

1-1-2011

Efficient channel allocation and medium access organization algorithms for vehicular networking

Zaydoun Yahya Rawashdeh
Wayne State University,

Follow this and additional works at: http://digitalcommons.wayne.edu/oa_dissertations

 Part of the [Computer Engineering Commons](#), [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Rawashdeh, Zaydoun Yahya, "Efficient channel allocation and medium access organization algorithms for vehicular networking" (2011). *Wayne State University Dissertations*. Paper 359.

**EFFICIENT CHANNEL ALLOCATION AND MEDIUM ACCESS
ORGANIZATION ALGORITHMS FOR VEHICULAR
NETWORKING**

by

ZAYDOUN RAWASHDEH

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2011

MAJOR: COMPUTER ENGINEERING

Approved by:

Advisor

Date

© COPYRIGHT BY

ZAYDOUN RAWASHDEH

2011

All Rights Reserved

DEDICATION

I dedicate this work to my parents, Rawashdeh Yahya and Amneh, for their continued support and encouragements. I also dedicate this work to my brothers and sisters for their love and support, they have always encouraged me toward excellence. Special thanks to all my friends.

TABLE OF CONTENTS

Dedication	ii
List of Figures	vii
List of Tables	x
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: BACKGROUND AND RELATED WORKS.....	10
2.1 Admission Control and Scheduling in VANET.....	10
2.1.1 Maximum Freedom Last Scheduling Algorithm for Downlinks of DSRC Networks.....	12
2.2 Clustering in VANET.....	16
2.2.1 Cluster-based Media Access Control in VANET.....	17
2.2.1.1 Media Access Concept for VANETs Based on Clustering.....	17
2.2.1.2 Cluster-Based DSRC Architecture for QOS Provisioning over Vehicle Ad Hoc Networks.....	20
2.2.2 Cluster Formation Algorithms.....	23
2.2.2.1 The Cluster-Based Location Routing.....	26
2.2.2.2 Clustering Formation for Inter-vehicle Communication.....	28

CHAPTER 3: The proposed methods for channel access and clustering in VANET	30
3.1 Channel Allocation for RSU Based on Virtual Task Finish Time	30
3.1.1 Motive and System Description.....	30
3.1.2 System Model	32
3.1.3 Problem Analysis	33
3.1.4 Time Shares Compensation	38
3.1.4 Algorithm Description	41
3.1.5 Allocation and Distribution of Time Shares	44
3.2 Media Access Technique for Cluster-based Vehicular Ad hoc Networks.	45
3.2.1 System Description	45
3.2.2 Cluster Formation and Cluster Members Functionality.....	47
3.2.3 The Cluster System Cycle.....	47
3.2.4 Delay Analysis	49
3.2.5 The Impact of IEEE 802.11 Distributed Coordinated Function (DCF) on the Delay of the Safety Message.....	52
3.3 A Novel Algorithm to Form Stable Clusters in Vehicular Ad hoc Networks.	53
3.3.1 System Overview and Assumptions	53
3.3.2 Clustering Process and Protocol Structure.....	55
3.3.3 Neighborhood Relationship	56

3.3.4 Cluster-Head Election Parameters	57
3.3.5 Cluster Formation Algorithm.....	60
3.3.6 Analysis of Cluster-Head Election.....	64
3.3.7 Cluster Maintenance	67
CHAPTER 4: SIMULATION AND PERFORMANCE EVALUATION.....	70
4.1 Performance Analysis of the Channel Allocation for RSU Based on Virtual Task Finish Time	70
4.1.1 Simulation Environment	70
4.1.2 Metrics for evaluation	72
4.2 Performance Analysis of the Media Access Technique for Cluster-based Vehicular Ad hoc Networks	75
4.2.1 Simulation Description	75
4.2.2 Metrics and Results.....	76
4.3 Simulation and Performance Evaluation of Cluster Formation Algorithm	79
4.3.1 Simulation Setup	79
4.3.2 Evaluation Criteria	81
4.3.2.1 Cluster Stability	81
4.3.2.2 Average Cluster Lifetime.....	85
4.3.2.3 Number of Clusters	87

4.3.2.4 Overhead for Clustering.....	90
CHAPTER 5: CONCLUSION	93
CHAPTER 6: FUTURE WORK	95
Bibliography	96
Abstract	105
Autobiographical Statement.....	107

LIST OF FIGURES

Figure 1.1 a) The DSRC spectrum b) channels 172 through 184 divided into 6 Service Channels (SCH) and one Control Channel (CCH)	2
Figure 2.1 RSU deployments along the roadside.....	13
Figure 2.2 VANET cluster	16
Figure 3.1 Vehicles in the RSU communication range.....	31
Figure 3.2 Illustrative example	37
Figure 3.3 time plan allocation	43
Figure 3.4 task share reduction	44
Figure 3.5 vehicles' positions with respect to the RSU	45
Figure 3.6 inter/intra-cluster communication links.....	46
Figure 3.7 cluster system cycle	48
Figure 3.8 message generation and transmission during a single cycle	51
Figure 3.9 neighborhood relationship of a given vehicle	57
Figure 3.10 suitability value, $\alpha=1$	59

Figure 3.11 clustering initiation process	60
Figure 3.12 cluster-head determination process	61
Figure 3.13 finalizing clustering process	62
Figure 3.14 vehicles' location with respect to the COV node	65
Figure 3.15 probability density function (PDF).....	67
Figure 4.1 tasks per minute	73
Figure 4.2 failure rate.....	74
Figure 4.3 effective cycle usage.....	75
Figure 4.4 earliest notifications.....	77
Figure 4.5 average safety messages delay caused by competition-based technique	78
Figure 4.6 (a) average cluster changes per vehicle	83
Figure 4.6 (b) average cluster changes per vehicle.....	83
Figure 4.6 (c) average cluster changes per vehicle	84
Figure 4.7 (a) average cluster lifetime	86

Figure 4.7 (b) average cluster lifetime	86
Figure 4.7 (c) average cluster lifetime	87
Figure 4.8 (a) average total number of formed clusters for TB, PB, and WB	88
Figure 4.8 (b) average total number of formed clusters for TB, PB, and WB	89
Figure 4.8 (c) average total number of formed clusters for TB, PB, and WB	89

LIST OF TABLES

Table I: Application range and tolerance time window for different VSC applications ..	52
Table II: Simulation parameters.....	76
Table III: The average and the standard deviation of the speed	80

CHAPTER 1: INTRODUCTION

Recent advances in wireless networks have led to the introduction of a new type of networks called Vehicular Ad Hoc Networks (VANET). This type of networks has recently drawn significant research attention since it provides the infrastructure for developing new systems to enhance drivers' safety [1-3]. Equipping vehicles with various kinds of sensing devices and wireless communication capabilities helps drivers to acquire real-time information about road conditions allowing them to react on time. For example, warning messages sent by vehicles involved in an accident enhances traffic safety by helping the approaching drivers to take proper decisions before entering the crash dangerous zone [4-5]. Moreover, information about the current transportation conditions facilitate driving by taking new routes in case of congestion, thus saving time and adjusting fuel consumption [6-7]. In addition to safety concerns, VANET can also support other non-safety applications that require Quality of Service (QoS) guarantee. This includes Multimedia (e.g., audio/video) and data (e.g., toll collection, internet access, weather/maps/ information) applications.

The communications in Vehicular Ad hoc Network (VANET) can be categorized into Vehicle to Roadside (V2R) units and Vehicle to Vehicle (V2V) communications [8]. The Inter-Vehicle Communication (IVC) operates in the 5.9 GHz band as shown in Figure 1.1 (a) and (b). The spectrum is divided into 7 channels, one of these channels is called the control channel, and the remaining six are called service channels.

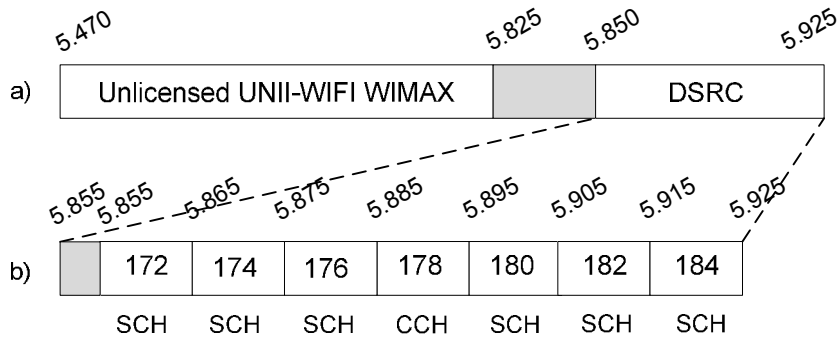


Figure 1.1 a) The DSRC spectrum b) channels 172 through 184 divided into 6 Service Channels (SCH) and one Control Channel (CCH)

The literature has numerous studies focusing on V2R and V2V communications. In this dissertation, we will study both V2R and V2V communications. For V2R communications, we focus on the resource managements; and for V2V communications, we focus on the network topology stability and media access organization using clustering techniques.

The RSUs will be deployed along the main roads to provide passing by vehicles with different safety and non-safety services. For the V2R communications, we mainly focus on the non-safety applications, where a bulk of data needs to be transferred within a short period of time (e.g., travel information, digital maps downloading, commercial advertisements, vehicles' software upgrades). Vehicles can stay in contact with the RSU for a short period of time. Therefore, the services provided by the RSU should be completed before the vehicles leave the RSU transmission range. In addition to that the RSU should also be capable of dealing with increasing number of vehicles requesting different services with different data sizes.

In overloaded conditions (during rush hours) tens of vehicles might request different services from a single RSU. These kinds of scenarios pose a very challenging problem

for the resource management algorithms used by the RSUs. One of the main challenges is channel allocation. Efficient channel allocation methods should manage bandwidth allocation in an optimal way to maximize the amount of data being exchanged. Another challenge is the Admission Control (AC) problem. The task of the AC is to decide whether to accept or reject the new request depending on whether the requirements of the new request will be fulfilled while maintaining the quality of service for all in-progress services. In the overloaded conditions, inefficient admission control methods might lead to congestions and delay of packets delivery, which results in an increase in the number of failed tasks.

Channel allocation and admission control methods proposed for single-hop wireless transmission targeted long-term flows like multimedia applications [9] [10]. However, these methods cannot perform well in high dynamic networks like VANETs. Therefore, channel allocation and admission control schemes for RSUs should take into consideration the short connection duration between the vehicles and the RSUs, the deadline constraints, and the resource sharing among multiple concurrent vehicles.

With the increase number of arrivals, the load on the RSU increases. When the RSU becomes close to overloaded situations, accepting more nodes means pushing the task finish time closer to the deadline (vehicle leaving time) as in [11] [12]. However, pushing the task finish time to the edges results in a very tight air-time transmission plan, which increases the risk of the task, thus, increasing the probability of task failure and increasing the number of failed tasks. This could easily happen due to fluctuations in channel conditions, especially when the vehicle moves away from the RSU where the signal strength becomes very weak.

In this dissertation, we propose a channel allocation algorithm used to generate an air-time transmission plan for each task until the task is completed. The algorithm focuses on reducing the risk of the vehicles that have been getting service for long period of time and are about to leave the RSU range, and keep the risk for the rest of the vehicles (the recently admitted ones need more time to leave the RSU transmission range) at the minimum level (the estimated task finish time should not exceed the vehicle departure time). The algorithm tries to allocate the channel such that the minimum requirements for the tasks are guaranteed and at the same time the system throughput is increased.

Our proposal is motivated by the fact that vehicles' movement is predictable and is restricted by the structure of the roads, the speed limits, and the traffic flow constraints. For example, the number of arrivals to/departures from the RSU region is predictable and can't exceed a certain number of vehicles per unit of time due to physical limitations like vehicles length, width, and the safe distance between two consecutive vehicles on the same lane. For example, two consecutive vehicles on the same lane can't depart from the RSU range at the same time. Consecutive vehicles on the same lane tend to have Safe Distance (SD) greater than 1.5 seconds; therefore, if the leading vehicle, v_i , leaves the RSU range at time t , then v_j , the following vehicle, will leave at least at time $t + SD$. Unlike MANET where the movement of the nodes is random and it's difficult to predict the number of arrivals and departures, the number of departures from the RSU region at the same time can't exceed n nodes on a n -lane road). Simulation results show that our algorithm maintains a higher number of admitted tasks and at the same time reduces the failure rate of those tasks. Our algorithm efficiently utilizes resources, such as bandwidth and time, compared to other admission techniques.

Our proposed algorithm calculates the expected physical task finish time for the arrivals and allocates a transmission plan for each admitted task. The vehicles that are closer than others (in terms of distance) to leave the RSU range will be treated differently. Those vehicles will be allocated a virtual transmission plan that is basically the expected task finish time plus an extra time called the Backup time, Δ_{bt} , that can be used in case the vehicle couldn't finish its task on time. This extra time (or part of it) will automatically be assigned to the next vehicles to leave the RSU range in case the first vehicle completed its task before the deadline. Our algorithm always re-evaluates the transmission plan of all admitted tasks and allocates virtual transmission plans accordingly.

For the V2V communications, we try to address the problem of medium access organization in high VANET dynamic environments. VANET topology, due to high nodes' mobility, changes rapidly, thus, introducing high communication overhead for exchanging new topology information [13,14]. Several control schemes for media access and topology managements have been proposed [13, 15, 16,17]. One of these schemes is establishing a hierarchical clustering structure within the network. The clustering allows the formation of dynamic virtual backbone that can be used to organize media access, to support QoS, and to simplify routing [13, 18,19]. Mainly, nodes are partitioned into clusters, each with a Cluster-Head (CH) node that is responsible for all management and coordination tasks of its cluster.

In order to have efficient channel access methods using VANETs' clustering schemes, it's very important to make VANET topology less dynamic by forming local strongly connected clusters, thus, increasing the stability of the network topology on the

global scale. Therefore, we focus on the medium access organization and the stability of the network using clustering in VANETs.

Clustering has been used as one of the methods to organize medium access in wireless networks [18, 20-23]. Media access techniques in cluster-based schemes should guarantee access fairly to all cluster members such that, every cluster member can have the chance to exchange its data. Different VANET clustering schemes proposed different media access protocols. However, most of these techniques fall into the following categories: Scheduled-based for intra-cluster communications, contention-based for inter-cluster communications and cluster-head to cluster-head communication for multi-hop data dissemination. In these proposed schemes, the scheduled-based technique is used to avoid interference among cluster members.

In this dissertation, we propose a new medium access technique that can be used for intra/inter cluster communications and management. This protocol integrates the centralized approach of cluster management and the universal way of forwarding data in VANET, where the farthest vehicle forwards data backward in an effort to increase the coverage area. In this technique, time is divided into cycles; each cycle is shared between service and control channels and divided into two parts. During the first part, leveraging Time Division Multiple Access (TDMA), service channel will be used for Intra-cluster management and safety message delivery within the cluster. In the second part, neighboring clusters will exchange safety messages and advertisements over the control channel using media contention-based techniques. In parallel with the second part, cluster members can use service channels to exchange non-safety data with one another and with members of neighboring clusters.

Ensuring stability is another major challenge for clustering algorithms especially in a highly dynamic environment. Thus, efficient clustering algorithms should not only focus on forming a minimal number of clusters as many existing algorithms do, but maintaining the current cluster structure and keeping the overhead at the minimum level. Most of the existing VANET clustering algorithms are derived from the MANET clustering schemes [13, 21-26]. However, these algorithms lack a technique to capture the mobility characteristics of VANET nodes and fall in a major drawback of forming clusters considering only position and direction of vehicles located in geographic proximity regardless of their high relative speed. We believe that the existence of group members in the same geographic area doesn't mean that they exhibit the same mobility patterns, e.g., vehicles on the left lanes move faster than the vehicles on the right lanes, and thus their relative speed might be very high.

Since the main goal of clustering is to make global topology less dynamic, we believe that, changes in the network topology on the global scale are directly related to the stability of local clustering structure. Therefore, in order to enhance their stability, clustering models need to be redefined so that they are characterized based on the full status elements: speed difference, location, and direction rather than considering only position and direction. In this thesis, we introduce a new clustering approach with the aim of increasing network stability and make it less dynamic. This approach takes the speed difference, in addition to the location and direction, into consideration. The proposed clustering algorithm runs on all nodes in a fully distributed fashion. This algorithm is used to divide the network nodes into clusters such that when the network is finally partitioned (clustered), the probability of partitioning along cluster boundaries is achieved

with high probability. This means vehicles with high mobility are grouped in one cluster and vehicles with low mobility are grouped in another cluster. We also propose a new multi-metric election method that can be used by nodes to determine their suitability to become cluster-heads (CH). A simulation was conducted to evaluate our method and compare it with the most commonly used clustering methods. The Simulation results show that our technique provides more local stable cluster structure which results in a more stable network structure on the global scale. The proposed method reduces the average number of clusters changed per vehicles and increases the cluster lifetime significantly.

The remaining chapters of this dissertation are organized as follows:

Chapter 2 – presents background information and review of the past research done on the admission control and scheduling methods used by RSUs, and on the cluster formation and cluster-based media access in VANET.

Chapter 3 – presents, in details, the proposed virtual task finish time admission control algorithm used by the RSU. It also presents a detailed explanation of our cluster formation algorithm followed by the cluster-based media access technique in VANET.

Chapter 4 – describes the simulation environment, the performance evaluation metrics, and the simulation results of the comparison between our proposed techniques with others.

Chapter 5 – presents the conclusion.

Chapter 6 – presents the future work.

CHAPTER 2: BACKGROUND AND RELATED WORKS

The literature has numerous studies proposing different methods for resource management in the wireless networks, especially the admission control and scheduling problems. There is also a large body of the literature studied the concept of clustering to organize the media access and to increase the stability of the mobile networks. This chapter presents an overview of the most recent works that have been developed for admission control problem and the clustering methods in VANET.

2.1 Admission Control and Scheduling in VANET

The main task of the Admission Control is to decide to admit or reject new requests (upload/download) depending on whether the requirements of the new task will be fulfilled while the requirements of all in-progress services are guaranteed. The admission control algorithm tries to determine how resources are allocated and makes its decision based on that. Different factors are taken into account, like the data size, the trip time of the vehicle under the RSU transmission range, the number of already admitted tasks, etc. Once the task is admitted, the RSU has to grant with high probability the completion of the task. Otherwise, the admission control algorithm is not efficient.

Many research papers proposed different admission control and scheduling methods for single hop wireless networks. Most of the proposed schemes targeted the long-term sessions like multimedia services [27, 28, 29]. For example, the authors of [30] took into consideration the coexistence of the Real-Time (RT) and Best Efforts (BE) services. The proposed method tries to improve the BE services by giving high priority to the RT packets only when they are close to their deadlines. In [9], the authors used the traffic

characteristics given in the traffic specification element of the IEEE 802.11e to derive the guaranteed rate for the flows. The impact of the road traffic dynamics (e.g., vehicle speed, density, and number of arrivals) and the Access Point's (AP) characteristics (e.g., transmission range and data transmission rate) on the amount of downloaded data was studied in [31]. In [32], the admission control was studied using the Earliest Deadline First (EDF) algorithm. In wireless communications, most of the admission control schemes, proposed for one-hop communication networks, mainly focus on long connection duration flows like multimedia services (audio/video). In [33], the authors studied the call admission for Voice over IP (VOIP) flows, where a technique called virtual career sensing was proposed to estimate the impact of the new flows on the admitted ones. Some other studies were proposed for the short connection durations like the RSU to vehicle cases. The authors of [34] proposed a lower layer optimization used for scheduling when multiple vehicles are in the range of the RSU. This work was built based on the opportunistic scheduling proposed in [35], which basically assign the channel to the node with good signal quality and ignore the others with weaker signal conditions. However, this method is unfair because the medium is shared among nodes with good Signal to Noise Ratio (SNR), and the vehicle that doesn't share the medium, because no other vehicles are simultaneously with good SNR, will get better throughput than others. In addition to that, vehicles that happen to be shadowed will never get a good SNR and therefore might never get the chance to transmit.

The authors of [36] proposed a scheduling method for download/upload between vehicles and RSUs. However, the authors didn't take into consideration that the channel status sometimes can be in bad conditions, and assumed that packets can be delivered as

long as the vehicle is within the RSU range. They also deal with the task as a single packet to be transmitted.

In [11], the authors focused on evaluating the risk of the admitted tasks. The objective is to determine whether all tasks can be admitted within a quantified risk [11]. They introduced a new metric to evaluate the risk of the task. The metric is used to estimate the total data size that can be transmitted before the vehicles depart from the RSU range. The proposed algorithm gives the task with the minimal transmission rate preference over others even if they are going to leave the RSU at the same time. The algorithm doesn't take into consideration the amount of resources reserved for the task to give it more priority to complete. It's unfair to make all vehicles have the same risk. The risk of the vehicles that have been using the bandwidth for long period of time should be kept at the minimum level, and be minimized as long as the vehicle progresses and comes closer to leave the RSU transmission range. The algorithm uses linear programming to determine the solution and to generate the transmission plan. The algorithm assumes that the solution is always feasible for the in-session tasks which might not be the case always. In the following we will discuss in details the work proposed in [12].

2.1.1 Maximum Freedom Last Scheduling Algorithm for Downlinks of DSRC

Networks

The authors of [12] proposed a Maximum Freedom Last (MFL) scheduling algorithm for V2R communications. The MFL algorithm was proposed to minimize task failure and to reduce the handoff rate under the maximum tolerable delay. The authors assume that the RSUs are fully deployed along the road side as shown in Figure 2.1. The

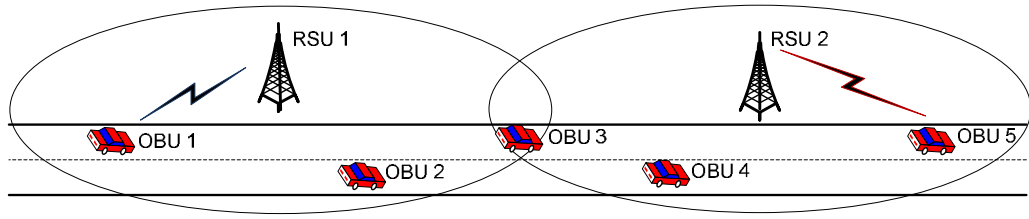


Figure 2.1 RSU deployments along the roadside

MFL algorithm schedules the service according to several factors like: the remaining dwell time of the service channel, the data size, the transmission time, and the maximum tolerable delay.

- System Operation

The RSU broadcasts a Road side Service Table (RST) via the Control Channel (CCH) to announce the service provisioning. The On Board Units (OBUs) compete to send OBU Service Table (OST). The response duration in which the OBUs can send OSTs is defined as the CCH wait time T_w . The number of the admitted OBUs in each cycle is restricted to γ . The RSU and the admitted OBUs use the Service Channel (SCH) for data transmission. The time duration of SCH is $T_{s,max}$. The MFL algorithm classifies OBUs as new, handoff, and ongoing OBUs. New OBUs are the ones that just sent the OSTs, the handoff are the ones that have just completed the handoff procedure but not listened to the RST, while the ongoing OBUs are the ones with unfinished data transmission. The RSUs send data to the OBUs according to the service list. If the SCH time $T_{s,max}$ expires during the transmission, then the transmission is suspended. The OBUs and the RSU jump to the CCH for a period T_w . If no high priority RST is received

after T_w , then the suspended task is resumed after the OBU and the RSU are jumped back to the SCH. If handoff OBUs or new OBUs request services during the CCH time, then the MFL scheduling algorithm is executed once again to create a new list and a data volume assignment table.

- The MFL Scheduling Procedure

The MFL algorithm assigns a higher service priority to the OBUs that have high chance to complete their service completely. Among those nodes, the ones with the highest degree of freedom will be served last. The OBU that has a lower transaction time and longer remaining SCH dwell time is considered the one that has a higher degree of freedom. The OBU with higher degree of freedom can tolerate longer transmission delay; therefore, other OBUs can be serviced before it. The authors defined a Weighting factor that is a function of queuing delay and maximum tolerable delay used to adaptively adjust the service priority and the service failure. The MFL algorithm runs in four phases as follows:

- A. *Initialization phase*: each OBU belongs to a service set A , is assigned with its scheduling parameters like virtual finish time FT_i and virtual start time ST_i . The FT_i is basically the SCH dwell time D_i , while the ST_i is $(FT_i - TX_i)$, where TX_i is the remaining transmission time. The OBUs that will finish their service before leaving the RSU are members of the $A+$ set, while the ones that will be partially serviced are members of $A-$.
- B. *Reverse Lineup Phase*: An iterative process is executed, as long as $A+$ is not empty, to construct a temporary list F based on the priority index of the OBUs

in $A+$. The OBU with a high priority index is added to the list and gets eliminated from set $A+$. The priority index of the OBU is higher when its queuing delay is large and still within the tolerable delay.

- C. *Transmission Time Pileup Phase*: in this phase, different parameters like the SCH dwell time and the queuing delay of OBUs in the $A-$ are updated. In addition to that, a service list is constructed by adding the OBUs in the temporary list F after being sorted in a reverse order to the list. The completely served OBUs are scheduled, and the algorithm enters the final phase
- D. *Partial Service Phase*: the OBUs that are selected in this phase will be partially served. The OBU that has the longest remaining SCH time in $A-$ group will be added to the end of the service list.

The proposed method doesn't take into consideration the change of the channel conditions as the vehicle approaches and leaves the RSU range. It doesn't also have a technique to evaluate the risk of the vehicles, especially the ones that have been getting service for long period of time. In the overloaded scenarios when the plan is very tight, the method tries to serve more vehicles by pushing the task finish time as close as possible to its leaving time. This increases the risk of the vehicle, thus, increasing the failure rate. This can easily happen because, as the vehicle moves away from the RSU, the channel conditions become unpredictable and the link between the RSU and the OBU becomes weaker. Another drawback is the proposed technique assumes that the RSUs are deployed along the roadside as shown in Figure 2.1. However, this is not realistic,

especially during the early phases of deployments whereas the RSUs will be sparsely deployed (this will be associated with very high cost).

2.2 Clustering in VANET

Clustering in VANET is basically grouping a set of vehicles that share the same mobility patterns in a logical entity called cluster (Figure 2.2). This group should elect a node called Cluster-head, which will be responsible for all inter/intra- cluster communications and managements. Clustering allows the formation of dynamic virtual backbone to organize media access, to support QoS and to simplify routing [13, 18, 19]. Ensuring stability is the major challenge for clustering algorithms especially in a high dynamic environment like VANET. A successful dynamic clustering algorithm should achieve a stable cluster topology with minimal communications overhead and minimal computational complexity [37]. Several issues having impact on the performance of the designed protocol need to be considered. These issues, proposed by authors of [38], are:

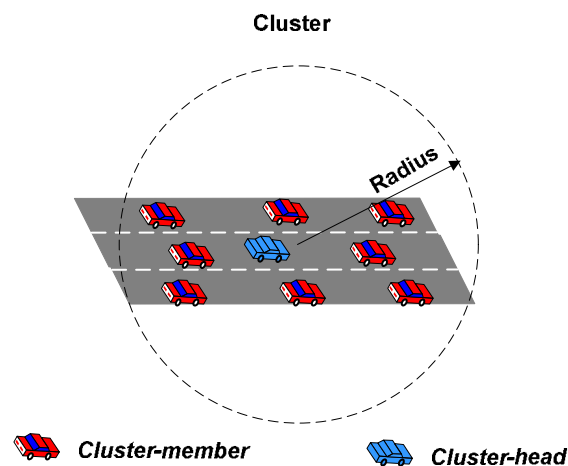


Figure 2.2 VANET cluster

- The clustering algorithm should consider the group mobility patterns.
- The algorithm must incur minimal clustering overhead, be it cluster formation or maintenance overhead.
- Network-wide flooding must be avoided.
- Optimal clustering may not be achieved, but the algorithm must be able to form stable clusters should any exists.

In Section 2.2.1, we summarize the most recent methods for cluster-based media access organization in VENET environment. In Section 2.2.2, we focus on the cluster formation algorithms used in VANET.

2.2.1 Cluster-based Media Access Control in VANET

Many research papers addressed the inter/intra-cluster organization and task coordination. In this subsection, we will briefly review two of the most recent cluster-based media access organization presented in [22,23].

2.2.1.1 Media Access Concept for VANETs Based on Clustering

The authors of the clustering algorithm [22] proposed a protocol for VANET cluster-based schemes that relies on the Cluster Based Location Routing (CBLR) [21] technique to form new clusters. In this method, the states of the nodes are similar to these used by the CBLR method. The only difference is that the node can be a member in more than one cluster and this node is called a Gateway.

- Cluster Formation

At the very beginning, the node is in the undecided state and waits for the HELLO messages from other nearby nodes for a certain period of time. Upon the reception of the messages, the node takes the appropriate decision to change its state. If no messages are received during this period, the node remains in the undecided state until it receives a message from a new node. To track the topology changes of the network, each node maintains two tables: one for all nodes it can hear and the other for the adjacent clusters. To build and update these tables, the nodes must exchange “HELLO” messages on a regular basis. The node must also include the ID of its cluster in the “HELLO” message. Each member node knows about only one cluster-head in its surrounding, whereas a gateway has more than one cluster-head in its table. Nodes get to know about cluster-heads either via “HELLO” messages received directly from the cluster-head or via information received from neighboring nodes.

Since VANET is very dynamic and the topology of the network changes very frequently, the cluster members and the cluster-head try to use their tables to decide on changes in their states. When a cluster member leaves the range of a cluster-head, it checks whether this is the only cluster it was a member of. If so, its state goes into an undecided state. But if the node is in more than one cluster, it stays in the member or gateway state. For the cluster-head, the case is little bit different. When two cluster-heads come into direct transmission ranges (They can receive their HELLO messages directly), one of them must give up its state and become a member of the other one. The decision of which one keeps its cluster-head state is based on a weighted factor W_v , which takes into account: the connectivity, the mobility, and the distance to the neighbors. The connectivity is given as the difference to the optimum number of nodes. The mobility is

calculated based on the difference of velocity of the nodes (the node with similar velocity to the most nodes in its neighborhood will cause less changes in the cluster membership than a node which is much faster or slower than the rest). Combining these measures, a weight factor can be calculated. This weight factor shows the suitability of a node to become a cluster-head, the smaller W_v is, the better it is qualified to be a cluster-head.

- The media access control protocol

The proposed algorithm presents a media access protocol that depends on the TDMA technique which divides the medium into time slots. These time slots are grouped into frames. The frame consists of two phases; the first one is called the direct link phase and the second is called random access phase. The cluster-head sends “HELLO” message at the beginning of each frame. Then the cluster-head sends a control message that contains information about the assignment of the slots. Each time slot will be given an ID and only the node with matching ID is allowed to send during this time slot. After that, nodes send their data according to the schedule sent by the cluster-head. In the direct link part of the frame, nodes within a single hop destination, communicate directly (no need for cluster-head). But for multi-hop connections, the CBLR technique can be used. In the second phase of the frame, nodes use the random access method to access the media. During this phase, nodes who are not members can join the cluster and register at the cluster-head as a cluster member. The length of this phase is variable and depends on the number of time slots the cluster members requested for their data.

2.2.1.2 Cluster-Based DSRC Architecture for QoS Provisioning over Vehicle Ad

Hoc Networks

In [23], the authors proposed new media access technique for VANET based on clustering. This method integrates the clustering algorithm, contention-free, and contention-based media access to support the real-time transmission of safety messages. In this method, the seven channels of the DSRC are assigned new functions and defined as follows: Ch178 is Inter-Cluster Control (ICC) channel, Ch174 is Inter-Cluster Data (ICD) channel, Ch172 is Cluster Range Control (CRC) channel, and the remaining channels are called Cluster Range Data (CRD). The authors of this method didn't use any particular technique for a cluster formation. They assume that the cluster-head is always at the center of the cluster. Each vehicle is assumed to have two DSRC transceivers. The cluster-head uses one transceiver for contention free over the CRC channel to collect and deliver safety messages as well as control packets within the cluster. The second transceiver is used to transmit the collected safety messages to nearby cluster-heads via the ICC channel. Each cluster member can use one transceiver to communicate with its cluster-head via the CRC channel, while the other one will be used to transmit all non-real-time traffic using one of the ICD/CRD channels assigned by the cluster-head. To accomplish the operation of the whole system, the proposed technique is divided into three core protocols, namely, the Cluster Configuration Protocol, the Inter-cluster Communications Protocols, and the Intra-Cluster Coordination and Communication Protocol.

- The Cluster Configuration Protocol

This core protocol is used by vehicles to control the transition from one state to another. Each vehicle in this method can operate under one and only one of the following states: Cluster-head (CH), Quasi-Cluster-head (QCH), Cluster Member (CM), and Quasi-Cluster Member (QCM). When the vehicle is in the CH state, one of its transceivers operates on the ICC channel to forward the collected safety messages to the neighboring clusters and the other transceiver uses the CRC channel to collect or broadcast safety messages from/to cluster members. If the vehicle is in the QCH state, this means that it is neither a CH nor a CM. In this state, one transceiver works on the ICC channel so that it can receive and send safety messages, while the other transceiver is turned off. If the vehicle switches to the QCH, it functions as cluster-head except for the ability in forming clusters. When the vehicle switches to the CM state, one transceiver works on the CRC channel to receive the consolidated safety messages and send their own safety messages as well as data reservation requests, while the other transceiver operates on the CRD/ICD channels. Finally, when the vehicle state becomes QCM, it uses one of its transceivers to operate on the ICC channel. Switching the transceiver to ICC guarantees that the vehicle can receive and send safety messages. This ensures that the vehicle can send and receive safety messages even if it temporally loses contact with the cluster-head. The second transceiver uses the CRC channel to be able to resume the communications with the previous cluster-head.

- The Inter-Cluster Coordination and Communication Protocol

In this protocol, the CH employs the TDMA technique over the CRC channel to send and receive safety messages. In the CRC channel, time axis is divided into time slots with equal length T . The length of the time slot depends on the number of cluster

members and the length of the cycle. The operation sequence of this core protocol can be summarized as follows: First, the cluster-head creates a schedule list specifying each vehicle when it can transmit according to the total number of cluster members. The cluster-head then distributes this list to all cluster members. Each cluster member receives the list and sends its safety message and data channel reservation requests during its own time slot. The cluster-head collects these messages and then broadcasts them back to the cluster members via the CRC channel. The cluster-head also transmits the collected messages to the neighboring clusters via the ICC channel. Finally, the cluster members can use the second transceiver by switching into ICD or CRD to send and receive non-real-time data. The communication between two nodes within the same cluster is performed using direct link without contention. Since vehicles are equipped with two transceivers, the safety messages and the non-real-time data can be serviced concurrently.

- The Inter-Cluster Communication Protocol

This protocol is used to organize the communication between neighboring clusters. The contention based technique will be used by the cluster members to access the media. The non-real-time traffic will be sent on the ICD channel. Therefore, vehicles from different clusters use the contention based method (IEEE 80.11) [39] to access the common ICD channel to send and receive this type of data. The real-time safety messages will be exchanged over the ICC channel. The CH, QCH, and QCM nodes from different clusters contend for the shared ICC channel to transmit the safety messages.

2.2.2 Cluster Formation Algorithms

In the previous section (2.2.1), we summarized some VANET clustering methods focusing on the media access organization. In this section, we try to summarize the most recent studies about cluster formation algorithms in VANET, and the main parameters that are considered during the process of cluster formations.

Most of the proposed VANET cluster formation algorithms were derived from MANET clustering schemes [37], [38], [40-48]. However, none of these methods considered all mobility characteristics of the VANET nodes. The clustering algorithm proposed in [24] is basically the Lowest ID used in MANET with a new modification. The authors included the leadership duration as well as the direction in the lowest ID algorithm to determine the node to be a cluster-head. The Leadership Duration (LD) is defined as the period the node has been a leader since the last role change. The higher the leadership duration, the more qualified the node is to be a cluster-head. Therefore, the cluster-head rule is: choose the node with the longest leadership duration and then choose the one with the lowest ID. The formation of clusters is based on beacon signals broadcasted by the VANET nodes. Each node announces itself as a cluster-head and broadcasts this to all neighbors. If it receives a reply from a neighboring node with a lower ID and a higher leadership duration, then the node changes its state to a cluster member. When a node leaves its cluster, it looks for another cluster in the neighborhood to join. If none of the neighboring nodes or the neighboring cluster-head satisfy the cluster-head election rules, then the node claims itself as a cluster-head.

The work in [24] was modified and presented in [25]. In addition to the LD and the Moving Direction (MD), the authors introduced the Projected Distance variation (PD),

which means distance variation of all neighbors over a period of time. Each node is associated with a utility Weight (uW) of 3 parameters (LD, PD, ID), where the ID is the identifier of the node. The LD parameter is given the highest weight. To define the total utility weight, a lexicographical ordering of the 3 parameters (LD, PD, ID) is used. For example, the utility weight (LD1, PD1, ID1) is greater than (LD2, PD2, ID2) if either $LD1 > LD2$ or $(LD1=LD2 \text{ and } PD1 < PD2)$ or $(LD1=LD2 \text{ and } PD1=PD2 \text{ and } ID1 < ID2)$. Based on this, the LD value has maximum importance and its value is the primary factor to determine the total uW. However, in both works [24] [25], the node that has higher connectivity degree might not be elected to lead the cluster if there is another node that has longer leadership duration. This will produce less stable cluster structure, because having longer leadership duration doesn't mean that the node has high connectivity degree that gives it the ability to lead the cluster.

In [49], the authors proposed a heuristic clustering approach for cluster-head elections that is equivalent to the computation of the Minimum Dominating Sets (MDS) used in graph theory. This approach is called Position-based Prioritized Clustering (PPC) and uses geographic position of nodes and the priorities associated with the vehicles traffic information to build the cluster structure. For clustering purposes, each node is assumed to broadcast a small amount of information of itself and its neighbors, which is referred to by 5-tuples (node ID, cluster-head ID, node location, ID of the next node along the path to the cluster-head, and node priority). A node becomes a cluster-head if it has the highest priority in its one-hop neighborhood and has the highest priority in the one-hop neighborhood of one of its one-hop neighbors. The priority of the node is calculated based on the node ID, current time and the eligibility function. A Node having

longer travel time has higher eligibility value, and this value decreases when the velocity of the node deviates largely from the average speed.

The authors of [50] proposed a cluster formation technique where nodes use the Affinity Propagation (AP) method to pass messages to one another. Basically, the proposed algorithm takes an input function of similarities, $s(i, j)$, which reflects how well suited data point j is to be the exemplar of data point i . Nodes exchange two types of messages: responsibility, $r(i, j)$, indicating how well suited j is to be i 's exemplar, and availability, $a(i, j)$, indicating the desire of j to be an exemplar to i . The nodes use the self responsibility, $r(i, i)$, and self availability $a(i, i)$, to reflect the accumulated evidence that node i is an exemplar. When a node's self responsibility and self availability become positive, that node becomes a cluster-head. The authors proposed that a clustering decision is made periodically every Clustering Interval (CI) period, and a clustering maintenance is performed in between CI. However, having cluster members make clustering decision every CI will increase the probability of re-clustering. Also the authors didn't take into consideration the speed difference among neighboring nodes.

In [45], the authors proposed a clustering technique for MANET applications. They introduced an Aggregate Local Mobility (ALM), which is a relative mobility metric that used the Received Signal Strength (RSS) at the receiving node as an indication of the distance between the sender and the receiver. However, the use of RSS is highly unreliable, especially in VANET environment, as indicated by other researchers [51]. The paper [45] also did not take the speed difference as a parameter to form clusters.

In [51], the authors basically use the Aggregate Local Mobility (ALM) proposed in [45], with some modifications, as a criterion for triggering cluster re-organization. Originally, the ALM is a relative mobility metric that uses the Received Signal Strength (RSS) at the receiving node as an indication of the distance between the sender and the receiver [45]. The ratio of the RSS of two successive periodic hello messages indicates the relative mobility between the two nodes. In [51], the authors used the location information embedded in the periodic hello messages to determine the relative mobility of the nodes instead of using the signal strength. In this technique, if two cluster heads come into direct communication range, they exchange more than one packet in a predefined period of time in order to consider the merging between the two clusters. In case merging takes place, the cluster-head with the lower ALM value maintains its role while the other gives up its role and becomes a member node in the new cluster. However, the nodes that lost their cluster-head due to merging or mobility and can't find nearby clusters to join, they will all become cluster heads almost at the same time. There will be a period where they will organize their minds as to who will be the new cluster-head. However, the authors did not take the speed difference of neighboring nodes into consideration.

2.2.2.1 The Cluster-Based Location Routing

The Cluster Based Location Routing (CBLR) [21] is a reactive [52], [53] type of protocols. The location of nodes is used by this method to improve the efficiency of the routing protocol. The operation of the proposed algorithm can be divided into four phases as follows: Cluster Formation, Location discovery, Routing of data packets, and Maintenance of location information.

- Cluster Formation.

The initial step of this technique is accomplished by the formation of the clusters. The formed cluster can have at least one cluster-head and zero or more cluster members. In general, the states of the nodes can be classified into: Undecided, Cluster-head, and Cluster Member. Initially every node is always in the undecided state. The node starts a timer and sends a “HELLO” message. If the undecided node receives a “HELLO” message from a cluster-head before the timer is expired, it becomes a member. Otherwise, it becomes a cluster-head. The cluster-head maintains a Cluster Table that contains the addresses and geographic locations of the member nodes, and a Cluster Neighbor Table that contains information about the neighboring clusters. The cluster-head frequently sends “HELLO” messages to inform others about its availability and to give chance for new members to join the cluster.

The network is divided into multiple clusters. The cluster-head takes the responsibility of exchanging data among neighboring clusters. The cluster neighbor tables are frequently distributed among clusters.

- Location discovery.

The protocol implements the reactive approach to communicate with the destination nodes. When a node needs to transmit data, it checks whether the destination is included in its cluster table. If the destination node is not included, it sends a Location Request (LREQ) packet. The cluster-head receives the LREQ and checks its table (The packet will be dropped if it has been received more than once). If the destination node is included in its table, it replies by uni-casting a Location Reply (LREP) packet to the

source node. Otherwise, it records the address of the received LREQ in its list and forwards the LREQ further to its neighboring cluster's head.

When the destination cluster-head receives the LREQ it extracts the information in the packet and records the location of the source node. The cluster-head sends a reply LREP via its neighboring cluster-head. The reply packet doesn't have to maintain a routing path, the path, instead, is determined from the location (the path traversed by the LREQ may be different from that traversed by the LREP).

- Routing of data packets.

Since both, the source and the destination nodes know their relative positions. The packets propagate from the source to the destination based on the location of the nodes. As the transmission is in the direction of the destination node, the path will be shorter than the other routing methods (In routing methods, the path found might not be the shortest one).

- Maintenance of location information.

The CBLR algorithm was designed to operate in very high mobile and dynamic environment. This method allows the sender to update its location information before sending every packet. Similarly, the receiver updates its location and then replies to the sender.

2.2.2.2 Clustering Formation for Inter-vehicle Communication

The Clustering formation for inter-vehicle communication [54] basically classifies vehicles into groups based on the speed range of vehicles. Vehicles that fall in the same

speed group belong to the same cluster. The authors defined 7 groups based on the minimum and maximum value of the speeds that the vehicles can use. The range of the speed difference is 15KMph for all groups except group 0 and group 6, which is 30KMph and 10KMph respectively. The authors adopted the "First Declaration Wins rule", which is basically a node that first claims to be a cluster-head remains as a cluster-head and rules the rest of nodes in its clustered area. According to the authors' definition, if a cluster member speed changes such that the node travels at a speed that is different from the group speed for a period of time, then, the node must update its clustering group and should seek for a new cluster even though the node is still under the transmission range of its current cluster-head. The authors proposed that the cluster-head adjust its transmission range when the density of the vehicles is very high. The cluster-head can reduce its transmission range to include less number of vehicles to reduce the management overhead. One of the drawbacks of this technique is that the first vehicle that claims to be the cluster-head may have its speed and location on the boundaries of both parameters. This cluster-head might lose the communications with its members soon. Moreover, having the cluster-head adjust its transmission range according to the speed of the group, makes the cluster members on the cluster boundary out of the transmission range of the cluster-head. Thus, these nodes will leave the cluster, which results in an increase of the cluster change rate.

CHAPTER 3: THE PROPOSED METHODS FOR CHANNEL ACCESS AND CLUSTERING IN VANET

3.1 Channel Allocation for RSU Based on Virtual Task Finish Time

3.1.1 Motive and System Description

The Admission Control (AC) and Scheduling schemes for RSUs are used to determine how resources can be allocated to the requests coming from passing by vehicles. If the requirements of the new arrival task can be fulfilled while guaranteeing the requirements of the current in-progress sessions, then the task is admitted otherwise it's rejected. Mainly, the AC is used to handle the situations when the RSU is close to the overloaded conditions. In such scenarios, accepting more nodes to increase system throughput and bandwidth utilization means pushing the task finish time to the edges (RSU departing time), which results in a very tight time allocation transmission plan assigned to the admitted vehicles as in [11] [12]. In this case, the risk of those tasks increases and the probability of task failure becomes higher. This could happen due to fluctuations in the channel conditions, especially when vehicles move away from the RSU where the signal strength becomes very weak.

To compromise between reducing the risk of the vehicles and increasing the system throughput, we propose a new technique with the goal of keeping the number of admitted tasks at higher levels, and at the same time reducing the risk of those tasks. The method focuses on reducing the risk of the vehicles that have been getting service for long period of time and are about to leave the RSU range. The motive behind our method comes from special characteristics and physical limitations (e.g., road structures, flows constraints,

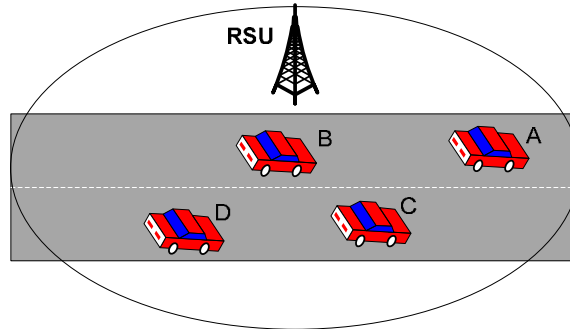


Figure 3.1 Vehicles in the RSU communication range

safe distance, etc) which, can be explained with the help of Figure 3.1 as follows: the figure shows that 4 vehicles getting service from the RSU. Consecutive vehicles on the same lane, e.g., *A* and *B*, try to keep Safe Distance (*SD*) (the safe distance is 1.5 – 3.0 sec). Vehicle *A* is closer than *B* (both are on the same lane) to depart the RSU region. Similarly *C* and *D* (both are on the same lane). Since vehicle *A* is closer than *B* (similarly *C* closer than *D*) to leave the RSU range, then vehicle *A* should be treated differently from vehicle *B*. Vehicle *A*'s risk should be minimized, however, vehicle *B* can tolerate some risk because vehicle *B* will not leave the RSU range before *A* does (if vehicle *A* leaves the RSU region at time t_A , then vehicle *B* will leave, at least, at time $t_A + SD$, where *SD* is the safety distance between *A* and *B*). Calculating the expected task finish time of vehicle *A* and allocating a virtual transmission plan that includes an extra reserved time for vehicle *A* (similarly *C*) can help minimizing the risk of the vehicle (this extra time will automatically be transferred to the transmission plan of vehicle *B* once it becomes the first vehicle on its lane to depart the RSU region). The reserved extra time is very small compared to the inter-arrival time, and this will not prevent the RSU from admitting more tasks.

3.1.2 System Model

The proposed work assumes that RSU and OBUs are equipped with DSRC [8] and GPS devices. Vehicles enter the communication range of the RSU and request a service. The RSU admits the new arrival as long as this admission doesn't increase the risk of the vehicles under service (especially the ones that have been getting service for long time and are about to depart the RSU range), and as long as the RSU is able to finish the task before the vehicle departs the RSU region.

For our method to function properly, it has to predict the position of the vehicle while it's under the RSU transmission range. After that, the method can use the predicted location of the vehicle with respect to the RSU position to set the transmission rate and to allocate the necessary time shares for the vehicle. Vehicle's future position can be predicted using the GPS and the mobility information (speed, acceleration, direction, current position, etc). Same as proposed by [11], channel status can be represented using data transmission rate and packets transmission failure. For data transmission rate, we use $r_{i,t}$ to denote the maximum transmission rate for vehicle i at time t . Since transmission over the wireless media is prone to errors, then we use $p_{i,t}$ to denote the error probability of transmission to vehicle i at time t (the transmission error probability can be determined experimentally by varying the distance between the RSU and the nodes and observe the number of failed packets over many runs). Same technique used in [11] [9].

In our proposed method, the time is divided into time slots called cycles c . Each cycle is of length c sec. Each task will be assigned a time share during each cycle once started to receive data flows from the RSU. Each task must have a minimal transmission

rate guarantee at each cycle once data flow begins. Upper layer applications require continuous communications to maintain their connectivity [11]. In the rest of this section, we use the terms task i and vehicle i interchangeably to refer to the same object.

3.1.3 Problem Analysis

The main goal of the admission control algorithms is to determine how resources are allocated to the new arrival tasks. The task will be admitted if the AC algorithm is able to allocate the necessary resources otherwise it's rejected. Once admitted, the task should be allocated sufficient resources to guarantee its successful completion. The admission control algorithms try to increase the system throughput and to reduce the risk of the admitted tasks. The trade-off between accepting more nodes to increase the system throughput thus increase system utilization and at the same time reducing the risk of these admitted tasks can be managed by optimal resource utilization.

Given that the max transmission rate and the transmission probability error can be determined in advance (test field results can be found in [55]), the part that can be used to control the amount of transferred data is the portion of time shares to each admitted task. For each task, the time allocation plan and the distribution of the time shares over the vehicle's trip under the RSU transmission range should be managed carefully to guarantee the successful completion of the task.

For task i , the amount of data transferred in cycle j is determined by the amount of the time share, $w_{i,j}c$ ($0 < w_{i,j} \leq 1$) allocated during the cycle, the maximum transmission rate, $r_{i,j}$, and the probability of transmission error $p_{i,j}$. Therefore, task i

should be allocated a time share in each cycle until it's finished successfully, this can be represented as follows

$$\sum_{j=1}^{d_i} r_{i,j} w_{i,j} c (1 - p_{i,j}) \geq DR. \quad (3.1)$$

where $r_{i,j}$ is the maximum physical transmission rate, DR is the remaining data, and d_i is the expected task finish time. Since $w_{i,j}$ is the only parameter the algorithm can control ($r_{i,j}$ and $p_{i,j}$ can be determined experimentally), then, $w_{i,j}$ should be allocated such that

$$r_{i,j} w_{i,j} (1 - p_{i,j}) \geq r_{i,j}^{min}, \text{ for all } j \quad (3.2)$$

$r_{i,j}^{min}$ is the min transmission rate guarantee for task i in cycle j . This means each task has a max number of cycles, M_i , as follows:

$$\sum_{j=1}^{M_i} c * r_{i,j}^{min} \geq DR, \quad (3.3)$$

In our proposed algorithm, the risk of the vehicle will be reduced automatically as it progresses and moves toward the borders of the RSU transmission range. Our algorithm differentiates between two types of vehicles. The vehicles that are closer to leave the RSU transmission range (in terms of distance as shown in Figure 3.1), and the other vehicles that are behind the ones in the front. This depends on the number of lanes. The road with *one*-lane will have, at any time, only one vehicle closer than others to the RSU range borders. For a road consisting of n -lanes, there will be n vehicles that are closer than others to the RSU transmission range borders (those vehicles might be on different lanes as shown in Figure 3.1 or might be on the same lane).

We define two sets, $Q1$ to store the admitted vehicles (tasks) that are closer to the RSU transmission range borders (the number of vehicles in $Q1$ depends on the number of lanes, so, for n -lanes road, there will be n vehicles in $Q1$), and $Q2$ to store the other admitted vehicles (tasks). For task i in $Q2$, the expected task finish time, d_i , is calculated, and based on that the task is allocated a transmission plan. The vehicles in $Q2$ are not going to leave the RSU before the ones in $Q1$, therefore, the expected task finish time of those vehicles ($Q2$ members) can't exceed their departing time l and can be expressed as follows:

$$d_i \leq l_i, \quad \forall i \in Q2 \quad (3.4)$$

For each task in $Q1$, the algorithm tries to minimize its risk, therefore it's very important to distinguish between the actual calculated expected task finish time, d_i , and the virtual task finish time, $d_{i,ver}$, that will be used to allocate the final air-time plan for those vehicles. The actual expected task finish time for $Q1$ members should always be:

$$d_i \leq l_i - \Delta_{bt}, \quad \forall i \in Q1 \quad (3.5)$$

where Δ_{bt} is an extra time called Backup Time, which can be used to transmit the remaining data in case the task couldn't be finished at time d_i . The algorithm calculates the virtual task finish time, $d_{i,ver}$, for each member of $Q1$ and $Q2$ as follows:

$$d_{i,ver} = \begin{cases} d_i + \Delta_{bt}, & \forall i \in Q1 \\ d_i, & \forall i \in Q2 \end{cases} \quad (3.6)$$

This means we need to allocate a transmission plan called Virtual Transmission Plan, $W_{i,ver}$, for $Q1$ members assuming that the task will be finished at $d_i + \Delta_{bt}$ instead

of d_i . Although, we know that the task will be finished at d_i , but we still reserve extra time Δ_{bt} so that the RSU can transmit the remaining data before the vehicle leaves the RSU range in case the task was not finished at d_i . Eventually, vehicles in $Q1$ will finish their tasks or leave the RSU range, and the vehicles (members of $Q2$) that become now closer to leave the RSU region are removed from $Q2$ and added to $Q1$. The Δ_{bt} (or part of it in case it is partially used by tasks in $Q1$) will automatically be transferred to those vehicles. This is a dynamic process that can be explained in the example shown in Figure 3.2 (a) and (b)

Based on the above, the virtual transmission plan for any $Q1$ member is: $W_{i,vir} = \Delta_{bt} + \sum_{j=1}^{d_i} w_{i,j}c$, and the total allocated transmission plan for all $Q1$ members is $W_{Q1} = \sum_{i \in Q1} W_{i,vir}$. The transmission plan for any $Q2$ member is: $W_i = \sum_{j=1}^{d_i} w_{i,j}c$, and the total allocated transmission plan for all $Q2$ members is $W_{Q2} = \sum_{i \in Q2} W_i$. So, the total transmission plan W_{total} for K ($K = Q1 \cup Q2$) is:

$$W_{total} = W_{Q1} + W_{Q2} \quad (3.7)$$

In our algorithm, the extra time Δ_{bt} is reserved and will dynamically be part of the air-time transmission plan allocated to the vehicles that are closer to leave the RSU range. As soon as those vehicles finish their task or leave the RSU region, this extra time will automatically be part of the air-time transmission plan of the next vehicle to leave the RSU region.

First, we explain our method using the example shown in Figure 3.2 (a) and (b), and then we show the algorithm used to implement our method. For simplicity, we show an

example of a *one-lane* road (Our algorithm can be implemented on n -lanes road). The example assumes departures every cycle (in real life departures of the same lane depend on the SD that can be between 1.5-3 sec.). Initially, $v1$ entered the RSU range first and is allocated a plan, and then $v2$ entered the RSU range and assigned a transmission. After that, $V3$ arrived and squeezed into the total allocated plan. Figure 3.2 (a) shows the state of the system after admitting $v3$. Figure 3.2 (a) shows the state of the system for cycles $j, j + 1, j + 2$, and $j + 3$. Since $v1$ is the first one to leave the RSU range, then it's a $Q1$ member. Therefore, our algorithm allocates a virtual transmission plan $d_{v1,vir}$ ($d_{v1,vir} = d_{v1} + \Delta_{bt}$) for $v1$ (the green color for $v1$ shown in Figure 3.2 (a)). Vehicles $v2$ and $v3$ have their expected task finish time d_2 and d_3 equal to their virtual task finish time $d_{v2,vir}$ and $d_{v3,vir}$ respectively, and this time is very close to their leave time l_2 and l_3 . Each vehicle is assigned a time share, $w_{vi,j}$ in cycle j such that $\sum_{i=1}^m w_{vi} \leq 1$,

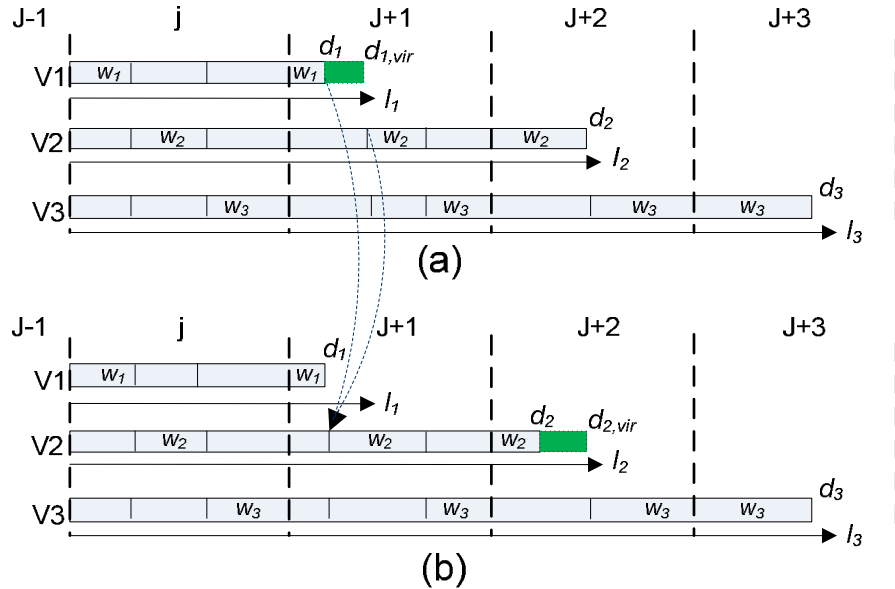


Figure 3.2 Illustrative example

$m = 3$ is the total number of tasks. The figure also shows the time shares assigned for each task in each cycle. Note that the time share for $v1$ in $j + 1$ includes the extra backup time Δ_{bt} (although $v1$ will finish its task at d_{v1}). If $v1$ is able to finish its task by d_{v1} (or anytime before $d_{v1,vir}$), then the remaining part of its share (the extra time Δ_{bt}) can be used (in addition to its original time share $w_{v1,j+1}$) by $v2$ to download more data during $j + 1$ thus bringing its task finish time d_{v2} earlier as shown in Figure 3.2 (b). Since our algorithm re-evaluates the transmission plan for each task every cycle, then, in the next cycle $j + 2$, $v2$ becomes the closer one to leave the RSU, therefore, the algorithm allocates a virtual transmission plan for $v2$ (the green color shows the gap between d_{v2} and $d_{v2,vir}$). The same procedure will be repeated for the other tasks.

In case the transmission plan is very tight (the task finish time is very close to the deadline) and the backup time is partially used by $Q1$ members, then allocating Δ_{bt} in the next cycle for each new $Q1$ member means reducing time share of other tasks ($Q2$ members). In this case, those tasks will be risked since their finish time might exceed their deadlines. Therefore, the proposed algorithm first tries to allocate resources for in-session tasks and then admit new tasks if there is enough room. Before discussing algorithm operation in details, we first explain the principle of resource compensation when the system is highly overloaded and the time transmission plan is very tight.

3.1.4 Time Shares Compensation

As mentioned earlier, once data flow starts, the task will be allocated a time share in each cycle until it's finished. Each cycle can be shared by multiple tasks and based on this we introduce the following definitions:

Definition-1: Tasks A and B form a compensation pair if both tasks have time shares in one or more cycles.

For a continuous flow of tasks admitted by the RSU, there will be a cycle, C_j , at which a new admitted task is scheduled to start such that no other early admitted tasks sharing the cycles $C_j, C_{j+1}, C_{j+2}, \dots$. And based on this we introduce the second definition

Definition-2: A new transmission plan W_{total} (with a new backup time) is created only when a new task, subsequent tasks as well, is admitted and scheduled to start getting service at cycle C_j such that no other already in-session tasks sharing the cycles $C_j, C_{j+1}, C_{j+2}, \dots$

The main goal of the algorithm is to guarantee the service for the in-session tasks before admitting new nodes. Therefore, if the backup time (or part of it) is used by $Q1$ members, then allocating backup time for the new tasks will be at the expense of $Q2$ members' shares (by reducing their shares), and if the transmission plan is very tight, then $Q2$ members might fail (their task finish time d_i will exceed their deadlines l_i). Let q be the number of tasks that are allocated backup time to be used during cycle C_j and δ be the portion of the backup time used by those tasks, then $(q * \Delta_{bt} - \delta)$ is always given to the next tasks ($Q2$ members) during C_j to help them transfer more data to bring their finish time as early as possible. Since the algorithm relies on Equations 3.1 through 3.11 (Equations 3.10 through 3.11 are shown in Section 3.1.4) to determine W_{total} , then reducing the share of the admitted tasks ($Q2$ members) to compensate the used part of the backup time will make the solution of these equations infeasible (d_i will exceed l_i because reducing the time share will delay the task finish time). To avoid risking all

$Q2$ tasks and at the same time guaranteeing with high probability the successful completion of all tasks belonging to the same W_{total} , the proposed algorithm tries to identify one task within W_{total} and reduces its share to compensate δ . The algorithm selects the task whose d_i is the latest (maximum) among all tasks belonging to the same W_{total} ; this task is the last one that started the service within W_{total} .

Reducing the time share of the selected task will delay its finish time and more reduction makes the task finish time exceed its deadline (section 3.1.4 shows how the algorithm handles this case). If h represents the task whose time share is reduced, then the amount by which the time share of the task is reduced is η ($\eta \leq \delta$), and the amount of data, S , the task would have received during η is:

$$S = \eta * r_{h,j}(1 - p_{h,j}) \quad (3.8)$$

where $r_{h,j}$ and $p_{h,j}$ are the transmission rate and the failure probability of task h during cycle j respectively. The task whose share is reduced should continue receiving the service. But, if $d_h > l_h$, then this will violate the rules of generating W_{total} using the algorithm. Therefore, to avoid violating the rules and make the algorithm be able to generate W_{total} , the algorithm, temporarily, assumes that the remaining data, DR_h , of task h is reduced by S . The algorithm continues generating the total transmission plan W_{total} using the new temporarily assumed $DR_{tmp,h}$ value. The temporary remaining data, $DR_{tmp,h}$, is calculated by:

$$DR_{tmp,h} = DR_h - S \quad (3.9)$$

The algorithm tries to allocate the transmission plan for task h based on $DR_{tmp,h}$ instead of DR_h (using $DR_{tmp,h}$ makes $d_h \leq l_h$). Every time the share of task h is reduced by η , the $DR_{tmp,h}$ is also reduced by S . The algorithm keeps monitoring task h and tries to increase its share if channel conditions improved and there is extra time to use.

Eventually, task h will be a $Q1$ member and will be eligible for extra time allocation. As long as S can be compensated once task h becomes a $Q1$ member, the algorithm will continue providing it with the required service. The algorithm allocates the whole extra backup time for task h to compensate S (remember d_h is the latest among all members in the current W_{total} and all other tasks that started the service after h belong to different W_{total}). The algorithm keeps checking whether S can be compensated or not. If S can't be compensated using the extra time, the algorithm will immediately drop task h from service, and will use the time that is supposed to be reserved for task h to admit new tasks. The algorithm tries to identify the tasks that might fail with high probability and attempts to drop them early to reduce the cost associated with waste of resources. Early dropping of the task that has consumed less amount of resources will reduce the cost associated with resources' wastage. Moreover, this task requires more resources in the future and if the task keeps using them, then all these resources is a waste. Therefore, early dropping allows us to use the resources efficiently.

3.1.4 Algorithm Description

The transmission plan is generated using linear programming technique as shown in Algorithms 1 and 2 of figures 3.3 and 3.4 respectively. The algorithm generates a list of

task finish time for admitted tasks by iteratively solving Equation (3.3) and Equation (3.10) through (3.11).

$$\sum_{j=1}^{d_{i,vir}} r_{i,j} w_{i,j} c(1 - p_{i,j}) \geq DR + \varepsilon, \quad \varepsilon = \begin{cases} r * \Delta_{bt}, & \forall i \in Q1 \\ 0, & \forall i \in Q2 \end{cases} \quad (3.10)$$

$$\sum_{i \in Q1, Q2} w_i \leq 1, \text{ for cycle } j \quad (3.11)$$

In equation (3.10), ε is used to reserve Δ_{bt} for $Q1$ members; the equation implies that the task should be completed by $d_{i,vir}$ cycles and this schedule should include Δ_{bt} as part of it. Equation (3.11) means the total shares in each cycle should not exceed 1.

At the beginning of the algorithm, we try to check whether the resources are sufficient for the in-session tasks by checking whether or not the solution of the Equations is feasible. If the solution is feasible, then the algorithm continues pushing the task finish time backward otherwise share reduction algorithm is called. The algorithm keeps pushing the task finish time by decrementing $d_{i,vir}$ by $\Delta\tau$ until the equations are violated. Then, the algorithm fixes $d_{i,vir}$ for all $Q1$ members and enters the inner loop, and repeats the same procedure for $Q2$ members starting from the vehicle whose task finish time is the min (those tasks have been getting service for long time and are about to leave the transmission range) as shown in line 16 of algorithm 1. The algorithm keeps decrementing until all d_i 's of $Q2$ members are fixed. The output of the algorithm is the total time plan consisting of the virtual time plan for $Q1$ and $Q2$ members.

If the solution is not feasible from the first iteration (lines 3 and 4 of algorithm 1), then share reduction algorithm is called. If no task was selected for share reduction, then the algorithm picks the task with max d_i (most likely this task started late and has not

consumed much of the bandwidth). The algorithm uses the same task for future share reduction. This task will be allocated the whole extra time once it becomes in $Q1$ to compensate the share that was reduced.

Algorithm 1: Time Plan Determination

```

1:  $K = Q1 \cup Q2$ 
2:  $d_{i,vir} = l_i, \forall i \in Q1, Q2$ 
3: test feasibility of eq. (3.2) (3.3) (3.10) (3.11)  $\forall i \in Q1, Q2$ 
4: if not feasible then
5:   Perform share reduction (Algorithm 2)
6: else
7:    $d_{i,vir} = d_{i,vir} - \Delta\tau, \forall i \in K$ 
8:   test feasibility of eq. (3.2) (3.3) (3.10) (3.11)  $\forall i \in Q1, Q2$ 
9:   if eq. are feasible then
10:    repeat steps 7 through 8
11:   else
12:     $d_{i,vir} = d_{i,vir} + \Delta\tau, \forall i \in K$ 
13:    fix  $d_{i,vir} \forall i \in Q1$ 
14:     $Q^2 \leftarrow \emptyset$ 
15:    while  $Q2 \neq \emptyset$ 
16:       $i \leftarrow \operatorname{argmin}_{i \in Q2} d_i$ 
17:       $d_{i,vir} = d_{i,vir} - \Delta\tau, \forall i \in Q2$ 
18:      test feasibility ( $Q1, Q2$ )
19:      if not feasible then
20:         $d_{i,vir} = d_{i,vir} + \Delta\tau, \forall i \in Q2$ 
21:         $Q2 = Q2 \cup Q^2$ 
22:        return  $Q2$ 
23:      else
24:         $Q2 = Q2 - \{i\}$ 
25:         $Q^2 = Q^2 \cup \{i\}$ 
26:      end if
27:    end while
28:    fix  $d_{i,vir} \forall i \in Q2$ 
29:  end if
30: end if

```

Figure 3.3 time plan allocation

Algorithm 2: Share Reduction

h : is the task whose share is reduced
 1: **if** h has not been determined
 2: $h \leftarrow \operatorname{argmax}_{h \in Q_2} d_h$ /* h should be within W_{total} */
 3: $DR_{new,h} = DR_h - S$ /* S is calculated using eq. 3.8 */
 4: **else**
 5: $DR_{new,h} = DR_{new,h} - S$
 6: **end if**
 5: **while** eq. (3.2) (3.3) (3.10) (3.11) not feasible
 6: $DR_{new,h} = DR_{new,h} - \beta$ /* β is smaller than S */
 7: **end while**

Figure 3.4 task share reduction

3.1.5 Allocation and Distribution of Time Shares

Sending more data to the vehicles that are in good signal quality is an efficient technique that has been widely used in the wireless networking [34] [11] (e.g., vehicles B and C in Figure 3.4 has high signal strength, while vehicles A and D have weaker signal strength). Since $r_{i,j}$ is determined based on the distance between the RSU and the vehicle, and $p_{i,j}$ can be derived from experiments, then the time share $w_{i,j}$ is the parameter that can be used to control the amount of data to be transmitted. The objective is to maximize the time share of the vehicles with high signal strength [11]. Therefore, for cycle j :

$$f = \max w_{i,j} \quad (3.12)$$

Subject to

$$\sum_{j=1}^{d_i} r_{i,j} w_{i,j} c(1 - p_{i,j}) \geq DR, \quad \forall i \in K \quad (3.13)$$

$$r_{i,j} w_{i,j} (1 - p_{i,j}) \geq r_{i,j}^{\min}, \quad (\text{for all } j) \quad (3.14)$$

$$(\alpha_i + \sum_{i \in K} w_i) \leq 1, \quad \alpha_i = \begin{cases} \frac{\Delta_{bt}}{c}, & \forall i \in Q1, \quad \text{only for the last } j \\ 0, & \forall i \in Q2 \end{cases} \quad (3.15)$$

$\alpha_i < 1$, and c is the length of the cycle. Equation (3.15) means Δ_{bt} should be part of the cycle at which the task of $Q1$ member is expected to finish.

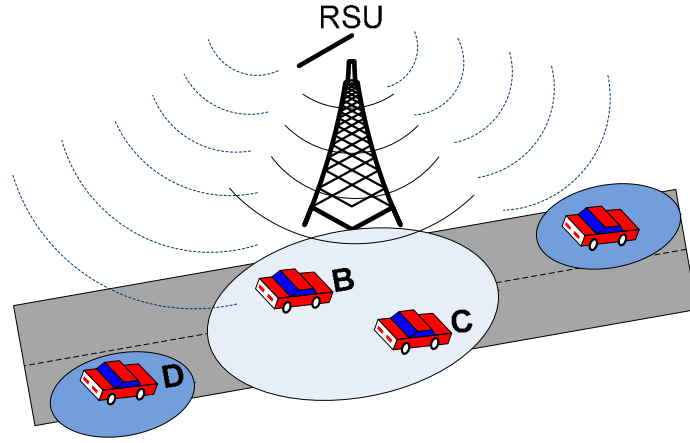


Figure 3.5 vehicles' positions with respect to the RSU

3.2 Media Access Technique for Cluster-based Vehicular Ad hoc Networks

3.2.1 System Description

The proposed protocol is a hybrid method that uses scheduled-based and contention-based approaches for Intra-Cluster and Inter-Cluster communications respectively. The design of our protocol is motivated by the fact that DSRC interface uses 7 non-overlapping 10 MHz channels. While the communication range of the control channel is 1000 or more meters, it is in the range of 30 to 400 meters for the service channels. Similar to [22], our proposed protocol takes advantage of the variation of communication

ranges of service and control channels such that, the control channel, CRL, will be used to deliver safety data and advertisements across neighboring clusters, and a service channel, called SRV, will be used to exchange safety and non- safety data within the cluster (Figure 3.6). Unlike [22] where each vehicle is assumed to have two DSRC interfaces, we think vehicles are very unlikely to have more than one DSRC interface. Therefore, we assume that each vehicle is equipped with a single DSRC interface and a GPS device. But, with one DSRC interface installed, the protocol must be designed to challenge the fact that DSRC interfaces demodulate one channel at a time [8]. This means, even though the DSRC interface has 7 channels, it can't use more than one channel at the same time. To solve this problem, we introduce the so called system cycle, which is divided into Scheduled-Based (SBP) and Contention-Based (CBP) sub-periods and repeat every T millisecond. Using this cycle, the proposed method can support numerous data delivery types having different requirements.

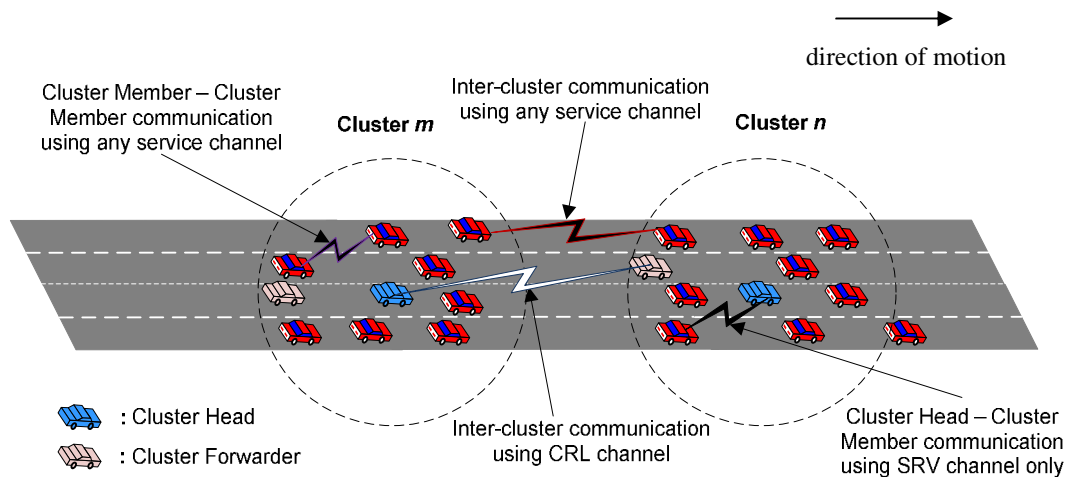


Figure 3.6 inter/intra-cluster communication links

3.2.2 Cluster Formation and Cluster Members Functionality

Once the cluster is formed, the cluster-head is elected, the cluster-head utilizes SRV channel and takes over the responsibility of all inter/intra-cluster management. The cluster-head takes the responsibility of accomplishing the following tasks:

1. Assigning time slots to all cluster members.
2. Processing and disseminating all received safety messages and advertisements.
3. Electing the Cluster Forwarder (CF) node.

The Cluster Forwarder is a cluster member that will be assigned the task of Forwarding all safety messages and advertisements backward to the nearby clusters via the CRL channel.

3.2.3 The Cluster System Cycle

The proposed protocol assumes a single system cycle that is shared between the SRV channel, the remaining service channels, and the CRL channel. As shown in Figure 3.7, the SRV channel consists of Cluster Members Period (CMP) and Cluster-head Period (CHP). CMP is divided into time slots. Each time slot can be owned by only one cluster member. The end of the CHP period is followed by the CBP period during which CRL is used by only CF and CH.

At the beginning of each cycle, all vehicles switch to SRV channel. Each system cycle starts with a frame sent by the cluster-head called the Start Frame (SF). This frame specifies the number of time slots before the SBP of the next cycle. All cluster members receive the frame and become synchronized with the cluster-head. During the CMP

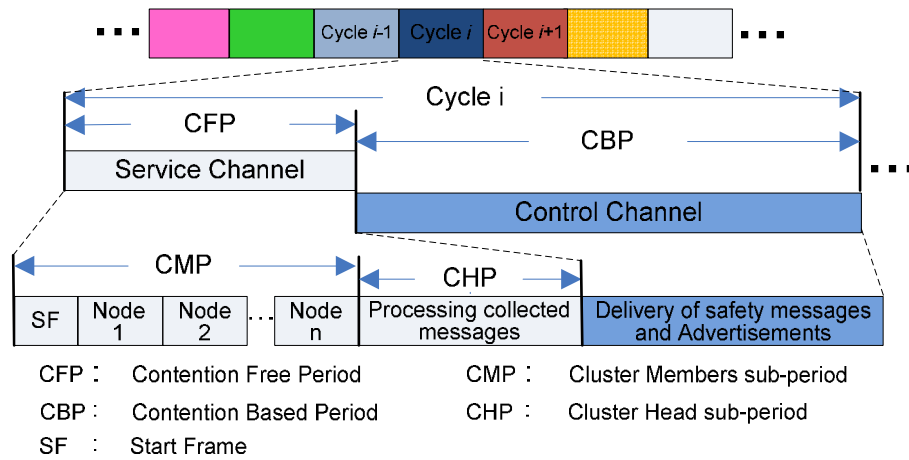


Figure 3.7 cluster system cycle

period each cluster member uses its time slot to send its status, safety messages and advertisement.

The *CHP* period follows the *CMP* period and is allocated to the cluster-head to process all collected messages. During the *CHP* period, the cluster-head processes the received messages and responds to all cluster members' requests. Vehicles remain listening to the SRV channel until the end of the SBP period. After that, they have the option to stay on the same channel, or switch to any other channel. By default, vehicles switch to the CRL channel. The cluster-head and the cluster forwarder must jump to the CRL channel at the beginning of the CBP period. During this period, the cluster forwarder competes for the media to send messages, while the cluster-head keeps receiving safety messages from neighboring clusters. Note that, concurrently during CBP cluster members can exchange data with one another and also with neighboring clusters via service channels, which have been dynamically scheduled by the cluster-head to specific cluster members during the CHP period.

3.2.4 Delay Analysis

The delay parameter is very crucial for the delivery of safety messages. All safety messages generated by cluster members are propagated to their destination via three steps as follows:

- Message transmission via the SRV channel within the cluster.
- Message delivery to neighboring clusters via the CRL channel.
- Message dissemination in the receiving cluster via the SRV channel.

The delay of the safety message, while transmitted on the SRV channel, is deterministic and subject to the upper bound of the SBP period. The length of the SBP, t_{SBP} , can be expressed by:

$$t_{SBP} = M\tau_s + t_{CHP} \quad (3.16)$$

Whereas τ_s is the time slot reserved for each cluster member (depending on the data transfer rate and the size of the safety message), t_{CHP} is the time needed by the *CH* to process the collected messages, and M can be defined using the following equation:

$$M = \frac{2rL}{G+Vl} \quad (3.17)$$

Vl is the average length of the vehicle; G is the average gap between two consecutive vehicles; l is the number of lanes per road, and r is the radius of the cluster. But, the delay of the safety message is nondeterministic while it's on the CRL channel, because of the IEEE 802.11 Distributed Coordination Function (DCF) [39] that depends on the Contention-Based method to get access to the media. So, in order to study the impact of the competition based method on safety message delay, we need to take into account the

following types of nodes contribute to the delay of the safety message, while on the CRL channel:

- The CF and CH nodes belonging to different nearby clusters.
- The nearby individual nodes in a non-clustered state.

If we denote the maximum tolerable delay of a safety message by S_{safety}^{max} , and the length of the system cycle period by T , then, in order to deliver safety messages on time, the following condition must be satisfied:

$$T < S_{safety}^{max} \quad (3.18)$$

The time at which a cluster member generates its safety message is very important to determine the maximum delivery time to notify cluster nodes, and to notify neighboring cluster nodes. As shown in Figure 3.8, the cluster member might generate the safety message either during the SBP sub-period or during the CBP sub-period. But it can only send it during the SBP period.

- If a safety message is generated and sent by any cluster member during *CMP* sub-period, all cluster members and neighboring cluster members must be notified on time. Therefore,
 - The max delay, denoted by $d_{cluster}^{max}$, to notify cluster members is: $d_{cluster}^{max} \leq t_{SBP}$.
 - The max delay, denoted by $d_{neighbor}^{max}$, to notify neighboring cluster members is: $d_{neighbor}^{max} \leq T + t_{SBP}$.
- If the vehicle generates safety message during the CBP sub-period, it can send it only during SBP of the next cycle. Therefore, the maximum delay to inform all cluster

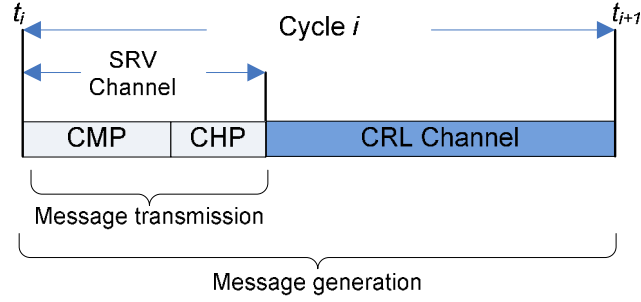


Figure 3.8 message generation and transmission during a single cycle

members and neighboring cluster members after the safety message generation is:

- The max delay to notify cluster members is: $d_{cluster}^{max} < T$.
- The max delay to notify neighboring cluster members is: $d_{neighbor}^{max} < 2T$.

Assuming that the cluster forwarder is able to send the safety message over the CRL channel at least once every cycle. Note that, as mentioned earlier, if some vehicles in the receiving cluster listen to the control channel while the CF node is sending safety messages, they can receive safety messages within a time that is less than $d_{neighbor}^{max}$, or even is less than T .

Before discussing the impact of IEEE 802.11 Contention-Based technique [39] on the delivery of the safety messages, we need to set T based on Equation (6). Since, the value of T depends on the maximum tolerable time of S_{safety}^{max} , we have to define this time first. Therefore, we refer to [58] where the authors demonstrated four types of Vehicular Safety Communication (VSC) applications -Stop/Slow Vehicles Ahead (SVA) Advisor, Emergency Electronic Brake Light (EEBL) Advisor, Forward Collision warning

Table I: Application range and tolerance time window for different VSC applications

VSC Applications	Application Range	Tolerance Time Window
<i>SVA</i>	300m	0.5 – 3.0 sec
<i>EEBL</i>	250m	0.3 – 2.0 sec
<i>FCW</i>	150m	0.3 – 1.0sec
<i>LCA</i>	100m	0.3 – 2.0 sec

(FCW), and Lane Change (& Blind Spot) Advisor (LCA). The application range for the safety messages and the time to receive these messages is shown in Table 1. Based on these results we set $S_{safety}^{max} = 300$ msec., and the length of $T = 90$ msec.

3.2.5 The Impact of IEEE 802.11 Distributed Coordinated Function (DCF) on the Delay of the Safety Message

The IEEE 802.11 DCF is employed to transmit messages across neighboring clusters. When the CF node has a packet to transmit, it senses the channel at the beginning of the CBP. If the channel is sensed idle for a duration called Distributed Inter-Frame Space (DIFS), the node waits for a random period of time called Back-off interval. If the channel remains idle, the node transmits its packet with probability one when the back-off counter reaches zero. If the channel is busy, the node freezes its back-off counter. In addition, the random back-off interval range is doubled after subsequent failed transmission attempt according to Binary Exponential Back-off (BEB) [39].

Many studies have been published analyzing the performance of the IEEE 802.11 DCF and the impact of this method on the important network metrics like throughput, delay, and fairness. In this dissertation, we use simulation to study the impact of the IEEE802.11 on the delivery of the safety messages transmitted over the CRL channel

among neighboring nodes.

3.3 A Novel Algorithm to Form Stable Clusters in Vehicular Ad hoc Networks

An efficient cluster formation algorithm is proposed for VANET environment with the aim of enhancing the stability of the network topology. This technique takes the speed difference as a parameter to create relatively stable cluster structure. A new multi-metric algorithm for cluster-head election is also proposed. The cluster formation algorithm runs in three phases, the cluster initiation followed by the cluster-head determination phase, and finally, the cluster finalizing phase. A suitability function is used by each node to determine its eligibility to become a cluster-head.

3.3.1 System Overview and Assumptions

The degree of the speed difference among neighboring vehicles is the key criterion for constructing relatively stable clustering structure. Neighboring vehicles cooperate with each other to form clusters. In general, vehicles build their neighborhood relationship using the position data embedded in the periodic messages. Usually, vehicles broadcast their current state to all other nodes within their transmission range r . Therefore, two vehicles are considered *r-neighbors* if the distance between them is less than or equal r .

Clusters are formed by vehicles traveling in the same direction (one way). Therefore, all *r*-neighboring nodes used in our analysis are limited to those vehicles traveling in the same direction. However, the speed levels among the *r-neighbors* vary and this variation might be very high; thus, not all *r-neighbors* are suitable to be included in one cluster,

and therefore, they are not good Candidate Cluster Members (CCM). In order to build relatively stable clustering structure, vehicles should consider only *r-neighbors* that are good CCM. Therefore, in this work, vehicles are required to classify their *r-neighbors* into Stable Neighbors (SN) and Non-Stable Neighbors. Two vehicles are considered *r-stable neighbors* if their relative speed is less than some predefined threshold, $\pm\Delta v_{th}$. Hence, only stable neighbors of the vehicle initiating the cluster formation request participate in the cluster formation process.

To show how the degree of the speed difference is used in our technique, we first introduce the statistical distributions of the vehicles' velocity. According to [59] [60] [61], the velocity can be modeled using the normal distribution with mean, μ , and variance, σ^2 , and its probability density function (*pdf*) is given by:

$$p_v(v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(v-\mu)^2}{2\sigma^2}} \quad (3.19)$$

The speed difference, Δv , between a vehicle and its *r-neighbors* follows normal distribution with *pdf* given as:

$$p_{\Delta v}(\Delta v) = \frac{1}{\sigma_{\Delta v}\sqrt{2\pi}} e^{-\frac{(\Delta v - \mu_{\Delta v})^2}{2\sigma_{\Delta v}^2}} \quad (3.20)$$

Where $\Delta v = v_1 - v_2$, $\mu_{\Delta v} = \mu_1 - \mu_2$, and $\sigma_{\Delta v}^2 = \sigma_1^2 + \sigma_2^2$. The probability that the speed difference between two *r-neighbors* falls within the threshold Δv can be obtained by:

$$p_{\Delta v}(-\Delta v_{th} < \Delta v < \Delta v_{th}) = \frac{1}{\sigma_{\Delta v}\sqrt{2\pi}} \int_{-\Delta v_{th}}^{\Delta v_{th}} e^{-\frac{(\Delta v - \mu_{\Delta v})^2}{2\sigma_{\Delta v}^2}} . d\Delta v \quad (3.21)$$

Note that, in (3.21), for a given Δv_{th} , the $p_{\Delta v}$ value decreases as $\sigma_{\Delta v}$ increases. Thus, the expected number of stable neighbors (SN) will vary. So, in order to avoid having high variation of this number, the threshold can be set as a function of the standard deviation, e.g., $\Delta v_{th} = \beta\sigma$. Thus, the threshold is a dynamic parameter which depends on the speed characteristics of the vehicles within the vicinity.

The stable neighbors of a given vehicle might not be stable with respect to each others; thus they can't belong to the same cluster. Therefore, in order to partition the network into minimum number of clusters, such that all cluster members are stable with respect to each other (fast moving vehicles in one cluster and slower moving vehicles in another cluster), not all vehicles are allowed to initiate the cluster formation process even though each vehicle can determine its stable neighbors. In the following section, we discuss which vehicle is a preferable one to initiate the clustering process.

3.3.2 Clustering Process and Protocol Structure

The Inter-Vehicle Communication (IVC) operates in the 5.9 GHz band to support safety and non-safety applications. The Dedicated Short Range Communications (DSRC) uses 75 MHz bandwidth (5.850-5.925 GHz) which is divided into 7 channels. One of the channels is called the control channel, and the remaining six are called service channels [8]. Vehicles are assumed to utilize the control channel to exchange periodic messages and gather information about their neighborhood, and use one service channel to define the cluster radius and perform all intra-cluster communication tasks. According to the DSRC specifications [8], the data link layer can provide a transmission range of up to 1000 meters for a channel. VANET applications can use a longer range (R) for the control channel so that a cluster-head can communicate with neighboring cluster heads

for safety message disseminations, and a shorter range (r) for a service channel that is used for intra-cluster managements. Using the control channel, vehicles can gather status information of other neighboring vehicles and then can build a complete picture about their neighbors which can even go beyond the cluster boundaries.

Since in our technique, slower vehicles will be in one cluster and faster vehicles will be in a different cluster, we can start the cluster formation process either from the slowest or fastest vehicle. For example, if we start with the slowest vehicle, then all the neighboring vehicles of this slowest vehicle that satisfy the speed threshold will be in the first cluster. The remaining vehicles will then go through the same cluster formation process to create other clusters. By extracting the velocity data embedded in the periodic messages, any vehicle can determine whether it has the *slowest velocity* among all its neighbors within R communications range. The slowest vehicle, in our method, is supposed to initiate the cluster formation process by sending a cluster formation request and *only* its stable neighbors participate in this process. The neighboring vehicles whose relative velocity, with respect to the slowest vehicle, is greater than the threshold, Δv_{th} , will not be grouped in the same cluster.

3.3.3 Neighborhood Relationship

The neighborhood term is directly associated with the transmission zone of the node. But, the DSRC is a multi-channel interface with different transmission ranges. Therefore, the neighborhood term needs to be re-defined according to the channel being used for the communications. To illustrate this, consider Figure 3.9, in which three vehicles l, m and n

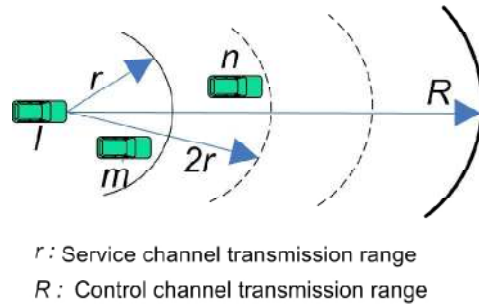


Figure 3.9 neighborhood relationship of a given vehicle

are located within geographical area. For node l , node n is considered a neighbor from the perspective view of the control channel, but not a neighbor from the perspective view of the service channel because the distance l to from n is greater than r which is the maximum range of the service channel. Node m is considered a neighbor from the perspective view of both service and control channels. As nodes exchange their status information via the control channel, it would be easy for node l to identify that node n is within $2r$ distance. Although neighborhood is built using the control channel, it will be represented using r -neighbors terminology. For example, node n is called a $2r$ -neighbor because it's within $2r$ distance.

3.3.4 Cluster-Head Election Parameters

The mobility information (*velocity, location, node degree, and direction*) of the nodes is exchanged via the control channel whose coverage area, R , is larger than that of the service channel, r , used to define the cluster boundary (radius). The mobility information of the *$2r$ -stable neighbors* is needed for the vehicle to initiate the cluster formation request, while cluster-head election information for any node is limited to the nodes that are within r distance from the node itself.

The priority of a node to become a cluster-head is determined by its suitability value that is computed based on the mobility information of its neighborhood. We denote the suitability value of the node by u , the speed by v , the position by p , and the stable nodal degree (the number of stable r -neighbors) by d . Thus, the suitability $u = f(d, v, p)$ is a function defined according to the following criteria:

- The suitability value of the vehicle is calculated by considering the mobility information of its stable neighbors only.
- Nodes having higher number of stable neighbors, maintaining closer distances to their stable neighbors, and having closer speed to the average speed of their stable neighbors should have higher suitability value, thus they are more qualified to be elected as cluster heads.

To calculate the suitability value, each vehicle has to find how close its position is to the mean position of all its d stable neighbors. The vehicle also determines how close its velocity is to the mean velocity of all its d stable neighbors. Since the distance of the vehicle to the mean position of its d stable neighbors can have large values, it's necessary to use the normalization technique to avoid having this parameter dominate the results of the calculation. The normalized mean distance, p_{norm} , of a node to its d stable neighbors can be found by having each node calculate the mean position, μ_p , and the standard deviation, σ_p , of all its d stable neighbors, thus, the p_{norm} can be calculated by:

$$p_{norm} = \frac{n_i^{pos} - \mu_p}{\sigma_p} \quad (3.22)$$

where n_i^{pos} is the position of the vehicle. The smaller the p_{norm} value, the closer the position of the vehicle to the mean position of its stable neighbors. The normalized mean speed v_{norm} can be calculated using the same way. The smaller the v_{norm} value, the closer the speed of the vehicle is to the mean speed of its neighbors. Finally, the suitability value, u , can be calculated as follows:

$$u = de^{-\alpha w} \quad (3.23)$$

Where $w = |p_{norm}| + |v_{norm}|$ and $0 < \alpha \leq 1$ indicates the sensitivity of u to w , the higher u value the more qualified the node is to become a cluster-head. Figure 3.10 shows the impact of the mobility parameters on the suitability. The Figure shows that the suitability of the node to win the cluster-head role decreases as the distance and the speed to d neighbors deviates very large from the mean.

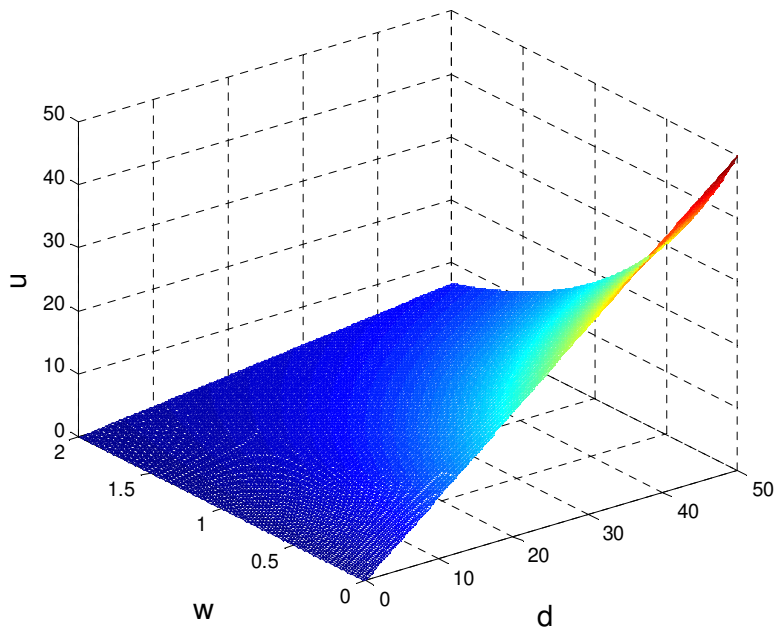


Figure 3.10 suitability value, $\alpha=1$

3.3.5 Cluster Formation Algorithm

In order to execute the algorithm, each vehicle is assumed to maintain and update a set of stable neighbors $SN(t)$ at time t , which contains the IDs of all $2r$ -stable neighbors. IDs are classified into two subsets: The $\Gamma(t)$ and the $\Lambda(t)$, which contain the IDs of the $2r$ -stable neighbors whose velocity is greater than and less than the velocity of the current vehicle respectively. At any time, there should be a vehicle whose speed is the slowest among its $2r$ -stable neighbors, and as a result, the $\Lambda(t)$ list maintained by this vehicle is empty. The pseudo codes of the algorithms (Algorithms 3-5) are shown in Figures 3.11-3.13. The algorithm basically requires that the slowest vehicle or the vehicle whose $\Lambda(t)$ members belong to other clusters originates the cluster formation process. This vehicle is called the Cluster Originating vehicle (COV). Line 3 in Algorithms 3, shows that COV sends the *InitiateCluster(CIDtmp)* with its ID as a temporary cluster ID to all $\Gamma(t)$. Then, as shown in Algorithms 4, all $\Gamma(t)$ non-clustered members react upon receiving this message by setting their cluster ID temporarily to be the ID of the COV as shown in line 3. Vehicles start calculating their suitability to become a CH as shown in line 4. Then, the vehicle calculates the waiting time, T_{wait} , before announcing its eligibility to become a cluster-head as shown in line 5. The vehicle waits for T_{wait} that is proportional to the suitability value of the vehicle. The higher the suitability value, the

Algorithm 3 Initiating Clustering Process

```

1: if  $\Lambda(t)$  is empty || ( $\Lambda(t)$  members  $\in$  other clusters) then
2:    $CID_{tmp} \leftarrow v_i.id$ 
3:   send InitiateCluster(CIDtmp)
4: end if

```

Figure 3.11 clustering initiation process

Algorithm 4 CH Competition and Determination

```

1: if  $v_j \in \Gamma(t)$  then
2:   On Receiving InitiateCluster( $CID_{tmp}$ )
3:    $v_j.CID \leftarrow CID_{tmp}$ 
4:    $v_j.Suitability()$ 
5:    $v_j.T_{wait} \leftarrow v_j.DeferTime()$ 
6:   while  $v_j.T_{wait} > 0$  do
7:     if FormCluster( $CH_{id}$ )  $\in \Gamma(t)$  then
8:       if received  $CH_{id} \in \Gamma(t)$  then
9:         QuitCompetition()
10:        Process FormCluster( $CH_{id}$ )
11:       end if
12:     else
13:       Decrement ( $v_j.T_{wait}$ )
14:     end if
15:   end while
16:    $v_j.STATUS \leftarrow CH$ 
17:    $CH_{id} \leftarrow v_j.id$ 
18:    $v_j.CID \leftarrow CH_{id}$ 
19:   Send FormCluster( $CH_{id}$ )
20: end if

```

Figure 3.12 cluster-head determination process

less the waiting time value. This can be seen in lines 6 through 15. If the vehicle receives a *FormCluster*(CH_{id}) message from any other vehicle belongs to $\Gamma(t)$ before its waiting time, T_{wait} , expires, then the vehicle determines that there are other vehicles belong to $\Gamma(t)$ that are more suitable to win the CH role. Therefore, the vehicle quits the competition and processes the received message. This is shown in lines 7 through 11. If the waiting time of the vehicle expires before any other vehicle sends the *FormCluster*(CH_{id}) message, then the current vehicle wins the cluster-head competition, changes its state to a cluster-head, and sets the cluster ID to be its own ID. This is shown in lines 16 through 18. Finally, the vehicle sends the *FormCluster*(CH_{id}) message with

Algorithm 5 Cluster Formation Finalization

```

1: if  $v_k \in \Gamma(t)$  then
2:   On Receiving  $FormCluster(CH_{id})$ 
3:   if  $v_k \in (\Gamma(t) \cap v_j.SN)$  then
4:      $v_k.STATUS \leftarrow CM$ 
5:      $v_k.CID \leftarrow CH_{id}$ 
6:   else
7:      $v_j.CID \leftarrow default$ 
8:     Reconstruct  $\Gamma(t)$ 
9:   end if
10: end if

```

Figure 3.13 finalizing clustering process

its own ID as the new cluster ID as shown in line 19.

Algorithm 5 shows the final stage of the clustering process. All vehicles in the $\Gamma(t)$ of the COV receive the $FormCluster(CH_{id})$ as shown in line 2. But, only *r-stable neighbors* of the winner (since the cluster boundary is defined by *r*), which belong to the $\Gamma(t)$ of the COV change their state to a Cluster-Member (CM) and change their temporary cluster ID to be the new cluster ID embedded in the received $FormCluster(CH_{id})$ as shown in lines 4 and 5. After that, the vehicle becomes a cluster member of the corresponding cluster. Vehicles that belong to $\Gamma(t)$ of the COV and couldn't associate with the cluster being formed, set their temporary cluster ID to the default (their own ID), modify their $\Gamma(t)$ and start the cluster formation process again, this is shown in lines 7 through 8.

According to the proposed algorithm, vehicles wait for a period of time before accessing the media to announce their eligibility to be a cluster-head. Media access is controlled by the Distributed Coordination Function (DCF) on the Media Access Control

(MAC) layer [39]. Usually, vehicles use the minimum Contention Window (CW_{min}) size value before accessing the media, and they double this size for each unsuccessful transmission until they reach the max Contention Window size (CW_{max}). In this work, vehicles wait for a period of time that is proportional to their suitability value before announcing their suitability to be a cluster-head as follows:

$$T_{wait} = \left\lfloor \frac{N_{max} - u}{N_{max}} * (CW_{max} - CW_{min}) + CW_{min} \right\rfloor \quad (3.24)$$

whereas N_{max} is the total number of vehicles in $\Gamma(t)$, u is the suitability value of the vehicle, and CW_{max} and CW_{min} are the maximum and the minimum contention window sizes respectively [39]. When there is more than one vehicle having the same T_{wait} , they will send the *FormCluster(CHid)* to announce their eligibility to become a CH at the same time. As a result, a collision occurs and none of them wins the competition. In this case, only those collided vehicles start new iterations of competition until one of them wins or the maximum number of iterations is completed. The length of the T_{wait} in iteration i is calculated as follows:

$$T_{wait} = \left\lfloor 10^i \left(\frac{N_{max} - u}{N_{max}} \right) - \left\lfloor 10^i \left(\frac{N_{max} - u}{N_{max}} \right) \right\rfloor * (CW_{max} - CW_{min}) + CW_{min} \right\rfloor \quad (3.25)$$

If the maximum number of iterations is used and nodes still collide, then each node picks a uniformly distributed random number between 0 - 9 and the one with the smaller value wins the competition. If the random numbers are the same, then the nodes will generate another pair and so on. Let s be the probability that a node will be able to announce its eligibility first time it generates a random number. The probability that a node will be able to announce its eligibility during the second time given the fact that it

failed to announce during the time is $(1 - s)s$. Similarly, the probability that a node will be successful during the third time given the fact that it failed during the first and second times is $(1 - s)(1 - s)s$ and so on. The node has to generate random numbers $\frac{1}{s}$ times before it can announce its eligibility. Therefore, if the node went through i iterations using equation (3.25) before it started generating random numbers, the average number of trials for eligibility announcement is $i + \frac{1}{s}$.

3.3.6 Analysis of Cluster-Head Election

During the cluster formation process, vehicles compete to win the cluster-head role. To find the average number of nodes that a vehicle (within the $2r$ neighbors) competes with during the cluster-head election, we first need to find the average number of the stable neighbors of the COV node within $2r$ communication range. So, if the COV node has N neighbors, then the probability that the COV node has K stable neighbors out of N follows the binomial distribution and can be calculated using:

$$P_{SN}(K) = \binom{N}{K} P_{\Delta v}^K (1 - P_{\Delta v})^{N-K} \quad (3.26)$$

where $P_{\Delta v}$ can be found using (9). Now, assume that vehicle i is one of the K nodes and let s be the average number of the r -stable neighbors of vehicle i . Let $P_{s,x}(s)$ be the probability that a vehicle that is x units (usually meters) away from the COV has s r -stable neighbors out of K . To calculate $P_{s,x}(s)$, we analyze it with a simplified assumptions by considering the part of the road, where all $\Gamma(t)$ of the COV are found, as a one dimensional problem as shown in Figure 3.14 (a). This simplified assumption is true

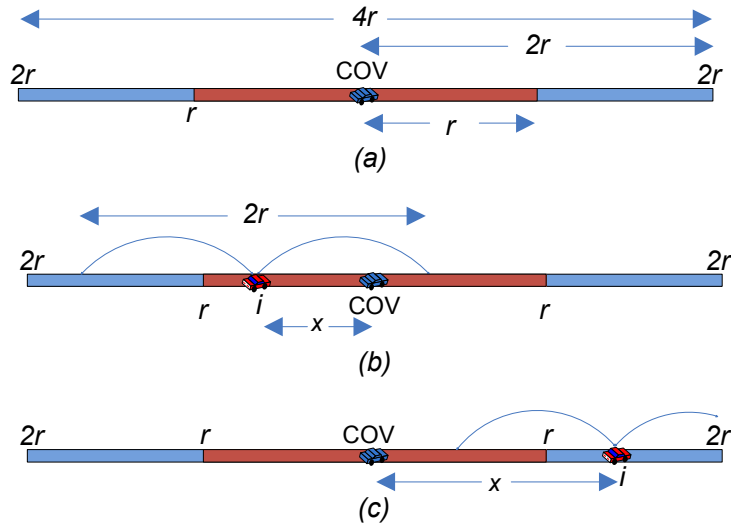


Figure 3.14 vehicles' location with respect to the COV node

since the roadway width is very small compared to the transmission range and thus it can be neglected. Figure 3.14 (a) shows the part of the road as a one dimensional line. This line represents the area covered by the $2r$ transmission range of the COV node. As shown in the figure, the COV node is located at the center of the line that is $4r$ long. Here, we are concerned about the number of the r -stable neighbors of any vehicle that can be placed anywhere on this line. If we randomly select vehicle i on this line that is x units away from COV, and randomly select another vehicle j on this line and try to find the probability that vehicle j is within r distance from vehicle i (the probability that both vehicles are r -neighbors). Then, depending on where the selected vehicles are located with respect to the center of this line (COV), we have to deal with only two cases. 1) The first case is when $x \leq r$ as shown in Figure 3.14 (b), in this case, vehicle i is within r distance from the COV node (the center), thus, the probability that vehicle j is a r -

neighbor of vehicle i is $1/2$. 2) The second case is when $r < x \leq 2r$ as shown in Figure 3.14 (c), in this case, the probability that vehicle j is r -neighbor of vehicle i is $\frac{3r-x}{4r}$. To generalize, we write the probability, q , that two stable nodes in the $2r$ transmission range of the COV are neighbors as:

$$q = \begin{cases} \frac{1}{2}, & \text{if } x \leq r \\ \frac{3r-x}{4}, & \text{if } r < x \leq 2r \end{cases} \quad (3.27)$$

Now, we can calculate $P_{s,x}(s)$ as follows:

$$P_{s,x}(s) = \binom{K-1}{s} q^s (1-q)^{K-s-1} \quad (3.28)$$

The Probability Distribution Function (PDF) is:

$$P_S(s < S) = \sum_{s=1}^{K-1} \binom{K-1}{s} q^s (1-q)^{K-s-1} \quad (3.29)$$

The expected value, $E(S)$, is:

$$E(S) = \sum_{s=1}^{K-1} s \binom{K-1}{s} q^s (1-q)^{K-s-1} \quad (3.30)$$

Figure 3.15 shows the PDF of the vehicles that are x units away from the COV node. The transmission range r is set to 200 units and 800 stable neighbors of the COV are uniformly distributed in the $4r$ radius. From the figure, it's obvious that vehicles that are closer to the COV have higher number of s r -stable neighbors out of the total stable neighbors of the COV.

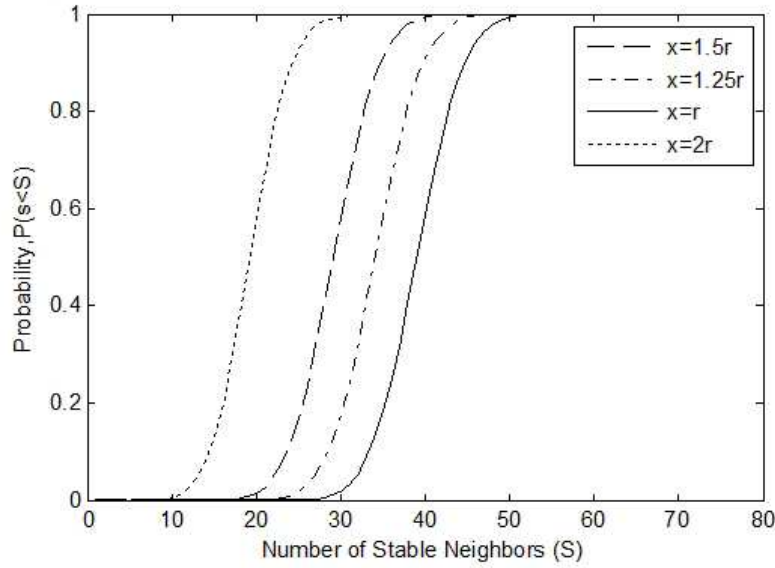


Figure 3.15 probability density function (PDF)

3.3.7 Cluster Maintenance

Due to the high dynamic nature of the VANET, vehicles keep joining and leaving clusters frequently, thus, causing extra maintenance overhead. The events that trigger the maintenance procedure can be summarized as follows:

- *Joining a cluster:* when a standalone (non-clustered) vehicle comes within r distance from a nearby cluster-head, the cluster-head and the vehicle check whether their relative speeds is within the threshold $\pm\Delta v_{th}$. If the speed difference is within $\pm\Delta v_{th}$, then the cluster-head will accept the vehicle and will add it to the cluster members list. If there are more than one cluster-heads in the vicinity that can be joined, the vehicle calculates the period of time, called the Residual Time (RT), it will remain in the transmission range r of these cluster-heads. The vehicle joins the cluster-head where it will stay for

the longest period of time. The RT could be computed from the information about the relative speed, current location, and the transmission range r as follows:

- If the standalone vehicle is following the cluster-head and its velocity at time t is less than that of the cluster-head, then

$$RT(t) = \frac{r - dis(n, CH)}{\Delta v}$$

where Δv is the speed difference, and $dis(n, CH)$ is the distance between the standalone vehicle, n , and the cluster-head, CH. The above formula can also be used when the standalone vehicle is followed by the cluster-head but its velocity is greater.

- If the standalone vehicle is following the cluster-head and its velocity at time t is greater than that of the cluster-head, then

$$RT(t) = \frac{r + dis(n, CH)}{\Delta v}$$

this formula can also be used when the standalone vehicle is followed by the cluster-head but its velocity is less.

- *Leaving a cluster:* when a cluster member moves out of the cluster radius, it loses the contact with the cluster-head over the service channel, r . As a result, this vehicle is removed from the cluster members list maintained by the cluster-head. The vehicle changes its state to a standalone if there is no nearby

cluster to join or there is no other nearby standalone vehicle to form a new cluster according to our cluster formation algorithm.

- *Cluster merging*: when two cluster-heads come within each other transmission ranges and their relative speed is within the predefined threshold Δv_{th} the cluster merging process takes place. The cluster-head vehicle that has less number of members gives up its cluster-head role and becomes a cluster-member in the new cluster. The other cluster members join that neighboring cluster if they are within the cluster-head's transmission range and the speed is within the threshold. If there is any other nearby clusters, then vehicles calculate their RT and join the cluster where they can stay for the longest period of time. Finally, vehicles that can't merge with the cluster nor can join a nearby cluster, start clustering process to form a new cluster according to our algorithm.

CHAPTER 4: SIMULATION AND PERFORMANCE EVALUATION

In this chapter, we show the performance analysis and the comparison of our proposed methods with other existing techniques.

4.1 Performance Analysis of the Channel Allocation for RSU Based on Virtual Task Finish Time

The performance analysis of our method was evaluated using simulation. The simulation environment and the evaluation criteria are explained in the following sections.

4.1.1 Simulation Environment

We developed a simulator using C++ with graphical interface to evaluate the performance of our method. The simulator is composed of four models, the mobility and the data network models that are simulator specific, and the mobility and the channel prediction models specific to the task virtual finish time algorithm.

A two-lane per direction road was simulated using C++ with graphical interface. In the simulation, the RSU was installed in the middle of a 1 Km road with a maximum transmission range of 250m. Vehicles arrive at the RSU region according to the Poisson process. Vehicles move on both directions of the road with a maximum speed that can't exceed the speed limit of that particular lane. The speed of the vehicles follows the normal distribution with mean, $\mu = 70Km$, and standard deviation $\sigma = 21Km$. Vehicles can change their current lane if there is a room in the next lane, and if the vehicle can

maintain a safe-distance (1.6-2.2 sec) with the vehicle ahead in the new lane. The safe-distance is also maintained between the lane changing vehicle and the vehicle behind it on the new lane. If the vehicle can't change lane, then it should decelerate and slowdown so that its speed matches the speed of the vehicle in the front. Once the vehicle changes its lane, it will adapt its speed to the average speed and speed limits of the new lane.

For the task virtual finish time algorithm implementation, we used a very simple distance prediction model that calculates the future distance based on the current mobility information. The algorithm uses coasting to predict the future position of the vehicle as follows: $pos^f = pos^c + St$, where pos^f and pos^c are the future and current positions of the vehicle, S is the current speed of the vehicle, and t is the time interval. For channel prediction model (setting the transmission rate and the transmission error probability), we adopted the results of the field tests presented in [55]. We evaluated our proposed method by setting Δ_{bt} to different values. The Δ_{bt} was set to 15msec and 25msec for each Q1 member. Each simulation run last for 600 sec. (only 575 sec. were considered to derive the final results). The results are an average of 10 runs of each scenario.

For comparison purposes, we adopted a method that allocates the air-time transmission plan based on the average transmission rate. For each arrival task, the method basically uses $r_{AvgR} \geq \frac{DR}{T_{dwell}}$ to find the average rate that is considered the minimum requirement guarantee to finish the task. The DR is the remaining data, and T_{dwell} is the estimated dwell time. If the task is admitted, then the algorithm tries to assign a time share to get the actual transmission rate for each admitted task such that $r_{i,n}w_{i,n}c(1 - p_{i,n}) \geq r_{AvgR}$ and $\sum_{i=1}^m w_i \leq 1$ for cycles $n = 1, 2, \dots$

4.1.2 Metrics for evaluation

To evaluate and compare the performance of both methods, we used two metrics: the task failure rate, R_f , and the percent of the effective usage of the cycle, C_{eff} :

- The task failure rate R_f rate can be defined as

$$R_f = \frac{N_f}{N_a} \quad (4.1)$$

where N_f represents the number of failed tasks, and N_a represents the total number of admitted tasks by the RSU.

- The fraction of the effective usage of the cycle C_{eff} represents the percent of the useful time (including the time used to retransmit failed packets of the successful tasks) of the cycle with respect to the length of the cycle. In general, the fraction of the cycle that is used by the failed tasks is considered a waste. Therefore, we use the waste percent per cycle, C_w , to calculate C_{eff} as follows:

$$C_{eff} = 1 - C_w \quad (4.2)$$

The percent of the waste per cycle, C_w , can be defined as the time used by all failed tasks divided by the simulation time. To calculate C_w , we tracked each admitted task during the simulation run, and then summed the portions of time shares of each failed task. Finally, we calculated the waste percent per cycle as follows:

$$C_w = \frac{\sum_{i=1}^{N_f} \sum_{n=1}^E w_{i,n} c}{Tc} \quad (4.3)$$

where N_f is defined in (4.1), $w_{i,n}$ is the time share of task i during cycle c , E is the effective number of cycles the failed task used to receive data from the RSU, T is the total number of cycles in the simulation run, and c is the length of the cycle in seconds (the cycle length is one second).

We first show the average number of admitted tasks per minute for both methods. Figure 4.1 shows that both methods, the AvgR-based and the Virtual Time (VT) based that uses the backup time, have almost the same admission rate for different loads. As shown in the figure, the admission rate of both methods decreases as the size of the load increases.

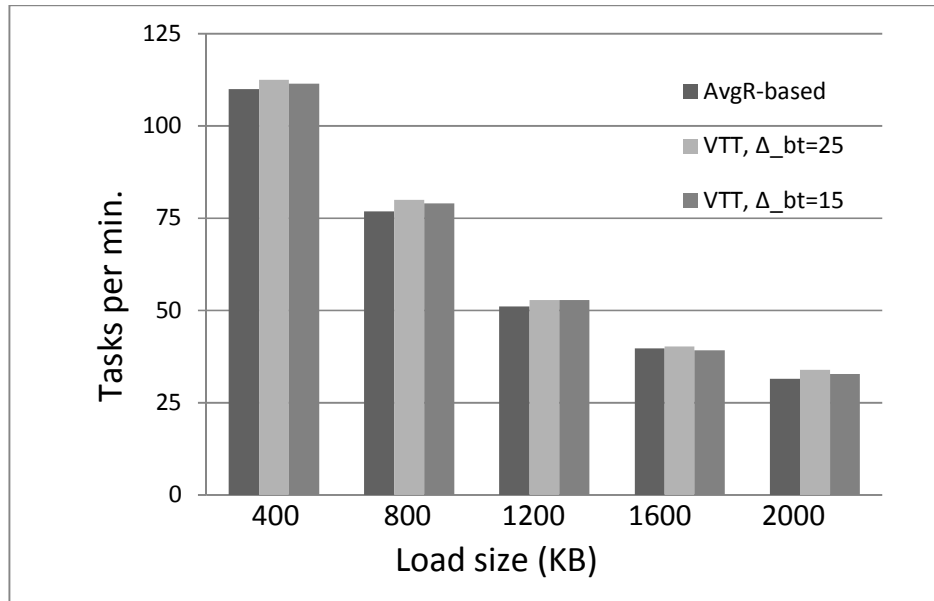


Figure 4.1 tasks per minute

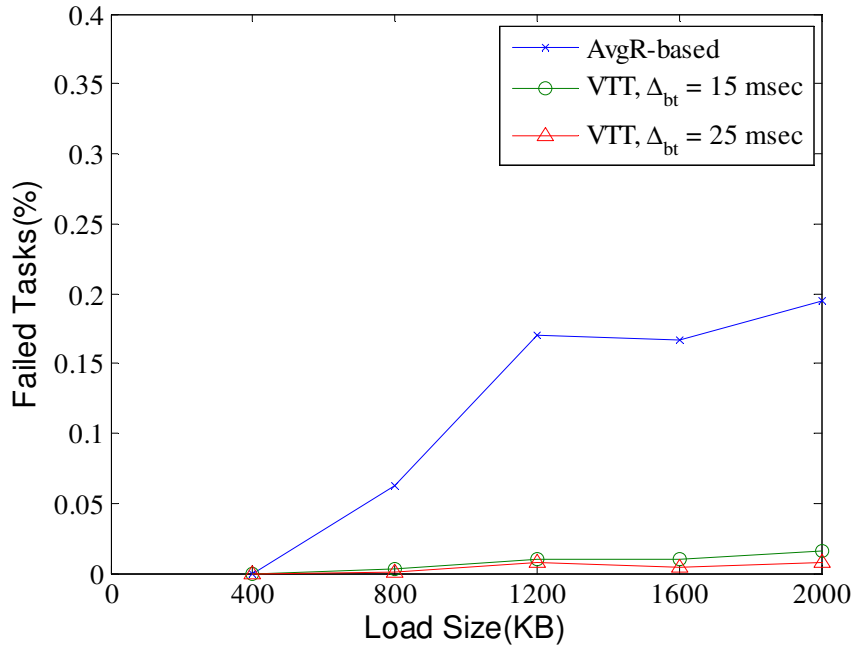


Figure 4.2 failure rate

In figure 4.1 both methods have almost the same number of admitted nodes, but the most important part is whether the admitted task can be finished successfully. Figure 4.2 shows the task failure rate for both methods. As shown in the figure, as the load size increases, the failure rate of the AvgR-based method increases because it doesn't evaluate the risk of the vehicles, and pushes the tasks' finish time to the edges. However, the task failure rate remains very low when the VT-based method is used because it always evaluates and reduces the risk of the vehicles.

Figure 4.3 shows how efficient both methods utilize the resources. The figure shows that our method uses the resources more efficiently than the AvgR-based method. The figure shows that as the load size increases our method outperforms the AvgR-based method. The wastage of resources is higher when using AvgR-based method because it

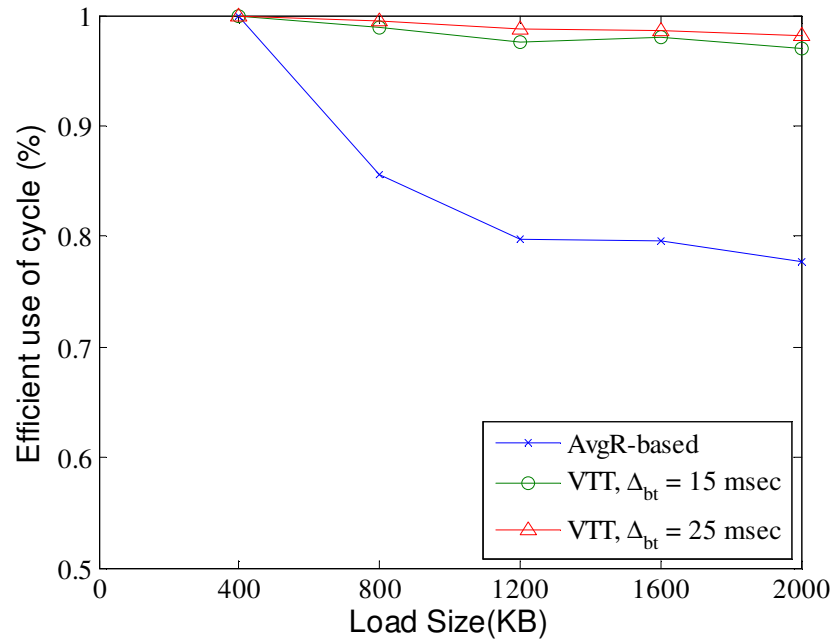


Figure 4.3 effective cycle usage (%)

doesn't allocate resources efficiently, which requires the task to use more resources to complete and if the task fails, then the whole resources allocated to the task is actually a waste.

4.2 Performance Analysis of the Media Access Technique for Cluster-based Vehicular Ad hoc Networks

4.2.1 Simulation Description

The protocol performance was evaluated via simulation using C++ with graphical interface. Vehicles are generated based on the headway distributions among vehicles. The arrival rate of the vehicles was modeled using Poisson distribution. For each generated vehicle, an average speed and acceleration is also generated using normal distribution

with mean μ and standard deviation σ . Vehicles move on the road and if the speed of the leading vehicle is slower than that of the following vehicle, the following vehicle changes the lane if there is a spot in the next lanes, otherwise it reduces its speed to match the speed of the leading vehicle. Vehicles move on the road and form non-overlapping clusters.

To evaluate the proposed protocol, and to compare the performance of this protocol with the classical clustering techniques, we generated different clusters with different densities by varying the mean headway (the time gap between successive vehicle arrivals). We varied the distance between two consecutive clusters and we also varied the average speed of different clusters, so the clusters in the back move faster. Eventually, clusters in the back enter the CRL channel transmission area of the CF node of the cluster in the front. Vehicles keep joining and leaving the clusters as long as they move on the road. Table 4.2 shows different Simulation parameters.

Table II: Simulation parameters

Road, Vehicles and Clusters' parameters	Safety message parameters	IEEE 802.11 parameters
SRV range = 200 m	$S = 200$ bytes	DIFS = 64 us
CRL range = 800 m	$T = 90$ msec	aSlotTime = 16 us
Average vehicle's length = 5 m	$t_{CHP} = 10$ msec	Max. contention window = 31
Number of Lanes = 4	$S_{safety}^{\max} = 300$ msec.	Number of retries = 7

4.2.2 Metrics and Results

Before discussing the performance metrics used to evaluate the proposed protocol and for the convenience, we refer to our protocol as *CF-Based* protocol, because cluster forwarder is used to relay safety messages backward, and we refer to protocols relying on

the cluster-head to send safety messages as *CH-based* protocol. To evaluate the performance of the proposed protocol, the data-related Metrics were considered as follows:

- The earliest notification, which shows how early in time CF-based protocols can forward safety messages compared to the CH-based protocols.
- The delay of the safety messages. This metric shows the impact of the contention-based technique on the delivery of safety messages. In this metric, we show the worst case scenario, and for this purpose, we increased the transmission range of the CRL channel, so more CF nodes compete to access the media. In addition, we force every vehicle to send safety messages during its time slot. At once, all safety messages are collected and sent in one package.

In Figure 4.4, x -axis shows the average speed difference between two consecutive

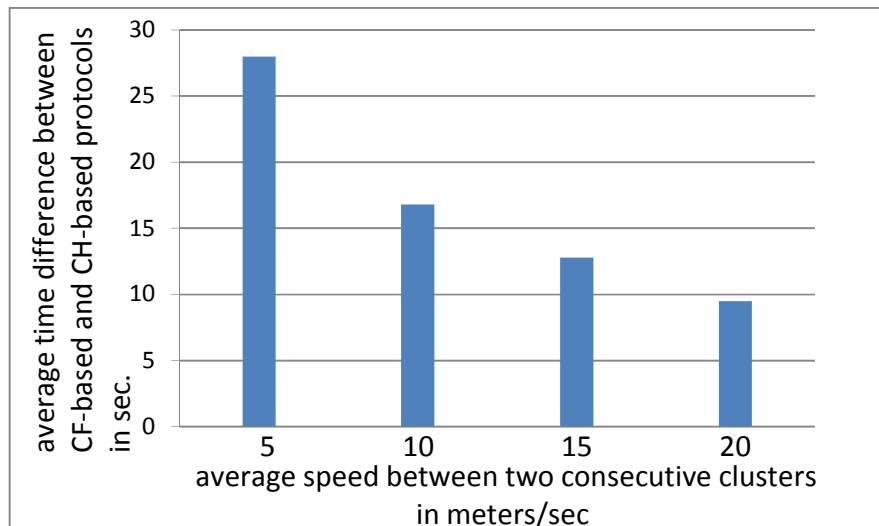


Figure 4.4 earliest notifications

clusters, and the y-axis represents the average time difference for notification between CF-based and CH-based protocols. When the cluster in the back is 5 meters/sec faster than the cluster in the front, CF-based protocol can notify the approaching cluster 28 seconds, on average, earlier than CH-based protocol. Due to the close proximity of the CF node to the approaching cluster, an early notification time is achievable. Therefore, our model performs more efficiently as compared to the CH-based model.

Figure 4.5 shows the delay of safety messages for different data transfer rates. The x-axis is similar to Figure 4.4 and the y-axis represents the delay of safety messages in msec. This figure demonstrates the worst case scenario, where the current CF node competes with three CF nodes from neighboring clusters to access the media. Safety messages collected and simultaneously broadcasted as a single package (without

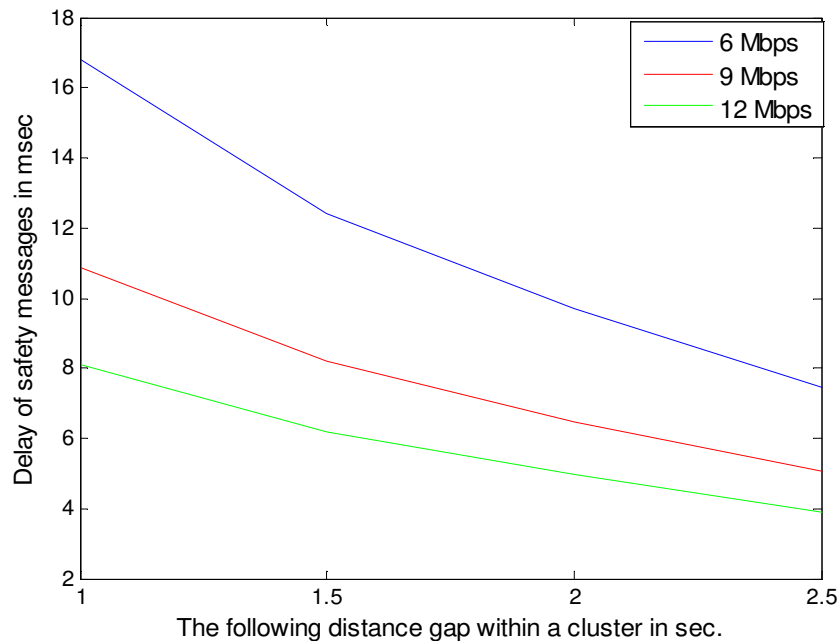


Figure 4.5 average safety messages delay caused by competition-based technique

compression) from all four clusters. From the figure, we note the density and delay are directly proportional. A decrease in cluster density, results in a decrease in the number of messages being sent, which therefore, results in a decrease in the delay.

4.3 Simulation and Performance Evaluation of Cluster Formation

Algorithm

An extensive simulation study was conducted to evaluate the performance of our protocol. The C++ was used to develop the simulation. In our simulation, we consider different road traffic and different network data parameters.

4.3.1 Simulation Setup

The highway traffic model used in this paper was built based on the car following model. The model is used to simulate the behavior of the vehicles on a 5-lane per direction highway. In the simulation, we monitor 400 vehicles on a highway of 15Km length for 650 sec. The arrival rate of the vehicles follows the Poisson process. We simulated three types of vehicles' speed taken from statistical measurements [59-61]. The speed of the vehicles on a given lane can't exceed the maximum speed limit of that lane. The speed assigned to the vehicles follows the normal distribution with average μ and standard deviation σ as shown in Table 3. In our simulation, we considered a major safety requirement that the vehicles should keep a safe-distance with the vehicles ahead. This will give any vehicle the ability to decelerate to avoid collision with the vehicle ahead if it can't change the lane. Another safety requirement is considered when a lane change takes place. Vehicles can change their current lane if there is room in the next lane and if the vehicle, the lane changing vehicle, can maintain a safe-distance with the

vehicle ahead in the new lane. Also the safe-distance is kept between the vehicle, lane changing vehicle, and the vehicle behind it on the new lane. If the vehicle can't change its current lane, then the safe-distance gives the vehicle the ability to decelerate and slowdown so that its speed matches the speed of the vehicle in the front. The density of the vehicles varies between (13 to 21 vehicle/Km/Lane) depending on the speed being used. For all simulation scenarios, the $\Delta v_{th} = \sigma$, e.g., for $\mu = 70\text{Km/h}$ and $\sigma = 21\text{Km/h}$, the $\Delta v_{th} = 21$. The performance of different Δv_{th} values can be found in [62].

We used different network parameters in the simulation. The data rate is set to 6 Mbps and the periodic messages are sent every 100 msec., the size of the message including the mobility information is 100 bytes. DSRC standard supports data rate in the range 6 to 27 Mbps [39]. However, various members of the Vehicle Infrastructure Integration (VII) Consortium use 6 Mbps data rate [63], [64] for road testing. Thus, we also decided to use 6 Mbps data rate. To study the performance of the clustering techniques for different cluster sizes, we used different transmission ranges for r and R . The transmission range for r was varied between 150 and 300 meters, while it's between 800 and 1000 meters for R . For media access, we used the IEEE802.11 standard [39]. We set the $CW_{min} = 15$, $CW_{max} = 1023$, $aSlotTime = 16\mu s$, $SIFS = 32\mu s$, and $DIFS = 64\mu s$.

Table III: The average and the standard deviation of the speed

$\mu(\text{Km/h})$	$\sigma(\text{Km/h})$
70	21
90	27
110	33

4.3.2 Evaluation Criteria

To show the performance of our proposed Threshold-Based (TB) technique, we compare it with the Weight-Based (WB) and the Position-Based (PB) methods proposed in [22] and [49] respectively. Originally, the WB method for MANET was proposed in [65], [66] and revised by Chatterjee, et al. [67] by introducing the combined weight metric. The algorithm assigns node weights based on the suitability of a node being a cluster-head. This algorithm basically takes into consideration the nodal degree, the transmission power, mobility, and battery power of the mobile nodes. Each one of these parameters is assigned a weight; the sum of these weights is 1. Then, the value of each parameter is multiplied by its weight and all the values are finally summed to produce the combined weight. The node with the lower combined weight is more suitable to become a cluster-head. The same algorithm was adopted by VANET clustering techniques [22], but without considering the battery power factor since it is not a crucial problem in VANET. In the simulation, we assigned all WB method parameters equal weights. For the PB method, the priority of the node is calculated based on the eligibility function. A Node having longer travel time has higher eligibility value, and this value decreases as the velocity of the node deviates largely from the average speed. We compare the three methods under the same environment variables. Each simulation run was repeated 10 times with different random seeds and the collected data was averaged over those runs

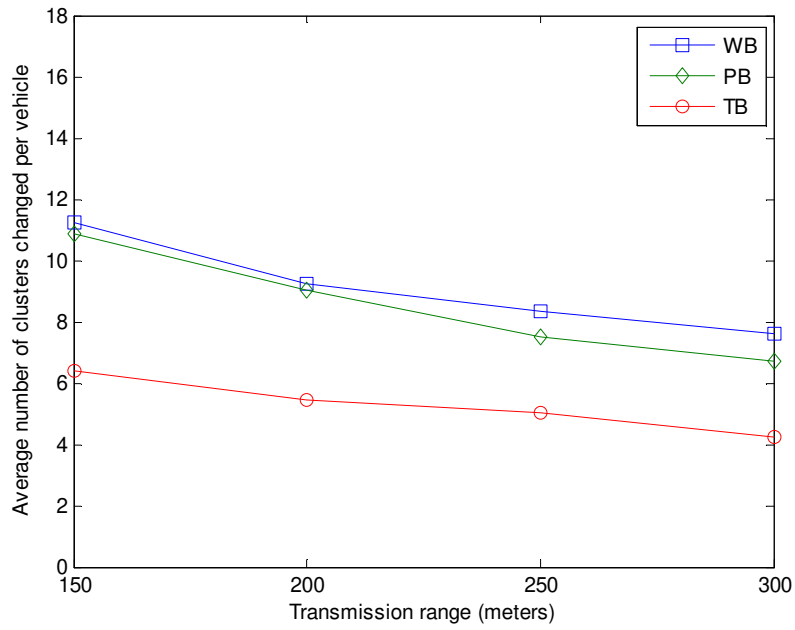
4.3.2.1 Cluster Stability

A clustering structure should be stable with respect to the nodes' motion, i.e., the cluster configuration should not change too much while the topology changes. In a high dynamic VANET, vehicles keep joining and leaving clusters along their travel route, and

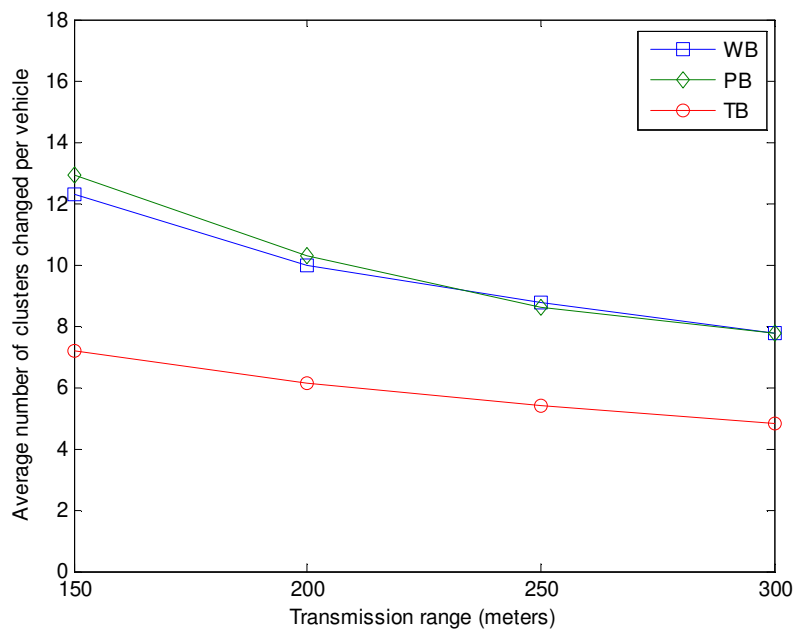
the Number of Cluster Changes (*NCC*) of the vehicle will vary depending on the clustering algorithms being used. Good clustering algorithms should be designed to minimize the number of cluster changes of the vehicle by minimizing vehicle transitions between clusters. The *NCC* of the vehicle during its lifetime can be used to evaluate the cluster stability. To find the *NCC* of the vehicle, we first introduce the basic transition events the vehicle encounters during its lifetime:

- e_1 – A vehicle leaves its cluster and forms a new one.
- e_2 – A vehicle leaves its cluster and joins a nearby cluster.
- e_3 – A cluster head merges with a nearby cluster.

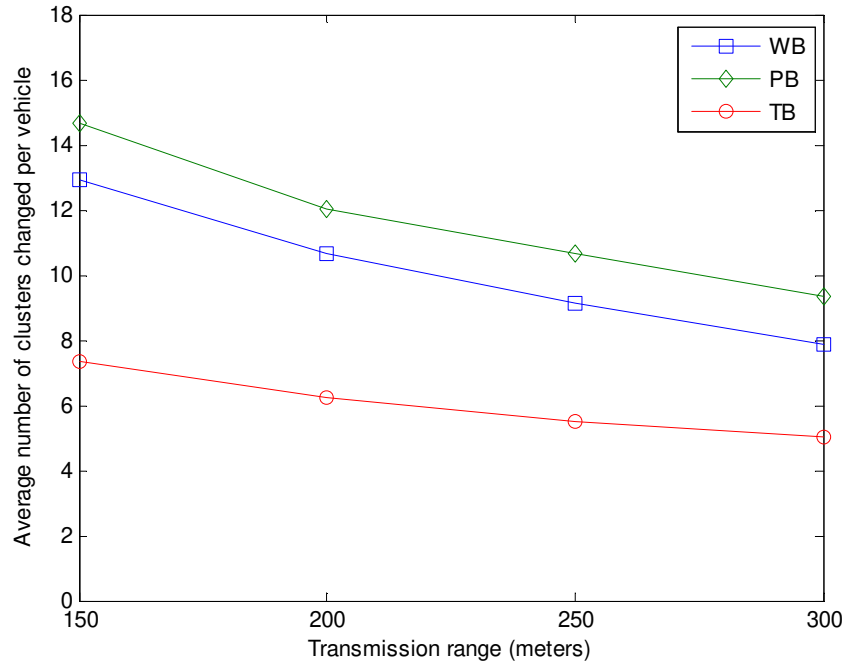
For each vehicle, the sum of all transition events (e_1 , e_2 , and e_3) defines the *NCC* of the vehicles over its lifetime. We compare the average *NCC* of the vehicles for the TB, WB, and PB methods when different speeds and different transmission ranges are used. In Figure 4.6 (a) (b) (c), the x-axis represents the transmission range, while the y-axis represents the average *NCC* of the vehicle. From Figure 4.6 (a) (b) (c), we can see that the average *NCC* produced by our TB technique is smaller compared to that produced by the WB and PB methods. This means our technique causes less number of cluster transitions for all different velocities and different transmission ranges. The figure shows that the average *NCC* of a vehicle is reduced by 34% to 46% compared to the WB and PB methods. We can see that the TB method performs even much better when the average speed becomes higher. Note also that the average speed increase has little impact on the number of clusters changed per vehicle when the TB method is used. This is because the threshold is a function of the speed deviation and it's always proportional to the speed



(a) $\mu=70\text{km/h}$, $\sigma=21\text{km/h}$, $\Delta v_{th} = \sigma$



(b) $\mu=90\text{km/h}$, $\sigma=27\text{km/h}$, $\Delta v_{th} = \sigma$



(c) $\mu=110\text{km/h}$, $\sigma=33\text{km/h}$, $\Delta v_{th} = \sigma$

Figure 4.6 average cluster changes per vehicle

regardless of its average value. The figures show that the average NCC of the vehicle decreases as the transmission range increases. This is because increasing the transmission range r , increases the probability that a vehicle stay connected with its cluster-head. The cluster stability can also influence the signaling overhead. A frequently changing clustering structure results in an increase in maintenance messages and thus increasing the load on nodes. From the figure, we can conclude that the TB method reduces the signaling overhead and the traffic load since it causes less number of transition between the clusters. We can also calculate the *average transition rate*, $\lambda_{tr,mean}$, between clusters as follows:

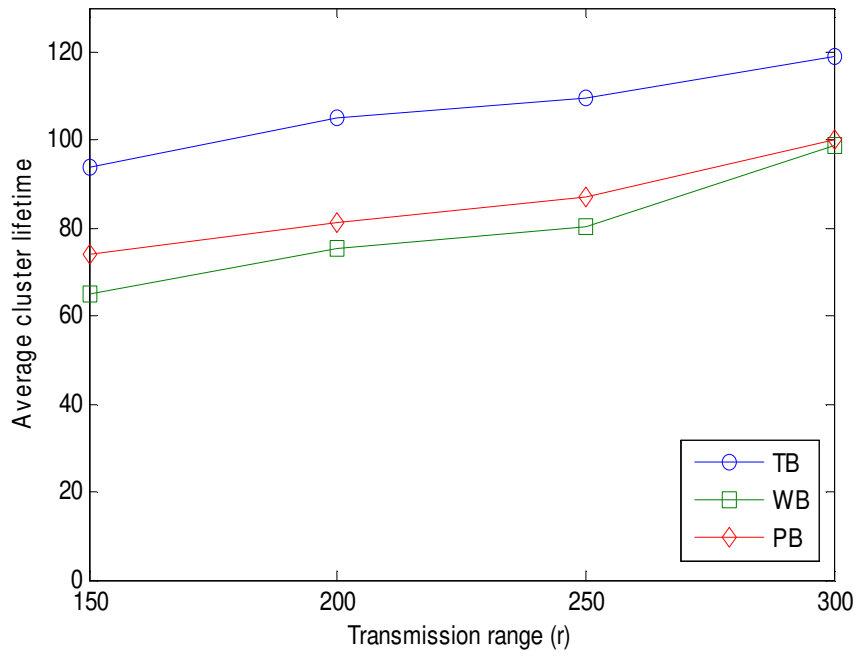
$$\lambda_{tr,mean} = \frac{1}{K} \sum_{i=1}^K \frac{NCC_i}{N_{i,life}} \quad (4.4)$$

Where, $N_{i,life}$ is the lifetime of vehicle i , NCC_i is the number of clusters vehicle i changes during its lifetime, and K is the total number of vehicles.

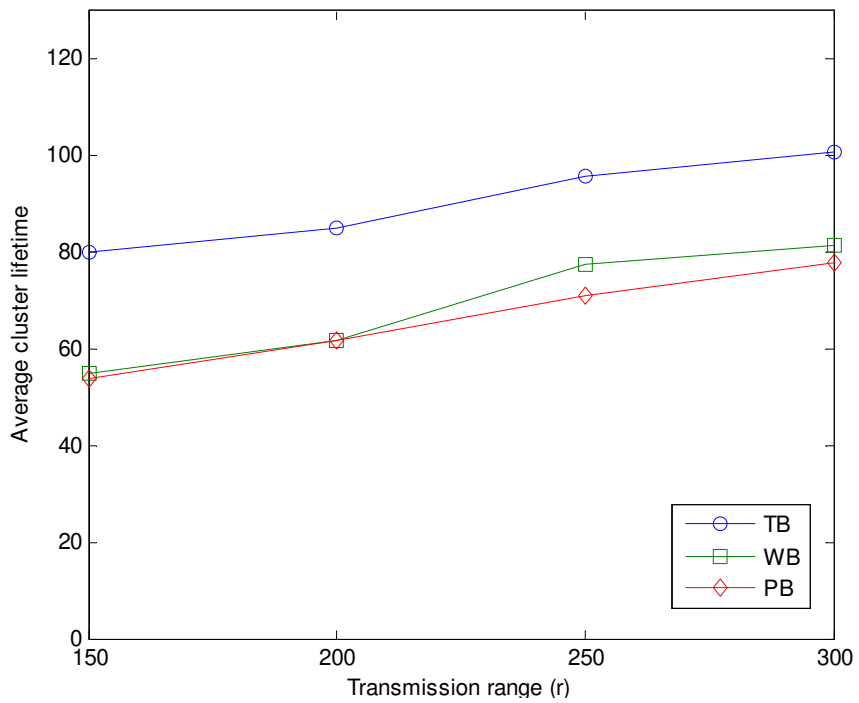
4.3.2.2 Average Cluster Lifetime

The average cluster lifetime is an important metric that shows the performance of the clustering algorithm. The cluster lifetime is directly related to the lifetime of its cluster-head. The cluster-head lifetime is defined as the time period from the moment when a vehicle becomes a cluster-head to the time when it is merged with a nearby cluster.

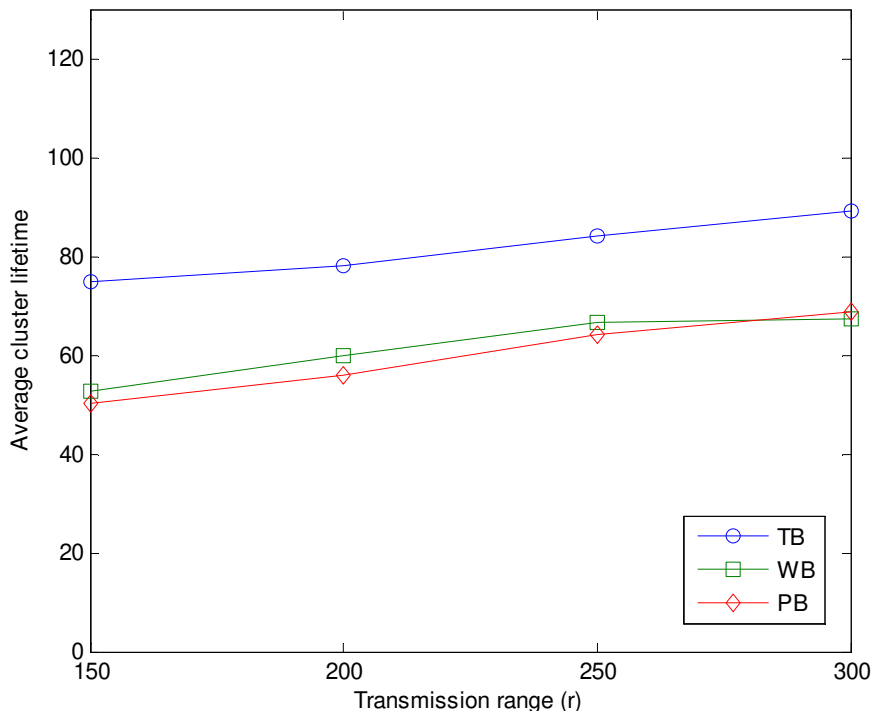
The average cluster lifetime produced by the TB, the WB and the PB methods is compared in different speed scenarios with different transmission ranges. Figure 4.7 (a) (b) (c) show that the average cluster lifetime is increased by 20% to 48% when the TB method is used compared to the WB and PB methods. This is due to the high variation of the speed difference among cluster members of the WB and the PB methods. This deviation leads to the following: first, in both methods, the probability that two cluster heads come into direct communication range is high which results in cluster merging. But, in the TB method, the cluster merging can't be performed unless the difference between the average speed of the cluster heads of both clusters are within the predefined threshold; second, the probability that the cluster members and the cluster-head get separated soon due to high mobility; especially when the cluster is composed of few nodes.



(a) $\mu=70\text{km/h}$, $\sigma=21\text{km/h}$, $\Delta v_{th} = \sigma$



(b) $\mu=90\text{km/h}$, $\sigma=27\text{km/h}$, $\Delta v_{th} = \sigma$



(c) $\mu=110\text{km/h}$, $\sigma=33\text{km/h}$, $\Delta v_{th} = \sigma$

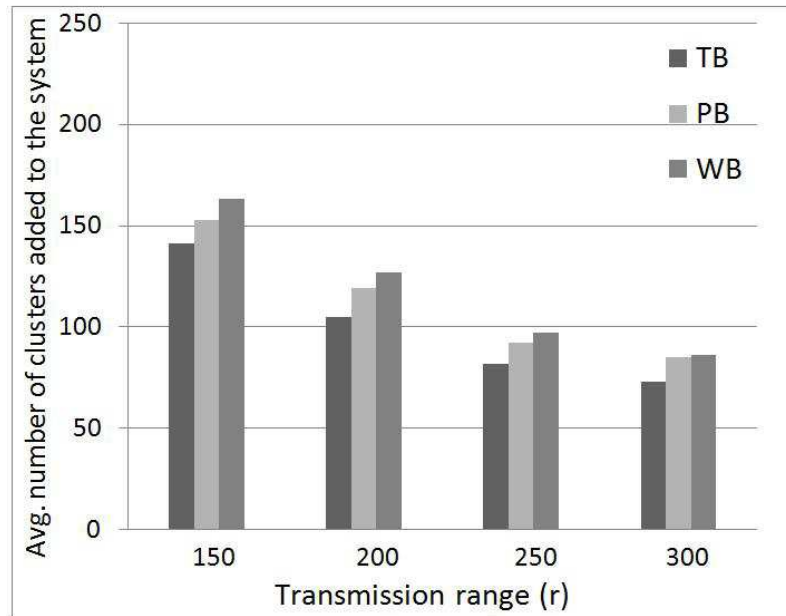
Figure 4.7 average cluster lifetime

4.3.2.3 Number of Clusters

Due to high dynamics of the VANET, clusters are created (new clusters added to the system) and vanished over time, and the total number of clusters created over a period of time defines the cluster formation rate. Good clustering algorithms should be designed to reduce the rate at which clusters are created and added to the system due to the mobility of the nodes. And this can be achieved by producing relatively stable clusters and by the ability of clustering method to maintain the current cluster structure stable as much as possible. In this paper, we compare the average number of clusters added to the system, we start counting each new cluster added to the system after the algorithm is executed by

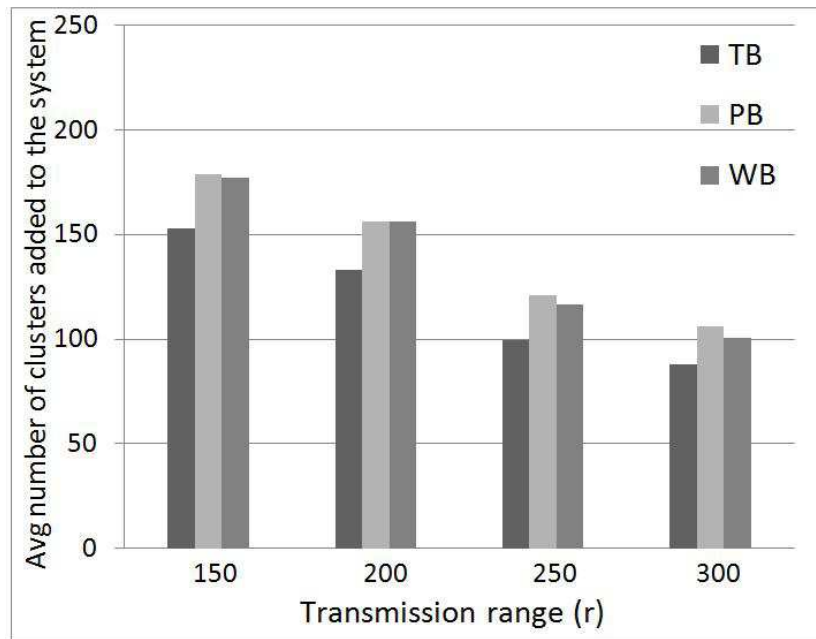
all nodes and the clusters are formed (e.g., when nodes leave their current clusters due to mobility and form a new cluster, or when two neighboring clusters merge to produce a new cluster). To evaluate this metric, the total number of clusters created and added is calculated for each run, then, the *average number of the total number of the created clusters*, C_{avg}^{total} , of all methods is taken over all runs for different transmission ranges.

Figure 4.8 (a) (b) (c) show the average number of the total number of the clusters, C_{avg}^{total} , added to the system over all simulation runs for different speeds and different transmission ranges. The figure shows that the C_{avg}^{total} produced by the TB method is always smaller compared to that produced by the WB and the PB methods and this number decreases as the transmission range increases. This is because the TB method uses the speed difference among vehicles as a parameter to create the clusters. Thus, the

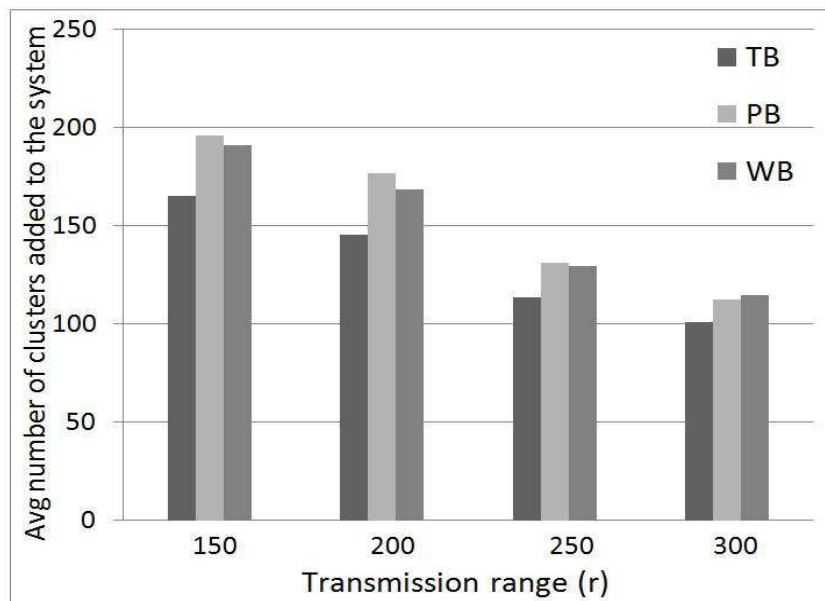


$\mu=70\text{Km/h}$ and $\sigma=21\text{Km/h}$, $\Delta v_{th}=\sigma$

(a)



$\mu=90\text{Km/h}$ and $\sigma=27\text{Km/h}$, $\Delta v_{th}=\sigma$
(b)



$\mu=110\text{Km/h}$ and $\sigma=33\text{Km/h}$, $\Delta v_{th}=\sigma$
(c)

Figure 4.8 average total number of formed clusters for TB, PB, and WB

clusters are more stable and have longer lifetime.

4.3.2.4 Overhead for Clustering

All clustering algorithms incur some additional signaling overhead to form and maintain their cluster structures. The clustering overhead consists of: HELLO packets overhead, cluster setup overhead and cluster maintenance overhead.

Overhead due to HELLO packets: HELLO packets are broadcast by vehicles every T_{HELLO} period. These packets carry local mobility information used to compute local variability, which will be used in cluster formation and cluster-head election. Each node sends one HELLO packet every T_{HELLO} period to maintain up-to-date neighborhood information. Thus, this overhead is the same for TB, WB and PB clustering techniques.

Overhead due to cluster setup: According to the TB cluster formation algorithm, the COV node sends one message to initiate cluster formation process (*InitiateCluster*). After receiving this message, the node that wins the cluster-head competition broadcasts a cluster formation message (*FormCluster*) to its neighbors with its ID embedded in the message. So for the cluster formation process, two messages are sent: one by the COV and the other one by the cluster-head winner node. Each non-clustered neighbor that satisfies the speed threshold joins this cluster by sending a message. So in the TB algorithm, if the average number of nodes in a cluster is K_{TB} , then the total number of messages to setup a cluster is $2 + K_{TB}$. For the PB algorithm, when a new node is powered up and none of its neighbors belong to other clusters, it announces itself as a cluster-head and sends a message to inform its neighbors about its new role. Neighbors that are in the registration phase (non-clustered) join this cluster by sending a join

message. So in the PB algorithm, the total number of messages to setup a cluster is $1 + K_{PB}$, where K_{PB} is the average number of members per PB cluster. In the WB algorithm, a node that claims to be a cluster-head sends a CH-HELLO message. All non-clustered neighbors join this cluster by sending a message. So in the WB algorithm, the total number of messages to setup a cluster is $1 + K_{WB}$, where K_{WB} , is the average number of members per WB cluster. In the TB technique the average number of nodes per cluster is less than that of the other two techniques. So if a TB cluster has at least two less members than the other two types of clusters, then the cluster setup overhead per cluster is less in TB technique than in other techniques.

Overhead due to cluster maintenance: Cluster maintenance is done periodically by all clustering methods. The three types of events that trigger topology change in VANET can be defined as follows: a node joins the network, a node leaves the cluster, and two cluster heads come into direct communication range. If the new node, that joins the network, has non-clustered neighbors, then those nodes will form a new cluster according to the rules used by each clustering method. The overhead of cluster formation was explained earlier. However, if the new node has a neighbor that is a cluster-head, then it will try to join the cluster by sending a join message to the cluster-head, and this cluster joining overhead is same for all three methods (TB, PB and WB). When two neighboring clusters merge, the cluster-head with less number of members will lose its role and join the other cluster and become a cluster member. The losing node sends one message in one period to inform its members about its decision. If the losing node has cluster members, then the members are subject to cluster reorganization. The cluster members either join any nearby clusters or form a new cluster if they couldn't find a cluster to join. Overhead for joining any nearby

clusters is the same for all three methods, and the overhead for cluster formation (cluster setup) is already presented before. The upper bound on the number of messages for cluster merging is equal to the average number of members per cluster, which is K_{TB} , K_{WB} and K_{PB} for TB, WB and PB techniques respectively.

CHAPTER 5: CONCLUSION

In this work, we proposed channel allocation and medium access organization for V2R and V2V communications respectively. We mainly focused on channel allocation for RSU access. The objective of the proposed method is to give the tasks that have been using wireless channel for long period of time the chance to complete, otherwise the resources allocated to those tasks is a waste. The proposed algorithm allocates a virtual transmission plan for vehicles that are closer than others to leave the RSU range. The basic idea is to calculate the expected task finish time and then allocate extra time as part of the transmission plan of the vehicle. This extra time can be used in case the vehicle couldn't finish its task on time. However, this extra time can be assigned to the next vehicle to leave the RSU in case the leading vehicle was able to finish the task on time. The algorithm reduces the risk of the vehicle as it progresses and moves toward the edge of the RSU transmission range. The performance of the algorithm was evaluated using simulation. The results show that the algorithm can reduce the task failure rate compared to other existing methods. The results also show that the proposed algorithm can use the resources efficiently and increases the throughput of the system.

We also proposed a hybrid media access method for cluster-based vehicular networks. This method integrates the centralization approach of cluster management and the universal way of forwarding data, where the farthest vehicle forwards data in an effort to maximize the opportunity of advanced notification. This method leverages contention-free and contention-based Media Access Control to support different requirements of safety and non-safety messages. This method relies on the cluster-head for intra-cluster management and on the cluster-forwarder for safety message dissemination. The

performance of the proposed method was evaluated via simulation program. The results show that, our method provides an early notification compared to the other methods that rely on cluster-head to send warning messages.

We also proposed a new VANET cluster formation algorithm that tends to group vehicles showing similar mobility patterns in one cluster. This algorithm takes into account the speed difference among vehicles as well as the position and the direction during the cluster formation process. After conducting a simulation experiment, we observe that our technique groups fast moving vehicles on the fast speed lanes in one cluster, while slow moving vehicles in another cluster. The simulation results show that our proposed algorithm increases the cluster lifetime and reduces vehicle transitions between clusters. The results show that our technique significantly increases the stability of the global network topology by reducing the rate at which clusters are created.

CHAPTER 6: FUTURE WORK

Intelligent transportation systems will rely on the V2V and V2R communications to increase drivers' and passengers' safety and comfort. Therefore, it's very important to develop new methods for medium access and channel allocation to support these types of applications. Since the mobility patterns of the vehicles are predictable, then new scheduling and channel allocation algorithms should take advantage of these characteristics to enhance their functionalities. The coexistence of different types of traffic (i.e., real-time and non real-time data) that have different requirements should also be considered. Channel allocation and admission control algorithms used by the RSU should react fast to the conditions and should be able to re-calculate the transmission plans in an optimal way to increase system throughput.

BIBLIOGRAPHY

1. IEEE 802.11 TGP, Wireless Access for Vehicular Environment (WAVE).
2. NOW (Network On Wheels) project, <[http:// www.network-on-wheels.de/](http://www.network-on-wheels.de/)>
3. Fleetnet project < [http:// www.et2.tu-harburg.de/fleetnet/](http://www.et2.tu-harburg.de/fleetnet/)>.
4. T. ElBatt, S. K. Goel, G. Holland, H. Krishnan, J. Parikh, "Cooperative collision warning using dedicated short range wireless communications," Proceedings of ACM VANET, New york, USA, 2006, Page(s): 1-9
5. Q. Xu, T. Mark, J. Ko, R. Sengupta, "Medium Access Control Protocol Design for Vehicle-Vehicle Safety Messages," IEEE Transactions on Vehicular Technology, Vol. 56, N. 2, pp.499-518, March 2007.
6. S. Dashtinezhad, T. Nadeem, B. Dorohonceanu, C. Borcea, P. Kang, L. Iftode, "TrafficView: A Driver Assistant Device for Traffic Monitoring based on Car-to-Car Communication," Proceedings of the IEEE Semiannual Vehicular Technology Conference, Milan, Italy, May 2004.
7. T. Nadeem, S. Dashtinezhad, C. Liao, L. Iftode, "TrafficView: Traffic Data Dissemination using Car-to-Car Communication," ACM Mobile Computing and Communications Review (MC2R), New York, USA, Vol. 8, No. 3, pp. 6-19, July 2004.
8. Standard Specification for Telecommunications and Information Exchange between Roadside and Vehicle Systems — 5 GH Band Dedicated Short Range Communications (DSRC) Medium Access Control (Mac) and Physical Layer (PHY) Specifications, ASTM E2213-03, September 2003.

9. C.-T. Chou, S.N. Shankar, and K.G. Shin. Achieving per-stream QoS with distributed airtime allocation and admission control in IEEE 802.11e wireless LANs. In Proceedings of IEEE Infocom'05, Page(s):1584 - 1595 vol. 3, 2005.
10. S. Sriram, T. Bheemarjuna Reddy, B. S. Manoj, and C Siva Ram Murthy. On the end-to-end call acceptance and the possibility of deterministic QoS guarantees in ad hoc wireless networks. In Proceedings of ACM MobiHoc'05, Page(s): 169 - 180, 2005.
11. B. Yu, C. Xu, "Admission Control for Roadside Unit Access in Intelligent Transportation Systems". In Proceedings of the 17th international workshop on the Quality of Service (IWQoS '09), page(s) 1- 9, SC, USA, 2009
12. C.-J. Chang, R.-G. Cheng, H.-T. Shih, and Y.-S. Chen, "Maximum Freedom Last Scheduling Algorithm for Downlinks of DSRC Networks," in: IEEE Transactions on Intelligent Transportation Systems: Volume 8, Number 2, IEEE, New York, 2007.
13. R.A Santos et al, "Supporting Inter-Vehicular and Vehicle-Roadside Communications over a Cluster-Based Wireless Ad-Hoc Routing Algorithm," Proceedings of the winter international symposium on Information and communication technologies, pp. 1-6, Trinity College Dublin, 2004.
14. M. Aoki and H. Fuji, "Inter-vehicle communication: Technical issues on vehicle control application," IEEE Commun. Mag., vol. 34, no. 10, pp. 99-93, Oct. 1996.
15. N. Wisitpongphan, O. Tonguz, J. Parikh, F. Bai, P. Mudalige, and V. Sadekar, "On the Broadcast Storm Problem in Ad hoc Wireless Network," in the IEEE 3rd , BROADNETS, San Jose, CA, USA, Oct. 1 2006, pp. 1-11.

16. H. Alshaer and E. Horlait, "An Optimized Adaptive Broadcast Scheme for Inter-Vehicle Communication," Proceedings of the 61st IEEE Vehicular Technology Conference (VTC 2005 Spring), Stockholm, Sweden, May 2005.
17. G. Korkmaz, E. Ekici, and F. Ozguner, "An efficient fully ad-hoc multihop broadcast protocol for inter-vehicular communication systems," Proceedings of the IEEE ICC'06, Istanbul, Turkey, June 2006.
18. C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," IEEE J. Sel. Areas of Communications., vol. 15, no. 7, pp. 1265-1275, Sep. 1997.
19. W. Chen and S. Cai, "Ad hoc peer-to-peer network architecture for vehicle safety communications," IEEE Communication. Mag., No 4, 2005, pp.100-107.
20. Reumerman, H.-J. Roggero, M. Ruffini, M, "The application-based clustering concept and requirements for intervehicle networks," IEEE Commun. Mag., No. 4, 2005, pp. 108-113.
21. R.A. Santos, R.M. Edwards and N.L. Seed, Inter Vehicular Data Exchange between fast moving road traffic using Ad-Hoc Cluster Based Location Algorithm and 802.11b Direct Sequence Spread Spectrum Radio, PostGraduate Networking Conference, 2003.
22. G. Yvonne W. Bernhard G. Hans Peter" Medium Access Concept for VANETs Based on Clustering," in the IEEE 66th VTC, Sept. 30 2007, pp. 2189-2193.
23. X. Zhang, H. Su, and H.H. Chen "Cluster-Based DSRC Architecture for QOS provisioning over Vehicle Ad Hoc networks," in IEEE GlobeCom'06, Nov. 2006, pp. 1-5.

24. P. Fan "Improving Broadcasting Performance by Clustering with Stability for Inter-Vehicle Communication," Proceedings of the 65th IEEE Vehicular Technology Conference (VTC 2007 Spring), Dublin, Ireland, April 2007.
25. P. Fan, P. Sistla, P. Nelson, "Theoretical analysis of a directional stability-based clustering algorithm for VANETs," Proceedings of the Fifth ACM international workshop on Vehicular ad hoc networks (VANET), San Francisco, CA, USA, 2008.
26. 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, November 2007.
27. Lei Huang; Kumar, S.; Kuo, C.-C.J.; , "Adaptive resource allocation for multimedia QoS management in wireless networks," Vehicular Technology, IEEE Transactions on , vol.53, no.2, pp. 547- 558, March 2004.
28. Yin Ge; Geng-Sheng Kuo; , "An Efficient Admission Control Scheme for Adaptive Multimedia Services in IEEE 802.16e Networks," Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th , vol., no., pp.1-5, 25-28 Sept. 2006.
29. Jing Liu; Zhisheng Niu; , "A Dynamic Admission Control Scheme for QoS Supporting in IEEE 802.11e EDCA," Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE , vol., no., pp.3697-3702, 11-15 March 2007.
30. W. Tan, W. Lau, and O. Yue. " Modeling Resource Sharing for a Road-side Access Point Supporting Drive-thru Internet ", In the proceeding of the sixth ACM international workshop on VehiculAr InterNETworking , VANET'09, Sep. 2009

31. Wha Sook Jeon; Dong Geun Jeong; , "Combined Connection Admission Control and Packet Transmission Scheduling for Mobile Internet Services," Vehicular Technology, IEEE Transactions on , vol.55, no.5, pp.1582-1593, Sept. 2006
32. V. Firoiu, J. Kurose, and D. Towsley. Efficient admission control for EDF schedulers. In Proceedings of IEEE Infocom' 97, Page(s):310 - 317 vol.1, 1997.
33. S. Shin, H. Schulzrinne. Call admission control in IEEE 802.11 WLANs using QP-CAT. In Proceedings of IEEE Infocom'08, Page(s):726 - 734, 2008.
34. D. Hadaller, S. Keshav, and T. Brecht. MV-MAX: Improving Wireless Infrastructure Access for Multi-Vehicular Communication. In Proc. CHANTS, 2006.
35. X. Liu, E. Chong, and N. Shroff. A Framework for opportunistic Scheduling in Wireless Networks. Computer Networks Journal, 41(4): 451-474, 2003.
36. Y. Zhang, J. Zhao, and G. Cao. On scheduling vehicle roadside data access. In Proceedings of ACM VANET'07 Workshop, Page(s): 9 - 18, 2007.
37. A. B. McDonald and T. F. Znati. A mobility-based framework for adaptive clustering in wireless ad hoc networks. IEEE Journal on Selected Areas in Communications, 17(8):1466-1486, Aug. 1999.
38. I. I. Er and W. K. Seah, .Mobility-based d-hop clustering algorithm for mobile ad hoc networks,. in IEEE Wireless Communications and Networking Conference, Atlanta, Georgia, USA, Mar. 2004.
39. IEEE Standard 802.11 - 1999; Wireless LAN medium access control (MAC) and physical layer (PHY) specifications; November 1999.

40. F. G. Nocetti, J. S. Gonzalez, I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks," *Telecommunication Systems* 22 (2003) 1-4, 205-220, 2003.
41. M. Gerla and J. T. C. Tsai, Multicluster mobile multimedia radio networks, *ACM-Baltzer Journal of Wireless Networks*, 1(3):255-265, 1995.
42. A.K. Parekh, Selecting routers in ad hoc wireless networks, *Proceedings of SBT/IEEE International Telecommunications Symposium*, August 1994.
43. mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11):1694-1701, 1981.
44. A. Ephremides, J. Wieselthier, and D. Baker. A design concept for reliable mobile radio network with frequency hopping signaling. In *Proceedings of IEEE 75*, pages 56-73, 1987.
45. P. Basu, N. Khan, and T. D. C. Little. Mobility based metric for clustering in mobile ad hoc networks. *Workshop on Distributed Computing Systems*, pages 413-418, 2001.
46. C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. *IEEE Singapore International Conference on Networks (SICON)*, pages 197- 211, Apr. 1997.
47. A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh. Max-min d cluster formation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM '00*, Vol. 1, pages 32-41, Mar. 2000.
48. C.R. Lin and M. Gerla, Adaptive clustering for mobile wireless networks, *IEEE Journal on Selected Areas in Communications* 15(7), pp. 1265-1275, 1997.

49. W. Zhiagang L. Lichuan Z. MengChu. A. Nirwan "A Position-Based Clustering Technique for Ad Hoc Intervehicle Communication,"IEEE Trans.on MAN, AND CYBERNETICS, vol. 38, No. 2, March 2008.
50. C. Shea, B. Hassanabadi and S. Valaee, "Mobility-based Clustering in VANETs using Affinity Propagation," Proceedings of the IEEE Globecom, Honolulu, HI, Nov.30 - Dec.4, 2009
51. E. de Souza, I. Nikolaidis, and P. Gburzynski, "A New Aggregate Local Mobility (ALM) Clustering Algorithm for VANETs," Proceedings of the IEEE ICC'10, May.23-27, Cape Town, South Africa, 2010.
52. R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. <http://www.ietf.org/rfc/rfc3561.txt>, July 2003. IETF Internet.
53. David B. Johnson, David A. Maltz, Yih-Chun Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). <http://www.ietf.org/internet-drafts/draft-ietfmanet-dsr-08.txt>, April 2003. IETF Internet Draft.
54. O. Kayis and T. Acarman, "Clustering formation for inter-vehicle communication," Proceedings of the 10th IEEE Intelligent Transportation Systems Conference (ITSC'07), Seattle, WA, USA, Sept. 30 - Oct. 3, 2007.
55. D. Hadaller, S. Keshav, T. brecht, et al. Vehicular opportunistic communication under the microscope. In proceedings of the 5th international conference on Mobile Systems, Applications, and Services (MobiSys '07), 2007.
56. T. K. Mak, K. P. Laberteaux, R. Sengupta. A MultiChannel VANET providing concurrent safety and commercial services. In Proceedings of ACM VANET'05 Workshop, Page(s): 1 - 9, 2005.

57. J. J. Meyers. Channel characterization and reliability of 5.8 GHz DSRC wireless communication links in Vehicular Ad Hoc Networks in suburban driving environment. Master thesis, Carnegie Mellon University, 2005.
58. F. Bai, H. Krishnan, "Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications," proceeding of the 9th International ITSC, 2006, pp. 355-362.
59. W. Schnabel, D. Lohse: "Grundlagen der Straenverkehrstechnik und der Verkehrsplanung," Bd. 1, 2. Aufl., Verlag frBauwesen, Berlin, 1997.
60. S.Yousefi, E Altman,R El-Azouzi,and M. Fathy,"Analytical Model for Connectivity Ad Hoc Networks," IEEE Trans. On Vehicular Technology, vol. 57, no, 6, pp. 3341-3356 Nov 2008.
61. M. Rudack, M. Meincke, and M. Lott,"On the dynamics of ad hoc networks for inter-vehicles communications (IVC)," Proceedings of the ICWN, 2002, pp. 40-48.
62. Zaydoun Yahya Rawashdeh and Syed Masud Mahmud, "Toward Strongly Connected Clustering Structure in Vehicular Ad hoc Networks," Proceedings of the 2009 IEEE 70th Vehicular Technology Conference: VTC2009-Fall, September 20-23, 2009, Anchorage, Alaska, USA
63. Radovan Miucic, Zeljko Popovic and Syed Masud Mahmud, "Experimental Characterization of DSRC Signal Strength Drops," Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, October 3-7, 2009, St. Louis, Missouri, USA
64. Vehicle Infrastructure Integration (VII) California Demonstration Evaluation, Final Evaluation Report, Kimley-Horn and Associates, Inc., July 11, 2006.
<[http:// www.viicalifornia.org/publications/vii demonstration evaluation.pdf](http://www.viicalifornia.org/publications/vii_demonstration_evaluation.pdf)>

65. S. Basagni, I. Chlamtac, and A. Farago, "A Generalized Clustering Algorithm for Peer-to-Peer Networks," Workshop on Algorithmic Aspects of Communication (satellite workshop of ICALP), Bologna, Italy, July 1997.
66. S. Basagni, "Distributed Clustering for Ad Hoc Networks," International Symposium on Parallel Architectures, Algorithms and Networks', Perth, June 1999, pp. 310-315.
67. M. Chatterjee, S.K. Sas, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," Proceedings of the 43rd IEEE Global Telecommunications Conference, San Francisco, CA, 2000, 1697-1701.

ABSTRACT**EFFICIENT CHANNEL ALLOCATION AND MEDIUM ACCESS ORGANIZATION ALGORITHMS FOR VEHICULAR NETWORKING**

by

ZAYDOUN RAWASHDEH**August 2011****Advisor:** Dr. Syed Masud Mahmud**Major:** Computer Engineering**Degree:** Doctor of Philosophy

Due to the limited bandwidth available for Vehicular Ad-hoc Networks (VANETs), organizing the wireless channel access to efficiently use the bandwidth is one of the main challenges in VANET. In this dissertation, we focus on channel allocation and media access organization for Vehicle-to-Roadside Units (V2R) and Vehicle-to-Vehicle (V2V) communications. An efficient channel allocation algorithm for Roadside Unit (RSU) access is proposed. The goal of the algorithm is to increase system throughput by admitting more tasks (vehicles) and at the same time reduce the risk of the admitted tasks. The algorithm admits the new requests only when their requirements can be fulfilled and all in-session tasks' requirements are also guaranteed. The algorithm calculates the expected task finish time for the tasks, but allocates a virtual transmission plan for the tasks as they progress toward the edges of the RSU range. For V2V mode, we propose an efficient medium access organization method based on VANETs' clustering schemes. In order to make this method efficient in rapid topology change environment like VANET, it's important to make the network topology less dynamic by forming local strongly connected clustering structure, which leads to a stable network topology on the global scale. We propose an efficient cluster formation algorithm that takes vehicles' mobility

into account for cluster formation. The results of the proposed methods show that the wireless channel utilization and the network stability are significantly improved compared to the existing methods.

AUTOBIOGRAPHICAL STATEMENT

Zaydoun Rawashdeh received the B.Sc. and M.Sc. degrees in Electrical and Computer Engineering from Saint Petersburg State Electro-technical University, Russia. He received his second M.Sc. degree in Computer Engineering from the University of Michigan, USA. From 1999 to 2004 he worked as a Computer Network Engineer at the University of Jordan. From 2007 to 2011, Rawashdeh worked as a Teaching Assistant in the ECE department at Wayne State University. Currently he works as a Wireless Research Engineer at the Crash Avoidance Metrics Partnership (CAMP) Vehicle Safety Communications 3 (VSC3) Consortium. He received the Outstanding Teaching Assistant Award at Wayne State University in 2009. Rawashdeh received the Student Paper Award in the 2010 ITS-Michigan Annual Meeting. He is a member of IEEE and Tau Beta Pi. His main research field is the Vehicular Ad hoc Networks. His other research interests include MANET and Embedded Systems. His recent publications:

1. Zaydoun Yahya. Rawashdeh and Syed Masud Mahmud "A Novel Algorithm to Form Stable Clusters in Vehicular Ad hoc Networks on Highways" submitted to the EURASIP Journal on Wireless Communications and Networking.
2. Zaydoun Yahya. Rawashdeh and Syed Masud Mahmud" Admission Control for Roadside Units Based on Virtual Air-time Transmissions," to appear in the Proceedings of the IEEE Global Communication Conference 2011.
3. Zaydoun Yahya. Rawashdeh and Syed Masud Mahmud" Communications in Vehicular Networks," in *Theory and Applications of Ad Hoc Networks*, InTech, Zieglergasse 14, 1070 Vienna, Austria, 2010.
4. Zaydoun Yahya. Rawashdeh and Syed Masud Mahmud, "Toward Strongly Connected Clustering Structure in Vehicular Ad hoc Networks," Proceedings of the 2009 IEEE 70th Vehicular Technology Conference: VTC2009-Fall, September 20–23, 2009, Anchorage, Alaska, USA.
5. Zaydoun Y. Rawashdeh and Syed M. Mahmud, "Teaching Real-Time Embedded Systems Networking and Assessment of Student learning, "Proceedings of the 2009 ASEE Annual Conference, Austin, Texas, June 14 – 17, 2009.
6. Zaydoun Yahya Rawashdeh and Syed Masud Mahmud, "Media Access Technique for Cluster-Based Vehicular Ad Hoc Networks," Proceedings of the IEEE 68th Vehicular Technology Conference: VTC2008-Fall, Calgary, Canada, September 21 - 22, 2008.
7. Zaydoun Yahya Rawashdeh and Syed Masud Mahmud, "Intersection Collision Avoidance System Architecture," Proceedings of the 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, January 10-12, 2008, Paper Number: FA1-S6-6.
8. Zaydoun Yahya Rawashdeh and Syed Masud Mahmud, "Effectiveness of Using Periodic Messages for Disseminating Warning Information in Vehicular Ad Hoc Networks," Proceedings of the 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, January 10-12, 2008, Paper Number: FA2-S7-6