, and CH change number.

Another multi-hop clustering algorithm DMCNF was proposed in [78]. The authors designed a neighborhood follow strategy. Each vehicle chooses a one-hop stable neighbor to follow, and the node with more followers and smaller average relative mobility are passively selected to be CH. Similar to HCA [27], each vehicle only needs to update its one-hop neighbors' information, which can reduce the overhead. The authors compared its performance to K-hop (K=3) [26], and DMCNF showed lower cluster overhead. DMCNF did not predefine the number of hops of the cluster; however, the maximum number of hop is four according to the simulation results.

Discussion: According to the research, multi-hop clusters show higher cluster stability, especially regarding the number of CH change, CM re-affiliation, and cluster lifetime. However, multi-hop cluster formation and cluster maintenance are more complicated, which will cost significant control overheads and long cluster formation time. During the research, it can be observed that some algorithms have predetermined the cluster size through the maximum number of hops from CH to CM and the cluster diameter. Simulation results of VMaSC [76] showed that cluster performance became worse when the maximum number of hops is bigger than three. In DMCNF [78], the maximum number of hops was tested to be four and most vehicles were presented in 1-hop and 2-hop clusters without any cluster size limitation. Moreover, according to the research, none of the clustering algorithms have evaluated the cluster construction delay. A long cluster formation time may cause unexpected information transmission delay.

2.4.3 Cluster maintenance approaches

Due to the dynamic topology of VANETs, frequent vehicle disconnections and re-connections may cause severe packet loss. Cluster maintenance is indispensable to reduce frequent vehicle re-clusterings and finally achieve more stable clustering performance. Clustering maintenance includes vehicle leaving, vehicle joining, and cluster merging. The maintenance methods will be presented in this section.

2.4.3.1 Vehicle leaving and joining process

During clustering process, each vehicle periodically broadcasts Beacon messages with their necessary traffic information to inform their vicinities of updating information. Typically,

the CH contains a list of all information of its member vehicles. When it loses the connection with a member vehicle, it deletes this member's information from the list. The disconnected car starts to find a new cluster to join. On the other hand, when a CH receives a request to join information from a vehicle, it checks whether this car could be its member or not, and sends back a confirmation message or just ignores the request.

2.4.3.2 Cluster merging process

Compared with the vehicle leaving and joining process, cluster merging process is more complicated. Cluster merging happens when two or more clusters can be represented by one merged cluster, which can reduce the number of clusters and increase the clustering efficiency. According to the literature, Cluster merging conditions are different in clustering algorithms. Usually, cluster merging is triggered when two CHs approach one another and become one-hop vicinities. In order to guarantee the stability of the merged cluster and to decrease vehicle re-clustering frequency, most existing cluster merging schemes require two neighboring CHs to stay in the transmission range (TR) of each other for a short time period, which is defined as contention time or merge interval (MI), instead of starting cluster merging immediately.

Usually, there are two common strategies to select the new CH in the merging cluster. The first one is to select the CH that is attached with more CMs, denoted as "CM-based", as adopted by [79], [45], and [58]. In [79], cluster merging takes place when two CHs come within each other's transmission range, and their speed difference is within the predefined threshold Δv_{th} . The CH that has a lower number of CMs simply gives up the CH role and becomes a CM in the new cluster. The rest CMs automatically join the neighboring cluster if they are in the transmission range of the CH and the speed difference is within the threshold. Similar to [79], in [45], when cluster merging happens, the cluster with fewer CMs is dismissed, and these CMs try to join other clusters, launching a new CM re-clustering stage. The "CM-based" strategy aims to reduce cluster member disconnections. However, such a strategy cannot guarantee the stability of link connections between the new CH and its members.

The second strategy is to select the CH that has better stability within its original cluster, as adopted by [7], denoted as "VMaSC-based". During cluster merging, two CHs compare their averaged relative speed, called AVGREL_SPEED in their original clusters, respectively. The CH with higher average relative speed gives up its CH role and affiliates to the CH with lower average relative speed as a CM. Similar to [7], both the cluster

merging schemes proposed in [42] and [80] select the new CH according to predefined vehicles' stability metrics (Aggregated Local Mobility (ALM) in [42] and Benefit Factor (BF) in [80]) in the original clusters. The intuition of this strategy is assuming that the CH's stability in its original cluster is representative for its stability in the newly merged cluster. However, in reality, a higher stability in the original cluster cannot guarantee a better cluster performance in the merged cluster.

Discussion: Due to the dynamic mobility pattern in VANETs, cluster merging process is indispensable and happens frequently. A better clustering method should smoothly manage cluster merging process, at the meantime, avoid unnecessary cluster re-affiliations. Most algorithms applied contention timer to defer cluster merging, ensuring that only the relative stable clusters can be merged. The merged CH is selected from the two merging CHs, which can guarantee most of the link connections with CMs. However, none of the previous clustering algorithms have analyzed the impacts of cluster merging process on clustering performance. In [22], the authors summarized the existing cluster merging methods and proposed a leadership-based clustering merging scheme (LCM). Moreover, a fair comparison of different merging schemes was given.

2.5 Classification of clustering algorithms

In this section, we propose a new classification of the existing clustering algorithms based on different context. During our research, it is easy to observe that clustering methods and metrics are becoming more and more diverse and complex. Therefore, classifying clustering algorithms based on their context is more reasonable and meaningful in research. Apart from pure clustering algorithms which are infrastructure-independent, hybrid clustering algorithms are becoming important with the rapid development of cellular network technologies. For this reason, we provide a comprehensive observation of the existing hybrid clustering algorithms in the second part of this section.

2.5.1 Clustering for Context-based Applications

In recent years, clustering mechanisms are applied for specific VANETs applications. Cluster nodes are treated as backbone nodes for information dissemination in VANETs. In this case, defining a proper clustering mechanism should be based on a specific context. Numerous context-based clustering algorithms have been proposed in recent years. Table 2.3 classifies the existing clustering algorithms according to their context.

2.5.1.1 Cluster-based information dissemination

CB-BDP [81] proposed a cluster-based beacon dissemination process, aiming at providing vehicles with a local vehicle proximity map of their vicinities which could be used for safety applications. The idea of CB-BDP is to combine cluster-based MAC with inter-cluster coloring scheme used to synchronize channel access between adjacent clusters. In the first phase, beacons in the same cluster are aggregated by CHs using the intra-cluster aggregation protocol. In the second phase, adjacent CHs exchange their cluster status using the multi-hop inter-cluster communication protocol. In the final phase, CHs broadcast the aggregated information to all their CMs through the intra-cluster dissemination protocol, providing each CM with a local vehicle proximity map. The authors used cluster coloring scheme to synchronize channel access between adjacent clusters and grouped sets of connected clusters into super-clusters where each super-cluster is colored independently. The CHs in the same super-cluster can always communicate with each other.

MCTC [82] is proposed for relay selection. The CH is selected as the vehicle that has the highest link connection value among its neighbors. The information exchanging is in a carry and forward fashion, taking into consideration of vehicles moving in different directions. The cluster relay (CR) is selected by CH. The CM with the lowest relative speed is chosen as CR.

In 2014, the author proposed a multi-homing based clustering method [83] for urban city scenario. It relies only on the vehicle's ability to send and receive wireless packets which identify the vehicle relationship. Clusters are created with redundant connections between nodes to increase the communication reliability. This algorithm is not location service dependent, like HCA [27]. The author proposed a new clustering metric which considered redundant CH connections, enabling the support for multi-homing. The value of the metric is represented as an 8-bit unsigned non-overflowing integer counter with the initial value of zero. A higher value means higher vehicle interconnections.

In [84], the authors proposed a pair of algorithms, sociological pattern clustering (SPC) and route stability clustering (RSC). It considers vehicles' social behaviors during the clustering process. The historic trajectories of vehicles can be stored in RSU. The CH selection method is based on virtual force, which is defined as Coulomb's law, according to their distance and their relative velocities. Every node computes the accumulated relative force applied to it along the axes x and y , and the total node and the relation of its force to its neighbor's. The node with the highest force among its neighborhood is considered as the most stable node and can become a CH. The social pattern includes the

type of vehicles, the height of the vehicle, the vehicle that tends to stay on the main street longer, and vehicles with driver behavior that is statistically smooth. The relative mobility parameters are calculated based on vehicle's current position and its future position. This protocol allows vehicles that are moving in the opposite directions to be clustered.

CF-IVC [53] proposed a cluster-based inter-vehicle communication method which firstly classifies vehicles into different speed groups. Vehicles will only join a CH of similar velocity. Code Division Access scheme is used to assign orthogonal codes to the previously identified vehicular nodes assuring collision-free data exchange among the nodes in the intra-cluster or inter-cluster communication. Vehicles are classified into speed groups based on the speed information of its GPS. In CF-IVC, the vehicle doesn't require the knowledge of absolute location of its neighbors to form a cluster or to communicate.

2.5.1.2 Routing

A Robust Mobility-Aware Clustering (RMAC) routing approach was proposed in [85]. In RMAC, the CH is selected based on the mobility metrics, including location, speed, and direction. After the vehicle identifies its 1-hop neighbors, the neighbors are sorted using the Bubble sort algorithm. The clusters are overlapped and a CH can also be a CM of another cluster. The network formed by RMAC is used to disseminate neighbor information using predominantly unicast packets synchronized by the CHs, such that nodes can construct neighbor tables in order to support geographic routing, by providing accurate location information of neighboring vehicles.

A cluster-based location routing (CBLR) [86] protocol was proposed for V2V communication, based on non-positional routing, by employing location information provided by GPS. In CBLR, only the gateway nodes can retransmit the packets.

In [83], the authors proposed a multihoming clustering algorithm (MCA-VANET), a new clustering algorithm with redundant cluster head connections, enabling the support for multihoming. Different from other clustering algorithms, all vehicles are claimed as CH in the initial state. The nodes can change state to a member vehicle if it has enough CH connected. Therefore, each vehicle has at two least one CH to be connected.

2.5.1.3 Traffic density estimation

The authors proposed a clustering algorithm for traffic density estimation at the intersection in [50]. Different from other clustering methods, MC-DRIVE introduced the imaginary points to control the cluster formation and CH election process. Clusters are formed

based on the moving directions at the intersection area.

In [80], the authors proposed a hybrid approach for traffic density estimation on the road, with the assist of RSUs. Based upon the estimated traffic density, the authors also introduced a new technique for traffic information generalization in low penetration terms of equipped vehicles. The approach showed higher accuracy on traffic density estimation.

2.5.1.4 Traffic safety

SRB [87] proposed a cluster-based broadcast protocol for safety applications in VANETs. It aims to reduce the effect of broadcast storm problem by limiting the number of packet transmissions. By receiving the messages from neighbor nodes, source node firstly detects the cluster including neighbors within the distance threshold. CH is selected as the farthest vehicle within the distance threshold.

2.5.1.5 Traffic prediction

In [88], the authors proposed an online traffic prediction method based on traffic clustering. The authors tested the real traffic data and designed an on-line neural network based traffic prediction algorithm. The network was modeled as a directed time-variant node-weighted graph. An affinity Propagation based traffic clustering is proposed. Each node pair calculates their similarity value. According to the tests, the authors concluded that the Average Relative Error (ARE) decreased when the maximum hop H increased from 1 to 4. When H was bigger than 4, ARE increased. Therefore, H was fixed as 4 during the simulation.

2.5.1.6 Data aggregation

In [104], the authors proposed a method, called CASCADE, for accurate aggregation of highway traffic information. CASCADE uses data compression to provide aggregation without losing accuracy. It aims to provide the information about the upcoming vehicles. With data compression, information can be transmitted more efficiently.

In [105], a secure cluster-based in-network information aggregation (SCB-INIA) algorithm was proposed. In SCB-INIA, the authors presented a new security mechanism for traffic efficiency application that uses HyperLogLog estimators to create bandwidth-efficient integrity proofs. SCB-INIA is claimed to be able to achieve high protection against plausible attacker models, and that it is more bandwidth efficient than a comparably secure security mechanism that does not employ clustering. However, the authors

Table 2.3: Classification of clustering algorithms*

Context		Algorithm	Year	Cluster performance		Network performance
				Macroscopic	Microscopic	
Application	Pure clustering	DMCNF[78]	2015	1, 2, 3	6	11, 16
		E-SP-CL[89]	2013	1, 3	7	
		UOFC[90]	2013	1, 3	6	
		VMaSC-1[76]	2013	1, 2,	6	16
		CCA[75]	2012	1, 2, 3, 4	6, 7	
		SP-CL[91]	2012	1, 3	7	
		TBC-2[79]	2012	1, 3	7	16
		FLBA[66]	2012	1, 2, 4		
		K-hop[26]	2011	1, 2	6	
		ALM[42]	2010	1, 3	7	
		DBC[60]	2009	2, 3, 4	6	
		MDMAC[92]	2008	3	6	16
		PPC[45]	2008			15
		AMACAD-2[65]	2011	1, 2, 3	10	
		APROVE-2[44]	2009	1, 2, 3	6	16
	UF[59]	2005		6		
	HCA[27]	2011	1, 3	7		
	Information dissemination	CB-BDP[81]	2015			12
		CFT[93]	2017	2, 4		15
		SFC[84]	2016	1, 3	7	
		MCTC[82]	2014	1, 3, 4	9	
		MCA-VANET[83]	2014		6	
		TC-MAC-3[94]	2013			
		MCMF[95]	2013			14, 15
		CSBP[96]	2007			14
		C-DRIVE[49]	2009			13
		LTE4V2X-3[97]	2012			12, 13
	Routing protocol	CONVOY[56]	2013	1, 3	7, 9	
		PassCAR[54]	2013	3		12, 15
		TACR[72]	2012			12, 16
		CAC[98]	2011			12, 13, 15, 16
		MI-VANET[99]	2010			12, 15
		VPC[100]	2010			12, 14, 15
		RMAC[85]	2009	1		10
		CBLR[86]	2004			12, 14, 16
	PC[38]	2003			14, 15	
	Traffic density estimation	MC-DRIVE[50]	2011	3		16
		CB-TIG[80]	2014	1, 3,		
	Security	ALCA[101]	2013	1, 2, 5		15
		VWCA[63]	2011	1, 2		12
		CBPKI[102]	2011	1, 3, 4, 5		
	Traffic safety	SRB[87]	2012			12, 14, 15
		C-RACCA[43]	2010			13, 14
	QoS	SBCA[103]	2012	1		12, 16
		CCP[48][47]	2006, 2007	1, 4		13, 14, 15, 16
Aggregation	CASCADE[104]	2015			12, 14, 15	
	SCB-INIA[105]	2015	4		16	
Target tracking	PBC-TT[106][107]	2014, 2017	1, 2,	6	12, 14, 16	
Topology discovery	CPTD[108]	2012	3		11, 16	
Traffic prediction	TC-OTP[88]	2012	3			
Hybrid clustering	Floating Car Data (FCD)	FCDOC[40]	2016			14
		GC-VDB[39]	2013	1, 3, 4		
		LTE4V2X[109][6]	2011			13, 16
	Gateway selection	FQGwS[110]	2015			13, 14, 15, 16
		CMGM[111]	2011			12, 13, 14, 15, 16
	Data transmission	VMaSC-LTE[7]	2015	1, 2, 3	6	12, 14, 16
		LTE4V2X[97]	2012			13, 16
	Collision avoidance	CA-ICA[112]	2013			12, 14
Cluster size optimization	DCSO[113]	2016	4		12	
Uplink transmission	C-HetVNETs[114]	2015			13, 14	
MAC protocol		CBMAC[115][46]	2007	1		12
		MCC-MAC[116]	2014			13
		DMMAC[117]	2013	1, 2, 4		14, 16
		TC-MAC-2[118]	2012			13
		DBA-MAC[119]	2007			13, 14
	CCP[48][47]	2006, 2007	1, 4		13, 14, 15, 16	

The numbers in this table indicate the corresponding performance metrics in Table 2.5.

only compared the overhead performance with another mechanism, which is considered inconceivable.

2.5.1.7 MAC clustering

In [115] and [46], the authors proposed a cluster-based medium access scheme for VANETs (CBMAC), in order to minimize the effect of hidden stations and further leads to a reliable data transmission. In CBMAC, CH takes over the responsibility to assign the bandwidth to its member vehicles in the cluster. CH selection method is a weighted-based method, where the node with the lowest weighted sum among its neighbors will be chosen as a CH. Three metrics are considered in the calculation of the weighted sum: connectivity, relative velocity, and relative distance. The basic structure of a Time Division Medium Access (TDMA) frame includes three parts: the first part is for the CH to broadcast basic information; the second part is for the CH to assign slots for its CMs, and the last random access part is for the CMs to transmit data.

DMMAC [117] proposed a distributed multichannel and mobility-aware cluster-based MAC(DMMAC) protocol. Vehicles organize themselves into more stable and non-overlapped clusters through channel scheduling and an adaptive learning mechanism integrated within the fuzzy-logic inference system (FIS). Every vehicle broadcasts its status message with a weighted stabilization factor. The vehicle with the highest weighted stabilization factor among its neighbors will elect itself as CH. The calculation of weighted stabilization factor is based on the vehicles average relative speed and the previous weighted stabilization factor. The triangular fuzzier is chosen to implement the FIS system, with the inter-distance and the relative velocity between two vehicles as the input parameters, and the vehicle's acceleration as an output value. In DMMAC protocol, each cluster uses a different sub-channel from its neighbors in a distributed manner to eliminate the hidden terminal problem. The proposed protocol can increase the system's reliability, reduce the time delay for vehicular safety applications, and efficiently cluster vehicles in highly dynamic and dense networks in a distributed manner.

2.5.2 Hybrid clustering

Most existing clustering algorithms are based on the creation, in a decentralized way, of dynamic clusters to self-organize a non-heterogeneous IEEE 802.11p vehicular network. With a highly dynamic environment such as vehicular networks, a decentralized clustering is not appropriate since it creates a large amount of overhead within the network. In

recent years, researchers start to focus on V2X communication type with the assist of cellular infrastructure instead of pure V2V communications. Table 2.4 summarizes the existing hybrid cluster architectures. It can be observed that most of the research works combine IEEE 802.11p with LTE cellular architecture, where IEEE 802.11p interface is used for V2V communication and LTE interface is used for V2I communication. The CH is selected by the base station. Application information is transmitted from the base station to CHs, and CHs broadcast the information to their CMs. From another direction, CHs are responsible for collecting and aggregating the data from their CMs. Then, CHs deliver the aggregated information to the base station. The hybrid clustering approaches summarized in Table 2.4 serve for different applications, including traffic data collection, information dissemination, gateway selection, and accident avoidance.

Table 2.4: Hybrid cluster architecture

Algorithm	Year	Application	Radius	V2V link	V2I link	Network Simulator	Traffic Simulator	Traffic Scenario
FCDOC [40]	2016	Floating Car Data application off-loading	1-hop	802.11p	LTE	OMNET++, Veins	SUMO, OpenStreetMap	City map of Rome and New York
DCSO [113]	2016	Cluster size optimization to reduce packet loss	N-hop	802.11p	LTE	OMNET++, Veins	SUMO	Highway
VMaSC-LTE [7]	2015	Safety message dissemination	N-hop	802.11p	LTE	NS3	SUMO	Straight road
FQGwS [110]	2015	Gateway selection	1-hop	802.11p	LTE-A	NS2	VanetMobiSim	Multiple-lane highway
C-HetVNETs [114]	2015	A framework for performance analysis	1-hop	802.11p	LTE	N/A	N/A	Urban with intersections
GC-VDB [39]	2013	Data collection	1-hop	802.11p	LTE	OPNET	OpenStreetMap SUMO	Highway & urban
CA-ICA [112]	2013	Intersection collision avoidance	1-hop	802.11b	LTE	NS3	VanetMobiSim	Urban with intersections
CMGM [120][111]	2011	Gateway selection	1-hop	802.11p	UMTS	NS2	N/A	Highway & urban
LTE4V2X [97]	2012	Data collection & data dissemination	N-hop	802.11p	LTE	NS3	VanetMobiSim	Highway
LTE4V2X [109][6]	2011	Floating Car Data (FCD)	1-hop	802.11p	LTE	NS3	VanetMobiSim	Urban

In [109], the authors proposed a framework for a centralized heterogeneous vehicular network using LTE, called LTE4V2X. All vehicles are assumed to have both the LTE and the IEEE 802.11p interfaces. The clustering process is managed by the central eNodeB-s. The framework is designed for Floating Car Data (FCD) application and is claimed to be able to deploy other applications. In [6], the authors analyzed the cluster performance of the proposed LTE4V2X [109] especially under highway scenario. In the last version of LTE4V2X in [97], the authors extended the proposed LTE4V2X framework to a multi-hop communication. Meanwhile, the adaptation of LTE4V2X framework for a data dissemination application has been introduced.

In [39], the authors proposed Greedy-based Clustering with Velocity and Direction re-

striction and cluster Bonus (GC-VDB) scheme, aiming for the extended Floating Car Data (xFCD) collection. In GC-VDB scheme, the CH vehicle should fulfill the following conditions: 1. Vehicles with the highest number of reachable nodes via V2V communication; 2. Vehicles satisfy the velocity and angle requirements (the velocity and moving direction should be limited); 3. Vehicles with the highest CH score (CS). The CH can perform data compression called xFCD payload, which increases the information efficiency. Both urban and highway scenarios were tested. However, it was not compared with other algorithms. Moreover, the authors proposed a new simulation architecture which allows for modeling of LTE cells under realistic user mobility. The simulation results have shown that the GC-VDB algorithm enables an enhancement of the cluster lifetime and a decrement of the total xFCD payload. Hence, the LTE utilization has decreased significantly.

In FCDOC [40], the authors employed a VANET-based multi-hop dissemination logic to spread control messages and elect designated nodes. The designated nodes (CH) are used to report vehicular data via LTE communications. The representative nodes are responsible for communicating aggregated FCD via the LTE infrastructure. The mechanism is simulated by a multi-layer simulation tool, constructed by SUMO, OMNET++, and Veins. This paper considers the real urban maps of the city centers of Rome and New York.

VMaSC-LTE [7] proposed a cluster-based architecture for VANET safety message dissemination. It is based on the work of VMaSC [76], a multi-hop clustering mechanism, proposed in 2013. The objective of this paper is to reduce the information transmission delay based on a hybrid architecture IEEE 802.11p and LTE. The authors designed a multi-hop clustering algorithm for this hybrid architecture. Compared with the previous pure VMaSC algorithm, the hybrid architecture has shown higher packet delivery ratio and lower packet transmission delay. However, the overhead has not been analyzed.

In [112], a cluster-based architecture for intersection collision avoidance (CA-ICA) is proposed based on heterogeneous networks. Vehicles approaching the intersection start to broadcast CAMs; however, packet collisions may happen when the node density is very high. Hence, instead of broadcasting the CAMs directly, the vehicle equipped with LTE interface will transmit CAMs to a base station and then be forwarded to vehicles on other roads. Since the CAMs should be broadcasted every 100ms, the clustering algorithm is proposed to reduce the number of data transmission. The vehicle approaching the intersection acts as the CH. It aggregates the cluster members' information in a single CAM and sends it to the corresponding base station via LTE interface. Then, the base station

will transmit CAM to the CH on the other roads, and these CHs broadcast this information directly to their members through an IEEE 802.11p interface. This heterogeneous network combines IEEE 802.11p and LTE, and it performs a higher packet delivery rate.

In [114], the authors proposed a framework of Cluster-Based Heterogeneous Vehicular Networks (C-HetVNETs). The performance analysis models of intra-cluster and inter-cluster communications were proposed based on Markov queuing model. Floating Car Data (FCD) applications are carried in the framework based on the designed model. In C-HetVNETs, two channel access interfaces are implemented, IEEE 802.11p for intra-cluster communication, and LTE for inter-cluster communication. The evolved NodeB (eNB) selects the vehicle that is closest to the center of the cluster to be CH. Clusters are formed only via one-hop of CH. Similar to the previous GC-VDB approach, CH will aggregate the data packets received from its CMs and can directly communicate with eNB. The analytical models are impressive; however, the authors did not mention which kind of simulators they implemented on.

In DCSO [113], the authors proposed a new heterogeneous clustering algorithm for dynamic cluster size optimization. The authors analyzed the impact of the maximum number of hops between a specific vehicle and its CH on the average cluster size, the data aggregation performance and the packet loss in the IEEE 802.11p network. Similar to AATR approach [63], the maximum number of hops should be adjusted to vehicle densities. Then, the authors proposed a new heterogeneous clustering algorithm, delegating the CH selection to the cellular base station. The simulation results showed that DCSO resulted in larger clusters for the same maximum number of hops compared to VMaSC-LTE algorithm [7]. As a consequence, data compression at the CH is more efficient in DCSO.

Discussion: With the rapid development of cellular network technologies, hybrid clustering via both DSRC and cellular technologies is becoming a trend in supporting more vehicular networks applications. Information can be delivered by the base station in a large area with lower latency, instead of a pure multi-hop delivering method. However, the requirement of having both the IEEE 802.11p and cellular access interfaces on the vehicle is still a challenge.

2.6 Performance Evaluation

Performance evaluations of clustering algorithms in VANETs are mostly based on network simulators because of the limitation of testing scales in real traffic environments. Moreover, using network simulators enables the same simulation environment when comparing

Table 2.5: Existing performance metrics

Domain		ID	Performance metric	Description
Cluster Performance	Macroscopic performance	1	CH/cluster lifetime	Avrg. time duration from a vehicle becoming a CH to giving up its state
		2	CM lifetime	Avrg. time duration from a vehicle becoming a CM to giving up its state
		3	No. of clusters/CH	Avrg. no. of clusters being formed during the simulation period
		4	Cluster size	Avrg. no. of vehicles in a single cluster
		5	Cluster efficiency	The percentage of vehicles participating in clustering process
	Microscopic performance	6	CH change rate	Avrg. no. of CH changes per unit time
		7	Cluster change rate	Avrg. no. of cluster changes per vehicle in a unit time
		8	State change rate	Avrg. no. of state changes per vehicle in a unit time
		9	Disconnection ratio	Avrg. percentage of disconnected vehicles
		10	CM reconnection ratio	Avrg. Percentage of vehicles that re-cluster within a given time
		11	No. vehicles per hop	Avrg. no. of vehicles per hop distance
Network Performance	12	Delivery ratio, success ratio	The percentage of vehicles that successfully receive the packets	
	13	Collision ratio, Packet loss ratio	The percentage of collision packets during packets transmission	
	14	End-to-End delay	Avrg. latency of data packets transmitted from source to the destination	
	15	Throughput	The rate of successful message delivery over a communication channel	
	16	Overhead	The ratio of the total no. of control packets to the total no. of data packets	

various algorithms. However, existing clustering algorithms were evaluated based on different assumptions, including traffic scenarios, channel models, vehicle mobility models, etc.. The criteria of evaluation are different; therefore, it is hard to compare different algorithms in a fair way. The rest of this section analyzes the performance evaluation methods of the existing clustering algorithms.

2.6.1 Performance metrics

According to the observation of existing clustering algorithms, there is a lack of fair comparison among these algorithms. Most clustering schemes aim to increase the **cluster stability**. However, many of them did not explain the term “**cluster stability**” and the corresponding performance metrics. In this section, the most mentioned performance metrics are summarized in Table 2.5, and are classified into two categories: cluster performance and network performance. Each performance metric is given a short description, and is assigned with identification from Arabic numbers from 1 to 16.

2.6.1.1 Cluster performance metrics

Cluster performance metrics describe the performance of clustering algorithms and reflect the stability of backbone nodes in the network. It can be described from the aspects of macroscopic and microscopic cluster performance, as proposed in [19]. The macroscopic cluster performance describes the overall cluster performance, including CH lifetime, CM lifetime, number of clusters, cluster size, and cluster efficiency. In general, a good clustering algorithm prefers a higher CH lifetime, higher CM lifetime, less number of clusters, large cluster size, and higher cluster efficiency. However, only the macroscopic cluster performance metrics cannot describe the details of communication links among vehicles in the network. The microscopic cluster performance metrics, including CH change rate, cluster change rate, state change rate, disconnection ratio, CM reconnection ratio, and the number of vehicles per hop, can describe precisely the **cluster stability**. From this aspect, a stable cluster should have lower CH change rate, lower cluster change rate, lower state change rate, smaller disconnection ratio, and higher CM reconnection ratio. Normally, a stable link connection requires the vehicle to maintain its current state for a longer duration. Therefore, a disconnected vehicle is expected to build a new link connection as soon as possible, in order to avoid unnecessary packet loss.

2.6.1.2 Network performance metrics

Another class of performance metric is network performance, describing the overall network performance, including packet delivery ratio, packet loss rate, collision ratio, end-to-end delay, throughput, and overhead. The network performance metrics are used to evaluate the context-based clustering schemes, such as information dissemination, routing, and traffic prediction. An efficient clustering algorithm inclines to perform higher packet delivery ratio, lower packet loss rate, lower collision rate, short end-to-end delay, larger throughput, and smaller overhead.

Discussion: The performance metrics of each clustering algorithms are summarized in Table 2.3. Clustering algorithms are listed according to their context. The metrics are represented by the ID numbers defined in Table 2.5. It can be summarized that pure clustering algorithms can be evaluated only by the clustering performance since there is no application related information in the network. On the contrary, the application related clustering algorithms should be analyzed from the aspects of both clustering performance and network performance. In summary, the performance evaluation metrics should strictly depend on the context of clustering algorithms, and these metrics have to be clarified before

Table 2.6: Comparison of urban simulation scenarios

Algorithm	Network topology	Traffic simulator; Network simulator	Transmission range (m)	No. of vehicles (density)	Vehicle Velocity (km/h)
SPC [84]	Erlangen, Germany	SUMO; N/A	300	80,120,160 veh; default: 120 veh	20-50, default: 42
C-DRIVE [50]	3*3 km, 7 inters	N/A; NCTUns	350-200	5-40 veh/lane	30.6-50.4
TC-MAC-1 [62]	2 inters, Norfolk,VA	N/A; NS3	150-300	20% trucks, 80% sedans; 60 veh/lane/km; (interval: 10 m)	MAX: 40,80,120
CA-ICA [112]	grid, 25 inters, 2*2 km, 2 dir, 2 lanes	VanetMobiSim; NS3	100	randomly	MAX: 50
C-HetVNETs [114]	8.5 km, 1 eNB	N/A; N/A	N/A	100-400 veh/km (uniformly)	120
CAC [98]	1 * 1 km, between inter: 350 m	N/A; N/A	100-500	300 veh (randomly) (interval: 15 m)	10-100
CBMAC [115][46]	City map Ulm in Germany	N/A; N/A	500	Day Avrg: 5 veh/km, MAX: 20 veh/km; Rush Avrg: 8 veh/km, MAX: 40 veh/km	N/A
MDMAC [92]	Washinton D.C., 1.087*0.942 km (TIGER)	VanetMobiSim; JiST/SWANS	70	500	Limit: 39.6-111.6; Avrg: 28.8
ALCA [101]	2 * 0.2 km (TIGER), with inters	MobiSim; N/A	200	400	30-50 miles/s
K-hop [26]	1 * 1 km	N/A; NS2	120	100	36-126
MI-VANET [99]	southern beijing, with inters, 1.7*1 km	VanetMobiSim; N/A	Bus: 400; Car: 150	100-250, 20% buses	0-108
DBC [60]	Washinton urban area, 2*3 km (TIGER),	VanetMobiSim; JiST/SWANS	N/A	100,200,300, 400,500 veh	Limit: 39.6-111.6; Avrg: 28.8
AMACAD [65]	1.5*1.5 km, blocks of 100 m	N/A; Java developer 10G	100-300	50 veh, 0-5 $veh/100m^2$	18-72
VMaSC-LTE [7]	5 km, 2 lanes, 2 dir	SUMO; NS3	200	100 veh	36-126
UOFC [90]	grid, 2 lanes, 4 lanes	iTETRIS; Octave (fuzzy logic)	250	N/A	MAX: 30,50,70
LTE4V2X [97]	8 km^2	VanetMobiSim	300	100-300 veh	90-145
TC-OTP [88]	Washington D.C.	N/A; N/A	N/A	N/A	N/A
FCDOC [40]	New York, Roma, OpenStreetMap	SUMO; OMNET++	N/A	NY: 70,96,110 veh/km^2 RM: 70,80,87 veh/km^2	50
MCA-VANET [83]	7.960*10.575 km, City Kirchberg	SUMO; NS3, Ovnis	100, 200, 300	395 veh	N/A
MDDC [67]	5 km, 2 lanes/road, 10 roads, 5,10 inters	N/A; C programme	250, 500	10-100 veh (interval:4 m)	MAX: 40,60,80 (MIN: 10)
HCA [27]	1.3*3.2 km	SUMO; OMNET++	200	10-100 veh (4 types)	MAX: 18-144

simulation evaluations.

2.6.2 Simulation scenarios

In the existing clustering algorithms, there is a lack of fair comparison between different algorithms, since various traffic scenarios are created in different algorithms. In this section, a detailed comparison of traffic scenarios is presented, including the comparison of the simulator, network topology, transmission range, vehicle density, and velocity. Table

Table 2.7: Comparison of highway simulation scenarios

Algorithm	Network topology	Traffic simulator; Network simulator	Transmission range (m)	No. of vehicles (density)	Vehicle Velocity (km/h)
E-SP-CL [89]	2 dir, 3 lanes/dir	SUMO; Custom	130	8-15, 5-9, 3-6, 2-5 veh/ km/lane.25% trucks	80-120
SP-CL [91]	2 km, 1 dir, 5 lanes	N/A; N/A	80, 125	8-15,5-9,3-6, 2-5,veh/km/lane	80-160
PPC [45]	10 km, 2/4 lanes	CORSIM; NS2	250	100 veh	60,100
UF [59]	N/A	N/A; Traffic simulation 3.0	150	N/A	40-140
TBC [79]	15 km, 5 lanes/dir	N/A; C++	r: 150-300; R: 800-1000	400 (13-21 veh/km/lane)	Avrg: 70,90,110; dev: 21,27,33
FLBA [66]	8 km, 1 dir, 4 lanes	MOVE, SUMO; NS2	200	50-400 veh/km	80-120
MCTC [82]	10 km, 2/4 lanes	N/A; MATLAB	N/A	250 (25 veh/km, 15 veh/km)	left lane: 100-130; right lane: 80-110
LTE4V2X [97]	3 lanes/dir, 8 km ²	VanetMobiSim; NS3	300	100-300	90-145
VWCA [63]	2 bands, 3 lanes/band, 2.5 * 0.05 km	N/A; MATLAB	static:300; dynamic:100-1000	10-350 veh	70-120
C-RACCA [43]	N/A	N/A; NS2	150	N/A	54-162
FQGwS [110]	multi-lane	VanetMobiSim; NS2, MATLAB	250	15-40	N/A
DMMAC [117]	8 km, 1 dir, 4 lanes	SUMO, MOVE; NS2	r: 300, TR: 2.5*300	0.05-0.4 veh/m	uniformly 80-120
CASCADE [104]	100 km	ASH; SWANS	300	500 veh	MAX: 108
DCSO [113]	5 km, 3 lanes	SUMO; OMNeT++, Veins	N/A	60 veh	Avrg: 60
ALM [42]	1 km, 2 dir, 2 lanes	SUMO; SIDE/SMURPH	N/A	N/A	LMS: 36,54,72, 90,108
VPC [100]	10 km, 2 lanes	N/A; NS2	250	100-400 veh	MAX: 72, 144
NDBC [121]	real map	TIGER; SWANS++	0-250	50-250 veh	N/A
APROVE [44]	3*3 km, 1 dir, 3 lanes, rectangular looped	MOVE, SUMO; NS2	250	100 veh	40,80,120,140
PassCAR [54]	5 km, 3 lanes	MOVE, SUMO; N/A	250	150-350 veh	80,100,120
CB-TIG [80]	3 km, 1 dir, 2 lanes	SUMO; OMNeT++, Veins	100-1000	N/A	60-120
TC-MAC [118] [94]	N/A	N/A; NS3	300	195 (5,12,21,50 veh/lane)	MAX: 104
MCMF [95]	50 km, 2 dir, 3 lanes/dir	N/A; N/A	500	1000 veh/dir	80-200
RMAC [85]	2 km, 1 dir, 4 lanes	freeway mobility generator; NS2	250	25,50,75 veh; 12 scenarios;	79.2-129.6
DBA-MAC [119]	8 km, 1 dir, 3 lanes	N/A; NS2	250	200,400,600 veh	72-108
CCP [48]	3 lanes/dir, circular loop, 2 km	N/A; N/A	N/A	12, 24, 40 veh/km/lane	72-180
PBC-TT [106] [107]	10 km	N/A; NS2, Tossim	50,100,250,500	50,100,150,200 veh	90-126
SCB-INIA [105]	5 km, 3 lanes	N/A; JiST/SWANS	300	0-500	N/A
SBCA [103]	4 lanes,1 dir	N/A; NS2	300	50,100,150 veh	90-126
QuickSilver [122]	3 * 0.2 km	ONE simulator; MATLAB	N/A	60,100 veh	Avrg: 54
CFT [93]	11 km	N/A; N/A	250-600 (per 50)	5-10 veh/km	60-120

2.7 and Table 2.6 summarize the traffic scenarios for highway and urban city respectively.

According to Table 2.7, the road topology is always designed as a straight, multi-lane, 2-direction highway road. In Table 2.7, 31 clustering algorithms designed for highway scenarios have been summarized; however, 15 of them have not specified the traffic simulator, 4 of them have not addressed the network simulator, and 7 of them only used MATLAB or C++ for network simulation, which is considered as unreliable.

In Table 2.6, the simulation parameters of 20 clustering algorithms for urban scenarios are summarized. Among these algorithms, 9 clustering algorithms use real city map, which are obtained from TIGER files or OpenStreetMap (OSM), for performance evaluation. The transmission range varies from 50m to 300m, which is a standard value of IEEE 802.11 Physical layer protocol. According to the theory of traffic flow [123], the transmission range is a function of the local density of vehicles, which is determined by vehicle movement and speed.

In [124], the researchers have introduced various network simulators and traffic simulators for vehicular networks, as well as pointed out the challenges of the simulation of VANETs. According to our summarization, even though NS2 project is no longer active since the year 2010, NS2 has been one of the top choices for researchers after the year 2010. Meanwhile, NS3 and OMNeT++ are becoming widely used because of their low complexity. Furthermore, the researchers have realized the importance of evaluating their algorithms under real traffic scenarios, especially for urban city scenarios.

2.7 Conclusion

In this chapter, we explored the existing clustering algorithms in VANETs and proposed a new classification considering also hybrid clustering architectures. In the beginning, we addressed the importance and indispensability of clustering algorithms in vehicular networks, as well as described the development of clustering algorithms. Then, we reviewed the existing surveys and highlighted their advantages and shortcomings.

According to our observation, we present clustering techniques from the aspects of cluster head selection, cluster construction, and cluster maintenance. We address the advantages and disadvantages of each method and provide a short conclusion. Furthermore, a classification of clustering algorithms is provided according to the different context. Generally, clustering schemes which provide high **cluster stability**, should ensure the following properties: (1) lower transmission overhead; (2) longer cluster head lifetime and longer cluster member lifetime; (3) less average number of state changes per vehicle; (4)

less cluster head changes. Based on our review, mobility metrics, such as vehicle's moving direction, velocity, and inter-vehicle distance, can improve **clustering stability**.

The last part of this chapter summarizes all of the performance evaluation methods and performance metrics in clustering algorithms. We observe that the simulation scenarios and performance metrics in clustering algorithms vary significantly, and it is hard to tell which clustering metric is more efficient and which method shows better performance. Therefore, we conclude that there is a lack of a fair evaluation of different clustering algorithms and clustering metrics. Moreover, we notice that a stochastic model for clustering algorithm is also required to understand clustering algorithms comprehensively.

According to our observation, we propose a mobility-based clustering algorithms under highway scenario in Chapter 3, aiming to increase the **cluster stability**. Furthermore, to analyze the impacts of different traffic parameters on clustering performance, we design a framework for clustering in Chapter 4. Based on this framework, we change the traffic scenarios and compare our algorithms with the benchmark algorithms.

MoDyC: A Mobility-based Scheme for Dynamic Clustering in VANETs

3.1 Introduction

In Chapter 2, we observe that vehicles' mobility patterns are effective clustering metrics that can improve **Cluster stability**. Therefore, in this chapter, we propose a dynamic mobility-based clustering scheme (MoDyC) for the purpose of establishing a stable network backbone for future data aggregation and information transmission. The proposed scheme is based on vehicles' mobility patterns, including moving direction, relative velocity, relative distance, and link lifetime. Different from some previous clustering schemes where nodes are static during cluster formation process, our scheme proposes a dynamic cluster formation process. A "temporary cluster head" is proposed to guide the cluster's construction. Besides, we introduce a "safe distance threshold" to limit the cluster size. The proposed clustering scheme is evaluated in terms of cluster stability, and its performance is compared with the benchmark algorithms, Lowest-ID [32] and VMaSC [7].

The chapter is organized as follows: Section 3.2 presents the proposed clustering algorithm from the aspects of cluster head selection, cluster formation, and cluster maintenance. Section 3.3 presents the simulation environment and the performance analysis of our scheme. Section 3.4 concludes this chapter and briefly introduces the future work.

3.2 Mobility-based Clustering Algorithm

The work focuses on proposing a new clustering algorithm based on V2V communication for highway scenario. It assumes that every vehicle is equipped with an On Board Unit (OBU) wireless transceiver/receiver and has a GPS receiver that can update vehicle's

location on the road. Meanwhile, each vehicle can calculate the relative velocity with respect to its one-hop neighbors, as well as detect the relative distance to its vicinities.

3.2.1 Cluster Definition

We suppose that vehicles enter the road segment one by one with a predefined traffic flow rate (The number of vehicles that are entering the road segment per hour.). Each vehicle moving on the road broadcasts a *Beacon* message at every Beacon Interval (BI). According to the clustering metrics that we have mentioned in Chapter 2, cluster head (CH) should be the vehicle that has higher relatively stability among its neighboring vehicles. Therefore, we choose the vehicle closest to the central geographical position of a cluster to be the CH, so that its neighbors should spend more travel time to leave the cluster, and the cluster is considered to be more stable. Cluster members (CMs) are selected from CH's one-hop neighbor set.

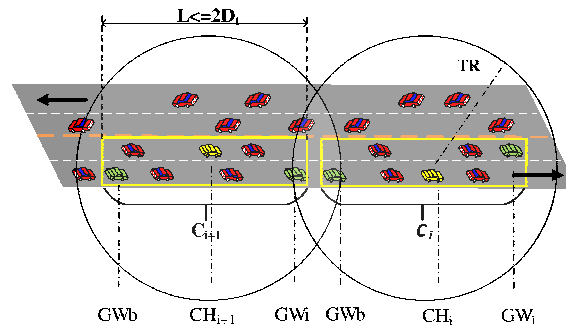


Figure 3.1: Clusters (TR : Transmission Range; L : cluster length; D_t : Safe Distance threshold; GW_i, GW_b : Gateway node.)

Figure 3.1 shows two clusters on a straight road, cluster C_i and cluster C_{i+1} (clusters are represented by rectangles). The cluster head is in the central position, and the length of the cluster is smaller than twice of CH's transmission range (TR). In our proposed clustering scheme, each cluster consists of two gateway nodes moving on the edge of the cluster: one is moving ahead, and another one is moving at the end of the cluster.

Due to the rapid changes in vehicle mobility, vehicles on the edge of CH's transmission range are considered not being stable enough and may cause frequent CM disconnections and CM re-clustering. To solve this problem, we introduce a "Safe Distance Threshold", denoted as D_t , which should be smaller than vehicle's transmission range, $D_t \leq TR$. Therefore, the vehicles within D_t range of the CH are considered as having more stable links with their CH. The size of the cluster is defined as $L \leq 2D_t$. Table 3.1 lists the notations used through this study.

Table 3.1: Notations

Notation	Description
TR	Transmission range
BI	Beacon Interval
D_t	Safe Distance threshold
MI	Merge Interval
V_i	Vehicle i
C_i	Cluster i
CH_i	Cluster head i
CM_i	Cluster member i
CHt_i	Temporary cluster head i
UN_i	Undecided Node i
$Dir(i)$	Moving direction of V_i
ΔD_{ij}	Relative distance between V_i and V_j
L_i	Length of cluster C_i
T_{UN}	Timer for UN transfer to CHt
T_{CHt}	Timer for CHt transfer to CH
T_{wb}	Timer for CH to monitor <i>Beacon</i> from its CM
CID	Cluster ID
CML_i	CM list of CH_i
L_{merge}	Length of the merged cluster
BL_i	Beacon list of CH_i , recording the received <i>Beacons</i>

3.2.2 Cluster State Transition

In the proposed clustering algorithm, a vehicle may have one of the following four states: Undecided Node (UN), Cluster Head (CH), Cluster Member (CM), and Temporary Cluster Head (CHt). The vehicles' states are specified in the following:

- **UN:** Initial state of all vehicles, which means that the vehicle does not belong to any clusters.
- **CH:** The leader of the cluster, which can communicate with all of its members. Each cluster has only one CH and each CH maintains a CM list, CML, recording the information of its CMs.
- **CM:** The normal vehicle which is a one-hop neighbor of a CH. A particular type of CM is the gateway node (GW), which is responsible for inter-cluster communication and is located on the edge of the cluster. Each cluster may have two gateway nodes:

GW_i , moving ahead of the cluster, and GW_b , moving in the end of the cluster.

- **CHt**: The temporary CH vehicle. It only appears at the beginning of cluster formation process and disappears when the CH is selected.

Figure 3.2 illustrates the possible state transitions of a vehicle. The vehicle starts with a UN state and sets a timer T_{UN} , during which it hears *Beacon* message from a CH or a CHt. In our study, *Beacon* messages, which are broadcasted by CH or CHt vehicles, are denoted as *Beacon_ACK* message, aggregating confirmation information. Each *Beacon_ACK* message contains an *ACK_list*, a list of node identifiers. If the UN vehicle does not hear any *Beacon_ACK* message until T_{UN} expires, it changes its state to CHt; otherwise, it changes the state to CM upon receiving a confirmation beacon message, called *Beacon_ACK* message, from a CH or a CHt.

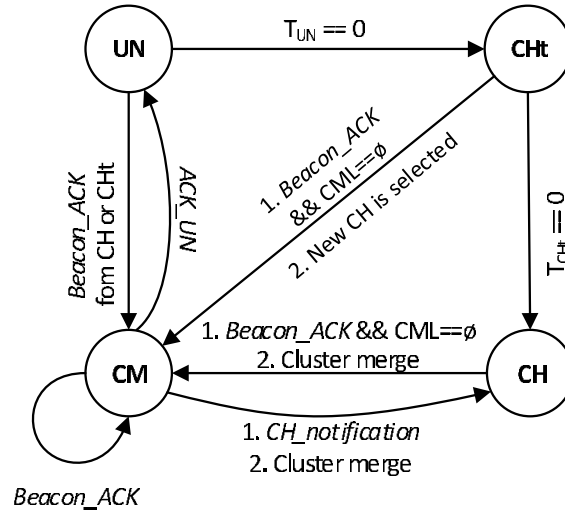


Figure 3.2: State transition machine

The CHt vehicle sets a timer T_{CHt} and initiates a cluster formation process which will be described in the next section. Upon *Beacon_ACK* message reception from a CH, CHt will change its state to CM if it does not have any followers, $CML = \emptyset$. In another situation, the CHt vehicle changes to CM during a CH selection procedure, described in the next section. Otherwise, the CHt vehicle changes to CH when T_{CHt} expires.

When a CH_i hears a *Beacon_ACK* message from a neighboring CH_j , it checks whether it has CMs or not. If its member list $CML_i = \emptyset$, CH_i changes the state to a CM of CH_j . Furthermore, when cluster merging happens, a CH vehicle can also change state to a CM of the merged cluster.

The CM vehicle will change the state to CH when it receives *CH_notification* message from its CHt, or when it is selected as CH in the merged cluster during cluster merging

process. A CM vehicle can hear *Beacon_ACK* messages periodically from its CH or CHt; otherwise, it changes the state back to UN if it is no longer the CM of the current cluster, which will be described in Section 3.2.6.2.

3.2.3 Cluster Formation

As has been shown in Figure 3.3, a vehicle i in the UN state, UN_i , tries to join an existing cluster by listening to the *Beacon* message from a CH or CHt during the time period T_{UN} . If UN_i fails to join an existing cluster when T_{UN} expires, it claims itself as a CHt node CHt_i , and sets its CID. Meanwhile, CHt_i starts a timer T_{CHt} and begins a cluster formation procedure, described in Section 3.2.4.

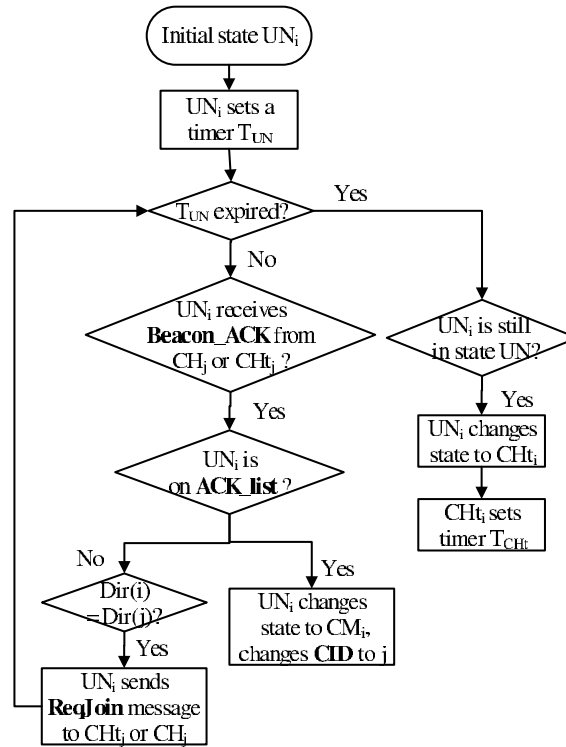


Figure 3.3: Cluster formation

During the time period T_{UN} , if UN_i hears a *Beacon_ACK* message from CH_j or CHt_j , it checks whether it is on the *ACK_list*. If yes, UN_i changes its state to CM directly and sets its cluster identifier $CID = j$; otherwise, it checks whether it is a CM candidate of CH_j or CHt_j . In this chapter, the CM candidate should be the vehicle which are moving in the same direction with its CH, $Dir(i) = Dir(j)$. UN_i sends a *ReqJoin* message to CH_j or CHt_j , if it is a CM candidate.

Upon receiving a *ReqJoin* message from vehicle V_i (V_i could be in the state UN or CH), CH_j checks the following conditions to confirm that the requester is a *qualified CM*:

(1) the relative distance between V_i and CH_j , ΔD_{ij} , should be smaller than the predefined D_t , $\Delta D_{ij} \leq D_t$; (2) V_i is not on CML_j (the CM list of CH_j).

If V_i is a qualified CM, CH_j adds the information of V_i into its CM list CML_j . Meanwhile, CH_j adds the identifier of V_i to its confirmation list, ACK_list , which will be broadcasted within its next *Beacon_ACK* message. It is noticed, only vehicles in the state CH or CHt can broadcast a *Beacon_ACK* message.

3.2.4 Cluster Head Selection

Similar to a CH, CHt can also add qualified CMs according to the conditions mentioned above. However, CHt only adds CMs which are in its neighborhood. Algorithm 1 describes a CH selection procedure initiated by a vehicle CHt_j .

Algorithm 1 CH selection process

```

while  $T_{CHt} \neq 0$  &&  $CHt_j$  is still in state CHt do
  if  $CHt_j$  receives ReqJoin from  $UN_i$  then
    if  $UN_i$  is moving behind  $CHt_j$  &&  $\Delta D_{ij} \leq D_t$  then
       $CML_j \leftarrow UN_i$ 
       $ACK\_list \leftarrow UN_i$ 
       $CHt_j$  broadcasts Beacon_ACK at next BI
    else
      if  $UN_i$  is moving behind  $CHt_j$  &&  $\Delta D_{ij} > D_t$  &&  $CML_j \neq \emptyset$  then
         $CHt_j$  chooses the farthest vehicle  $CM_k$  from  $CML_j$ 
         $CHt_j$  sends CH_notification to  $CM_k$ 
         $CHt_j \rightarrow CM_j$ 
         $CID \leftarrow k$ 
      end if
    end if
  end if
end while
if  $T_{CHt} == 0$  &&  $CHt_j$  is still in state CHt &&  $CML_j \neq \emptyset$  then
   $CHt_j \rightarrow CH_j$ 
end if
When  $CM_k$  receives CH_notification
 $CM_k \rightarrow CH_k$ 
 $CML_k \leftarrow CML_j$ 
 $CID \leftarrow k$ 
 $CH_k$  broadcasts Beacon_ACK

```

Upon receiving a *ReqJoin* message from UN_i moving behind, if CHt_j detects the relative distance $\Delta D_{ij} > D_t$ and $CML_j \neq \emptyset$, CHt_j selects the farthest CM CM_k in its CML_j to be the CH, and sends a *CH_notification* message, containing its CM list CML_j , to inform CM_k to become CH_k . Meanwhile, CHt_j changes to CM_j state and resets $CID = k$. After receiving a *CH_notification* message, CH_k adds CMs and broadcasts a *Beacon_ACK* to inform its CMs to reset $CID = k$. CH_k continues the cluster formation process via adding new CMs. In another case, if CHt_j is still in the state CHt and $CML_j = \emptyset$ when T_{CHt} expires, CHt_j claims itself as CH_j .

3.2.5 Gateway Node Selection

As soon as the CH is selected and the cluster is well formed, CH_k selects two CMs, which are moving on the edge, to be the GW nodes. However, it happens sometimes that two GW candidates have the same relative distance from their CH. To solve this problem, we introduce an estimated connection time between CH and CM, called link lifetime (LLT), to evaluate the link sustainability. A higher LLT represents a more sustainable link. CH will select the GW node which has larger LLT value. The work in [54] defines LLT, shown in Eq. (3.1), when two vehicles are moving in the same or opposite directions. Although vehicle position should be represented by x-coordinate and y-coordinate, this study assumes the trajectory of all vehicular nodes to be a straight line, as the lane width is small. Thus, the y-coordinate can be ignored. We denote the positions of V_k and V_j by x_k and x_j , respectively.

$$LLT_{kj} = \frac{-\Delta v_{kj} * \Delta D_{kj} + \Delta v_{kj} * TR}{(\Delta v_{kj})^2} \quad (3.1)$$

$$\Delta D_{kj} = |x_k - x_j| \quad (3.2)$$

$$\Delta v_{kj} = |v_k - v_j| \quad (3.3)$$

Note that the TR is the transmission range of the vehicle, v_k and v_j are the velocities of CH_k and CM_j , respectively.

3.2.6 Cluster Maintenance

Due to the highly dynamic nature of VANETs, vehicles keep joining and leaving clusters frequently, thus, causing extra maintenance overhead. In our proposed scheme, Clusters are dynamically moving on the road, with their CH inside of the clusters. When CH loses

all of its CMs, it becomes a UN node. Otherwise, it remains as CH until cluster merging process happens. Therefore, in our proposed scheme, the cluster maintenance procedure only deals with cluster merging and a vehicle leaving steps.

3.2.6.1 Cluster merging

The proposed algorithm allows the cluster to be overlapped. However, when two neighboring clusters C_i and C_{i+1} have a big overlapping area, as presented in Figure 3.4, cluster merging procedure is triggered. Instead of having two CHs, a single CH is selected. When the distance of two CHs is smaller than the predetermined threshold D_t , cluster merging procedure begins. To avoid frequent re-clustering, cluster merging is deferred. Instead of starting the cluster merging procedure immediately, the merging procedure begins if two CHs can always hear each other and are always within the range of D_t during the Merge Interval (MI). Once the cluster merging process begins, CH_{i+1} , moving behind, will send a *ReqMerge* message to CH_i , the CH moving ahead. Cluster merging process is described in Algorithm 2.

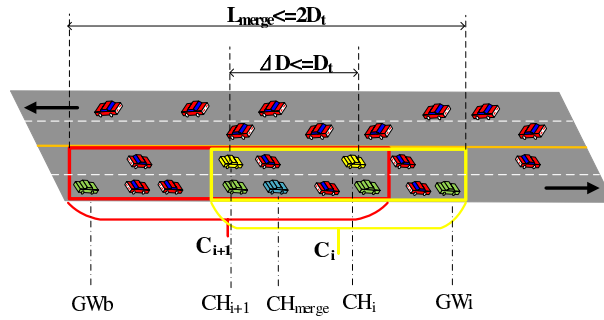


Figure 3.4: Cluster merging (L_{merge} : length of the merged cluster;
 D_t : Safe Distance threshold.)

Upon *ReqMerge* message reception from CH_{i+1} , CH_i estimates the potential merged cluster size L_{merge} . If $L_{merge} \leq 2D_t$, cluster merging is permitted and a CH for the merged cluster, called CH_{merge} , is selected, which is the nearest node to the geographical central position of the merged cluster. After selecting CH_{merge} , previous CHs will send a *ACK_merge* message, containing their CMs list CML , to CH_{merge} , and claims themselves as the CMs of CH_{merge} . The CH_{merge} adds all of the CMs to its CML and broadcasts a *Beacon_ACK* message to inform its CMs to change their CID .

Algorithm 2 Cluster merging process

Upon receiving *ReqMerge*

CH_i estimates L_{merge}

if $L_{merge} \leq 2 * D_t$ **then**

CH_i selects central CM CM_m to be CH_{merge}

CH_i and CH_{i+1} sends *ACK_merge* along with their *CML* to CM_m respectively

$CH_i \rightarrow CM_i$

$CH_{i+1} \rightarrow CM_{i+1}$

$CID \leftarrow m$

end if

Upon receiving *ACK_merge*

$CM_m \rightarrow CH_{merge}$

$CML_m \leftarrow CML_i, CML_m \leftarrow CML_{i+1}$

$CID \leftarrow m$

CH_{merge} broadcasts *Beacon_ACK*

3.2.6.2 Leaving a cluster

In the proposed approach, each CH creates and updates a *CML* dynamically. CH has to monitor the presence of its CMs per every waiting beacon interval, denoted as T_{wb} . Therefore, CH can detect CM disconnection as long as it does not receive the *Beacon* message from its CM at least T_{wb} time period. Moreover, each CH creates a beacon list (BL) in order to record the reception of its CMs' *Beacons*. Once a CH, for example CH_i , receives a *Beacon* message from CM_j , it checks whether CM_j is within the range of D_t . If $\Delta D_{ij} \leq D_t$, CH_i updates the information of CM_j and set $BL_i(j)$ to 1, indicating the reception of the information of CM_j ; otherwise, it deletes CM_j from CML_i .

3.2.6.3 CML and GW updating

Whenever a CH receives a *Beacon* message from its CM, it updates CM's information, for example, the position, in its *CML*. Therefore, every CH can monitor its *CML* dynamically. Once the *CML* is updated, GW_i and GW_b selection functions are triggered immediately, and cluster's gateway information will also be updated according to the process described in Section 3.2.5.

Algorithm 3 Leaving a cluster

```

When  $CH_i$  receives Beacon from  $CM_j$ 
if  $CM_j \in CML_i$  &&  $\Delta D_{ij} \leq D_t$  then
     $BL_i(j) \leftarrow 1$ 
     $CH_i$  updates the information of  $CM_j$  in  $CML_i$ 
end if
Every time when  $T_{wb} == 0$ 
for all  $CM_k \in CML_i$  do
    if  $BL_i(k) == 0$  then
         $CH_i$  deletes  $CM_k$  and  $BL_i(k)$ 
    else
        if  $BL_i(k) == 1$  then
             $BL_i(k) \leftarrow 0$ 
        end if
    end if
end for
Restart  $T_{wb}$ 

```

3.2.7 Important Messages

Table 3.2 presents a set of important messages transmitted during the clustering procedure, and the message dissemination types are demonstrated. Every message must contain the following parameters: message type, source ID, source state, cluster identifier CID , x-coordination x , y-coordination y , velocity v , and direction Dir . Compared to a simple *Beacon* message, *Beacon_ACK* adds a *ACK_list*, and is only broadcasted by a CH or CHt.

Table 3.2: List of important messages

Name of the message	Source	Dissemination type
<i>Beacon</i>	UN or CM	Broadcast
<i>Beacon_ACK</i>	CH or CHt	Broadcast
<i>ReqJoin</i>	Any single node	Towards a CH or CHt
<i>ReqMerge</i>	CH	Towards a CH
<i>CH_notification</i>	CH or CHt	Towards a new merged CH
<i>ACK_UN</i>	CH	Towards a CM

3.3 Performance Evaluation

In this section, we provide a deep analysis of our proposed clustering scheme and compare the clustering performance to the benchmark clustering algorithms Lowest-ID [32] and VMaSC [7]. Lowest-ID was originally proposed in MANET, simply selecting nodes with the lowest identify number among their neighbors to be CHs. VMaSC is the latest and most cited multi-hop clustering algorithm proposed in VANETs. Since both the proposed algorithm and the Lowest-ID algorithm are based on one-hop cluster, the VMaSC is implemented based on one-hop cluster in our simulation. All of the clustering algorithms are implemented on the network simulator NS2 [125], and the testing scenarios are all generated by Simulation of Urban MObility (SUMO) [77].

In the testing scenarios, the road topology consists of a two-lane and two-way road of length 15 *km*. Vehicles are deployed on the street with a predefined traffic flow rate (vehicles per hour), denoted as TFR. The maximum vehicle velocity, being allowed on the road, is called maximum lane speed (MLS). We consider 100 vehicles, 50 vehicles for each direction.

We first evaluate the impacts of "Safe Distance threshold" D_t . Traffic flow rate is set to 1200 vehicles per hour, and maximum lane speed is set to 20 *m/s*, which is considered a regular speed on the road. The value of D_t is set to be in the range of 100-200m, smaller than vehicle's transmission range. Therefore, cluster size is in the range of 200-400m, as defined in our algorithm.

The second simulation evaluates the impacts of the Beacon Interval (BI) on the cluster stability with the increased maximum lane speed (MLS). The set of MLS are specified as follows: 10, 15, 20, 25, 30, 35, 40 *m/s*; traffic flow rate is set to 1200 vehicles per hour. The BI is set to 0.5s, 1.0s and 2.0s respectively.

In the third simulation, we evaluate the impacts of the maximum lane speed (MLS) on cluster stability and compare the clustering performance of the proposed algorithm with Lowest-ID [32], denoted as LID, and one-hop VMaSC [7], denoted as VMaSC_1hop under the same context. The set of maximum lane speed are specified as follows: 10, 15, 20, 25, 30, 35, 40 *m/s*. The traffic flow rate is set to 1200 vehicles per hour. To make a fair comparison, T_{wb} is set to 5.0 s, the same value as *CH_TIMER* when implementing VMaSC [7], and the same value of information updating interval in LID [32].

For each testing scenario, simulation runs for 800s. The clustering process starts at time T_{start} , the time when all vehicles have entered the road, and ends at time T_{end} , before which most of the vehicles are still on the road. According to the testing scenarios, we

set $T_{start} = 160s$ and $T_{end} = 460s$. Therefore, the clustering simulation time is 300s. All of the simulations run ten times. According to previous related works(e.g., [26], [7], [41]), our simulation parameters are selected as illustrated in Table 3.3.

Table 3.3: Simulation parameters

Parameter	Value
Simulation time	300 s
MAC protocol	IEEE 802.11p
TR	200 m
Number of vehicles	100
Road length	15 km
Length of car	5 m
Acceleration rate	$2.6 m/s^2$
Deceleration rate	$4.5 m/s^2$
Maximum lane speed (MLS)	10-40 m/s
Traffic flow rate (TFR)	1200 <i>vehicles/hour</i>
D_t	100-200 m
BI	1.0 s
MI	10.0 s
T_{wb}	5.0 s
Propagation model	Two-Ray Ground
Number of iterations	10
Mobility model	Car-following model

3.3.1 Performance Metrics

The cluster performance metrics, used for cluster stability evaluation and comparison, are described as follows:

- **Average number of clusters:** as long as the CH is alive, there is a cluster. This metric allows us evaluating the quality of cluster formation. In the worst case, each vehicle represents an independent cluster; therefore, clustering algorithm is meaningless.
- **Average CH duration:** this metric represents the cluster's lifetime, the time interval between a vehicle becoming a CH and changing to another state. In general, a longer duration of CH represents a more stable cluster. In this chapter, the normalized average CH duration is the percentage time period of the total simulation

time.

- **Average CM duration:** it defines the average time interval from a node joining an existing cluster as a CM to leaving the connected cluster or to becoming a CH. The normalized average CM duration is the percentage time period of the total simulation time.
- **Average CH change rate (per second):** the cluster head change rate defines the number of state transitions from CH to another state per unit time.
- **Average state change (per node):** this metric indicates the number of state transitions in each vehicle during the clustering procedure.
- **Clustering efficiency:** it is defined as the percentage of vehicles participating in clustering procedure (vehicles which are not in UN state) during the simulation. A higher clustering efficiency means a better clustering performance.
- **CM disconnection frequency (per second):** it illustrates the total number of times that CMs lose the connections to their current CHs per unit time.

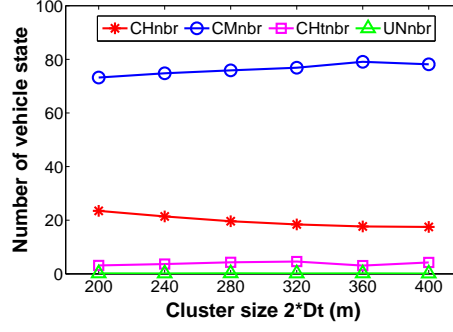
3.3.2 Results Analysis

3.3.2.1 Impact of "Safe Distance Threshold" D_t

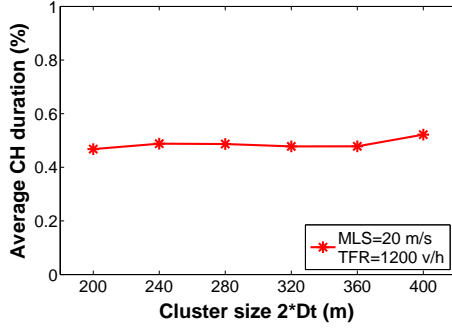
Figure 3.5 presents the impacts of "Safe Distance Threshold" D_t . In Figure 3.5(a), with the increased D_t , less clusters are organized during the simulation. This is because that more vehicles are combined in a cluster as CMs when the cluster length increases under the same traffic density. The numbers of vehicles in CHt and UN states (both are temporary states) remain stable when cluster size is becoming larger. Figure 3.5(b) shows the average CH duration, represented as the percentage of total simulation time. The average CH duration increases slightly but remains relatively stable, when D_t increases. Figure 3.5(c) illustrates that the average CM duration decreases slightly with the increased cluster size. We observe that D_t has small impacts on both the CH duration and CM duration.

3.3.2.2 Impact of Beacon Interval (BI)

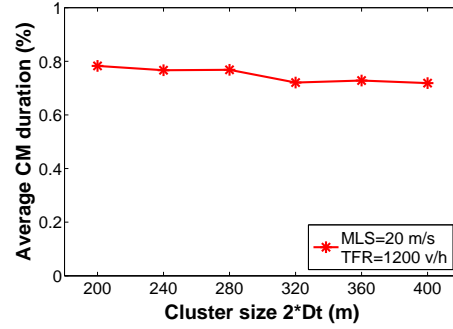
According to ETSI standard [13], the periodic Beacon message, called Cooperative Awareness Message (CAM) is broadcasted with the frequency 1-10Hz (0.1s-1s). Therefore, in our simulation, we set BI to 1.0s as the default value, and change BI to 0.5s and 2.0s respectively, in order to evaluate its impacts on our proposed algorithm.



(a) Average number of vehicles in each state



(b) Average CH duration (%)



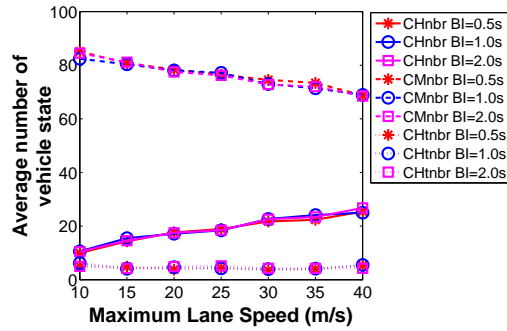
(c) Average CM duration (%)

Figure 3.5: Impacts of D_t on cluster performance

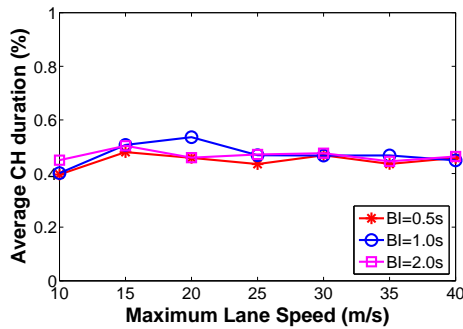
The results in Figure 3.6 show the cluster performance in terms of average number of vehicles in each state (Figure 3.6(a)), average CH duration (Figure 3.6(b)), and average CM duration (Figure 3.6(c)). From the results, we observe that BI has a slight impact on the cluster performance. According to the simulation results, BI is set to 1.0s in the rest of the simulation. In Figure 3.6(a), we observe that the number of vehicles in the state CH increases with the increased maximum vehicle velocity, and meanwhile, the number of CM vehicles decreases. This is because that with the increased vehicle velocity, some CMs may move out of the cluster and may become isolated vehicles. Then, if the isolated vehicle cannot successfully re-connect to another existing cluster, a new cluster will be formed, increasing the number of CHs.

3.3.2.3 Impact of maximum lane speed

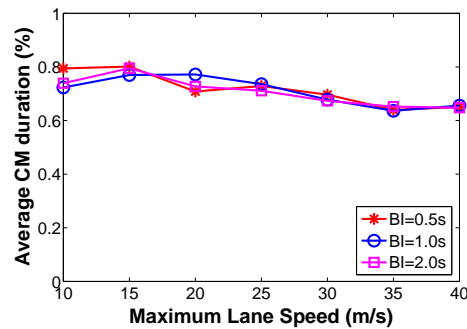
Figure 3.7 presents the averaged lifetime of each vehicle state with the increased maximum lane speed (MLS), in the proposed algorithm. We observe that when vehicle velocity increases from 10m/s to 40m/s, vehicle state lifetime is relatively stable. The CHt lifetime is very small because it is a temporary state which only appears at the beginning of a cluster formation process.



(a) Average number of vehicles in each state



(b) Average CH duration (%)



(c) Average CM duration (%)

Figure 3.6: Impact of BI on cluster performance with the increased maximum lane speed (MLS)

The results in Figure 3.8, 3.9, and 3.10, compare the cluster stability among the proposed clustering algorithm to Lowest-ID (LID) and VMaSC (VMaSC_1 hop), from the aspects of the vehicle state duration, the number of vehicle states, and the number of state changes, respectively.

Figure 3.8(a) and Figure 3.8(b) reveal the impacts of the maximum vehicle velocity on the averaged CH lifetime and CM lifetime. It is obvious that both the averaged CH and CM duration of LID and VMaSC_1hop decrease rapidly when MLS increases. In Figure 3.8(a), when vehicles move slowly on the road, both the mean CH duration of LID and VMaSC_1hop are higher than that of our scheme. However, their CH duration decreases rapidly with the increased MLS, especially for LID. The CH duration of VMaSC_1hop is always higher than that of our scheme until MLS becomes bigger than 33 m/s . This is because in our scheme, CHt assists cluster formation and CH is selected during the cluster formation process, while CH selection happens in the beginning in VMaSC_1hop and LID. In Figure 3.8(a), the CM duration in our scheme remains the highest one when MLS is bigger than 17 m/s . When MLS becomes bigger than 30 m/s , the CM duration of our scheme is almost two times of VMaSC_1hop. The results in Figure 3.9(b) and Figure

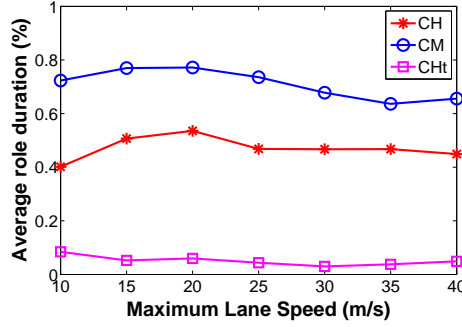
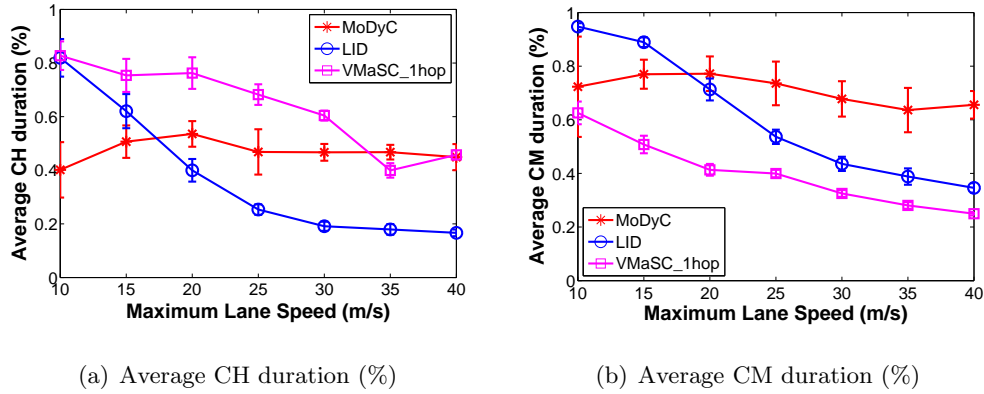


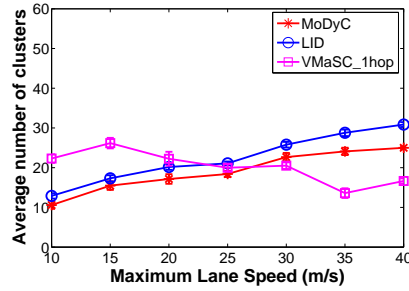
Figure 3.7: Average duration of each vehicle state of MoDyC

Figure 3.8: Comparison of vehicle states' lifetime under the impact of vehicle's MLS
(Confidential Interval: 95%)

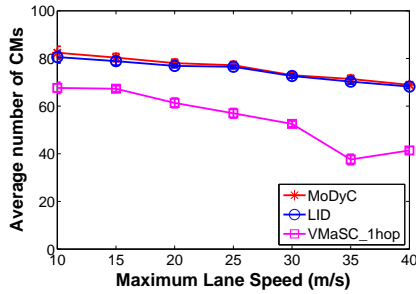
3.9(c) will explain this consequence that with the increased MLS, the cluster becomes less stable, and many vehicles change to the temporary state in VMaSC_1hop [7], therefore, reduces the average CH and CM duration.

The results in Figure 3.9 show that both the number of the CHs and the number of UNs in our scheme are slightly lower than the results of LID. Moreover, when MLS becomes larger, many CHs and CMs in VMaSC_1hop change to UN state (SE state in [7]). Therefore, the number of CHs and CMs of VMaSC_1hop in Figure 3.9(a) and in Figure 3.9(b) decrease and the number of UNs in Figure 3.9(c) grows quickly.

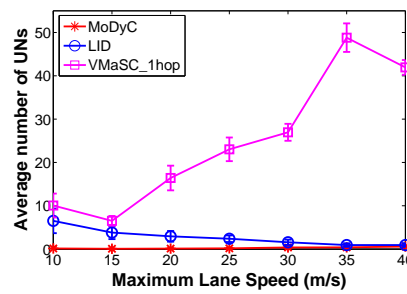
Figure 3.10 demonstrates the details of state transitions during the clustering process. The results in Figure 3.10(a) and Figure 3.10(b) reveal that both the CH change rate and vehicle state change times of LID grow quickly when MLS is increasing. In LID, CH changes its state as soon as it detects a neighbor vehicle is having an identifier smaller than itself. When vehicle velocity increases, vehicle's neighbor list changes considerably, causing more frequent CH change rate. CH change rates of VMaSC_1hop and our scheme are both very low and remain relatively stable in Figure 3.10(a). In our scheme, CH



(a) Average cluster number



(b) Average number of CMs



(c) Average number fo UNs

Figure 3.9: Vehicle state number comparison under the impact of vehicle's MLS

(Confidential Interval: 95%)

may change to a CM when cluster merging happens or when it loses all of its CMs, as mentioned in Section 3.2. In Figure 3.10(b), the number of state transitions for each vehicle in VMaSC_1hop and our scheme are higher than that of LID. This is because more vehicle states are defined in these two schemes compared to LID.

The CM disconnection frequency, shown in Figure 3.10(c), presents a similar growth trend compared to the results in Figure 3.10(b). It is because that vehicle state transition always happens when a CM loses the link connection with its current CH. Since state transition in LID is identifier-based, Figure 3.10(c) only compares our scheme and VMaSC_1hop. It is obvious that our scheme shows a very low CM disconnection frequency compared to VMaSC_1hop, indicating that our scheme provides higher cluster stability.

From the results in Figure 3.10(d), we observe that both LID and our scheme perform a very high clustering efficiency, which is close to 100% when MLS increases. It means that almost all of the vehicles on the road participate in clustering procedure during the simulation. However, with the growth of MLS, the clustering efficiency of VMaSC_1hop decreases significantly. It is because the number of UN nodes increases quickly when MLS becomes high, as shown in Figure 3.9(c).

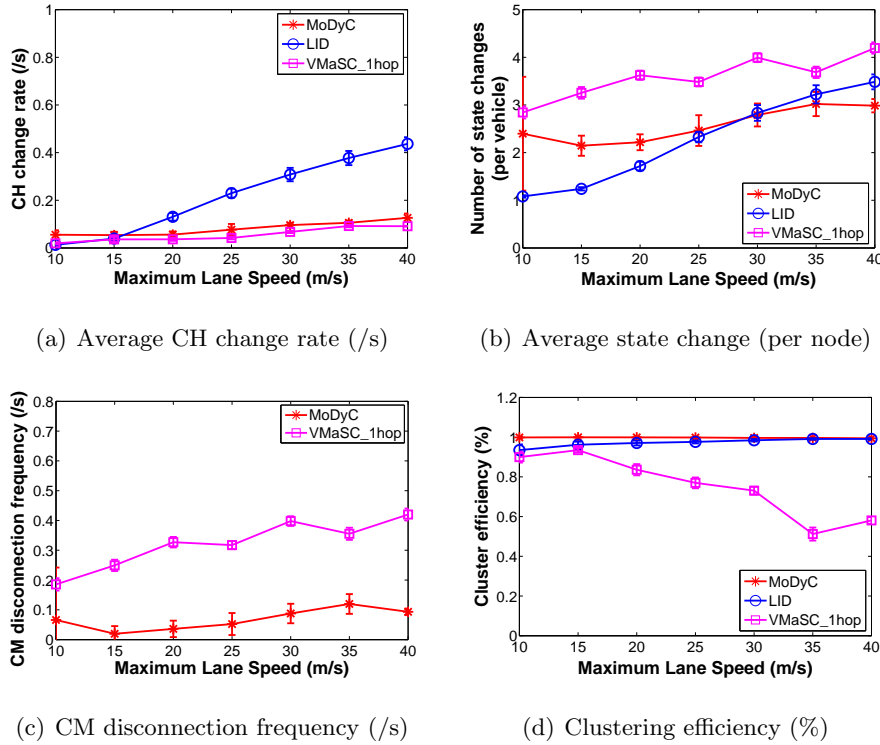


Figure 3.10: Cluster stability comparison under the impact of vehicle's MLS
(Confidential Interval: 95%)

3.4 Conclusion

In this chapter, we introduce a dynamic mobility-based clustering scheme for VANETs. In our scheme, vehicles are grouped in a single-hop based cluster, limited by a predetermined value "Safe Distance Threshold" D_t , where $D_t \leq TR$. CH is selected as the vehicle which is closest to the geographical center of a cluster, and CMs are within D_t range of the CH, moving in the same direction. A new vehicle state, called temporary cluster head CHt, is proposed in order to help cluster formation process. CHt only exists at the beginning of a cluster formation procedure. It changes its state to CH or CM as soon as the CH is selected. Cluster maintenance mechanisms are proposed, including cluster merging and leaving a cluster procedure.

Extensive simulations in NS2 with the vehicle mobility input from SUMO demonstrate the superior clustering performance of our scheme over LID and VMaSC schemes, regarding average cluster number, average CH duration, average CM duration, CH change rate, number of vehicle state changes, CM disconnection frequency, and clustering efficiency. The simulation results show that our proposed clustering scheme provides higher cluster stability even in a high dynamic traffic scenario.

Even though the proposed MoDyC shows good performance in terms of cluster stability and cluster efficiency compared with LID and VMaSC schemes, there still exist some limitations. In MoDyC, we simply use the inter-vehicle distance to construct vehicle clusters and clusters are constructed by adding vehicles one by one. Therefore, cluster formation process may create unnecessary time costs and extra overhead.

In the next chapter, we will present a more complete framework of clustering algorithm to not only solve the limitations mentioned above, but also analyze the impacts of different mobility metrics on clustering performance. At the same time, we will evaluate our framework under various traffic scenarios and compare it with the benchmark algorithms.

A Unified Framework of Clustering Approach (UFC) in VANETs

4.1 Introduction

Effective clustering algorithms are indispensable to solve the scalability problem in Vehicular Ad-hoc Networks (VANETs). Even though cluster stability has been improved in current existing clustering algorithms as summarized in Chapter 2, including MoDyC (described in Chapter 3), it is still hard to address which clustering metric performs the best. In this chapter, we propose a Unified Framework of Clustering approach (UFC), composed of three parts: i) Neighbor Sampling (NS); ii) Backoff-based Cluster head Selection (BCS); iii) BackUp Cluster Head based cluster maintenance (BUCH). Based on UFC, we implement three different mobility-based clustering metrics: vehicle relative position, relative velocity, and link lifetime; and we evaluate how these clustering metrics affect cluster's performance. Furthermore, we provide a detailed analysis of UFC with parameters optimization. Extensive comparison results among UFC, Lowest-ID, and VMaSC algorithms demonstrate that our clustering approach results in high cluster stability, especially under high dynamic traffic scenarios. The main contributions of this chapter are listed as follows:

- We propose a neighbor sampling (NS) scheme to filter out the unstable neighbors and to select the stable neighbor set. We assume that only vehicles in the stable neighbor set have the possibility to build connections with the cluster head.
- We propose a Backoff-based cluster head selection (BCS) scheme in order to reduce clustering management overhead. Each vehicle makes its own cluster head decision in a distributed manner by calculating its own backoff timer. The vehicle, with higher probability of being a cluster head, will set a smaller backoff timer. In our work,

three mobility-based clustering metrics are implemented to compute the backoff timer, including link lifetime, relative speed, and relative distance.

- We propose a BackUp Cluster Head based cluster maintenance (BUCH) scheme, in order to mitigate the influence of intermittent connectivity. Since clusters may overlap, each vehicle may hear more than one cluster head. Thus, a Backup cluster head list can be easily created in each vehicle to cache several backup cluster heads, in the case of loss of connection to the main cluster head.
- A detailed analysis of the proposed clustering scheme is presented by adjusting the corresponding metrics under different typical traffic scenarios, including both relative stable and high dynamic traffic scenarios.
- Clustering performance metrics are categorized into macroscopic and microscopic levels. A detailed comparison between UFC scheme, Lowest-ID scheme, and VMaSC scheme is presented, in terms of both macroscopic and microscopic performance metrics.

The rest of the chapter is organized as follows: Section 4.2 presents our proposed clustering algorithm from the aspects of cluster head selection, cluster formation, and cluster maintenance. Section 4.3 presents the simulation environment and analyzes UFC scheme from the view of parameter optimization; then, a fair comparison is given in this section. Section 4.5 concludes this chapter.

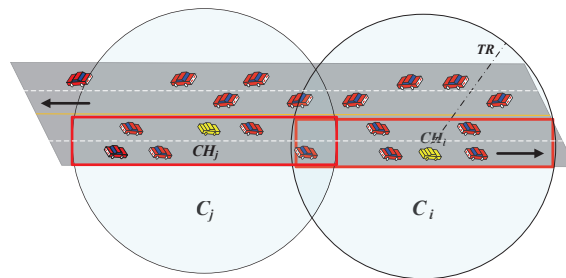


Figure 4.1: Clusters C_i and C_j

4.2 Description of The Framework

The Unified Framework of Clustering approach (UFC) is only based on V2V communication type. All vehicles are assumed to be equipped with a GPS system which provides vehicle's basic information, including vehicle's current location, velocity, and moving direction. Moreover, each vehicle can both calculate speed difference and detect relative

Table 4.1: Notations

Notation	Description
TR	Transmission Range
BI	Beacon Interval
$T_{collect}$	System timer for <i>Beacon</i> exchange and collection
T_W	Backoff timer for CH selection
Δv_{th}	Speed difference threshold between two vehicles
MAX_CM	Maximum number of CMs in a cluster
V_i	Vehicle i ; i is the ID of vehicle
N_i	Number of stable neighbors of V_i
SN_i	Stable neighbor set of V_i
$SN_i(j)$	Neighbor list entry of SN_i in V_i
CH	Cluster head
CM	Cluster member
CCM	Candidate cluster member
CML_i	CM List of CH_i
LLT_{ij}	Link lifetime between V_i and V_j
\overline{LLT}_i	Average LLT between V_i and its stable neighbors
Δv_{ij}	Speed difference between V_i and V_j
ΔD_{ij}	Distance between V_i and V_j
C_i	Cluster i , i is the ID of CH
$\Delta \overline{V}_{C_{ij}}$	Speed difference between C_i and C_j
$\Delta \overline{V}_{C_{th}}$	Speed difference threshold between two clusters
BCH	Backup cluster head
T_{cm}	Timer for CM to hear <i>Beacon</i> from its CH
T_{ch}	Timer for CH to hear <i>Beacon</i> from its CM
$BCHL_i$	Backup CH List in vehicle V_i
$BCHL_i(j)$	List entry of BCH_j , stored in $BCHL_i$ of V_i
MAX_BCH	Maximum number of BCHs in the BCHL
T_{bch}	Timer for CCM and CM to build BCHL

distance with respect to its vicinities.

Vehicles exchange their information periodically with their one-hop neighbors via *Beacon* messages at every Beacon Interval (BI). Information contained in *Beacon* message includes vehicle's identifier ID , vehicle's current state R , cluster identifier $ID_cluster$, current position (x, y) , current velocity v , and moving direction Dir .

Figure 4.1 presents an example of cluster network topology. Two clusters are presented on the road, with single cluster head in each cluster. Clustering procedure will be

described in the following sections: cluster head selection, cluster formation, and cluster maintenance. The notations used are presented in Table 4.1.

4.2.1 Vehicle states

In UFC, each vehicle operates in one of the following 4 states:

- Undecided Node (UN): Initial state of all vehicles, meaning that the vehicle does not belong to any cluster.
- Cluster Head (CH): The leader of the cluster, which can communicate with all of its members. Each cluster has only one CH.
- Cluster Member (CM): The vehicle which can directly be attached to an existing CH.
- Candidate Cluster Member (CCM): The vehicle which intends to be a CM of an existing cluster, but has not yet received a confirmation message.

The transition between two of these states are triggered by different events, presented through a state machine in Figure 4.2. The state transition process will be described in the following subsections, through the presentation of the main procedures of our UFC algorithm, NS, BCS, as well as BUCH.

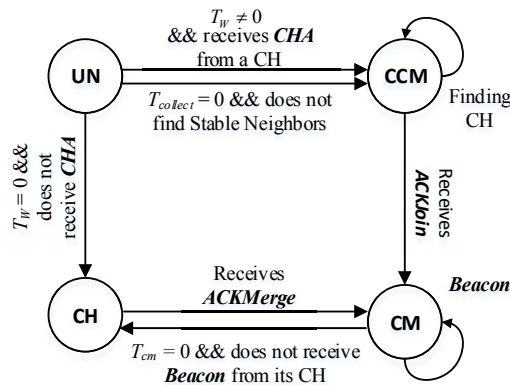


Figure 4.2: State transition

4.2.2 Neighbor Sampling (NS)

At the beginning of the clustering procedure, each node is in an initial state, indicated as UN node. The system starts a timer, called $T_{collect}$, during which vehicles exchange and collect *Beacons* to discover their one-hop vicinities, called Potential Neighbor set

(PN). However, not all vehicles in PN are ideally to be clustered. In order to reduce the redundant computation and useless message exchanges, the Neighbor Sampling process selects a set of stable neighbors from PN , denoted as Stable Neighbor set (SN), where $SN \subset PN$. The vehicle in SN is defined as the neighboring vehicle that presents a similar mobility pattern: (1) moving in the same direction; and (2) the speed difference Δv is smaller than the predetermined threshold Δv_{th} . The Neighbor Sampling process is presented in Algorithm 4.

Every vehicle V_i maintains a set of stable neighbors SN_i , containing several entries, indicated as $SN_i(j)$. Every neighbor list entry $SN_i(j)$ contains the following information:

Term	Description
(x_j, y_j)	position of vehicle V_j
v_j	velocity of vehicle V_j
LLT_{ij}	Link lifetime between vehicle V_i and V_j
Δv_{ij}	relative speed between vehicle V_i and V_j
ΔD_{ij}	relative distance between vehicle V_i and V_j

Algorithm 4 Neighbor sampling (NS)

```

while  $T_{collect} > 0$  do
  if  $UN_i$  receives Beacon from  $UN_j$  then
    if  $Dir(i) == Dir(j) \ \&\& \ \Delta v_{ij} \leq \Delta v_{th}$  then
       $UN_i$  calculates  $LLT_{ij}$ 
      if  $UN_j \in SN_i$  then
         $UN_i$  updates  $SN_i(j)$ 
      else
         $UN_i$  adds the entry  $SN_i(j)$  to  $SN_i$ 
         $N_i \leftarrow N_i + 1$ 
      end if
    end if
  end if
end while

```

It happens sometimes that some UN vehicles could not find any stable neighbors during $T_{collect}$ time period, where $SN_i = \emptyset$. Under these circumstances, such UN vehicles will directly change state to CCM, instead of participating in the Backoff-based CH Selection (BCS) process. Meanwhile, they will create a Backup CH List (BCHL) and try to find a suitable CH to follow, which will be explained in Algorithm 5 in Section 4.2.5.

4.2.3 Backoff-based CH Selection (BCS)

As we have observed in Chapter 2, almost all cluster head selection schemes are based on mobility metric exchanging and broadcasting. This kind of scheme increases the packet collision probability, as well as the management overhead. Instead of broadcasting cluster metrics, our proposed CH selection scheme allows each vehicle to set up its own backoff timer, T_W , in a distributed manner, waiting to broadcast a CH Announcement message *CHA*. The first vehicles broadcasting *CHA* messages among their neighbors will become initial CHs.

This work proposes two methods for CH selection. The first method is a metric-based method, including the following metrics: average Link Lifetime \overline{LLT}_i , average relative distance $\overline{\Delta D}_i$, and average relative speed $\overline{\Delta v}_i$. The second one is a random-based method. The details are described as follows.

4.2.3.1 Metric-based CH selection method

Link Lifetime (LLT), also called Link Expiration Time (LET) [75], describes the link sustainability, representing the duration of time when two vehicles remain connected. The work in [54] gives the definition of LLT, shown in Eq. (4.1), when two vehicles are moving in the same direction. Eq. (4.1) defines LLT calculation. Although vehicle position should be represented by x-coordinate and y-coordinate, this study assumes the trajectory of all vehicular nodes to be a straight line, as the lane width is small. Thus, the y-coordinate can be ignored. We denote the positions of V_i and V_j by x_i and x_j , respectively.

$$LLT_{ij} = \frac{-\Delta v_{ij} * \Delta D_{ij} + |\Delta v_{ij}| * TR}{(\Delta v_{ij})^2}. \quad (4.1)$$

Note that TR is the transmission range of a vehicle.

During $T_{collect}$ time period, for each received *Beacon* message from vehicle V_j , where $V_j \in SN_i$, vehicle V_i calculates the metric, LLT_{ij} , ΔD_{ij} , or Δv_{ij} , where $\Delta D_{ij} = x_i - x_j$, and $\Delta v_{ij} = v_i - v_j$. Then, vehicle V_i records the metric in its stable neighbor set SN_i .

When $T_{collect}$ is expired, V_i calculates the clustering metric \overline{LLT}_i , $\overline{\Delta D}_i$ or $\overline{\Delta v}_i$, according to Eq. (4.2), (4.3), or (4.4), respectively. The backoff time is calculated as in Eq. (4.5).

$$\overline{LLT}_i = \frac{\sum_{V_j \in SN_i} LLT_{ij}}{N_i} \quad (4.2)$$

$$\overline{\Delta D}_i = \frac{\sum_{V_j \in SN_i} |\Delta D_{ij}|}{N_i} \quad (4.3)$$

$$\overline{\Delta v_i} = \frac{\sum_{V_j \in SN_i} |\Delta v_{ij}|}{N_i}, \quad (4.4)$$

where N_i is the number of vehicles in SN_i of vehicle i .

$$T_{Wi} = T_{min} + (T_{max} - T_{min}) * M_i + \theta, \quad (4.5)$$

$$M_i = \begin{cases} 1 - \overline{LLT}_i / \Delta LLT_{max} \\ \overline{\Delta D}_i / \Delta D_{max} \\ \overline{\Delta v}_i / \Delta v_{max} \end{cases} \quad (4.6)$$

where M denotes the metric, T_{min} and T_{max} are set to 0s and 2s, respectively, LLT_{max} is given as the same time as the simulation time T_{sim} , ΔD_{max} is given as the same as TR , Δv_{max} is set to the same value as Δv_{th} , and $\theta \sim U(0, 0.1)$ follows a uniform distribution. Here, θ is added to the end of the equation to avoid the same T_W .

4.2.3.2 Random-based CH selection method

Different from metric-based method, when T_{colloc} expires, each vehicle chooses a random backoff time according to Eq. (4.7), where T_i follows a uniform distribution $T \sim U(0, T_{max})$, and $\theta \sim U(0, 0.1)$. According to experiments results, T_{max} is set to 2 seconds.

$$T_{Wi} = T_i + \theta_i \quad (4.7)$$

During T_{Wi} , if vehicle i receives a *CHA* message, it gives up CH competition process, cancels T_{Wi} , and changes its state from UN to CCM; otherwise, when T_{Wi} expires, vehicle i changes its state from UN to CH and broadcasts a *CHA* message to inform its vicinities of this state transition. The CH election process is described in Algorithm 5. A *CHA* message should contain the following information:

- Message type, denoted by *CHA*;
- Vehicle ID: denoted by *ID*;
- Vehicle location: denoted by (x_i, y_i) ;
- Stable neighbor list: denoted by SN_i ;
- Number of stable neighbors: denoted by N_i .

Algorithm 5 Backoff-based CH selection (BCH)

```

When  $T_{collect} == 0$ 
 $UN_i$  calculates  $\overline{M}_i$ 
 $UN_i$  calculates  $T_{W_i}$  and starts timer  $T_{W_i}$ 
 $flag\_T_W \leftarrow 1$ 
while  $T_{W_i} > 0$  do
  if  $UN_i$  receives  $A$  from  $CH_j$  then
     $flag\_T_W \leftarrow 0$ 
     $UN_i \rightarrow CCM_i$ 
    goto Cluster formation
  end if
end while
if  $flag\_T_W == 1$  then
   $UN_i \rightarrow CH_i$ 
   $CH_i$  broadcasts  $CHA$ 
end if

```

4.2.4 Cluster Formation

When a vehicle UN_j receives CHA message from CH_i , it immediately changes its state to a candidate cluster member CCM_j and sends a *ReqJoin* message to CH_i . Meanwhile, CCM_j sets a timer T_{ack} , waiting for a confirmation notification from CH_i . The cluster formation process is described in Algorithm 6. After CH_i receives the *ReqJoin* message from CCM_j , it firstly checks the total number of existing CMs in the cluster. If the number of existing CMs is less than the cluster capacity, denoted as MAX_CM , CH_i adds CCM_j to its cluster member list CML_i and sends back a *ACKJoin* message; otherwise, CH_i ignores the *ReqJoin* message. If CCM_j receives *ACKJoin* before T_{ack} expires, CCM_j changes its state to CM_j ; otherwise, CCM_j tries to join another cluster, which will be described in the next section.

4.2.5 Backup CH based vehicle re-clustering

Due to the high dynamic nature of VANETs network topology, vehicles keep joining and leaving clusters frequently. Indispensable vehicle re-clustering process guarantees that a CM can find a proper cluster to follow as long as it loses the contact with its current CH. However, a big delay in re-clustering process may lead to serious consequences, especially when delay-sensitive applications are implemented. To solve this problem, a caching cluster head scheme is proposed, aiming to reduce the vehicle re-clustering delay. A Backup CH List (*BCHL*) is created and updated in every CCM and CM node. Every time the

Algorithm 6 Cluster formation

```

if  $UN_j$  receives  $CHA$  then
   $UN_j \rightarrow CCM_j$ 
   $CCM_j$  sends  $ReqJoin$  to  $CH_i$ 
   $CCM_j$  starts timer  $T_{ack}$ 
  goto Waiting confirmation
end if

if  $CH_i$  receives  $ReqJoin$  from  $CCM_j$  then
  if  $N_i \leq MAX\_CM$  then
     $CML_i(j) \leftarrow CCM_j$ 
    sends  $ACKJoin$  to  $CCM_j$ 
  end if
end if

```

Algorithm 7 Waiting confirmation

```

while  $T_{ack} \geq 0$  do
  if  $V_i$  receives  $ACKJoin$  from its  $CH_j$  then
     $V_i \rightarrow CM_i$ 
  end if
5: end while

if  $V_i$  does not receive  $ACKJoin$  then
  goto BCH election
end if

```

vehicle loses the contact with its current CH, it starts to find the most qualified backup CH (BCH) to follow. The backup CH based cluster maintenance procedure is introduced in the following parts.

4.2.5.1 Backup CH List (BCHL) creation

Following the above description, every CCM node, for example CCM_i , which has not been clustered will set a timer T_{bch} , during which, CCM_i may hear *Beacon* messages from one or more than one CHs. For each detected CH_j , CCM_i only selects the qualified CHs and records their link lifetime LLT_{ij} and their relative speed Δv_{ij} in its Backup CH List $BCHL_i$ (Line 3-5 in Algorithm 8). The qualified CH should meet the following criteria: (1) moving in the same direction as CCM_i ; (2) $\Delta v_{ij} \leq \Delta v_{th}$.

Algorithm 8 Backup CH List BCHL creation

```

 $CCM_i$  starts timer  $T_{bch}$ 
while  $T_{bch} \geq 0$  do
  if  $CCM_i$  receives a Beacon from  $CH_j$  then
    if  $Dir_j == Dir_i \ \&\& \ \Delta v_{ij} \leq \Delta v_{th} \ \&\& \ Size(BCHL_i) \leq MAX\_BCH$  then
       $BCHL_i(j) \leftarrow (CH_j, LLT_{ij}, \Delta v_{ij})$ 
      BCHLi ranking
    end if
  end if
end while

```

4.2.5.2 Backup CH List (BCHL) ranking

In the $BCHL_i$ of vehicle V_i , BCHs are ordered according to their priorities: 1) BCH which has longer link lifetime LLT has higher priority; 2) if all LLT in the $BCHL_i$ are equal, BCH with less Δv will be given a higher priority. Then, the vehicle V_i chooses the BCH, which has the highest priority, and sends a *ReqJoin* message as long as V_i loses the connection with its current CH. Algorithm 9 presents how a vehicle reconnects to a new CH.

If V_i fails to join the new cluster, it sends a *ReqJoin* message to another BCH, which has the second highest priority on the $BCHL_i$, and waits for the confirmation. The vehicle V_i repeats this process until it successfully reconnects to a new CH. Apparently, when the size of the backup CH list $BCHL$ increases, the re-clustering delay increases. Thus, the size of $BCHL$ should be limited by the predetermined value MAX_BCH , which will be discussed in Section 4.3.

4.2.6 Cluster maintenance

Clustering maintenance process in UFC can be divided into the following parts: *CML* updating, *BCHL* updating, leaving a cluster, and cluster merging.

4.2.6.1 CML updating

Each CH maintains a dynamic CML. For each beacon interval, if CH hears a *Beacon* from one of its CMs, it updates the information in its *CML*. Each CH starts a control timer T_{ch} periodically, in order to monitor the *Beacons* of its CMs. Otherwise (i.e., if CH does not receive *Beacon* message from one of its CM during T_{ch} time period), CH removes this CM from its *CML*.

Algorithm 9 Backup CH (BCH) selection

```

 $V_i$  checks  $BCHL_i$ 
if  $BCHL_i == \emptyset$  then
  if  $V_i$  is a CCM node then
     $CCM_i$  restarts  $T_{bch}$ 
  else
    if  $V_i$  is a CM node then
       $CM_i \rightarrow CH_i$ 
    end if
  end if
else
  if  $Size(BCHL_i) \geq 1$  then
     $V_i$  chooses  $CH$  with the highest priority
     $V_i$  sends ReqJoin and starts  $T_{ack}$ 
    goto Waiting confirmation
  end if
end if

```

4.2.6.2 BCHL updating

The backup CH list ($BCHL$) is stored in a CCM/CM vehicle. When a vehicle hears a *Beacon* from a qualified BCH, as we have defined above, it updates the entry of $BCHL$, as well as records the BCHs. Similar to T_{ch} , timer T_{bch} is used in CCM/CM node, in order to monitor the $BCHL$. If a CCM/CM vehicle discovers that it could not hear an existing BCH any more during T_{bch} , it removes this BCH from its $BCHL$. The $BCHL$ is reordered as long as the entry is updated.

4.2.6.3 Leaving a cluster

Each CM sets a timer T_{cm} to monitor its connection with CH. If CM does not receive *Beacon* from its CH during T_{cm} , it considers itself out of the communication range of its current CH. Then, CM selects a BCH from its $BCHL$ and directly sends *ReqJoin* to this BCH. If $BCHL == \emptyset$, the CM claims itself as a new CH, and tries to create a new cluster, as described in Algorithm 9.

4.2.6.4 Cluster merging

When two neighboring CHs, CH_i and CH_j , are moving in the same direction within the transmission range of each other, the cluster merging procedure will be triggered. Instead

of starting merging immediately, we introduce a merge interval, called T_{merge} , to defer cluster merging. Cluster merging process starts only if two CHs can contact with each other consecutively during T_{merge} , and the average speed difference between two clusters $\Delta\overline{V}_{Cij}$, shown in Eq. (4.8), is smaller than the predefined threshold $\Delta\overline{V}_{Cth}$. The CH vehicle with less CMs will give up the leadership and another CH becomes the CH of the merged cluster, called CH_{merge} . CMs in the dismissed cluster will be automatically included in the merged cluster.

$$\Delta\overline{V}_{Cij} = |\overline{V}_{Ci} - \overline{V}_{Cj}| \quad (4.8)$$

$$\overline{V}_{Ci} = \frac{\sum_{V_k \in CML_i} |\Delta v_{ik}|}{N_i}, \quad (4.9)$$

where \overline{V}_{Ci} is the average speed difference of cluster C_i , $|\Delta v_{ik}|$ is the speed difference between CH_i and CM_k , and N_i is the number of CMs in the cluster C_i .

4.2.7 Main messages

Table 4.2 presents the important messages transmitted during the clustering procedures. The essential contents included in these messages are described, where T represents the message type, R is the state of the vehicle, L indicates the location, v is the speed and Dir is the moving direction. Table 4.3 introduces the message dissemination types.

Table 4.2: List of important messages

Message	Contents
<i>Beacon</i>	$\langle T, ID, R, IDcluster, L, v, Dir \rangle$
<i>CHA</i>	$\langle T, ID, R, IDcluster, L, v, Dir, SN, N, D \rangle$
<i>ReqJoin</i>	$\langle T, ID, R, IDcluster, L, v, Dir \rangle$
<i>ACKJoin</i>	$\langle T, ID, R, IDcluster, L, v, Dir, SN \rangle$
<i>ReqMerge</i>	$\langle T, ID, R, IDcluster, L, v, Dir, N, CML \rangle$
<i>ACKMerge</i>	$\langle T, ID, R, IDcluster, L, v, Dir, CML \rangle$

4.3 Performance Evaluation

In this section, we provide some insights into the operation of our proposed UFC approach. A detailed analysis of UFC approach will be presented by optimizing different parameters under various scenarios in the first part of the simulation. In the second part, we compare the performance of UFC to the simplest clustering algorithm Lowest-ID [32], as well as to the latest and most cited clustering algorithm VMaSC [7]. Since both the

Table 4.3: Message exchanging

Message	Source	Dissemination type
<i>Beacon</i>	all vehicles	broadcast
<i>CHA</i>	CH	broadcast
<i>ReqJoin</i>	CCM or CM	unicast towards CH
<i>ACKJoin</i>	CH	unicast towards CCM or CM
<i>ReqMerge</i>	CH	unicast towards CH
<i>ACKMerge</i>	CH	unicast towards CH

proposed algorithms and Lowest-ID are based on one-hop cluster, the one-hop VMaSC is implemented in our simulation. All of the schemes are implemented on NS2 [125]. The simulation configuration is described as follows.

4.3.1 Testing scenarios

We consider four testing scenarios, which are all generated by Simulation of Urban Mobility (SUMO) [77]. As far as we know, this is the first work in this field which evaluates the proposed algorithm in various traffic scenarios and provides the details of simulation setting. In all scenarios, there are 200 vehicles: 100 from east to west, and 100 from west to east. In each moving direction, there are two lanes. The length of the road is set to 10 km, which is equally divided into 8 segments. The traffic flow rate is set to 1500 vehicles per hour. The simulation runs for 600 seconds. The transmission range (TR) is 300 meters.

Table 4.4: Vehicle setting for scenario A.1

Max speed	Acceleration	Deceleration	Speed deviation
20 m/s	2.0 m/s^2	6.5 m/s^2	0.1

Table 4.5: Vehicle setting for scenario A.2

Type	Max speed	Acceleration	Deceleration	Speed deviation
1	20 m/s	2.9 m/s^2	7.5 m/s^2	0.7
2	20 m/s	2.9 m/s^2	7.5 m/s^2	0.3
3	20 m/s	2.0 m/s^2	6.5 m/s^2	0.1
4	20 m/s	1.5 m/s^2	5.5 m/s^2	0.3

These four testing scenarios consist of two relative stable traffic scenarios and two highly dynamic traffic scenarios, which are named as scenarios A.1, A.2, B.1, and B.2, respectively. There is only one vehicle type in scenario A.1, and there are four vehicle

Table 4.6: Vehicle setting for scenario B.1 and B.2

Type	Max speed	Acceleration	Deceleration	Speed deviation
1	35 m/s	2.9 m/s^2	7.5 m/s^2	0.7
2	25 m/s	2.9 m/s^2	7.5 m/s^2	0.3
3	20 m/s	2.0 m/s^2	6.5 m/s^2	0.1
4	10 m/s	1.5 m/s^2	5.5 m/s^2	0.3

Table 4.7: Testing scenario settings

Scenario	Vehicle type	Maximum Lane speed (m/s)
A.1	Table 4.4	8 segments (for each: 20)
A.2	Table 4.5	8 segments (for each: 20)
B.1	Table 4.6	8 segments (20,30,20,30,10,20,15,20)
B.2	Table 4.6	8 segments (20,15,25,30,25,20,15,20)

types in scenario A.2. The acceleration rate and deceleration rate are set according to the default values in SUMO. The maximal speed limit of the road is a constant in scenarios A.1 and A.2. In scenarios B.1 and B.2, there are four types of vehicles, and the maximal speed limit of each segment is different. The setting of vehicles in each scenario is shown in Tables 4.4, 4.5, 4.6, and the setting of each scenario is shown in Table 4.7. The mobility pattern of vehicles in scenarios B.1 and B.2 is more unpredictable than that in scenarios A.1 and A.2.

To better illustrate the differences of four traffic scenarios, we depict vehicles' speed distribution for each second from 300s to 500s, shown in Figure 4.3. It can be easily observed that vehicles' speed is more dynamic in Scenario B than Scenario A.

For each scenario, the simulation runs for 600s. The clustering process starts at time T_{start} , the time when all vehicles have entered the road. Vehicles establish CH/CM connections according to the clustering scheme. After the time T_{end} , all connections between CH/CM are automatically disconnected. T_{end} is the time which guarantees that $T_{end} - T_{start}$ is large enough, and most of vehicles are still on the road before T_{end} . In our simulation, we set $T_{start} = 300s$, and $T_{end} = 500s$. More simulation parameters and settings of MAC and PHY layers are illustrated in Table 4.8 and Table 4.9.

4.3.2 Performance metrics

Cluster stability could be defined from various aspects according to the implemented upper layer applications. In our simulation, we try to provide a detailed analysis of cluster performance from both macroscopic and microscopic levels, listed as follows. Macroscopic

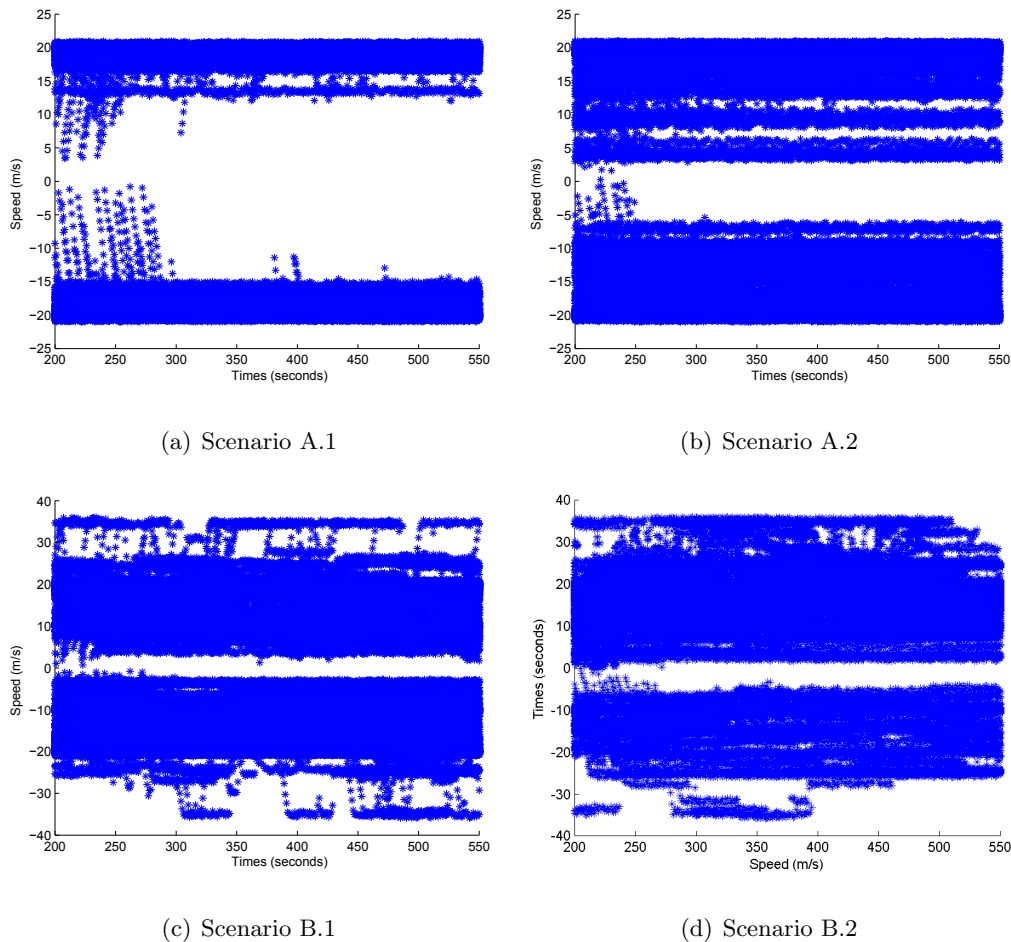


Figure 4.3: Vehicles' speed distribution

performance presents the overall cluster stability on the road. Microscopic performance shows vehicles' behaviors during the clustering procedure.

4.3.2.1 Macroscopic performance

- *Cluster head duration* presents the cluster's lifetime. It is the average time from a vehicle becoming a CH to giving up its state.
- *Cluster member duration* defines the average time from a node taking a CM state until changing to another state.
- *Number of clusters* defines how many clusters have been formed during the simulation period. A single CH without CMs also represents an independent cluster. Generally, an efficient clustering scheme prefers less formed clusters and more CMs in a single cluster.

Table 4.8: Default simulation parameters

Parameter	Value
Simulation time	200 <i>s</i>
T_{start}	300 <i>s</i>
T_{end}	500 <i>s</i>
Length of road	10 <i>km</i>
Number of vehicles	200
Transmission Range (TR)	300 <i>m</i>
Beacon Interval (BI)	1.0 <i>s</i>
<i>Beacon size</i>	66 bytes
<i>MAX_CM</i>	10
<i>MAX_BCH</i>	2
$T_{collect}$	3.0 <i>s</i>
T_{bch}	5.0 <i>s</i>
T_{ch}	5.0 <i>s</i>
T_{cm}	5.0 <i>s</i>
Δv_{th}	5.0 <i>m/s</i>
$\Delta \overline{V}_{Cth}$	10.0 <i>m/s</i>
T_{ack}	2.0 <i>s</i>
T_{merge}	5.0 <i>s</i>
Mobility model	Car-following model
Number of iterations	10

- **Clustering efficiency** is defined as the percentage of vehicles participating in clustering procedure during the simulation. A higher clustering efficiency means a better clustering performance.

4.3.2.2 Microscopic performance

- **Number of initial CHs** is the number of vehicles that are elected in the beginning of clustering procedure.
- **CM disconnection rate (per second)** illustrates the total number of link disconnections between CMs and their current CHs per unit time.
- **Average role change rate** presents the total number of state changes per second during the clustering procedure.

Table 4.9: Settings of PHY and MAC layer

Category	Parameter	Value
PHY	Propagation model	Two-Ray Ground
	Frequency/Channel Bandwidth	5.9GHz/10MHz
	Power Monitor Threshold	-174dBm
	Antenna	OmniAntenna
	Noise floor	-99dBm
	Carrier Sense Threshold	-94dBm
MAC	MAC protocol	IEEE 802.11p
	Retry Limit	7
	Header Length	40 μ s
	SlotTime/SIFS	13 μ s/32 μ s
	CWmin/CWmax	15/1023

- **CM re-clustering delay** is defined as the time interval of a CM from losing connection to successfully joining another cluster.
- **CM re-clustering success ratio** is defined as the percentage of successful CM reconnections after disconnections. A higher CM re-clustering success ratio guarantees the stability of the cluster.

4.3.3 Performance optimization of UFC

In Section 4.2, a new clustering approach has been introduced from the aspects of CH selection, cluster formation, and cluster maintenance. In this section, we will present how each method, proposed in UFC, influences the clustering performance.

4.3.3.1 Backoff timer T_W calculation

A Backoff-based CH selection method BCS has been presented in Section 4.2.3. Two potential methods for backoff time calculation are mentioned: a metric-based method and a random-based method. Three metrics have been proposed in the metric-based method, including link lifetime, relative distance, and relative speed. All vehicles calculate their individual T_W as soon as a system timer $T_{collect}$ expires. We compare these three metric-based methods and random-based method under the same context and analyze how T_W calculation method affects the cluster performance.

As has been addressed in Section 4.2.3, T_W is set to a value in $[T_{min}, T_{max}]$. We have tested the impact of T_{max} on the cluster performance, in which T_{max} is set to 0.5, 1.0,

2.0, 5.0, and 10.0s, respectively. According to the experimental results, we found that a larger T_{max} , for example when $T_{max} = 10.0s$, will cause longer cluster formation time, and a small T_{max} , for example when $T_{max} = 0.5$, will cause more packet collisions. Therefore, the value of T_{max} is fixed as 2 seconds, which shows the best performance in our simulation.

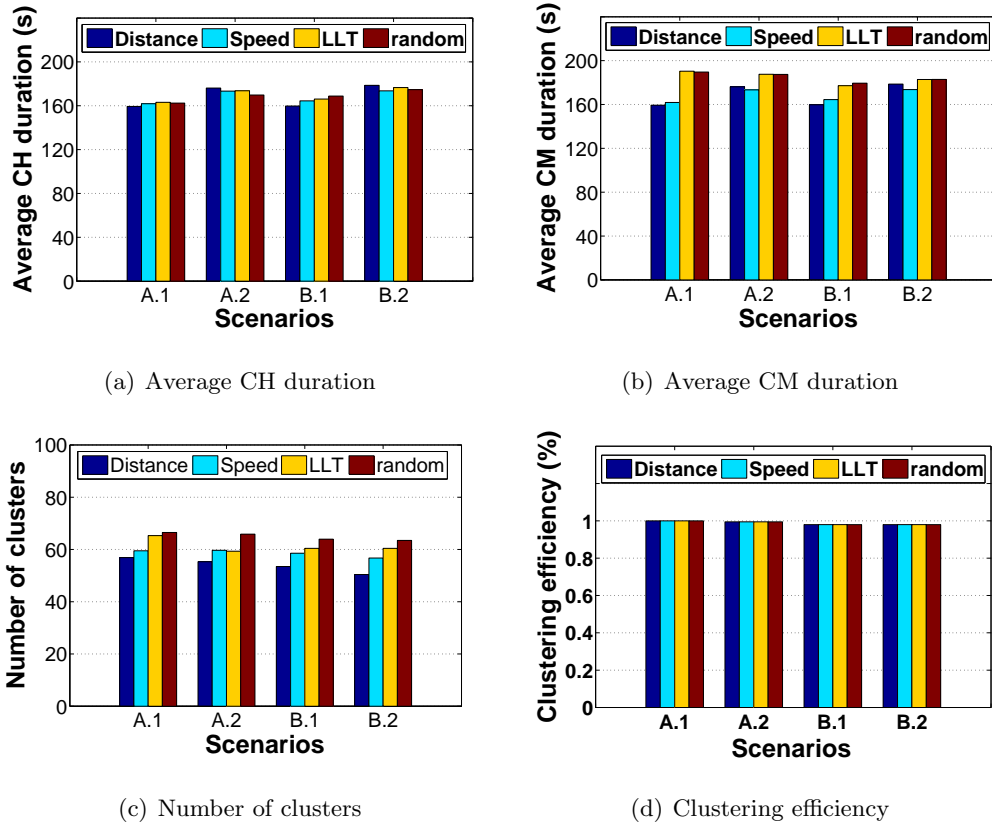


Figure 4.4: Impact of T_W calculation method on the macroscopic performance of UFC

Figure 4.4 compares the cluster macroscopic performance when applying different CH selection metrics. The comparison results of CH duration in Figure 4.4(a) shows small difference when implementing these four methods, and the CM duration is higher when implementing LLT-based and random-based methods, as shown in Figure 4.4(b). However, LLT-based and random-based methods create more clusters, comparing to distance-based method, shown in Figure 4.4(c). In Figure 4.4(d), clustering efficiency of these four methods are almost the same under each scenario. It means that the T_W calculation methods have no impact on the clustering efficiency. In addition, clustering efficiency slightly decreases when the traffic scenario becomes more dynamic (from scenario A.1 to B.2).

To further analyze the impacts of CH selection methods, we also compare the cluster microscopic performance, shown in Figure 4.5. Figure 4.5(a) shows that random-based method selects less initial CHs than other three methods. The results in Figure 4.5(b)

illustrate that the CH selection method does not have big impact on vehicle state change rate. In UFC, vehicle changes state from UN to CH or to CCM at beginning. Therefore, each vehicle changes state at least once during the simulation. Figure 4.5(c) presents CM disconnection rate. When traffic scenario becomes more dynamic, the CM disconnection rate decreases, and the differences of the CM disconnection rate among these four methods reduce. For example, under scenario B.2, the results are almost the same. We also notice that the value of random-based and speed-based method is always lower than that of LLT-based method under all scenarios. The comparison results of total CH change times in Figure 4.5(d) show that the performance of distance-based method is more sensitive to the setting of traffic scenarios, compared with other three methods.

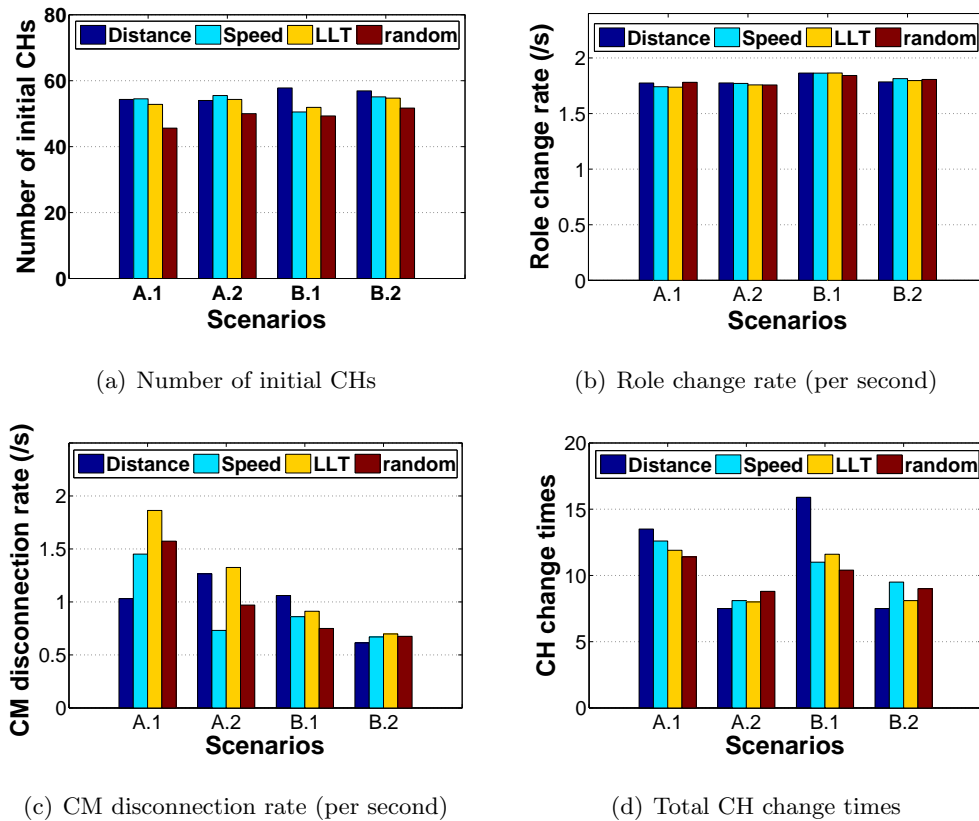


Figure 4.5: Impact of T_W calculation method on the microscopic performance of UFC

Therefore, from the results in Figure 4.4, we can conclude that LLT-based and random-based CH selection methods perform better in terms of the overall cluster stability with the price of creating more clusters.

4.3.3.2 Cluster capacity MAX_CM

Generally, cluster size is determined by vehicle's communication range in a single-hop clustering scheme. The increased vehicle density may cause high packet collision rate. Therefore, cluster size in UFC is controlled by a predetermined cluster capacity, i.e., the maximum number of CMs in the cluster, denoted by MAX_CM . In this section, we analyze the influence of cluster capacity on cluster performance by modifying MAX_CM under different scenarios. MAX_CM is set to 5, 10, 15, 20, and 200 vehicles per cluster, respectively. When MAX_CM is 200, it means that there is no limit on cluster capacity.

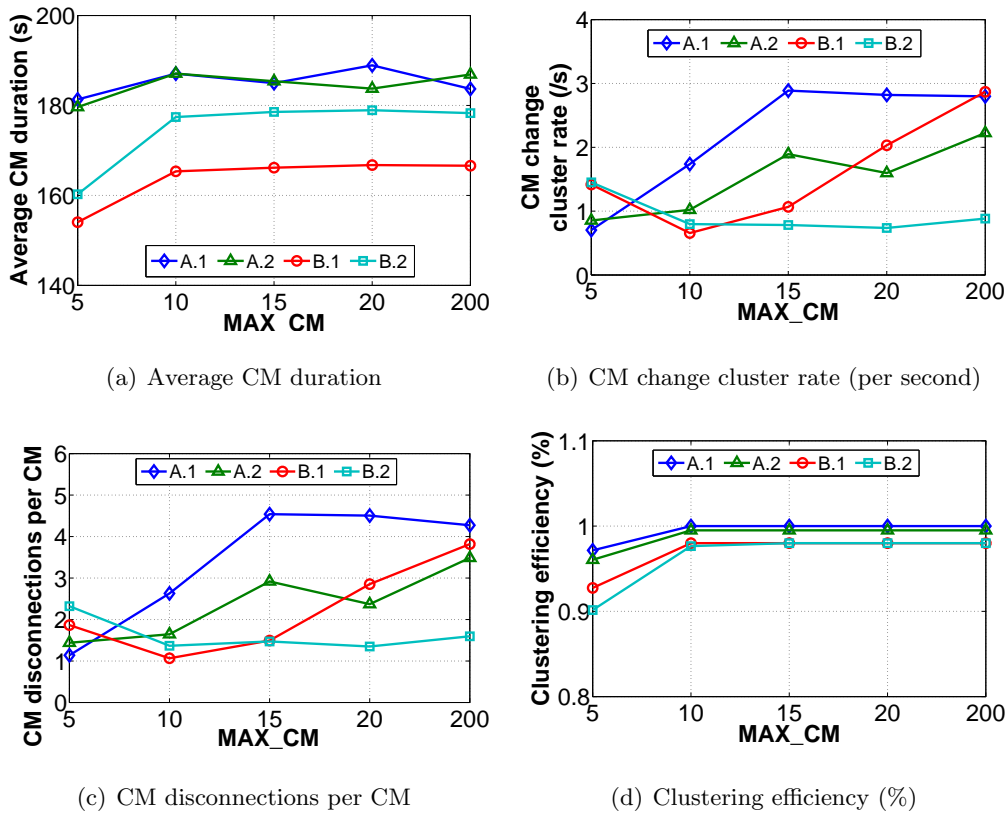


Figure 4.6: Impacts of cluster capacity (MAX_CM) on cluster performance of UFC

As shown in Figure 4.6(a), and Figure 4.6(d), CM duration and clustering efficiency have a sharp increase when MAX_CM changes from 5 to 10, and remain nearly stable with the increase of MAX_CM when MAX_CM is at least 10. It is because that when MAX_CM is too small, some original CMs are excluded from the new cluster due to the cluster capacity limit and become CHs during cluster merging. Such phenomenon reduces the mean CM duration. In addition, when MAX_CM is too small, some vehicles remain in the CCM state because their neighboring clusters reach the capacity limits. It reduces the clustering efficiency. When MAX_CM is larger than the average number of neighbors,

the above phenomenon seldom happens.

As shown in Figure 4.6(b), and Figure 4.6(c), CM change rate and the CM disconnection rate increase with the increase of MAX_CM for relative stable traffic scenarios (A.1 and A.2), and have a local optimal around $MAX_CM=10$ for dynamic traffic scenarios (B.1 and B.2). This is because, a cluster with more CMs has a higher probability to lose its CMs when compared to a cluster with less CMs. When the traffic is dynamic, cluster merging occurs frequently and causes frequent CM disconnections. In this case, when MAX_CM is too small, more CMs members will be excluded due to the cluster capacity limit during cluster merging, causing frequent CM disconnection.

In summary, we observe that, due to the comparison of different performance metrics, there is no single MAX_CM value that can optimize all performance metrics at the same time for different traffic scenarios. Nevertheless, when MAX_CM is 10, there is a good trade-off between these performance metrics. Therefore, in the rest part of the simulation, MAX_CM is fixed as 10.

4.3.3.3 Backup CH number MAX_BCH

Vehicle clusters are considered as a backbone structure during information dissemination, data aggregation, packet delivering, and etc. An unexpected vehicle's disconnection may result in losing an emergency message. In the proposed BUCH approach, our objective is to ensure that the disconnected vehicles can successfully join another existing cluster as soon as possible. Considering the management overhead caused by the Backup CH maintenance procedure in each CM and CCM vehicle, the number of BCHs maintained in backup CH list $BCHL$ should be carefully chosen to not only reduce re-clustering delay, but also avoid big overhead. The size of $BCHL$, denoted by MAX_BCH , is set to 1, 2, 3, 4, and 5, respectively. We evaluate the clustering performance in terms of CM re-clustering delay and CM re-clustering success ratio.

In Figure 4.7(a), CM re-clustering delay is always far less than 1.0s except when $MAX_BCH = 1$. When $MAX_BCH = 1$, it means that CM/CCM vehicle only caches the first qualified BCH it hears to its $BCHL$ without any prioritization. Therefore, this BCH may be unreliable and could not be the best choice for the disconnected vehicles, resulting in a lower CM re-clustering success ratio, revealed in Figure 4.7(b). When MAX_BCH is larger than 1, CM/CCM can prioritize the neighboring CHs it hears, and chooses the most reliable BCH as the target CH. Therefore, we can conclude that at least 2 BCH vehicles are required in order to both reduce the re-clustering delay and improve the

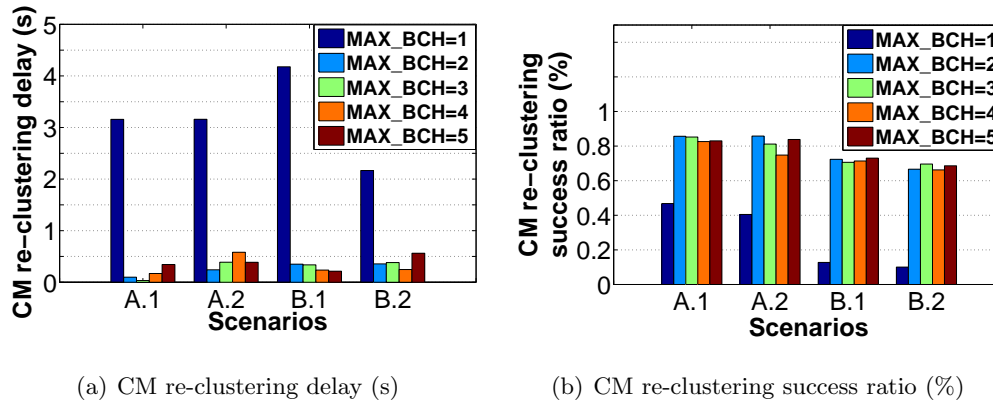


Figure 4.7: Impacts of MAX_BCH on CM re-clustering performance

re-clustering success ratio. Because there is no big difference in both the results in Figure 4.7(a) and in Figure 4.7(b) when MAX_BCH is larger than 1, and because recording redundant BCHs may cause a high $BCHL$ maintenance overhead and increase the CM re-clustering delay, MAX_BCH is fixed to 2 in the rest part of the simulation.

4.3.4 Performance comparison with VMaSC and Lowest-ID

In this section, we compare the cluster performance of the proposed UFC approach to Lowest-ID [32], and VMaSC [7] from both macroscopic and microscopic levels, under four traffic scenarios. Lowest-ID is one of the most famous clustering algorithms. According to the summary in [31], most proposed clustering algorithms in VANETs have chosen Lowest-ID as a benchmark to compare with. In addition, VMaSC is one of the latest and the most cited clustering algorithms, which provides detailed simulation parameter settings.

According to the definition of Lowest-ID [32], we consider the awareness of moving direction. "LID (all)" indicates the original Lowest-ID scheme which can cluster vehicles moving in all directions, while "LID (same dir)" indicates the optimized Lowest-ID scheme, in which only vehicles moving in the same direction can be clustered. In addition, the original VMaSC [7] clustering algorithm is a multi-hop based approach. Since both Lowest-ID and UFC are one-hop based clustering algorithm, we implement the one-hop VMaSC scheme in this study, denoted as "VMaSC_1hop".

We compare 2 sub-schemes of UFC, with or without Neighbor Sampling (NS) method, denoted as "UFC (w/o NS)" and "UFC (w/ NS)", respectively. When implementing NS method, vehicle only recruits stable neighbors, meeting the condition $\Delta v \leq \Delta v_{th}$. On the basis of the previous simulation results, the default MAX_CM and MAX_BCH are set

to 10 and 2, respectively. To make a fair comparison, we set the cluster updating interval to 5.0s in Lowest-ID scheme, which is same as T_{cm} in UFC and CH_TIMER in VMaSC scheme. The merging intervals are set to 5.0s both in UFC and VMaSC schemes. We repeat each simulation for 10 times. The nodes IDs in Lowest-ID are assigned randomly in each simulation.

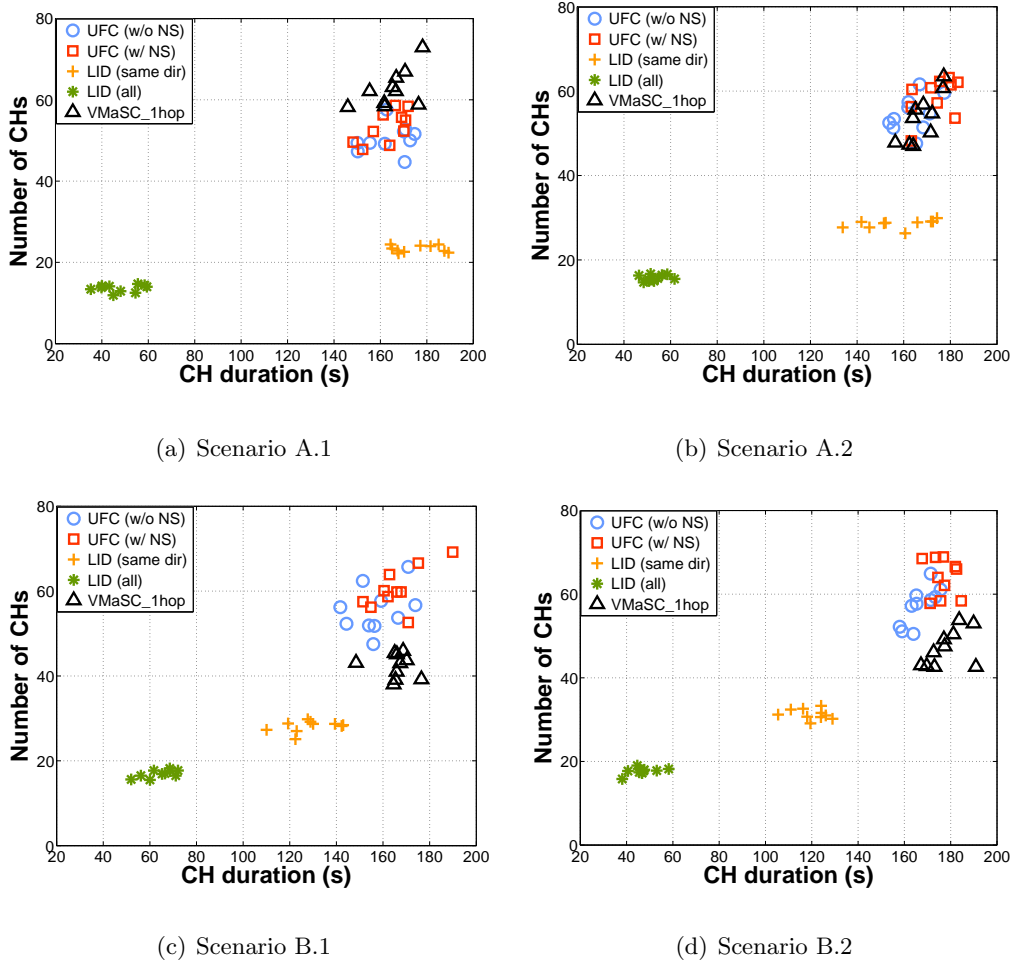


Figure 4.8: CH lifetime comparison between UFC, LID, and VMaSC

In Figure 4.8, we observe that LID (all) presents the minimum number of clusters and the smallest CH lifetime under all traffic scenarios. Lowest-ID scheme is a passive approach, in which, CM changes its CH as long as it hears a vehicle with a lower ID than its current CH; meanwhile, CH could become a CM as long as it hears a vehicle with a lower ID than itself. Therefore, LID (all) provides more chances for a CM to change its CH and for a CH to become a CM vehicle, since vehicles moving in opposite directions could also stay in the same cluster when they meet each other on the road. The frequent CH changing reduces CH lifetime, and the unlimited cluster size reduces the number of the created clusters. We also notice that under low dynamic scenario A.1, shown in Figure

4.8(a), LID (same dir) performs the best in terms of CH duration and number of clusters. However, when the traffic becomes more dynamic, especially in Figure 4.8(c) and Figure 4.8(d), the average CH duration of LID (same dir) decreases significantly and becomes smaller than that of two UFC sub-schemes and VMaSC_1hop scheme.

Table 4.10: Average CH lifetime (s)

Scenarios	Schemes				
	UFC		LID		VMaSC 1hop
	w/o NS	w/ NS	same dir	all	
A.1	163.943	163.112	175.556	47.92	164.875
A.2	163.814	173.645	156.969	53.164	167.801
B.1	157.386	166.112	128.589	64.2	165.779
B.2	166.626	176.57	119.717	46.762	178.243

Table 4.11: Average number of clusters

Scenarios	Schemes				
	UFC		LID		VMaSC 1hop
	w/o NS	w/ NS	same dir	all	
A.1	50.41	53.47	23.32	13.59	62.71
A.2	54.56	58.56	28.52	15.72	53.72
B.1	55.59	60.44	28.12	16.92	42.35
B.2	57.23	63.95	31.27	17.68	47.1

Furthermore, the results in Figure 4.8 presents the similar CH lifetime and number of clusters when implementing two UFC sub-schemes and VMaSC_1hop. Overall, UFC (w/ NS) performs better CH lifetime than UFC (w/o NS) scheme, especially in scenarios A.2, B.1, and B.2, which means that NS method is effective in increasing CH lifetime. Table 4.10 and Table 4.11 calculate the averaged CH lifetime and averaged number of clusters according to the results in Figure 4.8. In Table 4.10, we observe that the average CH lifetime of LID (all) is always the lowest one, about 65% lower than that of UFC schemes under all scenarios. Moreover, the average CH lifetime of UFC (w/ NS) remains the highest under scenario A.2 and B.1. Although VMaSC_1hop shows the highest average CH lifetime under scenario B.2, 178.243s, the value of UFC (w/ NS) is still the second highest one with a slight difference of 1.673s. However, the improvement of cluster stability

in VMaSC and UFC always cause the increase of the number of created clusters.

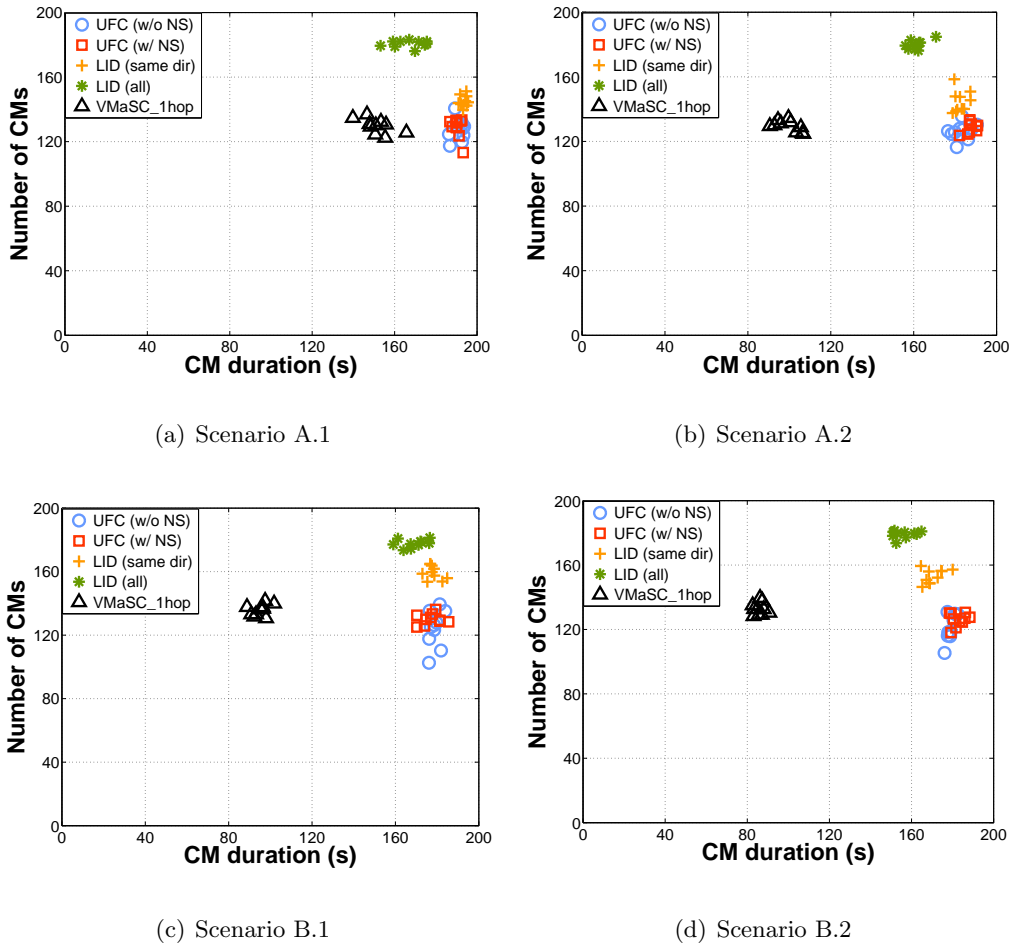


Figure 4.9: CM lifetime comparison between UFC, LID, and VMaSC

Figure 4.9 illustrates the CM lifetime and number of CMs. Apparently, LID (all) presents the maximum number of CMs in all scenarios. The reason is that vehicles moving in the opposite directions are allowed to stay in the same cluster and the cluster capacity is not limited in LID. Since the link connections between vehicles moving in the opposite directions are not stable, the CM lifetime of LID (all) is smaller than LID (same dir). LID (same dir) presents similar performance with two UFC sub-schemes under scenario A.1, A.2, and B.1, in terms of CM lifetime; and presents higher performance than UFC sub-schemes in terms of number of CMs under all scenarios. This is because the cluster capacity is not limited in LID. Furthermore, we are surprised by the performance of VMaSC_1hop in Figure 4.9, which shows good performance in Figure 4.8, but presents the lowest CM lifetime under all scenarios. We conjecture that the transitions between CM and unstable state node (denoted as SE in [7]) reduces the CM lifetime.

Table 4.12 and Table 4.13 calculates the averaged CM lifetime and averaged number

Table 4.12: Average CM lifetime (s)

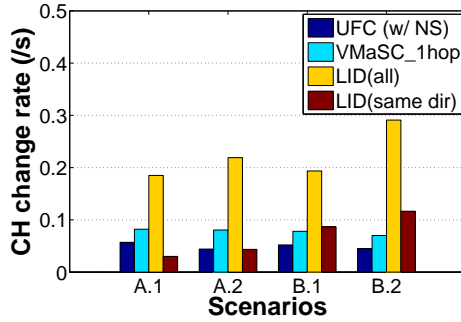
Scenarios	Schemes				
	UFC		LID		VMaSC
	w/o NS	w/ NS	same dir	all	1hop
A.1	191.065	190.395	192.032	166.986	151.436
A.2	182.959	187.582	182.484	161.418	99.604
B.1	177.129	178.963	178.107	168.706	95.073
B.2	179.313	182.796	170.514	156.194	85.897

Table 4.13: Average number of CMs

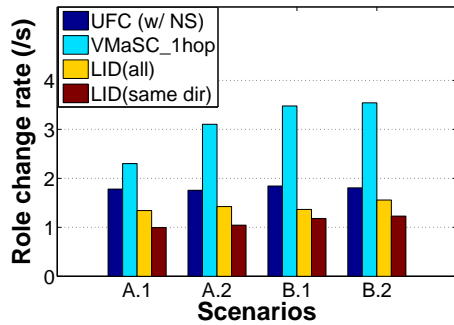
Scenarios	Schemes				
	UFC		LID		VMaSC
	w/o NS	w/ NS	same dir	all	1hop
A.1	127.89	128.73	145.01	180.63	129.72
A.2	126.33	128.74	144.65	179.82	129.46
B.1	124.3	130.06	159.43	177.68	135.81
B.2	121.96	126.09	153.2	179.05	132.74

of CMs based on the simulation results in Figure 4.9. Apparently, UFC (w/ NS) performs the highest cluster stability under scenario A.2, B.1, and B.2, as shown in Table 4.12. Even though the performance is worse than that of LID (same dir) and UFC (w/o NS) under scenario A.1, the tiny differences may be ignored, 0.67s with UFC (w/o NS) and 1.637s with LID (same dir). Besides, we observe that when the traffic scenario becomes more dynamic, the average CM lifetime decreases quickly in all schemes, except two UFC sub-schemes. We can conclude that UFC scheme is more robust to the change of traffic scenarios. Table 4.13 presents similar average number of CMs, except LID. LID (all) always performs the highest number of CMs under all scenarios.

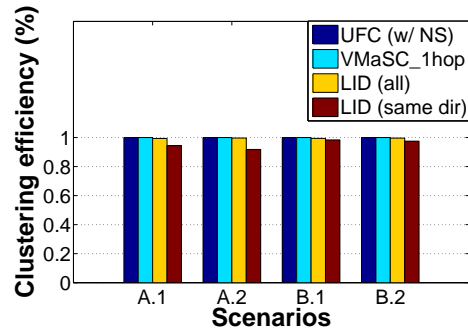
The simulation results in Figure 4.10 present the comparison between UFC (w/ NS), LID (same dir), LID (all), and VMaSC_1hop, in terms of CH change rate, role change rate, and clustering efficiency. In Figure 4.10(a), we observe that CH changes most frequently in LID (all) under all traffic scenarios, especially under B.2. Even though CH change rate of LID (same dir) is the lowest under scenario A.1, the value increases significantly and becomes larger than that of UFC (w/ NS) and VMaSC_1hop when traffic scenario becomes



(a) CH change rate (per second)



(b) Role change rate (per second)



(c) Clustering efficiency (%)

Figure 4.10: Cluster performance comparison between UFC and LID scheme

more dynamic. The results are consistent compared to the previous comparisons. On the contrary, the CH change rate of UFC (w/ NS) and VMaSC_1hop are both considered insensitive to the change of traffic scenarios, and UFC (w/ NS) always performs better than VMaSC_1hop.

In Figure 4.10(b), we observe that vehicles change the state more frequently when implementing UFC (w/ NS) and VMaSC_1hop, compared to both LID (all) and LID (same dir). Since LID scheme is a passive scheme and vehicles only change state based on their neighbors' identifiers, its role change rate is not accurate and not comparable with UFC (w/ NS) and VMaSC_1hop. In addition, the role change rate of VMaSC_1hop remains the highest under all scenarios, because many CMs change to unstable state as long as they lose the connections with their CHs. On the contrary, the role change rate of UFC (w/ NS) scheme remains stable while traffic scenario changes. This is because the proposed NS scheme guarantees more stable vehicle link connections and BUCH scheme allows CMs to find backup CHs instead of changing their state immediately.

Figure 4.10(c) illustrates the comparison results in terms of clustering efficiency. The clustering efficiency is always 100% with both UFC (w/ NS) and VMaSC_1hop approaches, under all traffic scenarios. This indicates that all vehicles on the road have participated

the clustering process. However, only part of vehicles are able to participate in clustering. LID (same dir) and LID (all) always miss some vehicles, which may lead to the clustering inaccuracy.

4.3.5 Summary of observations

From the results in Sections 4.3.3 and 4.3.4, we summarize the main observations as following.

- Importance of backup CH:** When there is no backup CH, the relative mobility based metric (speed based) does provide more stability, as reflected by CM disconnection rate in Figure 4.5(c). However, in a situation when there are at least 2 BCHs, it does not perform well compared to the link duration based metric (LLT based), and sometimes even to the random-based metric (reflected in Figure 4.4(a) and Figure 4.4(b), respectively). Nevertheless, the difference in CH duration and CM duration in such scenarios is just marginal. This illustrates the importance of backup CHs on the stability of clusters.
- Consistence of cluster stability provided by UFC:** The UFC clustering scheme performs better than both VMaSC scheme and the passive Lowest-ID clustering scheme, especially in terms of CH and CM lifetime, CM re-clustering delay, and vehicle state change rate. When the traffic scenario becomes dynamic, only the performance of UFC scheme shows stability, compared with other schemes. Therefore, we can conclude that our proposed UFC approach is robust enough and presents higher cluster stability, especially under high dynamic traffic scenarios, even though the trade-off is a higher number of created clusters.

4.4 UFC Analysis

4.4.1 Overhead analysis

Clustering overhead describes the total number of clustering related packets. In UFC, during the NS process ($T_{collect}$ time period), only *Beacon* messages are exchanged. When $T_{collect}$ expires, every vehicle calculates their own mobility metric and sets a timer T_W . The proposed Backoff-based CH Selection (BCS) scheme avoids broadcasting mobility metrics. Instead, only the selected CHs broadcast *CHA* messages (around 50 *CHA* messages according to the simulation results in Figure 4.5(a)). In the rest of the simulation, similar

to one-hop VMaSC, a vehicle sends a *ReqJoin* message to the CH when it wants to join a cluster. Once a CH confirms the join of this vehicle, it sends back an *ACKJoin*. In addition, during cluster merging, two kinds of messages have to be transmitted, *ReqMerge* and *ACKMerge*.

In one-hop VMaSC, besides the control messages (*ReqJoin*, *ACKJoin*, *ReqMerge* and *ACKMerge*), a CH advertisement message should be broadcast as long as a SE (State Election) [7] vehicle transits to CH. Thus, with time increasing, the total number of the broadcasted CH advertisement messages increases. We assume that, all vehicles broadcast *Beacon* messages for the same Beacon Interval (1.0s) in both UFC and one-hop VMaSC, and the total number of the control messages are almost the same. In this case, although the overhead of UFC is larger than one-hop VMaSC at the beginning of the simulation (*CHA* broadcasting), the overhead of one-hop VMaSC increases more rapidly than UFC when the simulation time increases. Moreover, since one-hop VMaSC presents a higher CH change rate and role change rate than UFC according to the simulation results in Figure 4.10(a) and 4.10(b) (around 1.5 times of UFC), the number of control messages will be higher than UFC.

4.4.2 UFC performance analysis

According the above simulation results, in general, the proposed UFC clustering algorithm with Neighbor Sampling performs better than other clustering algorithms, especially in terms of CH and CM lifetime, CM re-clustering delay, and vehicle state change rate.

- During the cluster formation process in UFC, only the Stable Neighbors which are moving in the same direction can be added in the same group. CH may change the state only when cluster merging happens. However, the original Lowest-ID, LID (all), can combine vehicles moving in the opposite directions, which greatly reduces the CH lifetime and increases the CH change rate. Moreover, since LID is a passive clustering algorithm, CH changes state as long as it hears another vehicle with lower ID.
- In UFC, instead of changing to an unstable state immediately, CM will remain in the state for a tiny period of time (0-0.4s, shown in Figure 4.7(a)) to find BCH when it loses connection with the current CH. The BUCH re-clustering scheme allows the disconnected CM to build up-link connection with an appropriate BCH as soon as possible, aiming at reducing the CM re-clustering delay and increasing CM lifetime.

On the contrary, in VMaSC, the disconnected CM will change state to State Election (SE) (unstable state) immediately, which decreases the CM lifetime.

In this work, the proposed UFC clustering algorithm has been tested under four traffic scenarios that are generated by SUMO, through modifying road conditions, vehicle velocity, speed acceleration, and speed deviation on one dimensional highway. The simulation results reveal that UFC shows superior performance comparing with LID and VMaSC algorithms. In the future work, we will verify the UFC performance under more practical settings, such as propagation model with fading and urban traffic scenarios with intersections, and re-design the CH selection metric accordingly.

4.5 Conclusion

In this chapter, we proposed a Unified Framework of Clustering approach, named UFC, in VANETs. This framework includes Neighbor Sampling (NS), Backoff-based Cluster head Selection (BCS), and BackUp Cluster Head based cluster maintenance (BUCH) schemes. NS scheme can filter out unstable neighbors in order to increase vehicle link stability. BCS scheme allows vehicles to make their own cluster head decisions in a distributed manner, which can reduce the clustering management overhead and save the cluster formation time. Moreover, BUCH scheme guarantees that the disconnected CMs could rebuild connections with other CHs as soon as possible, thus, effectively reduces the CM re-clustering delay.

To evaluate UFC's performance, we highlighted and well defined the performance metrics. Then, we compared UFC's clustering performance with the Lowest-ID and one-hop VMaSC algorithms under four different traffic scenarios generated by SUMO, and observed that UFC scheme performs better cluster stability than Lowest-ID and one-hop VMaSC, especially under high dynamic traffic scenarios. Meanwhile, UFC scheme shows steady performance under different traffic scenarios.

Although UFC shows good clustering performance in cluster stability, there have some limitations. The UFC has been tested under four traffic scenarios that are generated by SUMO; however, the simulation parameters, such as vehicle number, vehicle mobility model, and channel propagation model, are different under realistic scenarios. UFC needs to be double checked to verify its performance.

During our research work, we notice that cluster merging scheme may have potential impacts on clustering performance. Therefore, in the next chapter, we are interested in analyzing the impacts of different cluster merging schemes on clustering performance.

Impact of Merging Schemes on VANET Clusters Stability

5.1 Introduction

Through the observation of the existing clustering algorithms, it is easy to notice that all of the clustering algorithms incline to form a small number clusters with large cluster size. When two clusters approach one another, they intend to merge to form a single larger cluster. Then a new CH will be selected for the merging cluster. The process of combining two neighboring clusters into a single larger cluster is defined as a cluster merging process. An effective cluster merging process is always indispensable during clustering, to guarantee fewer vehicle disconnections and re-connections.

In Chapter 4, we have proposed a Unified Framework of Clustering approach (UFC) in VANETs. During the algorithm implementation, we have observed that it is easy to ignore the way of cluster merging. Moreover, in the literature, there is a lack of comparison of cluster merging schemes, which makes it hard to analyze the impact of this component on clustering performance. Therefore, in this chapter, we investigate the existing cluster merging schemes and propose a Leadership-based Cluster Merging (LCM) scheme based on our proposed clustering framework UFC. Then, a comprehensive comparison of different cluster merging schemes is given, and LCM is shown to achieve better performance on cluster stability.

The rest of this chapter is organized as follows: Section 5.2 discusses the related work of cluster merging approaches. Section 5.3 describes the proposed Leadership-based cluster merging scheme. Then, Section 5.5 introduces the simulation environment and provides a comprehensive comparison of the cluster performance of different cluster merging schemes.

Section 5.6 concludes this chapter.

5.2 Related Work

According to the literature, cluster merging process is always triggered when two CHs approach one another and become one-hop vicinities. In order to guarantee the stability of the merged cluster and to decrease vehicle re-clustering frequency, most existing cluster merging schemes require two neighboring CHs to stay in the transmission range (TR) of each other for a short time period, which is defined as contention time or merge interval (MI), instead of starting cluster merging immediately.

According to the research, there are two common strategies to select the new CH in the merging cluster. The first one is to select the CH that is attached with more CMs, denoted as “CM-based”, as adopted by [79], [45], and [58]. In [79], cluster merging takes place when two CHs come within each other’s transmission range, and their speed difference is within the predefined threshold Δv_{th} . The CH that has a lower number of CMs simply gives up the CH role and becomes a CM in the new cluster. The rest CMs automatically join the neighboring cluster if they are in the transmission range of the CH and the speed difference is within the threshold. Similar to [79], in [45], when cluster merging happens, the cluster with fewer CMs is dismissed, and these CMs try to join other clusters, launching a new CM re-clustering stage. The “CM-based” strategy aims to reduce cluster member disconnections. However, such a strategy cannot guarantee the stability of link connections between the new CH and its members.

The second strategy is to select the CH that has better stability within its original cluster, as adopted by [7], denoted as “VMaSC-based”. During cluster merging, two CHs compare their averaged relative speed, called AVGREL.SPEED in their original clusters, respectively. The CH with higher average relative speed gives up its CH role and affiliates to the CH with lower average relative speed as a CM. Similar to [7], both the cluster merging schemes proposed in [42] and [80] select the new CH according to predefined vehicles’ stability metrics (Aggregated Local Mobility (ALM) in [42] and Benefit Factor (BF) in [80]) in the original clusters. The intuition of this strategy is assuming that the CH’s stability in its original cluster is representative for its stability in the newly merged cluster. However, in reality, a higher stability in the original cluster cannot guarantee a better cluster performance in the merged cluster.

In this chapter, we propose a Leadership-based Cluster Merging (LCM) scheme. The term “leadership” indicates the capability of coordination with CMs. The main idea is to

assign the vehicle that has a better leadership to be the new cluster head of the newly merged cluster, to ensure that it can maintain stable connections with all members of the cluster. LCM scheme will select this cluster head, which not only has stable links with members in the original cluster but also has stable links with other members in another cluster to be merged. The strategy is denoted as a leadership-based cluster merging scheme, as the selected vehicle provides better coordination of the merged cluster.

5.3 Description of Cluster Merging Scheme

Based on the clustering framework which has been described in Chapter 4, we only focus on describing the cluster merging scheme in this section. Cluster merging process will be presented from two parts, merging condition check and new CH selection.

Assuming that clusters are well constructed according to the UFC clustering process, and Figure 5.1 presents an example of cluster network topology. Two clusters are presented on the road, with a single CH in each cluster, CH_f and CH_b .

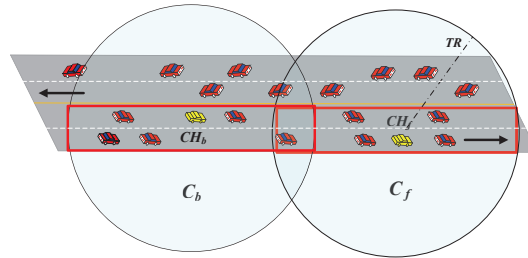


Figure 5.1: Cluster C_f and C_b .

5.3.1 Merging condition check

With time increasing, more and more single CMs may change state to CHs because of the CM-disconnections, which makes clustering meaningless. When two neighboring CHs are moving in the same direction within the transmission range of each other, cluster merging detection process will be triggered. Assuming that there are two clusters on the road, cluster C_1 and cluster C_2 with two CHs respectively, CH_1 and CH_2 (1 and 2 are node identifiers, and the ID of cluster $CHID$ is represented by the ID of its CH). As shown in Figure 5.1, the CH moving in front is denoted as CH_f , and the CH moving in the back is denoted as CH_b (b and f indicate the relative position of CH). Cluster merging detection process is triggered as long as CH_b receives a *Beacon* message from CH_f . CH_b will start a contention timer, called Merge Interval (MI), in order to avoid frequent CM re-clustering.

CH_b will check the merging condition only if it can receive the *Beacons* consecutively from CH_f during MI time period. Otherwise, these two clusters cannot be merged.

When MI expires, CH_b will check the merging conditions, listed as follows. If all of the conditions are satisfied, CH_b will send a *ReqMerge* message to CH_f .

- Two clusters are moving in the same direction;
- The number of CMs in the merged cluster is less than the predetermined cluster size MAX_CM , the value of MAX_CM will be addressed in Section 5.5;
- The difference between the mean relative speed of two clusters should satisfy $\Delta\overline{V}_{Cfb} \leq \Delta\overline{V}_{Cth}$. The mean relative speed of the cluster is described as the averaged speed difference between a CH and all of its CMs, shown in Eq. (5.1) and Eq. (5.2).

$$\Delta\overline{V}_{Cfb} = |\overline{V}_{Cf} - \overline{V}_{Cb}| \quad (5.1)$$

$$\overline{V}_{Cf} = \frac{\sum_{k \in C_f} |\vec{v}_f - \vec{v}_k|}{N_f}, \quad (5.2)$$

where \overline{V}_{Cf} is the average speed difference between CH_f and CMs of cluster C_f , \vec{v}_f and \vec{v}_k indicate the speed of CH_f and CM_k respectively, vehicle k is one of the CMs in cluster C_f , and N_f is the number of CMs in C_f .

Note that only clusters that have the same moving direction can be merged, in order to avoid frequent link disconnections between vehicles moving in the opposite directions. A predetermined cluster size MAX_CM is applied, indicating the maximum number of CMs in a cluster, to avoid overloading cluster's resource capacity, as well as to increase the cluster stability.

5.3.2 Leadership-based CH_{merge} selection

Upon the reception of *ReqMerge* message from CH_b , a leader in the newly merged cluster, denoted as CH_{merge} , should be selected by CH_f . The node with higher leadership is considered to have better link stability with neighboring members and will be selected as CH_{merge} . The definition of "leadership" is described as follows.

We define the stability factor $SF_{m,Cn}$ as the stability between vehicle m and all of the CMs in C_n , detailed in Eq. (5.3). It is represented by the averaged speed difference between a vehicle m and all of the CMs in cluster C_n (m and n are vehicles' identifiers

and can be the same).

$$SF_{m,C_n} = \frac{\sum_{k \in C_n} |\vec{v}_m - \vec{v}_k|}{N_n}, \quad (5.3)$$

where vehicle k is one of the CMs of cluster C_n and N_n is the number of CMs in C_n . A smaller SF_{m,C_n} indicates that the mobility pattern of vehicle m is more similar to the vehicles in the cluster C_n . Then, we denote the leadership of a vehicle m in cluster C_n as L_{m,C_n} , normalized between 0 and 1, shown in Eq. (5.4). A higher leadership represents a better link stability between vehicle m and the CMs in cluster C_n .

$$L_{m,C_n} = \frac{1}{1 + SF_{m,C_n}}. \quad (5.4)$$

Algorithm 10 Leadership-based merging CH selection

For cluster head moving in front CH_f , on receiving a *ReqMerge* message from CH_b :

CH_f calculates L_{f,C_f} and L_{b,C_f} , according to Eq. (5.3) and Eq. (5.4)

if $L_{b,C_f} > L_{f,C_f}$ then

CH_f sends back a *ACKMerge* message to CH_b , with $flag_CH_{merge} = 1$

CH_f broadcasts a *Change_CHID* to inform its current CMs to change their corresponding *CHID* to the ID of CH_b

CH_f changes its state to CM and changes its corresponding *CHID* to the ID of CH_b

else

CH_f sends back a *ACKMerge* message to CH_b , with $flag_CH_{merge} = 0$

CH_f becomes the CH_{merge}

end if

For cluster head moving back CH_b , on receiving a *ACKMerge* message from CH_f :

if $flag_CH_{merge} = 1$ then

CH_b becomes the CH_{merge}

else

if $flag_CH_{merge} = 0$ then

CH_b broadcasts a *Change_CHID* message to inform its current CMs to change their corresponding *CHID* to the ID of CH_f

CH_b changes its state to CM and changes its corresponding *CHID* to the ID of CH_f

end if

end if

For cluster member CM_i , on receiving a *Change_CHID* message from its current CH, CH_f or CH_b :

CM_i extracts the new CH identifier and changes its corresponding *CHID*

The details of the leadership-based merging CH selection process is described in Algorithm 10. Upon the reception of *ReqMerge* message from CH_b , the forward CH CH_f computes its leadership L_{f,C_f} and the leadership of CH_b in cluster C_f , denoted as L_{b,C_f} , according to Eq. (5.3) and Eq. (5.4). If the leadership of CH_b in the cluster C_f is higher than that of CH_f , where $L_{b,C_f} > L_{f,C_f}$, CH_b is more intuitive to be the leader becomes CH_{merge} . Otherwise, the forward CH_f keeps on being the leader and claims itself as CH_{merge} .

As long as the new CH is selected, CH_f sends back a *ACKMerge* message to CH_b , informing CH_b the information of CH_{merge} by *flag-CH_{merge}*. If CH_f keeps on being the leader, *flag-CH_{merge}* = 0, and CH_b broadcasts a message to inform its original CMs to change the identifier of CH, *CHID*. Meanwhile, CH_b becomes a CM of CH_f . On the contrary, if CH_b becomes the merging CH, where *flag-CH_{merge}* = 1, CH_f broadcasts a message to inform its original CMs to change their *CHID*, and itself becomes a CM of the merging cluster.

5.4 Scheme Analysis

In this section, we present a general model to analyze the rationality of both the existing and our proposed merging schemes. Without loss of generality, we consider the cluster C_1 is merged with the cluster C_2 , while the merging condition is already satisfied. We assume that the new CH after merging is one of the two original CHs (1 or 2), and it associates with all CMs in C_1 and C_2 , and another original CH.

We denote the current time when cluster merging happens as t_0 . $D^-(m, C_m)$ indicates the average link duration between CMs in cluster C_m and their CH m before t_0 , where $m = 1$ or 2 . Furthermore, we denote $D^+(n, C_m)$ as the average link duration between CMs in cluster C_m and the external CH n after merging, counting from the moment t_0 , where $(m, n) \in \{1, 2\} \times \{1, 2\}$. The superscript “-” and “+” represents the time before and after t_0 , respectively.

We denote \tilde{D} as the average CM duration of vehicles in these two clusters around the moment t_0 . Without loss of generality, we consider that the vehicle 1 is the new CH after merging. The CMs in C_1 originally have the association with the vehicle 1 uninterruptedly. Their link duration is $D^-(1, C_1) + D^+(1, C_1)$ on average. The CMs in C_2 originally break their association with the vehicle 2 at t_0 , and associate with the vehicle 1 afterward. Therefore these CMs have two periods of CM durations, which are $D^-(2, C_2)$ and $D^+(1, C_2)$ on average respectively. The vehicle 2 associates with the vehicle 1 only

after t_0 and its mean CM duration is $D^+(1, C_2)$. Since the total number of CM/CH connections around t_0 for vehicles in these two cluster is $N_1 + 2N_2 + 1$, the average CM duration of them, denoted as \tilde{D} in such a case is

$$\tilde{D} = \frac{[D^-(1, C_1) + D^+(1, C_1)]N_1 + D^-(2, C_2)N_2 + D^+(1, C_2)N_2 + D^+(1, C_2)}{N_1 + 2N_2 + 1}. \quad (5.5)$$

Following the same procedure, we can compute \tilde{D} in the case that the vehicle 2 is the new CH. Let δ be an binary variable to indicate whether node 1 or node 2 is the new CH, which is defined as

$$\delta = \begin{cases} 1, & \text{if vehicle 1 is the new CH} \\ 0, & \text{if vehicle 2 is the new CH.} \end{cases} \quad (5.6)$$

Then $\tilde{D}(\delta)$ can be generalized as a function of the CH selection decision, which is

$$\tilde{D}(\delta) = \frac{\alpha + \beta}{\gamma}, \quad (5.7)$$

where α and β are the sum of CM duration before and after t_0 , and γ is the total number of CM/CH connections around t_0 for vehicles in these two clusters. They are:

$$\alpha = N_1 D^-(1, C_1) + N_2 D^-(2, C_2), \quad (5.8)$$

$$\begin{aligned} \beta = & \delta N_1 D^+(1, C_1) + (1 - \delta)(N_1 + 1)D^+(2, C_1) \\ & + \delta(N_2 + 1)D^+(1, C_2) + (1 - \delta)N_2 D^+(2, C_2), \end{aligned} \quad (5.9)$$

$$\gamma = N_1 + N_2 + (1 - \delta)N_1 + \delta N_2 + 1. \quad (5.10)$$

When $\tilde{D}(1) - \tilde{D}(0) > 0$ (or ≤ 0), the vehicle 1 (or the vehicle 2) should be selected as the new CH, as that leads to a larger mean CM duration. We will show in the following that, different cluster merging schemes follows such rationality under different assumptions.

Under the assumption that $D^-(m, C_m)$ and $D^+(n, C_m)$ are equal to a constant for $m \in \{1, 2\}$ and $n \in \{1, 2\}$, the sign of $\tilde{D}(1) - \tilde{D}(0)$ is identical with $N_1 - N_2$. This is the intuition of the ‘‘CM-based’’ cluster merging schemes, such as [79], [45], and [58].

Under the assumption that N_1 is similar to N_2 , and $D^+(2, C_1)$ is similar to $D^+(1, C_2)$, the sign of $\tilde{D}(1) - \tilde{D}(0)$ is identical with $D^+(1, C_1) - D^+(2, C_2)$. This is the intuition of the ‘‘VMaSC-based’’ scheme in [7].

Our proposed LCM scheme is under the assumption that N_1 is similar to N_2 , and the sign of $D^+(2, C_1) - D^+(1, C_1)$ is identical with $D^+(1, C_2) - D^+(2, C_2)$. The intuition is that there is an exact one-one matching between CHs and clusters on the similarity in mobility pattern. Without loss of generality, we consider the vehicle 1 is the CH f in the front, and the vehicle 2 is the CH b in the back. Then the sign of $\tilde{D}(1) - \tilde{D}(0)$ is identical

with $D^+(f, C_f) - D^+(b, C_f)$. Notice that the leadership L_{m, C_n} is positively correlated with $D^+(m, C_n)$. Then the sign of $\tilde{D}(1) - \tilde{D}(0)$ is identical with $L_{f, C_f} - L_{b, C_f}$ calculated at the front vehicle f . Compared with “CM-based” and “VMaSC-based” schemes, our scheme does not require the assumption of equality between CM durations, which is more flexible.

5.5 Performance Evaluation

In this section, we provide some insights into three cluster merging schemes: the CM-based scheme; the VMaSC-based scheme; and the proposed leadership-based cluster merging (LCM) scheme, denoted as “Leadership-based”. In the simulation, three cluster merging schemes are implemented on UFC framework independently, and a detailed cluster performance comparison is presented.

All of the schemes are implemented on the Network Simulator NS2 [125]. Cluster Merge Interval (MI) is set to 2, 5, 10, 20, and 50 seconds respectively. Each simulation is repeated for 10 times and calculates the average value. The simulation configuration is described as follows.

5.5.1 Testing scenarios

In the simulation, four testing scenarios are generated by Simulation of Urban Mobility (SUMO) [77], aiming to observe the cluster performance under different traffic scenarios. In all scenarios, there are 200 vehicles: 100 from East to West, and 100 from West to East. Each moving direction has two lanes. The length of the road is set to 10 km, which is equally divided into 8 segments. Vehicles are generated with a certain traffic generation rate, 1500 vehicles per hour. The simulation runs for 600 seconds. The transmission range is $TR=300$ meters. These four testing scenarios are exactly the same to the scenarios which have been defined in Section 4.3.1.

The clustering process starts at time T_{start} , the time when all vehicles have entered the road. Vehicles are possible to establish CH/CM connections according to the clustering scheme. After the time T_{end} , all connections between CH/CM are automatically disconnected. T_{end} is the time which guarantees that $T_{end} - T_{start}$ is large enough, and most of vehicles are still on the road before T_{end} . In our simulation, we set $T_{start} = 350s$, and $T_{end} = 550s$.

In [126], according to the Canton of Zurich scenario, about 50% of vehicles have no more than 10 neighboring vehicles, and nearly 95% vehicles have less than 60 neighboring

vehicles. Therefore, in this simulation, the maximum number of vehicles in each cluster is set to 10, where $MAX_CM = 10$. Other simulation parameters are illustrated in Table 5.1.

Table 5.1: Default simulation parameters

Parameter	Value
Simulation time	200 <i>s</i>
T_{start}	350 <i>s</i>
T_{end}	550 <i>s</i>
Length of road	10 <i>km</i>
Number of vehicles	200
Transmission Range (TR)	300 <i>m</i>
Beacon Interval (BI)	1.0 <i>s</i>
Merge Interval (MI)	2, 5, 10, 20, 50 <i>s</i>
MAX_CM	10
T_W	2.0 <i>s</i>
$T_{collect}$	3.0 <i>s</i>
Δv_{th}	5.0 <i>m/s</i>
$\Delta \overline{V}_{Cth}$	10.0 <i>m/s</i>
T_{ACK}	2.0 <i>s</i>
Mobility model	Car-following model
Propagation model	Two-Ray Ground
MAC protocol	IEEE 802.11p
Frequency/Channel Bandwidth	5.9GHz/10MHz
Number of iterations	10

5.5.2 Performance metrics

We study the performance of clustering schemes from both macroscopic and microscopic aspects through the simulation. The macroscopic performance presents the overall cluster stability, and the microscopic performance shows vehicles' behaviors during the clustering procedure. The detail of the definition of performance in these two aspects is as follows:

5.5.2.1 Macroscopic performance

- *Cluster head duration* presents the cluster's lifetime. It is the average time interval from a vehicle becoming a CH to giving up its state. In general, a longer CH lifetime means a more stable cluster.
- *Cluster member duration* defines the average time interval from a node joining an existing cluster as a member in CM state to becoming another state.

5.5.2.2 Microscopic performance

- *Average role change rate (per second)* presents the total number of vehicles' state changes in one second.
- *Cluster member disconnection rate (per second)* describes the total number of disconnections between CMs and their CHs per second.

Summarizing both macroscopic and microscopic performance, a good clustering scheme requires not only higher mean CH lifetime and CM lifetime, but also lower mean role change rate and CM disconnection rate.

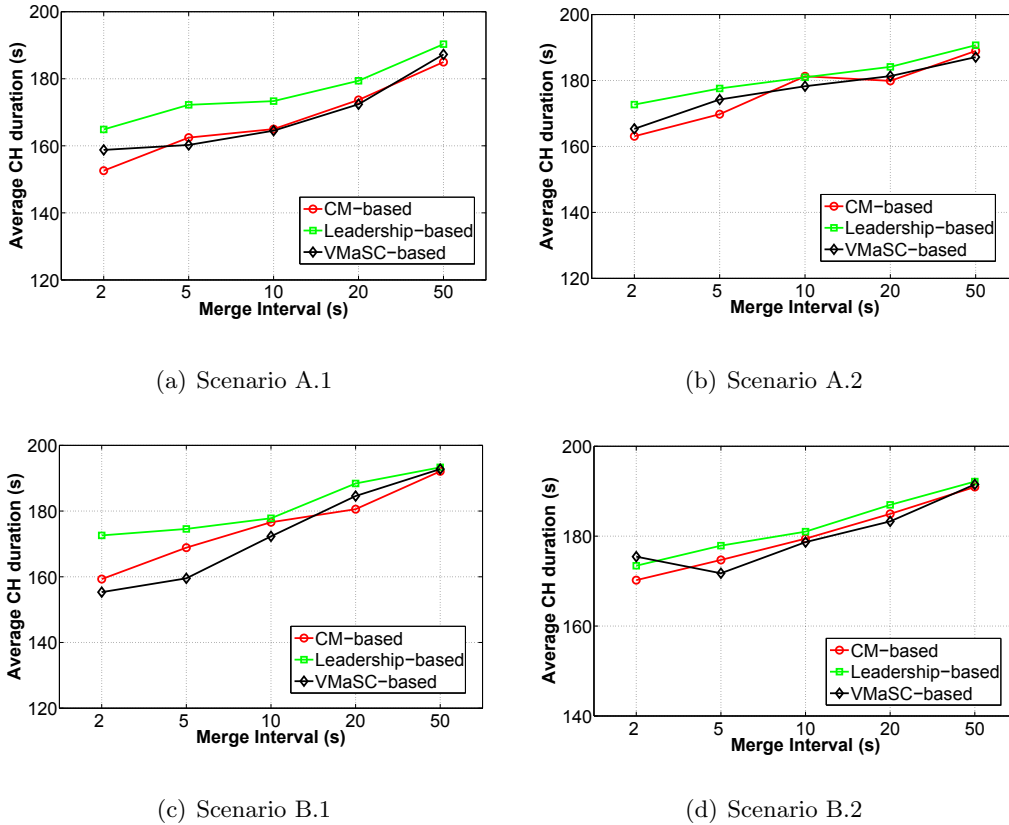


Figure 5.2: Impacts of MI on CH lifetime.

5.5.3 Performance analysis

The macroscopic performance, i.e., the mean CH lifetime and mean CM lifetime, versus cluster Merge Interval (MI), is shown in Figure 5.2 and Figure 5.3. In the *A* series scenarios, the MLS of vehicles are the same, while that is not a constant in the *B* series scenarios. Therefore, the cluster size is similar in the *A* series scenarios as there is no traffic congestions. Since the traffic mobility is quite unpredictable in the *B* series scenarios, the mean link duration between CMs and different neighboring CHs can be regarded as more or less similar.

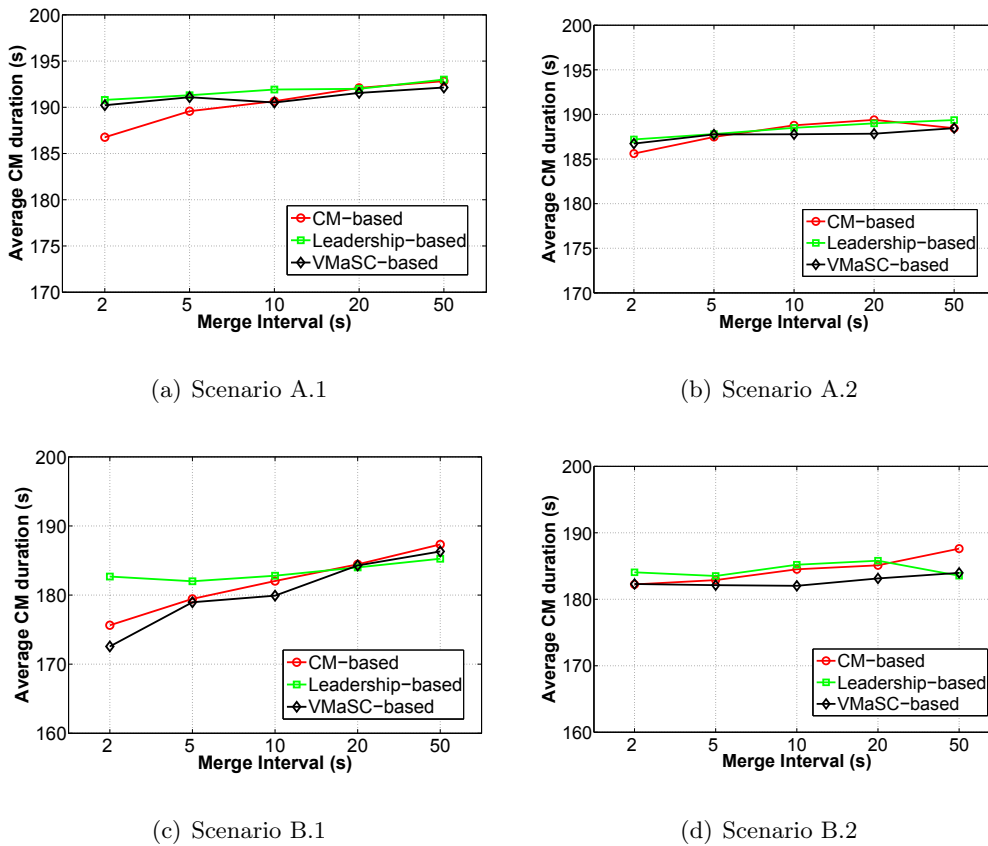


Figure 5.3: Impacts of MI on CM lifetime.

According to our analysis in Section 5.4, the CM-based scheme is more suitable in the *A* series scenarios, and the VMaSC-based scheme is more suitable for the *B* series scenarios, which is confirmed by the simulation results. When MI increases, the mobility pattern of the two clusters that pass the merging condition check becomes more and more similar. Therefore the CM-based scheme outperforms the VMaSC-based scheme when MI is sufficiently large. Comparing with the two benchmarks, the Leadership-based scheme overall achieves the best performance, in regardless of traffic scenarios and the setting of MI, as it has a better prediction on the stability between CMs and potential CHs.

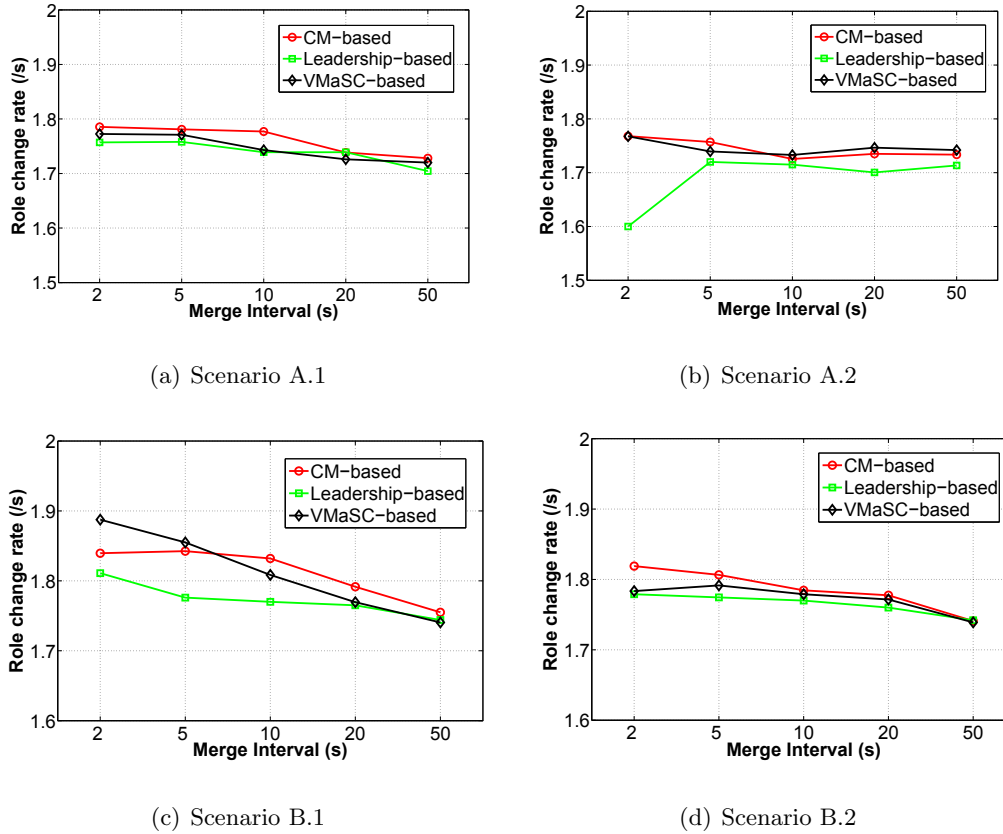


Figure 5.4: Impacts of MI on the average role change rate.

The microscopic performance, i.e. the role change rate, and the CM disconnection rate, is shown in Figure 5.4 and Figure 5.5. Since the Leadership-based scheme targets to ensure a stable cluster after cluster merging, it achieves the lowest role change rate and lowest CM disconnection rate by avoiding frequent cluster splitting and CM/CH de-association. Its advantage over VMaSc-based scheme shows that the stability between CMs and different CHs provide more information than the stability in the current cluster. The CM-based scheme, on the contrary, shows poor performance especially for the CM disconnection rate, as it does not take into consideration of the stability of the merged cluster.

We can observe that both macroscopic and microscopic performance are improved with the increase of MI. Nevertheless, it is not desirable to always utilize a large MI, as this may result in many small sized clusters, which degrades the efficiency of clustering in VANETs for data communication. The simulation results reveal that the proposed Leadership-based scheme is less sensitive to the change of MI. Therefore, LCM is able to ensure both good macroscopic and microscopic performance and high clustering efficiency at the same time.

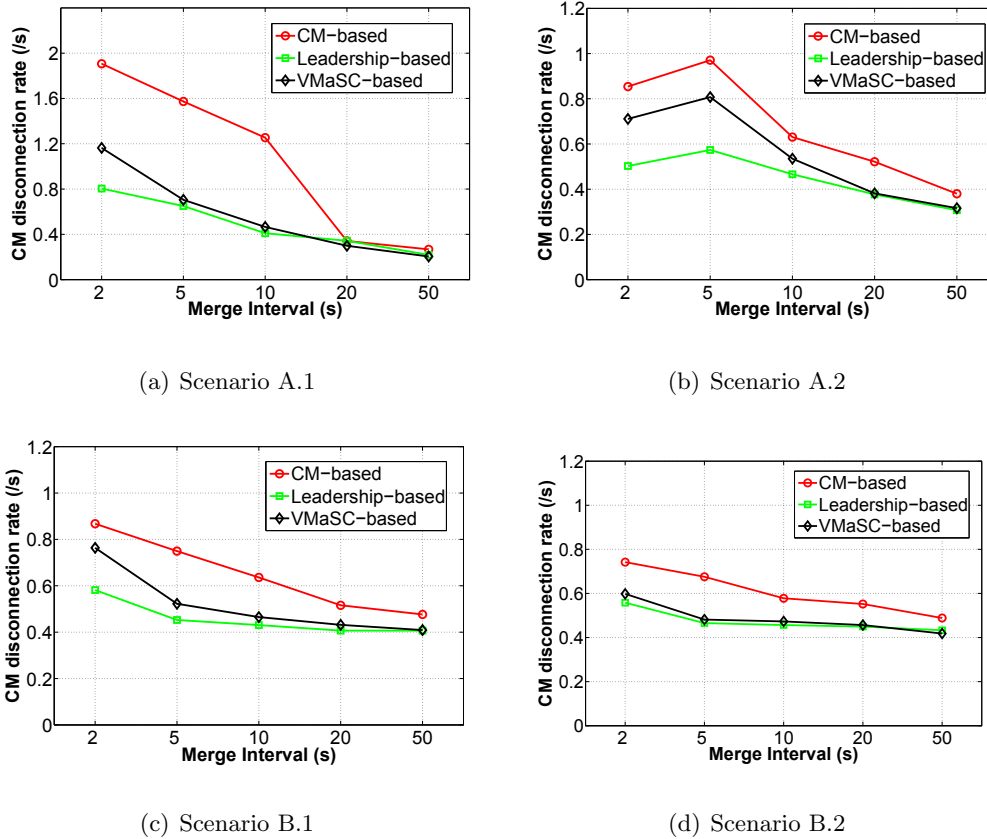


Figure 5.5: Impacts of MI on CM disconnection rate.

5.6 Conclusion

In this chapter, we investigate the impact of cluster merging process on cluster stability. A Leadership-based Cluster Merging (LCM) scheme is proposed and is compared with other cluster merging methods based on the same framework UFC. The LCM scheme assigns the vehicle that has better link stability with all members in the newly merged cluster to be the cluster head. The intuition of LCM is to have a better perception of new events and better cluster stability, to be compatible with popular applications such as platooning in vehicular networks. The simulation results conducted by NS2 and SUMO show that LCM achieves better cluster stability, compared with VMaSC-based, and CM-based cluster merging schemes. To deep understand the advantage of LCM, we also present a general model to analyze the rationality of both the existing and our proposed schemes.

According to our work, we have proved that cluster merging scheme does influence cluster stability, although it may seem like a small part of the clustering process. The proposed LCM scheme shows better performance in enhancing the cluster stability, but we still consider that the linear highway-style scenario is a little simplistic. It would

be interesting to evaluate the performance in a more complex environment, such as a Manhattan-style grid road network.

In this work, we have presented a general model to analyze the rationality of cluster merging schemes. To further understand the entire clustering process, we propose a stochastic model in the next chapter.

A Stochastic Model for Vehicle Clustering Performance Analysis

6.1 Introduction

In Chapter 4, we proposed a framework of clustering algorithm, named UFC, and evaluated it by comparing with other algorithms on NS2. Based on this framework, we analyzed the impact of cluster merging method on clustering performance, proposed a new cluster merging scheme LCM, and presented a general model for evaluation. In this chapter, we propose a stochastic model, in order to have a deep understanding of clustering process and UFC algorithm.

According to the literature, some researchers have already provided stochastic analysis of clustering algorithms and intra-cluster connections. In [127], the authors proposed a stochastic analysis for single-hop communication link of vehicles. A discrete time finite-state Markov chain with state-dependent transition probabilities is introduced to model the distance headway. Based on this model, the authors continued to analyze the single-hop cluster stability in [128], and proposed a discrete-time lumped Markov chain to model the time variations of a system of distance headways. In [129], the researchers modeled a VANET cluster with a Markov chain, taking into account the impact of channel fading and vehicle mobility. All of these works modeled the variance of headways and analyzed inter-vehicle connections. However, we believe that a stochastic model is needed to investigate the entire clustering process. We also believe that the model needs to be based on reasonable experimental results rather than just assumptions.

In this chapter, a stochastic model for clustering performance analysis is presented, based on the simulation results of our proposed clustering framework in Chapter 4. Vehi-

cles' state transitions during clustering process are modeled as a discrete-time finite-state Markov chain. An in-depth analysis of the clustering algorithm can be achieved through this model, including cluster's lifetime and cluster member's lifetime. Numerical results are presented to evaluate the proposed model, which shows high consistency between analytical and simulation results. Furthermore, a long-term clustering performance can be predicted through the proposed model.

The rest of this chapter is organized as follows: Section 6.2 describes the proposed stochastic clustering model, and the analysis is given in Section 6.3. Then, Section 6.4 provides the simulation settings and numerical results. Section 6.5 concludes the chapter and briefly presents the future work.

6.2 Analytic Model

According to the clustering framework UFC described in Chapter 4, each vehicle operates one of the four states (UN, CCM, CM, and CH) at a time during vehicle clustering process. With clustering proceeding, vehicles make the corresponding state transitions due to different trigger events, as shown in Figure 4.2. Therefore, we model the vehicle's state transition process as a discrete-time finite-state Markov chain X_k , aiming to analyze and predict the clustering performance, such as CH lifetime and CM lifetime.

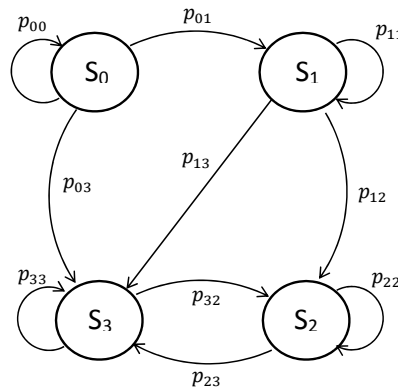


Figure 6.1: Markov chain model of vehicle clustering state transition

The Markov chain X_k , as shown in Figure 6.1, has 4 states $S = \{s_0, s_1, s_2, s_3\}$, representing UN, CCM, CM, and CH, respectively. The state is denoted by s_i , for all i , $0 \leq i \leq 3$, and $s_i \in S$. Within a time step T , the transition probability from s_i to s_j is

$$P\{X_{k+1} = s_j | X_k = s_i\} = p_{ij}. \quad (6.1)$$

According to the process of our clustering framework, the probability transition matrix

P is given by

$$P = \begin{pmatrix} p_{00} & p_{01} & 0 & p_{03} \\ 0 & p_{11} & p_{12} & p_{13} \\ 0 & 0 & p_{22} & p_{23} \\ 0 & 0 & p_{32} & p_{33} \end{pmatrix} \quad (6.2)$$

and satisfy $\sum_{j \in S} p_{ij} = 1$, $i \in S$.

Let $\mathbf{Q}(t_0)$ be the initial distribution when clustering process starts at $t = t_0$. Since all vehicles are UN nodes at the beginning, $\mathbf{Q}(t_0) = \{1, 0, 0, 0\}$. After k time steps, the distribution can be described as $\mathbf{Q}(t_0 + kT) = \mathbf{Q}(t_0) \cdot P^{(k)}$.

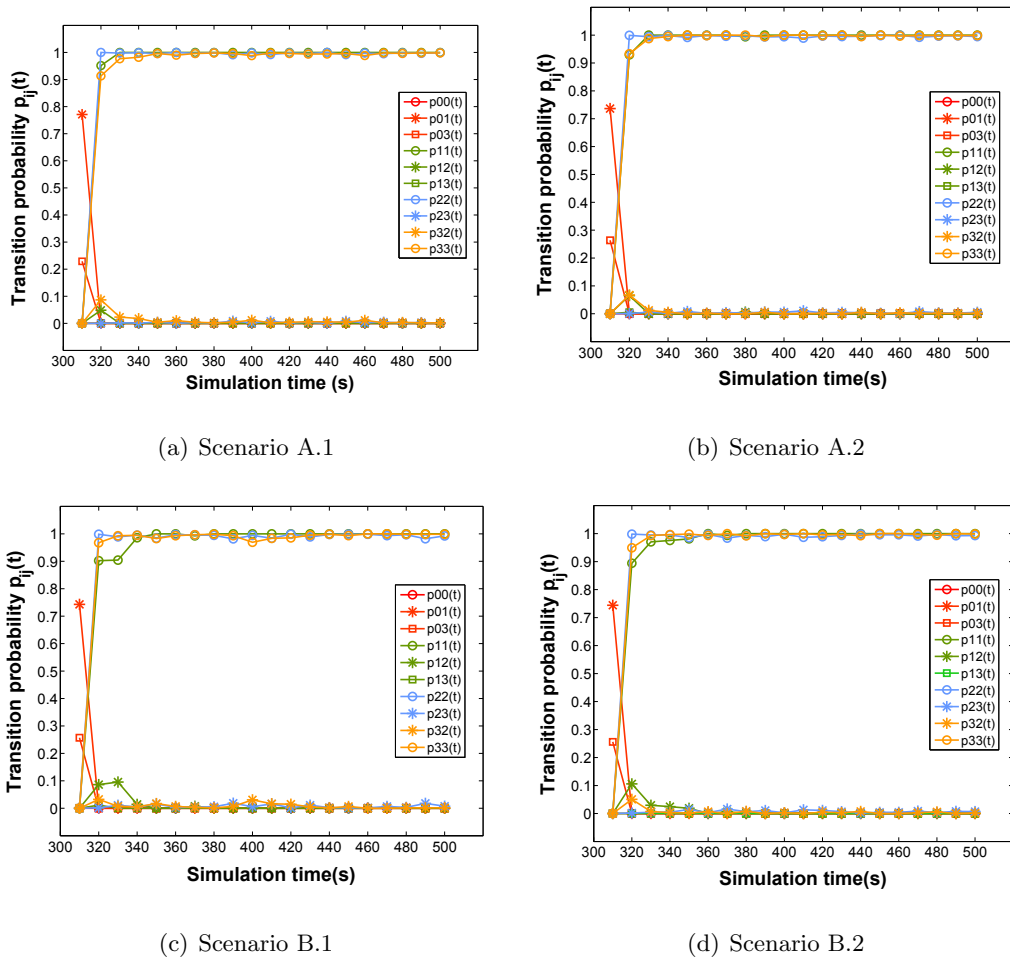


Figure 6.2: State transition probabilities $p_{ij}(t)$

From the experimental results of [21], we can obtain the vehicles' state distribution at each time step (every 10 seconds). Therefore, the state transition probabilities can be calculated at each time step, shown in Figure 6.2. It can be observed that the state transition probability is not constant and varies when time increases. At the beginning 20 seconds, a large number of state transitions occurred due to the CH selection and cluster

formation process. Then, vehicles enter the cluster maintenance process and are becoming more stable with fewer state transitions. Therefore, the transition probabilities are related to time t , and the probability transition matrix is written as $P(t)$. The Markov chain is non-homogeneous.

$$P(t) = \begin{pmatrix} p_{00}(t) & p_{01}(t) & 0 & p_{03}(t) \\ 0 & p_{11}(t) & p_{12}(t) & p_{13}(t) \\ 0 & 0 & p_{22}(t) & p_{23}(t) \\ 0 & 0 & p_{32}(t) & p_{33}(t) \end{pmatrix} \quad (6.3)$$

To simplify this model, a clustering process can be treated as two parts: the unstable clustering period with a large variance of state transition probabilities, denoted as L_1 ; and the stable clustering period with a small variance of state transition probabilities, denoted as L_2 . Assuming that the clustering process starts at time $t = t_0$, and there are m and n time slots in L_1 and L_2 , respectively; then, $L_1 = [t_0, t_0 + mT]$, and $L_2 = [t_0 + mT, t_0 + (m + n)T]$. Here, m and n depend on the time duration of CH selection, simulation time, and the value of each time slot T .

Generally, m is very small and $p_{ij}(t)$ can be obtained directly from the experimental results when $t_0 \leq t \leq t_0 + mT$. When $t \geq t_0 + mT$, since the variance of $p_{ij}(t)$ is very small, $p_{ij}(t)$ can be approximated as a constant value, and the proposed Markov chain can be changed to a time-homogeneous Markov chain. Here, we use the averaged state transition probability $\overline{p_{ij}}$ in L_2 time period instead of $p_{ij}(t)$,

$$\overline{p_{ij}} = \frac{\sum_{k=1}^n p_{ij}(t_0 + (m + k)T)}{n}, i, j \in S. \quad (6.4)$$

Therefore, the expected time that a vehicle spends in each Markov state is composed of two parts, given by

$$E_{s_i} = E_{s_i}[L_1] + E_{s_i}[L_2], \quad (6.5)$$

where $s_i \in S$.

6.3 Analysis of Clustering Performance

Based on the proposed Markov chain model in the previous section, the average expected time that a vehicle spends in the state s_i is the period from a vehicle entering the state s_i until changing to another state s_j after several time slots. To calculate the expected time that the vehicle spends in a specific state, we can simplify the 4-state Markov chain to a 2-state birth and death Markov chain, including the current state and non-current state, which are represented by state A and state B in Figure 6.3.

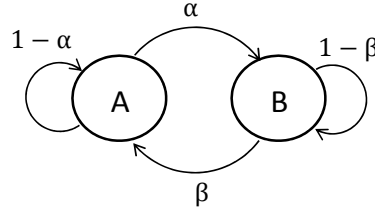


Figure 6.3: 2-state Markov chain model

α indicates the probability that a vehicle gives up the current state and changes to another state in a single time slot T , and β indicates the probability that a vehicle change to the current state from other states. Based on Eq. 6.5, the expected time that a vehicle spends in state A is

$$E_A = E_A[L_1] + E_A[L_2]. \quad (6.6)$$

When $t_0 \leq t \leq t_0 + mT$, $E_A[L_1]$ is given by

$$\begin{aligned} E_A[L_1] &= \beta(t_1) \cdot (m-1) \cdot T + (1-\beta(t_1)) \cdot \beta(t_2) \cdot (m-2) \cdot T \\ &+ (1-\beta(t_1)) \cdot (1-\beta(t_2)) \cdot \beta(t_3) \cdot (m-3) \cdot T + \dots \\ &+ (1-\beta(t_1)) \cdot (1-\beta(t_2)) \cdot \dots \cdot (1-\beta(t_{m-2})) \cdot \beta(t_{m-1}) \cdot T \\ &= \sum_{j=1}^{m-1} \left\{ \left\{ \prod_{i=1}^{j-1} [1-\beta(t_i)] \right\} \cdot \beta(t_j) \cdot (m-j) \cdot T \right\}, \end{aligned} \quad (6.7)$$

where i means the i th time slot during L_1 , $1 \leq i \leq m$, and $\beta(t_i)$ is the transition probability from state B to state A at the i th time slot.

When $t_0 + mT \leq t \leq t_0 + (m+n)T$, $E_A[L_2]$ is given by

$$\begin{aligned} E_A[L_2] &= \bar{\alpha} \cdot T + (1-\bar{\alpha}) \cdot \bar{\alpha} \cdot 2T + (1-\bar{\alpha})^2 \cdot \bar{\alpha} \cdot 3T + \dots \\ &+ (1-\bar{\alpha})^{(n-1)} \cdot \bar{\alpha} \cdot nT + \left[1 - \sum_{i=1}^n \bar{\alpha} \cdot (1-\bar{\alpha})^{(i-1)} \right] \cdot nT \\ &= \sum_{i=1}^n \left\{ \bar{\alpha} \cdot (1-\bar{\alpha})^{(i-1)} \cdot i \cdot T \right\} \\ &+ \left[1 - \sum_{i=1}^n \bar{\alpha} \cdot (1-\bar{\alpha})^{(i-1)} \right] \cdot nT, \end{aligned} \quad (6.8)$$

where $\bar{\alpha}$ is the averaged state transition probability from state A to state B during L_2 , which can be calculated from to Eq.6.4.

To calculate the averaged time duration in each state, we should obtain the value of $\bar{\alpha}$, $\beta(t_i)$, and also the state distribution at last time step $\mathbf{Q}(t_{i-1})$.

6.3.1 Cluster head lifetime

The cluster head lifetime represents the lifetime of the cluster. It is the time period from a vehicle entering the state CH until changing to another state. Based on the 2-state Markov chain model in Figure 6.3, CH is presented by state A and non-CH states (including UN, CCM, and CM) are represented by a unique state B . Corresponding to the model in Figure 6.1, $A = \{s_3\}$, and $B = \{s_0, s_1, s_2\}$. Any state in B can change to s_3 , and s_3 can only change to s_2 or remains in s_3 . Therefore, the probability α and β can be given by

$$\begin{cases} \bar{\alpha} = \overline{p_{32}}, \\ 1 - \bar{\alpha} = \overline{p_{33}} = 1 - \overline{p_{32}}, \\ \beta(t_i) = [Q_{s_0}(t_{i-1}) \cdot p_{03}(t_i) + Q_{s_1}(t_{i-1}) \cdot p_{13}(t_i) \\ + Q_{s_2}(t_{i-1}) \cdot p_{23}(t_i)] / [Q_{s_0}(t_{i-1}) + Q_{s_1}(t_{i-1}) + Q_{s_2}(t_{i-1})]. \end{cases}$$

where $\overline{p_{32}}$ and $\overline{p_{33}}$ are the averaged transition probabilities in L_2 , which can be obtained based on Eq. 6.4. t_i is the time after i th time slot, and $Q_{s_j}(t_{i-1})$ represents the distribution of s_j in $(i-1)$ th time slot.

6.3.2 Cluster member lifetime

The average CM lifetime is the time period from a vehicle entering the state CM until changing to another state. In Figure 6.3, state A represents the state CM, and state B represents non-CM states (including CCM and CH state). Corresponding to the model in Figure 6.1, $A = \{s_2\}$, and $B = \{s_1, s_3\}$. Therefore, the probability α and β are given by

$$\begin{cases} \bar{\alpha} = \overline{p_{23}}, \\ 1 - \bar{\alpha} = \overline{p_{22}} = 1 - \overline{p_{23}}, \\ \beta(t_i) = [Q_{s_1}(t_{i-1}) \cdot p_{12}(t_i) + Q_{s_3}(t_{i-1}) \cdot p_{32}(t_i)] / \\ [Q_{s_1}(t_{i-1}) + Q_{s_3}(t_{i-1})], \end{cases}$$

where $\overline{p_{23}}$ and $\overline{p_{22}}$ are the averaged transition probabilities during L_2 , and t_i is the time after i th time slot during L_1 . $Q_{s_j}(t_{i-1})$ represents the distribution of s_j in $(i-1)$ th time slot.

6.3.3 Candidate cluster member lifetime

The averaged CCM lifetime is the time period from a vehicle becoming a CCM until changing to a CH or CM. In Figure 6.3, state A represents the state CCM, and state B

represents non-CCM states (including CH, CM, and CH). Corresponding to Figure 6.1, $A = \{s_1\}$, and $B = \{s_0, s_2, s_3\}$. Therefore, the probabilities α and β can be given by

$$\begin{cases} \bar{\alpha} = \bar{p}_{12} + \bar{p}_{13}, \\ 1 - \bar{\alpha} = \bar{p}_{11} = 1 - \bar{p}_{12} - \bar{p}_{13}, \\ \beta(t_i) = p_{01}(t_i), \end{cases}$$

where \bar{p}_{12} and \bar{p}_{13} are the averaged transition probabilities during L_2 , and t_i is the time after i th time slot during L_1 . When a vehicle change from the state CCM to other states, it will never change back, and the CCM state can only be changed from the initial state UN.

6.3.4 Time duration in initial state

The averaged UN lifetime is the time period from a vehicle in initial state until changing to another state. According to Figure 6.3, the state A represents the state UN, and the state B represents non-UN states (including CH and CCM state). Corresponding to Figure 6.1, $A = \{s_0\}$, and $B = \{s_1, s_3\}$. The state s_0 can change to s_1 or s_3 with probability p_{01} or p_{03} . All vehicles are in s_0 when clustering starts at t_0 , and a vehicle that leaves s_0 can never come back, $\beta = 0$.

Since s_0 is the initial state and there have no UN nodes anymore during L_2 , the expected time that a vehicle spends in s_0 , E_{s_0} equals to the time during L_1 , which is calculated by

$$\begin{aligned} E_{s_0} &= E_{s_0}[L_1] = \alpha(t_1) \cdot T + (1 - \alpha(t_1)) \cdot \alpha(t_2) \cdot 2T \\ &+ (1 - \alpha(t_1)) \cdot (1 - \alpha(t_2)) \cdot \alpha(t_3) \cdot 3T + \dots \\ &+ (1 - \alpha(t_1)) \cdot (1 - \alpha(t_2)) \cdot \dots \cdot (1 - \alpha(t_{m-1})) \cdot \alpha(t_m) \cdot mT \\ &= \sum_{j=1}^m \left\{ \prod_{i=1}^{j-1} (1 - \alpha(t_i)) \right\} \cdot \alpha(t_j) \cdot j \cdot T. \end{aligned} \quad (6.9)$$

The transition probability $\alpha(t_i)$ is obtained from the following equation:

$$\alpha(t_i) = p_{01}(t_i) + p_{03}(t_i). \quad (6.10)$$

6.3.5 Prediction of state distribution during the clustering process

At the beginning of the clustering process $t = t_0$, all vehicles are in UN state s_0 , and the initial distribution is $\mathbf{Q}(t_0) = \{1, 0, 0, 0\}$. Let $Q_{s_i}(t)$ be the probability that a vehicle stays in s_i at time t , the state distribution at time t is described by

$$\mathbf{Q}(t) = \{Q_{s_0}(t), Q_{s_1}(t), Q_{s_2}(t), Q_{s_3}(t)\}. \quad (6.11)$$

We assume that when $t \rightarrow \infty$, the system is becoming stable. There are only CH, CM, and CCM vehicles on the road with a stationary distribution \mathbf{Q}^* . \mathbf{Q}^* can be calculated from to the following equations:

$$Q_{s_j}^* = \sum_{s_i \in S} Q_{s_i}^* p_{ij}, s_j \in S. \quad (6.12)$$

$$\sum_{s_j \in S} Q_{s_j}^* = 1. \quad (6.13)$$

where p_{ij} can be represented by $\overline{p_{ij}}$, which is calculated from to Eq. 6.4.

6.4 Results and Discussion

This section presents the numerical results of the estimated time that a vehicle spends in each state E_{s_i} and the vehicles' state distributions.

In the simulation, four testing scenarios are generated by Simulation of Urban Mobility (SUMO) [77], and the detailed definition of these testing scenarios can be found in Section 4.3.1 in Chapter 4. In all scenarios, there are 200 vehicles, and the mobility pattern of vehicles in scenario B.2 is more unpredictable than the others.

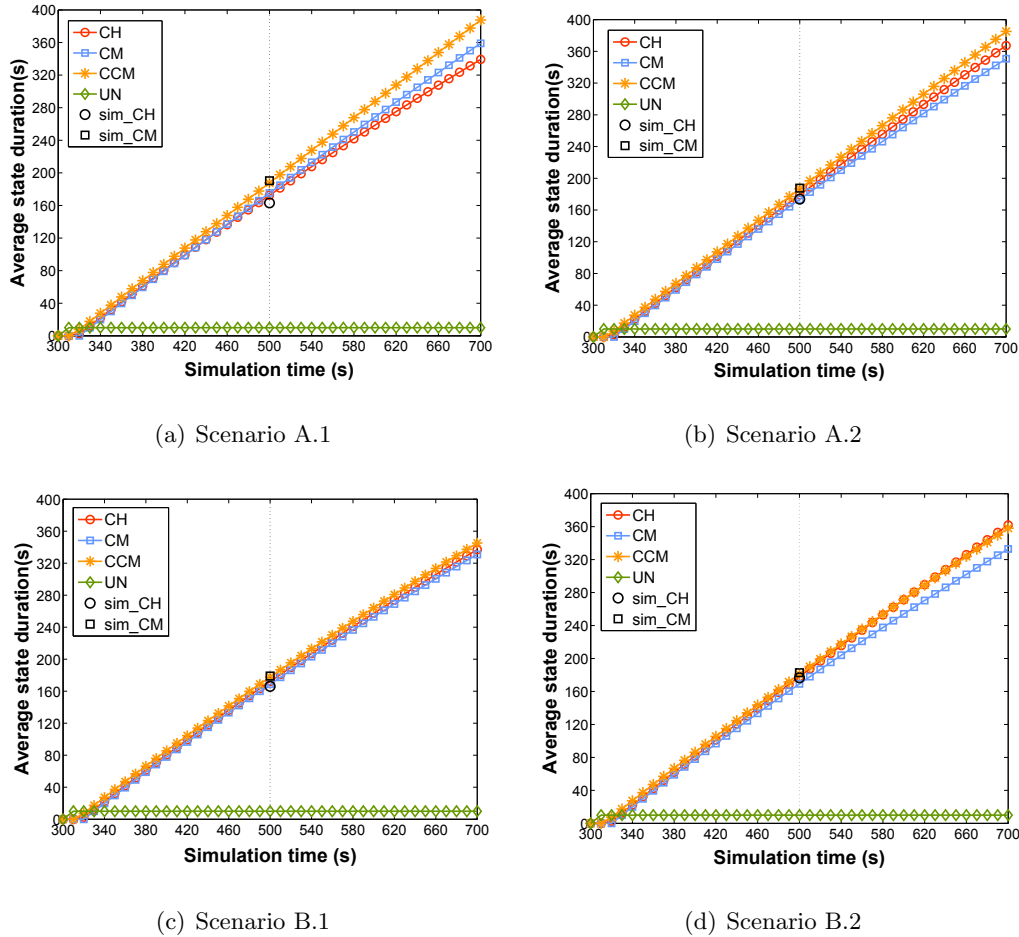
In previous simulations in NS2, the simulation time starts at $t = 300s$ and ends at $t = 500s$. In our simulation, clustering process starts at $t_0 = 300s$, and each time slot is 10s, $T = 10s$. From Figure 6.2 and analysis in Section 6.2, the clustering process is separated into two parts L_1 and L_2 with m and n time slots, respectively. It can be observed that most state changes happen in L_1 , around two time slots. After clusters are formed, there are only few state changes. Therefore, m is fixed to 2, $m = 2$. Based on the previous experimental simulation results in NS2, we can predict the long-term clustering performance through the proposed model, where $t \geq 500s$, and have a comprehensive analysis of the clustering algorithm.

6.4.1 Vehicle state lifetime

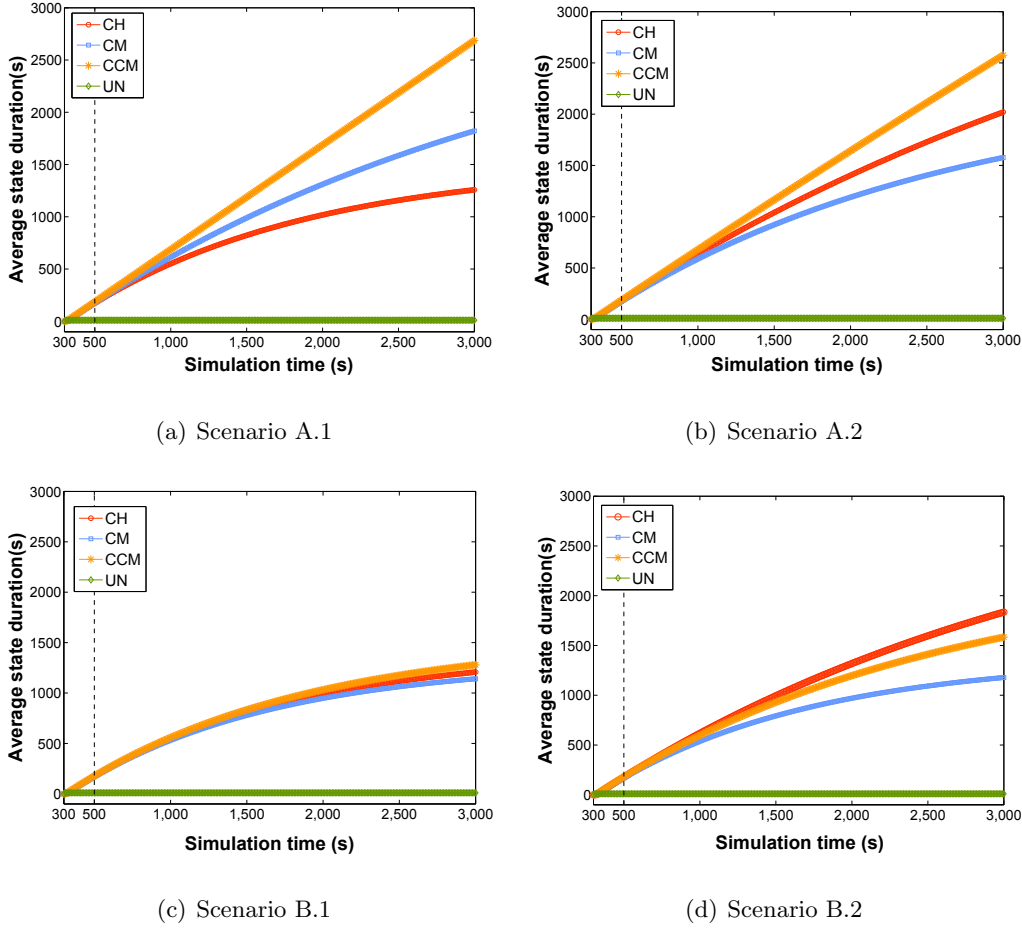
We compare the estimated lifetime to the simulation results of NS2 when $t = 500s$, obtained from Table 4.10 and Table 4.12. To predict the future clustering performance, the simulation time is set to 700s and 3000s in MATLAB, respectively ($t = 700s$, $n = 38$, and $t = 3000s$, $n = 298$).

Table 6.1: Simulation results of the averaged CH and CM lifetime on NS2

	A.1	A.2	B.1	B.2
CH lifetime (s)	163.112	173.645	166.112	176.57
CM lifetime (s)	190.395	187.582	178.963	182.796

Figure 6.4: Expected state duration E_{s_i} ($t = 700s$)

The cluster stability can be represented by CH lifetime and CM lifetime. Figure 6.4 and Figure 6.5 illustrate the expected state duration under different scenarios. The simulation ends at $t = 700s$ and $t = 3000s$, respectively. To better observe the lifetime at the beginning of clustering process, Figure 6.6 presents the variations of E_{s_i} in the first 3 time slots. In the first time slot, all of the state lifetimes are 0, except UN state. The CH lifetime increases in the second time slot since CHs have been selected during this time. From the third time slot when $t = 320s$, the CH lifetime is approximated to a linear increase with the increased simulation time, as shown in Figure 6.4. However, when the simulation time continues to increase, the state lifetime increases slowly, especially under dynamic scenarios, as illustrated in Figure 6.5(c) and 6.5(d).

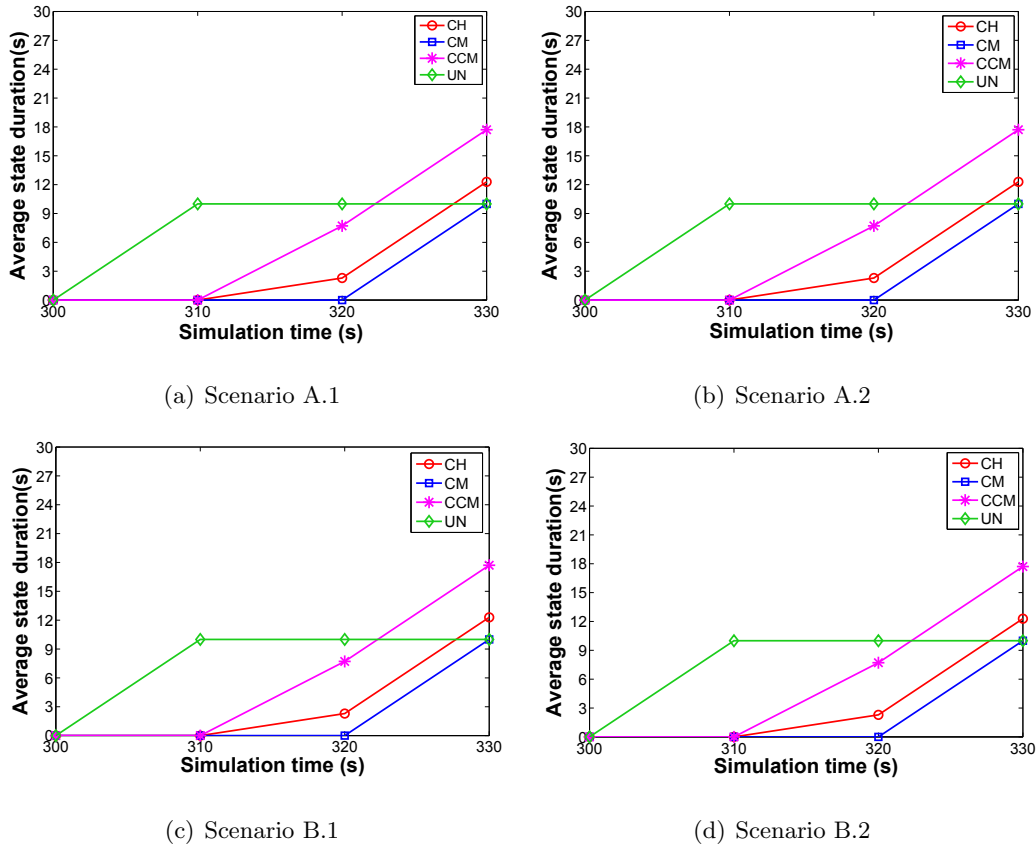
Figure 6.5: Expected state duration $E_{s_i} (t = 3000s)$

A similar trend can be found for CM and CCM lifetime. The CM lifetime remains 0 until the third time slot. It is because CCM vehicles have to ask CHs to join the cluster during the second time slot. Furthermore, since the UN vehicles only appear in the first time slot, analyzed in Section 6.3.4, the duration of UN remains as a constant value, 10s, when time increases.

Table 6.1 shows the simulation results of the average CH and CM lifetime in [21], when the simulation ends at 500s, $t = 500s$. The results are marked as black data points in Figure 6.4. It can be observed that when $t = 500s$, there has a small difference between the estimated value and the simulation value.

6.4.2 Prediction of future state distribution

In Section 6.3.5, we have analyzed the state distribution $\mathbf{Q}(t)$. In order to observe the trend of the clustering process, we set the simulation time as $t = 3000s$, $n = 298$. Figure 6.7 shows the simulation results of state distribution when time increases. It can be observed

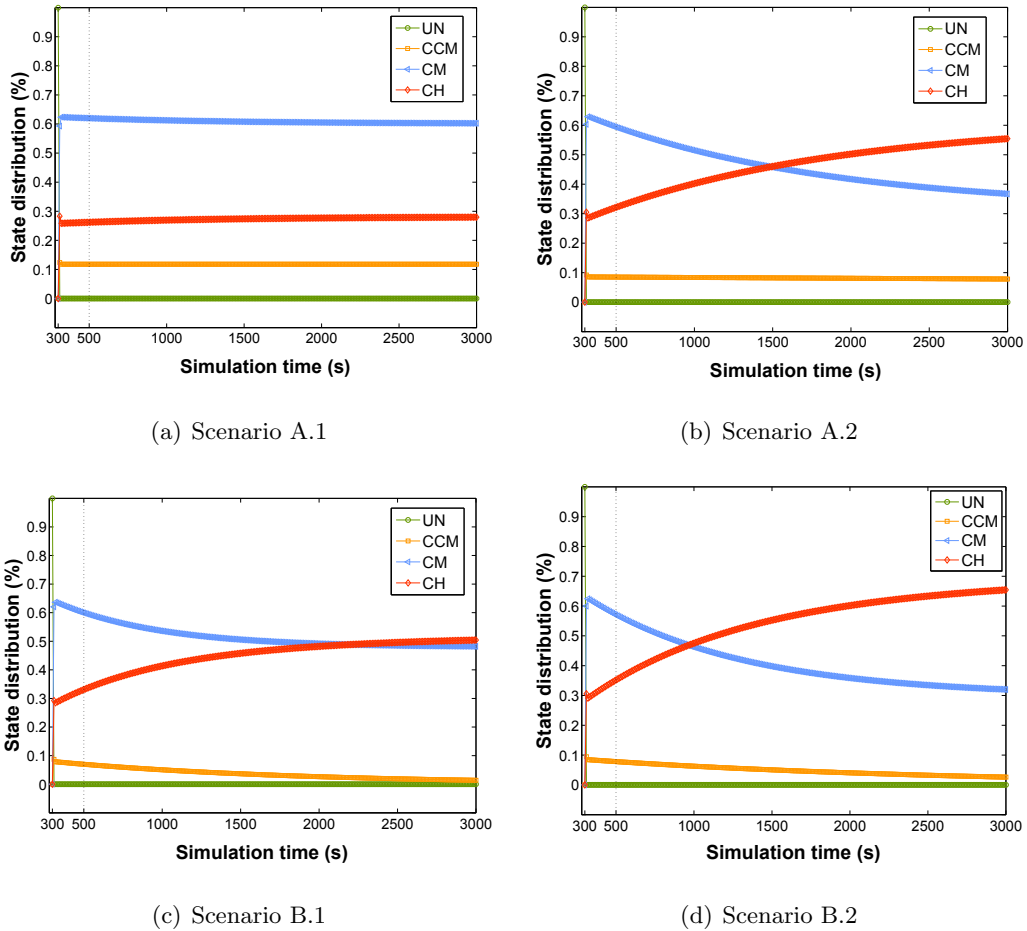
Figure 6.6: Expected state duration E_{s_i} (300 – 330s)

that when time increases, the number of the CHs and CMs are becoming stable. Moreover, when the traffic scenario is more dynamic, such as scenario B.2, shown in Figure 6.7(d), more CMs change to CHs in a shorter time. However, when the traffic scenario is more static, such as A.1, shown in Figure 6.7(a), clusters are more stable.

Table 6.2: Stationary distribution \mathbf{Q}^*

	$Q_{s_0}^*$ (UN)	$Q_{s_1}^*$ (CCM)	$Q_{s_2}^*$ (CM)	$Q_{s_3}^*$ (CH)
A.1	0	0.1180	0.6001	0.2819
A.2	0	0	0.3333	0.6667
B.1	0	0	0.4830	0.5170
B.2	0	0	0.2913	0.7087

According to the analysis in Section 6.3.5, we can obtain the stationary distribution under different traffic scenarios, shown in Table 6.2. It can be observed that all vehicles are in either CH or CM state when $t \rightarrow \infty$, except for scenario A.1. It is because the traffic is less dynamic with a small number of state change. Moreover, in scenario A.2, B.1, and B.2, the number of CHs are becoming more than the number CMs. It is because,

Figure 6.7: State distribution $\mathbf{Q}(t)$ ($t = 3000s$)

with time increases, more vehicles leave the cluster and become single CHs without CMs. When the traffic scenario is more dynamic, more CMs change to single CHs during a specific time, which is the same as illustrated in Figure 6.7(d).

6.5 Conclusion

In this chapter, a discrete-time finite-state Markov chain model for clustering framework UFC is proposed, aiming to analyze the cluster stability, including CH lifetime and CM lifetime. Each vehicle clustering state is modeled as a state in Markov chain. In the proposed Markov chain model, the state transition probabilities are obtained from previous simulation results of UFC on the network simulator NS2. Therefore, the expected time that a vehicle spends in each state can be estimated. Furthermore, the proposed model enables the prediction of long-term clustering performance.

Even though the proposed stochastic model enables us to deep understand the clustering process, there still have some limitations. The proposed Markov chain model only takes

into account of vehicles' state transitions without details; however, each state transition can be triggered by different events. In the future work, a more accurate and complicated event-related model is required.

Cluster-based Emergency Message Dissemination in VANETs

7.1 Introduction

As we have mentioned in Chapter 1, road safety application depends mostly on the dissemination of warning messages to deliver information to the concerned vehicles. The requirements of applications and VANETs inherent characteristics make data dissemination an essential and challenging task.

In previous chapters, we have proposed a framework of clustering algorithm UFC for VANETs, as well as evaluated the cluster performance. Based on UFC, cluster merging process has been discussed, and a stochastic model has been designed to model the clustering process. In this chapter, we design an emergency message dissemination algorithm combined with the UFC-based clustering algorithm. We compare our proposed algorithm to the one combined with VMaSC [7] clustering algorithm. The comparison results are presented and our algorithm shows better performance in terms of packet delivery ratio and packet delivery delay.

The rest of this chapter is organized as follows: Section 7.2 presents the proposed cluster-based emergency message dissemination algorithm. Section 7.3 presents the application scenario and the performance evaluation of our scheme. Section 7.4 concludes this chapter and briefly introduces the future work.

7.2 Description of the scheme

This section describes the proposed emergency message dissemination algorithm based on our previous clustering algorithm UFC in Chapter 4, which can be named as UFC-

based emergency message dissemination algorithm. The primary goal of this algorithm is to disseminate warning messages to all vehicles inside the region of interest (ROI). In this case, our algorithm should solve the broadcast storm and intermittently connected network problems.

We assume that each vehicle is equipped with a GPS device to obtain its current geographical location, velocity, and moving direction. We also assume that all vehicles periodically exchange their location, moving speed, and driving direction enclosed in their periodic CAM messages [13].

All of the vehicles are running on a straight highway in two directions. The accident vehicle will broadcast a warning message $Msg_{warning}$ immediately when the accident happens, as shown in Figure 7.1. The objective is to let all vehicles on this road segment to receive this warning message successfully. Assuming that vehicles have already been clustered according to our clustering algorithm UFC, the emergency message dissemination can be realized by cluster head vehicles (CH) and cluster member vehicles (CM). Moreover, to improve the data dissemination efficiency and reduce the transmission latency, the warning message should be transmitted in both directions.

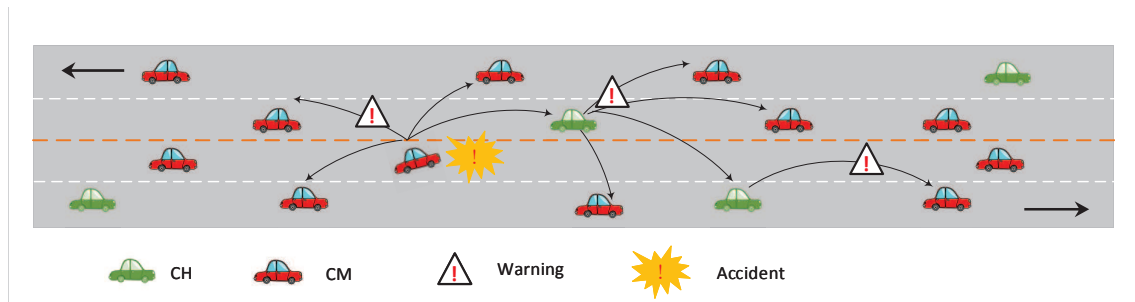


Figure 7.1: Emergency message dissemination

When a CH receives $Msg_{warning}$ from an accident vehicle or other vehicles, it firstly checks whether it has already received this message or not. If it is the first time that the CH receives this message, it broadcasts $Msg_{warning}$ to inform its CMs, as well as stores $Msg_{warning}$ in its buffer. If the CH has already received $Msg_{warning}$ but did not relay this warning message, it broadcasts $Msg_{warning}$ immediately.

In our algorithm, when the CH receives a warning message for the first time, it can store this warning message for a short time period, named $T_{CH_{store}}$, which equals to the Time-To-Live (TTL) of the warning message. During this time period, the CH will rebroadcast the warning message periodically with a relay interval, named T_{RI} , except when it changes its state. Therefore, the moving CH is able to inform more vehicles on the road, especially when the vehicle density is very low.

When a CM receives $Msg_{warning}$ from an accident vehicle, it firstly checks whether it has already received this message or not. If it is the first time the CM receives this message, it will deliver the message to its belonged CH. If the CM has already received $Msg_{warning}$ but has not delivered the message, the CM will deliver $Msg_{warning}$ to its CH directly. If a CM receives $Msg_{warning}$ from a relaying node, it will check whether the message is delivered by its belonged CH. If not, the CM relays the message to its belonged CH.

7.3 Performance Evaluation

In this section, we evaluate the proposed cluster-based emergency message dissemination algorithm based on the UFC-based [21] and VMaSC [7] clustering algorithms, respectively. Since UFC is based on single-hop cluster, the single-hop VMaSC is implemented in our simulation, named VMaSC_1hop. All of the algorithms are implemented on the Network Simulator NS2 [125]. The simulation configuration is described as follows.

Table 7.1: Vehicle setting for scenario A.1

Max speed	Acceleration	Deceleration	Speed deviation
20 m/s	2.0 m/s^2	6.5 m/s^2	0.1

Table 7.2: Vehicle setting for scenario A.2

Type	Max speed	Acceleration	Deceleration	Speed deviation
1	20 m/s	2.9 m/s^2	7.5 m/s^2	0.7
2	20 m/s	2.9 m/s^2	7.5 m/s^2	0.3
3	20 m/s	2.0 m/s^2	6.5 m/s^2	0.1
4	20 m/s	1.5 m/s^2	5.5 m/s^2	0.3

Table 7.3: Vehicle setting for scenario B.1 and B.2

Type	Max speed	Acceleration	Deceleration	Speed deviation
1	35 m/s	2.9 m/s^2	7.5 m/s^2	0.7
2	25 m/s	2.9 m/s^2	7.5 m/s^2	0.3
3	20 m/s	2.0 m/s^2	6.5 m/s^2	0.1
4	10 m/s	1.5 m/s^2	5.5 m/s^2	0.3

Table 7.4: Testing scenario settings

Scenario	Vehicle type	Maximum Lane speed (m/s)
A.1	Table 7.1	8 segments (for each: 20)
A.2	Table 7.2	8 segments (for each: 20)
B.1	Table 7.3	8 segments (20,30,20,30,10,20,15,20)
B.2	Table 7.3	8 segments (20,15,25,30,25,20,15,20)

7.3.1 Testing scenarios

In the simulation, we generate four testing scenarios by Simulation of Urban Mobility (SUMO) [77], aiming to observe the cluster performance under different traffic scenarios. In all scenarios, there are 200 vehicles: 100 from East to West, and 100 from West to East. The length of the road is set to 10 *km* which is equally divided into 8 segments, and vehicles are generated with a certain traffic generation rate, 1500 vehicles per hour.

Table 7.5: Default simulation parameters

Parameter	Value
Simulation time	200 <i>s</i>
T_{start}	300 <i>s</i>
T_{end}	500 <i>s</i>
$T_{accident}$	340 <i>s</i>
Length of road	10 <i>km</i>
Number of vehicles	200
Traffic generation rate	1500 <i>vehicles/h</i>
Transmission Range (TR)	200-500 <i>m</i>
CAM frequency	1.0 <i>Hz</i>
$T_{CH_{store}}$	30 <i>s</i>
T_{RI}	5 <i>s</i>
CAM size	66 bytes
Mobility model	Car-following model
Number of iterations	10

For a more realistic mobility behavior, these four testing scenarios consist of two relative static traffic models and two highly dynamic traffic models, which are named as scenarios A1, A2, B1, and B2 respectively. There is only one vehicle type in scenario A1, shown in Table 7.1, with the same maximum speed of 20 *m/s* and the same speed acceler-

ation rate, and there are four vehicle types in scenario A2 with the same maximum speed of 20 m/s and different speed acceleration rates, shown in Table 7.2. For scenarios B1 and B2, both of them have four vehicle types with different maximum velocities, as shown in Table 7.3, but the maximal speed limit of each segment is different which is presented in Table 7.4. It can be observed that the mobility pattern of vehicles in scenarios B2 is more unpredictable than the others.

In our simulation, clustering process starts at $T_{start} = 300s$ and ends at $T_{end} = 500s$. An accident happens at $T_{accident} = 340s$, at 4km away from the left beginning of the road segment, on the lane with the direction from west to east. We focus on the immediate consequences of an accident on a highway. The crashed vehicle starts to generate and broadcast a warning message immediately after the collision to warn all of the vehicles as quickly as possible in a distributed way. The transmission range changes from 200m to 500m, and the CAM messages are broadcasted with a frequency 1.0Hz. More simulation parameters and settings of MAC and PHY layers are illustrated in Table 7.5 and Table 7.6.

Table 7.6: Settings of PHY and MAC layer

Category	Parameter	Value
PHY	Propagation model	Two-Ray Ground
	Frequency/Channel Bandwidth	5.9GHz/10MHz
	Power Monitor Threshold	-174dBm
	Antenna	OmniAntenna
	Noise floor	-99dBm
	Carrier Sense Threshold	-94dBm
MAC	MAC protocol	IEEE 802.11p
	Retry Limit	7
	Header Length	40 μs
	SlotTime/SIFS	13 μs /32 μs
	CWmin/CWmax	15/1023

7.3.2 Performance metrics

In this work, we use two metrics to evaluate the cluster-based emergency message dissemination algorithm:

- **Packet Delivery Ratio (PDR)**: It indicates the percentage of vehicles that have successfully received a single event-related message during the simulation.

- **Average Packet Delivery Delay (APDD):** It represents the average time from the crashed vehicle generating the first warning message until the message is finally received by the destination node.

7.3.3 Performance analysis

Figure 7.2 shows the numerical results of averaged packet delivery ratio with the increase of vehicle's transmission range. Normally, the transmission range is limited up to 350m. However, we are interested in the impacts of transmission range on the performance of the algorithm, and set the transmission range from 200m to 500m in our simulation.

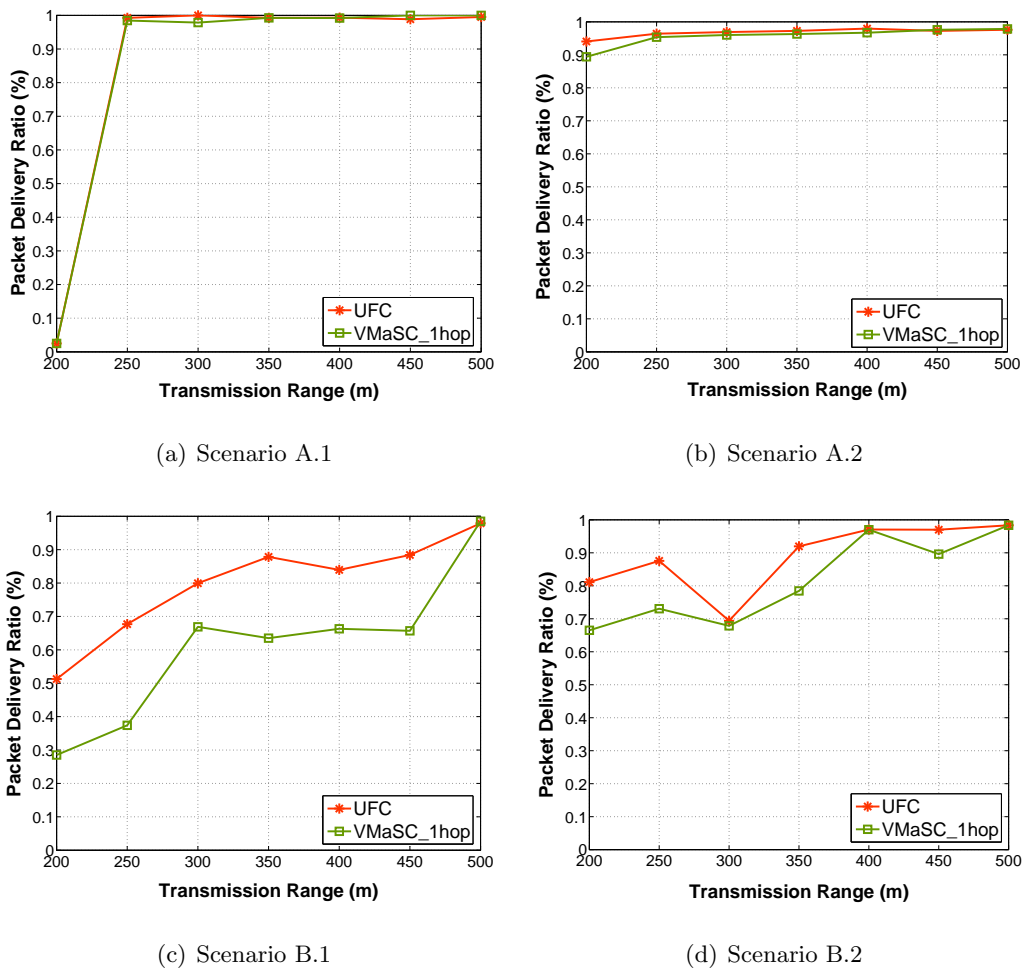


Figure 7.2: Packet Delivery Ratio (%)

It can be observed that the packet delivery ratio increases with the increase of vehicle's transmission range. When the traffic scenarios are less dynamic, as scenario A.1 and A.2, the packet delivery ratio can achieve almost 100%. In Figure 7.2(a), when the transmission range is 200m, the packet delivery ratio is extremely low. It is because that there has less neighboring vehicles around the accident vehicle and information can not be delivered

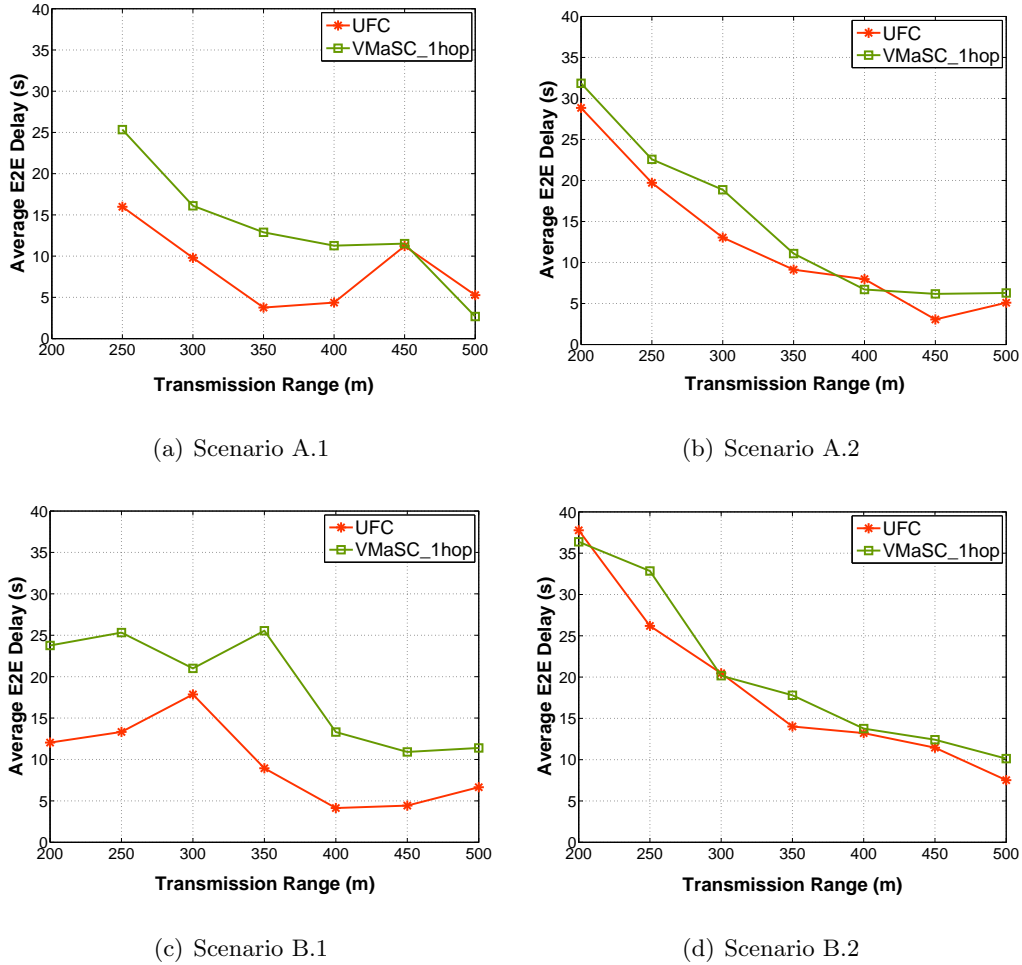


Figure 7.3: Average Packet Delivery Delay (s)

successfully.

Moreover, comparing with VMaSC_1hop, our cluster-based data dissemination algorithms shows higher packet delivery ratio, especially in dynamic traffic scenarios, as shown in Figure 7.2(c) and 7.2(d).

Figure 7.3 presents the numerical results of the averaged packet delivery delay with the increase of transmission range. When the transmission range increases, the packet delivery delay decreases slightly. It means that it spends less time to let all of the vehicles on the road to receive the warning message. Since the packet delivery ratio is extremely low when the transmission range is 200m in scenario A.1, its packet delivery delay is meaningless and can be ignored, as shown in Figure 7.3(a).

Moreover, it can be observed that our algorithm performances lower average packet delivery delay than VMaSC_1hop-based emergency message dissemination algorithm. We can conjecture that it is because of the BackUp CH based cluster maintenance scheme (BUCH) of UFC algorithm. The disconnected CM can be reconnected as soon as possible

with BUCH scheme; therefore, the warning message can be delivered successfully with fewer packet loss.

To analyze the performance of our algorithm in detail, Figure 7.4, 7.5, 7.6, and 7.7 presents the averaged packet delivery ratio of each traffic scenario respectively. It can be observed that the packet delivery ratio increases with the increased simulation time at the beginning. After a short time period, the packet delivery ratio becomes stable. It is because that the accident vehicle stops broadcasting the warning message and the relaying vehicles stop delivering the message.

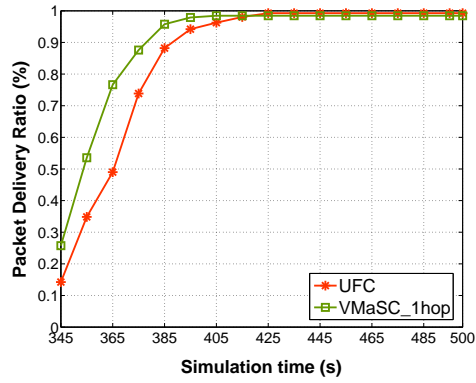
It can be observed that when the transmission range increases, the packet delivery ratio becomes stable within fewer time. In other words, the warning messages are delivered to the target vehicles within fewer time. Moreover, when the traffic scenario becomes more dynamic, as shown in Figure 7.6 and 7.7, more time is need for the packet delivery ratio to be stable. In other words, it spends more time for the warning messages to be delivered to the target vehicles.

When comparing two cluster-based data dissemination algorithms, our UFC-based emergency message dissemination presents higher packet delivery ratio, especially in dynamic scenarios B.1 and B.2, as presented in Figure 7.6 and 7.7.

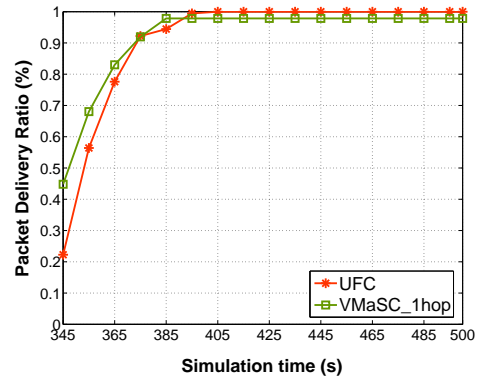
7.4 Conclusion

In this chapter, we proposed an emergency message dissemination algorithm combined with our proposed clustering algorithm UFC [21], called UFC-based emergency message dissemination algorithm. We also combine this message dissemination algorithm with VMaSC [7] clustering algorithm and compare their performances. The numerical results show that our UFC-based emergency message dissemination algorithm presents superior performance in both packet delivery ratio and packet delivery delay. On the other hand, the results also prove the advantages of UFC clustering algorithm.

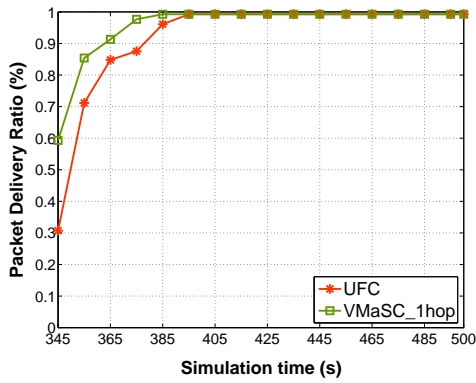
In the future work, the message type is not limited to emergency message. Clusters are able to handle de dissemination of different types of messages at the same time, such as weather report message, video message, and traffic condition message. The cluster head vehicles can handle these messages with a content caching and information aggregation use cases.



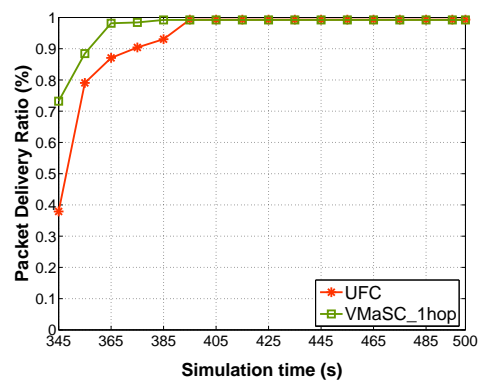
(a) TR=250



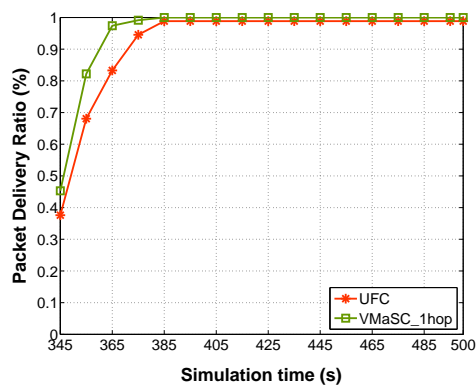
(b) TR=300



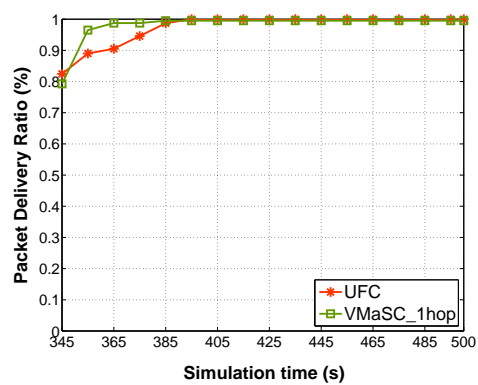
(c) TR=350



(d) TR=400

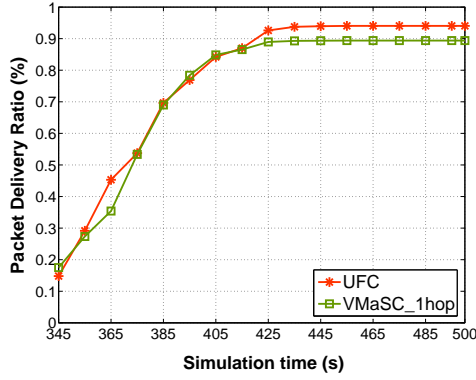


(e) TR=450

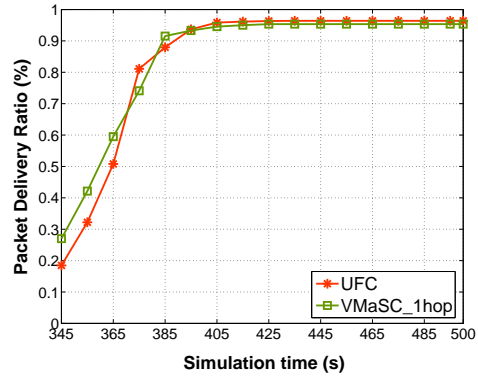


(f) TR=500

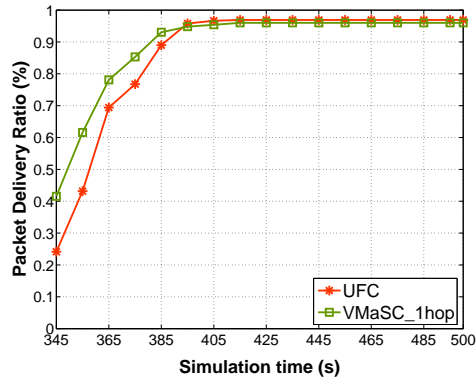
Figure 7.4: Packet Delivery Ratio (%) under Scenario A.1



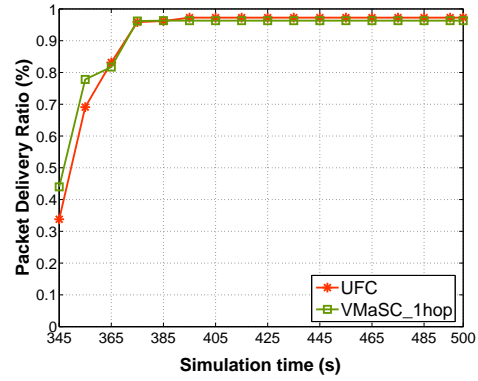
(a) TR=200



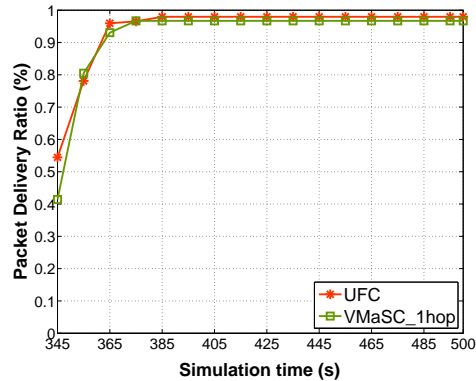
(b) TR=250



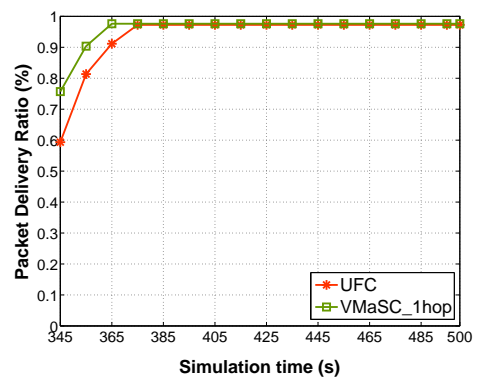
(c) TR=300



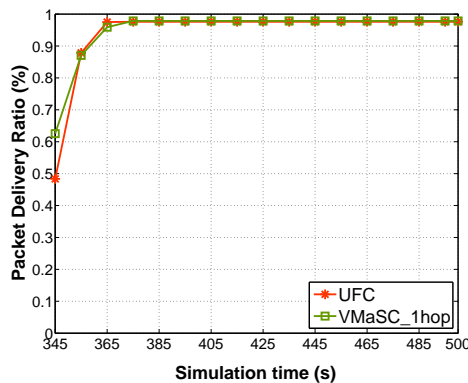
(d) TR=350



(e) TR=400

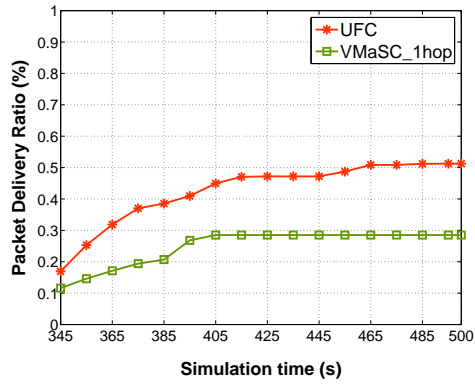


(f) TR=450

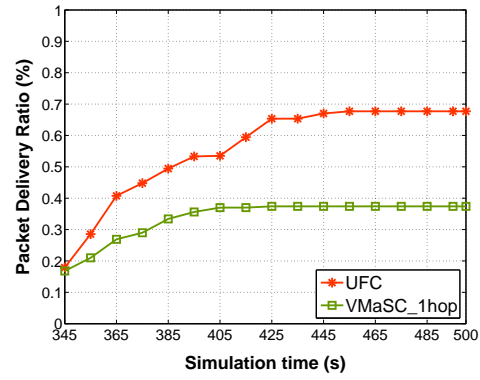


(g) TR=500

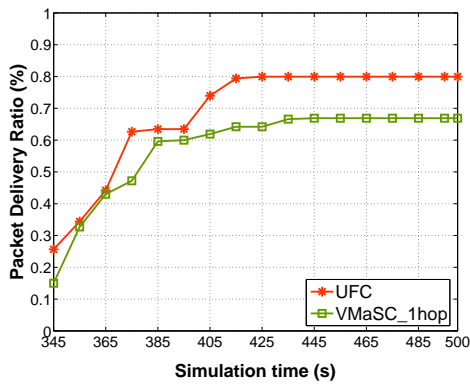
Figure 7.5: Packet Delivery Ratio (%) under Scenario A.2



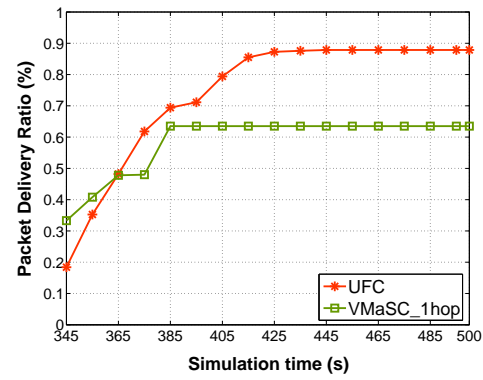
(a) TR=200



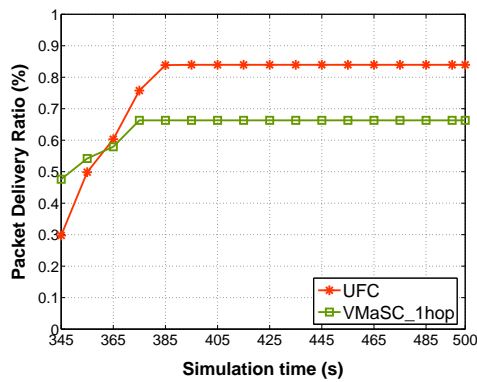
(b) TR=250



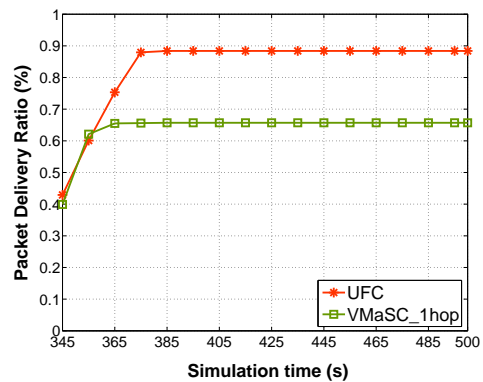
(c) TR=300



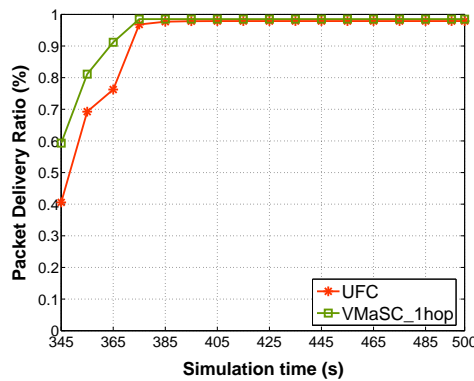
(d) TR=350



(e) TR=400

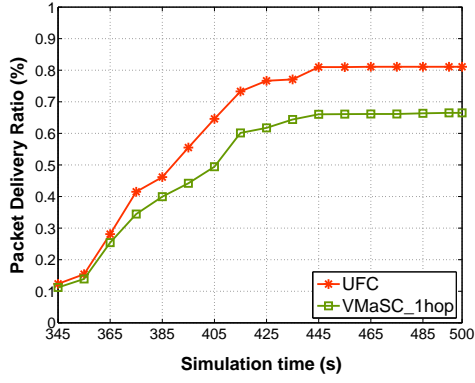


(f) TR=450

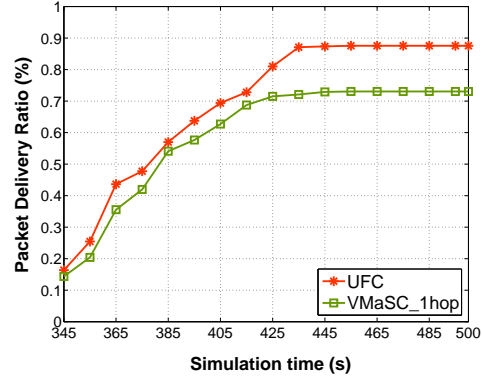


(g) TR=500

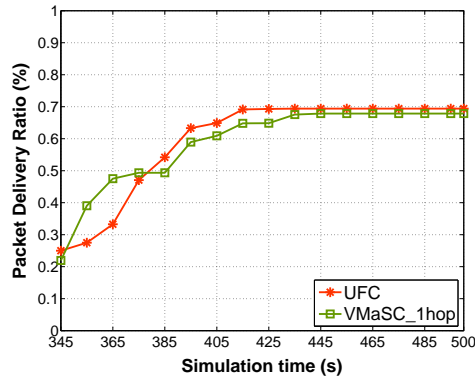
Figure 7.6: Packet Delivery Ratio (%) under Scenario B.1



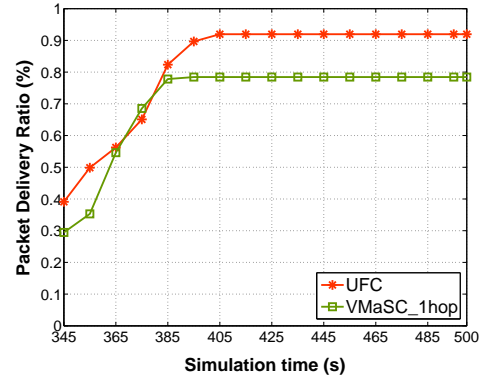
(a) TR=200



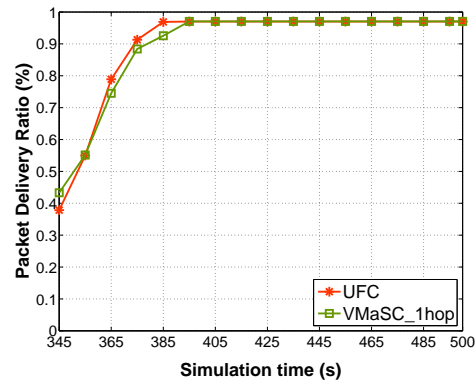
(b) TR=250



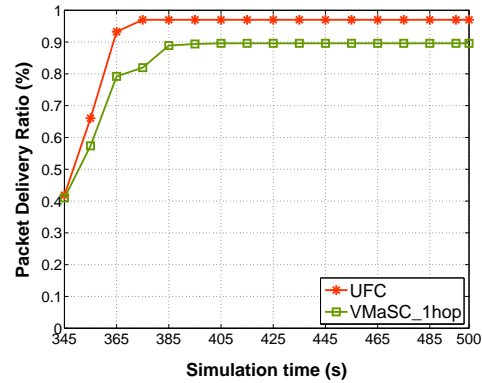
(c) TR=300



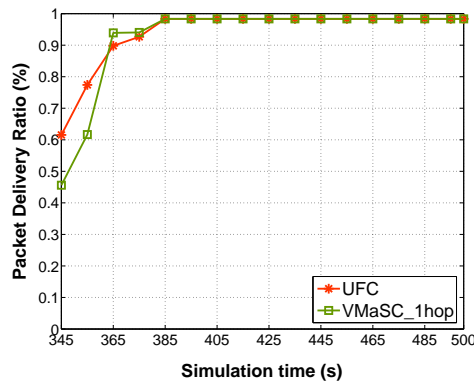
(d) TR=350



(e) TR=400



(f) TR=450



(g) TR=500

Figure 7.7: Packet Delivery Ratio (%) under Scenario B.2

Conclusions and Perspectives

In this manuscript, we have addressed the characteristics and challenges of VANETs. We have studied the necessity of clustering algorithms to enhance inter-vehicle connections by investigating several clustering algorithms. Firstly, we proposed a new mobility-based scheme for dynamic clustering (MoDyC) to improve the cluster stability. Secondly, to further analyze the impact of different clustering metrics, we designed a unified framework of clustering algorithm (UFC), which provides a platform for clustering evaluation. Furthermore, based on UFC, we introduced a leadership-based cluster merging scheme (LCM) and studied the impacts of cluster merging schemes on clustering performance. Next, we proposed a stochastic model for clustering performance analysis. In the end, we derived a cluster-based data dissemination method for content caching. This chapter summarizes the thesis by outlining the contributions above and by discussing future research directions.

8.1 Conclusions

First of all, we studied the existing clustering algorithms and the cluster-based data dissemination algorithms in VANETs in Chapter 2. A new classification of these algorithms has been proposed according to their context. Moreover, we have summarized their methods of evaluation, such as performance metrics, traffic scenarios, and simulation tools.

Based on our review in previous chapter, we have pointed out that using mobility metrics for vehicle clustering is more effective in improving **cluster stability**. Therefore, in Chapter 3, we proposed a mobility-based scheme for dynamic clustering (MoDyC) which uses inter-vehicle distance to control the cluster size. To evaluate MoDyC's clustering performance, we compared it with LID and 1-hop VMaSC algorithms. Meanwhile, we analyzed the impacts of cluster size (D_t) and maximum lane speed (MLS) on MoDyC's

cluster stability. Although MoDyC shows improved cluster stability, especially under high dynamic traffic scenarios, the cluster construction process that adds vehicles one by one may create unnecessary time costs and extra overhead. Moreover, we are interested in using alternative clustering metrics instead of inter-vehicle distance and investigating their suitabilities.

In order to solve the problems that we found in MoDyC and to investigate the impacts of different clustering metrics, we proposed a complete and efficient unified framework for clustering (UFC) in Chapter 4. UFC is composed of three parts: Neighbor Sampling (NS), Backoff-based Cluster head Selection (BCS), and BackUp Cluster Head based cluster maintenance (BUCH). The impacts of three clustering metrics have been evaluated: inter-vehicle distance, relative velocity, and link lifetime. Moreover, we compared UFC with LID and 1-hop VMaSC under four traffic scenarios. The simulation results show that UFC presents better cluster stability (particularly lower CM disconnection frequency and lower role change frequency), especially under highly dynamic network topologies.

During our research, we noticed that cluster merging is easily overlooked, and many clustering algorithms did not mention their cluster merging methods. Therefore, we are interested in investigating the importance of cluster merging in the clustering process. Based on our proposed clustering framework UFC, we proposed a leadership-based cluster merging scheme (LCM) in Chapter 5 and compared its performance with CM-based and VMaSC-based cluster merging schemes. Moreover, a general model has been proposed to analyze their impacts on cluster stability. The results have proved that cluster merging method does influence clustering performance, although the impact is not that significant.

In Chapter 6, we presented a stochastic model of clustering process, aiming to further analyze the cluster stability, including CH and CM lifetime. The clustering process can be treated as a sum of several vehicle state transitions. Therefore, we proposed a discrete-time finite-state Markov chain model based on UFC, and the expected time that a vehicle spends in each state can be estimated. The model was evaluated by MATLAB and the results, including CH and CM lifetime, showed high consistency to the simulation results of NS2. Furthermore, we predicted the state distribution in long-term clustering process when the simulation time is set to 700s and 3000s respectively.

In Chapter 7, we proposed a cluster-based emergency message dissemination scheme combined with our proposed clustering algorithm UFC. When an accident happens, the emergency message should be disseminated to all of vehicles on the road. The CH uses the Store-Carry-Forward method to broadcast the emergency message periodically to its

CMs. The CMs receives the message and relays the message to its belonged CH. This proposed data dissemination scheme shows higher packet delivery ratio and lower packet delivery delay.

8.2 Perspectives

This thesis provided several important solutions to enhance vehicles' connections. As part of the future research, we would like to investigate the following aspects concerning improvements of our proposed methods.

8.2.1 Extension of UFC

Based on our proposed framework UFC, we can evaluate not only the clustering metrics that we have mentioned in Chapter 4 but also some complex and novel metrics, such as driver's behaviors, vehicle's trajectory history and social patterns (e.g. type of vehicles, including taxi, bus, private car, and trucks).

The proposed UFC algorithm can further solve the information dissemination in vehicular networks. UFC forms one-hop clusters, and CHs form the backbone of the vehicular networks. Instead of pure broadcast, CMs send information directly to the associated CHs. The CHs aggregate the received information and forward it to the destined vehicles or the related geographic regions. The transmission over the CH backbone can be in a carry-and-forward manner, to avoid expensive overhead on maintaining the end-to-end path. To reduce the information dissemination latency, it is also possible to make use of fixed infrastructures, such as RSUs. The CH can contact RSUs to disseminate information in the backbone, formed by RSUs, and let the RSU, the closest one to the destined vehicles or regions, to forward the information to the corresponding CH. For data with different delay requirements, the CH can decide whether to aggregate or not the received information.

Moreover, the further deployment of UFC can take the cluster size optimization into consideration. Vehicles can pre-compute a table of optimal cluster size subject to different combinations of road condition, the requirement of applications, packet sizes, etc., then the CH will decide on-line about the suitable cluster size based on its location condition by table checking.

8.2.2 Extension of stochastic model for clustering

The proposed stochastic model in Chapter 6 can be extended by taking into consideration the real events which cause the state transition in the Markov chain model. For instance, the state transition from UN to CCM are triggered by two events, as mentioned in Figure 4.2, therefore, we can treat the state transition from UN to CCM as two sub-state transitions. In this case, four-state transition model will be expended to a more precise and complicated N-state transition model, where N indicates the number of events that can cause state transitions.

Furthermore, based on the proposed stochastic model, we can also model the process of cluster maintenance, the stability of clusters, and cluster-based data dissemination. All of the parameters that may influence the clustering performance can be taken into consideration, such as cluster size, traffic density, channel condition, vehicle speed, and road topology.

8.2.3 Perspectives of clustering algorithms

Most of the clustering algorithms for VANETs are implemented using simulators and are evaluated under several assumptions. A real testbed will enable the evaluation of clustering algorithms in small-scale real traffic scenario. Moreover, as an effective approach to support various vehicle applications, clustering by machine learning techniques may improve clustering efficiency and accuracy.

Since clustering algorithms require high cooperation among vehicles, vehicles are required to share their personal information with neighbors. In this case, how to balance the collaboration and privacy is still an open issue. Moreover, not all of the sharing information are convinced enough, therefore, how to detect the malicious vehicles is very important during the cooperation process.

When considering vehicles' privacy and cooperation security, the performance of the clustering algorithms may be affected. For example, the detection of a malicious vehicle may cause information transmission latency. Therefore, how to balance the clustering performance and the clustering security is still an open challenge.

With the development of cellular network technologies, such as LTE and 5G, many hybrid network architectures are proposed for vehicular networks, such as LTE-D2D, LTE-V2X, and 5G-V2X systems. In these hybrid network architectures, clustering algorithms are becoming indispensable and are highly recommended to support vehicle services. However, the feasibility of these systems is still a big challenge.

In the end, in order to meet specific requirements of new ITS applications, including both safety and non-safety applications, a framework of clustering algorithms is needed which can enable various vehicle cooperation modes. Moreover, we can design a cross layer clustering algorithms which makes full use of the information of the communication environment (such as channel condition and antenna direction) and traffic scenarios (such as network topology and traffic lights).

Bibliography

- [1] 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.
- [2] Y. Z. Hassnaa Moustafa, *Vehicular Networks: Techniques, Standards, and Applications*, 2nd ed. Auerbach Publications, 2009.
- [3] ITS-Intelligent Transportation Systems. [Online]. Available: http://www.etsi.org/plugtests/ITS_CMS/Home.htm
- [4] Y. Zhuang, J. Pan, Y. Luo, and L. Cai, "Time and Location-Critical Emergency Message Dissemination for Vehicular Ad-Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 187–196, January 2011.
- [5] Y. Bi, H. Shan, X. S. Shen, N. Wang, and H. Zhao, "A Multi-Hop Broadcast Protocol for Emergency Message Dissemination in Urban Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 736–750, March 2016.
- [6] G. Rémy, S. M. Senouci, F. Jan, and Y. Gourhant, "LTE4V2X - impact of high mobility in highway scenarios," in *Global Information Infrastructure Symposium - GIIS 2011*, Aug 2011, pp. 1–7.
- [7] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multi-Hop Cluster based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2015.
- [8] "5G America V2X solution." [Online]. Available: http://www.5gamericas.org/files/2914/7769/1296/5GA_V2X_Report_FINAL_for_upload.pdf

-
- [9] J. B. Kenney, “Dedicated short-range communications (dsrc) standards in the united states,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [10] ETSI, “Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band,” *EN 302 663 v. 1.2.0*, November 2012.
- [11] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, “Lte for vehicular networking: a survey,” *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, May 2013.
- [12] T. Mangel, T. Kosch, and H. Hartenstein, “A comparison of UMTS and LTE for vehicular safety communication at intersections,” in *2010 IEEE Vehicular Networking Conference*, Dec 2010, pp. 293–300.
- [13] “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,” *ETSI EN 302 637-2 V1.3.1*, pp. 1–44, Spt 2014.
- [14] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, “Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2377–2396, Fourthquarter 2015.
- [15] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, “5G Network Slicing for Vehicle-to-Everything Services,” *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, Dec 2017.
- [16] “ETSI TR 102 638 V1.1.1, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions.”
- [17] 3GPP, “Technical Specification Group Services and System Aspects; Study on enhancement of 3GPP Support for 5G V2X Services (Release 15),” *TR 22.886 v. 15.1.0*, December 2016.
- [18] C-ITS, “C-ITS Platform Final Report Phase 1,” January 2016.
- [19] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Véque, “A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs),” *Vehicular Communications*, vol. 9, no. Supplement C, pp. 233 – 241, 2017.

- [20] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Véque, “A new mobility-based clustering algorithm for vehicular ad hoc networks (VANETs),” in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 1203–1208.
- [21] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Véque, “A Unified Framework of Clustering Approach in Vehicular Ad Hoc Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–14, 2017.
- [22] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Véque, “A Study of the Impact of Merging Schemes on Cluster Stability in VANETs,” in *2017 IEEE 28th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications - PIMRC2017*, Oct 2017.
- [23] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Véque, “A stochastic model for vehicle clustering performance analysis,” in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.
- [24] M. Chatterjee, S. K. Das, and D. Turgut, “WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks,” *Cluster Computing*, vol. 5, no. 2, pp. 193–204.
- [25] F. D. Tolba, D. Magoni, and P. Lorenz, “Connectivity, energy and mobility driven clustering algorithm for mobile ad hoc networks,” in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*. IEEE, 2007, pp. 2786–2790.
- [26] 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*, ser. MobiWac '11. New York, NY, USA: ACM, 2011, pp. 19–26.
- [27] E. Dror, C. Avin, and Z. Lotker, “Fast randomized algorithm for hierarchical clustering in vehicular ad-hoc networks,” in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, June 2011, pp. 1–8.
- [28] L. Khoukhi, A. E. Masri, A. Sardouk, A. Hafid, and D. Gaiti, “Toward Fuzzy Traffic Adaptation Solution in Wireless Mesh Networks,” *IEEE Transactions on Computers*, vol. 63, no. 5, pp. 1296–1308, May 2014.

- [29] S. Vodopivec, J. Bešter, and A. Kos, “A survey on clustering algorithms for vehicular ad-hoc networks,” in *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*, July 2012, pp. 52–56.
- [30] A. E. Masri, A. Sardouk, L. Khoukhi, A. Hafid, and D. Gaiti, “Neighborhood-Aware and Overhead-Free Congestion Control for IEEE 802.11 Wireless Mesh Networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5878–5892, Oct 2014.
- [31] 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, Firstquarter 2017.
- [32] C. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 15, no. 7, pp. 1265–1275, Sep 1997.
- [33] L. Khoukhi and S. Cherkaoui, “FuzzyMARS: a fuzzy logic approach with service differentiation for wireless ad hoc networks,” in *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 2, June 2005, pp. 839–844 vol.2.
- [34] J. Y. Yu and P. H. J. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *IEEE Communications Surveys Tutorials*, vol. 7, no. 1, pp. 32–48, First 2005.
- [35] S. Basagni, “Distributed clustering for ad hoc networks,” in *Parallel Architectures, Algorithms, and Networks, 1999. (I-SPAN '99) Proceedings. Fourth International-Symposium on*, 1999, pp. 310–315.
- [36] P. Basu, N. Khan, and T. D. Little, “A mobility based metric for clustering in mobile ad hoc networks,” in *Distributed computing systems workshop, 2001 international conference on*. IEEE, 2001, pp. 413–418.
- [37] G. Chen, F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic, “Connectivity based k-hop clustering in wireless networks,” in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, Jan 2002, pp. 2450–2459.
- [38] T. J. Kwon, M. Gerla, V. K. Varma, M. Barton, and T. R. Hsing, “Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad

- hoc/sensor networks,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1210–1220, Aug 2003.
- [39] C. Ide, F. Kurtz, and C. Wietfeld, “Cluster-Based Vehicular Data Collection for Efficient LTE Machine-Type Communication,” in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, Sept 2013, pp. 1–5.
- [40] P. Salvo, I. Turcanu, F. Cuomo, A. Baiocchi, and I. Rubin, “LTE floating car data application off-loading via VANET driven clustering formation,” in *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Jan 2016, pp. 1–8.
- [41] B. Hassanabadi, C. Shea, L. Zhang, and S. Valaee, “Clustering in vehicular ad hoc networks using affinity propagation,” *Ad Hoc Networks*, vol. 13, Part B, pp. 535 – 548, 2014.
- [42] E. Souza, I. Nikolaidis, and P. Gburzynski, “A New Aggregate Local Mobility Clustering Algorithm for VANETs,” in *Communications (ICC), 2010 IEEE International Conference on*, May 2010, pp. 1–5.
- [43] T. Taleb, A. Benslimane, and K. B. Letaief, “Toward an effective risk-conscious and collaborative vehicular collision avoidance system,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1474–1486, March 2010.
- [44] C. Shea, B. Hassanabadi, and S. Valaee, “Mobility-based clustering in vanets using affinity propagation,” in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov 2009, pp. 1–6.
- [45] Z. Wang, L. Liu, M. Zhou, and N. Ansari, “A position-based clustering technique for ad hoc intervehicle communication,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 2, pp. 201–208, 2008.
- [46] Y. Gunter, B. Wiegel, and H. P. Grossmann, “Cluster-based Medium Access Scheme for VANETs,” in *2007 IEEE Intelligent Transportation Systems Conference*, Sept 2007, pp. 343–348.
- [47] H. Su and X. Zhang, “Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3309–3323, Nov 2007.

- [48] X. Zhang, H. Su, and H. h. Chen, "Cluster-based multi-channel communications protocols in vehicle ad hoc networks," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 44–51, October 2006.
- [49] N. Maslekar, M. Boussedjra, J. Mouzna, and L. Houda, "Direction based clustering algorithm for data dissemination in vehicular networks," in *Vehicular Networking Conference (VNC), 2009 IEEE*, Oct 2009, pp. 1–6.
- [50] N. Maslekar, J. Mouzna, H. Labiod, M. Devisetty, and M. Pai, "Modified c-drive: Clustering based on direction in vehicular environment," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 845–850.
- [51] M. A. Togou, A. Hafid, and L. Khoukhi, "A Novel CDS-Based Routing Protocol for Vehicular Ad Hoc Networks in Urban Environments," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.
- [52] L. Khoukhi and S. Cherkaoui, "A quality of service approach based on neural networks for mobile ad hoc networks," in *Second IFIP International Conference on Wireless and Optical Communications Networks, 2005. WOCN 2005.*, March 2005, pp. 295–301.
- [53] O. Kayis and T. Acarman, "Clustering formation for inter-vehicle communication," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, Sept 2007, pp. 636–641.
- [54] S.-S. Wang and Y.-S. Lin, "Passcar: A passive clustering aided routing protocol for vehicular ad hoc networks," *Computer Communications*, vol. 36, no. 2, pp. 170 – 179, 2013.
- [55] N. Maslekar, M. Boussedjra, J. Mouzna, and H. Labiod, "A stable clustering algorithm for efficiency applications in vanets," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, July 2011, pp. 1188–1193.
- [56] V. Véque, F. Kaisser, C. Johnen, and A. Busson, "CONVOY: A New Cluster-Based Routing Protocol for Vehicular Networks," *Vehicular Networks*, pp. 91–129, 2013.
- [57] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 195–207, Feb 1993.

- [58] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Vehicular Communications*, vol. 2, no. 2, pp. 110 – 123, 2015.
- [59] P. Fan, J. G. Haran, J. Dillenburg, and P. C. Nelson, *Ad-Hoc, Mobile, and Wireless Networks: 4th International Conference, ADHOC-NOW 2005, Cancun, Mexico, October 6-8, 2005. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, ch. Cluster-Based Framework in Vehicular Ad-Hoc Networks, pp. 32–42.
- [60] S. Kuklinski and G. Wolny, "Density based clustering algorithm for VANETs," in *Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, April 2009, pp. 1–6.
- [61] S. Dhaou, C. Soumaya, and K. Lyes, "Multi-priority queuing for electric vehicles charging at public supply stations with price variation," *Wireless Communications and Mobile Computing*, vol. 15, no. 6, pp. 1049–1065.
- [62] S. A. Mohammad and C. W. Michele, "Using traffic flow for cluster formation in vehicular ad-hoc networks," in *IEEE Local Computer Network Conference*, Oct 2010, pp. 631–636.
- [63] A. Daeinabi, A. G. P. Rahbar, and A. Khademzadeh, "VWCA: An efficient clustering algorithm in vehicular ad hoc networks," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 207 – 222, 2011.
- [64] Y. X. Wang and F. S. Bao, "An entropy-based weighted clustering algorithm and its optimization for ad hoc networks," in *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, Oct 2007, pp. 56–56.
- [65] M. Morales, C. S. Hong, and Y. C. Bang, "An adaptable mobility-aware clustering algorithm in vehicular networks," in *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, Sept 2011, pp. 1–6.
- [66] K. A. Hafeez, L. Zhao, Z. Liao, and B. N. W. Ma, "A fuzzy-logic-based cluster head selection algorithm in vanets," in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 203–207.

- [67] M. Kakkasageri and S. Manvi, “Multiagent driven dynamic clustering of vehicles in VANETs,” *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1771 – 1780, 2012.
- [68] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [69] 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, theory, Algorithms and Applications of Wireless Networked RoboticsRecent Advances in Vehicular Communications and Networking.
- [70] M. Togou, A. Hafid, and L. Khoukhi, “SCRP: Stable CDS-Based Routing Protocol for Urban Vehicular Ad Hoc Networks,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–10, 2016.
- [71] A. Coloni, M. Dorigo, V. Maniezzo *et al.*, “Distributed optimization by ant colonies,” in *Proceedings of the first European conference on artificial life*, vol. 142. Paris, France, 1991, pp. 134–142.
- [72] R. R. Sahoo, R. Panda, D. K. Behera, and M. K. Naskar, “A trust based clustering with ant colony routing in vanet,” in *Computing Communication Networking Technologies (ICCCNT), 2012 Third International Conference on*, July 2012, pp. 1–8.
- [73] M. Fathian, G. R. Shiran, and A. R. Jafarian-Moghaddam, “Two new clustering algorithms for vehicular ad-hoc network based on ant colony system,” *Wireless Personal Communications*, vol. 83, no. 1, pp. 473–491, 2015.
- [74] M. Fathian and A. R. Jafarian-Moghaddam, “New clustering algorithms for vehicular ad-hoc network in a highway communication environment,” *Wireless Networks*, vol. 21, no. 8, pp. 2765–2780, 2015.
- [75] W. Li, A. Tizghadam, and A. Leon-Garcia, “Robust clustering for connected vehicles using local network criticality,” in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 7157–7161.
- [76] 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)*, April 2013, pp. 2381–2386.

- [77] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO—simulation of urban mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.
- [78] 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, pp. 1–12, 2015.
- [79] 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, pp. 1–13, 2012.
- [80] H. R. Arkian, R. E. Atani, A. Pourkhalili, and S. Kamali, "Cluster-based traffic information generalization in Vehicular Ad-hoc Networks," *Vehicular Communications*, vol. 1, no. 4, pp. 197 – 207, 2014.
- [81] Y. Allouche and M. Segal, "Cluster-Based Beaconing Process for VANET," *Vehicular Communications*, vol. 2, no. 2, pp. 80 – 94, 2015.
- [82] F. Chiti, R. Fantacci, and G. Rigazzi, "A mobility driven joint clustering and relay selection for IEEE 802.11p/WAVE vehicular networks," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 348–353.
- [83] S. Vodopivec, J. Bešter, and A. Kos, "A multihoming clustering algorithm for vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.
- [84] L. A. Maglaras and D. Katsaros, "Social Clustering of Vehicles Based on Semi-Markov Processes," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 318–332, Jan 2016.
- [85] R. T. Goonewardene, F. H. 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, pp. 148–158, June 2009.
- [86] R. A. Santos, R. M. Edwards, and A. Edwards, "Cluster-based location routing algorithm for vehicle to vehicle communication," in *Proceedings. 2004 IEEE Radio and Wireless Conference (IEEE Cat. No.04TH8746)*, Sept 2004, pp. 39–42.
- [87] A. M. Vegni, A. Stramacci, and E. Natalizio, "SRB: A Selective Reliable Broadcast protocol for safety applications in VANETs," in *Mobile and Wireless Networking*

- (*iCOST*), *2012 International Conference on Selected Topics in*, July 2012, pp. 89–94.
- [88] B. Zhang, K. Xing, X. Cheng, L. Huang, and R. Bie, “Traffic clustering and on-line traffic prediction in vehicle networks: A social influence perspective,” in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 495–503.
- [89] L. A. Maglaras and D. Katsaros, “Enhanced spring clustering in vanets with obstruction considerations,” in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, June 2013, pp. 1–6.
- [90] I. Tal and G. M. Muntean, “User-Oriented Fuzzy Logic-Based Clustering Scheme for Vehicular Ad-Hoc Networks,” in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, June 2013, pp. 1–5.
- [91] L. A. Maglaras and D. Katsaros, “Distributed clustering in vehicular networks,” in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2012, pp. 593–599.
- [92] G. Wolny, “Modified dmac clustering algorithm for vanets,” in *2008 Third International Conference on Systems and Networks Communications*, Oct 2008, pp. 268–273.
- [93] Q. Luo, C. Li, Q. Ye, T. H. Luan, L. Zhu, and X. Han, “CFT: A Cluster-based File Transfer Scheme for Highway,” *ArXiv e-prints*, Jan. 2017.
- [94] M. S. Almalag, S. Olariu, M. C. Weigle, and S. El-Tawab, “Peer-to-peer file sharing in VANETs using TC-MAC,” in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, March 2013, pp. 84–89.
- [95] H. Venkataraman, R. Delcelier, and G.-M. Muntean, “A moving cluster architecture and an intelligent resource reuse protocol for vehicular networks,” *Wireless Networks*, vol. 19, no. 8, pp. 1881–1900, 2013.
- [96] P. Fan, “Improving broadcasting performance by clustering with stability for inter-vehicle communication,” in *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, April 2007, pp. 2491–2495.

- [97] G. Rémy, S. M. Senouci, F. Jan, and Y. Gourhant, “LTE4V2X - Collection, dissemination and multi-hop forwarding,” in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 120–125.
- [98] S. T. Cheng, G. J. Horng, and C. L. Chou, “Using Cellular Automata to Form Car Society in Vehicular Ad Hoc Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1374–1384, Dec 2011.
- [99] J. Luo, X. Gu, T. Zhao, and W. Yan, “MI-VANET: A New Mobile Infrastructure Based VANET Architecture for Urban Environment,” in *2010 IEEE 72nd Vehicular Technology Conference - Fall*, Sept 2010, pp. 1–5.
- [100] S.-S. Wang and Y.-S. Lin, “Performance evaluation of passive clustering based techniques for inter-vehicle communications,” in *The 19th Annual Wireless and Optical Communications Conference (WOCC 2010)*, May 2010, pp. 1–5.
- [101] N. Kumar, N. Chilamkurti, and J. H. Park, “ALCA: Agent Learning—based Clustering Algorithm in Vehicular Ad Hoc Networks,” *Personal Ubiquitous Comput.*, vol. 17, no. 8, pp. 1683–1692, Dec. 2013.
- [102] T. Gazdar, A. Benslimane, and A. Belghith, “Secure Clustering Scheme Based Keys Management in VANETs,” in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, May 2011, pp. 1–5.
- [103] A. Ahizoune and A. Hafid, “A new stability based clustering algorithm (SBCA) for VANETs,” in *37th Annual IEEE Conference on Local Computer Networks - Workshops*, Oct 2012, pp. 843–847.
- [104] K. Ibrahim and M. C. Weigle, “CASCADE: Cluster-Based Accurate Syntactic Compression of Aggregated Data in VANETs,” in *2008 IEEE Globecom Workshops*, Nov 2008, pp. 1–10.
- [105] S. Dietzel, A. Peter, and F. Kargl, “Secure Cluster-Based In-Network Information Aggregation for Vehicular Networks,” in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–5.
- [106] S. Khakpour, R. W. Pazzi, and K. El-Khatib, “A prediction based clustering algorithm for target tracking in vehicular ad-hoc networks,” in *Proceedings of the Fourth ACM International Symposium on Development and Analysis of Intelligent Vehic-*

- ular Networks and Applications*, ser. DIVANet '14. New York, NY, USA: ACM, 2014, pp. 39–46.
- [107] S. Khakpour, R. W. Pazzi, and K. El-Khatib, “Using clustering for target tracking in vehicular ad hoc networks,” *Vehicular Communications*, vol. 9, pp. 83 – 96, 2017.
- [108] 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.
- [109] G. Remy, S. M. Senouci, F. Jan, and Y. Gourhant, “LTE4V2X: LTE for a Centralized VANET Organization,” in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Dec 2011, pp. 1–6.
- [110] G. e. m. 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, Feb 2015.
- [111] A. Benslimane, T. Taleb, and R. Sivaraj, “Dynamic Clustering-Based Adaptive Mobile Gateway Management in Integrated VANET - 3G Heterogeneous Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 559–570, March 2011.
- [112] L. C. Tung, J. Mena, M. Gerla, and C. Sommer, “A cluster based architecture for intersection collision avoidance using heterogeneous networks,” in *2013 12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, June 2013, pp. 82–88.
- [113] J. Garbiso, A. Diaconescu, M. Coupechoux, and B. Leroy, “Dynamic cluster size optimization in hybrid cellular-vehicular networks,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 557–563.
- [114] Q. Zheng, K. Zheng, L. Sun, and V. C. M. Leung, “Dynamic Performance Analysis of Uplink Transmission in Cluster-Based Heterogeneous Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5584–5595, Dec 2015.
- [115] Y. Guenter, B. Wiegel, and H. P. Grossmann, “Medium Access Concept for VANETs Based on Clustering,” in *2007 IEEE 66th Vehicular Technology Conference*, Sept 2007, pp. 2189–2193.

- [116] F. Yang, Y. Tang, and L. Huang, "A multi-channel cooperative clustering-based MAC protocol for VANETs," in *2014 Wireless Telecommunications Symposium*, April 2014, pp. 1–5.
- [117] 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, Oct 2013.
- [118] M. S. Almalag, S. Olariu, and M. C. Weigle, "TDMA cluster-based MAC for VANETs (TC-MAC)," in *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2012, pp. 1–6.
- [119] L. Bononi and M. di Felice, "A Cross Layered MAC and Clustering Scheme for Efficient Broadcast in VANETs," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, Oct 2007, pp. 1–8.
- [120] T. Taleb and A. Benslimane, "Design Guidelines for a Network Architecture Integrating VANET with 3G and beyond Networks," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec 2010, pp. 1–5.
- [121] P. Fan, A. Mohammadian, P. C. Nelson, J. Haran, and J. Dillenburg, "A novel direction-based clustering algorithm in vehicular ad hoc networks," in *Transportation research board 86th annual meeting*, no. 07-1673, 2007.
- [122] R. Crepaldi, M. Bakht, and R. Kravets, "QuickSilver: Application-driven Inter- and Intra-cluster Communication in Vanets," in *Proceedings of the Third ACM International Workshop on Mobile Opportunistic Networks*, ser. MobiOpp '12. New York, NY, USA: ACM, 2012, pp. 69–76.
- [123] S. Maerivoet and B. De Moor, "Traffic Flow Theory," *ArXiv Physics e-prints*, Jul. 2005. [Online]. Available: <http://adsabs.harvard.edu/abs/2005physics...7126M>
- [124] R. Stanica, E. Chaput, and A.-L. Beylot, "Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations," *Computer Networks*, vol. 55, no. 14, pp. 3179 – 3188, 2011, deploying vehicle-2-x communication.
- [125] The Network Simulator - ns-2. [Online]. Available: <http://isi.edu/nsnam/ns/>
- [126] O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Worm Epidemics in Vehicular Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2173–2187, Oct 2015.

- [127] K. Abboud and W. Zhuang, "Stochastic analysis of a single-hop communication link in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2297–2307, Oct 2014.
- [128] K. Abboud and W. Zhuang, "Stochastic Modeling of Single-Hop Cluster Stability in Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 226–240, Jan 2016.
- [129] H. Wang, R. P. Liu, W. Ni, W. Chen, and I. B. Collings, "VANET Modeling and Clustering Design Under Practical Traffic, Channel and Mobility Conditions," *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 870–881, March 2015.

List of publications

Peer-reviewed international journals

1. **M. Ren**, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, “Clustering algorithms in Vehicular Ad-hoc Networks: A comprehensive survey,” submitted.
2. **M. Ren**, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, “A Unified Framework of Clustering Approach in Vehicular Ad-hoc Networks,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1-14. 2017.
3. **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*, Volume 9, 2017, Pages 233-241, ISSN 2214-2096.

Peer-reviewed international conferences with proceedings

1. **M. Ren**, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, “A stochastic model for vehicle clustering performance analysis,” in *IEEE International Conference on Communications (ICC), Kansas City, 2018*. IEEE. (accepted)
2. **M. Ren**, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, “A Study of the Impact of Merging Schemes on Cluster Stability in VANETs,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Montreal, 2017*. IEEE.
3. **M. Ren**, L. Khoukhi, H. Labiod, J. Zhang, and V. Vèque, “A new mobility-based clustering algorithm for vehicular ad hoc networks (VANETs),” in *Network Operations and Management Symposium (NOMS), Istanbul, 2016 IEEE/IFIP*, (pp.

1203-1208). IEEE.

4. J. Zhang, **M. Ren**, H. Labiod and L. Khoukhi, "Link Duration Prediction in VANETs via AdaBoost," in *IEEE GLOBECOM, Singapore, 2017*, IEEE.
5. J. Zhang, **M. Ren**, H. Labiod and L. Khoukhi, "Every dog has its day: A comparative study of clustering algorithms in VANETs," in *IEEE Symposium on Computers and Communications (ISCC), Heraklion, 2017*, pp. 383-389.

Résumé Français

I Introduction

Un réseau ad-hoc véhiculaires ou VANET (Vehicular Ad hoc Network) est un réseau auto-organisé constitué d'une collection de véhicules communicants en mouvement [1]. Les systèmes de transport intelligents (ITS, Intelligent Transport System) visent deux objectifs principaux : la sécurité des transports, la fluidité du trafic routier. Dans ce cadre, de nouvelles fonctionnalités ont été ajoutées aux véhicules comme la perception de l'environnement grâce à de nombreux capteurs, la communication avec les autres véhicules ainsi que des centres de contrôles ainsi que la synthèse de ces informations et la réaction à des événements grâce à un ordinateur embarqué. Pour cela, les constructeurs développent des véhicules de plus en plus intelligents. Les véhicules peuvent communiquer directement avec d'autres véhicules grâce à des technologies sans fil V2V (véhicule à véhicule) ou indirectement via des infrastructures routières en V2I (véhicule à infrastructure) ou I2V (infrastructure à véhicule) [1]. De plus, les véhicules peuvent également se connecter aux piétons via la communication V2P (Véhicule à Piétons). En parallèle, le développement de technologies cellulaires, telles que Long-Term Evolution (LTE), permet aux véhicules de communiquer directement avec le réseau cellulaire. Collectivement, ces connexions sans fil sont désignées communications V2X (véhicule-à-tout). Elles peuvent prendre en charge de nombreux cas d'utilisation des systèmes de transport intelligents coopératifs (C-ITS), comme les services liés à la sécurité ou non. La Figure B.1 présente les futurs systèmes de transport intelligents proposés par l'Institut Européen des Normes de Télécommunication (ETSI) [3].

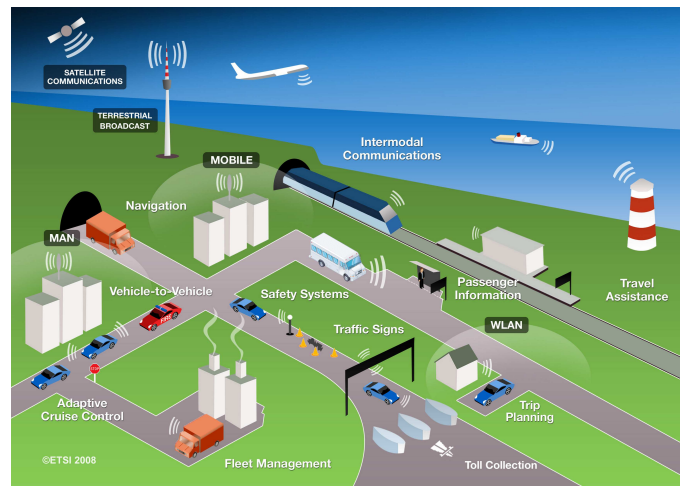


Figure B.1: STI-Systèmes de Transport Intelligents

I.a Caractéristiques et défis des VANET

Par rapport au réseau mobile ad hoc traditionnel (MANET), les caractéristiques et les défis spécifiques aux VANET ont reçu l'attention de nombreux chercheurs ou industriels de différents domaines pour étudier les applications, les technologies, les protocoles et les normes de ces réseaux. En tant que MANET typique, VANET hérite des caractéristiques similaires de MANET; cependant, ces réseaux présentent également des spécificités :

- Mobilité prévisible des véhicules : la mobilité des véhicules est limitée par les structures routières, les conditions de circulation, les feux de circulation et les panneaux routiers.
- Différents modèles de mobilité des véhicules : un VANET se caractérise par des modèles de mobilité variables, en fonction du type de véhicule. Ainsi, les vitesses des camions, des voitures et des motos sont généralement différentes.
- Topologie très dynamique du réseau : En raison de la vitesse rapide des véhicules sur la route, la topologie du réseau ad hoc change rapidement. Un véhicule roulant à une vitesse élevée peut dépasser rapidement le véhicule qui le précède, ce qui provoque l'instabilité des connexions inter-véhicules. Par conséquent, la communication entre ces véhicules sera coupée.
- Réseau sensible à la densité de véhicules : La nuit en ville ou sur une autoroute, la circulation est réduite et la distance entre les véhicules peut être trop grande pour assurer la continuité du réseau. Au contraire, dans un embouteillage, si tous les véhicules sont en communication, le canal radio sera congestionné. Par conséquent,

le réseau n'est pas connexe quand la densité de véhicules est faible et au contraire saturé, si la densité est élevée.

- Échelle de réseau illimitée : La taille du réseau peut être énorme dans les VANET, par exemple, une grande ville. En l'absence d'un contrôleur central dans le VANET, la plage de transmission limitée ne peut prendre en charge que les communications de véhicule à courte portée, ce qui n'est pas suffisant pour prendre en charge certains services VANET.

I.b Technologies

Les communications entre véhicules, Roadside Unit (RSU) et infrastructures sont connues sous le nom de la communication véhicule-à-tout (V2X). La communication V2X peut être prise en charge par de nombreuses technologies d'accès sans fil. Certaines de ces technologies de communication prennent en charge les communications de manière distribuée (V2V). En revanche, d'autres technologies se basent sur une infrastructure centralisée dans HetVNET (Cellular-V2X).

Les communications V2V: La communication V2V permet à des véhicules voisins géographiquement d'échanger et de partager des informations directement sans infrastructure. Par conséquent, les messages d'urgence peuvent être délivrés avec un délai plus court. Le standard américain de communications dédiées à courte portée (DSRC) a été développé pour ce type de communications [9]. DSRC est basé sur le protocole IEEE 802.11p, qui est proposé particulièrement pour la communication V2V.

Les communications Device-to-Device (D2D) sont l'une des techniques qui supportent la communication V2V dans HetVNET [14]. Dans LTE-D2D, les équipements utilisateur (UE) voisins peuvent communiquer directement entre eux. Cependant, les équipements doivent découvrir leur pair avant toutes les communications directes. Dans ce cas, le processus de découverte peut augmenter la latence ce qui n'est pas acceptable pour la transmission de messages d'urgence. Par conséquent, DSRC est plus approprié aux communications V2V.

Les communications Cellular-V2X (C-V2X): La communication C-V2X permet la connexion avec les réseaux cellulaires. Le réseau cellulaire peut fournir une couverture plus importante et un débit de données plus élevé. Dans ce cas, plus de services peuvent être supportés par le réseau cellulaire. Actuellement, les technologies cellulaires les plus populaires sur lesquelles les chercheurs se concentrent sont LTE et la cinquième Génération (5G), aussi appelées respectivement LTE-V2X [11] et 5G-V2X [15].

I.c Applications

Les applications C-ITS peuvent être classées en trois catégories, la sécurité routière, l'efficacité du trafic et les applications de divertissement. Les applications de sécurité routière visent à éviter les accidents de la circulation et à protéger la vie des usagers de la route. Les applications d'efficacité du trafic se concentrent sur l'amélioration de l'efficacité du trafic, par exemple en évitant les embouteillages qui induisent pollution et perte de temps. Enfin, les applications de divertissement visent à offrir des trajets plus confortables et pratiques aux usagers de la route. La description détaillée de chaque type d'application est présentée comme suit.

- Applications de sécurité routière: Ces applications peuvent détecter et recueillir des informations d'urgence à partir des capteurs du véhicule. Les messages d'urgence doivent être transmis ou diffusés aux véhicules dans une zone cible avec une courte latence et une grande précision.
- Applications d'efficacité du trafic: Ce type d'applications nécessitent généralement une plus grande portée. Par rapport aux applications de sécurité ; la latence et la précision de la transmission sont moins contraignants.
- Applications de divertissement: Ces applications sont prises en charge par les différentes technologies de communication. Les applications, telles que l'accès Internet, le téléchargement de contenus, le partage de l'information, nécessite une large bande passante et une grande capacité de stockage.

I.d Motivations et contributions

Parmi les applications C-ITS mentionnées ci-dessus, l'application de sécurité routière est la plus critique en temps et la plus difficile à satisfaire. Pour ce type d'applications, la transmission de l'information nécessite généralement une courte latence, une grande précision et une grande fiabilité. De plus, le problème de la dymanicité des VANETs ne peut pas être résolu par une architecture de réseau plate sans contrôleur central.

Dans le réseau plate sans contrôleur central, les communications sont très limitées dans une zone et les messages urgents ne peuvent pas être envoyé rapidement aux véhicules éloignés. Ainsi, les chercheurs ont proposé une architecture de réseau hiérarchique pour résoudre ce problème, dans lequel les véhicules sont virtuellement organisés en différents groupes, appelés **clusters**. Cet architecture peut résoudre le problème de la limitation

d'échelle de réseau. Par ailleurs, les connections entre véhicules dans chaque groupe peuvent être renforcées en utilisant de clusters.

Dans chaque groupe de véhicules, il existe un chef et un ou plusieurs membres. Le chef est responsable de la communication entre les groupes (inter-clusters), et les membres du groupe peuvent communiquer directement avec leur chef [1]. La topologie de réseau de véhicules étant très dynamique, cela peut provoquer l'instabilité des groupes de véhicules. C'est l'inconvénient majeur de la hiérarchisation car cela entraîne des échanges de messages supplémentaires pour maintenir la structure. Dans ce cas, il faut trouver une méthode efficace pour construire des groupes de véhicules stables. Notre premier objectif dans cette thèse est de *concevoir un algorithme de clustering dans les VANETs afin de garantir la stabilité du cluster*.

En fonction de l'objectif de la hiérarchisation, des métriques à optimiser ont été proposées pour construire des clusters et ont été prouvées être très efficaces. Par exemple, la vitesse, la direction, et la distance relative.

Cependant, il est assez difficile d'identifier quelle métrique de clustering fonctionne mieux pour la stabilité du cluster. De plus, étant donné que les algorithmes de clustering sont proposés selon différents scénarios, il est difficile d'évaluer équitablement ces algorithmes. Dans ce cas, le deuxième objectif de cette thèse est de *concevoir un framework générique servant à définir des algorithmes de clustering dans les VANETs*, afin d'évaluer équitablement les différentes métriques et d'analyser leur impact.

Nous avons observé que le processus de maintenance de cluster joue un rôle important pour assurer la stabilité du cluster. En tant que partie indispensable du processus de maintenance de cluster, la méthode de fusion de cluster peut avoir une certaine influence sur les performances. Cependant, la méthode de fusion de cluster n'a pas été toujours considérée. Sur la base de notre framework proposé, cette thèse *évalue également les impacts des schémas de fusion de cluster sur les performances de clustering et conçoit un nouveau schéma de fusion de clusters*. Pour comprendre en profondeur l'algorithme de clustering, un autre objectif de cette thèse est de *proposer un modèle stochastique pour l'analyse des performances de cluster*.

Comme mentionné ci-dessus, les applications de sécurité routière dans les VANET dépendent principalement de la diffusion d'informations aux véhicules concernés. Cependant, la diffusion d'informations souffre du problème de tempête de diffusion et des interférences entre les messages périodiques de beacons échangés à 1-saut. L'architecture de réseau hiérarchique, basée sur les clusters, peut effectivement résoudre ces problèmes

en évitant les transmissions de messages redondants. Les méthodes de mise en cache de contenu permettent le stockage temporaire d'informations moins urgentes afin que les messages urgents puissent être envoyés le plus rapidement possible. Le dernier objectif de cette thèse est de *proposer un schéma de la dissémination des données sur les clusters*.

En résumé, l'objectif principal de cette thèse est de proposer des algorithmes de clustering efficaces pour supporter une transmission de données fiable dans les VANETs.

II État de l'art

Ce chapitre passe en revue les développements d'algorithmes de clustering pour les VANETs et résume les algorithmes de clustering existants ainsi que les algorithmes de diffusion de données basée sur le cluster à partir des aspects suivants : méthodes de clustering, algorithmes de clustering contextuels et évaluation des performances de cluster. Une nouvelle classification des algorithmes de clustering est proposée en fonction du contexte. De plus, ce chapitre présente une analyse complète des méthodes d'évaluation des performances des algorithmes de clustering, y compris les métriques de performance, les outils de simulation, les scénarios de trafic, etc.

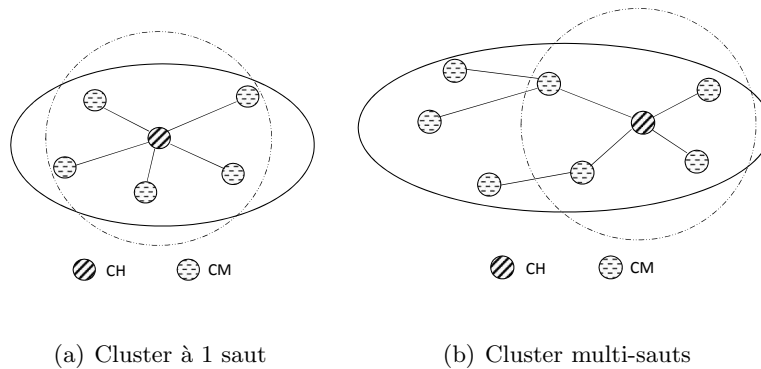


Figure B.2: Topologies de cluster

II.a Histoire des algorithmes de clustering

Chaque cluster ou groupe virtuel de véhicules a un chef de cluster (CH-Clusterhead). Un CH est suivi par plusieurs membres du cluster (CM-Cluster Member). La topologie de cluster peut être classée en deux catégories: clusters 1 saut et clusters multi-sauts, comme le montre la Figure B.2. Généralement, on construit un cluster 1-saut sur la base de la couverture radio de CH (TR). Chaque CM peut communiquer directement avec son

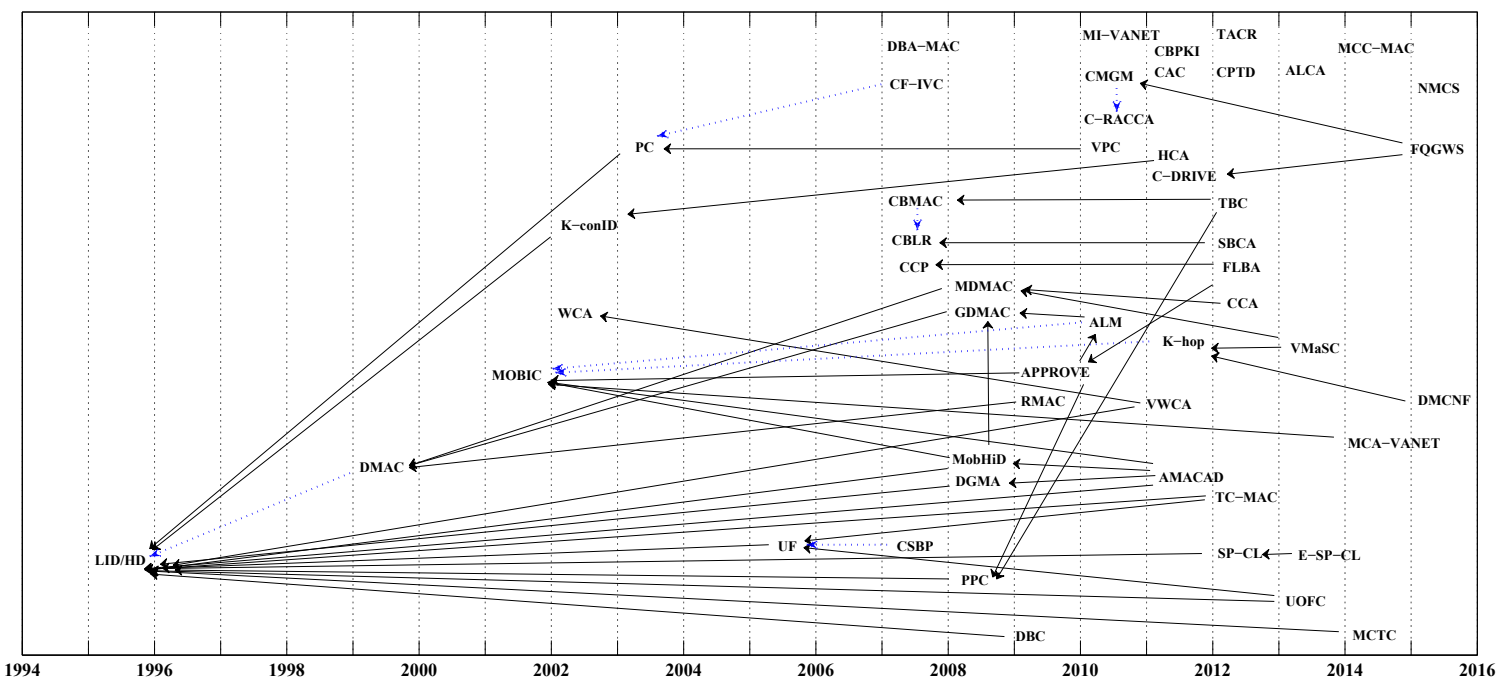


Figure B.3: Le developement des algorithmes de clustering dans les VANETs. (Couleur bleu: sans comparaison; Couleur noir: avec comparaison.)

CH via une communication V2V. Deux CM différents peuvent communiquer entre eux directement ou via leur CH. Ces dernières années, plusieurs clusters multi-sauts ont été proposés afin de construire des clusters plus stables.

Les techniques de clustering VANETs ont d'abord été utilisés dans les MANET au début des années 1990, et ont été améliorés pour les VANET ces dernières années. La Figure B.3 présente un plan pour le développement des algorithmes de clustering proposés depuis le début des années 1990 à l'année 2017. L'axe des abscisses indique l'année, et l'ordonnée montre les algorithmes proposés. Dans cette figure, on indique les relation entre ces algorithmes existants. La flèche noire indique que l'algorithme suivant est comparé à l'algorithme original dans la simulation, et la flèche bleue montre que l'algorithme suit le cadre original mais sans comparaison.

II.b Mécanismes de clustering dans VANETs

La construction de cluster de véhicules est une procédure dynamique en raison de la grande mobilité des véhicules et des communications intermittentes. Les véhicules doivent obtenir les informations de leurs voisins, y compris leur identité, position, vitesse, etc. Le CH potentiel sera sélectionné en fonction de critères, tels que la mobilité relative, l'intensité du signal reçu, la durée de vie de lien, etc. Les clusters sont établis par l'ajout de membres

potentiels. Dans [31], un flux de l'algorithme de clustering a été introduit, indiquant les étapes générales.

II.b1 Approches de sélection du CH

Dans l'algorithme de clustering, la partie la plus importante est la sélection d'un CH qui est capable d'atteindre la plus grande stabilité parmi ses voisins. Dans la plupart des algorithmes de clustering, les CH sont sélectionnés au début de la construction du cluster. Ensuite, le processus de formation de cluster est contrôlé par les CH sélectionnés.

Au lieu d'utiliser une seule métrique au début, comme l'identifiant du nœud dans l'approche Lowest-ID (LID) [32], de plus en plus de méthodes de sélection de CH combinent plusieurs métriques, comme la vitesse relative, la distance relative, le nombre des voisins et le lien entre deux véhicules. Par ailleurs, la stratégie de sélection de CH dépend également du contexte.

II.b2 Approches de formation du cluster

Les clusters sont formés en fonction de plusieurs critères prédéfinis, y compris la couverture radio le rayon du cluster, le nombre maximum de membres, etc. Bien que les critères de formation du cluster varient dans chaque algorithme, il n'y a que deux topologies de cluster: cluster 1-saut et cluster multi-sauts.

Single-hop: La majorité des algorithmes de clustering établissent des clusters à 1-saut. La formation du cluster dépend de l'information de véhicules qui communiquent directement entre eux. La topologie de ces clusters permet de réduire le délai de formation de cluster et le délai de gestion du cluster, car moins d'échanges d'informations sont nécessaires.

Multi-hop: Il est facile d'observer que la topologie de cluster multi-sauts devient la tendance dans la conception de cluster. La structure des clusters multi-hop est plus stable avec moins de déconnexions et de ré-affiliation. De plus, certaines applications nécessitent une transmission d'informations à grande échelle, et une topologie de cluster multi-hop peut augmenter l'efficacité de la transmission d'informations.

II.b3 Approches de maintenance de cluster

En raison de la topologie dynamique des réseaux VANET, les déconnexions et les re-connexions fréquentes entre les véhicules dans le cluster peuvent provoquer une importante perte de paquets. La maintenance du cluster est indispensable pour réduire les fréquents

de re-clustering et obtenir des clusters plus stables. La maintenance de cluster inclut le départ du véhicule, l'assemblage du véhicule et la fusion des clusters.

Pendant le processus de clustering, chaque véhicule diffuse périodiquement des messages *Beacon* pour informer leurs voisins de la mise à jour des informations. Quand le CH perd la connexion avec un CM, il supprime les informations de ce membre. La voiture déconnectée du cluster cherche à trouver un nouveau cluster à se joindre.

Comparé au processus de départ et d'assemblage du véhicule, le processus de fusion de clusters est plus complexe. En raison du modèle de mobilité dynamique dans les VANET, le processus de fusion de cluster est indispensable et se produit fréquemment. La fusion de cluster se produit lorsque deux ou plusieurs clusters sont proches et peuvent être regroupés dans un seul grand cluster, ce qui conduit à réduire le nombre de clusters. Habituellement, la fusion de clusters est provoquée lorsque deux CHs se rapprochent et deviennent des voisinages à 1-saut. Afin de garantir la stabilité de cluster après la fusion et réduire la fréquence de re-clustering, la plupart des méthodes de fusion de clusters exigent que deux CHs voisins restent dans la distance de transmission (TR) l'un de l'autre pendant une période, définie comme l'intervalle de fusion (MI), au lieu de commencer la fusion de cluster immédiatement.

II.c Classification des algorithmes de clustering

Nous proposons une nouvelle classification des algorithmes de clustering en fonction des différents contextes. En dehors des algorithmes de clustering qui sont indépendants de l'infrastructure, les algorithmes de clustering hybride dans HetVNET deviennent importants en raison du développement du réseau cellulaire.

II.c1 Clustering pour les applications basées sur le contexte

Ces dernières années, plusieurs mécanismes de clustering ont été appliqués pour des applications spécifiques de VANET. Les nœuds de cluster sont traités comme des nœuds du backbone pour la dissémination des données dans les VANET. Dans ce cas, la définition d'un algorithme de clustering doit être basée sur un contexte précis. Le Tableau B.1 montre une classification des algorithmes de clustering en fonction de leur contexte.

II.c2 Cluster hybride

La plupart des algorithmes de clustering existants sont basés sur la création de clusters dynamiques de manière décentralisée pour auto-organiser un réseau véhiculaire non

Table B.1: Classification des algorithmes de clustering*

Context		Algorithm	Year	Cluster performance		Network performance
				Macroscopic	Microscopic	
Application	Pure clustering	DMCNF[78]	2015	1, 2, 3	6	11, 16
		E-SP-CL[89]	2013	1, 3	7	
		UOFC[90]	2013	1, 3	6	
		VMaSC-1[76]	2013	1, 2,	6	16
		CCA[75]	2012	1, 2, 3, 4	6, 7	
		SP-CL[91]	2012	1, 3	7	
		TBC-2[79]	2012	1, 3	7	16
		FLBA[66]	2012	1, 2, 4		
		K-hop[26]	2011	1, 2	6	
		ALM[42]	2010	1, 3	7	
		DBC[60]	2009	2, 3, 4	6	
		MDMAC[92]	2008	3	6	16
		PPC[45]	2008			15
		AMACAD-2[65]	2011	1, 2, 3	10	
	APROVE-2[44]	2009	1, 2, 3	6	16	
	UF[59]	2005		6		
	HCA[27]	2011	1, 3	7		
	Information transmission	CB-BDP[81]	2015			12
		CFT[93]	2017	2, 4		15
		SFC[84]	2016	1, 3	7	
		MCTC[82]	2014	1, 3, 4	9	
		MCA-VANET[83]	2014		6	
		TC-MAC-3[94]	2013			
		MCMF[95]	2013			14, 15
		CSBP[96]	2007			14
		C-DRIVE[49]	2009			13
		LTE4V2X-3[97]	2012			12, 13
	Routing protocol	CONVOY[56]	2013	1, 3	7, 9	
		PassCAR[54]	2013	3		12, 15
		TACR[72]	2012			12, 16
		CAC[98]	2011			12, 13, 15, 16
		MI-VANET[99]	2010			12, 15
		VPC[100]	2010			12, 14, 15
		RMAC[85]	2009	1		10
		CBLR[86]	2004			12, 14, 16
	PC[38]	2003			14, 15	
	Traffic density estimation	MC-DRIVE[50]	2011	3		16
		CB-TIG[80]	2014	1, 3,		
	Security	ALCA[101]	2013	1, 2, 5		15
		VWCA[63]	2011	1, 2		12
		CBPKI[102]	2011	1, 3, 4, 5		
	Traffic safety	SRB[87]	2012			12, 14, 15
		C-RACCA[43]	2010			13, 14
	QoS	SBCA[103]	2012	1		12, 16
		CCP[48][47]	2006, 2007	1, 4		13, 14, 15, 16
	Aggregation	CASCADE[104]	2015			12, 14, 15
		SCB-INIA[105]	2015	4		16
Target tracking	PBC-TT[106][107]	2014, 2017	1, 2,	6	12, 14, 16	
Topology discovery	CPTD[108]	2012	3		11, 16	
Traffic prediction	TC-OTP[88]	2012	3			
Hybrid clustering	Floating Car Data (FCD)	FCDOC[40]	2016			14
		GC-VDB[39]	2013	1, 3, 4		
		LTE4V2X[109][6]	2011			13, 16
	Gateway selection	FQGwS[110]	2015			13, 14, 15, 16
		CMGM[111]	2011			12, 13, 14, 15, 16
	Data transmission	VMaSC-LTE[7]	2015	1, 2, 3	6	12, 14, 16
		LTE4V2X[97]	2012			13, 16
	Collision avoidance	CA-ICA[112]	2013			12, 14
Cluster size optimization	DCSO[113]	2016	4		12	
Uplink transmission	C-HetVNETs[114]	2015			13, 14	
MAC protocol		CBMAC[115][46]	2007	1		12
		MCC-MAC[116]	2014			13
		DMMAC[117]	2013	1, 2, 4		14, 16
		TC-MAC-2[118]	2012			13
		DBA-MAC[119]	2007			13, 14
	CCP[48][47]	2006, 2007	1, 4		13, 14, 15, 16	

Les chiffres de ce Tableau indiquent les mesures de performance correspondantes dans le Tableau B.3.

hétérogène. Dans un environnement très dynamique, le clustering décentralisé n'est pas approprié car il crée une surcharge de contrôle importante au sein du réseau. Au cours de ces dernières années, les chercheurs ont commencé à se concentrer sur le type de communication V2X avec l'aide de l'infrastructure cellulaire au lieu des communications V2V. Le Tableau B.2 résume les architectures de cluster hybrides existantes.

Table B.2: Architectures de cluster hybrides

Algorithm	Year	Application	Radius	V2V link	V2I link	Network Simulator	Traffic Simulator	Traffic Scenario
FCDOC [40]	2016	Floating Car Data application off-loading	1-hop	802.11p	LTE	OMNET++, Veins	SUMO, OpenStreetMap	City map of Rome and New York
DCSO [113]	2016	Cluster size optimization to reduce packet loss	N-hop	802.11p	LTE	OMNET++, Veins	SUMO	Highway
VMaSC-LTE [7]	2015	Safety message dissemination	N-hop	802.11p	LTE	NS3	SUMO	Straight road
FQGwS [110]	2015	Gateway selection	1-hop	802.11p	LTE-A	NS2	VanetMobiSim	Multiple-lane highway
C-HetVNETs [114]	2015	A framework for performance analysis	1-hop	802.11p	LTE	N/A	N/A	Urban with intersections
GC-VDB [39]	2013	Data collection	1-hop	802.11p	LTE	OPNET	OpenStreetMap SUMO	Highway & urban
CA-ICA [112]	2013	Intersection collision avoidance	1-hop	802.11b	LTE	NS3	VanetMobiSim	Urban with intersections
CMGM [120][111]	2011	Gateway selection	1-hop	802.11p	UMTS	NS2	N/A	Highway & urban
LTE4V2X [97]	2012	Data collection & data dissemination	N-hop	802.11p	LTE	NS3	VanetMobiSim	Highway
LTE4V2X [109][6]	2011	Floating Car Data (FCD)	1-hop	802.11p	LTE	NS3	VanetMobiSim	Urban

On peut observer que la plupart des travaux de recherche combinent le standard IEEE 802.11p avec l'architecture cellulaire LTE, où l'interface IEEE 802.11p est utilisée pour la communication V2V et l'interface LTE est utilisée pour la communication V2I. Le CH est sélectionné par la station de base. L'information d'application est transmise de la station de base aux CH, et les CH diffusent l'information à leurs membres. D'un autre côté, les CH prennent et recueillent des données de leurs CM. Ensuite, les CH transmettent l'information agrégée à la station de base. Les approches de clustering hybride, résumées dans le Tableau B.2, servent à différentes applications, notamment la collecte de données de trafic (FCD), la diffusion des données, la sélection de passerelle ou la prévention des accidents.

II.d Évaluation des performances

L'évaluation des performances des algorithmes de clustering dans les VANET est principalement basée sur des simulateurs de réseau en raison de la limitation des échelles de test dans les environnements de trafic réels. Toutefois, les algorithmes de clustering existants ont été évalués en fonction de différentes hypothèses, notamment les scénarios de trafic,

les modèles de canaux, les modèles de mobilité des véhicules, etc. Les critères d'évaluation étant différents, par conséquent, il est difficile de comparer ces divers algorithmes. Le reste de ce paragraphe analyse les méthodes d'évaluation des performances.

Table B.3: Métriques de performance

Domain		ID	Performance metric	Description
Cluster Performance	Macroscopic performance	1	CH/cluster lifetime	Avg. time duration from a vehicle becoming a CH to giving up its state
		2	CM lifetime	Avg. time duration from a vehicle becoming a CM to giving up its state
		3	No. of clusters/CH	Avg. no. of clusters being formed during the simulation period
		4	Cluster size	Avg. no. of vehicles in a single cluster
		5	Cluster efficiency	The percentage of vehicles participating in clustering process
	Microscopic performance	6	CH change rate	Avg. no. of CH changes per unit time
		7	Cluster change rate	Avg. no. of cluster changes per vehicle in a unit time
		8	State change rate	Avg. no. of state changes per vehicle in a unit time
		9	Disconnection ratio	Avg. percentage of disconnected vehicles
		10	CM reconnection ratio	Avg. Percentage of vehicles that re-cluster within a given time
		11	No. vehicles per hop	Avg. no. of vehicles per hop distance
Network Performance	12	Delivery ratio, success ratio	The percentage of vehicles that successfully receive the packets	
	13	Collision ratio, Packet loss ratio	The percentage of collision packets during packets transmission	
	14	End-to-End delay	Avg. latency of data packets transmitted from source to the destination	
	15	Throughput	The rate of successful message delivery over a communication channel	
	16	Overhead	The ratio of the total no. of control packets to the total no. of data packets	

Selon l'observation des algorithmes de clustering existants, la plupart des algorithmes de clustering visent à augmenter la stabilité du cluster. Cependant, beaucoup d'entre eux n'ont pas expliqué le terme "**stabilité du cluster**" et les paramètres de performance correspondants. Les métriques de performance utilisées sont résumées dans le Tableau B.3, et sont classées en deux catégories: **performance du cluster** et **performance du réseau**. Chaque métrique de performance reçoit une brève description et est identifiée avec les chiffres arabes de 1 à 16.

Les métriques de performance de chaque algorithme de clustering sont résumées dans le Tableau B.1. Les algorithmes de clustering sont présentés en fonction de leur contexte. Les métriques sont représentées par les numéros d'ID définis dans le Tableau B.3. En résumé, les algorithmes de clustering purs peuvent être évalués juste par les performances du cluster car il n'y a pas d'informations liées à l'application. Au contraire, les algorithmes

de clustering liés à l'application doivent être analysés à partir des aspects des performances de clustering et des performances du réseau. En résumé, les métriques d'évaluation des performances doivent dépendre strictement du contexte des algorithmes de clustering, et ces métriques doivent être clarifiées avant les évaluations de simulation.

III MoDyC: Un schéma basé sur la Mobilité pour le clustering dynamique dans les VANET

Dans ce chapitre, nous proposons un algorithme de clustering basé sur la mobilité dynamique (MoDyC) dans le but d'établir un backbone de réseau stable. Le schéma proposé est basé sur les modèles de mobilité des véhicules, y compris la direction du déplacement, la vitesse relative, la distance relative et la durée de vie de la liaison. Un "chef de cluster temporaire" est proposé pour guider la construction du cluster. Par ailleurs, nous introduisons un "seuil de sécurité à distance" pour limiter la taille du cluster. L'algorithme de clustering proposé est évalué en termes de **stabilité du cluster**, et sa performance est comparée aux algorithmes de référence, Lowest-ID [32] et VMaSC [7].

III.a Description de l'algorithme

Nous supposons que les véhicules entrent dans le segment routier avec un débit de circulation prédéfini (le nombre de véhicules qui entrent dans le segment routier par heure). Chaque véhicule se déplaçant sur la route, diffuse un message *Beacon* à chaque intervalle (BI). Selon les métriques de clustering que nous avons mentionnées auparavant, le chef de cluster (CH) devrait être le véhicule qui a une stabilité la plus élevée parmi ses voisins. Par conséquent, nous choisissons le véhicule le plus proche de la position géographique centrale d'un groupe comme le CH, de sorte que le cluster soit plus stable. Les membres de cluster (CM) sont l'ensemble de voisins à 1-saut du CH.

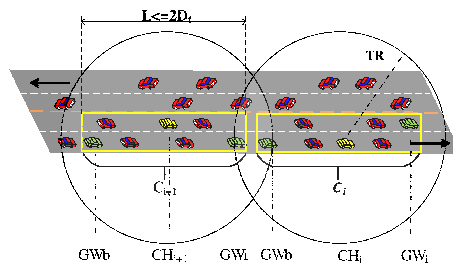


Figure B.4: Clusters (TR : Transmission Range; L : cluster length; D_t : Safe Distance threshold; GW_i, GW_b : Gateway node.)

La Figure B.4 montre deux clusters sur la route droite, cluster C_i et cluster C_{i+1} (les

clusters sont représentés par des rectangles). Le CH est au centre et la longueur du cluster est inférieure à deux fois la distance de transmission de CH (TR). Dans notre méthode de clustering, chaque cluster consiste en deux nœuds de passerelle se déplaçant sur le bord du cluster: l'un avance et l'autre se déplace à la fin du cluster.

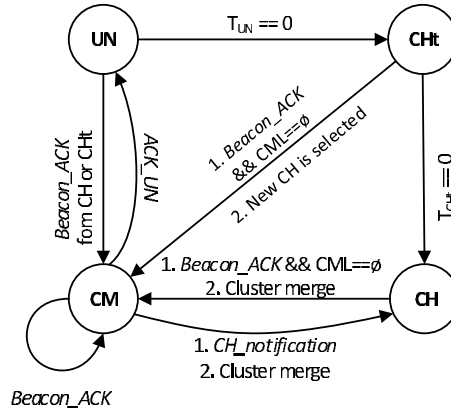


Figure B.5: Machine d'états finis

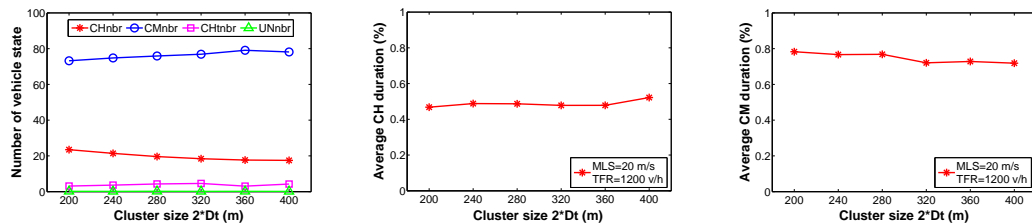
En raison des changements rapides dans la mobilité des véhicules, les véhicules se trouvant en bord de couverture radio de CH sont considérés comme n'étant pas assez stables et peuvent provoquer des déconnexions fréquentes. Pour résoudre ce problème, nous introduisons un seuil appelé "Safe Distance Threshold", noté D_t , qui doit être plus petit que la distance de transmission, $D_t \leq TR$. Par conséquent, les véhicules dans la gamme D_t du CH sont considérés comme ayant des liens plus stables avec leur CH. La taille du cluster est définie par $L \leq 2D_t$.

Dans l'algorithme de clustering proposé, un véhicule peut avoir l'un des quatre états suivants: nœud indéterminé (UN), chef de cluster (CH), membre de cluster (CM) et chef de cluster temporaire (CHt). La Figure B.5 illustre les transitions d'état possibles d'un véhicule.

III.b Évaluation des performances

Cette section présente quelques résultats de simulation pour analyser les performances de MoDyC, y compris les impacts de différents paramètres sur la stabilité du cluster.

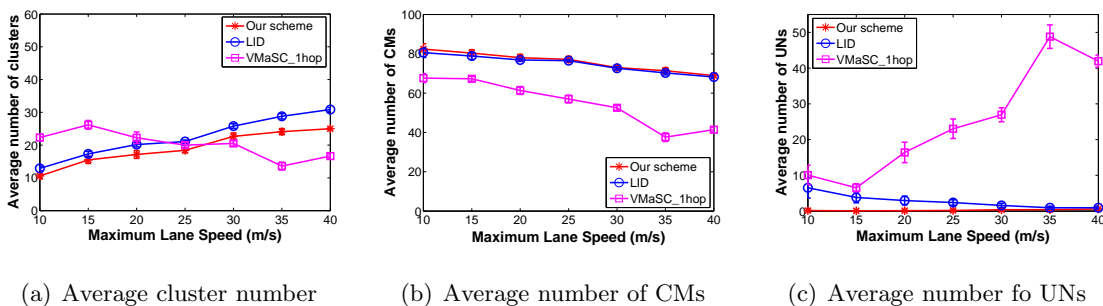
La Figure B.6 présente l'impact du seuil "Safe Distance Threshold" D_t . Dans la Figure B.6(a), avec l'augmentation de D_t , moins de clusters sont organisés pendant la simulation. En effet, plus de véhicules sont ajoutés dans un cluster comme CM lorsque la taille du cluster augmente sous la même densité de trafic. Le nombre de CHt et de UN est stable lorsque la taille des clusters devient plus grande. La Figure B.6(b) montre la durée moyenne de CH, représentée par le pourcentage du temps de simulation total (200s).



(a) Average number of vehicles (b) Average CH duration (%) (c) Average CM duration (%)
in each state

Figure B.6: Impacts de D_t sur les performances du cluster

La durée moyenne de CH augmente légèrement mais reste relativement stable lorsque D_t augmente. La Figure B.6(c) illustre que la durée CM moyenne diminue légèrement avec l'augmentation de la taille du cluster. Nous observons que D_t a un faible impact sur la durée du CH et sur la durée du CM.



(a) Average cluster number (b) Average number of CMs (c) Average number fo UNs

Figure B.7: Comparaison de nombre d'état de véhicule sous l'impact de MLS de véhicule
(Intervalle de confiance : 95%)

Les résultats de la Figure B.7 montrent que le nombre de CH et le nombre de UN dans notre schéma sont légèrement inférieurs aux résultats de LID. De plus, lorsque la vitesse maximale de voie (MLS: Maximum Lane Speed) devient plus grand, beaucoup de CH et de CM dans VMaSC_1hop passent à l'état UN. Par conséquent, le nombre de CH et CMs de VMaSC_1hop dans la Figure B.7(a) et dans la Figure B.7(b) diminue et le nombre d'UN dans la Figure B.7(c) augmente rapidement.

La Figure B.8 montre les détails des transitions d'état au cours du processus de clustering. Les résultats de la Figure B.8(a) et de la Figure B.8(b) révèlent que le temps de changement de CH et les temps de changement d'état du véhicule augmentent rapidement lorsque le MLS augmente. Les taux de changement de CH de VMaSC_1hop et notre schéma sont très bas et restent relativement stables sur la Figure B.8(a). Dans notre schéma, CH peut devenir un CM lorsque la fusion de cluster arrive ou quand il perd tous

ses membres. Dans la Figure B.8(b), le nombre de transitions d'état pour chaque véhicule dans VMaSC_1hop et notre schéma sont plus élevés que celui de LID. C'est parce que plus d'états de véhicule sont définis dans ces deux systèmes par rapport à LID.

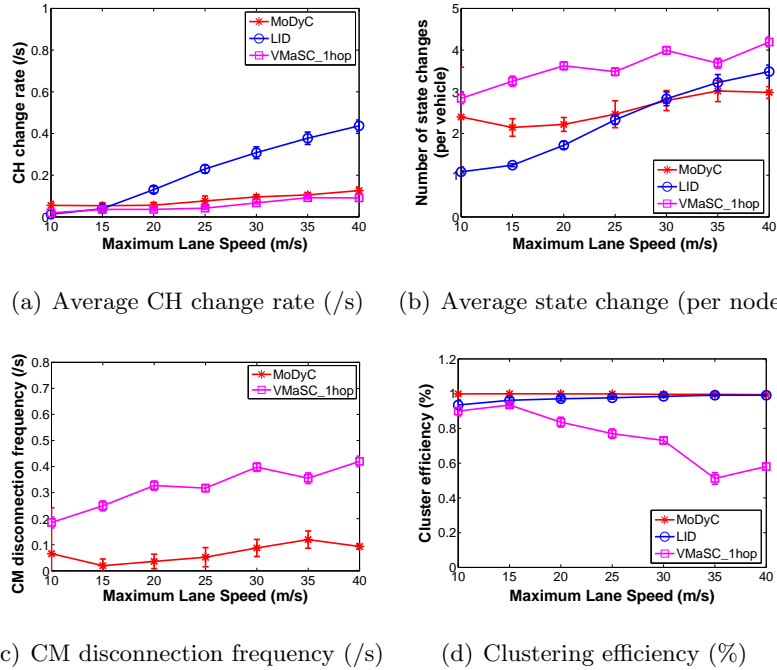


Figure B.8: Comparaison de la stabilité du cluster sous l'impact de MLS
(Intervalle de confiance : 95%)

La fréquence de déconnexion de CM, représentée sur la Figure B.8(c), présente une tendance de croissance similaire à celle des résultats de la Figure B.8(b). En effet, la transition d'état du véhicule se produit toujours lorsqu'un CM perd la connexion de liaison avec son CH actuel. La Figure B.8(c) compare notre schéma et VMaSC_1hop. Il est évident que notre schéma montre une fréquence de déconnexion CM très faible par rapport à VMaSC_1hop, indiquant que notre schéma fournit une plus grande stabilité de cluster.

A partir des résultats de la Figure B.8(d), nous observons que LID et notre schéma ont une efficacité de clustering (cluster efficiency) très élevée, qui est proche de 100 % lorsque le MLS augmente. Cela signifie que presque 100 % des véhicules participent à la procédure de clustering pendant la simulation. Cependant, avec la croissance de MLS, l'efficacité de clustering de VMaSC_1hop diminue rapidement, cela est dû au fait que le nombre de UNs augmente rapidement lorsque MLS devient haut, comme le montre la Figure B.7(c).

IV L'approche Unified Framework of Clustering (UFC)

Dans ce chapitre, nous proposons une approche de clustering (UFC), composée de trois parties: i) Neighbor Sampling (NS); ii) Sélection de CH basée sur le backoff (BCS); iii) Cluster Maintenance basée sur BackUp CH (BUCH). Sur la base du framework UFC, nous implémentons trois métriques de clustering basées sur la mobilité: la position relative du véhicule, la vitesse relative et la durée de vie de la liaison. De plus, nous fournissons une analyse détaillée de l'approche UFC avec l'optimisation des paramètres.

IV.a Description de UFC

Dans l'approche UFC, chaque véhicule est dans l'un des 4 états: nœud indécis (UN), chef de cluster (CH), membre de cluster (CM) et membre candidat de cluster (CCM). La Figure B.9 présente les transitions d'état possibles d'un véhicule.

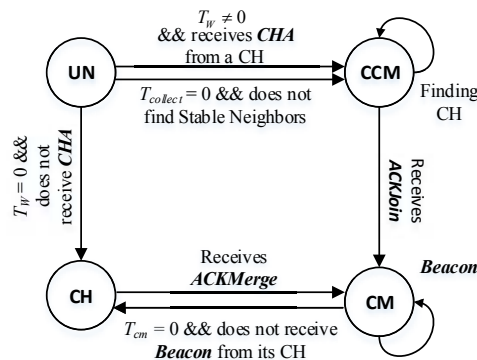


Figure B.9: Machine d'états finis

IV.a1 Neighbor Sampling (NS)

L'état *Neighbor Sampling (NS)* permet aux véhicules de filtrer les voisins instables afin de réduire les déconnexions du CM et d'améliorer la stabilité du cluster. Chaque véhicule a un ensemble de *stables neighbors (SN)*, et les voisins stables (*stable neighbors*) doivent présenter un modèle de mobilité similaire: (1) se déplacer dans la même direction; et (2) avec une différence de vitesse Δv inférieure à un seuil prédéterminé Δv_{th} .

IV.a2 Backoff-based CH Selection (BCS)

L'état *Backoff-based CH Selection (BCS)* permet à chaque véhicule de configurer son propre temporisateur de backoff, T_W , de manière distribuée, en attendant de diffuser un message d'annonce CH *CHA*. Les premiers véhicules diffusant des messages *CHA* parmi leurs voisins deviendront des CH initiaux.

Ce travail propose deux méthodes pour la sélection de CH. La première méthode est une méthode basée sur la métrique, incluant les métriques suivantes : durée de vie moyenne du lien \overline{LLT}_i , distance relative moyenne $\overline{\Delta D}_i$ et vitesse relative moyenne $\overline{\Delta v}_i$. La seconde est une méthode aléatoire.

IV.a3 Backup CH based vehicle re-clustering (BUCH)

Dans ce travail, un méthode en utilisant le *Backup CH* est proposé, afin de réduire le délai de re-clustering du véhicule. Une liste de Backup CH (*BCHL*) est créée et mise à jour dans chaque nœud, CCM et CM. Chaque fois que le CM perd la connections avec son CH actuel, il commence à trouver le *Backup CH* (*BCH*) le plus qualifié (avec la durée de lien la plus élevée) à suivre.

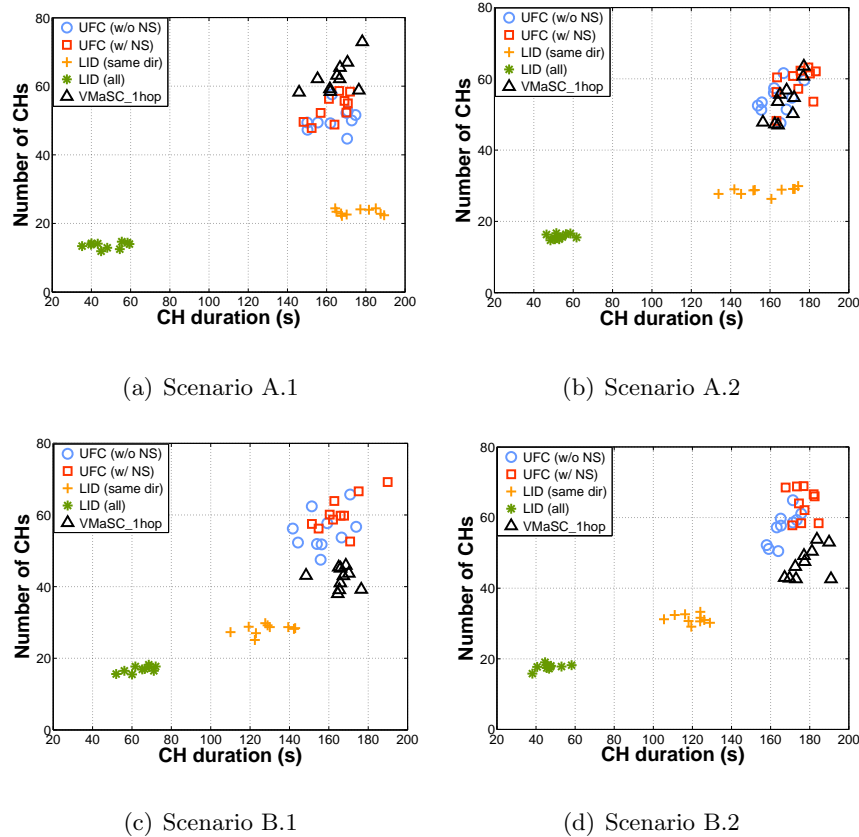


Figure B.10: Comparaison de la durée de vie de CH entre UCF, LID, et VMaSC

IV.b Évaluation des performances

Dans la Figure B.10, nous observons que LID (all) présente le nombre minimum de clusters et la plus petite durée de vie de CH dans tous les scénarios. Dans, LID (all), il y a plus de chances pour un CM devenir un CH et pour un CH de devenir un CM, puisque les

véhicules se déplaçant dans des directions opposées peuvent également rester dans le même groupe lorsqu'ils se rencontrent sur la route. La modification fréquente du CH réduit la durée de vie du CH et la taille illimitée du cluster réduit le nombre de clusters créés. Nous remarquons également que dans le scénario A.1 qui est à faible dynamique, LID (same dir) donne les meilleurs résultats en termes de durée de CH et de nombre de clusters. Cependant, lorsque le trafic devient plus dynamique, notamment sur la Figure B.10(c) et la Figure B.10(d), la durée moyenne de CH de LID (same dir) diminue rapidement et devient plus petit que celui de deux sous-schémas UFC et du schéma VMaSC_1hop.

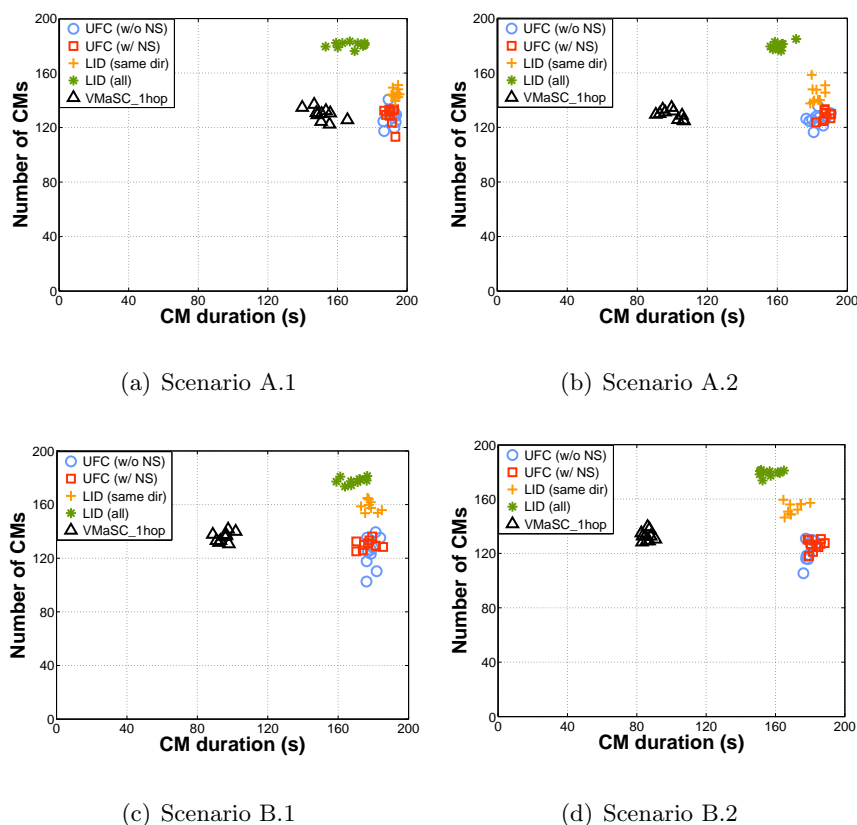
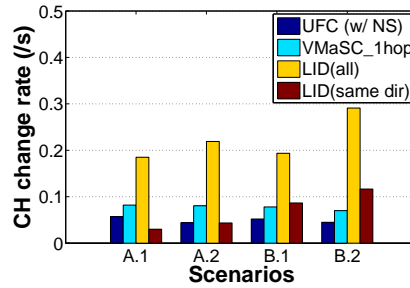


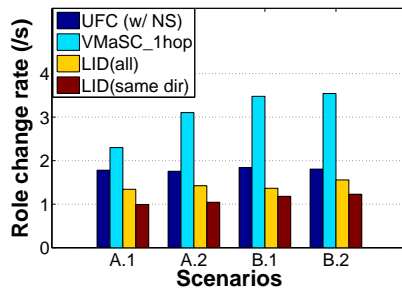
Figure B.11: Comparaison de la durée de vie de CM entre UFC, LID, et VMaSC

La Figure B.11 illustre la durée de vie de CM et le nombre de CM. Apparemment, LID (all) présente le nombre maximum de CM dans tous les scénarios, parce que que les véhicules dans des directions opposées sont autorisés à rester dans le même cluster et que la capacité du cluster n'est pas limitée dans le LID. Étant donné que les connexions de liaison entre les véhicules se déplaçant dans les directions opposées ne sont pas stables, la durée de vie de CM dans LID (all) est inférieure à celle de LID (same dir). LID (same dir) présente des performances similaires avec deux sous-schémas UFC dans les scénarios A.1, A.2 et B.1, en termes de durée de vie CM; et présente des performances plus élevées que les sous-systèmes UFC en termes de nombre de CM dans tous les scénarios. Cela est

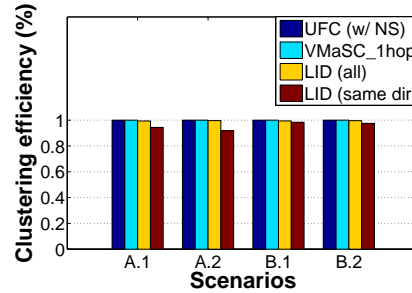
dû au fait que la capacité de cluster n'est pas limitée dans LID.



(a) CH change rate (per second)



(b) Role change rate (per second)



(c) Clustering efficiency (%)

Figure B.12: Comparaison des performances de clustering entre l'UFC et LID

Les résultats de la simulation (Figure B.12) présentent la comparaison entre UFC (w/ NS), LID (same dir), LID (all), et VMaSC_1hop, en termes de taux de changement CH, taux de changement de rôle, et l'efficacité du clustering. Dans la figure B.12(a), nous observons que CH change le plus fréquemment dans LID (all) dans tous les scénarios de trafic, en particulier sous B.2. Même si le taux de changement CH de LID (même direction) est le plus bas dans le scénario A.1, la valeur augmente significativement et devient plus grande que celle de UFC (w/ NS) et VMaSC_1hop lorsque le scénario de trafic devient plus dynamique. Au contraire, le taux de changement CH de UFC (w/ NS) et VMaSC_1hop sont tous considérés comme insensibles aux changements de scénarios de trafic, et UFC (w/ NS) fonctionne toujours mieux que VMaSC_1hop.

Dans la Figure B.12(b), nous observons que les véhicules changent d'état plus fréquemment lors de l'implémentation de UFC (w/ NS) et VMaSC_1hop, comparés à LID (all) et LID (same dir). Le taux de changement de rôle du UFC (w/ NS) reste stable, alors que le scénario de trafic change. En effet, la méthode NS proposée garantit des connexions intervéhicules plus stables et le méthode BUCH permet aux CM de trouver des BCH au lieu de changer immédiatement d'état.

La Figure B.12(c) illustre les résultats de la comparaison en termes d'efficacité de

clustering. L'efficacité du clustering est toujours de 100% avec les approches UFC (w/ NS) et VMaSC_1hop, dans tous les scénarios de trafic. Cela indique que tous les véhicules sur la route ont participé au processus de clustering. LID (same dir) et LID (all) manquent toujours certains véhicules, ce qui peut conduire à l'inexactitude du clustering.

V L'impact de la fusion de clusters sur la stabilité de cluster dans les VANET

Dans cette section, nous étudions les méthodes de fusion de clusters existants et proposons une nouvelle méthode de fusion de clusters basée sur le leadership (LCM). Ensuite, une comparaison complète des différentes méthodes de fusion de cluster est donnée. LCM montre de meilleures performances sur la stabilité des clusters.

V.a Description de LCM

En se basant sur le framework de clustering (UFC) proposé dans la Section B, nous nous concentrons uniquement sur la description de la méthode de fusion de cluster. Le processus de fusion des clusters comporte deux parties: la vérification des conditions de la fusion et la sélection du nouveau CH.

Lorsque deux CH voisins se déplacent dans la même direction en couverture l'un de l'autre, un processus de détection de la fusion sera activé. Tant que le CH se déplaçant derrière CH_b reçoit un message *Beacon* du CH se déplaçant devant CH_f , il va démarrer un temporisateur, appelé intervalle de fusion (MI). CH_b ne vérifie les conditions de fusion que s'il peut recevoir les *Beacon* consécutifs à partir de CH_f pendant la période MI. Sinon, ces deux clusters ne peuvent pas être fusionnés. Lorsque MI expire, CH_b vérifie les conditions de fusion, listées comme suit: 1) Deux clusters se déplacent dans la même direction; 2) Le nombre de CM dans le cluster fusionné est inférieur à la taille de cluster prédéterminée; 3) La différence entre la vitesse relative moyenne de deux clusters doit être inférieure à une valeur prédéfinie.

Une fois la fusion effectuée, deux CH choisissent l'un d'entre eux comme nouveau CH dans le cluster fusionné. Ce CH devrait avoir un leadership plus élevé, ce qui signifie qu'il devrait être plus stable au sein de tous les membres dans le cluster fusionné.

V.b Évaluation des performances

Les performances macroscopiques, comme la durée moyenne de vie de CH et la durée moyenne de vie de CM, par rapport à MI, sont représentées sur la Figure B.13. Lorsque MI augmente, la mobilité de ces deux clusters qui peuvent être fusionnés devient de plus en plus similaire. Par conséquent, le schéma CM-based montre de meilleures performances que celui basé sur VMaSC lorsque MI est assez grand. Par rapport à ces deux schémas, le LCM obtient globalement les meilleures performances, car il a une meilleure prédiction de la stabilité des communications entre les CM et les CH potentiels.

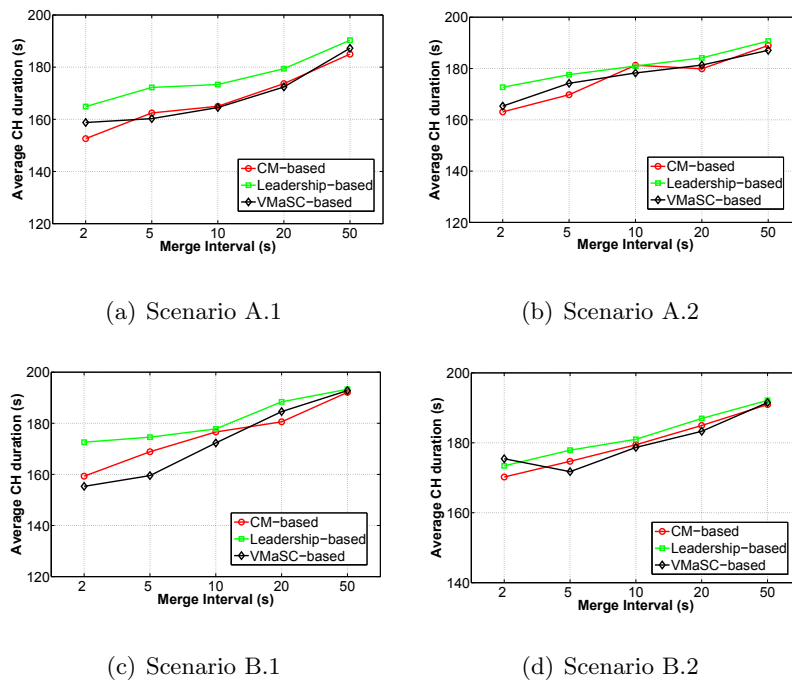


Figure B.13: La durée de vie de CH sous l'impact de MI

Les performances microscopiques, comme le taux de changement de rôle et le taux de déconnexion de CM, qui sont présentées dans la Figure B.14. Les performances de LCM sont les meilleures avec le taux de déconnexion de CM le plus bas, parce que la méthode LCM vise à assurer un cluster stable après la fusion. Les performances de CM-based sont pires en fonction du taux de déconnexion de CM (le plus haut), parce qu'il n'a pas pris en considération la stabilité de connections.

Nous pouvons observer que les performances macroscopiques et microscopiques sont améliorées avec l'augmentation de MI. Cependant, il n'est pas souhaitable d'utiliser un grand MI, car cela peut entraîner de nombreux clusters de petites tailles, ce qui diminue l'efficacité de clustering dans les VANET pour la dissémination de données. Les résultats de simulation montrent que le LCM est moins sensible au changement de MI. Par conséquent, LCM est capable de garantir de bonnes performances macroscopiques et micro-

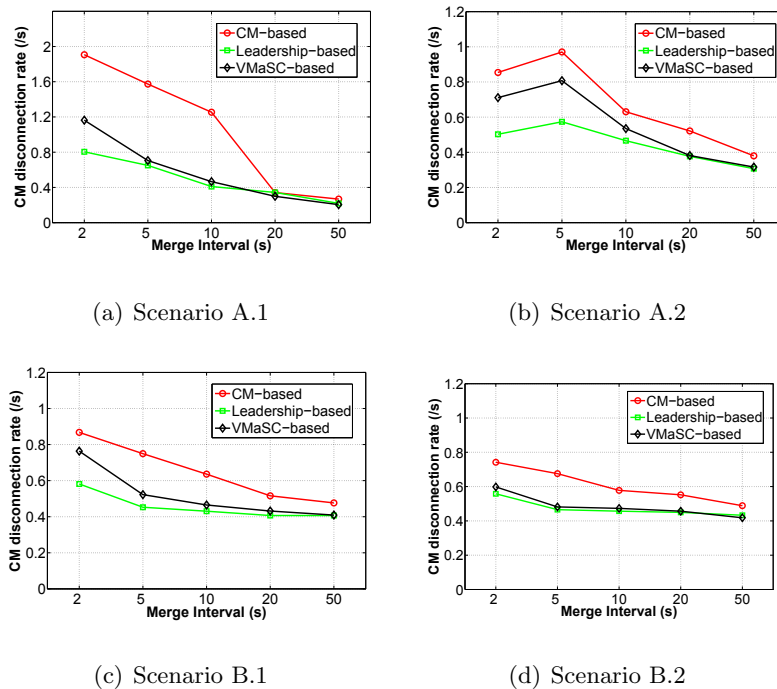


Figure B.14: Le taux de déconnexions de CM sous l'impact de MI

scopiques et une bonne efficacité de clustering.

VI L'analyse des performances du clustering des véhicules

Dans ce paragraphe, nous présentons un modèle stochastique pour l'analyse des performances de clustering, basé sur les résultats de la simulation de l'UFC proposé dans la Section B. Les transitions d'état des véhicules au cours du processus de clustering sont modélisées sous la forme d'une chaîne de Markov à états finis en temps discret. Une analyse approfondie de l'algorithme de clustering peut être réalisée selon ce modèle, y compris la durée de cluster celle de CM. Les résultats montrent une grande cohérence entre les résultats analytiques et ceux de simulation. En outre, les performances de clustering peuvent être prédites par le modèle proposé lorsque le temps de simulation continue d'augmenter.

VI.a Description du modèle

Nous modélisons le processus de transition d'état du véhicule sous la forme d'une chaîne de Markov à états finis en temps discret X_k . La chaîne de Markov X_k a 4 états $S = \{s_0, s_1, s_2, s_3\}$, qui sont notés respectivement UN, CCM, CM et CH. Pour simplifier ce modèle, un processus de clustering peut être traité comme deux parties: la période de

clustering instable avec une grande variance de probabilités de transition d'état, notée L_1 ; et la période de clustering stable avec une petite variance des probabilités de transition d'état, notée L_2 . Par conséquent, le temps prévu pour qu'un véhicule passe dans chaque état de Markov est composé de deux parties, données par

$$E_{s_i} = E_{s_i}[L_1] + E_{s_i}[L_2], \quad (\text{B.1})$$

Pour calculer le temps prévu pour que le véhicule passe dans un état spécifique, nous pouvons simplifier la chaîne de Markov de 4 états en une chaîne de Markov de naissance et de mort de 2 états, y compris l'état actuel et non actuel, qui sont représentés par l'état A et l'état B dans la Figure B.15.

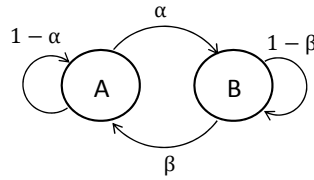


Figure B.15: La chaîne de Markov de 2-état

Basé sur Eq. B.1, le temps que passe un véhicule dans l'état A est

$$E_A = E_A[L_1] + E_A[L_2]. \quad (\text{B.2})$$

Au début du processus de clustering $t = t_0$, tous les véhicules sont dans l'état UN s_0 et la distribution initiale est $\mathbf{Q}(t_0) = \{1, 0, 0, 0\}$. Soit $Q_{s_i}(t)$ la probabilité qu'un véhicule reste dans s_i à l'instant t , la distribution d'état à l'instant t est décrite par

$$\mathbf{Q}(t) = \{Q_{s_0}(t), Q_{s_1}(t), Q_{s_2}(t), Q_{s_3}(t)\}. \quad (\text{B.3})$$

VI.b Évaluation des performances

La Figure B.16 illustre la durée de l'état sous différents scénarios. La simulation se termine à $t = 3000s$. La durée de vie de CH augmente dans la deuxième slot puisque les CH ont été sélectionnés pendant cette période. A partir du troisième slot où $t = 320s$, la durée de vie de CH est approchée d'une augmentation linéaire avec l'augmentation du temps de simulation. Cependant, lorsque le temps de simulation continue d'augmenter, la durée de vie de l'état augmente lentement, en particulier dans les scénarios dynamiques (B.1 et B.2), comme illustré dans les figures B.16(c) et B.16(d).

Une tendance similaire peut être trouvée pour la durée de vie de CM et CCM. La durée de vie de CM reste 0 jusqu'au troisième slot, car les véhicules CCM doivent demander aux

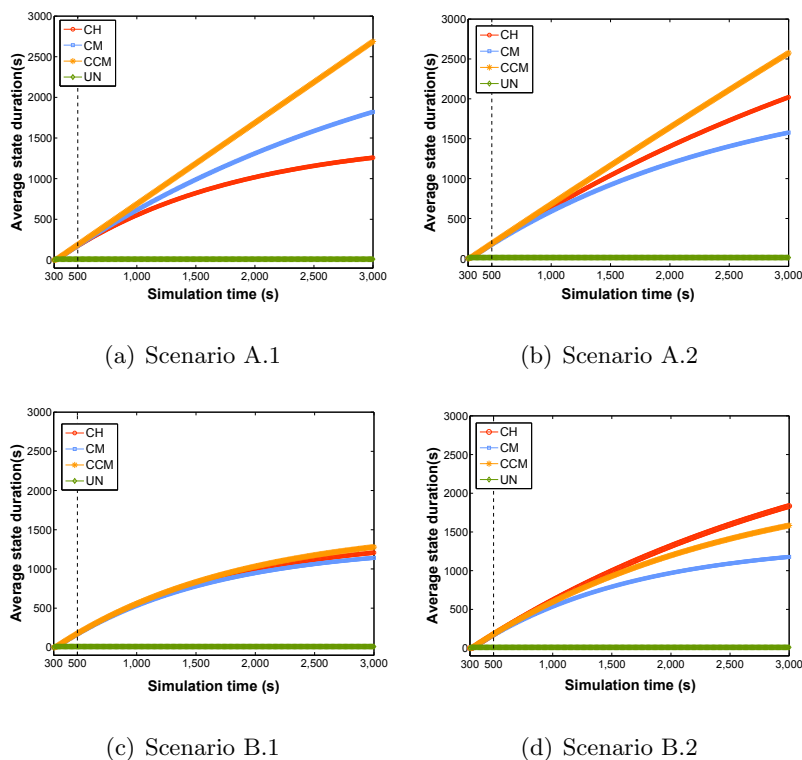


Figure B.16: La durée d'état prévue E_{s_i} ($t = 3000s$)

CH de rejoindre le cluster pendant le deuxième slot. De plus, puisque les UN n'apparaissent que dans le premier slot, la durée de UN reste une valeur constante lorsque le temps augmente.

La Figure B.17 montre les résultats de simulation de distribution d'états quand le temps augmente. On peut observer que lorsque le temps augmente, le nombre de CH et de CM devient stable. De plus, lorsque le scénario de trafic est plus dynamique, tel que le scénario B.2 montré dans la Figure B.17(d), plus de CM changent aux CH. Cependant, lorsque le scénario de trafic est plus statique, comme A.1 illustré dans la Figure B.17(a), les clusters sont plus stables.

VIII Conclusions

Dans cette thèse, nous avons abordé les caractéristiques et les défis de VANET. Nous avons étudié la nécessité d'algorithmes de clustering pour améliorer les connexions intervéhicules. Dans un premier temps, nous avons proposé un schéma basé sur la mobilité pour le clustering dynamique (MoDyC) afin d'améliorer la stabilité du cluster. Les performances de l'MoDyC sont supérieures à ceux du LID en terme de la stabilité de cluster, notamment dans les scénarios très dynamiques. Cependant, la construction des clusters prennent beaucoup de temps. Deuxièmement, pour analyser les impacts des différentes métriques de

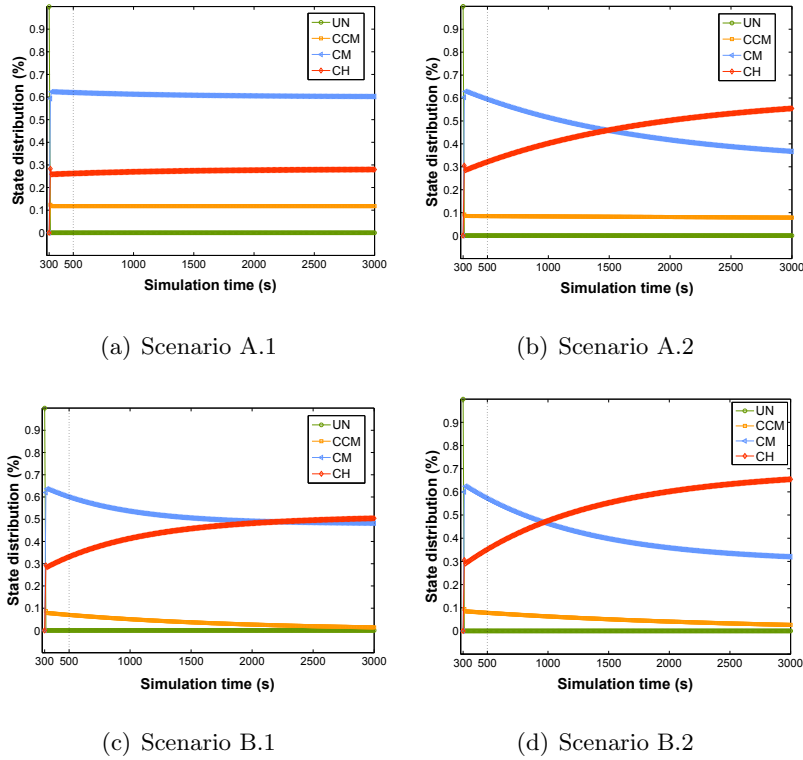


Figure B.17: La distribution des états $\mathbf{Q}(t)$ ($t = 3000s$)

clustering, nous avons conçu un cadre de clustering (UFC). En outre, sur la base de l'UFC, nous avons introduit une méthode de fusion des clusters basée sur le leadership (LCM) et étudié l'impact des méthodes de fusion de clusters sur les performances de clustering. Les performances de l'UFC sont supérieures à ceux du LID et VMaSC notamment en terme de taux de déconnexions de CM. Par ailleurs, nous avons observé que la méthode de fusion de clusters est aussi un élément important qui peut influencer les performances de clustering. Ensuite, un modèle stochastique a été proposé pour l'analyse de performances de clustering. Les résultats de l'analyse et de la simulation sont correspondants. Avec ce modèle, nous pouvons également prévoir la distribution d'états des véhicules. Finalement, nous avons proposé une méthode de dissémination de données basée sur clustering afin de délivrer le message d'urgence. Le ratio de la transmission des données (PDR) et le temps de propagation de bout en bout sont supérieurs à ceux du VMaSC.

Mengying REN

Doctorat : Ingénierie Sociotechnique des Connaissances, des Réseaux et du Développement Durable

Année 2018

Diffusion des données basée sur les clusters dans les réseaux ad-hoc véhiculaires (VANETs)

En raison de l'essor rapide des technologies de communication sans fil et de la demande croissante de services dans les réseaux ad hoc véhiculaires (VANETs), le besoin d'algorithmes de clustering efficaces est nécessaire pour résoudre le problème de mise à l'échelle et le besoin croissant d'applications dans les VANETs. Cependant, à cause de la nature dynamique des VANETs, les algorithmes de clustering doivent assurer la stabilité du cluster. Pour cela, l'objectif de la thèse est de proposer un cadre générique servant à définir des algorithmes de clustering afin d'améliorer la stabilité du cluster et d'augmenter l'efficacité de la transmission de données. Dans cette thèse, nous introduisons un algorithme de clustering (MoDyC) qui se base sur la métrique de mobilité pour former des clusters. Ensuite, nous proposons un cadre générique servant à définir des algorithmes de clustering appelé UFC. Nous évaluons l'impact des métriques de clustering et des méthodes de fusion de clusters sur le critère de stabilité. Nous proposons une nouvelle méthode de fusion de cluster appelée LCM. Cette méthode basée sur la métrique Leadership permet d'augmenter la stabilité du cluster. Nous analysons les performances des algorithmes de clustering proposés. Dans le but de modéliser le processus de clustering, nous proposons un modèle basé sur les chaînes de Markov. A partir de ce modèle, nous analysons la stabilité du cluster. Cette thèse se termine par l'étude d'un scénario de diffusion de messages d'urgence en utilisant un schéma de diffusion de données en cluster avec cache de contenu.

Mots clés : réseaux locaux sans fil- réseaux ad hoc de véhicules - systèmes de transport intelligent - données, transmission.

Cluster-based Data Dissemination in Vehicular Ad-hoc Networks (VANETs)

However, because of the dynamic network topology of VANETs, clustering algorithms are required to guarantee the cluster's stability. Therefore, this thesis aims to propose a framework for clustering algorithms for VANETs, to improve cluster's stability and to increase the efficiency of data transmission. In this thesis, we firstly investigate a new mobility-based scheme for dynamic clustering (MoDyC) in VANETs using the mobility-based clustering metric to construct clusters. Then, we propose our framework for clustering algorithm named UFC. Based on this framework, we evaluate the impacts of different clustering metrics and cluster merging schemes on cluster's stability. Meanwhile, a leadership-based cluster merging scheme (LCM) is presented to increase the cluster stability. To analyze the cluster stability, we propose a Markov chain model to model the clustering process and to investigate its performance. Finally, this thesis studies a scenario, aiming to disseminate vehicular emergency messages through cluster-based data dissemination scheme.

Keywords: wireless LANs - vehicular ad hoc networks (computer networks) - intelligent transportation systems - data transmission systems.

Thèse réalisée en partenariat entre :

