Cluster-Based Target Tracking in Vehicular Ad Hoc Networks

by

Sanaz Khakpour

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Applied Science

in

Computer Science

The Faculty of Business and Information Technology

University of Ontario Institute of Technology (UOIT)

Oshawa, Ontario, Canada

January 2015

Abstract

Recently Vehicular Ad-hoc Networks (VANETs) have drawn the attention of academic and industry researchers due to their potential applications in enabling Intelligent Transportation System (ITS), including safe driving, entertainment, emergency response, and content sharing. Another potential application for VANET lies in vehicle tracking, where a tracking system is used to visually track a specific vehicle or to monitor a particular area. In this case, and in similar applications such as multimedia content sharing, a large volume of information is required to be transferred between vehicles, which can easily congest the wireless network in a VANET if not designed properly. The development of low-delay, low-overhead, and precise tracking system in VANET is a major challenge requiring novel techniques to guarantee performance and reduce network congestion.

Among the several proposed data dissemination and management methods implemented in VANETs, clustering has been used to reduce data propagation traffic and to facilitate network management. However, clustering for target tracking in VANETs is still a challenge. In this thesis, we propose two clustering algorithms for vehicle tracking in VANETs. These algorithms provide a reliable and stable platform for tracking specific vehicles based on their visual features under various conditions. These algorithms have also been tested and evaluated in the context of vehicular tracking under various scenarios. Performance evaluation results demonstrate that the proposed schemes provide a more stable clustering structure with reduced overhead.

ACKNOWLEDGEMENTS

I am sincerely grateful to Dr. Khalil El-Khatib and Dr. Richard W. Pazzi for their guidance and encouragement during my master’s studies.

I would also like to thank my friends in the Advanced Networking Technology and Security (ANTS) Lab specially Leandro Ballico for his help.

The financial support of Natural Science and Engineering Research Council of Canada, Developing Next Generation Intelligent Vehicular Networks and Applications (NSERC DIVA) is gratefully acknowledged.

Lastly, I would like to offer special gratitude to my family specially my parents, my aunt, and my spouse for their unconditional love, constant support, and encouragement.

PUBLICATIONS

The following publications are resulted from the work in this thesis:

Sanaz Khakpour, Richard Werner Pazzi, and Khalil El-Khatib. 2013. A distributed clustering algorithm for target tracking in vehicular ad-hoc networks. In *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications* (DIVANet '13). ACM, New York, NY, USA, 145-152. DOI=10.1145/2512921.2512933 http://doi.acm.org/10.1145/2512921.2512933

Sanaz Khakpour, Richard W. Pazzi, and Khalil El-Khatib. 2014. A prediction based clustering algorithm for target tracking in vehicular ad-hoc networks. In *Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications* (DIVANet '14). ACM, New York, NY, USA, 39-46. DOI=10.1145/2656346.2656412 http://doi.acm.org/10.1145/2656346.2656412

Table of Contents

Table of Figures

List of Tables

1. Introduction

Vehicular Ad hoc Networks (VANET) play an important role in Intelligent Transportation Systems (ITS) by providing critical information about roads and traffic condition, sending safety messages, and providing entertainment for passengers. In VANETs, vehicles can connect to each other for many purposes such as exchanging safety and infotainment messages. A special characteristic of VANET nodes, compared to nodes of other ad hoc networks such as MANET, is the abundant on-board processing resources of the vehicles which make them suitable platforms for processing complex algorithms for various applications.

Over the last few years, a number of research have been conducted on VANETs, mainly focusing on routing techniques and data dissemination under various road and traffic conditions [1] [2] [3], localization of nodes [4] [5], location privacy protection [6], communication security [7], social networking and advertisement [8, 9].

While VANET is still in its infancy, a number of applications which are not safety related have been proposed in the literature. One of the main applications is target tracking, where an object vehicle is located and tracked using some of the on-board vehicle sensors such as cameras. Such applications may be used by police agencies to locate a specific vehicle with particular visual features such as license plate information, color, model, etc. Even though police agencies might rely on pre-installed security camera infrastructure across the city, the cost of installing cameras to cover all roads can be very high. Also, there is a probability of losing the target in non-monitored areas. However, many vehicles on the roads are getting equipped with front and rear cameras and on-board communication capabilities that can be used as parts in a mobile tracking system. Another application of this system is in passive monitoring to collect video footage of incidents that happened in areas where security camera systems are unavailable, and using therefore only the cameras of passing by vehicles may be relied upon

One of the challenges in continuous monitoring systems in VANET is bandwidth availability, which can be a limiting factor especially when there are multiple sources in close proximity streaming video data [10, 11]. A traditional solution to control bandwidth usage in ad hoc networks is to segment the network into clusters and select one representative (cluster head) for each cluster to act as a connection point to the cluster [12]. However in a highly dynamic environment such as VANET, selection of appropriate metrics for cluster head election and cluster membership can be a challenge as vehicles constantly enter and leave the clusters.

1.1 Problem Statement

The main goal of this thesis is to provide an appropriate framework for vehicle tracking in VANETs. Target tracking can be simply performed if the target vehicle has a Global Positioning System (GPS), with the location data communicated to external entities. However, we assume that such devices are not available or have been turned off on target vehicles. In order to solve this issue, we rely on visual identification of target vehicle using the on-board cameras of neighboring vehicles and reporting the location and visual information of the target to a control center. The control center is assumed to be a police station looking for a special vehicle based on its visual description and is interested to acquire location and visual information of the target.

Therefore, if a suitable framework is not provided, every vehicle that detects the target will broadcast location and visual information of the target towards the control center. In VANETs, nodes communicate with each other through multi-hop message transmission. In case the control center is located in a multi-hop distance from the target, there is a high probability of network congestion, packet collision and packet loss because of concurrent transmission of target's information by all the neighboring vehicles in a multi-hop manner [13, 14]. Also, the control center might receive duplicate messages which are unnecessary and redundant. This problem is due to unavailability of a central aggregator node to collect information from neighboring vehicles and to process and aggregate them. The other concern in such a system is data overload in the control center due to direct transmission of target's information by all target's neighbor vehicles to a central entity.

In order to address these problems, we have considered a cluster-based framework to organize the network. Therefore, target's neighbor vehicles which can detect the target join a cluster and select a leader node or cluster head (CH). The neighbor nodes send their target's information to the leader. The leader node is responsible for aggregating the information and sending it to the control center. So, instead of every node sending its information to the control center separately, there is only one node responsible for delivering the information to the control center.

The challenges towards designing a high-performance and efficient clustering algorithm mostly include clustering stability improvement and control overhead reduction. Due to high velocity of vehicles in VANET, the changes in the cluster structure can be so high. As a result, the cluster memberships change so rapidly. Also, the eligibility of current CH may change so fast which causes high number of cluster head changes. Any changes in the cluster structure require control messages transmission in the cluster to inform other nodes about the change. Broadcasting of control messages causes overhead in the cluster [15, 16]. So, it is of great importance to propose appropriate cluster membership and CH selection rules that help to increase cluster member and cluster head lifetime as much as possible; whereas, providing application requirements. Also, the CH should have information from all the member nodes which is retrieved from periodic control messages transmission. The other critical challenge in clustering algorithms is fast growth of control overhead in the cluster. Control packets can congest the cluster if not managed properly. Therefore, employing ideas to decrease control overhead of a cluster structure is a necessary step towards an efficient clustering protocol.

The other problem we are addressing in this thesis is data packets dissemination from the CH to control center. After the CH aggregates received information from cluster members, it will transmit the information to the control center. The challenge is sending large volumes of data to a multi-hop distant destination. If the information is being broadcasted from the CH towards control center, congestion may happen in the network. Also, if every node carries target's information and sends it to the control center, packet collision happen which results in low delivery ratio. This causes tracking errors due to high packet loss rate and low quality of received visual information and inaccurate location information. Also, re-transmission of data packets leads to high delay and reduces bandwidth efficiency [13]. Therefore, we need to design efficient algorithm to transfer the information from CH to control center with high delivery ratio and low control overhead. The other technique to address such a problem is transmission of information in carry-and-forward manner. This method increases delivery ratio; but, causes higher delay.

1.2 Thesis Contribution

The main contribution of this thesis is two cluster-based target tracking algorithms for vehicular ad hoc networks. The main purpose of both algorithms is to provide an appropriate cluster-based framework for communication of vehicle tracking information to a central entity. We use clustering techniques in both algorithms to reduce congestion and packet loss in the network and increase delivery ratio. Besides, we concentrate on improvement of clustering performance and functionality along with preparation of an appropriate framework for target tracking. The main concerns we addressed in this regard are clustering overhead deduction, cluster stability improvement, and proposing reliable and application-based clustering metrics to serve properly towards target tracking requirements.

The proposed DCTT algorithm is a distributed clustering algorithm that uses mobility features of nodes for cluster formation and leader selection. The distributed structure makes the cluster less vulnerable to topology change which is very important in highly dynamic VANET environment.

The second proposed algorithm is called PCTT which is a centralized clustering protocol. PCTT uses prediction techniques for cluster management and cluster head (CH) selection. Using prediction reduces clustering overhead considerably. Besides, application of prediction-based CH selection rules helps in reduction of cluster structure changes and improvement of cluster stability.

The simulation results represent better performance of both proposed algorithms in comparison to the following approaches.

A. Structureless target tracking system for VANETs

B. Modified DMAC (MDMAC) clustering algorithm [17] adapted for target tracking in VANETs

We have also studied performance of the proposed centralized and distributed clustering algorithms for target tracking application and the simulation results present better performance of centralized approach in terms of overhead reduction and cluster stability than the distributed clustering version. However, the distributed algorithm performs better in lost CH scenarios in terms of delay reduction.

Last but not least, we have conducted a comprehensive survey on VANET clustering protocols and categorization of these protocols based on their CH selection criteria.

1.3 An overview of the proposed clustering algorithms

In this section a brief overview of our proposed algorithms and their main properties are provided. The detailed description of the protocols is presented in Chapter 3 and Chapter 4. The first proposed algorithm is called Distributed Cluster-based Algorithm for Target Tracking in VANETs (DCTT). DCTT is a distributed multi-hop clustering algorithm used for detection and tracking of vehicles based on their visual information such as license plate and colour. In this algorithm nodes should send periodic control packets in order to inform other member nodes and the cluster head about their status. The cluster head selection metric we use for DCTT is referred to as Tracking Failure Probability (TFP) which is a percentage representing a node’s movement similarity to the target. The second proposed algorithm is denoted as Prediction-based Clustering Algorithm for Target Tracking in Vehicular Ad-Hoc Networks (PCTT). This algorithm benefits from prediction-based cluster head selection metric and cluster maintenance functions. In order to decrease clustering control overhead the CH uses a prediction function to estimate cluster members behaviour instead of receiving their information periodically. This technique improves clustering overhead significantly. PCTT is a centralized algorithm and the cluster head is the central managing entity that is responsible for most of the clustering decisions. Because the cluster head can be exposed to failures we have considered a backup mechanism which is selection of a candidate cluster head (CCH). The current cluster head is responsible for selecting a candidate cluster head that can take over the responsibility in case the current cluster head fails. The cluster head selection metric in this algorithm is called Observation Time (OBT) which represents the duration of time the target spends in the field of view of a cluster member. A member node with the highest OBT value is eligible to be the cluster head.

1.4 Thesis Organization

This thesis is organised into six chapters. Chapter One provides a brief introduction to target tracking and clustering in VANETs and outlines the challenges in target tracking, together with a quick review on the contributions of our thesis. Chapter Two presents a comprehensive literature review of VANET's features and applications, cluster-based techniques for VANET environment, and target tracking in these networks. In chapter Three and Four we propose two cluster-based target tracking algorithms for VANETs (DCTT and PCTT). Chapter Five presents the simulation scenarios and results of our proposed algorithms. Finally, this thesis concludes in Chapter Six.

2. Literature Review

In this Chapter, we will provide an introduction to ad hoc networks including VANETs and MANETs. To this end, a brief review of VANET clustering and performance metrics of a cluster-based algorithm for VANET as well as target tracking in VANET will be provided. Finally, we will revisit some of the cluster-based protocols in VANETs and MANETs environments.

2.1 Wireless Ad hoc Networks

Wireless ad hoc networks are decentralized networks of nodes that communicate without any pre-defined infrastructure. These networks can be formed for a short time period according to arisen needs and requirements [18]. Ad hoc networks consist of wireless nodes (mobile or fixed) that can be spread throughout large areas. Nodes communicate with each other via wireless links without any pre-installed infrastructure using broadcast messages and multi-hop communications. Wireless ad hoc networks are categorized into various types including wireless sensor networks (WSNs), mobile ad hoc network (MANETs), vehicular ad hoc networks (VANETs) and wireless mesh networks (WMNs) [18].

2.1.1 Wireless Sensor Networks (WSNs)

Wireless Sensor Networks (WSN) are distributed networks of autonomous sensor nodes deployed in specific places for monitoring purposes [19] [20]. The sensor nodes can be fixed or mobile. These nodes acquire information from their area (based on their application) and send the information to a central entity called sink node. Considerable challenges in WSN area include energy consumption, limited memory, and restricted processing power [21]. A great number of researches on WSNs are dedicated to energy management which focus on increasing network lifetime. Inaccessibility of sensor nodes and deployment in dangerous or hardly accessible areas such as battlegrounds makes it almost impossible to recharge the nodes or replace the batteries. Therefore, many researches in this area concentrate on energy management mechanisms and reducing power consumption without affecting application requirements [22].

2.1.2 Mobile Ad hoc Networks (MANETs)

Mobile Ad hoc Networks (MANET) consist of mobile nodes communicating with each other through wireless links [23]. The neighbour nodes that are in the transmission range of each other can communicate directly. However, if the distance between two nodes is more than the possible transmission range, messages should be transferred through multi-hop communications. MANET has been used mostly for military applications and some civilian applications [24]. The main challenges posed by MANETs are topological changes due to node movement, link bandwidth variations, and power management [24].

2.1.3 Vehicular Ad Hoc Networks (VANET)

Vehicular ad hoc network (VANET) is a special kind of MANET that consist of vehicles using dedicated short-range communication (DSRC) and WAVE (wireless access in vehicular environment) protocol [25]. VANETs are self-organized and self-managed networks capable of working without any pre-installed infrastructure [26]. These networks are composed of mobile nodes that are vehicles equipped with wireless interfaces and communicate with each other through unstructured vehicle to vehicle (V2V) or structured vehicle to roadside infrastructure (V2I) communications. Roadside infrastructures are provided to enable vehicles to connect to external networks such as the Internet [27].

The major purpose of VANET deployment is enabling vehicular communication for special purposes such as reporting traffic conditions, driver’s and passenger’s conditions, sending emergency and collision warnings, monitoring roads surfaces and weather conditions, data sharing, and other safety-related purposes, just to mention a few [28]. VANET is the principal framework for intelligent transportation systems (ITS). ITS is proposed with the purpose of designing vehicle operations, assisting drivers to obtain needed information for safety and entertainment purposes, traffic management, and providing convenience for passengers. ITS is expected to grow as its ultimate goal is the realization of a safe and accident-free driving environment. Automatic toll collection and driving assistance systems may be cited as examples. ITS applications generally require numerous messages being transferred via multiple hops between vehicles to travel from source to destination.

VANET's applications are divided into the following main categories:

A. Navigation safety and driver safety application:

The main purpose behind VANET deployment is defined as providing a safe driving environment as well as pleasant driving experience. The main focus of inter vehicle communication (IVC) is navigation safety. These applications include warnings about road problems, traffic sign conflicts, road conditions, assistance in lane-changing, crash prevention and survivability, and reporting driver's condition [29], [30]. According to the research in [31], safety-related applications are classified by the Vehicle Safety Communications (VSC) into traffic light conflict warnings, curve speed warning, emergency brake lights, pre-crash sensing, cooperative forward collision warning, lane-change warning, and stop sign movement assistant. Some of these applications require vehicle-to-vehicle (V2V) communications, whereas others necessitate vehicle-to-roadside infrastructure (V2I) communication.

B. Emergency routing

These applications include forwarding information during an earthquake, thunderstorm or other natural disasters when network infrastructure is not able to work properly to send data [26]. In the case of natural disasters like an earthquake or a hurricane, the power lines may go down. Therefore the communication infrastructure will not function properly either because of loss of power or due to the congestion in the network [29]. VANET is a network that can still operate under these conditions since it can reconfigure itself to be able to send and receive information. VANET's protocols are designed in such a way as to be capable of functioning without any infrastructure which makes it well suited for emergency situations [29].

C. Entertainment and advertisement applications

Entertainment applications include social networking, content sharing, and location-based roadside advertisement aimed at providing a convenient and pleasant travelling experience for passengers. In this regard, some content sharing protocols are introduced, which may be described as follows [29]:

- Car Torrent is proposed by the UCLA group [32]. This protocol is a BitTorrent style content sharing protocol in wireless sensor networks which uses a proximity-based content sharing method instead of the rarest first piece selection.
- Ad Torrent [33] uses network coding for downloading content. This scheme is based on the idea that downloading from a multi-hop access point or Long-Term Evolution (LTE) might be time consuming and not practical because of traffic overload. Therefore, in this scheme downloading from neighbors is proposed. A vehicle will download any needed piece of information from the nearby vehicles and third parties. The difference between Car Torrent and Ad Torrent is the dissemination of segments in Ad Torrent [29].

D. Monitoring and Tracking

VANET has been used for monitoring traffic conditions and as a communication infrastructure for transmission of monitoring information gathered for various applications. Some of these applications include traffic monitoring and congestion prediction [34, 35], acoustic noise pollution monitoring [36], monitoring of pollution in urban areas [37], and medical monitoring during disasters when most network infrastructures are unavailable [38]. All of these applications use VANET as a framework for transmitting the gathered information due to availability of vehicles and VANET system in most of the areas. The other surveillance application of VANET is monitoring and tracking the moving vehicles based on their visual characteristics. We refer to this application as target tracking using vehicular networks. The VANET monitoring and tracking system requires vehicles to be equipped with cameras capable of detecting particular visual features including license plate, color, accident damage, etc. Our proposed cluster-based VANET tracking systems [39, 40] may also be used as a framework for monitoring and reporting of a specific region for a variety of reasons as long as vehicles exist in the area.

2.2 Different Characteristics of VANET vs. MANET

Vehicular networks have distinctive characteristics and networking properties as compared to MANETs, rendering MANET protocols inapplicable to VANET applications [26, 30]. Some of VANET's special features may be cited as the following:

- Rapid topology changes due to high relative mobility between vehicles.
- Variable velocity of nodes which requires VANETs to have an infrastructureless dynamic topology with partial infrastructure support.
- Fragmented inter vehicle communications and frequently broken connectivity.
- Dependency of topology changes to driver's behavior and reactions to received messages.
- Different communication requirements due to the need to send safety messages which demands reliable, accurate and timely delivery of messages [41, 42].
- Predictable mobility models of vehicles.
- Constrained mobility freedom because of the obligation to drive on the roads.
- Ability to retrieve location information via an external system such as GPS.
- A lack of need for complex power management techniques due to availability of abundant power supply on vehicles.
- Sufficient storage and processing capabilities.
- Variable network density in various areas and during different times of the day.

2.3 Clustering technique in VANETs

A beneficial technique to organize ad hoc networks and group the nodes into smaller segments is called clustering. Clustering is helpful in large scale distributed networks for simpler management and information aggregation of each network segment [43].

Classification of the nodes into clusters is performed according to special application requirements in order to provide a conveniently manageable network. In cluster-based routing protocols, nodes are compared to each other and the most similar nodes based on their movement patterns are selected to join the same cluster. The comparison criteria between nodes are defined based on protocol’s application requirements. Applying clustering techniques to VANET applications is beneficial and is being used widely [44].

Clustering has been mostly used for data dissemination and routing in VANETs [1, 2]. Employing cluster-based techniques for target tracking in VANET is still a challenge and has not been used frequently.

The main entities of a cluster are: cluster members (CM), cluster head (CH), and gateway nodes (GW). CH is the leader node responsible for cluster management and communication with other clusters or infrastructures in the network. CH is also responsible for relaying information between nodes in the cluster or from cluster nodes to other clusters. CMs are the nodes which join a cluster based on their features and similarities. These nodes are responsible to send their information and application-based data to CH in specific time intervals. CMs of one cluster are not supposed to communicate with CMs or CHs of other clusters. GW nodes are the shared nodes between two clusters. These nodes can contribute to the communication between two clusters.

2.4 Clustering Advantages for VANETs

In complex distributed and large scale networks, clustering is helpful for network management and data aggregation [43]. Due to VANET's special characteristics it would be effective to introduce an aggregator node responsible for data aggregation in a specific part of the network. The aggregator node may be referred to as the leader node or CH. CH’s role is to build and maintain the cluster structure for communication of application-specific data. The CH receives messages from member nodes in its area and aggregates these messages. The other nodes out of cluster area will only receive the aggregated message instead of receiving all of the messages from every node separately. This method is helpful in sending safety or hazard messages in VANETs. The vehicles around the hazard area will send messages to a leader member instead of broadcasting their messages in the entire network. The leader gathers and processes the information and communicates with other parts of the network. Clustering method helps in dividing the network into smaller segments which are easier to manage. Much research has been done on clustering techniques for VANETs [44], [45], [46], [47]. The major reasons to use clustering are: Increasing network scalability by creating network segments [48], reducing the number of messages being transmitted within the network [44], decreasing congestion in both V2V and V2I communications [48] [49], providing optimal quality of service (QoS) and applicable routing of messages [50], coping with variable network connectivity, which is caused by link breakage and density variations [51], decreasing contention and hidden terminal problems [52] . Dealing with the dynamic topology of VANETs and adapting to rapid topology changes are other important benefits of clustering in VANET environment [44]. In the process of clustering, the entire network is divided into smaller segments which are less dynamic than the global network since relative mobility between nodes in a cluster is less than relative mobility in the entire network. The aim is to choose the best appropriate nodes with more similar mobility patterns to join the same cluster [50]. As mentioned in [51], in MAC protocols, clustering helps in reducing channel contention, providing fair channel access, and increasing network capacity by controlling the topology and organizing medium access [50] [51]. As well, Using cluster-based techniques to reduce the effect of handoff latency in VANETs and to minimize packet loss caused by handoff, is proposed in [46, 53]. A Network Mobility (NEMO) based handoff scheme is introduced in [53] which is based on dividing the network into clusters and using inter-cluster communications to receive the available access points before handoff.

2.5 Cluster Stability and efficiency Features

Cluster stability is measured by various performance metrics that will be explained in this section. All of the clustering algorithms are attempting to improve these features in order to create more stable and robust clustering protocols that can function properly in VANET's highly dynamic environment and can adapt to frequent topology and density changes. The following are the main stability and efficiency features considered in most clustering algorithms. Improvement of these stability features would help to the design and implementation of an efficient and stable clustering algorithm.

Cluster head lifetime: is the time interval a cluster head is active and responsible for cluster maintenance and management. Most of the clustering algorithms try to increase the cluster-lifetime and to decrease CH changes as much as possible in order to decrease changes in the cluster structure.

Cluster member lifetime: is the interval between the times a vehicle joins the cluster as a member until it leaves the cluster. Increasing the cluster member's lifetime contributes to a more stable and robust clustering algorithm. The reason lies in the reduced number of changes in the cluster structure due to the existence long-living cluster members.

CH change number (CH change rate): is described as the number of CH changes during the simulation time [46] [52]. The CH selection criteria should be designed in a way to decrease the number of CH changes as much as possible; and yet satisfy the application requirements. A robust and stable clustering algorithm results in fewer changes in the cluster structure.

Average number of clusters: As mentioned in [52] network contention can be decreased when the number of formed clusters decreases. However, decreasing the number of clusters results in increased cluster sizes which is not always advantageous. Therefore, a trade-off should be made between the number of formed clusters and the cluster sizes.

Cluster lifetime: The definition of cluster lifetime depends on the application and design of the algorithm. For instance, in most algorithms, cluster lifetime depends on CH lifetime and if the CH is lost, the cluster structure does not exist anymore. However, losing CH in VANET's extremely dynamic environment is highly probable. Therefore, consideration of substitution techniques to assign a new CH in such scenarios without re-clustering can make considerably improve to algorithm's performance. A widely used technique is selection of a secondary CH or candidate CH to take the responsibility in case a CH is lost [45]. This method helps in improvement of CH lifetime metric and reduces delay caused by re-clustering. In Chapter 3, we will present the concept of assigning priorities to nodes for our distributed clustering algorithm (DCTT) [39]. This method contributes to CH lifetime increase and prevents re-clustering in lost CH situations.

Control overhead: overhead is caused by sending clustering control packets in the network. Control packets are necessary for cluster maintenance task and maintaining the cluster structure. In order to reduce delay and increase delivery ratio in the cluster, the overhead should be reduced. A few techniques may be used to reduce overhead, such as applying passive clustering techniques [15, 16], and prediction of member nodes behavior instead of sending their information frequently.

Convergence time: is the amount of time needed to create clusters and select a CH for each cluster. In fact, convergence time period indicates the initialization phase length. Convergence time is an essential performance metric which should be decreased to guarantee a fast and efficient clustering algorithm [17].

Packet delivery ratio: is the ratio of total number of received packets to the total number of sent packets in a cluster. This value demonstrates successful packet delivery in the network. Packet delivery ratio has been measured in many clustering algorithms as a performance metric [16]. Higher packet delivery ratio indicates better performance of the clustering algorithm.

End-to-end delay: is the average time required to deliver a packet from a source to a destination. End-to-end delay depends on various factors in the network such as network density, cluster size, communication range, and so forth. Due to frequent changes in VANET topology and structure, there is a crucial need to decrease delay. Also, vital applications of VANET such as driving safety and hazard notifications require fast delivery of messages to destination.

2.6 Clustering Stability and Efficiency Improvement in VANET

Recently a considerable research is being conducted on increasing clustering efficiency and cluster stability in VANET. Due to the dynamic nature of VANET, designing efficient clustering protocols with high cluster stability is a challenging task which requires novel ideas and techniques. The most popular methods used in many VANET clustering algorithms are categorized as the following:

- **Appropriate CH selection metric**

The CH is a crucial entity in clustering protocols which should be a long-living node and should be chosen based on application requirements. Proposing an appropriate CH selection metric can help in assigning the most eligible node as CH and increasing CH lifetime which serves towards stabilization of cluster structure. An advantageous technique for CH selection is to employ prediction of node's behavior to select a node that is an appropriate CH for a longer time period [44].

- **Appropriate cluster membership rules**

In most VANET clustering algorithms, cluster members are selected based on their relative mobility and movement direction [45], [48], [44]. Typically, in VANET clustering algorithms, the nodes moving on a different direction from the cluster are not added to it. The reason lies in the instability caused by short-time membership of these nodes. However, in some applications and under special conditions adding different direction nodes might be helpful. Likewise, it would be helpful to decrease the number of CM changes and increase CM lifetime. The concept of candidate cluster members and cluster member level is proposed in DCTT algorithm (Chapter 3) [39]. A candidate CM or a lower level member is a node which does not completely comply with CM requirements; but is highly probable to become an eligible CM in a near future due to its special characteristics. Adding these nodes to the cluster will increase the stability by decreasing cluster membership changes.

- **Reduction of CH changes**

Changing the CH requires making adjustments to cluster structure. Therefore, decreasing the number of CH changes would help in maintaining cluster structure and increasing cluster stability. In most clustering algorithms, CH is defined as the least relatively mobile node compared to all other cluster members. The CH should be evaluated at each defined time interval and re-selected if needed based on CH selection rules. Due to rapid changes in VANET topology, there is a high probability that the current CH would lose its eligibility quickly. Although another node might be more appropriate to be the CH, most algorithms do not change the CH so frequently in order to reduce the number of changes as much as possible. Adding a threshold to change the current CH is the solution that we have used in DCTT clustering protocol (Chapter 3) [39]. The threshold should be calculated carefully so as not to sacrifice the application requirements for clustering stability. The other approach to decrease the number of CH changes is to engage prediction mechanisms for CH selection. This technique selects a node which will be an eligible CH for a longer time interval compared to all other member nodes.

- **Association of nodes to cluster instead of CH**

When cluster member nodes are associated to CH, they use the CH ID and as soon as the CH changes, the cluster structure needs to be changed as well [45]. In this case the number of cluster formation (re-clustering) will increase and the cluster lifetime decreases. However, a solution to such a problem is making the cluster structure independent of CH. This method helps in increasing cluster lifetime and reducing overhead caused by running the initialization phase frequently.

- **CH Recovery Techniques**

A CH is a vital entity in a cluster. In some algorithms if the CH is lost, the cluster structure is broken and the initialization phase is required to run again. To avoid switching between cluster maintenance and initialization phases, some algorithms select a candidate CH (CCH) to take the responsibility in case of losing the current CH [45]. Candidate CH selection adds a level of stability to the algorithm and prevents delay caused by re-clustering in case the CH is lost. The other helpful method in case of losing the CH is to assign priority to member nodes. The same procedure as in CCH selection will be applied to give priority to nodes at each time interval based on the defined application metrics. The nodes are supposed to advertise their priority and inform all the member nodes about it. CMs create a member list and save the priority values of the nodes. This method is helpful in the selection of the next CH between nodes without a need for an active CH. The problem with using this method is the high overhead caused by sending beacon messages to announce the priorities. This technique helps in creating robust and stable clusters which do not solely rely on CH to continue their activities. More details on this approach is presented in our distributed clustering algorithm (Chapter 3) [39].

- **Overhead reduction technique**

Prediction-based approaches have been employed to decrease overhead caused by sending and receiving control messages for cluster maintenance in VANET algorithms. In Chapter 4, we apply a prediction-based approach to CH in order to acquire cluster members' information [40]. In this algorithm, the prediction function of CH predicts member nodes' behavior. Therefore, members do not need to send their information periodically to the CH unless they find out the predicted information do not match their actual status. This approach helps reduce the control overhead. Furthermore, the idea of passive clustering is used for reducing the clustering overhead. Passive protocols send control messages inside data packets. This concept is proposed by Gerla et. al in [15] and is used in many MANET and VANET clustering protocols e.g. [16] [2] [54].

2.7 Target Tracking in VANETs

Since vehicles are available almost everywhere, and given the rapid advancement of modern techniques for vehicles, VANETs are considered the right and proper infrastructure for various applications such as tracking and monitoring. VANETs can be used when a police agency is looking for a specific vehicle with specific visual features such as license plate, color, model, and so on. If the police agency relies solely on fixed and pre-installed security camera infrastructure across the city, there is a high probability that it would not find the target promptly, or it might even lose track of the target vehicle altogether in non-monitored areas. Therefore, camera-equipped vehicles are a future reality, and the use of communication capabilities on future vehicles would constitute the most efficient tracking system.

We define vehicle tracking as the ability to detect a target vehicle based on its visual features and continuously track the vehicle by sending position information on it to a central entity. The detection process may be based on any visual processing algorithm including license plate detection, logo, and color recognition algorithms e.g. [55-59]. However, our focus is the communication framework for continuous tracking based on ad hoc communication, which is a new topic to the best of our knowledge.

The topic of vehicle tracking has been studied mostly under localization and visual detection of moving vehicles and not as a specific VANET tracking framework. Ramos et al. [60] argue that vehicle tracking differs from tracking in traditional ad hoc networks due to various mobility models of vehicles. According to the authors, a cooperative target tracking system requires a motion model of the target, measurements of target's position, a data association model to associate measurements to the right target, and a Bayesian filer to estimate parameters of the motion model considering the measurements. The filtering task may be done by variations of the Bayesian filter such as Kalman Filter, Extended Kalman Filter (EKF), and Unscented Kalman Filter (UKF). In [60], target tracking is referred to as an estimation problem and defined as accurate and precise localization of the target. Numerous vehicle tracking researches focus on recognition of visual features of vehicles such as license plate and color [61] [62] [63] [64]. In [64] the localization challenge is defined as differences between location acquired by on-board cameras and the actual location. Calculating the precise location of vehicles (localization) has been a challenge and studied widely under the area of localization. A considerable number of researches focusing on vehicle tracking are based on positioning methods such as GPS and rely on the localization accuracy of such systems. Some of their research focus on vehicle tracking applications using smartphone's GPS and compare the functionality and accuracy of various GPS systems [65] [66]. In [67] an application based on iPhone's GPS receiver [68] is proposed. The application acquires data from GPS and sends it to a central entity for processing of traffic flow on the roads which is performed by FreeSim [69]. The authors evaluated location accuracy and reliability of data obtained from iPhone's GPS with the information received from vehicle's tracking system. Prior to 2007, most vehicle tracking systems was based on GPS and satellite transmitters, which was costly in usage and implementation [70]. However, it has been a long time since 1960's when GPS was started to deploy worldwide. Yet, GPS signal can still be unavailable in some places such as tunnels, and of course not every vehicle is equipped with GPS receiver[71]. Furthermore, in some circumstances such as tracking a stolen vehicle, it can be assumed that the tracked vehicle will have a disabled GPS.

To rectify the above-mentioned problems, we propose a cluster-based framework to continuously track a target vehicle. We use the proposed localization and visual detection techniques for VANETs. However, the focus of this thesis is on the communication framework for tracking a target vehicle cooperatively, without having access to its positioning system.

2.8 Clustering Technique for Target Tracking in VANETs

Of interest to the research work presented here is the challenge of dividing large networks such as VANET into multiple segments to improve applications performance by decreasing overhead and therefore facilitating management. Many clustering algorithms have been proposed for monitoring and tracking in WSN and MANET [72] [73] [74]. As mentioned in Section 2.2, different characteristics of MANET and WSN make their algorithms non-applicable to VANETs. The clustering structure needed for tracking a moving target vehicle differs from other cluster-based applications. As illustrated in **Error! Not a valid bookmark self-reference.**, the cluster should be formed around the target and move along with the target in order to track it continuously. Accordingly the clustering metrics and CH selection criteria would be different from other applications. For example, in cluster-based routing algorithms, the cluster is mostly formed based on movement similarity of nodes; however, in target tracking application all the metrics should be defined based on target's movement pattern. For instance, movement similarity between a node and the target should be used for cluster membership and CH selection decisions. The goal of target tracking is that the nodes around the target (which can detect the target) would be able to gain information about the target and do not lose track of the target. Thus, these nodes join a cluster which moves along with the target. The member nodes send their information about the target to the CH instead of sending it to the central entity. The CH should be a node which has the most similar movement pattern to the target to be able to track the target for the longest time interval. Therefore, all nodes should compare their movement pattern to target and the most appropriate node should be selected as CH.

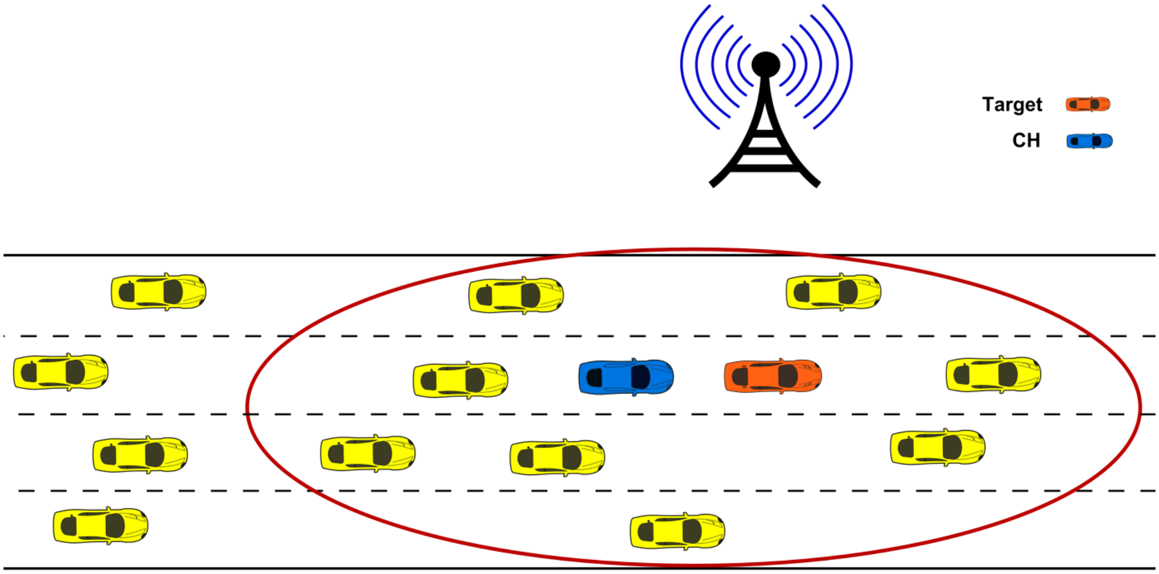


Figure 1. Clustering Technique for Target Tracking in VANET

2.9 An Introduction to VANET Clustering Algorithms

Communication between vehicles through VANETs is becoming a popular topic in research and industry. A number of research works are being carried out to improve communication techniques and create a more reliable and safe communication framework for exchanging high priority messages between vehicles. As mentioned earlier, clustering is a beneficial technique for ad hoc networks such as VANETs. Recently, numerous clustering techniques have been proposed for VANETs [44, 45, 48, 50]. Most of the proposed algorithms use vehicles' mobility features to calculate mobility metric between nodes. Mobility metric is used to make clustering decisions such as accepting nodes as cluster members or selecting a node as cluster head or candidate cluster head.

The most commonly used mobility metrics include relative velocity and distance between two vehicles. Some other protocols use relative acceleration which makes the protocol more applicable to real-world scenarios. There are other cluster membership factors such as packet transmission delay, received signal strength, and link expiration time that can be used based on protocol requirements. In this section, some of the clustering algorithms used for VANET environments are being introduced and explained. We have categorized the algorithms based on their cluster head selection criteria. In Table 1. **Characteristics of** the cluster membership rules are listed and can be a categorization feature for the algorithms. Most of the protocols use the same mobility features to compare mobile nodes. However, the calculated mobility metric and cluster membership rules and CH selection rules are among the distinguishing features of the protocols. The mobility features used by most of the algorithms to calculate their CH selection metric include distance and relative velocity. Some algorithms go further and consider acceleration in their approach, which results in more practical and applicable to real-world protocols as cited earlier. In this section we have considered these factors for categorizing the algorithms and have classified them based on their CH selection criteria as follow:

A. *Total Forces (calculated based on distance, direction and relative velocity)*

Maglaras et al. proposed a clustering algorithm for vehicular networks called spring clustering (Sp-Cl) [44]. The main idea behind Sp-Cl algorithm is to use forces as the mobility metric between nodes and the basis of cluster creation and CH selection. These forces are calculated based on relative mobility and distance between two pairs of nodes and determine whether two nodes are eligible to join the same cluster. The negativity or positivity of forces is based on the movement direction of vehicles. Two nodes apply positive force to each other if they move in the same direction and negative forces if they are driving in the opposite direction. Nodes moving in the opposite direction are not supposed to be in the same cluster. The distance, movement direction, and relative speed of nodes, are the parameters used to estimate the force between each pair of nodes. If the total forces applied to a vehicle are negative, it is not considered a candidate cluster member candidate. Negative value of total forces of a vehicle shows that all other nodes are moving away from it. The total amount of forces applied to each node along the *x*-axis and *y*-axis is used as CH selection metric. This value is referred to as "suitability value" and is calculated based on neighbor nodes' mobility and distance information. A stable node is a node with a movement pattern most similar to the nodes in its neighborhood. The most stable node in the cluster is elected as CH. In case a CM's total forces value exceeds its CH, the CM will leave its cluster and becomes a CH for the new cluster. Further, if two CHs meet each other, their clusters merge and the CH with the highest value takes over the CH duty. In order to select the most appropriate CH, a prediction-based parameter is used to evaluate the driver's behavior. As mentioned in [44] vehicles that keep a predictable movement pattern or stay at almost the same speed, are more eligible to be selected as CH. A vehicle node with more stable movement patterns may be detected by predicting its future behavior based on its previous driving patterns.

The experimental result of the Sp-Cl shows a better performance of the algorithm in comparison to Low-ID [75] method which is a MANET clustering protocol. The average number of cluster changes is calculated for different transmission ranges and various densities. The change rate increases as the transmission range decreases. However, cluster change rate per node in Sp-Cl is less than in the Low-ID algorithm. Furthermore, the average number of created clusters increases by decreasing the transmission range. Still, the average number of clusters formed in Sp-Cl algorithm is less than Low-ID. Besides, the average cluster lifetime of Sp-Cl is higher than Low-ID and is decreased when the transmission range is decreased.

B. *Velocity Difference*

In some clustering algorithms, the cluster membership metric is not calculated based on distance or relative speed between nodes, but the received signal strength, and packet delivery delay, which are useful metrics in multi-hop clustering scenarios. Ahizoune et al propose a stability based clustering algorithm for VANETs (SBCA) [45]. In SBCA, cluster membership is based on the strength of received signal from the CH. However, the CH is chosen based on velocity difference between a node and its neighbors. In this paper, the idea of selecting a secondary CH (SCH) to take over the responsibility in case of loss of the primary CH (PCH) is advanced. Selection of a secondary CH (SCH) helps in forming more stable clusters, and reduces the overhead of re-clustering in case of losing the primary CH with less overhead. The PCH selects the SCH at each time interval based on velocity and distance difference of nodes compared to PCH. A mobility prediction method based on driver's behavior is used on the PCH node to predict the time it will exit the cluster. This prediction technique helps in informing the SCH to be ready to take up the PCH role when the time comes. In this algorithm the PCH is the central entity which makes all the clustering decisions. Considerable concern in this algorithm arises when the SCH exits the cluster boundaries suddenly, or the PCH is lost before the SCH is chosen. Another beneficial feature of this algorithm is to associate member nodes with cluster instead of the CH. This feature prevents re-clustering when the CH is altered. Therefore, when the change occurs, the cluster structure remains stable and the member nodes are informed about the new selected CH. To shed more light, in some clustering algorithms, member nodes join a CH instead of a cluster. So, each time the CH changes, the cluster should be formed again. The simulation results presented by the authors show a better performance of SBCA in comparison to CCP [47]. By increasing the density in the network, average cluster lifetime is increased. However, overhead also increases as a result of increased density, which is due to more message exchange between nodes. A drawback in the design of SBCA which makes it non-applicable to real-world scenarios is a lack of rules for opposite direction vehicles because it has been assumed that all vehicles are moving in the same direction on a highway.

C. *Network Criticality (based on Link Expiration Time (LET))*

Li et al. proposed an algorithm called criticality-based algorithm (CCA) [76]. The main idea behind CCA is to use local network criticality as basic metrics for clustering. Network criticality is a global metric which demonstrates sensitivity of a network graph to topological changes in the network. It has been argued in [76] that the idea of network criticality is derived from the concept of “Random Walk Betweenness” of a node. Random walk betweenness is the total number of times a node "k" is met when information is sent from a specific source to a specific destination. The value of criticality in the network is calculated as the normalized average number of random walk betweenness of a node. The lower value of network criticality shows less sensitivity to network changes. The value of network criticality for a node pair is calculated as point-to-point network criticality which evaluates the total commute time between the node pair and illustrates the sensitivity of nodes to topology changes. Another value called localized criticality of a node is determined by considering all the paths between a node *i* and all its neighbors. Local network criticality shows robustness of a node and its suitability to be the CH. The weight matrix is required to calculate network criticality of a node pair. Therefore, link expiration time (LET) is introduced as a mobility metric which is used to assign weight to network graph. LET represents the amount of time two nodes stay connected to each other and is computed based on relative velocity and distance between two nodes. LET value is a prediction-based value calculated based on current information of nodes and assuming the same pattern for the next time intervals. As mentioned in section 2.6 prediction improves clustering performance in VANET environment, if the prediction intervals are assigned properly. The simulation results reveal that the changes on average number of clusters and average cluster size in CCA are less than MDMAC protocol. Furthermore, CH changes and member changes in CCA are less than MDMAC [17], which indicates a better performance of CCA algorithm compared to MDMAC algorithm. It is noteworthy that CCA and MDMAC are implemented as 1-hop and 2-hop algorithms. The results represent less CH and CM changes in multi-hop clusters.

D. *Spatial Dependency (based on distance, relative velocity, and relative acceleration)*

Considering acceleration as a mobility parameter in the algorithm helps in designing more realistic scenarios. The algorithms proposed in [48] and [51] consider acceleration in their mobility metric calculations. Dynamic clustering algorithm (DCA) proposed by Fan et al. in [48] takes acceleration value of nodes into account for protocol design. The mobility metric used in DCA algorithm is called Spatial Dependency (SD) which demonstrates movement similarity between two neighbor nodes. The mobility parameters used in the SD calculation are distance, relative velocity, and relative acceleration. The mobility value of each node in the cluster is calculated as the normalized total SD value of the node with all its neighbors. This value is called cluster relation (CR). A node with the highest CR value is chosen as the CH among its neighbors. The main characteristics of DCA algorithm as compared to Lowest-ID and Max Degree protocols include high cluster stability, and longer cluster head life-time when the transmission range of vehicles are increased.

E. *Fuzzy-Logic System (based on distance, relative speed and acceleration*

Hafeez et al. propose a fuzzy logic-based cluster head selection algorithm for VANETs [51]. The authors assert that some factors of VANET systems such as driver's behavior and inter vehicle distance are not predictable. Therefore they use fuzzy logic to handle this situation. The proposed algorithm is able to predict the future speed and position of vehicles using a fuzzy logic system. A learning mechanism is implemented to make more precise predictions based on the driver’s behavior. Using prediction in clustering approaches improves performance of the algorithm mostly in highly mobile scenarios such as VANETs. The most important aspect of using prediction is to decrease control messages overhead of cluster by reducing the number of required communication messages to establish and maintain cluster structure. In some cases the mobility metric is also calculated based on prediction and the decisions are made based on future behavior of nodes which is quiet beneficial in VANET's dynamic environment. In this system the membership functions of fuzzy system are defined as: inter distance, relative speed, and acceleration functions. The network model is a multi-lane one-way highway and only vehicles moving in the same direction are able to communicate with each other. A Control Channel Interval (CCI) is used as synchronization time period. Vehicles connect to control channel and send their safety messages in this period. At every CCI, vehicles receive information about their neighbors and calculate a value called "Stabilization Factor" (SF). SF is used to select the best cluster head in the cluster. The evaluation results show a better performance of proposed fuzzy-based algorithm compared to APROVE [52] and CMCP [47] in terms of average CH and CM lifetime and average cluster size. Furthermore, the impact of increasing the vehicle density in the network and increasing the prediction time interval on the protocol performance is studied in this paper. The results demonstrate improvement in the average CH lifetime, average cluster size, and average CM lifetime when vehicle density in the network is increased. This is because of the reduction of inter vehicle distance and re-election of previous CHs. Additionally, the accuracy of the algorithm degrades slightly due to the increase of the prediction time interval. The reason for low changes is the learning mechanism in the algorithm which allows the protocol to adapt to driver's behavior.

Another fuzzy-logic based clustering protocol is proposed in [77] for visual touristic guide on vehicles. This system can help tourists watch videos of touristic areas around them based on their interests. This algorithm is a multi-hop, distributed, fuzzy-logic based clustering algorithm which considers vehicles location, velocity, movement direction, and user interest as clustering metrics. A value called cluster head eligibility or CHE is calculated by their proposed fuzzy logic controller for each vehicle and is broadcasted in the network to select the most eligible CH. The CHE value is calculated by fuzzy logic controller based on the following inputs: average velocity, average distance, and average compatibility which is related to interest and is calculated based on a factor called interest vector. The performance evaluation of the protocol shows better performance in terms of CH lifetime, stability and mean number of clusters in comparison to lowest-ID protocol [75].

F. *Packet Transmission Delay*

Most of the proposed algorithms can work properly under 1-hop cluster size; however, designing multi-hop clustering protocols is challenging and requires profound scrutiny and analysis of clustering features to assure performance in large clusters (multi-hop). A multi-hop clustering approach is proposed by Zhang et al. in [46]. Packet transmission delay is used as mobility metric in this algorithm. The packet transmission delay of two consecutive beacon messages received by a vehicle from the same sender is used to show the relative mobility between two vehicles. The aggregate mobility which is the basis of CH selection is calculated by using relative mobility of vehicles. Vehicles are compared with their N-hop neighbors and the one with lowest aggregate mobility is being selected as CH. This idea helps in increasing cluster stability. The most common metrics used to calculate relative mobility between nodes in VANETs are relative speed, distance, and signal strength. As mentioned in [46], these metrics are not helpful in multi-hop clustering scenarios. The main reason is fading effects caused by obstacles between vehicles. Therefore, using packet transmission delay as clustering metric is a beneficial idea mostly in multi-hop clusters. The proposed protocol has been evaluated under two, three, and five hop scenarios on freeway mobility and Manhattan mobility models. The results show that CH duration is higher in freeway scenarios because of strong connection between vehicles and less mobility compared to city scenarios. Also, by increasing the maximum allowed speed in the network, the CH and CM lifetime in both scenarios are decreased. However, increasing the number of hops has positive effect and increases CH and CM lifetime in all scenarios.

G. Similarity Function Based on Euclidean Distance

in some VANET clustering algorithms such as [52], statistical approaches are used to calculate mobility metrics between vehicles. In this paper, a distributed mobility metric based on a statistical approach called affinity propagation is proposed in order to increase cluster stability. Cluster stability is defined as high CH and CM lifetime and lower CH change rate. The concept of affinity propagation is referred to as a clustering technique used in data mining and statistics. In this approach data points (nodes) send values to each other by messages. The transferred values include availability and responsibility of each data point. In each cluster, an exemplar is selected to be the representative of the cluster. A similarity function is defined to show suitability of a node to function as the cluster exemplar. In this algorithm, the concept of affinity propagation is applied for clustering in vehicular networks. The proposed algorithm is called Affinity PROpagation for VEhicular networks or APROVE [52]. The basic features of this algorithm include distributed function of the algorithm and stability of clusters due to using appropriate mobility metric for similarity function calculation. Besides, the idea of predicting the future position of nodes based on their current position and velocity is used in similarity function calculation of APROVE algorithm. The similarity function of a node pair is estimated based on Euclidean distance between the current position of nodes and their future position. Consideration of future distance requires using prediction based on current velocity. Another parameter used in similarity function calculation of nodes is self-similarity. The appropriate CH is selected based on similarity function of nodes. Evaluation of APROVE protocol was performed under various prediction intervals and maximum speeds. The results show that performance decreases by increasing the speed. Also, the optimal prediction interval is estimated to be 30 seconds in this algorithm which is a reasonable time interval for a very dynamic network. Furthermore, the results show superior performance of APROVE compared to MOBIC in terms of CH and CM lifetime and cluster change rate. However, MOBIC creates fewer clusters in the network in all the scenarios compared to APROVE. The problem with APROVE is the long convergence time due to the need for exchanging all the affinity messages. Also, the CH selection algorithm should run any time the timer expires, which causes high overhead.

H. First Deceleration Wins (FDW)

Cluster management in VANETs requires a large number of messages to be exchanged periodically to obtain a comprehensive knowledge of the network. It would be very helpful to reduce the number of communication messages in such a vast and dynamic network. Passive Clustering (PC) is proposed by Gerla et al. to decrease the overhead caused by exchanging periodic beacon messages to gain information about neighbor nodes and avoid cluster initialization phase [15]. The principal point of PC is to send essential clustering information in data packets. If there is no data packet ready to be delivered, the delivery of clustering information will be postponed. Wang et al. propose three different passive clustering techniques called VPCs to use for VANET routing purpose [16]. The proposed algorithms use passive cluster-based techniques for VANET environment. PC algorithm [15] uses FDW method to select the CH, in which the first ready node to be the CH, is selected as CH. VPC algorithms use the same technique to elect the first CH in the cluster-formation phase. However, the random selection of CH and GW nodes is combined with some weight based methods to assign priority to nodes. The distinction point of the three proposed algorithms is the CH election metric i.e. vehicles density, link quality and link sustainability respectively used in VPC1, VPC2, and VPC3. Vehicle density is calculated by counting the number of reply messages each node receives from its neighbors after sending an advertisement message and is used in VPC1 algorithm. A node with more neighbors is suitable to be the CH. The link quality metric which is used in VPC2 algorithm is represented as reliability level of links. Expected Transmission Count (ETX) is used to show reliability and high quality of links and indicates the bi-directional transmission quality of a link. The other metric used for VPC3 is called link sustainability. The connection time between two vehicles is used in order to evaluate sustainability of a routing path. This metric is called "Link Expiration Time" or LET and is calculated based on relative speed and distance between vehicles. LET is considered a prediction-based metric because it relies on the current status of nodes and determines the future behavior to make clustering decisions.

I. *Connectivity Degree (based on distance and relative speed)*

Rawshdeh et al. propose a Threshold Based (TB) clustering algorithm in [50]. In TB, identification of candidate cluster members is made by using the degree of speed difference. The position information of vehicles is sent in periodic messages. Each node calculates its nodal degree, which is the number of r-neighbors. The neighbor nodes are classified into stable neighbors (SN) and unstable neighbors (UN). SNs are supposed to be candidate cluster members. Candidate cluster members move in the same direction and have more similar speed. The probability density function for speed of each vehicle is estimated to find the probability that relative speed of two vehicles are in a defined threshold or not. The nodes which maintain their relative speed in the threshold are assumed to be appropriate candidate cluster members. The suitability function is used to verify eligibility of a node to be CH. To calculate the suitability function, a parameter called connectivity degree should be defined. The nodes with closer distance to their neighbors and closer relative speed to average speed of neighbors are supposed to have higher connectivity degree and are more probable to become CH.

J. *Node ID as weight value*

Modified DMAC (distributed and mobility-adaptive clustering) protocol is proposed in [17] to make DMAC protocol appropriate for VANET environment. Distributed clustering for ad hoc networks (DMAC) [78] is a general clustering protocol for mobile environments and this feature makes it less beneficial for VANET’s highly dynamic nature. Specific features of MDMAC algorithm are mentioned as: avoiding to add nodes with short connectivity time to the cluster, avoiding to add opposite direction nodes compared to cluster's movement direction. The proposed algorithm uses the idea of weight based clustering in which the weights of nodes are assigned based on their ID and node connectivity. Node connectivity is represented as the number of neighbors of each vehicle node. The cluster membership rule of MDMAC is based on prediction of connection time of nodes. This value is referred to as freshness and is an estimated value based on the current distance and velocity of nodes. MDMAC algorithm contradicts with some of the DMAC algorithm properties as cited in [17]. MDMAC is a multi-hop clustering algorithm and nodes can be n-hops far from CH. MDMAC helps in creating more stable clusters with fewer changes compared to DMAC. However, the overhead of MDMAC is higher due to its connectivity time estimation property, which requires more messages passing between nodes.

2.10 MANET Clustering Algorithms

The main approaches used in VANET clustering algorithms are derived from MANET protocols. As explained in Section 2.2, MANET protocols are not appropriate to be used in VANET environment due to their different characteristics and features. However, adjusting MANET algorithms and considering VANAET's characteristics in the design procedure can be used as methods to implement clustering algorithms suitable for VANET. Some of the most popular MANET clustering protocols include MOBIC [43] and Lowest-ID [75]. In this chapter some of the most popular MANET clustering algorithms have been reviewed briefly.

Lowest-ID is a maximum two-hop clustering algorithm proposed by Gerla et al. for mobile ad hoc networks [75]. This protocol is a simple clustering approach which uses the ID of nodes as the only clustering metric. Lowest-ID does not consider mobility of a vehicle in CH selection decisions. Nodes are supposed to broadcast messages to their neighbors in order to exchange clustering information. A node with lowest ID among all its neighbors is selected as CH. The CH only receives messages from nodes which have higher ID than itself. Any node which receives messages from more than one CH is a gateway (GW) node and other nodes are ordinary members.

MOBIC extends the concept of MANET clustering by considering the idea of relative mobility between nodes [43]. The main idea behind MOBIC is to compare nodes with their neighbors based on their mobility metrics and to add them to appropriate clusters. A node with lowest relative mobility compared with its neighbors is selected as CH. A CH with high relative mobility compared to its neighbors results in poor cluster stability. The mobility metric proposed in MOBIC does not require location information about nodes.

Relative mobility is calculated based on received signal strength of two consecutive messages from the same neighbor node. MOBIC is a weight based and one-hop clustering protocol. The clustering scheme used for MOBIC is similar to Lowest-ID algorithm [75]. A notable property of MOBIC includes the merging process of two clusters. When two CHs meet, the merging time is postponed for CCI time interval. The CCI or cluster contention interval is introduced as a waiting time for cluster merging process. After this waiting time if two CHs are still in each other's range, their clusters are supposed to merge and the one with lowest ID takes over the CH responsibility. The evaluation results represent a better performance of MOBIC in terms of CH changes because of using relative mobility instead of node ID.

As mentioned earlier, passive clustering is an advantageous technique to reduce control overhead in clustering algorithms. There exists a considerable number of passive clustering algorithms proposed for wireless ad hoc networks such as MANETs including FWD [15], GRIDS [79], EFPC [80], EAPC [81], PCBRP [82], and KHPCBRP [83].

The idea of passive clustering for wireless ad hoc networks was proposed by Gerla et al. in [15]. Cluster stability and faster convergence are the benefits of PC algorithm. A novel CH selection technique called first declaration wins (FDW) is proposed in [15]. FDW suggests selection of the first ready node as CH instead of using weight based methods. The network activity and clustering state of a node represents its readiness as a CH. The selected CH might not be the best eligible CH based on application requirements; but, it is selected faster than weight based methods. However, the CH lifetime, which is an important stability metric, can be affected adversely.

GRIDS [79] is an energy-aware passive clustering protocol which uses periodic polling and geographical repulsion. The CH and Gateway (GW) nodes selection criteria depend on energy levels of nodes. The CH nodes do not change frequently unless there is a CH collision which is entering the 1-hop neighborhood of another CH.

Rangaswamy et al. proposed a passive clustering algorithm for MANETs which is called PCBRP [82]. PCBRP is a multi-hop (max 2-hops) algorithm and the cluster formation is based on node proximity. The clusters consist of three node states including CH, GW, and ordinary nodes. The ordinary nodes are not supposed to broadcast any messages and the CH and GW nodes are the critical cluster nodes. Among various nodes competing for CH state, a node with lowest ID takes the responsibility.

A multi-hop passive clustering algorithm for MANET environment called KHPCBRP is proposed in [83]. This algorithm is based on CBRP [54] and the simulation results show better performance of KHPCBRP in comparison to CBRP in terms of overhead. The algorithm has been tested under 2-hop and 3-hop scenarios and in both cases the overhead is reduced. The concept of prepared CH (PCH) is proposed to reduce re-clustering overhead by replacing the current CH with a more eligible node. The FDW rule is used to select the CH. Given the fact that clustering procedure is an on-demand process and the data messages are used for clustering, the overhead is reduced considerably and the clustering is done faster. Also, because of creating large clusters with multi-hop clustering approach, re-clustering is reduced, resulting in higher cluster stability.

Table 1. Characteristics of Cluster-Based VANET Algorithms

Protocol	CH selection metric	Clustering metric	Stability features	Other features	Cluster size	Simulation Environment
SP-CI [44]	Total Forces (Distance, Direction, relative speed)	Force based (Distance, Direction, relative speed)	The lowest mobile and most predictable nodes become CH Same direction nodes join cluster	Distributed	-	Highway Direction
DCA [48]	Spatial Dependency (SD) (Distance, relative velocity, relative acceleration)	Spatial Dependency (SD) (Distance, relative velocity, relative acceleration)	Same direction nodes join cluster	Distributed No prediction	-	-
SBCA [45]	PCH: velocity difference SCH: Distance, relative speed	Received signal strength of two consecutive beacon messages	Secondary CH, Prediction of CH lifetime	Centralized Prediction of expiration time of PCH	-	Highway (4 lane) All vehicles are same direction
Fuzzy-Logic [51]	Fuzzy logic rules Distance, speed, acceleration	No clustering metric mentioned	Prediction of speed and position	Prediction-based CH selection	-	highway (one directional, 4 lane)
Multi-hop [46]	Aggregate relative mobility based on transmission delay	Relative mobility based on Transmission delay of 2 consecutive beacon messages	Using transmission delay to overcome fading effect in multi-hop scenarios	Distributed	Multi-hop	Freeway mobility and Manhattan mobility model
APROVE [52]	Affinity Propagation Messages	Similarity Function based on current and future Euclidean Distance between nodes	Distance prediction	Distributed	1-hop	Highway

CCA [76]		Localized network criticality of a nodes	Node pair network criticality	Prediction-based calculation of LET		1-hop and 2-hop	-
VPC [16]	VPC1	Vehicle density	-	Passive clustering to reduce overhead, prediction based metric (LET) Combination of FDW and weight based metric to assign priority	Distributed Prediction based LET metric (VPC3)	-	Highway (one way, multi-lane)
	VPC2	Link quality (ETX, bi-transmission quality of a link)	-				
	VPC3	Link sustainability (LET, link expiration time)	-				
TB [50]		Suitability value (Si) based on average distance from neighbors and speed difference with neighbors	Relative speed less than a threshold and is in a specified range	Relative speed threshold	Distributed Weight based algorithm (WB), and TB with different relative speed thresholds	2-hop	Multi-lane highway
MDMAC [17]		Weight-based (node ID, and node connectivity or number of neighbors)	Freshness value: estimation of connection time	Prediction-based CM selection metric Same direction nodes join cluster	Distributed Direction-based Prediction-based (cluster membership rules)	Multi-hop	Multi-lane highway
Fuzzy-Logic II [77]		CHE value (fuzzy controller output based on average velocity, distance, and compatibility)	Location, direction, velocity, and passenger interest	Same direction nodes join cluster	Distributed Direction-based	Multi-hop	2 and 4 lane highway implementation

DCTT [39]	TFP (Tracking Failure Probability) based on relative velocity and distance	Target detection and distance from the target	Cluster member level TFP threshold Same direction nodes join cluster	Distributed Direction-based	Multi-hop	Multi-lane highway
PCTT [40]	OBT (Observation Time)	Target detection and distance from the target	Prediction-based CH selection metric Prediction-based cluster maintenance Same direction nodes join cluster Cluster member level Resign Timer to increase CH lifetime Candidate cluster head (CCH) selection	Centralized Direction-based	Multi-hop	Multi-lane highway

3. **Proposed DCTT Algorithm:** A Distributed Cluster-based Algorithm for Target Tracking in Vehicular Ad Hoc Networks

3.1 Assumptions and Definitions

The proposed Distributed Cluster-based Algorithm for Target Tracking (DCTT) clustering algorithm is designed for the purpose of vehicle tracking in VANETs. This algorithm assumes that vehicles have front and rear cameras and can detect visual features of a target such as license plate information and color. Localization of the target is performed by visual processing. In this algorithm, a central entity such as a police station is seeking help to find a specific target and receive its visual and location information periodically. This entity is called Command and Control Centre (CC) and is a node located in multi-hop communication distance from the target. The CC broadcasts the target's information in the network with the purpose of informing vehicles about target's existence. The DCTT algorithm is designed to help in building a cluster, with the cluster head responsible for collecting target's information from all vehicles that can detect the target, aggregating the information, and forwarding the information to the CC. It is noteworthy to mention that we are not sending the actual video information in our simulations. Table 2. **DCTT Term Definitions**defines the terms used in this algorithm.

Table 2. DCTT Term Definitions

CH	Cluster Head
CM	Cluster Member
CC	Control Center
CCM	Control Center Message
CMM	Cluster Member Message
CHM	Cluster Head Message
TDV	Target Detection Value
TFP	Tracking Failure Probability
OBN	Observer Nodes
LP	License Plate
ML	Member List

3.2 Tracking Failure Probability (TFP) as CH Selection Metric

The proposed algorithm assumes all vehicles are aware of their location and velocity using GPS devices. The location of the target is unknown since we assume there is no access to its GPS information. Each vehicle calculates its distance from the target by visual processing. A considerable research is been done on visual distance calculation that can be used in this algorithm to acquire target's distance [84-87]. To acquire coordinates of the target, the DCTT algorithm relies on digital map and the calculated distance. The coordinates and the distance information are used to find velocity of the target at any time.

Tracking Failure Probability (TFP) is a mobility metric which represents movement similarity of a node relatively to the target. In order to calculate TFP between a vehicle *C* and the target vehicle *T* at time *t*, it is required to have the distance between node *C* and *T* and their velocity vectors at that time. Assume that D_{CT_t} is the distance between node *C* and target at time *t*. We define a value called Valid Distance Range (VDR), which is used to normalize the distance between any node and the target. This range is the farthest acceptable distance from CH that depends on the communication range of nodes and the number of allowed hops in the cluster. The normalized distance is calculated as follow:

$$(1)\ D_{C_{Nt}} = \frac{D_{CT_t}}{VDR}$$

We are interested in the velocity vector of vehicles rather than their speed value. Velocity vector shows the movement direction of a vehicle along with its velocity. In this way we can differentiate between nodes moving in the same and opposite directions. The angle θ is the velocity vector angle between vehicle *C* and the target vehicle. If vehicles *C* and *T* move in the same direction, the velocity vector angle between them will be zero degree and if they move in opposite direction, θ it will be 180 degrees. The velocity vector $\bar{V}_{C_t}$ is defined as:

$$(2)\ \bar{V}_{C_t} = V_{C_t} \cos\theta$$

To find the normalized value of velocity vectors, we need to define a value called Valid Velocity Range (VVR). VVR is the difference between minimum and maximum allowed speed in the network. The values $\bar{V}_{C_{Nt}}$ and $\bar{V}_{T_{Nt}}$ are normalized velocity vectors of vehicle *C* and target *T* respectively.

(3) $\bar{V}_{C_{Nt}} = \frac{\bar{V}_{C_t}}{VVR}$

(4) $\bar{V}_{T_{Nt}} = \frac{\bar{V}_{T_t}}{VVR}$

Two values α and β are defined as Distance and speed Efficiency Factors. These values are coefficients of distance and velocity to control efficiency of these metrics for each vehicle. We assume the effects of velocity and distance are the same on TFP calculations. Therefore, the value of α and β are assumed to be equal.

(5) $\alpha = \beta = 0.5$

In the following formula, the TFP value of node *C* at time *t* has been represented as $TFP\ (C)_t$. A node's TFP value indicates its eligibility to become the CH. A node with lowest TFP value is selected as the CH.

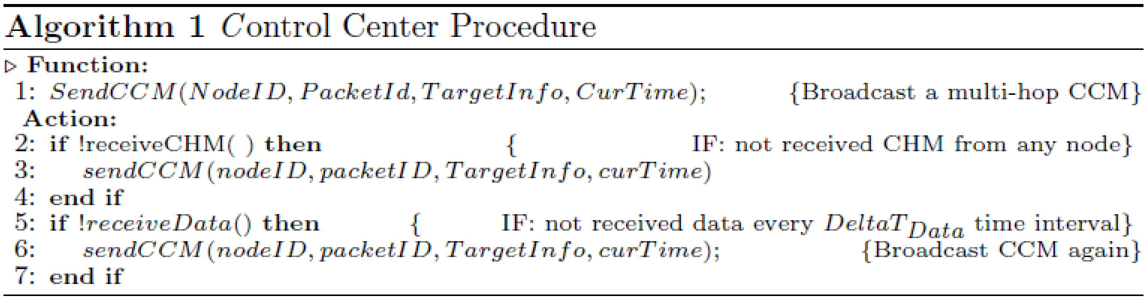
(6) $\boldsymbol{TFP\ (C)_t = 100 * (\alpha D_{C_{Nt}} + \beta \left|\bar{V}_{C_{Nt}} - \bar{V}_{T_{Nt}}\right|)}$

3.3 Algorithm Description

The DCTT algorithm is divided into three phases: *initialization*, *cluster maintenance*, and *tracking*. In the initialization phase, the cluster is created and the initial cluster head is selected. In the processing phase, each node (including CH and CMs) performs its different tasks for cluster maintenance, and in the tracking phase, the target is tracked

3.3.1 Control Center Functions

The Control Center (CC) broadcasts a "Control Center Message" (CCM) to the entire network with the target vehicle's information such as license plate, color, and other features and then waits to receive response messages from vehicles. When CC receives a response message from any vehicle that has detected the target, it stops broadcasting and waits for the target’s information. The CC may also send the CCM to specific areas in the network if it has a rough idea about the location of the target. At any point later, if the CC stops receiving information from the CH regarding the specified target (after a pre-defined time interval) it will assume the cluster no longer exists and will start broadcasting the target’s information again in the network. The control center procedure is described in Algorithm 1.



Algorithm 1 Control Center Procedure

```
▷ Function:
$SendCCM(NodeID, PacketId, TargetInfo, CurTime)$;        {Broadcast a multi-hop CCM}
Action:
if !receiveCHM( ) then        {        IF: not received CHM from any node}
    $sendCCM(nodeID, packetID, TargetInfo, curTime)$
end if
if !$receiveData()$ then        {        IF: not received data every $DeltaT_{Data}$ time interval}
    $sendCCM(nodeID, packetID, TargetInfo, curTime)$;        {Broadcast CCM again}
end if
```

3.3.2 Initialization Phase

Any vehicle that receives a CCM from the CC and which can detect the target responds to CC and starts the initialization process (Algorithm 2).

We have defined a flag called *Target Detection Value (TDV)*. Any node that can detect the target sets its TDV to true. The vehicles that detect the target are referred to as “Observer Nodes (OBN)”. The OBNs start broadcasting Cluster Member Message (*CMM)* and receive response messages from their N-hop neighbors. OBNs check the TDV field in the response messages. If TDV in the message is set to true, the sender node will be added to a list called “Member List (*ML*)” with the *TDV* equal to a true value. If *TDV* field is not true, and the neighboring node cannot detect the target but it is in the communication range of OBNs, it is also added to *ML* with a false *TDV* field. OBNs calculate their Tracking Failure Probability (*TFP)* based on the formula cited in Section 3.2. *TFP* displays which vehicle has a closer movement pattern to the target and is more appropriate to be the cluster head. In this algorithm, the cluster is moving with the target to be able to track it continuously; thus, it would be more efficient to choose a node with more similar movement pattern to the target as cluster head.

A critical issue considered in this phase is that there may not be any other vehicle to respond to the first vehicle which has detected the target because there is no other vehicle in its communication range. This may happen in sparse areas such as suburban roads. In such a case, the first vehicle which is an Observer Node is responsible for keeping the location history of the target as long as the target is moving in its Field of View (*FOV*), and send it to the CC.

In DCTT algorithm, cluster members are divided into two groups. The first group is *OBN*s that are level-1 cluster members (CM-L1). OBNs contribute to the tracking task as they can detect the target. The second group is level-2 members (CM-L2). These nodes are not able to detect the target at current time; but, are highly probable to observe the target in a near future. In Figure 2, vehicle C is not able to detect the target at current time. However, if the target moves faster, vehicle C will be able to see the target at time t_2. Besides, if the target moves slower, it will enter the rear FOV of vehicle C at time t_1. We argue that adding both groups of nodes to the cluster as cluster members would prevent re-clustering and increase cluster stability.

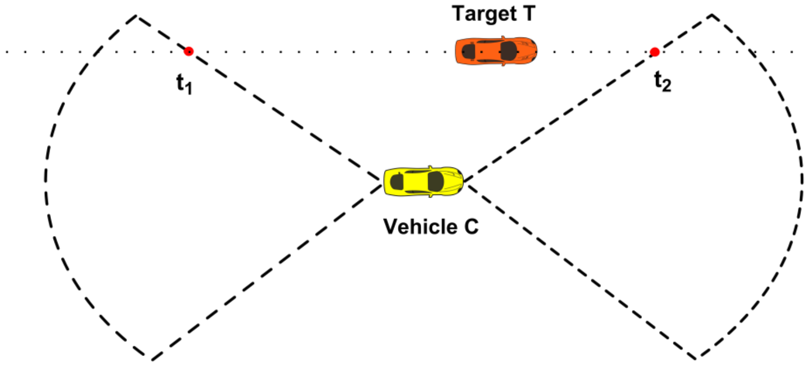


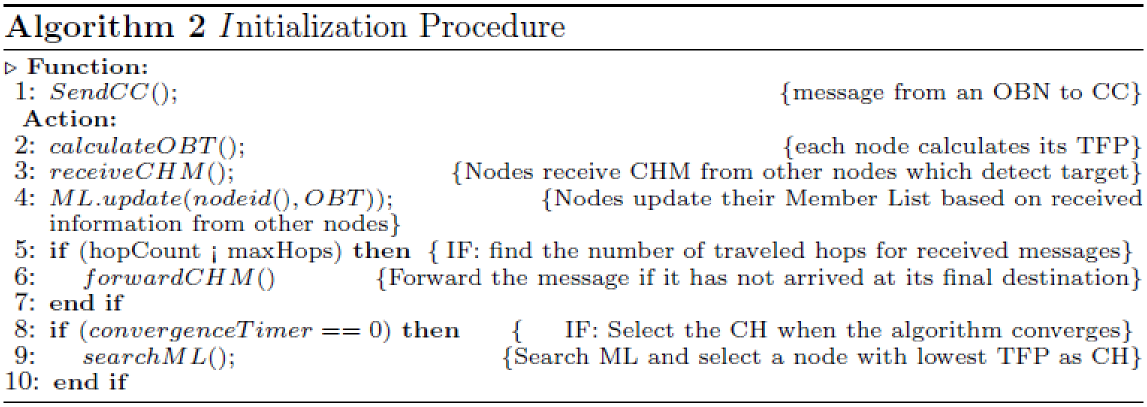
Figure 2. Cluster Member Level

These two groups have different tasks due to their different characteristics. Level-2 members may rapidly transform to level-1 members and vice versa. If we only add *OBN*s to the cluster, the speed of changes will be very high since nodes swap places between these 2 levels quite frequently.

An important point to be considered in the design of our algorithm is to connect nodes to the cluster instead of linking them to the cluster head. As a result, there would be no need to alter the membership of all nodes in the case of changing or losing CH. Also, this idea would help avoid switching to the initialization phase all over again every time the CH changes. This same concept has been used in design of SBCA algorithm [45] in order to create more stable clusters and decrease overhead.

Both level-1 and level-2 nodes join the cluster as members (CM-L1 and CM-L2). Member nodes keep and update the member list when they receive information from other nodes. CMs are supposed to calculate their *TFP* and send it to other members. The node with the lowest *TFP* becomes the cluster head. *TFP* keeps updating as the nodes move. The CH will change according to changes of *TFP* during the maintenance phase.

After the initialization phase, the initial cluster is created and the CH is selected. The initialization phase may be repeated only if there are no cluster members available and the cluster is decommissioned. The purpose of our design is to avoid switching to the initialization phase from the cluster maintenance phase frequently. This goal can be attained only if the clusters are sufficiently stable and re-clustering is not done repeatedly. For example, there should be a recovery mechanism in place in case the CH is lost so as to ensure stability of the cluster structure. This is a crucial prerequisite in designing a robust clustering algorithm.



Algorithm 2 *I*nitialization Procedure

```
▷ Function:
$SendCC()$;                                   {message from an OBN to CC}
Action:
$calculateOBT()$;                             {each node calculates its TFP}
$receiveCHM()$;                               {Nodes receive CHM from other nodes which detect target}
$ML.update(nodeid(), OBT)$;                   {Nodes update their Member List based on received information from other nodes}
if (hopCount ¡ maxHops) then                  { IF: find the number of traveled hops for received messages}
    $forwardCHM()$                            {Forward the message if it has not arrived at its final destination}
end if
if ($convergenceTimer == 0$) then             {    IF: Select the CH when the algorithm converges}
    $searchML()$;                             {Search ML and select a node with lowest TFP as CH}
end if
```

3.3.3 Cluster Maintenance Phase

This phase is divided into CH functions and CM's functions as described in the next subsections.

A. <u>Cluster Head Functions</u>

The initial CH is selected by CMs in the initialization phase. Thus, there is no need for a CH to announce itself. CH is responsible for managing the cluster by sending messages at every ΔT time intervals to find new cluster members and add them to its member list (ML). The member list of the CH is updated by the information received from the new and current members. The TFP of all nodes is saved in the ML. Furthermore, the cluster head calculates its own TFP every ΔT_{TFP} time interval, and compares it with other values in its ML. This comparison helps the CH to check if it is still a valid CH or should quit and hand over the responsibility. A "Safe Threshold" to change the CH is defined because the *TFP*s are changing very quickly and we do not want to change the CH too frequently. Therefore, a CM will become the CH only if its TFP value is lower than the current CH's TFP value with a safe threshold. Changing the CH requires every member vehicle to update its information about the cluster. Therefore, we try to reduce the cluster changes as much as possible by choosing the most appropriate long-lasting CH. Besides, we aim to increase CH's lifetime by defining a safe threshold and using it in TFP comparison function as represented in Algorithm 3.

CH is responsible for managing the cluster by adding new members and removing the old members which are no longer eligible cluster members. In order to do so, CH sends CHM to member nodes periodically and receives CMM from them in response. It is important to note that here vehicles moving in the opposite direction of the target are excluded from the cluster because these nodes would be unstable cluster members, thus decreasing cluster stability. By using velocity vector rather than only speed value in our formula, we address direction when calculating TFP. When two vehicles are moving in opposite directions, TFP moves beyond the acceptable range to join the cluster. Therefore, in normal conditions, they are not added to the cluster. However, in some cases such as in sparse areas where the number of cluster members are less than one node, opposite direction nodes would also join the cluster, in order to acquire information about the target. The idea is to design an algorithm that is capable of adapting to different conditions while maintaining cluster stability.

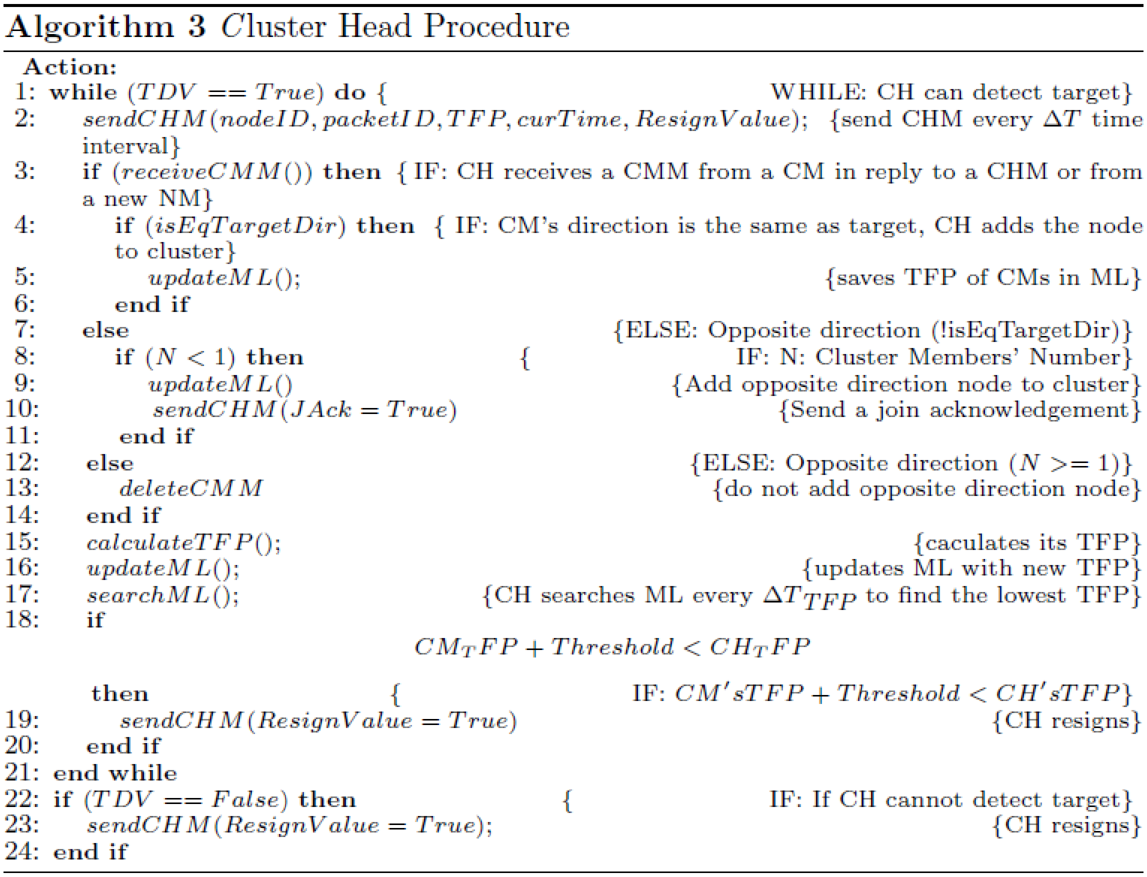
At any point that CH is not able to detect the target or its TFP value is higher than other nodes with a safe threshold, it broadcasts CHM with the “Resign Field” set to true and sends integrated information to CC (the data which was not sent previously). However, in some cases it is impossible for CH to send “Resign Message” to members because of not having any connection with the cluster. We call this situation “Lost CH”. In the case of losing cluster head, a node with highest TFP value will be chosen to take the cluster head’s responsibility. This value is exchanged between nodes in the messages they send to each other in the form of multi-hop broadcast messages. Therefore, all nodes know about TFP value of the other members. In this condition, if the current CH is lost without any notice, there will be no need to go to the initialization phase and restart the algorithm. A node with the lowest TFP takes the responsibility and becomes the new CH.

Our algorithm is robust to lost CH scenarios and works properly under these conditions. This is due to the fact that all the member nodes know about the latest TFPs of other nodes and can choose the best node as CH without being forced to start the initialization phase again. Here, the cluster structure is not broken in case of losing CH and thus there is no need to cause delay by stopping tracking and running the initialization phase to find a new CH. In [45] the authors propose to use a “Candidate CH” for further stability. In DCTT algorithm we assume that candidate cluster heads might be exposed to lost CH scenarios and therefore it would be more reliable to consider a range of options rather than only one choice. As well, it is possible that the CH is lost before being able to choose a candidate cluster head.

B. *Cluster Members Functions*

In DCTT algorithm we define two categories of cluster members. The first category is level-1 members which we refer to as OBN, and the second category are level-2 members.

As mentioned before, OBNs are responsible for tracking the target continuously and sending its information (such as location information) to CH at defined time intervals. Also, OBNs should calculate their TFP value repeatedly at a defined time interval (ΔT_{TFP}) and send it to all CMs. CMs are supposed to receive CHM at every defined time intervals or “Resign Message” from CH. In case a CM does not receive any of these messages, it may be because the member has gone out of cluster boundaries or the CH is lost.

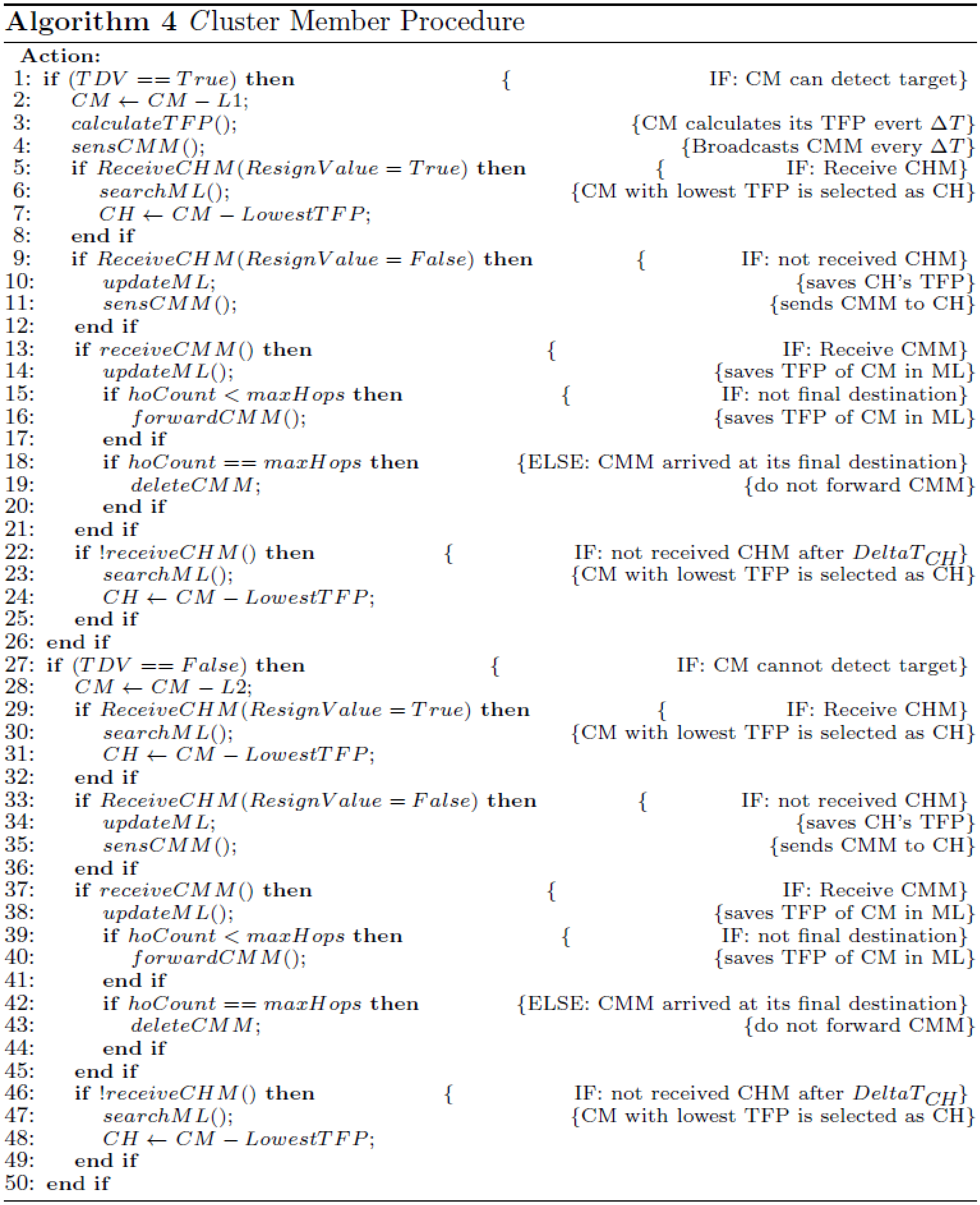


Algorithm 3 *C*luster Head Procedure

```
Action:
while ($TDV == True$) do {                                  WHILE: CH can detect target}
    $sendCHM(nodeID, packetID, TFP, curTime, ResignValue)$;  {send CHM every $\Delta T$ time interval}
    if ($receiveCMM()$) then  {IF: CH receives a CMM from a CM in reply to a CHM or from a new NM}
        if ($isEqTargetDir$) then  { IF: CM's direction is the same as target, CH adds the node to cluster}
            $updateML()$;                                   {saves TFP of CMs in ML}
        end if
    else                                          {ELSE: Opposite direction (!isEqTargetDir)}
        if ($N < 1$) then              {            IF: N: Cluster Members' Number}
            $updateML()$                          {Add opposite direction node to cluster}
            $sendCHM(JAck = True)$                          {Send a join acknowledgement}
        end if
    else                                             {ELSE: Opposite direction ($N >= 1$)}
        $deleteCMM$                                    {do not add opposite direction node}
    end if
    $calculateTFP()$;                                                {caculates its TFP}
    $updateML()$;                                              {updates ML with new TFP}
    $searchML()$;              {CH searches ML every $\Delta T_{TFP}$ to find the lowest TFP}
    if
                    $CM_TFP + Threshold < CH_TFP$
    then                  {              IF: $CM'sTFP + Threshold < CH'sTFP$}
        $sendCHM(ResignValue = True)$                                  {CH resigns}
    end if
end while
if ($TDV == False$) then            {                   IF: If CH cannot detect target}
    $sendCHM(ResignValue = True)$;                                     {CH resigns}
end if
```

If the CH is lost without any notice, all CMs check their ML and find a node with the highest priority and select it as CH. All members send a request message to the new CH in the form of CMM as a confirmation. Algorithm 4 shows the cluster members functions.

As mentioned in Control Center Procedure, CC will check its updates from CH and if there was a problem and it did not receive updates, we assumes the cluster does not exist anymore and the algorithm goes to the initialization phase again.

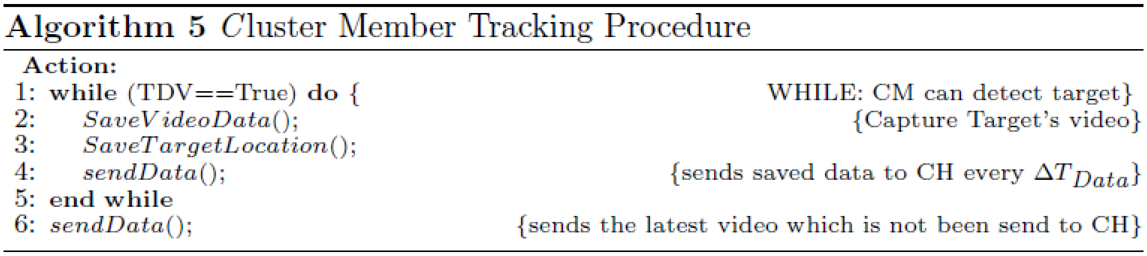


Algorithm 4 *C*luster Member Procedure

```
Action:
if ($TDV == True$) then    {IF: CM can detect target}
    $CM \leftarrow CM - L1$;
    $calculateTFP()$;    {CM calculates its TFP evert $\Delta T$}
    $sensCMM()$;    {Broadcasts CMM every $\Delta T$}
    if $ReceiveCHM(ResignValue = True)$ then    {IF: Receive CHM}
        $searchML()$;    {CM with lowest TFP is selected as CH}
        $CH \leftarrow CM - LowestTFP$;
    end if
    if $ReceiveCHM(ResignValue = False)$ then    {IF: not received CHM}
        $updateML$;    {saves CH's TFP}
        $sensCMM()$;    {sends CMM to CH}
    end if
    if $receiveCMM()$ then    {IF: Receive CMM}
        $updateML()$;    {saves TFP of CM in ML}
        if $hoCount < maxHops$ then    {IF: not final destination}
            $forwardCMM()$;    {saves TFP of CM in ML}
        end if
        if $hoCount == maxHops$ then    {ELSE: CMM arrived at its final destination}
            $deleteCMM$;    {do not forward CMM}
        end if
    end if
    if $!receiveCHM()$ then    {IF: not received CHM after $DeltaT_{CH}$}
        $searchML()$;    {CM with lowest TFP is selected as CH}
        $CH \leftarrow CM - LowestTFP$;
    end if
end if
if ($TDV == False$) then    {IF: CM cannot detect target}
    $CM \leftarrow CM - L2$;
    if $ReceiveCHM(ResignValue = True)$ then    {IF: Receive CHM}
        $searchML()$;    {CM with lowest TFP is selected as CH}
        $CH \leftarrow CM - LowestTFP$;
    end if
    if $ReceiveCHM(ResignValue = False)$ then    {IF: not received CHM}
        $updateML$;    {saves CH's TFP}
        $sensCMM()$;    {sends CMM to CH}
    end if
    if $receiveCMM()$ then    {IF: Receive CMM}
        $updateML()$;    {saves TFP of CM in ML}
        if $hoCount < maxHops$ then    {IF: not final destination}
            $forwardCMM()$;    {saves TFP of CM in ML}
        end if
        if $hoCount == maxHops$ then    {ELSE: CMM arrived at its final destination}
            $deleteCMM$;    {do not forward CMM}
        end if
    end if
    if $!receiveCHM()$ then    {IF: not received CHM after $DeltaT_{CH}$}
        $searchML()$;    {CM with lowest TFP is selected as CH}
        $CH \leftarrow CM - LowestTFP$;
    end if
end if
```

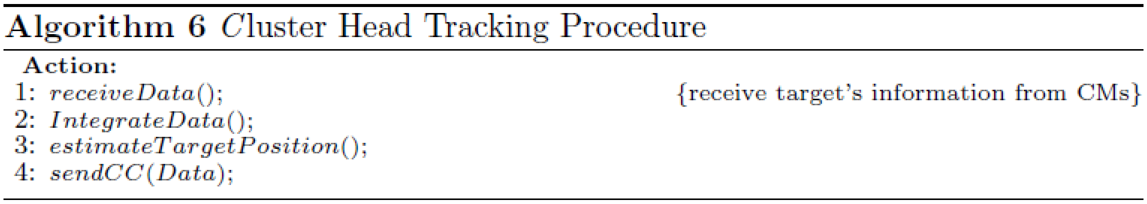
3.3.4 Tracking Phase

In this algorithm, tracking is done by all OBNs and the CH. Tracking includes taking continuous visual and location information of the target and sending this information to the CC in specified time intervals (ΔT_{Data}). CMs send target's information to CH and they are not responsible for sending this information directly to CC. After the initialization phase, CH should integrate all the information received from other nodes about the target and send it to the CC. This phase includes two procedures related to CMs and CH. The tracking functions of CMs and CH are illustrated in algorithms 5 and 6 respectively.



Algorithm 5 *C*luster Member Tracking Procedure

```
Action:
while (TDV==True) do {                          WHILE: CM can detect target}
    SaveVideoData();                            {Capture Target's video}
    SaveTargetLocation();
    sendData();                                 {sends saved data to CH every $\Delta T_{Data}$}
end while
sendData();                                     {sends the latest video which is not been send to CH}
```



Algorithm 6 *C*luster Head Tracking Procedure

```
Action:
receiveData();                                  {receive target's information from CMs}
IntegrateData();
estimateTargetPosition();
sendCC(Data);
```

4. **Proposed PCTT:** A Prediction Based Clustering Algorithm for Target Tracking in Vehicular Ad-Hoc Networks

4.1 Algorithm Characteristics and Features

In this section we provide a quick review of special characteristics and features of Prediction Based Clustering Algorithm for Target Tracking (PCTT) algorithm and compare its features with DCTT protocol.

4.1.1 Centralized vs. Distributed

The PCTT algorithm is a centralized ad-hoc clustering algorithm. CH is the central entity which is in charge of cluster management and tracking. Such maintenance decisions as calculating the CH selection metric, selecting the best CH at each time, and granting permission to join, are performed by the CH. The list of all members in the cluster are also kept and updated by the CH and there is no need for member nodes to keep any member list or make managerial decisions.

The pros and cons of such systems should be taken into consideration in order to ensure functionality and optimal performance. One of the main concerns about centralized systems is the huge processing overhead and abundant resource requirements for the central processing entity. Although this is true in some ad-hoc networks such as wireless sensor networks (WSN), and mobile ad-hoc networks (MANET), it is not an issue in vehicular ad-hoc networks. The reason is availability of ample processing and power resources on vehicles which makes VANET systems unique in comparison to other ad-hoc networks. One of the major advantages of having a central management entity in VANETs is reducing network overhead by decreasing the number of messages required to be sent between vehicles in order to transfer critical information. In distributed systems where the network is designed without any central entity, all the nodes are required to broadcast messages in the network so as to transfer information. But in centralized networks, the nodes are only required to send their vital information to the central entity at specified time intervals instead of flooding them into the entire network regularly.

Also, devising mechanisms such as prediction functions in the central entity can reduce the number of required messages. These methods may reduce the bandwidth requirement and decrease overhead in the network.

However, a central node is a single point of failure, and it cannot be solely relied in crucial applications. For instance, in our tracking algorithm, the central entity, which is the CH, is supposed to collect all the targets information from member nodes and process this information before sending it to the central entity. Therefore, if the CH is lost without notice, an important part of the information will be lost as well and cannot be retrieved easily. Also, the algorithm has to switch to the initialization phase and start over again. Therefore, we need to devise a technique to help in such situations. In DCTT algorithm (Chapter 3), this problem did not exist because of DCTT's distributed structure. The concept we apply in PCTT algorithm is considering a candidate cluster heads to take responsibility in the case of losing the current CH. The method of choosing and handing over the responsibility is described in the algorithm description section (Section 4.4).

4.1.2 Prediction Mechanism

VANET is a dynamic network consisting of high speed nodes moving throughout roads with movement restrictions due to speed limits, road shapes and conditions, and driver's behavior. Employing prediction procedures in such networks is feasible, simple, and beneficial. The simplicity of prediction is due to predictable driver behavior due to road barriers and conditions.

Because of rapid changes in node's location and speed in short time periods, it would be much preferable in VANETs to rely on predicted information rather than use the current information for future decisions since it is conducive to designing more efficient protocols. In this algorithm we rely on prediction to find out the next position of nodes, as well as calculate the CH selection metric. In this section we explain these two procedures briefly.

A. *Prediction based CH selection metric*

Our proposed CH selection metric for PCTT is the time period the target spends in the field of view of each vehicle. This time value is referred to as Observation Time (OBT), which is described in section 4.2.2 extensively. In DCTT (Chapter 3) we calculated the CH selection metric (TFP) based on the current movement pattern of each node as compared to the target, such as relative velocity and distance. Each node was supposed to send its TFP value to other nodes for future decisions. Therefore, every decision was made based on the previous information, considering the transmission and processing delays. Assume vehicle C calculates its TFP value for time t_0 and broadcasts this value in the cluster. The CH will receive this value at time t_1 after a short time interval (due to transmission delay). Therefore, the CH is making decisions based on received data, which is the old data calculated at time t_0, not t_1. The point is, vehicle C's position might have changed during this time interval, which is not considered in making clustering decisions. Thus, estimating the future behavior of nodes for making cluster maintenance decisions helps create a more efficient clustering algorithm for a dynamic VANET environment.

In PCTT algorithm, we predict the future movement of nodes to calculate their CH selection metric (OBT) and rely on the predicted movement patterns for making clustering decisions. We consider the current conditions of nodes and develop a movement function for each node based on existing metrics. This movement function is then used to predict future behavior of each vehicle. Should the condition remain unchanged, the movement function will be deemed valid for the next prediction periods.

B. *Employing prediction to calculate next location of nodes*

In clustering techniques, the CH is supposed to have information about cluster members. If the CH can predict this information, instead of receiving it periodically through beacon messages, the overhead will be decreased significantly. Clearly, by relying on prediction, fewer messages are required to maintain a cluster structure. For instance, in PCTT the CH predicts the future location and velocity of member nodes instead of receiving this information regularly. However, there is always a probability that a node's movement pattern changes and the prediction do not match reality. To address these concerns, a correction mechanism should be considered in every prediction-based method.

In PCTT algorithm we have considered prediction functions in CH and all member nodes. The CH receives the initial information about nodes and uses them as input for prediction function. Afterwards, CH will predict the next location of all members and will use the predicted information for maintenance decisions. The member nodes are also predicting their own next locations for the same time interval by the same prediction mechanism. If a node encounters a contradiction between its predicted location and the actual next location, it will inform the CH. But, if the prediction is correct within a certain error threshold, no beacon message will be sent by CMs to CH. This error correction mechanism does not provide so much overhead in the network but determines the information accuracy. In Figure 3, the prediction mechanism of this algorithm is illustrated. The other error correction method is to reset the predictions periodically. This means CH asks the nodes to send their current information at particular time intervals and use the actual information for the next round of prediction. The reset time interval is a longer period which does not cause much traffic in the network regularly. This process is beneficial when prediction denial messages are lost.

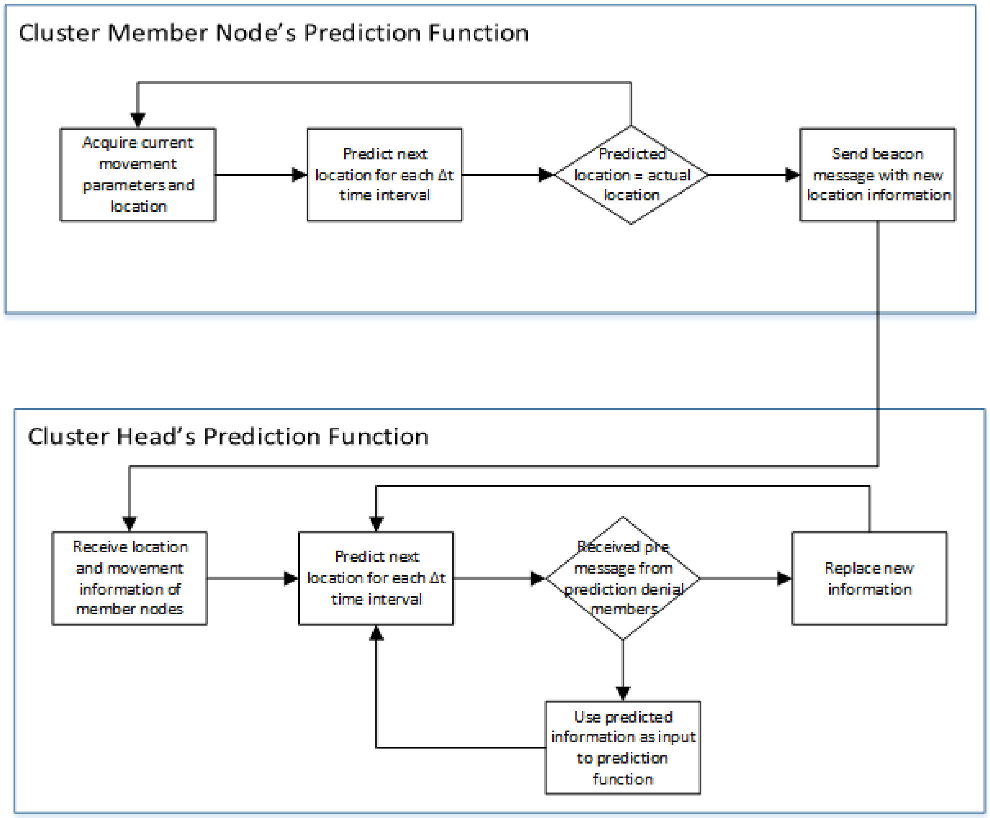


Figure 3. Prediction Mechanisms of the CH and CMs

4.2 Overview of Terms and Attributes

The main objective of PCTT algorithm is to continuously track a specific target based on its visual features. PCTT benefits from hybrid cluster-based and prediction-based techniques to acquire high levels of accuracy and efficiency while tracking a target in dynamic VANET environment. Some of the important terms used in this thesis are defined in Table 3.

Table 3. PCTT Term Definitions

CH	Cluster Head
CM	Cluster Member
CCH	Candidate Cluster Head
NM	Non-Member Node
CC	Control Center
CHM	Cluster Head Message
CMM	Cluster Member Message
CCM	Control Centre Message

TDV	Target Detection Value
OBT	Observation Time
FOV	Field of View
ΔT_R	Reset Time Interval
ΔT	Prediction Time Interval
ΔT_{Data}	Data Transmission Time Interval
RT	Resign Timer
ML	Member List
$InfoList$	Information List of CM nodes

The communication messages being sent between nodes are categorized into three types: Control Centre Message (CCM), Cluster Head Message (CHM) and Cluster Member Message (CMM). The messages fields are illustrated in Figure 4, Figure 5, and Figure 6.

NodeID	**PacketID**	**TargetInfo**	**Time**

Figure 4. Control Centre Message (CCM)

NodeID	**PacketID**	**Current Time**	**RT**	**OBT**	**JAck**	**CCH_ID**

Figure 5. Cluster Head Message (CHM)

NodeID	**PacketID**	**TDV**	**Current Time**	**Current Position**	**NodeInfo**	**Prediction Denial**

Figure 6. Cluster Member Message (CCM)

4.3 Observation Time as CH Selection Criteria

In this algorithm we consider CH as a node which can observe the target for a longer period of time. Clearly, a node that has the target in its field of view for a longer time interval is more probable to be selected as CH. In this condition, the cluster head selection metric is considered “Observation Time” or the amount of time the target spends in the field of view of each vehicle. We refer to this time value as OBT.

The Field of View (FOV) of each vehicle is defined as a semi-triangular shape (parts of a circle with radius *r*) in the front and rear of each vehicle which can vary based on the camera type. There is also an FOV angle that describes how wide this field can be. The assumed FOV in this algorithm is shown in Figure 7.

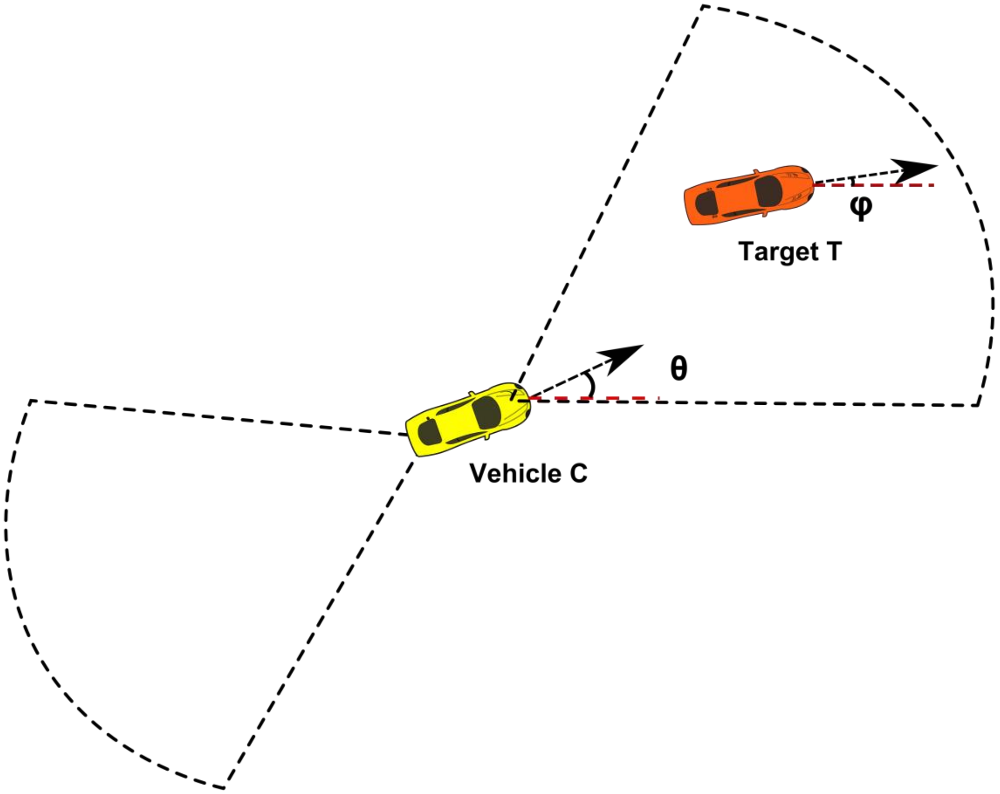


Figure 7. Field of View (FOV) and Velocity Vector of vehicles

Due to the fact that vehicles can move in different directions and with various velocities we need to define the movement pattern of each vehicle with a variety of variables to be able to calculate OBT for each vehicle. In calculating the OBT for a vehicle, we rely on the current parameters of the vehicle and predict its future behavior and use these values for cluster maintenance.

We assume a member vehicle C of the cluster can detect the target. At time t, vehicle C is at location $(\mathrm{xc}_t, \mathrm{yc}_t)$ and its velocity is Vc_t. Because vehicle C is moving, the FOV around this vehicle is moving as well. We consider vehicle C is moving with an angle of θ with X axis. In this case the velocity vector of vehicle C can be extracted in to two velocity vectors across X and Y axes as shown in Figure 8. The movement formula of the FOV of vehicle C is represented in equation 6. We have substituted the velocity vector on each axis in equation 6 and attained equation 7.

$$(6)\ (x-(xc_t+V_{cx}T))^2+(y-(yc_t+V_{cy}T))^2 = r^2$$

$$(7)\ (x-(xc_t+vc_t\cos\theta\ T))^2+(y-(yc_t+vc_t\sin\theta\,T))^2=\ r^2$$

The FOV of vehicle C moves as vehicle C is moving on its path. We assume vehicle C is located on the center of its circular FOV as illustrated in Figure 8. The FOV's center which is the vehicle C's location, changes based on the movement pattern of vehicle C as time passes. The FOV of vehicle C and its movement direction based on its velocity vector angle is shown in Figure 8.

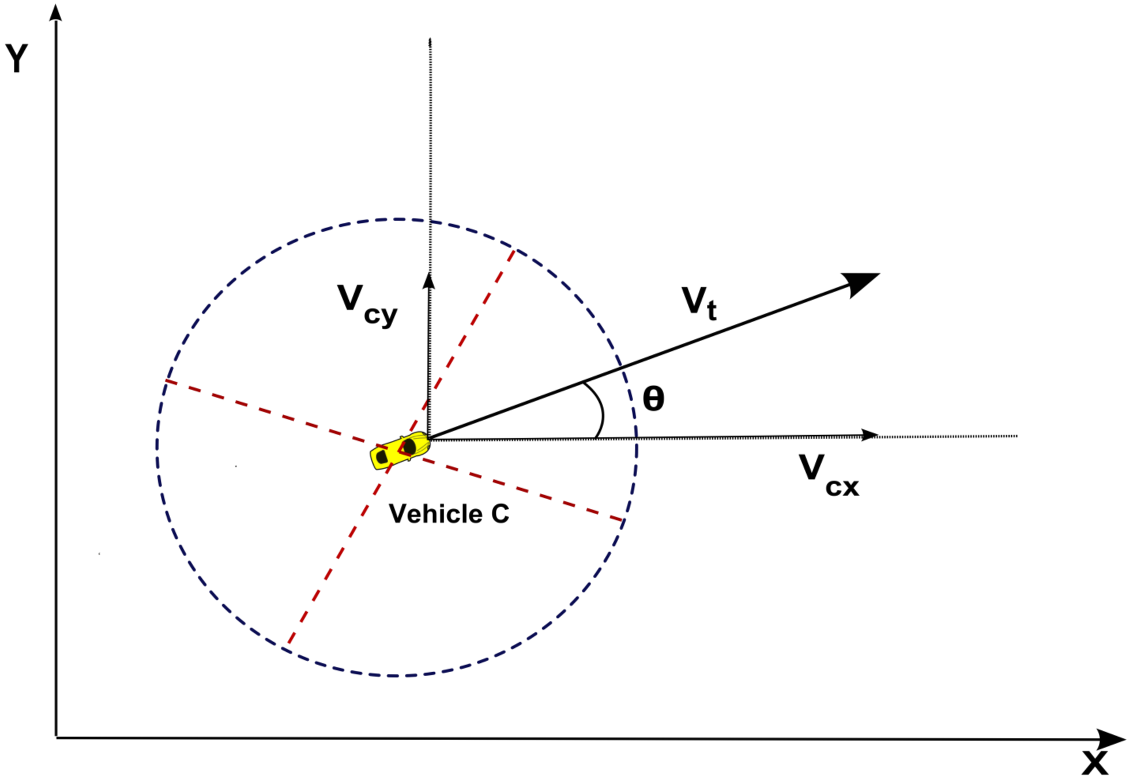


Figure 8. Velocity Vector Extraction for a member vehicle moving with $\boldsymbol{\theta}$ angle

The following equations show the movement pattern of target T if it moves with an angle of φ with the X axis:

$$(8)\begin{cases}x=vt_t\cos\varphi\ \ T+xt_t\\ y=vt_t\sin\varphi\ \ T+yt_t\end{cases}$$

Here we have the movement formulas for both vehicle *C* and its FOV and also for target *T*. We are interested to find out how long target T stays in the FOV of vehicle *C*. In order to find this time value, we substitute equation 8 in equation 7. Equation 9 represents the substituted formula based on relative movement:

$$(9)\ [vt_t\ \cos\varphi\ T + xt_t - (xc_t + vc_t \cos\theta\ T)]^2 + [vt_t\ \sin\varphi\ T + yt_t - (yc_t + vc_t \sin\theta\, T)]^2 = r^2$$

We need to have the movement function as a function of time (*T*) in order to be able to find out the OBT value. So, by solving the previous formula we will conclude the following quadratic equation of *T*:

$$(10)\ [(vt_t \cos\varphi - vc_t \cos\theta)^2 + (vt_t \sin\varphi - vc_t \sin\theta)^2]\ \boldsymbol{T}^2 + [(2 * xt_t * vt_t \cos\varphi - 2 * xc_t * vc_t \cos\varphi) + (2 * xc_t * vc_t \cos\theta - 2 * xt_t * vc_t \cos\theta) + (2 * yt_t * vt_t \sin\varphi - 2 * yc_t * vt_t \sin\varphi) + (2 * yc_t * vc_t \sin\theta - 2 * yt_t * vc_t \sin\theta)]\ \boldsymbol{T} + [(xt_t - xc_t)^2 + \left(yt_t - yc_t\right)^2 - r^2] = 0$$

Using formula 10, we can calculate the time period that the target stays in the FOV of vehicle C. Besides, by this formula we can compute whether the target will stay in the FOV of vehicle C forever based on the current situation or whether it will not enter the FOV at all (based on current conditions).

4.4 Algorithm Description

The PCTT algorithm is a hybrid cluster-based and prediction-based target tracking algorithm to continuously track a target vehicle and report its location to a control center, which can be assumed to be a central police station. This algorithm can help with sending target’s information such as location and visual data to a central node, which can serve different purposes such as active and passive monitoring. The proposed algorithm can use any visual recognition algorithm to find and track a target and send its information to a central entity. A considerable research is conducted on visual object detection and vehicle features recognition. These protocols include license plate recognition [55, 56, 59], and vehicle logo and color detection [57, 58] which are helpful for our proposed protocols in order to locate the target in the first place. We rely on visual feature detection of the target based on these visual processing algorithms.

We have employed clustering and prediction techniques in the design of PCTT to help with stability and functionality improvement and overhead reduction. PCTT comprises cluster creation or initialization phase, cluster management phase, and target tracking phase. The main entities of the algorithm which participate in tracking and maintenance phases are control center, non-member nodes seeking membership, cluster member nodes, cluster head node, and candidate cluster head node(s).

In this section we explain the tasks and procedures of each entity separately and introduce pseudo codes to describe each entity's functions.

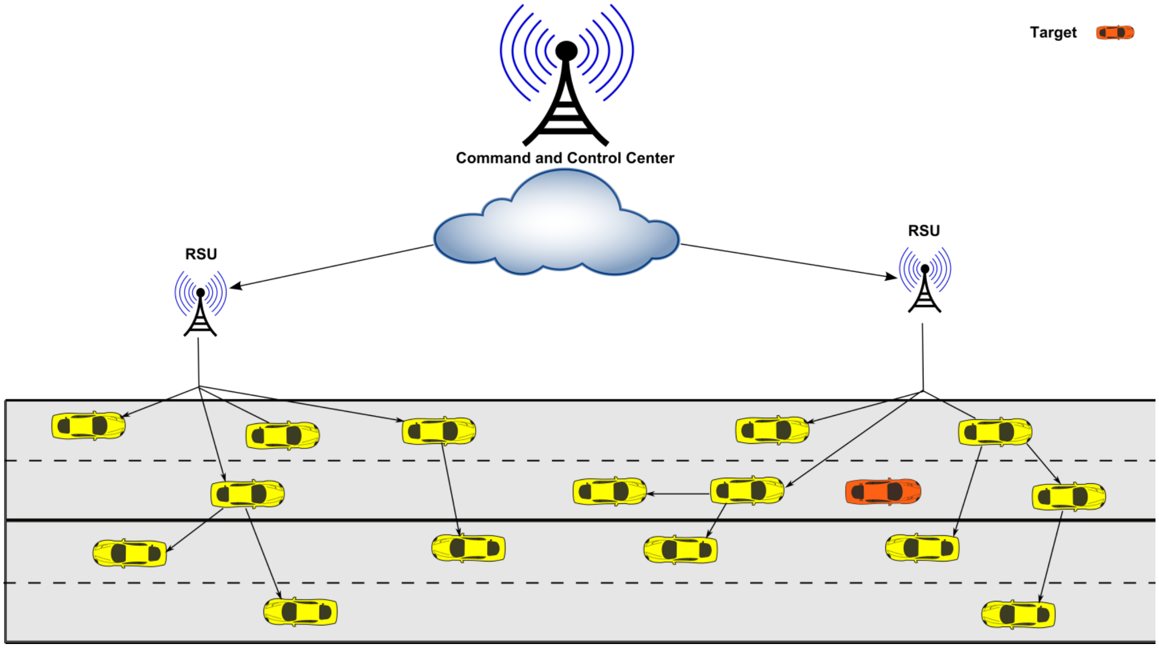
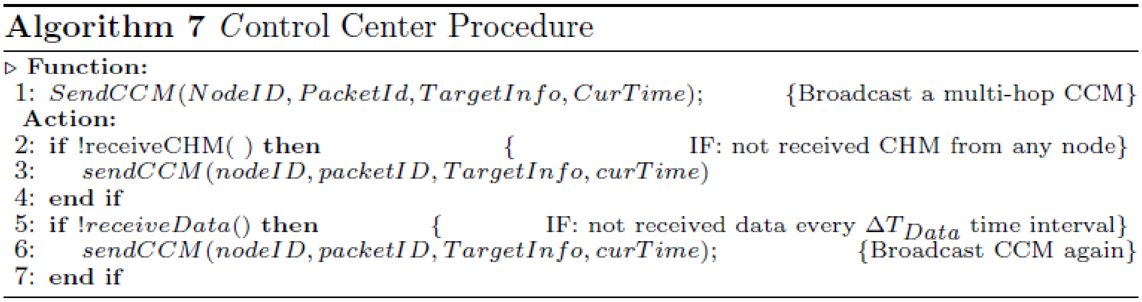


Figure 9. Control Center Functions

4.4.1 Control Center Functions

The command and control center (CC) is a central immobile entity such as a police station searching for a particular vehicle. This entity is interested in receiving location information about the target such as its location and visual information. The control center broadcasts the information of the target in the entire network in the form of control center messages (CCM) and waits to receive message from the nodes which can detect the target (Figure 9). As soon as the CC receives a message it stops broadcasting CCM and waits to receive tracking information from the selected CH. If CC does not receive data messages, it will start broadcasting the target's information in the network again. The function of CC is presented in Algorithm 7.

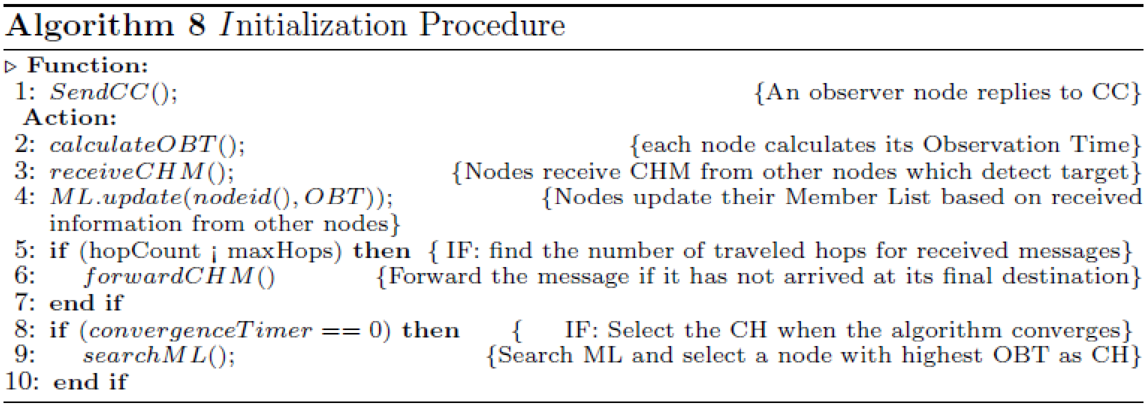


Algorithm 7 Control Center Procedure

```
▷ Function:
SendCCM(NodeID, PacketId, TargetInfo, CurTime);        {Broadcast a multi-hop CCM}
Action:
if !receiveCHM( ) then        {        IF: not received CHM from any node}
    sendCCM(nodeID, packetID, TargetInfo, curTime)
end if
if !receiveData() then        {        IF: not received data every $\Delta T_{Data}$ time interval}
    sendCCM(nodeID, packetID, TargetInfo, curTime);        {Broadcast CCM again}
end if
```

4.4.2 *Initialization Phase*

All the nodes which have received the control centre’s message (CCM), and are able to detect the target, participate in the initialization phase. These nodes are called “Observer Nodes” as explained in Chapter 3. Observer nodes (OBNs) start to form the first cluster by calculating their observation time (OBT) and broadcasting this time value throughout the network in their N-hop neighborhood. We assume in the initial cluster every cluster member is trying to become a CH, therefore all nodes broadcast a CHM and send their OBT value in this message in order to share this value with their neighbor nodes. All nodes keep received OBT values on a list called member list (ML) and search their ML after a defined time period to select a node with the highest OBT value as initial CH. The initialization procedure is represented in Algorithm 8. After the initialization phase, the cluster is formed and the initial CH is selected as displayed in Figure 10.



Algorithm 8 Initialization Procedure

```
▷ Function:
SendCC();        {An observer node replies to CC}
Action:
calculateOBT();        {each node calculates its Observation Time}
receiveCHM();        {Nodes receive CHM from other nodes which detect target}
ML.update(nodeid(), OBT));        {Nodes update their Member List based on received
information from other nodes}
if (hopCount ¡ maxHops) then  { IF: find the number of traveled hops for received messages}
    forwardCHM()        {Forward the message if it has not arrived at its final destination}
end if
if (convergenceTimer == 0) then        {    IF: Select the CH when the algorithm converges}
    searchML();        {Search ML and select a node with highest OBT as CH}
end if
```

4.4.3 *Cluster Maintenance Phase*

In this phase all the nodes cooperate to manage the cluster and provide an efficient, stable, and scalable cluster structure. The entities of this phase are Cluster Head (CH), Cluster Members (CM) and non-member nodes (NM).

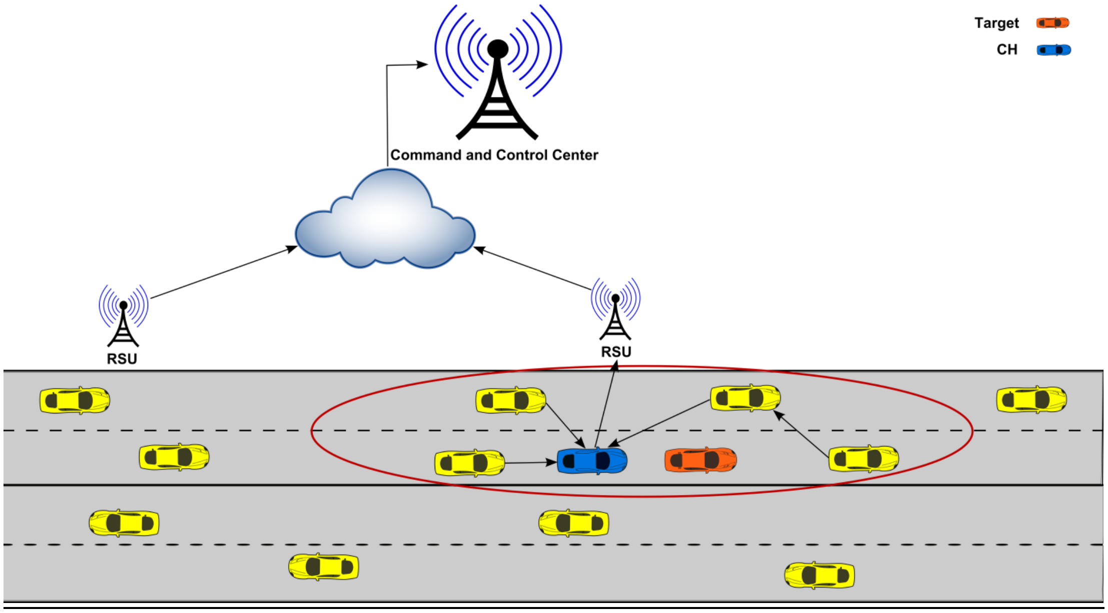


Figure 10. Cluster is formed and CH is selected after initialization phase

A. *Cluster Head Functions*

In PCTT algorithm, CH is the central management entity unlike DCTT algorithm (Chapter 3), in which all the nodes had a role in managing the cluster. Therefore, this algorithm is considered a centralized algorithm. The CH is responsible for cluster maintenance in order to make a reliable platform for target tracking. As discussed in section 4.1, a centralized algorithm requires a technique to prevent the central point of failure problem, which we have solved by selecting one or more candidate cluster heads. The cluster maintenance function of CH is illustrated in Algorithm 9.

The selected CH starts its tasks by sending a CHM in the cluster. The CHM is supposed to be sent regularly at each ΔT_R time interval. This time period is referred to as "Reset Time Interval", which we will define at the end of this section. Any node that receives CHM and can detect the target replies by sending a cluster member message (CMM).

After the CH receives a CMM from a member node, it checks if the source node is moving in the same direction as that of the target. Only the nodes moving in the target's direction are supposed to receive the membership approval. However, in exceptional cases where there is no CM, a node moving in the opposite direction is considered a CM in order not to lose track of the target.

The novel technique we use in our algorithm employs the prediction procedure, rendering the algorithm more efficient by reducing clustering overhead. We have considered a prediction function for the CH which receives the location and movement information of each vehicle at time t_0 and predicts their location for the next time interval (every ΔT Time interval) until it receives prediction denial messages from CMs. The predicted information is used to calculate OBT for every member periodically. The ML is updated according to recently calculated OBT values. Therefore, instead of relying on the actual location information sent from member vehicles, the CH relies on the predicted information to make clustering decisions. Subsequently, the CH searches the ML periodically to select the best CH and candidate cluster head (CCH).

The important point about our prediction mechanism is that every CM also predicts its future location by the same prediction procedure. Thus, prediction for each member takes place at two stages: first on the member side which is calculated by the prediction mechanism of the member itself; and second on the CH side, which is calculated by the prediction functions of the CH. Since both parties use the same prediction mechanism, the predicted information for vehicle C on both sides (on vehicle C itself and on the CH side) should be the same. Using this prediction method, there is no need for vehicle C to send its information to the CH at each ΔT time interval, because the CH is capable of predicting that information.

The other technique is that each vehicle predicts its own behavior for the next ΔT time interval and compares the predicted information with the actual information. To shed more light, vehicle C, which is a cluster member, predicts its location at time t_0 for time t_1. Then at time t_1, vehicle C is supposed to compare the predicted location for time t_1 with its actual location at time t_1. If the actual information conforms to the predicted information, vehicle C will not send any prediction denial message to CH; however, if the predicted information does not match the actual information, vehicle C is supposed to send its new information to CH. If no prediction denial message is received by CH, the CH assumes that its prediction about vehicle C is consistent with the real information on vehicle C’s side; and relies on this information for the next prediction round. Conversely, if the prediction denial message is received by the CH, it is expected to update the ML with the most recent information and rely on the new information for the next cluster management decisions. This concept helps prevent a huge number of messages being transferred between member nodes and the CH for cluster maintenance and management process. Figure 11 shows the prediction mechanism on CM and CH entities.

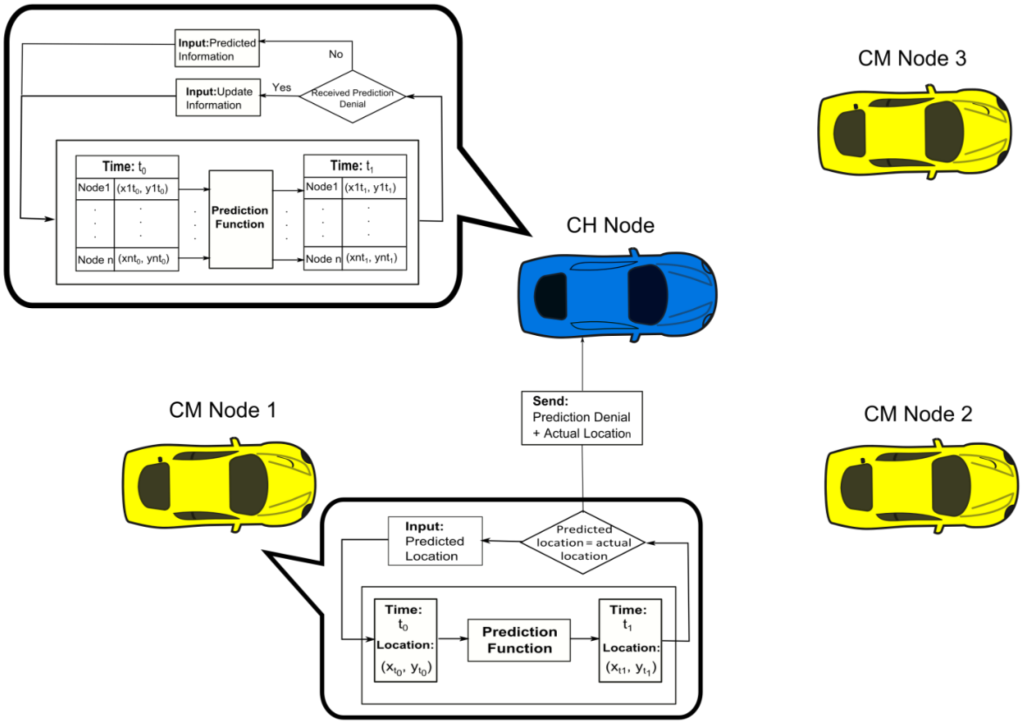
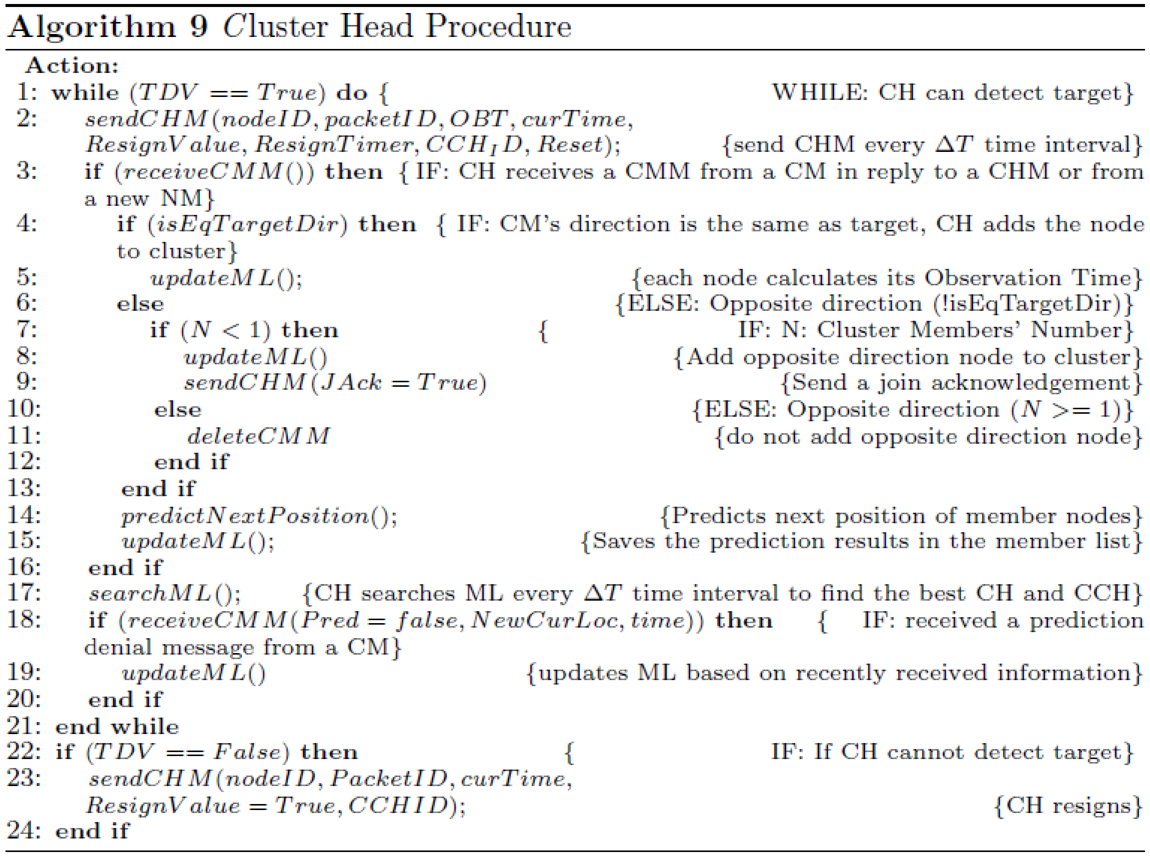


Figure 11. Prediction Procedure on CH and CM side

We have also considered a "Reset Time Interval" or ΔT, which is a time period that the CH asks nodes to send their current information to update its member list. Because CH depends on prediction unless it receives a prediction denial message (PreDenial), consideration of a reset interval is necessary to reduce error probability. This reset time helps retrieve accurate information in case of any errors or message losses. After the reset time interval, all the information is reset, which is like starting from scratch, with fresh and accurate information. Besides, the reset time interval is larger than the prediction time interval because we do not want to congest the network with excessive number of control messages.

(11) $\Delta T_R > \Delta T$

In centralized clustering algorithms, selecting a CCH can help in reducing failure probability in case of losing the current CH. The procedure we apply in assigning CCH is set up in such a way as to increase CH lifetime. The CCH is a vehicle which can detect the target for a longer time period than other member vehicles except the current CH. However, if a vehicle has a higher OBT than the current CH, we select it as CCH instead of changing it to CH. In this case, a timer called "Resign Timer" (RT) is set to CH's current OBT. This timer indicates the amount of time the current CH is still capable of detecting the target. When the CH selects a CCH, it sends RT to announce its resign time. Therefore, even if the CCH has more potential to become the CH, it should wait until the current CH is unable to see the target. This concept contributes to cluster stability by decreasing unnecessary changes of the CH. The CH should also send the latest information about member nodes and the unsent data regarding the target to the CCH.



Algorithm 9 *C*luster Head Procedure

```
Action:
while ($TDV == True$) do {    WHILE: CH can detect target}
    $sendCHM(nodeID, packetID, OBT, curTime,$
    $ResignValue, ResignTimer, CCH_ID, Reset)$;    {send CHM every $\Delta T$ time interval}
    if ($receiveCMM()$) then { IF: CH receives a CMM from a CM in reply to a CHM or from a new NM}
        if ($isEqTargetDir$) then { IF: CM's direction is the same as target, CH adds the node to cluster}
            $updateML()$;    {each node calculates its Observation Time}
        else    {ELSE: Opposite direction (!isEqTargetDir)}
            if ($N < 1$) then {    IF: N: Cluster Members' Number}
                $updateML()$    {Add opposite direction node to cluster}
                $sendCHM(JAck = True)$    {Send a join acknowledgement}
            else    {ELSE: Opposite direction ($N >= 1$)}
                $deleteCMM$    {do not add opposite direction node}
            end if
        end if
        $predictNextPosition()$;    {Predicts next position of member nodes}
        $updateML()$;    {Saves the prediction results in the member list}
    end if
    $searchML()$;    {CH searches ML every $\Delta T$ time interval to find the best CH and CCH}
    if ($receiveCMM(Pred = false, NewCurLoc, time)$) then {    IF: received a prediction denial message from a CM}
        $updateML()$    {updates ML based on recently received information}
    end if
end while
if ($TDV == False$) then {    IF: If CH cannot detect target}
    $sendCHM(nodeID, PacketID, curTime,$
    $ResignValue = True, CCHID)$;    {CH resigns}
end if
```

B. Cluster Members Functions

In PCTT cluster members are categorized into two types. The first level members (CM-L1) are able to detect the target. The second level members (CM-L2) are the nodes inside the cluster range; but they cannot see the target at this point of time. Due to the FOV shape of each vehicle, and the rapid movement of vehicle nodes, there is a high probability that a level 2 member will change into a level 1 member in a short time period and vice versa. So, we add both groups of nodes which are in the N-hop communication range of the CH as cluster members. However, members of each level have different tasks.

A level 2 member does not cooperate directly in cluster maintenance and tracking tasks. These nodes are mostly considered to be intermediate nodes which take part in forwarding the messages. Yet, they are aware of clustering information such as CH ID, CCH ID, and Target ID. As soon as they can detect the target they will be able to use their saved information about the cluster to adapt as CM-L1 immediately.

Compared to level 2 members, a level 1 member is an active cluster member and collaborates directly in cluster maintenance and tracking tasks. A level 1 member is supposed to reply to CH by sending CMM when the CH asks by sending a CHM. when a CM does not receive any message after a timeout interval, it assumes that it has gone out of cluster boundaries and turns into the non-member (NM) state.

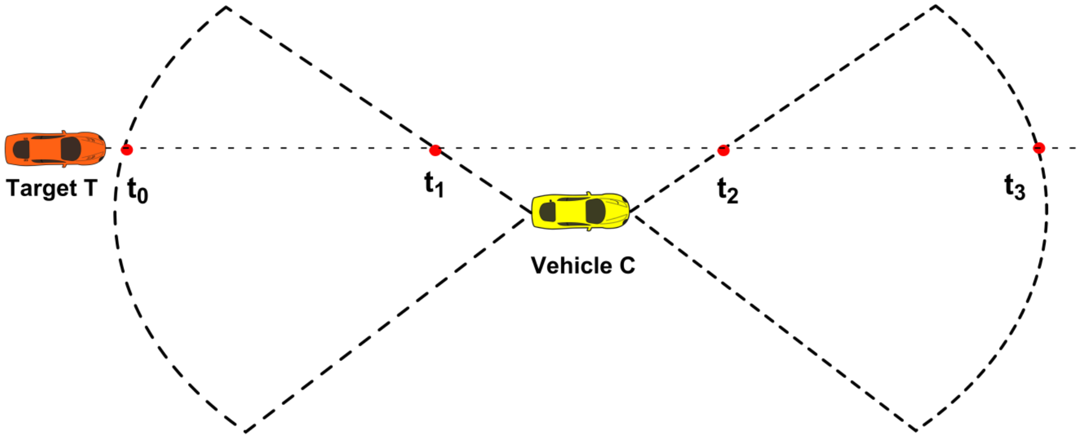


Figure 12. Effect of FOV shape on cluster member level idea

In Figure 12, we display the reason for adding level 2 nodes to the cluster. It is assumed that vehicle C is not moving and is fixed at its position; but the target is moving in the shown direction. It is clearly displayed that Target T enters the FOV of vehicle C at time t_0, and gets out the rear FOV at time t_1. The target will be out of the both fields of view of vehicle C for a short time period and then enters the front FOV of vehicle C at time t_2. Therefore, if we unjoin vehicle C from cluster as soon as the target exits the rear FOV at time t_1, we need to re-join it to the cluster at time t_2, which causes a lot of changes and overhead in the cluster and decreases cluster stability.

The CM tasks are divided into three categories as follow:

I. Prediction

Using prediction mechanisms help decrease the number of communication messages required to maintain cluster structure in VANETs. In this algorithm, every CM-L1 is expected to predict its own next position after each ΔT time interval. So, member vehicle

C which is at position (xc_{t0}, yc_{t0}) at time t_0, calculates its next position (xc'_{t1}, yc'_{t1}) for time t_1. At time t_1, this node arrives at its real position (xc_{t1}, yc_{t1}). Vehicle C will compare (xc'_{t1}, yc'_{t1}) and (xc_{t1}, yc_{t1}) to find out if the predicted position for time t_1 matches the actual position at time t_1. If the prediction and the actual position are equivalent, the CM will not send any message to CH to announce its location because the CH is predicting vehicle C's position with the same prediction mechanism and knows where vehicle C is based on that prediction. However, if vehicle C discovers that its predicted position for time t_1 is different from its actual position at time t_1, it sends a CMM and sets the Prediction Denial (PreDenial) field to true and sends its new information to the CH. The new information will be the input of prediction function for the next prediction round for both CM and CH. The prediction functions of CM and CH is illustrated in Figure 11.

II. Response to CH

When a CM-L1 receives a CHM, it means either the reset time is due or the CH wants to announce some important information such as a new CCH or its resigning time. If the Reset field in CHM is set to true, the CM should send its latest location information to the CH. Also, the CM should save any updated information sent in CHM in its InfoList. The InfoList is a short list on the CM side which saves critical cluster information received from the CH. There is no need for a CM to keep other members' information. If the reset field in the CHM is false, the message is considered an informing message and the CMs should only update their information accordingly and do not need to reply. In case a CM has been selected as CCH, it should watch the Resign Timer (RT) and as soon as the RT is up, this node should switch to the CH state.

PCTT algorithm is a multi-hop clustering algorithm where we consider a maximum number of hops which represents our cluster boundary and test the algorithm under various hop numbers. If, the number of travelled hops for the CHM is fewer than the maximum number of hops (maxHops), the node acts like an intermediate node and forwards that message. Otherwise, it assumes the message has arrived at its final destination and deletes the message.

III. Forwarding CMM

Unlike DCTT algorithm (Chapter 3), CM nodes do not need to know about other CMs because the CH is performing all required calculations for maintaining the cluster. The CH receives all essential information from CMs directly. According to this principle, when a CM receives a CMM from its neighbor nodes, it is supposed to forward the message if the maximum number of hops is not reached. Otherwise, it should just ignore and delete the message. In other words, the CMs merely play the role of an intermediate node for other CMs.

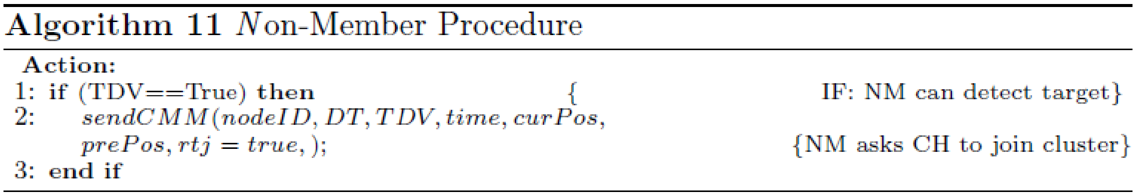
The functions of cluster members are presented in Algorithm 10.

Algorithm 10 Cluster Member Procedure

```
Action:
if ($TDV == True$) then    {IF: CM can detect target}
    $CM \leftarrow CM - L1$;
    $predictNextPosition()$;    {CM predicts its next location every $\Delta T$ time interval}
    if ($PredictedPosition! = ActualPosition$) then    {IF: Prediction on CM side is false (CM checks prediction every $\Delta T$ time interval}
        $sendCMM(nodeID, TDV, PreDenial = false, time, NewcurPos, isEqTargetDir)$;    {CM sends updated information to CH}
    end if
    if ($receiveCMM()$ then    {IF: received message from other members}
        if ($hopCount < maxHops$) then {IF: CMM is not arrived at its final destination based on hop count}
            $ForwardCMM()$;    {CM aacts like an intermediate node and forwards CMM}
        end if
        if ($hopCount == maxHops$) then    {IF: max number of hops is reached, CM deletes CMM}
            $deleteCMM$;
        end if
    end if
    if ($receiveCHM()$) then    {IF: received message from CH}
        if ($CHM.Reset == True$) then    {IF: received reset message from CH every $\Delta T_R$}
            $sendCMM(nodeID, TDV, PreDenial, time, curPos, isEqTargetDir)$;    {CM replies with a CMM}
        end if
        if ($CHM.CCHID == NodeID$) then    {IF: The CM is selected as CCH}
            $CM \leftarrow CCH$;
            $updateInfoList()$;    {save RT(Resign Timer)to keep the CH resign time}
        end if
        if ($CHM.CCHID! = NodeID$) then    {IF: The CM is not been selected as CCH}
            $updateInfoList()$;    {saves CHM information such as CCH and RT}
        end if
        if ($RT == 0$) then    {IF: RT is up}
            if ($NodeID == CCHID$) then    {IF: CM is the CCH}
                $CH \leftarrow CCH$;
            end if
            if ($NodeID! = CCHID$) then    {IF: CM is not the CCH}
                $updateInfoList()$;    {CM updates its infoList with CHID}
            end if
            if ($hopCount < maxHops$) then    {IF: CHM is not arrived at its final destination}
                $ForwardCHM()$;
            end if
        end if
    end if
else    {ELSE: CM cannot detect target(TDV=False)}
    $CM \leftarrow CM - L2$;
    if ($receivedCMM()$ then    {IF: CM-L2 received a CMM from other member nodes}
        if ($hopCount == maxHops$) then    {IF: CMM has arrived at its final destination}
            $deleteCMM$;
        else
            $ForwardCMM()$;
        end if
    end if
    if ($receivedCHM()$) then    {IF: CM-L2 received a CHM}
        $updateInfoList()$;    {save the RT and CCHID}
    end if
    if ($hopCount < maxHops$) then
        $ForwardCHM()$;    {CM-L2 acts as an intermediate node}
    end if
    if ($!receiveCMM()||!receiveCHM()$) then    {IF: CM-L2 does not receive any message}
        $NM \leftarrow CM$;    {CM switches to a non-member (NM)}
    end if
end if
```

Non-Member Nodes Function

A node which is not a CM or a CH is considered a non-member node (NM). If a NM detects the target, it tries to join the cluster. Not all NM nodes run the NM procedure, but only the ones that can detect the target. A NM node cannot join the cluster immediately. It requires a Join Acknowledgment (JAck) from the CH. This procedure is applied because a NM node can be a node moving in the opposite direction, which is not supposed to join the cluster under existing circumstances. However, the CH is in charge of granting a permission to join to the opposite direction nodes in exceptional conditions when there are not adequate members to perform tracking successfully. Immediately after the NM node receives the JAck from CH it turns into CM state. The NM procedure is explained in Algorithm 11.

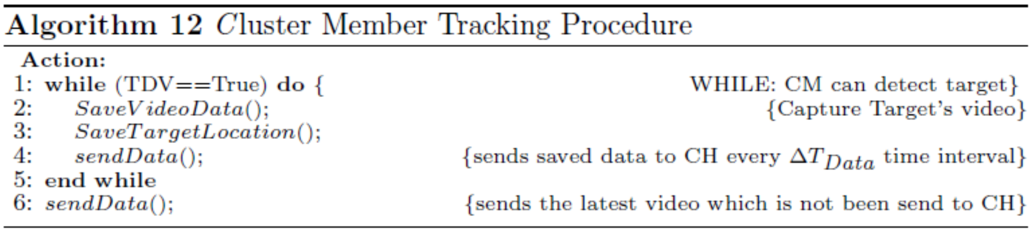


Algorithm 11 Non-Member Procedure

```
Action:
if (TDV==True) then                                   {IF: NM can detect target}
    $sendCMM(nodeID, DT, TDV, time, curPos,$
    $prePos, rtj = true, );$                          {NM asks CH to join cluster}
end if
```

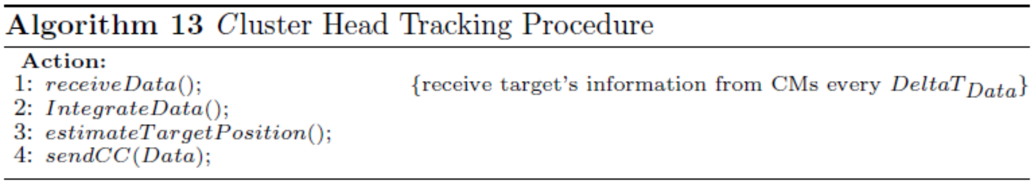
4.4.4 Tracking Phase

In this algorithm we define tracking as capturing continuous visual and location information of the target and reporting it to CC at each specified Data Time Interval (ΔT_{Data}). The CM-L1 nodes and CH are the active nodes participating in tracking the target. Level 2 members are not capable of detecting the target and acquiring its location information because the target is not in their FOV. The CM-L1 nodes capture target's information and send it to CH every ΔT_{Data}. As soon as the target goes out of FOV of a CM, the CM stops the tracking task; but it needs to send its unsent data to the CH. In case a CM loses the target at time $\mathrm{t_n}$ before the end of the ΔT_{Data} interval, it is supposed to send the CH the last captured data which has not been sent yet. This information includes the visual and location information taken from time $\mathrm{t_{n-1}}$ to $\mathrm{t_n}$.As the central tracking entity, the CH is responsible for integrating the received information received from the member nodes and sending it to CC. Based on the information received from all members; the CH estimates the target's position and reports its accurate coordinates to the control center. The CH and CMs tracking procedures are illustrated in Algorithm 12 and 13.



Algorithm 12 Cluster Member Tracking Procedure

```
Action:
while (TDV==True) do {                         WHILE: CM can detect target}
    SaveVideoData();                           {Capture Target's video}
    SaveTargetLocation();
    sendData();                                {sends saved data to CH every $\Delta T_{Data}$ time interval}
end while
sendData();                                    {sends the latest video which is not been send to CH}
```



Algorithm 13 Cluster Head Tracking Procedure

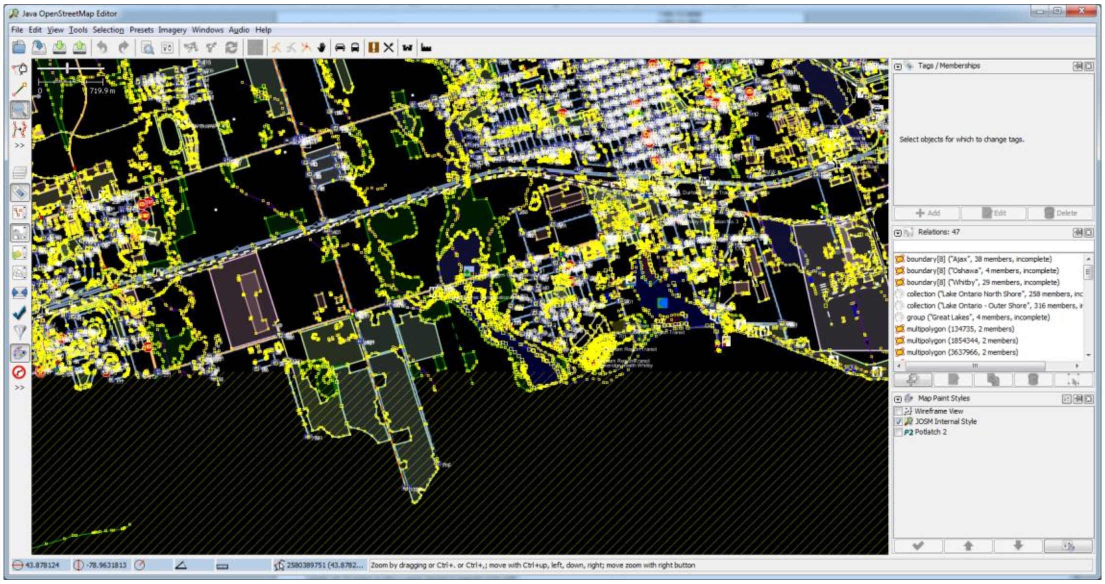
```
Action:
receiveData();                 {receive target's information from CMs every $DeltaT_{Data}$}
IntegrateData();
estimateTargetPosition();
sendCC(Data);
```

5. Evaluation of Proposed Protocols

In this chapter, the scenarios are explained and the simulations results are represented. Due to the large scale of VANETs, the proposed algorithms were evaluated through simulation. The simulators that are used to generate the vehicle traces and create the communication framework include the Simulator of Urban Mobility (SUMO), NS-2, and Tossim.

5.1 Simulation Environment

Simulator of Urban Mobility (SUMO) is used for traffic simulations of VANET projects [88]. We have used SUMO to generate vehicle traces for our algorithms. The chosen simulation environment for our experiments includes 10 km of Ontario Highway 401 from the city of Oshawa to Ajax. The map is extracted from OpenStreetMap website [89] by using Java OpenStreetMap Editor (JOSM) [90]. The street shapes, traffic lights, and all the default downloaded objects can be edited and re-configured in JOSM. The downloaded Highway 401 in JOSM environment map is shown in Figure 13. In order to simulate the communication framework between nodes for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication we have used NS-2 and Tossim. NS-2 is a discrete event simulator designed for network researches [91]. TinyOS SIMulator (Tossim) is a network Simulator for TinyOS applications. Tossim is a discrete event simulator that is designed for wireless networks [92, 93].



Select objects for which to change tags.

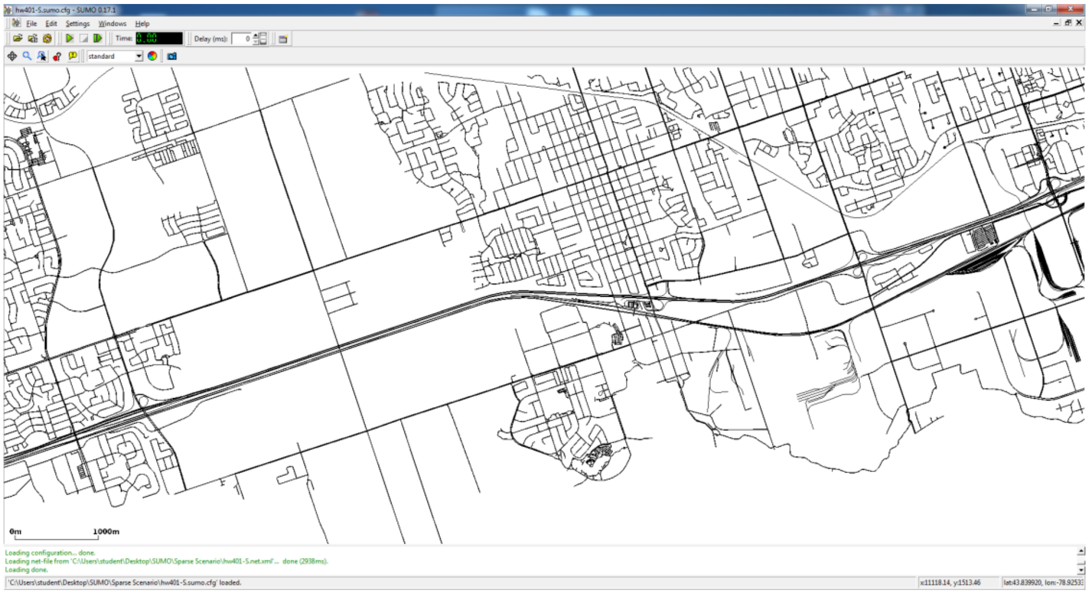
boundary[8] ("Ajax", 38 members, incomplete)
boundary[8] ("Oshawa", 4 members, incomplete)
boundary[8] ("Whitby", 29 members, incomplete)
collection ("Lake Ontario North Shore", 258 members, inc
collection ("Lake Ontario - Outer Shore", 316 members, i
group ("Great Lakes", 4 members, incomplete)
multipolygon (134735, 2 members)
multipolygon (1854344, 2 members)
multipolygon (3637966, 2 members)

☐ Wireframe View
☑ JOSM Internal Style
☐ Potlatch 2

Zoom by dragging or Ctrl+. or Ctrl+,; move with Ctrl+up, left, down, right; move zoom with right button

Figure 13. Simulation Environments in Java OpenStreetMap Editor (JOSM)

Figure 14 shows the map we use for our traffic generation in SUMO. This map includes the part of highway 401 that we use for simulations during different times of a day. Figure 15. **A Part of Simulation Highway in SUMO including Vehicle Streams** shows a part of our simulation environment including the vehicles on the highway 401.



```
Loading configuration... done.
Loading net-file from 'C:\Users\student\Desktop\SUMO\Sparse Scenario\hw401-S.net.xml'...  done (2938ms).
Loading done.
'C:\Users\student\Desktop\SUMO\Sparse Scenario\hw401-S.sumo.cfg' loaded.
x:11118.14, y:1513.46
lat:43.839920, lon:-78.92533
```

Figure 14. Simulation Environments in SUMO

We have considered various density scenarios, e.g., sparse, medium density, and dense to evaluate the proposed frameworks under different circumstances. As noticed, during the day-time and mostly in the early mornings, a huge traffic is moving on highway 401 from east to west. Therefore, by implementing a dense scenario we can evaluate our protocol's performance under high traffic network. In a medium density scenario, the distances between nodes are longer compared to dense scenario. However, there are numerous vehicles that can detect the target and can join the cluster. The last scenario we implemented is a low density network. For our application, a sparse network is not an ideal situation, because the track of target might be lost due to unavailability of vehicle nodes around the target. However, if the number of vehicle nodes around the target is more than one vehicle, there is still a chance of being able to track the target.

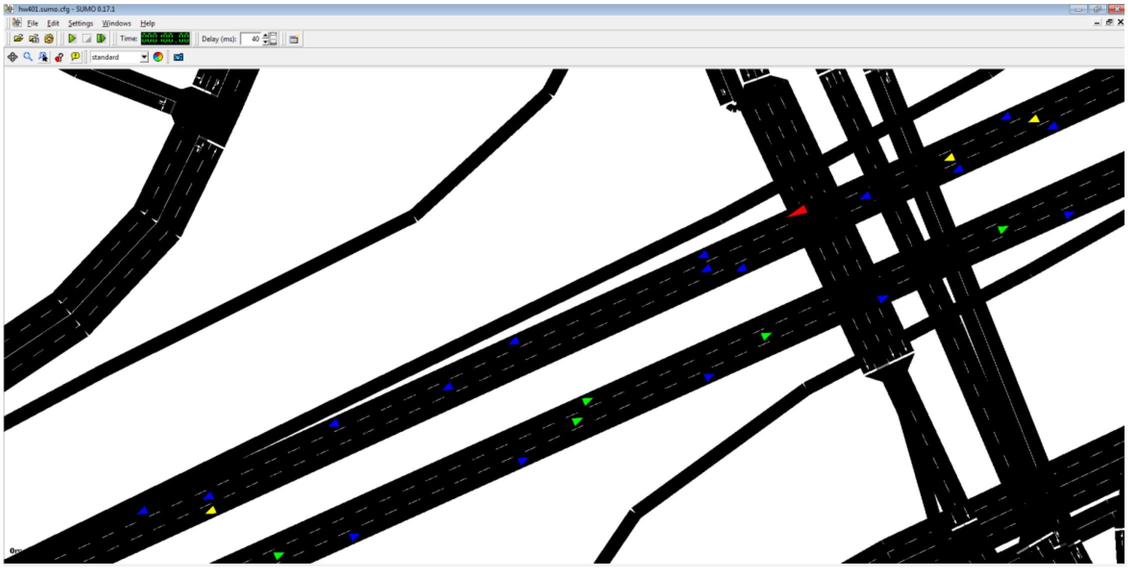


Figure 15. A Part of Simulation Highway in SUMO including Vehicle Streams

In order to create more real-world scenarios, we have defined various flows of vehicles with different movement patterns. The vehicle flows have different speed range and take different routes. In this case we make sure the cluster members and CH will not always be the same, and the cluster structure will change as it happens in the real world. This assumption helps in realistic evaluation of our proposed protocols.

5.2 Definition of Some Functions and Techniques

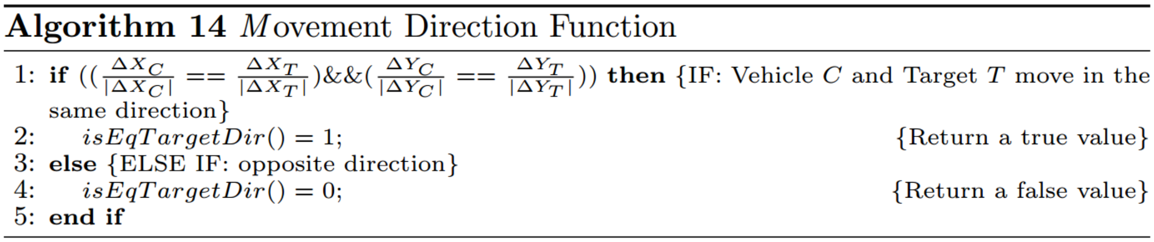
Some of the main techniques and functions used in implementation of the proposed protocols are explained in this section.

5.2.1 Movement Direction of Vehicles

Resolution of nodes' movement direction is a necessary step in order to avoid opposite direction nodes from joining the cluster. We have implemented a function to acquire moving direction of moving vehicles in the simulation environment as displayed in Algorithm 14. This function uses the position of a vehicle *C* and the target *T* at times t_0 and t_1. Then, based on two acquired positions, the movement directions for vehicle *C* and Target *T* are calculated. In case both vehicles are moving on the same direction, this function returns a true value. But if the movement directions are different, the returned value will be false. As mentioned before, in most scenarios the opposite direction nodes should not join the cluster in order to decrease cluster changes as much as possible. The assumptions for movement direction calculations are defined in Table 4.

Table 4. Assumptions for Movement Direction Calculation

Vehicle C's location at time t_0 and t_1 respectively:	$(X_{C0}, Y_{C0}), (X_{C1}, Y_{C1})$
Target T's location at time t_0 and t_1 respectively:	$(X_{T0}, Y_{T0}), (X_{T1}, Y_{T1})$
Movement pattern of target *T* from time t_0 to t_1 :	$\begin{cases} \Delta X_T = X_{T1} - X_{T0} \\ \Delta Y_T = Y_{T1} - Y_{T0} \end{cases}$
Movement pattern of vehicle C from time t_0 to t_1 :	$\begin{cases} \Delta X_C = X_{C1} - X_{C0} \\ \Delta Y_C = Y_{C1} - Y_{C0} \end{cases}$



Algorithm 14 *M*ovement Direction Function

```
if $((\frac{\Delta X_C}{|\Delta X_C|} == \frac{\Delta X_T}{|\Delta X_T|})\&\&(\frac{\Delta Y_C}{|\Delta Y_C|} == \frac{\Delta Y_T}{|\Delta Y_T|}))$ then {IF: Vehicle $C$ and Target $T$ move in the same direction}
    $isEqTargetDir() = 1;$                                   {Return a true value}
else {ELSE IF: opposite direction}
    $isEqTargetDir() = 0;$                                   {Return a false value}
end if
```

5.2.2 Target Detection Value (TDV)

The other important variable we need to calculate while the vehicles are moving is Target Detection Value (TDV). The TDV value determines if the target is inside the field of view of a vehicle *C* or not. Calculation of TDV for the proposed protocols is a delicate issue. The reason is the shape of the FOV shape as illustrated in Figure 16.

As displayed in Figure 16, vehicle *C* is located at position (X_c, Y_c) at time *t*. The target *T* is located at position (X_T, Y_T) at the same time. In our simulation environment, we are not capable of using visual processing directly to find the TDV value. We only have access to position information of each vehicle. The implemented function is applicable to all movement models.

It is assumed that vehicle C is moving with an angle θ with X-axis. The challenge of TDV calculation in this thesis originates from the FOV shape which is a part of a circle with a defined angle δ as illustrated in Figure 16. In order to detect whether target (red vehicle) is inside FOV of vehicle *C* (yellow vehicle) we assume vehicle C is the center of a new axes system. Therefore, the current X and Y axes should be rotated and mapped to a new location. Then the coordinates of target in the new axes system is calculated which will be (X_{Tnew}, Y_{Tnew}). Afterwards, we presume a line connecting vehicle *C* to target *T*. This line is shown as the green line with the length of σ in Figure 16. The value of σ represents the distance between vehicles *C* and target *T*. The angle between this line and the new X-axis represents whether the target is inside or outside of the FOV of vehicle *C*. This angle is represented as ω and is smaller than $\delta/2$ if the target is inside FOV of vehicle *C*. The assumption for target detection value calculations are displayed in Table 5. The defined steps are represented in algorithm 15.

Table 5. Assumptions for Target Detection Value Calculation

Vehicle *C*'s location at time t_0 :	(X_{c0}, Y_{c0})
Vehicle *C*'s location at time t_1	(X_{c1}, Y_{c1})
Target's location at time t_1 :	(X_T, Y_T)

The following formulas are used to calculate required parameter for TDV computations:

$$(12)\quad \theta = tan^{-1}\frac{Y_{c1}-Y_{c0}}{X_{c1}-X_{c0}}$$

$$(13)\quad X_{Tnew} = \big((X_T - X_{c1}) * \cos\theta\big) + \big((Y_T - Y_{c1}) * \sin\theta\big)$$

$$(14)\quad Y_{Tnew} = \big((Y_T - Y_{c1}) * \cos\theta\big) - \big((X_T - X_{c1}) * \sin\theta\big)$$

$$(15)\quad \omega = \left|\tan^{-1}\frac{Y_{Tnew}}{X_{Tnew}}\right|$$

$$(16)\quad \sigma = \sqrt{|(Y_{c1} - Y_T)^2 + (X_{c1} - X_T)^2|}$$

Algorithm 15 *T*arget Detection Function

```
if $\omega \leq \frac{\delta}{2} \&\& \sigma \leq r$ then {          IF: Target $T$ is inside FOV of vehicle $C$}
    $TDV = 1$;                                                 {Set TDV field to true}
else {ELSE IF: Target $T$ is not inside FOV of vehicle $C$}
    $TDV = 0$;                                                 {Set TDV to a false value}
end if
```

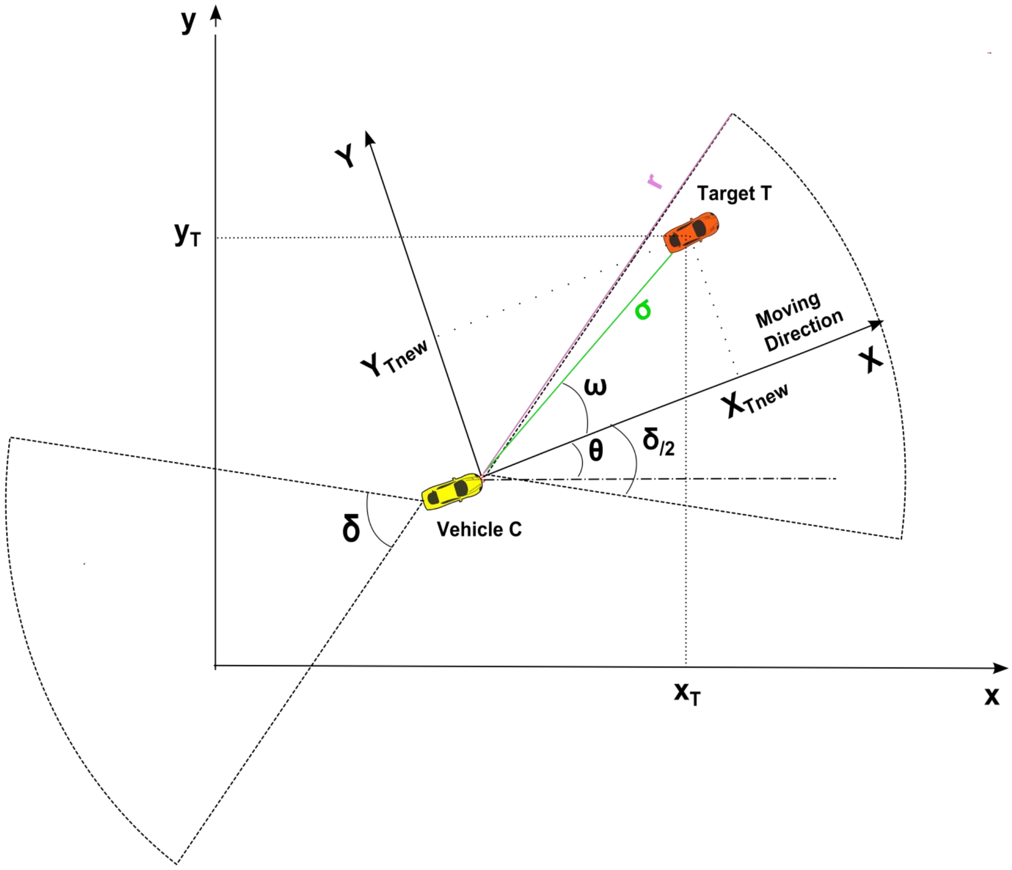


Figure 16. Target Detection Value (TDV) Calculation

5.2.3 Semi-Passive behavior

As explained in Section 2.9, passive clustering is proposed in [15] for MANETs. In passive clustering, nodes do not send control messages separately. They attach required control information to data packets and send them during the data message transmission interval. VANET is a dynamic network and the nodes are changing their status frequently. Applying passive techniques to VANETs may decrease control overhead considerably. However, by employing passive techniques, clustering decisions would not be accurate and precise enough. We propose a technique to decrease control overhead but not eliminate them completely. We call this technique semi-passive clustering, because it is a combination of traditional and passive clustering.

In this technique we assume nodes send data packets every ΔT_{Data} time interval. The control message interval is supposed to be ΔT which is smaller than data time interval as displayed in Figure 17. At each control time, the member node checks the data time interval. If the current time is close to the next data delivery time interval, the CM will not send a control message. Instead, it will attach the required control fields to the data packet and will send the data packet at the next data delivery time. Elimination of unnecessary control messages decreases the control overhead due to reduction of some control message fields.

Here we assume each data time interval is equal to four control time interval as displayed in Figure 17. A member node checks the current time and defines whether it should send a control message or not as shown in equation 17.

$$(17) \quad \begin{cases} \text{if } (\mathrm{T_{current}} < \frac{3n\Delta t}{4}) \Rightarrow \text{Send control message} \\ \text{if } (\mathrm{T_{current}} \geq \frac{3n\Delta t}{4}) \Rightarrow \text{Do not send control message} \end{cases}$$

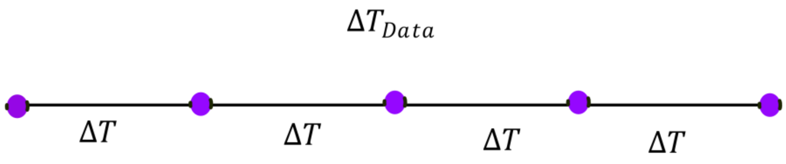


Figure 17. Data and Control Time Interval

This method should be tested under various data and control time interval. Semi-passive technique can be applied to both DCTT and PCTT algorithms in order to decrease control overhead. Evaluation of this method is left for future work.

5.2.4 Message Aggregation and Compression by the Cluster Head

We assume or protocols are capable of capturing target's video and location information and sending the information to the control center. However, we do not send actual real-time video information in the simulations. In our cluster-based target tracking algorithms, the CH is responsible for aggregating the video packets received from member nodes and sending the compressed information to the control center. It is important to note that the video sequences have been captured from multiple cameras with different positions and view angels. Target tracking with multiple cameras brings about the challenge of multi-view video coding and compression. This concept has been studied widely under the areas of multi-media, computer vision, and image processing [94-97]. Multiple video streams captured by various cameras provide more realistic depth information and help in covering larger areas [98]. However, the video sequences from various cameras may contain considerable similarities and correlations [99]. For target tracking purpose, the CH detects video sequence similarities by applying an algorithm proposed in [97]. This compression technique captures similarities of motion vectors from each view, using a 3D motion estimation (3D ME) technique. This technique is applied to four standard video sequences i.e. Exit, Ball room, Vassar, and break Dancing for evaluation purpose. The authors assert that a frame from one view has a similarity of 51% to 93% to the same frame from other views [97]. The similarity ratio varies based on the scenarios. In more dynamic scenarios, similarities between various view frames are less; but, in less dynamic circumstances, more similarities may be found. Compared to the tested scenarios in [97], we have assumed that in this thesis the video frames captured by each vehicle can have a similarity rate of 60%. Therefore, the CH applies a compression technique on received video sequences and eliminates redundant frames before forwarding the information to the control center. Employing such a technique, bandwidth usage and data overload in the network would decrease significantly.

5.3 Performance Metrics

The performance metrics evaluated in this thesis are categorized into two groups in order to represent performance of cluster-based techniques and its effects on dissemination of tracking data to the central entity. The evaluation metrics are described as followings:

$\underline{Clustering\ Overhead}$

The overhead in the cluster is caused by sending control messages for cluster management. These messages include information about cluster entities and are transmitted periodically in the cluster. The control overhead metric represents the percentage of control packets to the total transmitted packets in the cluster. The lower value of control overhead shows better performance of a clustering algorithm. The control overhead of a clustering protocol is calculated as follow:

$$Clustering\ Overhead = \sum Control\ packets \Big/ \left(\sum Control\ packets + \sum Data\ packets\right)$$

$\underline{Cluster\ Head\ Lifetime}$

In a clustering algorithm, the CH changes as time passes based on conditions and protocol requirements. At every defined time interval the eligibility of the current CH should be evaluated in order to select the best CH. The CH lifetime is the time interval a node is selected as CH until it gives up its CH role. The longer CH lifetime represents better fewer changes in the cluster structure and improved cluster stability. In this thesis the CH lifetime metric is represented in milliseconds.

$\underline{Cluster\ Member\ Lifetime}$

Cluster member lifetime shows the average time a node spends in the cluster. The membership time is calculated for each member node separately and the average value is represented as cluster member lifetime. A higher value of cluster member lifetime defines better performance of a clustering protocol.

$\underline{Packet\ Delivery\ Ratio}$

This metric represents the percentage of delivered packets to destination that is the percentage of successful deliveries in the network. Packet delivery ratio is calculates as follow:

$$Packet\ Delivery\ Ratio = \sum Number\ of\ received\ packets \Big/ \sum Number\ of\ sent\ packets$$

The greater value of delivery ratio shows better performance of the protocol. In this thesis total delivery ratio represents successful delivery of target's information from every cluster member to cluster head and from the cluster head to the control center.

$\underline{End\text{-}to\text{-}End\ Delay}$

End-to-End delay is the average time takes for a packet to arrive to a defined destination. In this thesis, the end-to-end delay is referred to as the average time it takes for a packet to travel from a cluster member to the control center.

The end-to-End delay is calculated as follow:

$$End - to - End\ Dela = \sum (\,arrive\ time - \ send\ time\,)\ \ / \ \sum Number\ of\ sent\ messages$$

5.4 Scenarios and Algorithms

5.4.1 Structureless Target Tracking Algorithm

We assume our algorithms will be useful for sending location information and streaming of video information about the target although we have not simulated the actual video streaming scenarios. In order to achieve this goal, the proposed algorithms should be able to manage large amounts of information without affecting the performance negatively.

We have simulated a structureless, carry and forward scenario for tracking and information delivery to a base station to represent the necessity of having a structured cluster based target tracking algorithm for VANETs. In this scenario every vehicle is responsible of retrieving location information of the target and sending it to the control center as soon as it arrives into its communication range. Using this method, delivery ratio may decrease significantly due to separate packet transmission of nodes to the same base station which causes unavoidable packet loss. Furthermore, delay of carry and forward method is so high and we cannot rely on such a framework for real-time vehicle tracking and reporting purposes.

The other structureless technique for vehicle tracking may be mentioned as flooding which is not appropriate for transmission of large data packets. In flooding, a vehicle that detects the target sends visual and location information of the target directly to the control center. The control center may be located in a multi-hop communication distance from the vehicles. Therefore, vehicles broadcast target's information in order to inform the control center. Information about the target needs to travel a multi-hop distance in order to arrive at control center. The problems caused by this method are as following:

- The control center is probable to get congested by large amount of packets received from each node separately mostly in dense networks. The reason is every node sends target's information directly to control center instead of sending it to a central aggregator node like CH.
- The network may get congested by the numerous large data packets being broadcasted in a multi-hop manner.
- The received visual information on the control center includes redundant frames due to lack of a central entity i.e. CH to aggregate the information received from multiple view cameras. Transmission of redundant information is a waste of bandwidth.
- In the flooding algorithm, every node sends target's location information separately to the control center. The location information received from each vehicle node might not be accurate because it is acquired by visual processing. In the proposed clustering algorithms (DCTT and PCTT), the CH receives all location information and estimates approximate location of the target before sending it to the control center. This technique increases the target's location

accuracy information which is received at control center. However, in a flooding algorithm, there is not a central node responsible for determining target's location accuracy that may result in receiving inaccurate tracking information in the control center. Furthermore, redundant location information utilizes the bandwidth by traveling a multi-hop distance and may overload the network in dense network scenarios.

5.4.2 Adapted MDMAC Protocol for Target Tracking

As mentioned in Section 2.9 MDMAC [17] is a modification of the DMAC [78] algorithm that makes it suitable for VANET networks. The clustering metric is called freshness value which represents which nodes are eligible to be in the same cluster. The freshness value is transferred between nodes in HELLO messages. The cluster head selection metric in this algorithm is a constant weight value such as node ID. The other distinctive properties of MDMAC algorithm are preventing opposite direction nodes to join the cluster, and forming multi-hop clusters. These characteristics make the algorithm appropriate to apply for target tracking in VANETs. We have used the clustering properties of MDMAC and have adapted this algorithm to target tracking application for VANETs. The simulation results show better performance of DCTT and PCTT algorithms in comparison to MDMAC for target tracking purpose.

Using constant weight as CH selection metric is not appropriate for VANET clustering algorithms. The reason lies behind the high mobility of nodes which causes rapid topological changes in the network. Therefore, a weight metric should be calculated based on proper mobility features of nodes such as velocity, distance, acceleration, and connectivity time. Relying on a constant weight as CH selection metric causes cluster instability by decreasing CH lifetime and increasing number of CH changes. Besides, CH change requires more control messages to be transferred between vehicles in order to update cluster information which increases cluster overhead.

5.4.3 Routing Algorithm for Dissemination of Information from Cluster Head to Control Center:

In this thesis, it is assumed that there are a number of base stations along the road to receive the information from the CH and relay it to the control center. In order to send aggregated information from the CH to the control center, we have implemented two different methods.

Store, Carry, and Forward

In this method, the CH keeps the aggregate information received from member nodes, until it arrives into the communication range of a base station. Then it will send the information to the base station. Depending on the road conditions, network density, and the number of base stations, delay may increase rapidly in this method.

Multi-hop Routing

Using this method, the CH forwards the aggregated packets through multi-hop routing every data time interval. In case the CH broadcasts target's information out of cluster through multiple hops without acquiring knowledge of the network, there is a high probability of collision, packet loss. Therefore, we need to implement a method to avoid such problems. We have used the concept of control packet transmission to acquire information about the neighboring nodes before sending out the data messages. In this method, the Cluster head broadcasts a control message to its neighborhood and will send the data messages only if it receives an acknowledgement from a node. It is possible that the CH receives more than one acknowledgement. In this case, it calculates the distance between the nodes and the closest base station and choses the closest node to the base station as the forwarder node and sends aggregated information to that node.

We have implemented and tested both methods under various numbers of base stations for both proposed algorithms. The simulation results are represented in Section 5.5.3.

5.5 Simulation Results

We have implemented and tested the proposed algorithms under different scenarios in order to represent the effects of different parameters on clustering performance. As well, we have compared performance of both proposed algorithms with a traditional VANET clustering algorithm called MDMAC which is adapted for target tracking purpose. The simulation results are presented in this chapter. It is noteworthy to mention that the proposed algorithms require all vehicles to be equipped with cameras and specific wireless and networking technologies for vehicle localization and communication purposes respectively which may be a future advancement.

5.5.1 DCTT Algorithm Results

In this section the simulation results of DCTT algorithm under different scenarios are presented. The simulation assumptions are displayed in Table 6.

Table 6. Simulation assumptions

Parameter	Value
Simulation environment	Highway
Simulation environment length	10 km
Simulation Time	600 sec
Number of nodes	50, 100, 150, 200
Data packet length	1000 Byte
Data packet frequency	0.5 Hz
Control packet frequency	1 Hz
Transmission rate	1 Mbps
Communication range	50, 100, 250, 500 meter
Vehicle speed	25 - 35 m/s
Traffic type	UDP
Number of base stations	2 - 100
Mac protocol	IEEE 802.11

Effects of network density

We have implemented DCTT algorithm under various node numbers to demonstrate the effects of network density on clustering performance. The simulation parameters are illustrated in Table 7.

Table 7. DCTT Simulation assumptions under various vehicle numbers

Node numbers	50, 100, 150, 200
Transmission Range	100 meters
Velocity range	25-35 meters/sec

Number of base stations	50

Figure 18 displays the effect of number of nodes on the CH lifetime metric. In this algorithm, we have considered a threshold for changing the CH. This threshold has a substantial impact on CH lifetime. The threshold is defined in a way to decrease changes as much as possible. Therefore, unlike other algorithms, when the number of nodes increases, the CH lifetime will not decrease. Besides, the CH lifetime may increase when network density is higher. The evaluation results in Figure 18 displays that increasing the number of nodes has a positive effect on the CH lifetime. The reason is appropriate CH selection metric which is not affected so much by cluster structure changes because the selected CH is a node with the most similar movement pattern to the target.

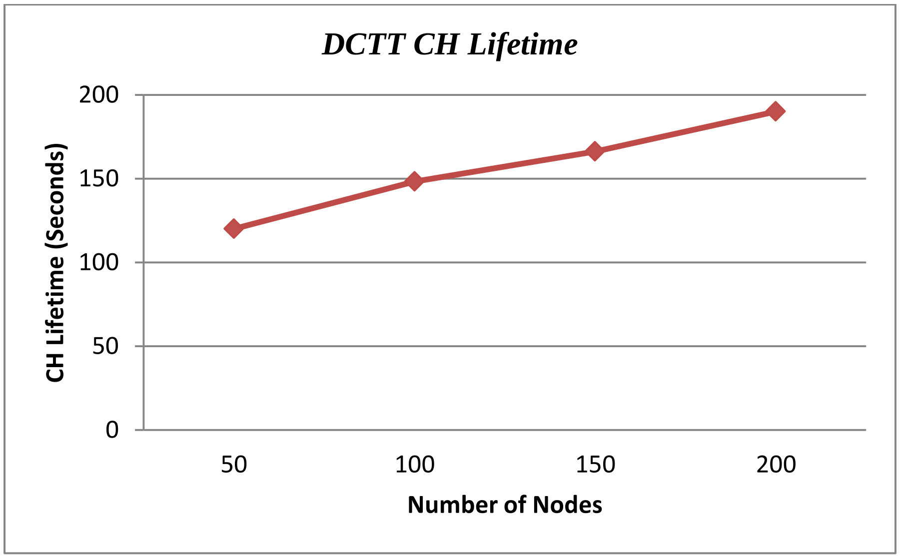


Figure 18. DCTT CH Lifetime under Different Numbers of Nodes

Figure 19 represents the effects of network density changes on packet delivery ratio. In dense networks more vehicles are capable of detecting the target. Therefore, the number of cluster members increase which results in more data message transmission in the cluster. As the number of messages increase, the probability of packet collision increases as well. As a result, packet delivery ratio drops. Here we have considered carry and forward method for packet delivery from the CH to the base station.

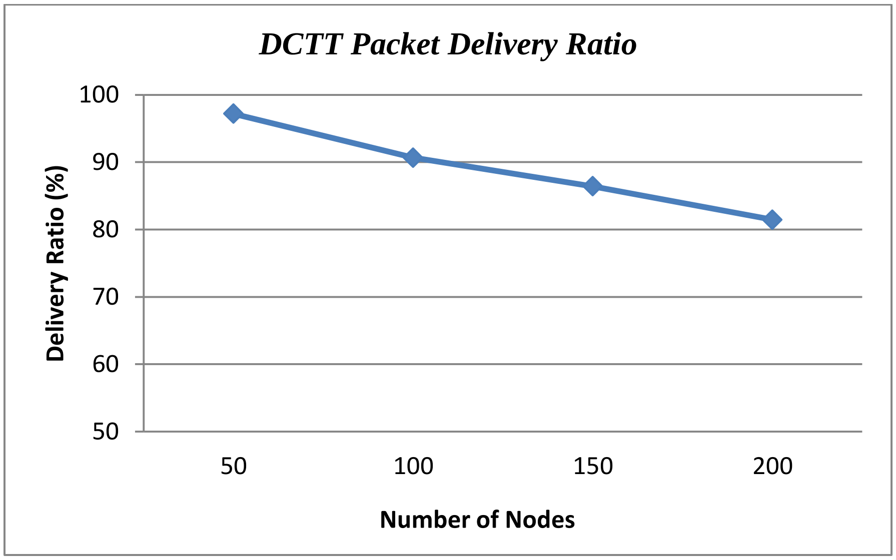


Figure 19. DCTT Packet Delivery Ratio under Different Numbers of Nodes

Figure 20 displays effects of number of nodes on clustering overhead. As the number of nodes increase in the network, the number of cluster members increase consequently. The more number of cluster members send more control messages in the cluster that results in increased control overhead. However, as compared to MDMAC algorithm in Section 5.5.4 the clustering overhead of DCTT is lower which represents better performance of DCTT algorithm.

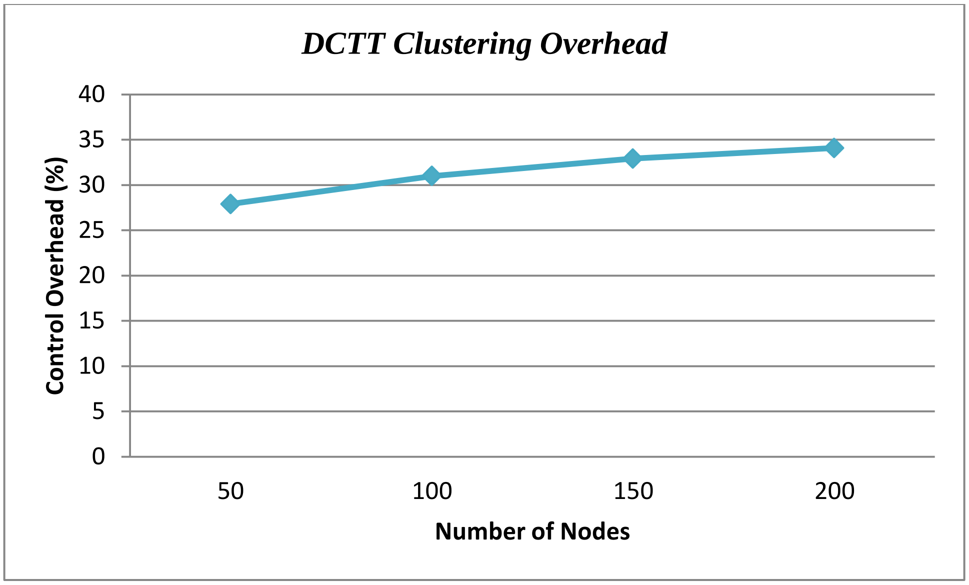


Figure 20. DCTT Clustering Overhead under Different Numbers of Nodes

Effect of TFP Change Threshold

TFP value is the CH selection metric as described in in Section 3.2. A node with the lowest TFP value is selected as CH. The TFP value of member nodes changes as their movement parameter change during the simulation period. The current CH is responsible for selecting the best CH at each time interval. However, if we do not define a threshold for changing the CH, the changes will increase significantly. By defining a change threshold we decrease the number of CH changes. However, defining a very high threshold causes inaccuracy in CH selection and affects the protocol performance negatively. Considering the simulation results we conclude that a threshold value higher than 10 would affect the protocol performance negatively by causing inaccuracy in CH selection. Figure 21 displays the effects of TFP threshold on CH lifetime.

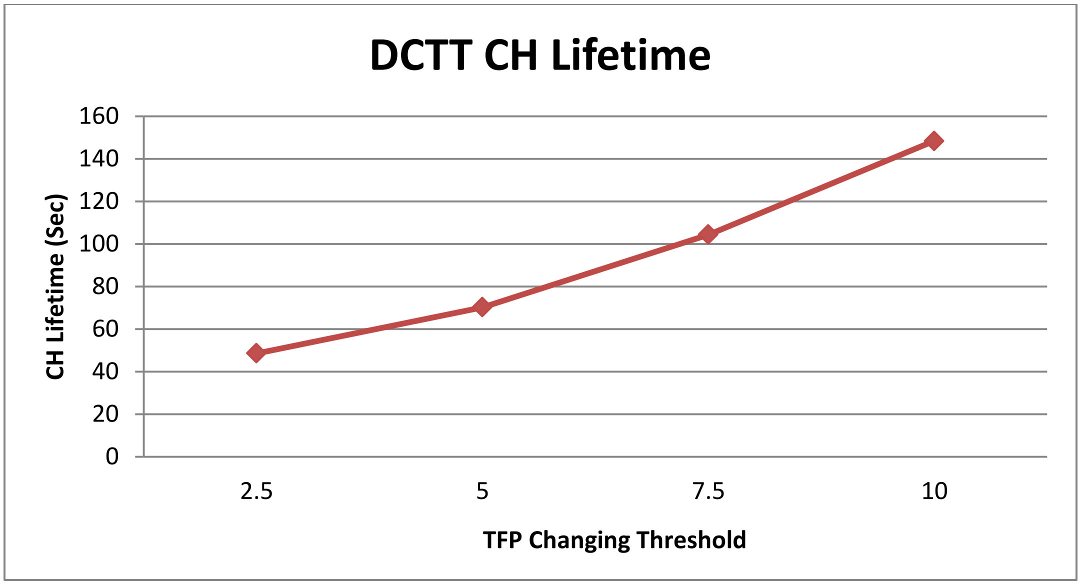


Figure 21. Effects of TFP Threshold on CH Lifetime in DCTT Protocol

Effect of Transmission Range

The simulation assumptions for this scenario are represented in Table 8.

Table 8. DCTT Simulation assumptions under various transmission ranges

Transmission Range	50, 100, 250, 500 meters
Node numbers	100
Data message frequency	0.5 Hz
Control message frequency	1 Hz
Velocity range	25-35 meters/sec

The impact of increasing transmission range on clustering performance is positive. As mentioned in [100], communication range up to 1000 meters is accepted in IEEE 802.11p. It has been cited in [101] that an efficient communication range for WAVE is approximately between 100 to 300 meters. The maximum transmission range in this thesis is assumed to be 500 meters. By increasing the transmission range, the cluster's size increases as well. Therefore, more vehicles join the cluster and stay in the cluster for a longer time period. As a result, CH lifetime and CM lifetime will increase as displayed in Figure 23 and Figure 24. Besides, successful message deliveries inside the cluster and from the CH to the closest base station will increase as illustrated in Figure 22. Increasing the transmission ranges of nodes would help increase the covered areas in the cluster by member nodes. Therefore, packet drops due to unavailability of an intermediate node will decrease which results in higher packet delivery ratio.

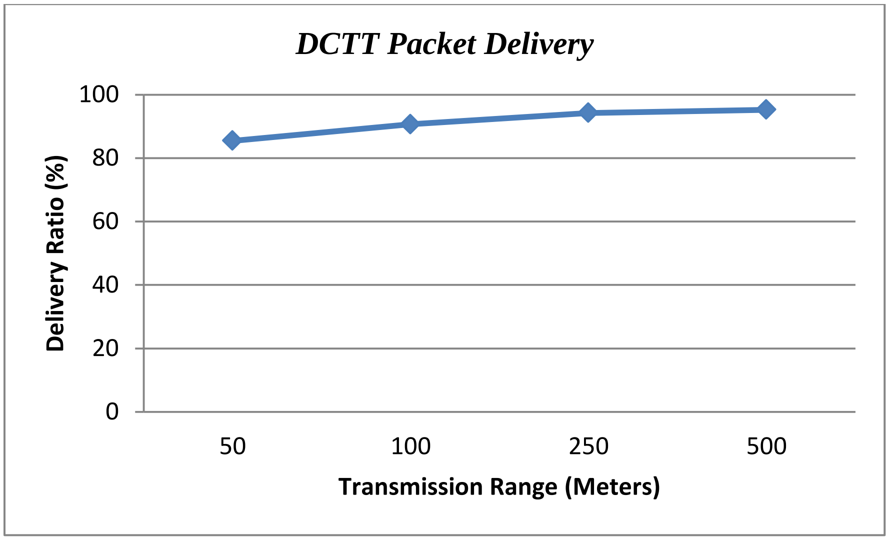


Figure 22. DCTT Packet Delivery Ratios under Various Transmission Ranges

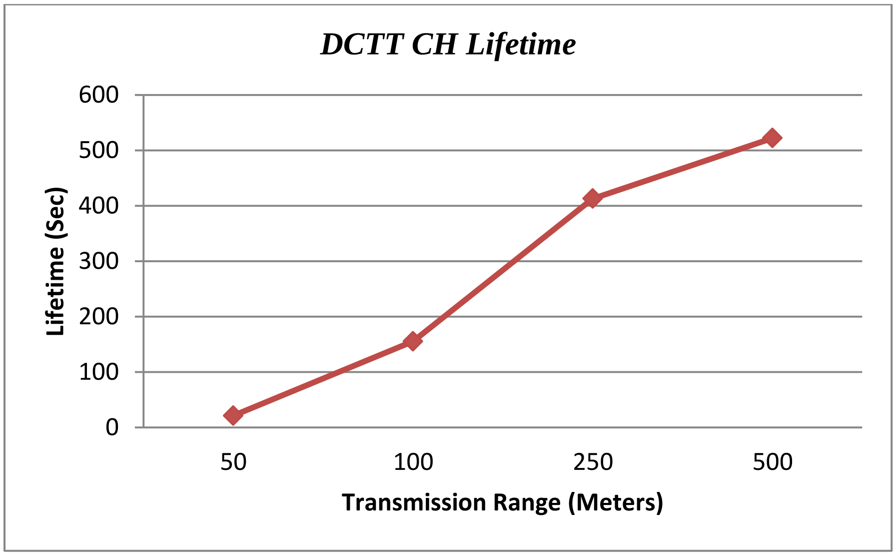


Figure 23. DCTT CH Lifetime under Various Transmission Ranges

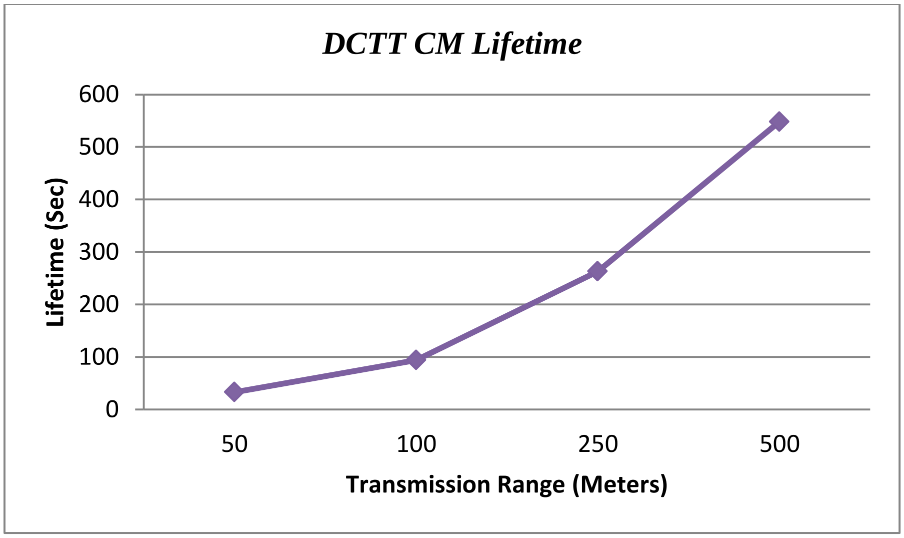


Figure 24. DCTT CM Lifetime under Various Transmission Ranges

Effect of Maximum Velocity

In this section we evaluate the effects of speed range. The simulation assumptions are presented in Table 9.

Table 9. DCTT Simulation assumptions under various maximum velocities

Velocity range	25-70 meters/sec
Transmission Range	100 meters
Node numbers	100

The velocity difference between vehicles in a cluster is an important reason for fast topological changes in the cluster. As well, high velocity of vehicles causes instability in the cluster structure. Therefore, clustering performance of a VANET clustering protocol is degraded when vehicles move faster. Figure 25 and Figure 26 show the effect of maximum velocity change on CH and CM lifetime. The number of CH and CM changes increase as the maximum velocity increases. As a result, the CH lifetime and the CMs lifetime decrease which reduces cluster stability.

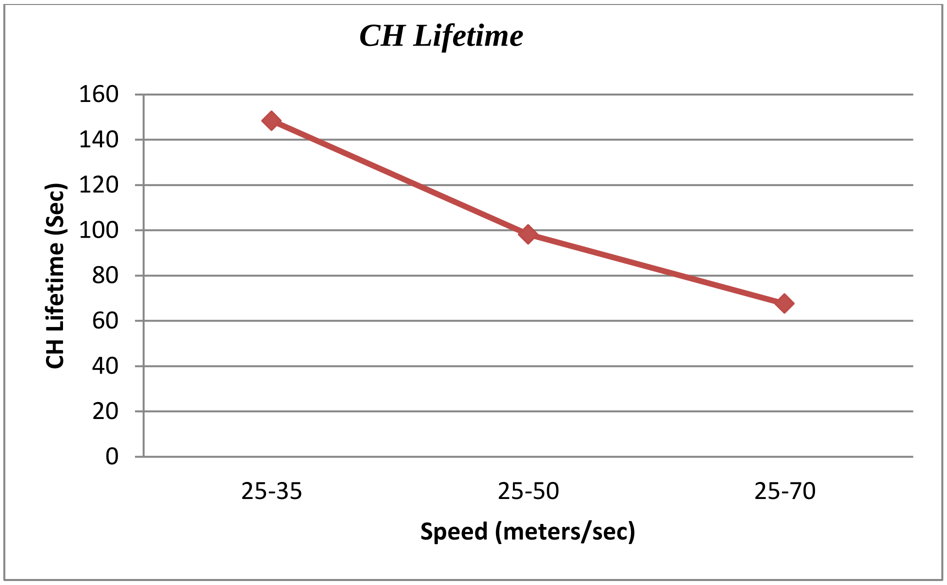


Figure 25. DCTT CH Lifetime under Various Speed Ranges

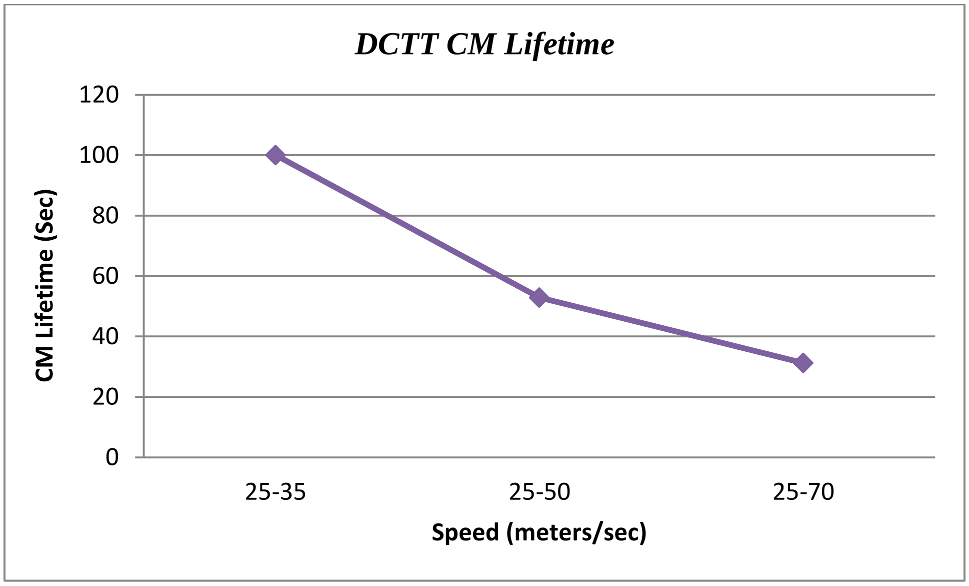


Figure 26. DCTT CM Lifetime under Various Speed Ranges

5.5.2 PCTT Algorithm Results

We have simulated PCTT algorithm under various network densities and various speed ranges. The results are presented in this section.

Effects of network density

The simulation assumptions for PCTT algorithm under various network densities are defined in Table 10. **PCTT Simulation assumptions under various node numbers**. Besides, the general simulation assumptions are illustrated in Table 6. Simulation assumptions

Table 10. PCTT Simulation assumptions under various node numbers

Node numbers	50, 100, 150, 200
Transmission Range	100 meters
Velocity range	25-35 meters/sec
Number of base stations	50
Reset Time	5 Sec

The results represented in Figure 27, Figure 28, and Figure 29 show the same trend as compared to DCTT algorithm under various network densities. However, the results have been improved because of the prediction-based CH selection and prediction-based cluster management techniques employed by PCTT algorithm.

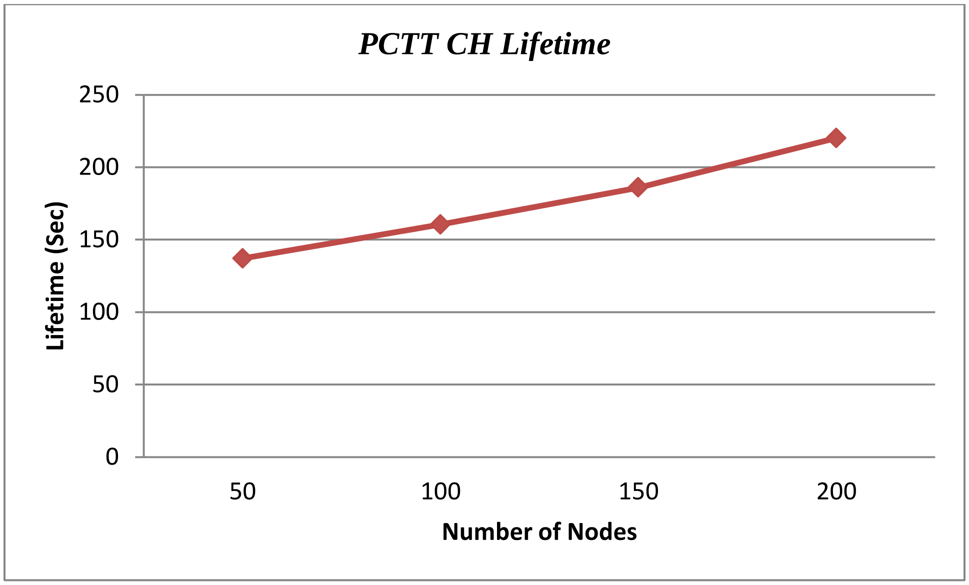


Figure 27. PCTT Cluster Head Lifetime under Various Node Numbers

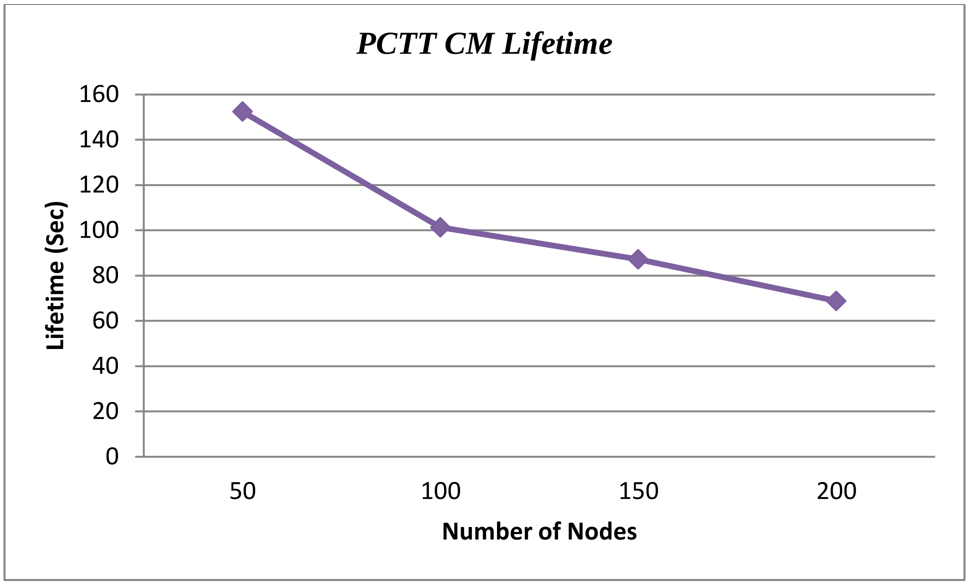


Figure 28. PCTT Cluster Member Lifetime under Various Node Numbers

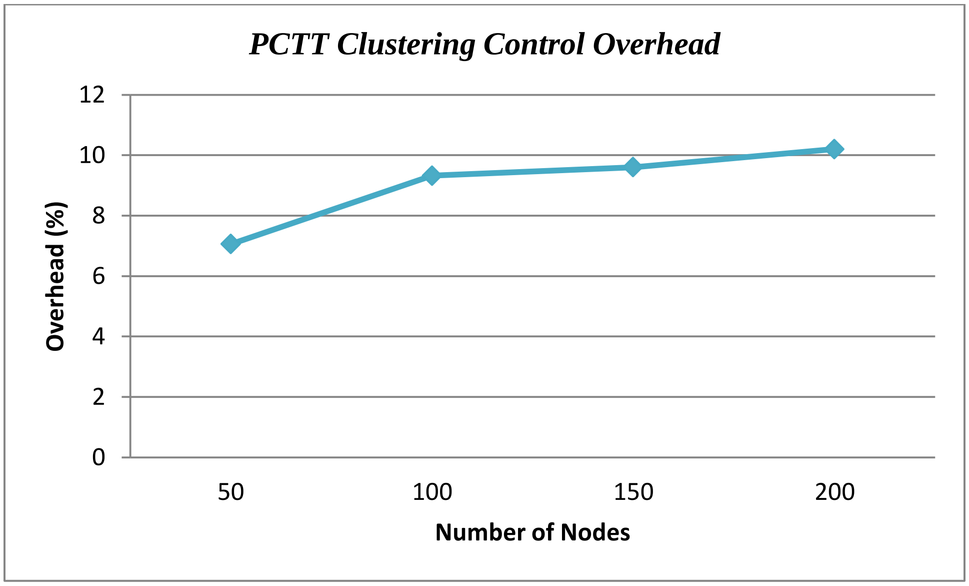


Figure 29. PCTT Clustering Overhead under Various Node Numbers

5.5.3 Results of Routing Algorithm for transferring messages from the CH to the CC

In this section we evaluate two methods for sending data messages from the CH to the control center. We consider a few numbers of base stations that are connected to the control center and are located along the road to receive target's information from the CH and send it to the control center. We have assumed various numbers of base stations to evaluate protocol performance in different scenario. The methods we use for sending target's information to a base station are carry-and-forward method and multi-hop routing as explained in Section 5.4.3. In this section we represent the simulation results of each method.

Store, Carry, and Forward

We have evaluated this method under different number of nodes and different number of base stations. In this method, the CH receives target's location information from member nodes and aggregates the information. It will not send the information until it arrives into the communication range of a base station. Therefore, if the number of base stations increases, the end-to-end delay metric will decrease as displayed in Figure 30. It is noteworthy that increasing the number of base stations along the road increases network setup cost. Therefore, there is always a trade-off between decreasing the delay and increasing the number of base stations.

When the number of nodes increases in the cluster, average end-to-end delay increases because of more message transmissions from nodes to the CH. Also, the CH gathers these messages and waits to arrive to the communication range of a base station to send the information. A large message requires more time to be transferred from the CH to the base station. Therefore, the average end-to-end delay increases.

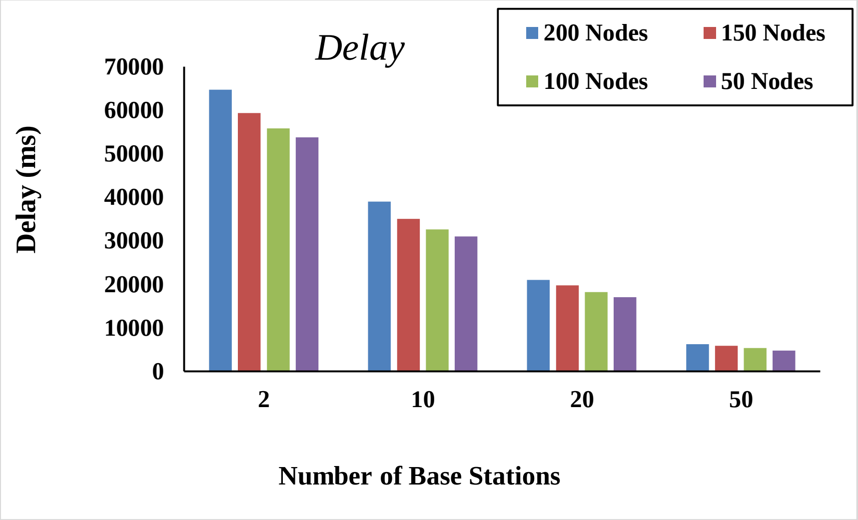


Figure 30. End-to-End Delay of Carry-and-Forward Method under Different Number of Nodes and Base Stations

The delivery ratio of store, carry-and-forward method is represented in Figure 31 under various numbers of base stations and nodes. The results represent that delivery ratio of carry-and-forward method is less than multi-hop method mostly when the number of base stations is low and the network density is high. The reason is the CH should store all the received messages until it arrives into the communication range of a base station. Therefore, when the base stations are located far from each other, it will take a long time for the CH to arrive at the communication range of a base station and send the information.

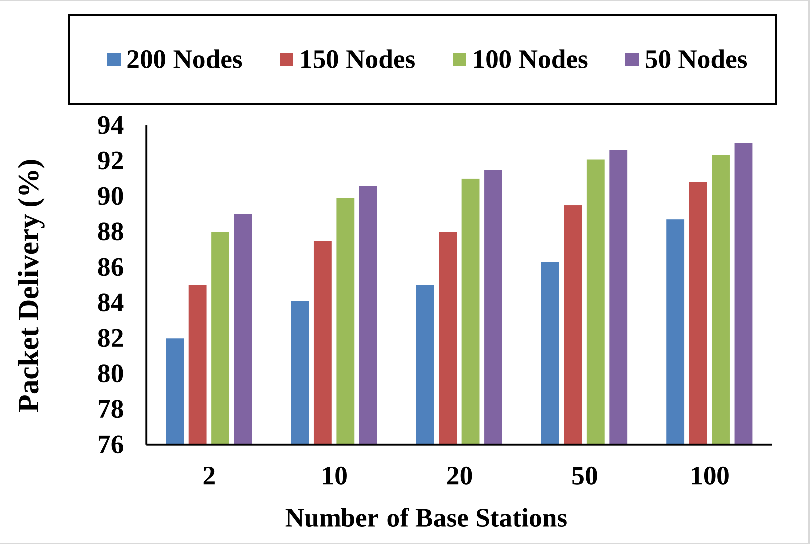


Figure 31. Packet Delivery Ratio of Carry-and-Forward Method under Different Number of Nodes and Base Stations

Multi-Hop Routing

A method that can reduce the delay considerably is multi-hop transmission of the information to the closest base station as soon as the CH processes the data. However, this method may decrease delivery ratio if the information is broadcasted by CH to the neighboring nodes without knowledge of CH's neighborhood. As explained in Section 5.4.3, in order to improve packet delivery, we have used the concept of control message transmission by the CH to acquire knowledge of its neighboring nodes. Therefore, the CH will only send the target's information when it receives an acknowledgement from a neighbor node confirming its availability. Using this method, we have improved packet delivery compared to structure-less algorithms. Besides, the average end-to-end delay is improved as compared to the carry-and-forward scenario.

The effect of number of base stations on packet delivery ratio is displayed in Figure 32. Multi-hop routing technique with control messages guarantees high packet delivery even when the distance between the CH and the next base station is long. The only cost we are adding in order to achieve high delivery and low delay is a little control overhead in the network.

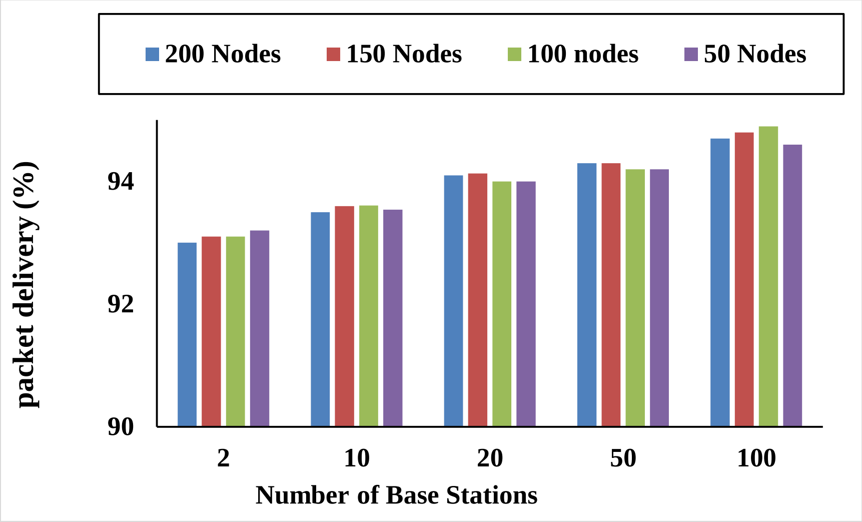


Figure 32. Packet Delivery Ratio of Multi-Hop Routing Method under Different Number of Nodes and Base Stations

The end-to-end delay of multi-hop routing method is displayed in Figure 33. A delicate point in this figure is when the number of base stations are low, the delay increases by decreasing the number of nodes. In VANETs low density scenarios can sometimes have negative effect on performance. For instance, in this scenario the CH checks its neighborhood before sending a message to the next hop. When there is a long way to the next base station, and the density is low, the CH may not find an available neighboring node to send the information. Therefore, it has to wait and find other available nodes. During this time, the information is being buffered which will increase end-to-end delay.

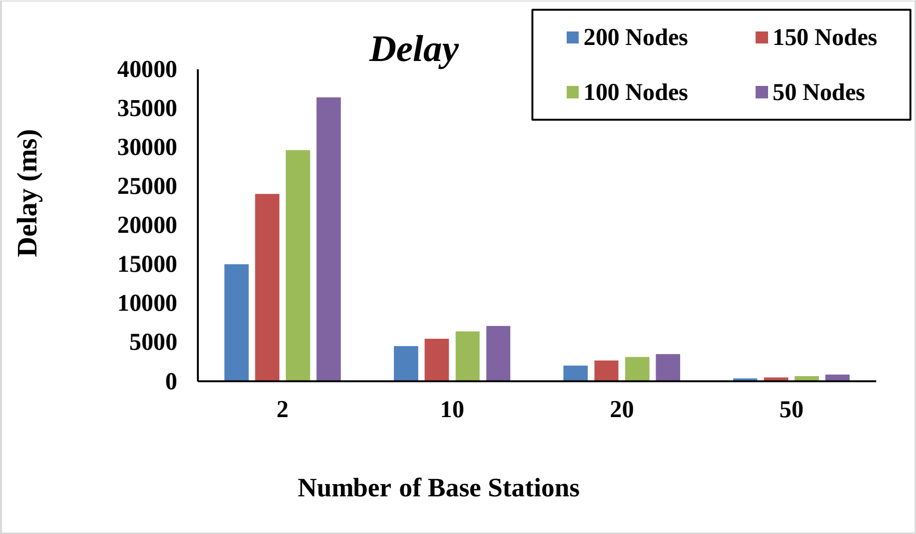


Figure 33. End-to-End Delay of Multi-Hop Routing Method under Different Number of Nodes and Base Stations

We have plotted the packet delivery ratio of both carry and forward and multi-hop routing methods in 3D graph in Figure 34 and Figure 35. The 3D graphs can represent the results more clearly in order to compare the methods in terms of delivery ratio.

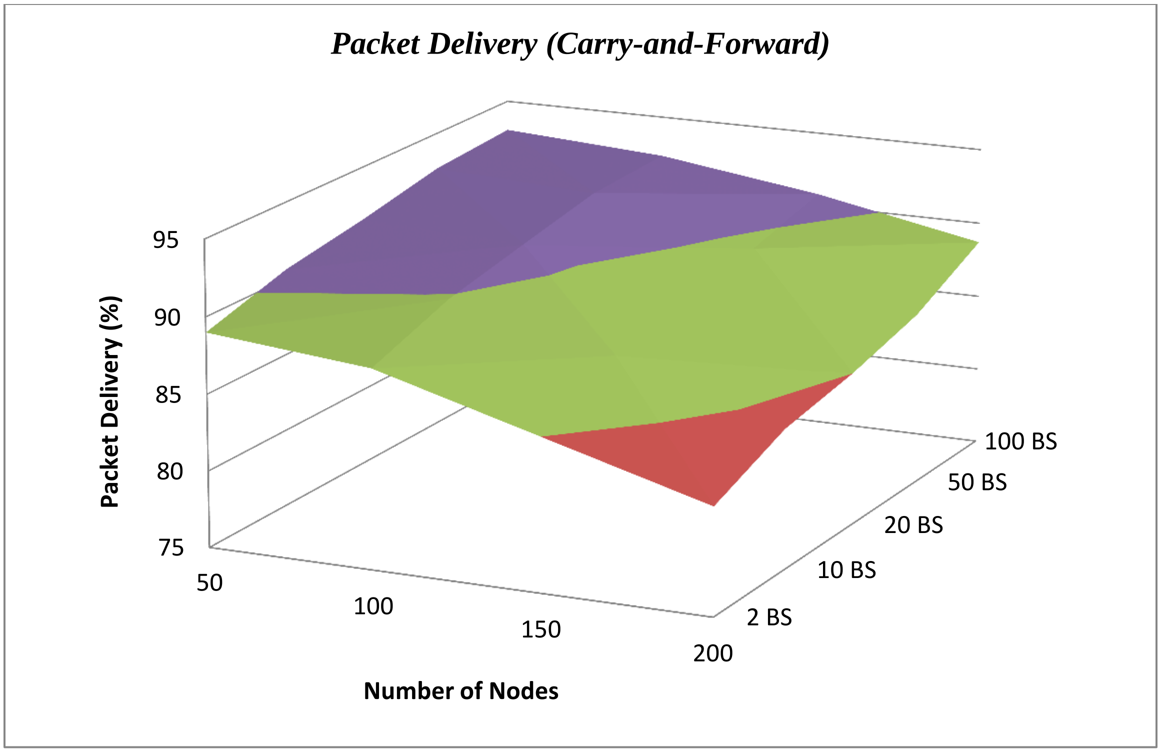


Figure 34. Delivery Ratio of Carry and Forward Method in 3D Graph

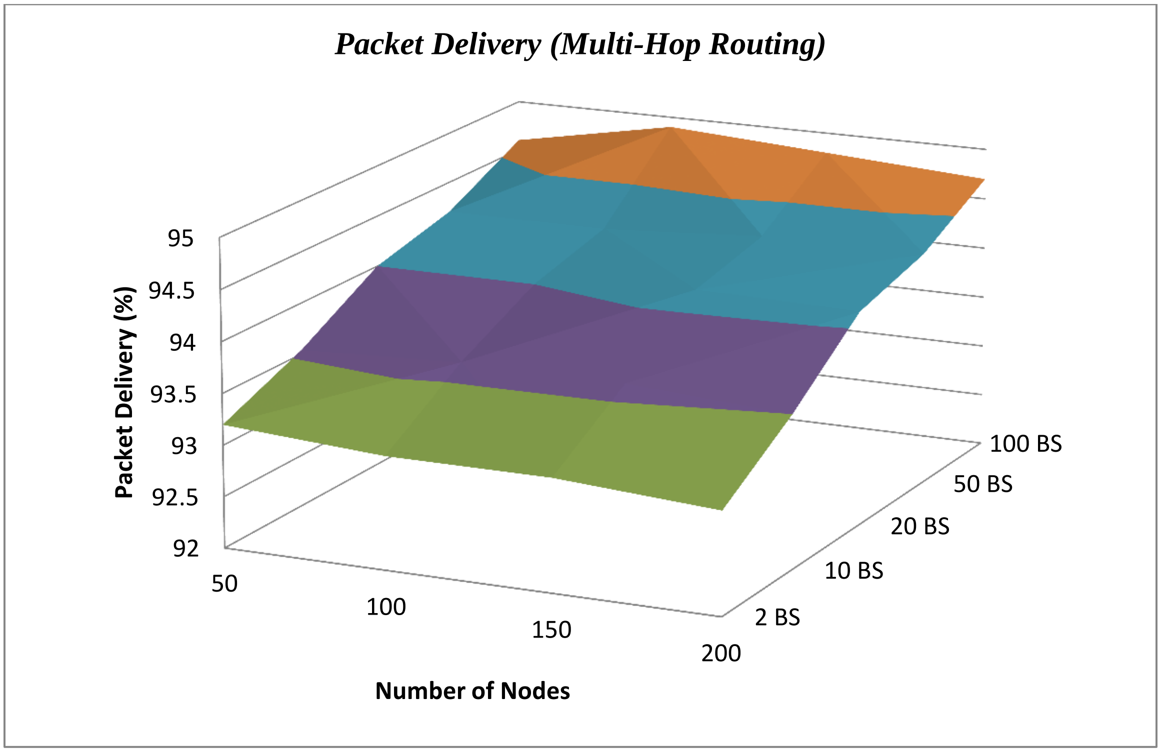


Figure 35. Packet Delivery Ratio of Multi-Hop Routing Method in 3D Graph

5.5.4 Comparison of DCTT, PCTT and MDMAC

We have compared our proposed algorithms DCTT and PCTT with an adapted version of MDMAC algorithm. The simulation results are represented in Figure 36, Figure 37, and Figure 38.

Figure 36 displays significant improvement of clustering control overhead by PCTT algorithm as compared to both DCTT and MDMAC. As explained before, PCTT algorithm benefits from a prediction-based mechanism in both cluster members and cluster head. The cluster head predicts member nodes' behaviour, and the member nodes predict their own behaviour as well. Therefore, a node only sends a control message when its prediction about its own behaviour does not match the real behaviour. This method is so much beneficial in terms of overhead reduction mainly in highway scenarios due to predictable movement of vehicles.

The control overhead of adapted MDMAC protocol is considerably higher than PCTT and DCTT. The reason lies in the need to send control overhead frequently because of the cluster head selection metric requirement. The CH selection metric in this algorithm is node ID. Therefore, nodes need to send their information to the CH as soon as they can so that the CH knows about the memberships and selects the best CH at each time interval. Using node ID as CH selection metric in VANETs affects the clustering performance negatively. Due to very dynamic nature of VANETs it is very important to consider an appropriate CH selection metric which decreases the changes as much as possible.

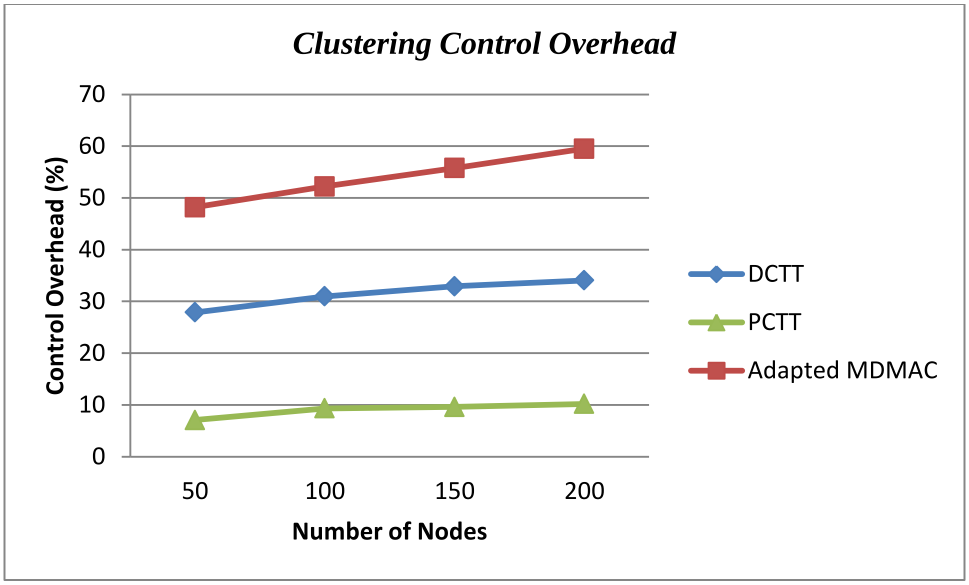


Figure 36. Comparison of Clustering Control Overhead between DCTT, PCTT, and Adapted MDMAC Protocols

Figure 37 displays CH lifetime of the proposed algorithms as compared to adapted MDMAC algorithm. The simulation results represent better performance of PCTT algorithm because of the prediction-based CH selection metric we introduced in Section 4.3. Using this technique, the most appropriate CH which will be an eligible CH for the longest time interval will be selected. Also, the concept of resign timer is defined to prevent the current CH from resigning if the secondary CH is more qualified. In this case, a CH remains in its role as long as it is eligible. The CH lifetime of adapted MDMAC algorithm is lower than all other algorithms because of the CH selection metric. Furthermore, the CM lifetime of the algorithms are displayed in Figure 38. The CM lifetime of PCTT and DCTT algorithm are almost the same. However, the cluster members using MDMAC algorithm have the shortest lifetime because of frequent changes of the cluster structure and the CH changes.

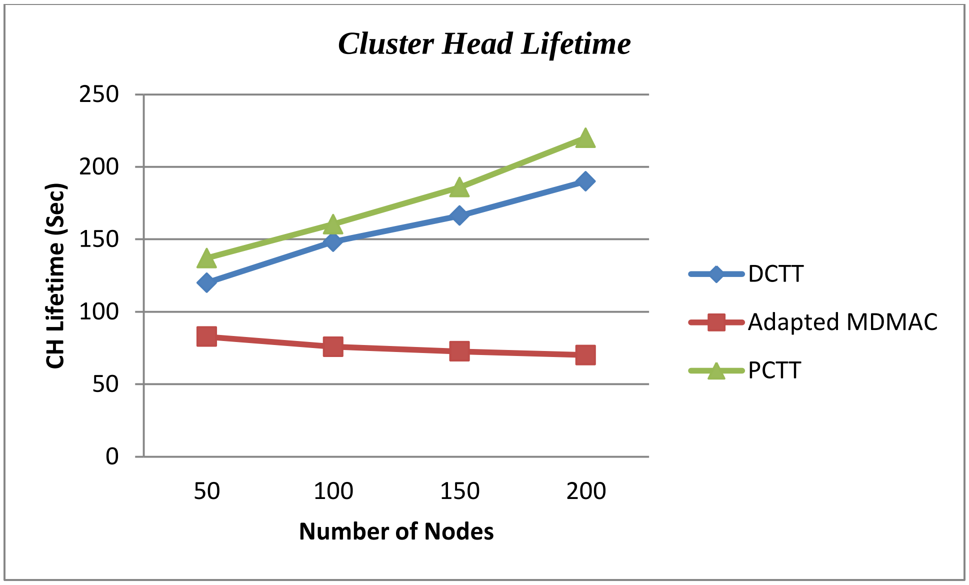


Figure 37. Comparison of Cluster Head Lifetime between DCTT, PCTT, and Adapted MDMAC Protocols

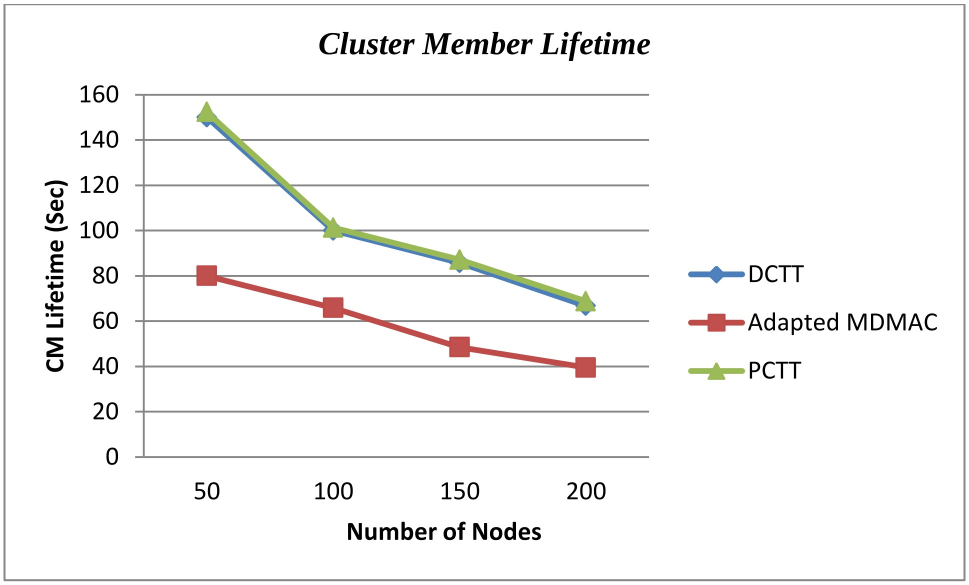


Figure 38. Comparison of Cluster Member Lifetime between DCTT, PCT, and Adapted MDMAC Protocols

5.5.5 Structureless Target Tracking Algorithm Results

In this section, performance of the structure-less carry and forward algorithm is demonstrated in terms of packet delivery ratio. The properties of this algorithm for target tracking are described in Section 5.4.1. As mentioned, using a structure-less algorithm to transfer tracking information to the control center causes packet loss on the base stations side which will result in reduced delivery ratio. This problem has been solved by using the proposed cluster-based algorithms and relying on the cluster head to aggregate the information and forward it to the control center. We have implemented the structure-less algorithm under different node numbers and base station numbers to represent the performance.

Node Number Effect

As the number of node increases, the delivery ratio decreases due to increased number of message broadcasts in the network. In dense areas, more vehicles are capable of detecting the target. Therefore, more packets are broadcasted in the network in order to inform the control center about target's location. Without an appropriate framework, the number of packet collisions increase significantly as the number of nodes increase which results in reduced delivery ratio. The delivery ratio of structure-less algorithm under various node numbers and base station numbers are represented in Figure 39.

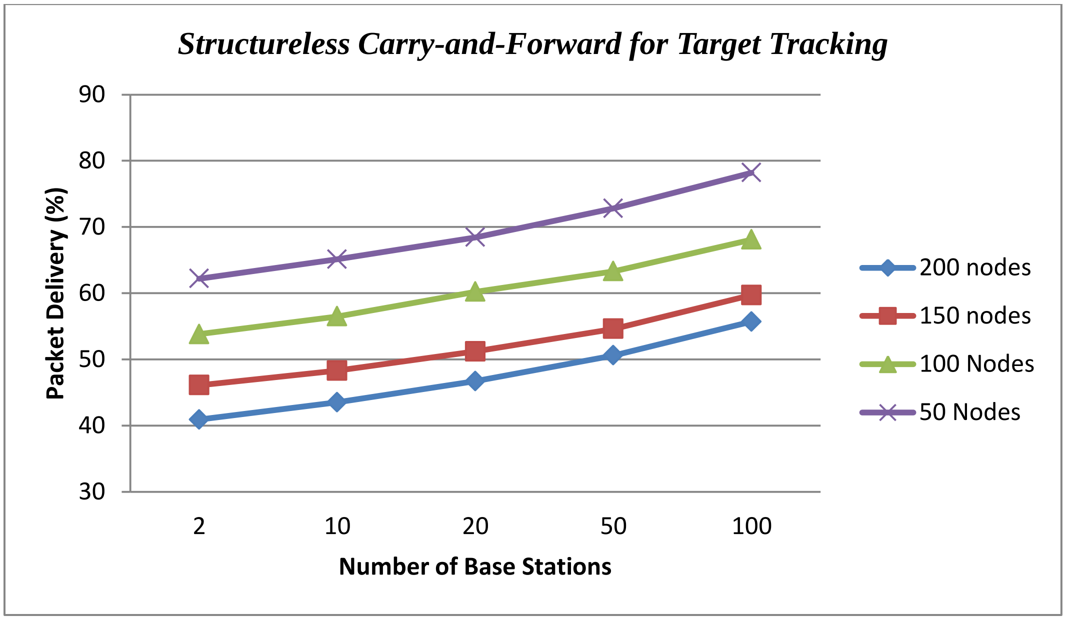


Figure 39. Packet Delivery Ratio of Structureless Carry-and-Forward Algorithm for Target Tracking under Different Number of Nodes and Base Stations (Displaying Node Number Effect)

Base Station Number Effect

Increasing the number of base stations along the highway affects the performance positively. Yet, the cost of installing numerous base stations may be so high which should be taken into consideration in the protocol design.

In the scenario where the base stations are installed at every 5 kilometers distance, delivery ratio for dense networks can be as low as 40% which indicates 60% packet loss. The simulation results are represented in Figure 40.

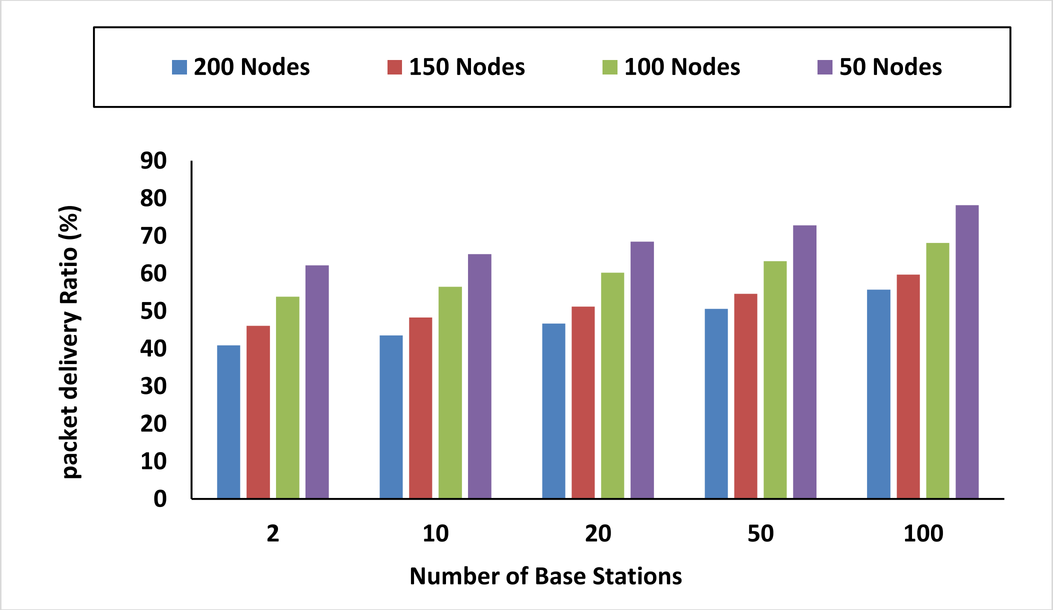
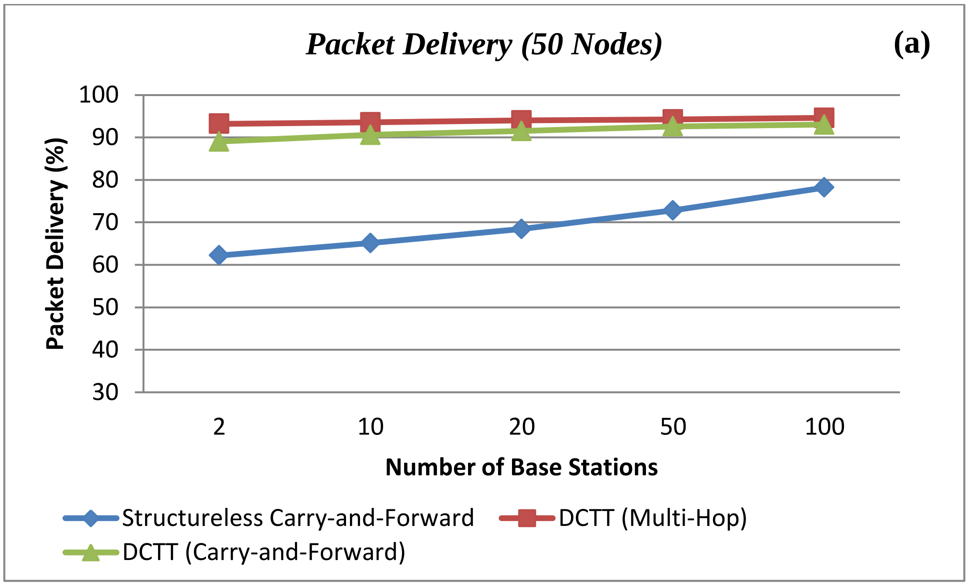


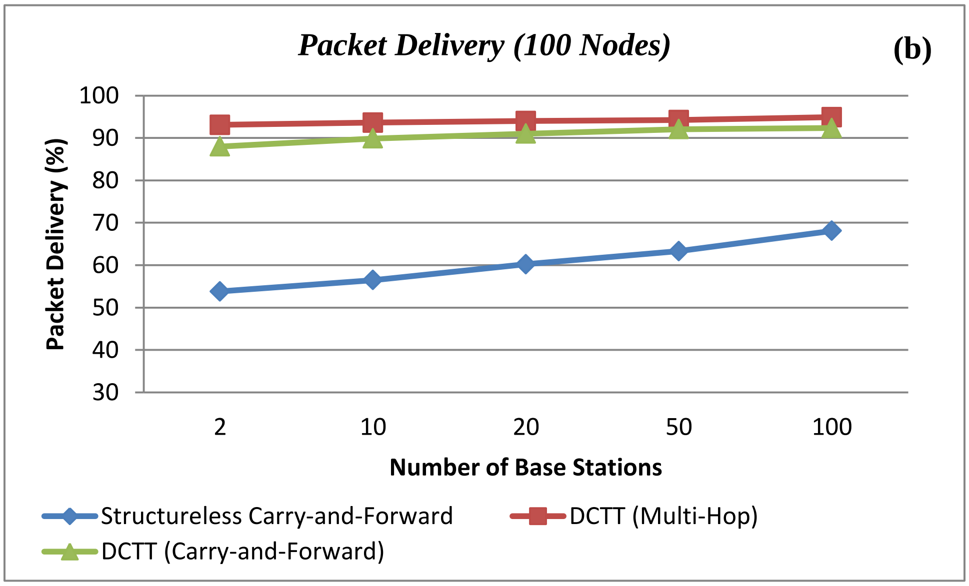
Figure 40. Packet Delivery Ratio of Structureless Carry-and-Forward Algorithm for Target Tracking under Different Number of Nodes and Base Stations (Displaying Base Stations Effect)

5.5.6 Comparison of proposed algorithms with the structureless carry and forward algorithm

In this section we represent results of the clustering algorithms for target tracking as compared to a structureless algorithm in order to display better performance of clustering in terms of packet delivery. As explained in Section 5.4.1, in the structureless algorithm, not any aggregator node such as CH is selected to manage the information and every node is responsible for sending its information separately. Therefore, the probability of collision and packet drops on the base stations side is high which results in low delivery ratio.

In Figure 41, the Structureless carry and forward target tracking algorithm is compared with DCTT algorithm in two different scenarios while transferring information from the CH to the control center. The algorithms have been tested under various numbers of nodes and base stations. In all scenarios, DCTT algorithm shows better performance as compared to using a structureless carry and forward target tracking algorithm. Besides, the performance of DCTT algorithm shows more improvement when multi-hop routing is used for information delivery from the CH to the control center.





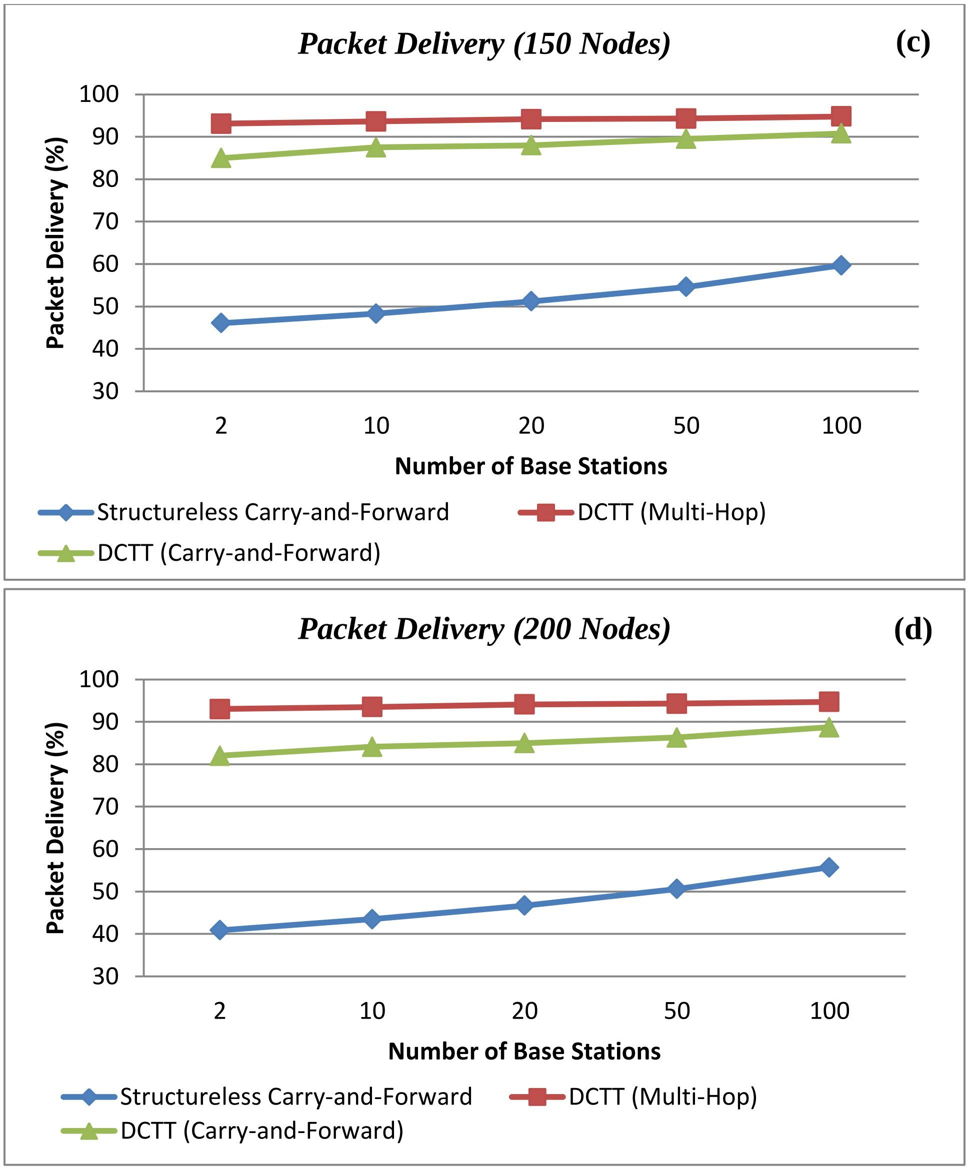


Figure 41. Packet Delivery Ratio Comparison in Structureless Carry-and-Forward Algorithm for Target Tracking, DCTT with Carry and Forward Information Delivery, and DCTT with Multi-Hop Routing Algorithms under Various Number of Base Stations and Different Numbers of Nodes: (a) 50 Nodes, (b) 100 Nodes (c) 150 Nodes (d) 200 Nodes

6. Conclusion and Future Works

6.1 Conclusion

In this thesis, we proposed a cluster-based communication framework for vehicle tracking in VANETs. The major purpose of this framework is to avoid information broadcast and multi-hop data dissemination by each vehicle separately in order to inform the control center about the target. This information can congest the network easily if not managed properly by an appropriate algorithm. We proposed two cluster-based algorithms named DCTT and PCTT. DCTT algorithm is the basic cluster-based target tracking framework that is designed to work in a distributed manner. PCTT algorithm is a centralized and prediction-based algorithm which improves clustering performance considerably. The performance of clustering algorithms is represented in terms of clustering overhead, cluster head lifetime, and cluster member lifetime. The simulation results represent better performance of PCTT algorithm because of its prediction-based cluster maintenance, and cluster head selection mechanisms. As well, the performance results of DCTT algorithm display significant stability and overhead improvement as compared to adapted MDMAC algorithm.

Furthermore, we have tested the vehicle-to-Infrastructure (V2I) communication framework in our algorithms by extending two techniques for information dissemination from the cluster head towards the control center. The carry-and-forward method is compared with a multi-hop routing algorithm. The multi-hop algorithm benefits from control message transmission in order to acquire information about its neighborhood before sending information. The simulation results display considerable performance improvement of the multi-hop routing algorithm in terms of packet delivery and end-to-end delay.

Last but not least, a structureless carry and forward target tracking algorithm for VANETs is implemented so as to demonstrate the necessity of a cluster-based protocol for target tracking in VANETs.

6.2 Future Works

As a future work, the proposed algorithms can be extended for multiple targets tracking and reporting to different central stations. Multiple targets tracking using cluster-based approach requires techniques to manage cluster formation mostly in areas where targets are close to each other. Management of nodes which can participate in both multiple clusters and proper usage of their video information should be considered.

In Section 5.2.3, the concept of semi-passive clustering was introduced. Applying this technique to DCTT and PCTT protocols is a beneficial method to reduce clustering control overhead. As a future work, the concept of semi-passive clustering can be applied to the proposed algorithms for performance improvement.

The other important concept is providing privacy mechanisms in order to protect location and other information of the target and other vehicles from being revealed to unauthorized vehicles and base stations. Location privacy is an important issue in a tracking application, as the only authorized entity to have access to the information, is the central base station which is considered to be a police station. Furthermore, security of such a system should be taken into consideration as a future work in order to prevent malicious nodes from sending false information or disrupting the communication.

Last but not least, performance of PCTT and DCTT algorithm should be evaluated under various city scenarios. In order to extend the proposed algorithm to work properly under city scenario, a prediction procedure to predict target’s future behavior considering the network’s map should be implemented. The concern in cities is the sudden change of route by target when it arrives at junctions, intersections, or exits. The target’s sudden route change may cause the cluster to lose the target. Therefore, re-clustering should be performed which causes delay in the network. In order to prevent this problem the prediction method of the CH can take advantage of map-matching technique to consider various future behaviors of target. In this case the CH will be able to inform the control center to communicate with vehicles in the probable future locations of target and inform them about the target. Therefore, these vehicles can form a cluster in advance considering the information sent from the control center. Other techniques may be applied to extend the proposed algorithms to work properly under city scenarios. Furthermore, implementation of both proposed protocols under real-time video streaming to the control center is recommended. The challenges of this method would be appropriate bandwidth management to avoid network congestion.

References

1. Maia, G., et al. *Data dissemination in urban Vehicular Ad hoc Networks with diverse traffic conditions*. in *Computers and Communications (ISCC), 2013 IEEE Symposium on*. 2013.

2. Tiecheng, W. and W. Gang. *TIBCRPH: Traffic Infrastructure Based Cluster Routing Protocol with Handoff in VANET*. in *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*. 2010.

3. Rezende, C., et al., *A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection.* Ad Hoc Networks, 2014. **17**(0): p. 1-17.

4. Gribben, J. and A. Boukerche, *Location error estimation in wireless ad hoc networks.* Ad Hoc Networks, 2014. **13, Part B**(0): p. 504-515.

5. Boukerche, A., et al., *Vehicular ad hoc networks: A new challenge for localization-based systems.* Computer communications, 2008. **31**(12): p. 2838-2849.

6. Emara, K. *Location privacy in vehicular networks*. in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*. 2013.

7. Abrougui, K. and A. Boukerche, *Efficient group-based authentication protocol for location-based service discovery in intelligent transportation systems.* Security and Communication Networks, 2013. **6**(4): p. 473-484.

8. Hu, X., et al., *A semantics-based multi-agent framework for vehicular social network development*, in *Proceedings of the first ACM international symposium on Design and analysis of intelligent vehicular networks and applications*. 2011, ACM: Miami, Florida, USA. p. 87-96.

9. Rigolin Ferreira Lopes, R., et al. *Social and location-based collaboration mechanism to manage wireless connectivity context data*. in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. 2012.

10. Weicheng, Z., et al. *Multilevel Cluster-Based Information Fusion in Vehicle Ad Hoc Networks*. in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. 2013.

11. Chung-Ming, H., L. Tzu-Hua, and T. Kuan-Cheng. *Bandwidth aggregation over VANET using the geographic member-centric routing protocol (GMR)*. in *ITS Telecommunications (ITST), 2012 12th International Conference on*. 2012.

12. Balasubramaniyan, R. and M. Chandrasekaran. *A new fuzzy based clustering algorithm for wireless mobile Ad-Hoc sensor networks*. in *Computer Communication and Informatics (ICCCI), 2013 International Conference on*. 2013.

13. Xiaoxiao, J., C. Xiang, and D.H.C. Du. *Multihop transmission and retransmission measurement of real-time video streaming over DSRC devices*. in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*. 2014.

14. Qadri, N.N., et al., *Multi-source video streaming in a wireless vehicular ad hoc network.* Communications, IET, 2010. **4**(11): p. 1300-1311.

15. Gerla, M., T. Kwon, and P. Guangyu. *On-demand routing in large ad hoc wireless networks with passive clustering*. in *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE*. 2000.

16. Sheng-Shih, W. and L. Yi-Shiun. *Performance evaluation of passive clustering based techniques for inter-vehicle communications*. in *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*. 2010.

17. Wolny, G. *Modified DMAC Clustering Algorithm for VANETs*. in *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on*. 2008.

18. Cadger, F., et al., *A Survey of Geographical Routing in Wireless Ad-Hoc Networks.* Communications Surveys & Tutorials, IEEE, 2013. **15**(2): p. 621-653.

19. Lima-Filho, A.C., et al., *Embedded System Integrated Into a Wireless Sensor Network for Online Dynamic Torque and Efficiency Monitoring in Induction Motors.* Mechatronics, IEEE/ASME Transactions on, 2012. **17**(3): p. 404-414.

20. Alam Bhuiyan, M.Z., et al. *Energy and bandwidth-efficient Wireless Sensor Networks for monitoring high-frequency events*. in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*. 2013.

21. Thakkar, A. and K. Kotecha, *Cluster Head Election for Energy and Delay Constraint Applications of Wireless Sensor Network.* Sensors Journal, IEEE, 2014. **14**(8): p. 2658-2664.

22. Alippi, C., et al., *An Adaptive Sampling Algorithm for Effective Energy Management in Wireless Sensor Networks With Energy-Hungry Sensors.* Instrumentation and Measurement, IEEE Transactions on, 2010. **59**(2): p. 335-344.

23. Demigha, O., W.K. Hidouci, and T. Ahmed, *On Energy Efficiency in Collaborative Target Tracking in Wireless Sensor Network: A Review.* Communications Surveys & Tutorials, IEEE, 2013. **15**(3): p. 1210-1222.

24. Gupta, S.K.S. and P.K. Srimani, *Adaptive core selection and migration method for multicast routing in mobile ad hoc networks.* Parallel and Distributed Systems, IEEE Transactions on, 2003. **14**(1): p. 27-38.

25. Liu, B., *Next generation vehicular traffic management enabled by vehicular ad hoc networks and cellular mobile devices*. 2011, University of California at Davis. p. 113.

26. Ren-Junn, H., H. Yu-Kai, and L. Yen-Fu. *Secure Communication Scheme of VANET with Privacy Preserving*. in *Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on*. 2011.

27. Yi, Q. and N. Moayeri. *Design of Secure and Application-Oriented VANETs*. in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. 2008.

28. Chang, Y.-T., et al., *A survey of handoff schemes for vehicular ad-hoc networks*, in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. 2010, ACM: Caen, France. p. 1228-1231.

29. Gerla, M. and L. Kleinrock, *Vehicular networks and the future of the mobile internet.* Comput. Netw., 2011. **55**(2): p. 457-469.

30. Yousefi, S., M.S. Mousavi, and M. Fathy. *Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives*. in *ITS Telecommunications Proceedings, 2006 6th International Conference on*. 2006.

31. Hartenstein, H. and K.P. Laberteaux, *A tutorial survey on vehicular ad hoc networks.* Communications Magazine, IEEE, 2008. **46**(6): p. 164-171.

32. Nandan, A., et al. *Co-operative downloading in vehicular ad-hoc wireless networks*. in *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*. 2005.

33. Nandan, A., et al. *AdTorrent: digital billboards for vehicular networks*. in *Proc. of IEEE/ACM International Workshop on Vehicle-to-Vehicle Communications (V2VCOM), San Diego, CA, USA*. 2005.

34. Rong, D., et al. *VANET based traffic estimation: A matrix completion approach*. in *Global Communications Conference (GLOBECOM), 2013 IEEE*. 2013.

35. Gramaglia, M., M. Calderon, and C.J. Bernardos, *ABEONA Monitored Traffic: VANET-Assisted Cooperative Traffic Congestion Forecasting.* Vehicular Technology Magazine, IEEE, 2014. **9**(2): p. 50-57.

36. Radu, D., et al. *Acoustic noise pollution monitoring in an urban environment using a VANET network*. in *Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on*. 2012.

37. Mohan, R.S., R. Sachin, and U. Sakthivel. *Vehicular Ad Hoc Network Based Pollution Monitoring in Urban Areas*. in *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on*. 2012.

38. Noshadi, H., et al. *Remote Medical Monitoring Through Vehicular Ad Hoc Network*. in *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*. 2008.

39. Khakpour, S., R.W. Pazzi, and K. El-Khatib, *A distributed clustering algorithm for target tracking in vehicular ad-hoc networks*, in *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*. 2013, ACM: Barcelona, Spain. p. 145-152.

40. Khakpour, S., R.W. Pazzi, and K. El-Khatib, *A prediction based clustering algorithm for target tracking in vehicular ad-hoc networks*, in *Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications*. 2014, ACM: Montreal, QC, Canada. p. 39-46.

41. Chaurasia, B.K., et al. *Suitability of MANET Routing Protocols for Vehicular Ad Hoc Networks*. in *Communication Systems and Network Technologies (CSNT), 2012 International Conference on*. 2012.

42. Tayal, S. and M.R. Tripathy. *VANET-Challenges in Selection of Vehicular Mobility Model*. in *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on*. 2012.

43. Basu, P., N. Khan, and T.D.C. Little. *A mobility based metric for clustering in mobile ad hoc networks*. in *Distributed Computing Systems Workshop, 2001 International Conference on*. 2001.

44. Maglaras, L.A. and D. Katsaros. *Distributed clustering in vehicular networks*. in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*. 2012.

45. Ahizoune, A. and A. Hafid. *A new stability based clustering algorithm (SBCA) for VANETs*. in *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*. 2012.

46. Zhang, Z., A. Boukerche, and R. Pazzi, *A novel multi-hop clustering scheme for vehicular ad-hoc networks*, in *Proceedings of the 9th ACM international symposium on Mobility management and wireless access*. 2011, ACM: Miami, Florida, USA. p. 19-26.

47. Hang, S. and Z. Xi, *Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular <emphasis emphasistype="italic">Ad Hoc</emphasis> Networks.* Vehicular Technology, IEEE Transactions on, 2007. **56**(6): p. 3309-3323.

48. Wei, F., et al. *A mobility metrics based dynamic clustering algorithm for VANETs*. in *Communication Technology and Application (ICCTA 2011), IET International Conference on*. 2011.

49. Tizghadam, A. and A. Leon-Garcia, *Autonomic traffic engineering for network robustness.* Selected Areas in Communications, IEEE Journal on, 2010. **28**(1): p. 39-50.

50. Rawshdeh, Z.Y. and S.M. Mahmud. *Toward Strongley Connected Clustering Structure in Vehicular Ad Hoc Networks*. in *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*. 2009.

51. Hafeez, K.A., et al. *A fuzzy-logic-based cluster head selection algorithm in VANETs*. in *Communications (ICC), 2012 IEEE International Conference on*. 2012.

52. Shea, C., B. Hassanabadi, and S. Valaee. *Mobility-Based Clustering in VANETs Using Affinity Propagation*. in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. 2009.

53. Zhenxia, Z., A. Boukerche, and R.W. Pazzi. *A Novel Network Mobility Management Scheme for Vehicular Networks*. in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. 2010.

54. Jiang, M., J. Li, and Y.C. Tay, *Cluster Based Routing Protocol ({CBRP}) Functional Specification*. 1998.

55. Bo, L., et al., *Component-Based License Plate Detection Using Conditional Random Field Model.* Intelligent Transportation Systems, IEEE Transactions on, 2013. **14**(4): p. 1690-1699.

56. Kai-Hsiang, L., T. Hao, and T.S. Huang. *Robust license plate detection using image saliency*. in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. 2010.

57. Kai, Z., et al. *Hybridization of appearance and symmetry for vehicle-logo localization*. in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. 2012.

58. Shuyuan, Y., et al. *Vehicle logo recognition based on Bag-of-Words*. in *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*. 2013.

59. Zhou, W., et al., *Principal Visual Word Discovery for Automatic License Plate Detection.* Image Processing, IEEE Transactions on, 2012. **21**(9): p. 4269-4279.

60. Ramos, H.S., et al., *Cooperative target tracking in vehicular sensor networks.* Wireless Communications, IEEE, 2012. **19**(5): p. 66-73.

61. Yuexian, Z., et al. *A moving vehicle segmentation method based on clustering of feature points for tracking at urban intersection*. in *Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on*. 2010.

62. Ya, L., et al. *Optical Flow Based Urban Road Vehicle Tracking*. in *Computational Intelligence and Security (CIS), 2013 9th International Conference on*. 2013.

63. Sivaraman, S. and M.M. Trivedi, *A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking.* Intelligent Transportation Systems, IEEE Transactions on, 2010. **11**(2): p. 267-276.

64. Wen-hui, L., et al. *Co-training algorithm based on on-line boosting for vehicle tracking*. in *Information and Automation (ICIA), 2013 IEEE International Conference on*. 2013.

65. Menard, T., et al. *Comparing the GPS capabilities of the Samsung Galaxy S, Motorola Droid X, and the Apple iPhone for vehicle tracking using FreeSim_Mobile*. in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. 2011.

66. SeokJu, L., G. Tewolde, and K. Jaerock. *Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application*. in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. 2014.

67. Menard, T. and J. Miller. *FreeSim_Mobile: A novel approach to real-time traffic gathering using the apple iPhone&#x2122*. in *Vehicular Networking Conference (VNC), 2010 IEEE*. 2010.

68. *Apple, http://www.apple.com/.*

69. Miller, J. and E. Horowitz. *FreeSim - a free real-time freeway traffic simulator*. in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*. 2007.

70. Montgomery, J. *A real-time traffic and weather reporting system for motorists*. in *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*. 2005.

71. Barani, H. and M. Fathy, *An algorithm for localization in vehicular ad-hoc networks.* Journal of Computer Science, 2010. **6**(2): p. 168.

72. Naderan, M., M. Dehghan, and H. Pedram, *Upper and lower bounds for dynamic cluster assignment for multi-target tracking in heterogeneous WSNs.* Journal of Parallel and Distributed Computing, 2013. **73**(10): p. 1389-1399.

73. Mansouri, M., et al., *Secure and robust clustering for quantized target tracking in wireless sensor networks.* Communications and Networks, Journal of, 2013. **15**(2): p. 164-172.

74. Hajiaghajani, F., et al. *HCMTT: Hybrid clustering for multi-target tracking in Wireless Sensor Networks*. in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. 2012.

75. Gerla, M. and J. Tzu-Chieh Tsai, *Multicluster, mobile, multimedia radio network.* Wireless Networks, 1995. **1**(3): p. 255-265.

76. Weiwei, L., A. Tizghadam, and A. Leon-Garcia. *Robust clustering for connected vehicles using local network criticality*. in *Communications (ICC), 2012 IEEE International Conference on*. 2012.

77. Tal, I. and G.M. Muntean. *User-Oriented Fuzzy Logic-Based Clustering Scheme for Vehicular Ad-Hoc Networks*. in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*. 2013.

78. Basagni, S. *Distributed clustering for ad hoc networks*. in *Parallel Architectures, Algorithms, and Networks, 1999. (I-SPAN '99) Proceedings. Fourth InternationalSymposium on*. 1999.

79. El Ghanami, D., T.J. Kwon, and A. Hafid. *GRIDS: Geographically Repulsive Insomnious Distributed Sensors – An Efficient Node Selection Mechanism Using Passive Clustering*. in *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing*. 2008.

80. Kwon, T., et al., *Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad hoc/sensor networks.* Proceedings of the IEEE, 2003. **91**(8): p. 1210-1220.

81. Hafid, A.S., F. Chender, and T. Kwon. *Energy Aware Passive Clustering in Wireless Mobile Networks*. in *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*. 2008.

82. Rangaswamy, A. and H.K. Pung. *Enhancement of passive cluster based routing protocol for mobile adhoc networks*. in *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*. 2002.

83. Chunhua, Z. and T. Cheng. *A K-Hop Passive Cluster Based Routing Protocol for MANET*. in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*. 2009.

84. Jia, H., et al., *Point of Interest Detection and Visual Distance Estimation for Sensor-Rich Video.* Multimedia, IEEE Transactions on, 2014. **16**(7): p. 1929-1941.

85. Nan, J., L. Wenyu, and W. Ying, *Learning Adaptive Metric for Robust Visual Tracking.* Image Processing, IEEE Transactions on, 2011. **20**(8): p. 2288-2300.

86. Doskocil, R., et al. *Measurement of distance by single visual camera at robot sensor systems*. in *MECHATRONIKA, 2012 15th International Symposium*. 2012.

87. Seung-Eun, Y. and K. DaeEun. *Distance estimation method with snapshot landmark images in the robotic homing navigation*. in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. 2010.

88. *Simulation of Urban Mobility (SUMO), www.dlr.de/ts/sumo.*

89. *OpenStreetMap, http://www.openstreetmap.org/.*

90. *Java OpenStreetMap Editor (JOSM), https://josm.openstreetmap.de/.*

91. *Network Simulator NS-2, http://www.isi.edu/nsnam/ns/.*

92. Levis, P., et al., *TOSSIM: accurate and scalable simulation of entire TinyOS applications*, in *Proceedings of the 1st international conference on Embedded networked sensor systems*. 2003, ACM: Los Angeles, California, USA. p. 126-137.

93. *TinyOS, http://www.tinyos.net/.*

94. Feng, S., et al., *Asymmetric Coding of Multi-View Video Plus Depth Based 3-D Video for View Rendering.* Multimedia, IEEE Transactions on, 2012. **14**(1): p. 157-167.

95. Chakareski, J., *Wireless Streaming of Interactive Multi-View Video via Network Compression and Path Diversity.* Communications, IEEE Transactions on, 2014. **62**(4): p. 1350-1357.

96. Soon-Young, L., et al., *Correspondence Matching of Multi-View Video Sequences Using Mutual Information Based Similarity Measure.* Multimedia, IEEE Transactions on, 2013. **15**(8): p. 1719-1731.

97. Paul, M., et al. *Multi-view video compression using dynamic background frame and 3D motion estimation*. in *Computer and Information Technology (ICCIT), 2012 15th International Conference on*. 2012.

98. Susheel Kumar, K., et al. *Multiple Cameras Using Real Time Object Tracking for Surveillance and Security System*. in *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*. 2010.

99. Xun, G., et al., *Wyner–Ziv-Based Multiview Video Coding.* Circuits and Systems for Video Technology, IEEE Transactions on, 2008. **18**(6): p. 713-724.

100. Bilstrup, K., et al. *Evaluation of the IEEE 802.11p MAC Method for Vehicle-to-Vehicle Communication*. in *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*. 2008.

101. Jafari, A., S. Al-Khayatt, and A. Dogman. *Performance evaluation of IEEE 802.11p for vehicular communication networks*. in *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*. 2012.