Electrical Engineering, Computing and Mathematical Sciences

Broadcasting Protocol for Effective Data Dissemination in Vehicular Ad Hoc Networks

Lakshmanaprakash Sanmugaraja

This thesis is presented for the degree of

Doctor of Philosophy

Of

Curtin University

December 2018

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: ...S. Lenh C...........

Date: .21./.12./.18........

To

My parents, My Uncle, My wife Ramya

and My daughter Manasadevi

Acknowledgements

I would like to express the deepest gratitude to my thesis supervisors Dr. Wong Kiing Ing, and Prof C. Palanichamy and thesis chairperson Associate Prof. Zhuquan Zang for their motivation and the substance of genius, constant support and encouragement during my PhD studies. Without their guidance and persistent help, this research would not have possible.

I would like to extend my appreciation to Prof Clem Kuek, Dean Research and Development, for his necessary administrative guidance and help at all level. In addition to that, I thank Associate Prof Dr. Lenin Gopal for his encouragement and insightful comments for my thesis preparation and completion. I am grateful to Faculty of Engineering and Science, Curtin University Sarawak Malaysia for the financial and administrative support they have provided me during the course of my study.

I am also thankful to all my HDR friends and colleagues, especially Dr.JaisonJeevanandam, Dr.CalebAcquah and Mr.Veeramani Shanmugam for their help and encouragement during my research. Lastly and most importantly, I give grateful acknowledgement to my parents, my in-laws, my wife and my daughter for their moral support in my research.

Abstract

Vehicular Ad-Hoc Networks (VANETs) are distinct types of mobile ad-hoc networks where vehicles are equipped through sensing, computing and communicating equipment. This equipment exchanges messages with each other and with the infrastructure on the roadside units, to make an intelligent communication for road traffic safety and other proficiency purposes. However, the network topology of VANET is based on time and location varying due to the fickle density of the vehicles and road network, respectively. To address the data dissemination challenge, a broadcasting algorithm that works with the local information attained as a beacon message is proposed. The beacon message is analyzed and categorized as either a connected dominating set or not a connected dominating set. These nodes been enhanced with the neighbor elimination method.

In addition, a brief time-out estimation mechanism is used for message retransmissions when one vehicle is detected in the dominating set. A poorly managed mechanism for message dissemination can overflow the network with replicated information and increase the amount of collisions due to disputes between vehicles for accessing the wireless medium. These difficulties are known as broadcast storm problem. However, the previous study has addressed this broadcasting storm issue based on position, statistical distance, local topology, timer, and trajectory or map-based methods. We recommend a modified Broadcast Conquest and Delay De-synchronization mechanism using preference zone identification method. In the proposed modified broadcast suppression mechanism, vehicle inside the preference zone acquires the highest priority to communicate and waiting delay of all vehicles is calculated using the delay de-synchronization approach. Although data dissemination is possible in all direction, the performance of data dissemination in the opposite direction is investigated. With comprehensive simulation-based evaluation, the performance of the proposed method has compared with the modern approaches based on its reliability and messaging efficiency. It is envisioned that the application of this protocol will eventually reduce the time delay of data dissemination, avoids information jamming (broadcast storm) and enhances the performance of VANETs.

Keywords: VANET, Data dissemination, broadcasting storm, connected dominating set.

Table of Contents

Lists of Figures

List of Tables

Conference and Publications

- "Resolving Broadcasting Storm using Enhanced Suppression Mechanism" , at One Curtin International Postgraduate Conference, from 10-12 Dec 2017, Miri.
- "Connected Dominating Sets based Routing Algorithm for Intersections in VANETs", at North Borneo Research Colloquium 2016, 19 Apr 2016, Miri.
- "A Comprehensive Study and Survey of VANET Simulators" , at 10th CUTSE conference , 7 Nov 2015 , Miri.
- "VANET Features and Challenges: A Comprehensive Review " , at University Malaysia Sarawak, FENG Post Graduate Research Colloquium 2015

CHAPTER 1-Introduction and Background

1.1 Introduction to VANET

In the contemporary development of automobile industry, the intelligent transport system (ITSs) has used the innovative wireless communication tools to develop the road transport as a user comfort. ITS bids vehicular ad hoc networks (VANETs) to deliver communications amongst vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) as described by [1].

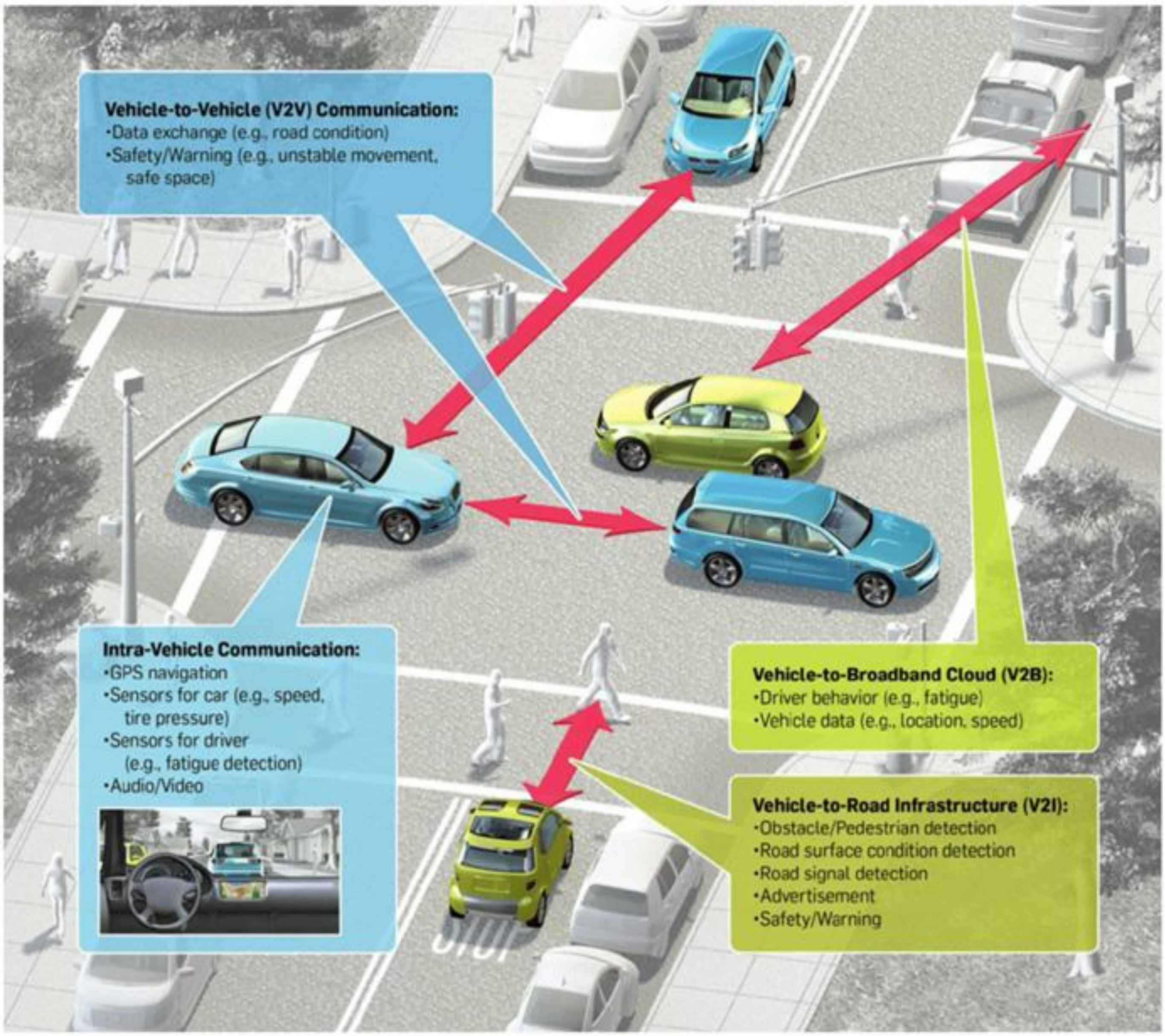


Figure 1.1: VANET Communication Types [3]

Non-profitable organization like CAR-2-CAR Communication Consortium, SAFEPOST and other major automobile industries are constantly looking for the way to improve the road user comfort and safety to the next level as described by [2]. A vehicular communication type of VANET is shown in Figure1.1.

1.2 Technical Specifications

FCC (Federal Communication Commission)has assigned the frequency bands of 75 MHz in 5.9 GHz to licensed DSRC (Dedicated Short Range Communication) in order to increase the bandwidth and reducing the latency of VANETs communications that was reported in [4]. Europe uses 20 MHz from the 5.9 GHz for the applications associated with mobile user safety. Table 1.1 exhibits standards of DSRC used in various countries for ITS systems effectively as described by [5]. According to [6] VANETs enables a new class of applications with high data transfer rates (in the range of 6 to 54Mbps) for vehicles traveling at speeds of around 120 mph. In [7],[8], the WAVE (Wireless Access in the Vehicular Environments) architecture that cooperatively enables both Vehicle-to-Infrastructure (V2I) and vehicle-to-vehicle (V2V) wireless communications is discussed. According to the IEEE 1609 standards, the upper layer handles the operating utilities and complexity related to the Short range communications.

Features	USA	EUROPE	JAPAN
Communication	Half-duplex	Half-duplex	Half-duplex (OBU)/Full duplex (RSU)
Radio Frequency	5.9 GHz	5.8 GHz	5.8 GHz
Band	75 MHz bandwidth	20 MHz bandwidth	80MHz bandwidth
Channels	Downlink: 7	4	7
Channel Separation	10 MHz	5 MHz	5 MHz
Data Transmission rate	Down/Up-link 3-27 4 MBits/s	Down-link/500 Kbits/s Up-link/ 250 Kbits/s 3-27 MBits/s	Down/Up-link 1or 4 MBits/s
Coverage	1000 meters	15–20 meters	30 meters (max)
Modulation	OFDM	RSU: 2-ASK OBU: 2-PSK	2-ASK, 4-PSK

Table1.1: DSRC- Standards in developed countries [5].

The purpose of WAVE is based on the activities of IEEE P1609. It's their sub versions are exhibited in table 1.2.

Table 1.2 IEEE P sub versions and their Activities [8],[5].

Protocol standard IEEE P	Purpose
P1609.1	Management activities
P1609.2	Security related activities
P1609.3	Network layer activities
P1609.4	Physical channel parameters

Some recently completed VANET related projects are shown in the Table 1.3. Vehicular networks have been used in lots of traffic related applications. They are classified as applications relating to safety and non-safety. Safety applications include applications like cooperative driving, simple message exchanges and accident avoidance messages. Whereas, the non-safety application includes applications contain road traffic information, internet access, toll related amenity, multimedia games and video sharing.

Table 1.3: Various VANET projects recently funded in EU, USA and JAPAN [9].

Country	Year	Project
USA	2002-2004	"Vehicle Safety Communications Consortium "
	2006-2009	"Vehicle Safety Communications Two Consortium"
	1998-2004	Intelligent Vehicle Initiative
	2004-2009	Vehicle Infrastructure Integration
	2014	Implementation of a V2I Highway Safety System and Connected Vehicle Test-bed
	2018-2020	NeTS: Small: VC-VANET: A Sustainable Vehicle-Crowd Based Vehicular Ad Hoc Network Supporting Mobile Cloudlet Computing
European Union	2001	Car- to- Car Communication Consortium
	2004-2008	PReVENT
	1996-2003	CHAUFFEUR I AND II
	2008	Carlink
	2000-2013	FleetNet
	2006-2010	Cooperative Vehicles and Infrastructure System
	2000-2013	CarTalk 2000

	2015	Building an intelligent transport information system platform for smart cities
Japan	2000	Demo
	2006	JARI
	2001-2005	Advanced Safety Vehicle Programe-3
	2005-2007	Advanced Safety Vehicle Programe-4
	2011-2015	Advanced Safety Vehicle Programe

Vehicles intermittently broadcast small packets called beacon message that is used to identify the current position, altitude and longitude of the updated location of the vehicles as proposed by [10]. The beacon message carries the various critical, constant ever-changing constraints such as power supply information, relative address, physical location, time-stamp, signal capability, bandwidth available, current temperature, and pressure as stated by [11]. This beacon signal awakens nodes from sleep mode because they were in idle state for a long time and synchronizes them to the sending nodes. Without any fixed infrastructure, the wireless ad-hoc networks is designed instantly; here the “Hello” signals function as beacons to communicate to nearby mobile nodes about active neighbors. The famous routing protocols such as, on demand distance vector routing, table-based routing, and associativity - based routing—use “beacon” signals for routing table formation. However, VANETs are dynamic and frequently suffers from the network disconnection problem: it renders harsh environments for the wireless communications.

In [12], the major characteristics of route maintenance and updating of routing tables for mobile nodes based ad-hoc networks are described elaborately. Even, these routing related protocols used in the traditional wireless networks and mobile ad-hoc networks such as “Ad-hoc On-demand Distance Vector” (AODV) protocol and “Dynamic Source Routing protocol” (DSR) are not suitable for the VANETs. To address the unambiguous routing problems of VANET, a meticulous design and implementation methods are required to cover the purpose of wireless networks.

Since, sensors are the integral part of the vehicular network, the drawbacks of the sensors are identified, in vehicular network; and it is inferred that the sensors are restricted to less than hundred meters. In urban surroundings, the traffic conditions often changes dynamically. It is also observed that the VANET specific protocol is also affected by the concern of ambiguous road environment, the divergence in the dimensions of the intersections in a distinct area, uneven slopes, trees, traffic lights, and signboards. Hence, it is impractical to redesign the existing roads in urban areas. So, VANET exclusively needs a routing protocol for covering the large distance areas, with data exchange patterns that supports both one-to-many and one-to-one communication types.

Testing and deploying of VANETs comprises high cost and work force. Hence, simulation is used before real implementation. VANET Simulations contain large and diverse scenarios which takes the distinct characteristics of vehicular surroundings into account. In MANETs the random waypoint model (RWP) is applied for generating mobility model as suggested by [13]. In vehicular networks, nodes moves along streets only, which prompt the need for a road model implementation for better visuals. Another significant feature of VANET is that nodes don't move separately, their movement is based on well-established vehicular road traffic replicas. Henceforth, the outcomes for MANETs design models may not be implemented straightaway. Likewise, speed also various here (MANETs- 0 to 5 m/s, in VANETs 0 to 40 m/s). According to [14] the categories of the simulation software's are:

- Vehicular mobility generators
- VANET simulators
- Network simulators

Vehicular mobility generators are used to escalate realism. The, Network simulators gives the packet-level simulation of node data traffic and various routes of the road. The third category i.e. VANETs simulators used to develop IVC models, which gives drivers response, based on the signals.

1.2 Objectives

This research project aims to develop a new broadcasting protocol for VANET to solve the issues of broadcasting storm problem, intersection identification and route selection. The integration of all these modules gives more efficient way of data dissemination.

Hence, the research objectives are as follows:

a) To enhance data dissemination protocol for making better decisions in finding the available Paths without the need of recognizing the intersections
b) Investigate the broadcast storm problem.
c) To develop a model that disseminates traffic data to a vehicle moving from the opposite direction.
d) Evaluate and validate the performance of the protocol with the standard performance metrics using simulation.

1.3 Significance of the Study

The novel aspect of the proposed protocol of VANETs in the perspective of design of data dissemination is its ability to adapt to various road scenarios in urban, rural and spares areas. Every node looking for data transforming autonomously decides whether to send a received broadcast message that is associated to scalability property of local data obtained via the beacon messages. This leads to the message overhead reduction that has not been addressed previously. DV-CAST protocol proposed by [15] is the only protocol that addresses the various connectivity condition problems in VANETs. However, the drawback of the proposed approach [15] is that it is restricted to the traffic structure of highway. During message forwarding, the node in "opposite direction" is not well-defined as neighbor nodes for urban circumstances with various roads join at an intersection [15].

Secondly, the broadcast storm problem is handled in a better way by using the broadcast conquest and de-synchronization approach to decrease the problem of end-to-end delay that arises from flooding. Lastly, the traffic data dissemination may be possible in any direction.

Dissemination in the opposite direction improves performance in various vehicular scenarios including urban traffic, highway traffic, dense and sparse areas. From the literature survey, these three challenges have not been addressed simultaneously in any of the existing protocols. Hence, the gap will be duly addressed and the performance of protocols is evaluated based on high reliability, less end-to-end delay, and lowest amount of retransmission in the VANETs simulation software as metrics.

1.4 Chapter Summary

In the above section, the technical aspects of VANET are discussed along with the, design considerations of the protocols used in the establishment of VANET. The real time implementation for this kind of networks involves huge amount of money and labor, so the latest simulation is considered as the platform for testing and implementation of new protocol. The next chapter describes about the reviews of earlier works and their loopholes, which leads to identify the research gaps for the future work.

CHAPTER 2-Literature Review

This section describes about the literature review of the earlier work. First, it describes the major routing protocols and their drawbacks; next it discusses the various broadcasting algorithms with the challenges and, drawbacks. Followed by, dominating sets solving method, broadcasting storm issues, intersection problem description, route selection and optimization method and the gaps in their approach.

2.1 Routing Aspects

In recent years, with improvements in the field of computing and communication, more importance has been given to VANET research in order to improve the convenience of the user. The high mobility, numerous changes in topology and inadequate lifetime are main features of VANET that makes routing decisions more challenging. Some factors like road layout and different environments such as city and highway make the comparison between different routing techniques difficult. A routing protocol regulates the exchange of message information between two communication objects. It consists of route establishment, then forwarding of data based on the current information and route maintenance based on any link failures. Nodes in VANETs join, leave and rejoin the network that dynamically leading to a frequent data interruption hence deciding a standard routing solution for all VANETs scenarios is not possible.

Inter vehicle communication remains a challenge, despite of the various routing protocols that have been developed for VANETs. Topology-based and geographically based protocols classification is as shown in Figure 2.1 as described by [16]. In [12], VANETs specific protocols based on their types of routing, how they use location information, and how they assessed (i.e. Simulators and various simulation scenarios) are categorized. To deliver user safety related information to the neighbors, VANET uses unicast, multicast/geocast and broadcast message passing protocols. Unicast routing is operated on a source to destination basis. Multicast method is defined as distribution of multicast packets from an individual source to all possible members in the communication range. Then, Geocast approach is for delivering packets to a geographic region. Broadcast protocols send source packets to all vehicles in the network.

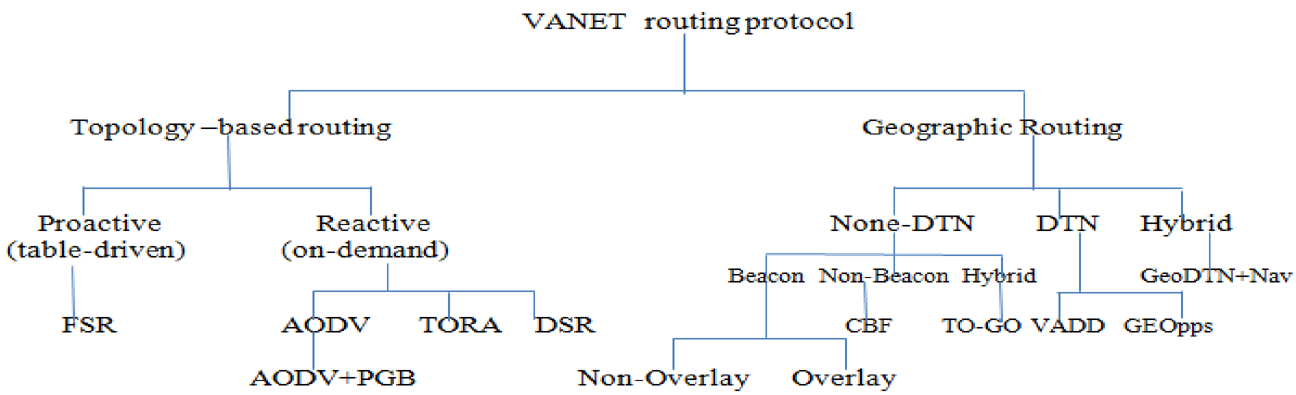


Figure 2.1: Classification of routing protocols in VANET [17]

Adhoc networks are modelled by the unit graph construction. In this graph construction, two nodes A and B are assumed as neighbor nodes and the Euclidean distance between the nodes is defined as R (transmission distance). The value of R is considered as identical values for all pair of nodes in the graph. The broadcasting algorithms of [18],[19] used the unit graph methods which results in reduced amount of rebroadcasting of nodes. Here, network condition is considered as global information, so nodes having the best value among the others will perform broadcasting. The disadvantage of this method is that it is not reliable. The method is said to be not reliable since only if there are no collisions during data transmission; the proposed broadcasting algorithm guarantees the data transmission to all the receiving nodes that are attached to the source node. Apart from that if, there are no collisions, it guarantees the message delivery to all intended neighboring nodes at presents.

In [20], the time division multiple accesses method is used, here every node assigned with a distinct time slot. According to [19], broadcasting redundancy is minimized by trimming down the packet loss due to assertion or collision and possibly intensify the dependability of data transmission. A multipoint electrical relay technique is used by [21] to improve the reliability of broadcasting.

2.2 Broadcasting Algorithms

The proposal of [22]describes a generic broadcasting protocol for adhoc type networks linked to the approach of address-searching. In [23] cluster-based association of links are established. Cluster head is nominated among the clusters; each has a direct connection to each of the cluster's nodes. Here, source node propagates the message to its near by cluster heads (CH), which built virtual spanning tree of all available CHs by transmitting the message to all. More specifically, the content is dissipated to all neighbouring CHs. The authors of [24] proposed a distributed clustering design, which is originated by all nodes whose ‘id’ is the lowermost in the middle of all its neighbours. This node broadcasts their decision to create clusters among the neighbours. Every node discovers the broadcast by its neighbours and chooses the last-place id among neighbouring cluster heads, if any exists. If all neighbours with a lower id have sent their verdicts and none have confirmed a CH, the node decides to make its own cluster head and send its id as a cluster. Furthermore, it selects neighboring CH with the lowest id and transmits each selection. Each node can determine its cluster and only one cluster can contain exactly one information during the entire process.

A combination of clustering and broadcasting algorithm are designed in [22], which does not have connection over-head intended for support cluster structure nor updating neighbourhood data. They use the cluster structure to update the current road traffic information by adding 2 bits to each on-going message. This approach has universal parameters and is not trustworthy. In addition, they have very poor delivery proportion. The reports only get 35 per cent of message in retransmissions for of flooding environments. The exceptional case of simulating broadcasting mission while every node are placed on a straight line is premeditated in [25], [26], they consider only the inter-vehicle communications scenarios like highway. The authors in [27] represents energy efficient broadcast tress of wireless networks having adjustable transmission radios for processing. In [28], a centralized algorithm for static network needs the geometric information such as distance or positions are processed and analysed. The survey shown in [29] indicates only worst case scenarios and ideally ignored invariant multipliers in the calculation of time complexity.

2.3 Dominating Sets

Let 'F' be a wireless network graph. A set is dominating if all F nodes are in either the set or neighbours of nodes in the set. A node may consider being an internal node of graph 'F', only if it belongs to a dominating set. Connected dominating set is used in [30] to make the routing base, the internal node selection is based on the searching space value on the route identification process. Then, the authors[30] introduce the intermediate node and inter-gateway node concept, which calculates connecting dominating sets in wireless networks. To cut the use of number of intermediate and inter-gateway nodes, they introduce two rules. Let Y (*s*) be an open set of all neighbours of a node s and let $Y[s] = Y(s) \cup \{s\}$ be the corresponding closed neighbour set.

Rule 1: Nodes *t* and *s* are considered here as two intermediate nodes. If $Y[t] \subseteq Y[s]$ in F and $id(t) < id(s)$, then node *t* is not an inter-gateway node for the given graph. It can be further elaborate as, if any neighbor of node t is also a neighbor of *s* and t is connected to *s* and has lower *id*, then any path via *t* can be set by a path via *s* and, thus, node *t* is 'covered' by node *s*.

Rule 2: By applying Rule1 we get, *s* and *w* are two inter-gateway neighbors of an inter-gateway node *t*. If $Y[t] \subseteq Y[t] \cup T(w)$ in F and $id(t) = \min\{id(t), id(s).id(w)\}$, then node *t* is declared a non-gateway node. In other words, if each neighbor of *t* is neighbor of *s* or *w*, where *s* and *w* are two connected neighbors of *t*, the *t* of gateway nodes can be destroyed. With the help of the location information, every node decides whether it is an inter-gateway, intermediate, or simply gateway node with time complexity $O(k^3)$ calculation time.

Two centralized algorithms are proposed by [31]. Algorithm 1 is based on a greedy algorithm concept, where connected dominating set grow from an apex with the highest degree. In successive iterations, the dominating set is selected as a single node or a brace of nodes, which are neighbors to the current dominators and have a significant yield. The yield of a node is calculated by adding neighbors of the node that are not dominated. According to the authors, Algorithm 1 produce a CDS of size at most $2(1 + H(\Delta)) \cdot |OPT_{DS}|$, (here H-harmonic function, Δ- maximum degree, and OPTDS - OPTimal Dominating Set.)

Algorithm 2 is implemented in multiple stages. At the start, all nodes are assigned in white color. In every iteration of the primary phase, a node is reduced to the maximum number of pieces is assigned to black color; subsequently white neighbors are assigned to gray node. A portion is categorized as a black node or a white associated component. In the subsequent phases, pieces are recursively joined by picking a sequence of two apices. At long last, all black hubs procedure a connected dominated sets (CDS) of the network. The authors of [32] design a distributed algorithm which has three stages. The rooted span tree is built in the algorithm's preliminary stage, and then a spanning tree is developed using maximal independent set (MIS) in the subsequent phase and a dominant MIS-based tree in the final phase. Perhaps with the help of internal nodes, the dominant tree nodes form a CDS.

[33]Demonstrate that its procedure has an estimate factor of 8 and has a low time complexity O(n) and message complexity as O(n log n), which is message ideal. In their research, however, the preservation of CDS in a wireless atmosphere is not investigated. A Timer-based protocol namely MTCDS is proposed by [34], this is a Type-I distributed calculation algorithm. In the beginning phase, the node with the smallest ID is designated as the initiator in a scattered way, and the initiator becomes the primary node in the ruling dominating set. In the next stage, nodes in the dominant set convey their position messages occasionally. When a status message is received, a neighboring node twists a timer that turns the node into a dominator when it ends. A node renovates its clock in order to replicate changes near it, and a superior number of exposed neighbors lead to a reduced timer. A node without revealed neighbors never finishes the clock. Nodes with more exposed neighbors will from now on become the dominators in every cycle. With the modified network topology author generates and maintained the CDS in their approach. Those changes incorporate pulling back of the initiator, extracting of a new dominator, and appending a fresh node into the network.

The authors of [35] describes A tree-based algorithms similar to MT-CDS , in which several initiators are used to acquire lesser CDS size. Their processes yields connect neighboring trees in the third phase. A cluster head based algorithm is proposed by [33], which employs node

IDs to choose required dominators in the clustering stage and joins them by pathways among detached dominators in the linking stage. Comparing the algorithms of [36] and [31], we found one of the dissimilarity occurs how the weakly connected dominating set is constructed.. With this distributed information from the algorithms, a unique arbitrator node selects new dominator. Creation of a rooted spanning tree is implemented in the next phase, where root node is fixed as an arbitrator. Next, Connected dominating sets are grows from the arbitrator node in the next phase. To select a minimal dominant set a priority method is used.

According to [37], a priority approach is employed to choose a minimal dominating set (MDS). In the next step, more gateway nodes from the MDS form CDS successfully. The factors like node ID, energy level, and mobility are considered as priority factors for set formation. While comparing, all the algorithms referred above uses only one factor in dominator selection, but authors used many factors to give preference. In [38], the authors have proposed two algorithms, Algorithm I is based on distributed method and algorithm II use cluster head construction method to form the CDS. Algorithm I: a spanning forest is built in the first phase. The generated fragments are connected by applying the distributed minimum spanning tree scheme during the next stage. In Algorithm II a fragment is selected randomly, subsequently the extensions are added which forms the dominating sets. Multiple-node association is treated as a separate single-node drive in all mobile environments and CDS is updated according to the processing order.

2.4 Broadcasting Storm Issues

In VANET, the broadcasting alert message contains an information like a road accident or traffic jam in nearby location, which are propagated to its nearby vehicles. Blindly broadcasting theses redundant packets may tip to collisions between neighboring vehicles. Research community refers this as a broadcast storm problem. According to [18, 39], a problem caused by overflow of a repeated information over a network resulting from, data contention and collision is known as broadcast storm problem.

The broadcast storm has a very high impact on the link layer, usage of bandwidth, system processing power, packet collisions, and more significantly the facility interruption. An opportunistic cluster choice based algorithm is proposed by [40], which selects only one cluster head, which competently rebroadcasts messages like emergency information, locations and regulatory information. The drawback is that, it allows vehicles to spread message selectively within its transmission range, which causes network overload, and message duplication. To improve the broadcast storm problem [39] suggests, the re-broadcasting messages from the node furthermost away from the broadcaster, despite the consequences of the node density. They used these three approaches to resolve the broadcast storm problem and they are depicted in the Figure 2.2., namely

- weighted p-Persistence broadcasting technique
- slotted 1-Persistence
- slotted p-Persistence broadcasting

According to weighted p-Persistence broadcasting, re-broadcasting probability is calculated as $p_{ij}=D_{ij}/R$ (pij - forwarding probability, D_{ij} - relative distance and R- Transmission range) on a per-packet based. The weighted p-Persistence regulates superior possibility mobile nodes that are situated remotely from the broadcaster, only when the packet header is accessible and available using the GPS information. According to second technique, the probability is assigned as one at the time slot TS_{ij}, that is considered as $TS_{ij}= SN_{ij} *\tau$ (where, τ - one-hop delay, SN_{ij} -slot number). If duplicate packets are received from multiple senders, means, it proceeds with the smallest D_{ij} node value for processing. Further this approach also requires R (average transmission) value in certain range of the fixed slot size.

In slotted p-Persistence broadcasting: each node buffers the message for a certain period. Retransmit with probability one, only when nodes in neighborhood does not prevent the rebroadcasting of "die out" message. Calibration of this arrangement also based on the reforwarding probability (p) value preferred. However, result shows that this approach achieves 70 percentage packet loss ratio and data redundancy only. The simulation scenario was done for one-dimensional highway network wherein, the packet loss ratio was found at high-loss ratio to a magnitude of 90% in the worst case of slotted p-Persistence approach.

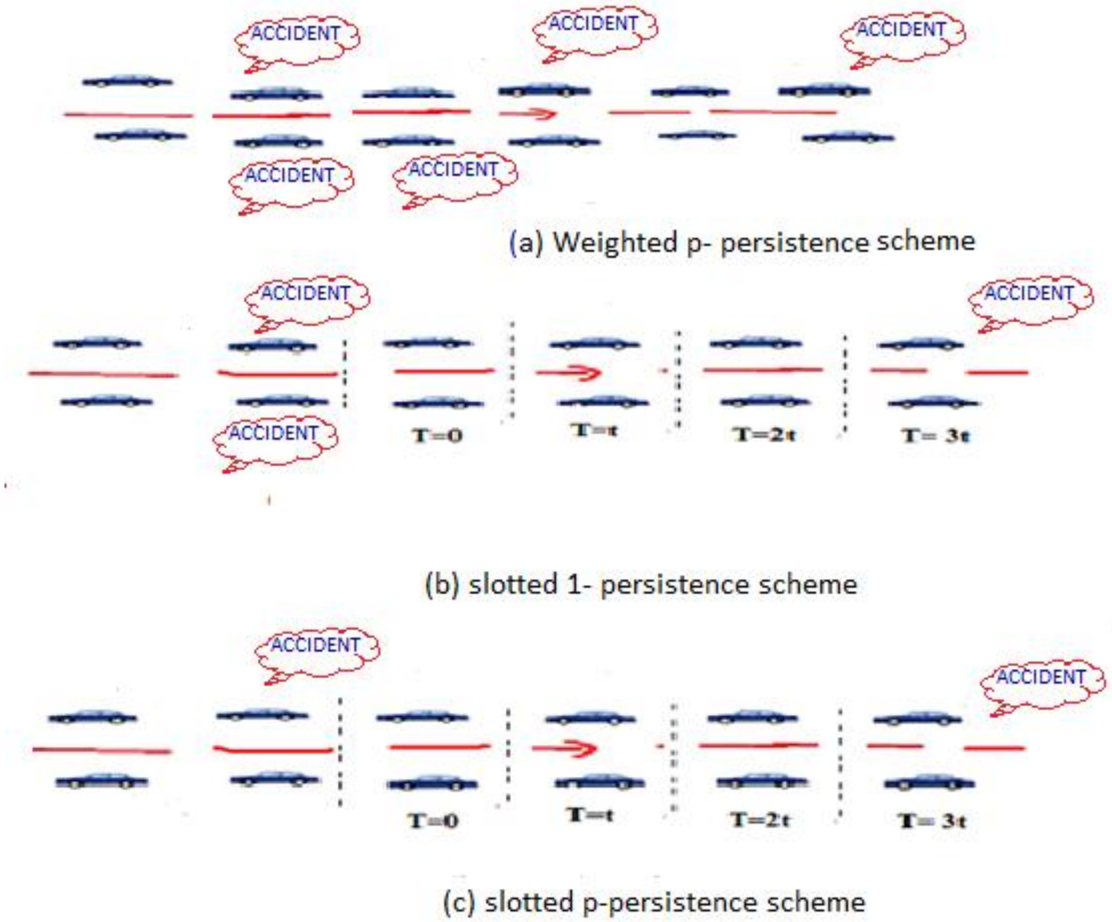


Figure 2.2: various slotted persistence schemes [41]

In [18] based on the physical parameters like distance and location are proposed to cut the redundancy and separate timing of rebroadcasts is used. The advantage of this method is found to be, higher percentage in packet delivery rates and lesser number of retransmissions. Nevertheless, they are found to be not reliable with regards to scenarios. In the probabilistic method, with a given probability *p*, each node rebroadcast its first copy. As indicated by counter-based plan, rebroadcasts of messages just conceivable when it have lesser than C neighbor's esteem. In location-based scheme, the retransmission is occurs only when it covers more area than the threshold A covers. The cluster algorithm of [24] is used to get lowest cluster ID for cluster based schemes. Then, any of the above three methods is applied on its border nodes and cluster heads. Table 2.1 exhibits some of the pros and cons of the existing works in tabular format.

Previous work	Significance	Limitation
Tonguz & Wisitpongphan (2007)	The rebroadcasting messages from the vehicle furthest away from the broadcaster, regardless of the node density	Reduction in packet loss ratio achieved 70 % for Highways.
Bae et al. (2012)	Rebroadcasting is performed by the vehicle furthest away from the broadcaster.	Only applicable with *slotted p-persistence* scheme.
Vegni et al, (2013)	*Opportunistically* selecting neighboring nodes, which acting as relay nodes.	Cluster aggregation occurs in a *random fashion*, due to the non-homogeneous nature of the urban areas.
Kumar &Dave (2013)	Probabilistic data aggregation.	Increasing probabilities or decreasing WT as the distance from the sending vehicle increase & for nodes outside R

Table 2.1 Pros and cons of existing works in broadcasting storm issue.

2.5 Intersection Problems

Another main issue of VANET is to design a routing protocol appropriate for road intersections. The greedy forward technique is used for transferring the data to the next available node. However, this technique causes unnecessary routing. Therefore, [42]proposed a position-based routing scheme called A-STAR. The drawback of this method is connectivity, because they utilize fixed vehicular information based on the existing paths to discover their destination for city buses. Likewise, in ASTAR, forwarding packet information involving consecutive junctions was done in simple greedy forwarding mechanism without considering the vehicle current direction and speed at which it travel. Global Positioning System (GPS) is employed in VANETs to find the next existing node on the path was proposed in [4].

Deploying the greed and perimeter nodes within the network to find the new routes between nodes are suggested in [43]. They include, 1) Support for MAC-layer failure feedback approach to deal with the time-out interval, 2) interface queue traversal, 3) Planarization of the graph. The drawback of this wireless protocol is that it has not incrementally updated the planarization upon receipt of beacon information from a neighbor, to keep the planarized graph maximally up-to-date. In [44], the problem of cross-links is solved using stateless routing methods. A distributed graph planarization technique is used in Cross-Link Detection Protocol. By exercise the proactive search technique and elimination of the cross-links increases message overhead complication. A planar graph approach is created as a base for fundamental topology that eliminates the cross-links with a mutual witness algorithm applied by [45].

As explained by [17], GeoCross protocol eradicate cross-links to evade the looping problem. With updated natural planar features of Urban maps are used in GeoCross protocol. In terms of performance comparisons with Greedy Perimeter Coordinator Routing, GeoCross's packet delivery ratio is higher and average delay is lesser. Nevertheless, the previously discussed protocol does not resolve the dilemma of intersections message forwarding. According to GPSR protocol, packet forwarding is performed when nodes are geographically closer to its immediate neighbor. After local optimum is attained, a recovery approach is applied for packet forwarding. This process continued until nodes met local maximum value. Forwarding of data in GPCR protocol is performed using the existing junctions and streets with greedy technique.

An intersection based protocol is proposed in GyTAR. An enhanced greedy method applied is for dynamic intersection selection and data forwarding. With help of the digital map, forwarding node is materialize for the adjacent junctions. Based on curve metric distance and traffic density at the junctions, unit values are assigned in the table. The junction among maximum unit is selected for packet forwarding. Once junction is chosen, in the subsequent phase the packets are forwarded. A brief summary in tabular format exhibits the significance and its limitation in table 2.2.

Previous work	Significance	Limitation
Karp &Kung (2000)	Greedy forwarding mode, Perimeter mode.	Lacks information about the network topology, it can potentially go through loops
Seet et al. (2004)	Anchor path computed using Dijkstra's least-weight path algorithm	It selects paths with *higher connectivity* for packet delivery leads to a higher average end-to-end delay.
Korkmaz et al. (2006)	HUNTER vehicle which tries to select the closest vehicle to the inter section	Segments require a *repeater* because of existing buildings in the urban areas.
Lee et al. (2010)	Removes cross- links dynamically to avoid routing loops in urban VANETs	Produces *higher hop count* than to environme ntal cross-links.
K. Chandram ohan & P. Kamalakk annan(2015)	NDA-CDS uses Aloha-based Collision Correction	Broadcast rate efficiency on data packets being traversed in VANET is 27.04% only

Table 2.3- Summary of pros and cons in tabular format- inter section issues.

2.6 Route Selection and Optimization

The routing decisions are influenced by geographical information as describe by [22] in A-STAR protocol. Along with that packet header comprises of list of intersections and number of next hop counts for traversing. Street scenarios are assorted based on updated traffic density information. CAR- Connectivity-Aware Routing algorithm [23] is implemented in three phases:

(1) Identification of destination node and feasible path to attain.

(2) Message transmission over identified path.

(3) Route maintenance.

Contention-Based Forwarding (CBF) [27, 41] is location based unicast forwarding routing algorithm, implemented without neighborhood information. The decision on the forwarding is built on the actual location of the vehicles in the current GPS position. Directional Greedy Routing (DGR) uses both position first forwarding and direction first forwarding techniques .Position first forwarding tries to discover the adjacent node which leads to the target node as the next hop. Along with those, the individual node nearest to the target will be preferred as next hop. [33, 29]. “Landmark Overlays for Urban Vehicular Routing Environments” LOUVRE routing protocol uses overlay nodes. Here, urban intersections are assumed as overlay nodes. Precondition for construction an overlay link is, only when path promises the multi-hop routing among the two overlay vehicles based on traffic density. In [31] overlay network link is assessed. Formerly, routing is executed effectively on the overlay node network, assured the packet release.

“Advanced Greedy Forwarding (AGF)” was proposed by [32], they combine the speed, node movements, and the total travel time into a beacon packet. It is more improved in PDR. “Predictive Direction Greedy Routing (PDGR)” predict the upcoming neighbors easily. This protocol estimates the weighted score for existing, in progress and future possible neighbors. It exercises two-hop neighbors’ method to produce possible future neighbors. Next hop is decided in [33], by approving all these weighted scores. In [35] an abstract neighbor table is used to find x hop neighborhood. The neighbors with smallest matrix value will be selected for the next hop iteration. It split the entire plot into tiny areas and has only one agent neighbor per area. GpsrJ+ protocol reduces hop count number by purge the redundant stops at different connection of the map. The selection of next segment depends on two-hop neighbor beacon intersection. The subsequent forwarding neighbor node selects a different direction. In the next iteration the packet is forwarded to the nearest junction neighboring node [36]. If it approaches local maxima after that it employs the perimeter mode.

A junction node in the ASTAR algorithm has updated information on the existing road connectivity. From the available nodes in the junction, one is randomly selected as master node. This master node generates the link to the next junction for forwarding. This link information also broadcasted to auxiliary master nodes [37]. It computes the geographic gap from current vehicle location to the target nodes and forwards the packet information to immediate node that has fewer distances. Some of the significant and limitations of the previous works are tabulate in table 2.4.

Previous work	Significance	Limitation
Chen et al. (2001)	Relayed and stored temporarily at moving nodes while waiting for opportunities to be forwarded further	Suitable only for *localized* applications with *delay tolerant*
Niculescu & Nath (2003)	Requires that, nodes know their position relative to a coordinate system	Since the *trajectory* does not explicitly encode *intermediate* members of the path.
Schwartz et al. (2011)	In dense networks- o*ptimized broadcast suppression technique* used and in sparse networks, the *store-carry-forward*	Not suitable for *urban* and *dense* network
Hai- tao et al (2016)	Straight road and direction ,position speed of the neighbor nodes	Highway and three lane area only

Table 2.4- Brief summary of the existing works.

2.7 Chapter Summary

This chapter described about the earlier work done in field of VANET in terms of safety related classification aspects. Firstly, the various routing protocols, their classifications, advantages and drawbacks. Next, in specifically about broadcasting protocols and selection of dominating sets are studied. Then one of the internal design aspects broadcasting storm issue and the impact it creates in delay of the message forwarding and the various ways to resolve this issue is elaborated. Next, the intersection section issue in the urban area been addressed with the previous works and their drawbacks are identified as research gap. The next chapter will describe in detail about the proposed protocol design and implementation.

CHAPTER 3- Protocol Design

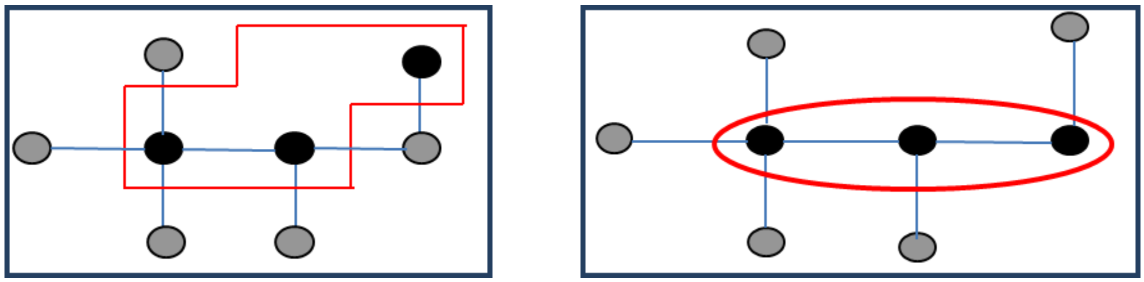
This section elaborates the details of the proposed protocol design and the methods used. The steps involved in the protocol design are constructing connected dominating sets and the broadcasting protocol. Both intersection problem and broadcasting storm issues are addressed in the broadcasting protocol. The proposed protocol is implemented in the simulation software environment. It is evaluated and compared with the existing protocols.

3.1 Design Overview

The proposed protocol is an adaptive and distributed algorithm suitable for all mobility situations of VANETs. It adjusts its behavior automatically without tracking the degree of mobility that the vehicles sense. Every node decides autonomously whether or not to transmit the received message. These choices are based on the local information obtained by periodic beacons from its neighbors. The design of the proposed protocol consists of construction of connected dominated sets with neighbor elimination approach, decision-making to deal with the intersection in the traffic, addressing the broadcasting storm issues, and design of direction selection for packet forwarding.

3.1.1 Design of Connected Dominating Sets

A node participates in message broadcast, does not retransmit it instantly. The nodes wait to confirm if retransmissions from additional nodes have already covered the neighborhood regions and then start its transmission. This information is enough to compute the Connected Dominating Set (CDS) described by [29]. A dominating set (DS) is a subset of all nodes so that each node is in the DS or adjacent to a number of nodes in the DS. A connected dominating set (CDS) is a subset of the nodes such that it constructs a DS, and all the nodes in the DS are connected (shown in Figure 3.1).



(A) (B)

Figure 3.1: (A) Dominating Sets (B) Connected Dominating Sets (CDS).

In detail, F = (*V*, *E*) is a graph considered as a CDS and it is assumed as a partition of nodes K $\subseteq$ *V* such that K is a DS of F*'s* and the sub-graph of F. The CDS is required to get smallest possible connected dominating set among F. It is also comparable to discover a spanning tree with highest amount of leaves. In recent research, CDS is defined as a vital problem in wireless network and investigated accurately in optimization and computer engineering. Recently, the trivial Ω (2*n*) enumerative algorithm is the best exact algorithm to form CDS with all possible subsets of nodes. In the proposed trivial Ω (2*n*) enumerative algorithm, the formed set of CDS is the general (unconnected) version [46], [47], [48]. The best case algorithm for DS has running time complexity of O(1.5137*n*) [46].

Branch and Reduce paradigm is the major technique that supports CDS for constructing as a fast exponential time based algorithm. The reduction rules are applied first and next branch on two or more sub problems, which are recursively solved. Worst-case running time and average running time analysis are based on reduction and branching rules. This leads to linear recurrences based on upper bound values. In case of non-standard measures, Measure and Conquer method proposed by [46], gives the enhanced study of Branch and Reduce algorithms.

Let *F* = (*V*, *E*) be an un-directed graph. The open *neighborhood* of a node *t* is designated by $N(t) = \{p \in V: pt \in E\}$, in addition to that the closed neighborhood of *t* is indicate by $N[t] = N(t) \cup \{t\}$. The sub-graph of *F* induced by a set K, $K \subseteq V$ is designated by *F* [*K*]. A set $S \subseteq V$ of nodes of *G* is *connected*, if *G*[*S*] is connected.

By Assuming: (I) the graph is connected one.

(II) The minimum connected dominating set takes cardinality value at least two.

According to the final hypothesis, we can deliberate the *total* variant of CDS, where each node *v* *dominates* its neighbors *N* (*v*), but not the node *t* itself.

Assign two subsets node M and N, with the constraints as $|M| \geq 2$ and *F* [*K*], is connected where (*M-selected* nodes, *N-discarded* nodes). The recursive algorithms that discover an optimum solution *OPT*, if any, under the constraint that every node in *K* and no node in *N* belong to *OPT*: $K \subseteq OPT$ and $N \cap OPT = \emptyset$. In order to resolve CDS, it is adequate to guess two adjacent nodes *t'* and *t''* of several optimum solution and run the algorithm above on the instance $(M, N) = (\{t', t''\}, \emptyset)$. Obviously, the illustration is infeasible after $V \setminus N$ is not a connected dominating set.

For further elaborations the algorithm defines some notations. The *available* nodes $L = V \setminus (M \cup N)$ are the nodes which are neither selected nor discarded. An available node *c* is a *candidate* if it is neighboring to *S* and a *promise* if its elimination formulates the example infeasible, i.e. $V \setminus (N \cup \{c\})$ is not a connected dominating set of *F*. Instinctively, a candidate is a node that might be upended to *K* in the existing step, while a promise is a node that must be added to *K* at some point. We articulate that a node is *dominate* if it is adjacent to a node in K, otherwise it is *free*. By *R*, we designate the set of the free nodes as

$$R = V \setminus \cup_{c \in S} N(c).$$

The algorithm stops the progress if either the example is infeasible, or *K* is a (connected) dominating set. In the primary case the algorithm returns the value as *no*, while in the subsequent one it returns the value as $OPT = K$. Else the process carry out approximately reduction on the problem example, and then it branches on one or more subproblems, which are recursively solved. In case of each subproblem, the algorithm enhances accessible nodes to either K or *N*, but always-keeping *K* connected. The excellent explanation of the subproblems, which reduces the size $|OPT|$ of the result, is the answer to the original question.

The reduction rules decide the connected dominated set formation:

(a) If there is a contender *c* which is a promise, select it (add it to *K*);

(b) If there are two contenders *c* and *d* (which by (a) are not promises) such that $N(c) \cap R \subseteq N(d) \cap R$, discard *c* (add it to N);

(c) If there is an available node *c, which* does not dominate any free node, discard *c*.

The branching section assessed using the following regulations:

(A) If there is a contender *c* which dominates at least three free nodes $d1$, $d2$, and $d3$, or this overlooks an available node w such that, after selecting c, d does not dominate some free node, two subproblems of the branch are:

- $(M1, N1) = (M \cup \{c\}, N)$
- $(M2, N2) = (M, N \cup \{c\})$

(B) If there is a candidate *v* which dominates a unique free node *d*, let $Q = \{q1, q2. \ . \ . \ qk\} = N(d) \cap L \setminus N[c]$ be the set of the available neighbors of *d*, *which* are not in the closed neighborhood of *c*.

Three subproblems of the branch are:

- $(M1, N1) = (M, N \cup \{c\})$
- $(M2, N2) = (M \cup \{c, d\}, N)$
- $(M3, N3) = (M \cup \{v\} N \cup \{w\} \cup Q)$

Here w is discarded. Likewise, one of the q_i's could be a promise. In such situations, one or more subproblems are classified into infeasible state, and the algorithm simply halts. The following cases are example of this type of situations.

(C) If there is a candidate *c*, which dominates two free nodes $d1$ and $d2$, name $d1$ and $d2$ such that if $d2$ is available (a promise), so is $d1$.

Let $Q_i = \{q_{i,1}, q_{i,2} . \ . \ . \ q_{i,ki}\} = N(c_i) \cap L \setminus N[c]$, be the available neighbors of d_i, *which* are not in the closed neighborhood of *c*.

The following sub-cases are considered here:

(C.1) if $d1$ and $d2$ are adjacent, $d1$ is available and d2 is discarded, three subproblems of the branch are:

- $(M1, N1) = (M, N \cup \{c\})$
- $(M2, N2) = (M \cup \{c, d1\}, N)$
- $(M3, N3) = (M \cup \{c\}, N \cup \{d1\} \cup Q1)$

(C.2) if $d1$ and $d2$ are adjacent and both available, four subproblems of the branch are:

- $(M1, N1) = (M, N \cup \{c\})$
- $(M2, N2) = (M \cup \{c,d1\},N)$
- $(M3 , N3) = (M \cup \{c,d2\},N \cup \{d1\})$
- $(M4, N4) = (M \cup \{c\}, N \cup \{d1,d2\} \cup Q1 \cup Q2)$

(C.3) Otherwise (either $d1$ and $d2$ are not adjacent, or they are adjacent, and both discarded), Five subproblems of the branch are:

- $(M1, N1) = (M, N \cup \{c\})$
- $(M2, N2) = (M \cup \{c, d1\}, N)$
- $(M3, N3) = (M \cup \{c, d2\}, N \cup \{d1\})$
- $(M4, N4) = (M \cup \{c\}, N \cup \{d1, d2\} \cup Q1)$
- $(M5, N5) = (M \cup \{c\}, N \cup \{d1, d2\} \cup Q2)$

3.1.2 Correctness & Proof of the Algorithm:

Computation of a minimum cardinality CDS.

A reduction rule arrives to a feasible state, *if* it does not adjust the optimum value. Reduction rule have the following case:

Case (a): is feasible since eliminating a candidate v that is a promise would lead to an infeasible case.

Case (b) is feasible since if $c \in OPT$, then $OPT\_ = OPT \cup \{d\} \setminus \{c\}$ is a feasible result of cardinality at most $|OPT|$.

Case (c): is feasible since all the existing neighbors of c are already connected to M, and thus removing c from any feasible reply maintain the solution feasible.

The branching rules take the following feasible values. First, it examines how Mi induces an attached subgraph of the original graph. A branching rule is *feasible* if at least one subproblem preserves the optimum value. Branching rule (A) is trivially feasible: every connected dominating set either contains candidate c or does not.

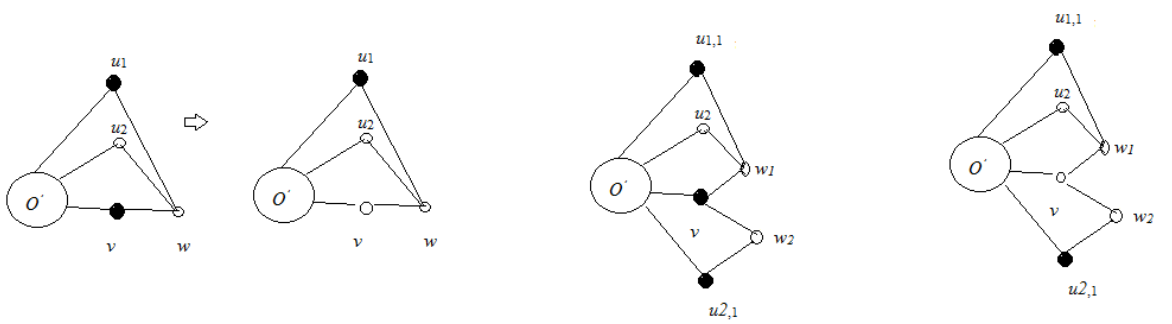


Figure 3.2: Block nodes selection for cases (B) and (C.3) branching.

Branching rule (B): It is adequate to demonstrate that if we select c and discard d, then we must also discard Q. The result of $(M', N') = (M \cup \{c\}, N \cup \{d\})$ is an optimum solution (OPT). In particular, $c \in OPT$ and d $/\in OPT$. Assume by contradiction that some $q_i \in Q$ also belongs to OPT, that is $OPT = Z' \cup \{c, q_i\}$ for a proper choice of Z'. Since $d /\in OPT$, $OPT' = Z' \cup \{q_i\}$ is also linked. Furthermore, since c dominates only d, and d is dominated by q_i as well, OPT' is a dominating set shown in Figure 3.2. Thus OPT' is a connected dominating set of size $|OPT| - 1$, which is a contradiction.

The feasibility of (C.3) follows by monitoring that if we choose c and discard both $d1$ and $d2$, then we must also reject either $Q1$ or $Q2$ (or both). This can be demonstrated with the same set of case (B) arguments. The rest of the two cases are more difficult to address. In first case (C.2): It is adequate to illustrate that, if we select c and discard both the d_i's, then we can also discard $Q1$ and $Q2$. By the similar argument used in case (C.3), we already know that in the optimum solution OPT to $(M \cup \{c\}, N \cup \{d1, d2\})$ we must discard either $Q1$ or $Q2$. For sake of contradiction, presume that $OPT = Z\_ \cup \{c, q1, i\}$ hold one $q1, i \in Q1$ and no node in $Q2$ (a symmetric analysis holds if OPT contains one $q2, j \in Q2$ and no node in $Q1$). Since $d1$ and $d2$ are neighboring, and $d1$ is available, by replacing c with $d1$ in OPT we attain another realistic answer of the same cardinality. Thus, we do not need to judge this case as if OPT is the optimum answer to the original problem. Then, the algorithm will find an answer with same cardinality value for $(M1, N1) = (M, N \cup \{v\})$ subproblem.

The similar arguments of case (C.1) show that, if we choose c and we reject both $d1$ and $d2$, then we can also reject $Q1$. Therefore, the feasibility of (C.1) is achieved here. Note that, differently from case (*C*.2), we cannot employ a symmetric argument to explain that also $Q2$ can be discarded. Since $d2 \in N$, also $(M1, N1) = (M, N \cup \{c\})$ optimum solution to cannot contain $d2$.

3.2 New Broadcasting Algorithm

The proposed protocol is an adaptive and distributed broadcast protocol, that appropriate for an extensive mobility environment. The major difficulty of any broadcast protocol is its flexibility to the diverse vehicular scenarios in real time environments. The protocols must have higher coverage area at the expense of as less transmission as conceivable, despite of whether the network is vastly disconnected or enormously dense. This protocol is based on enhancing Connected Dominated Set (CDS) and Neighbor Elimination (NES) models using the currently available neighborhood information. In addition, our design estimates the network connectivity using the ideal communication devices. Since actual communication links are far from perfect, the protocol utilize the broadcast acknowledgments to ensure the gathering of the information or retransmit it when failure. A received message is acknowledged in its entire lifetime. At expiration, node’s buffer removes this acknowledgment and no further acknowledgments are distributed. Given that broadcast messages are uniquely identified, since they are acknowledged. GPS receivers are equipped in Vehicles; local topology is updated by Periodic beacon messages.

The sender node physical positions are integrated inside the beacons, which are sufficient to compute backbone of CDS after each round. The source node broadcast the message. After, receiving the message for the first instance, each node initializes two lists namely X and Y.

List X - comprise of all nodes that received the message.

List Y- those neighbors which are in need of the message.

If a node is not present in the CDS table, it chooses longer time-out than the CDS nodes, consequently the latter first reacts. Each node updates X, Y and its time-out for each additional copy received and its individual message sent. It transmits if Y is not empty at the end of the timeout period. The message is buffered in both ways till it expires. On behalf of each message received, Y and X are restructured based on the presence or lack of acknowledgment. Nodes that are no longer single-hop neighbors are detached from these lists.

Irrespective of earlier decisions, all nodes obtaining the broadcast message verify that Y is not empty. If so, they initiate a fresh time-out. Acceptance of received broadcast messages is also sent back to periodic beacons. Nodes those were included in X, since they thought they received the message, but did not truly obtain it, were then removed from X and inserted in Y. This process will be repeated for each message. The beacon size therefore increases linearly with the number of simultaneous dissemination iterations.

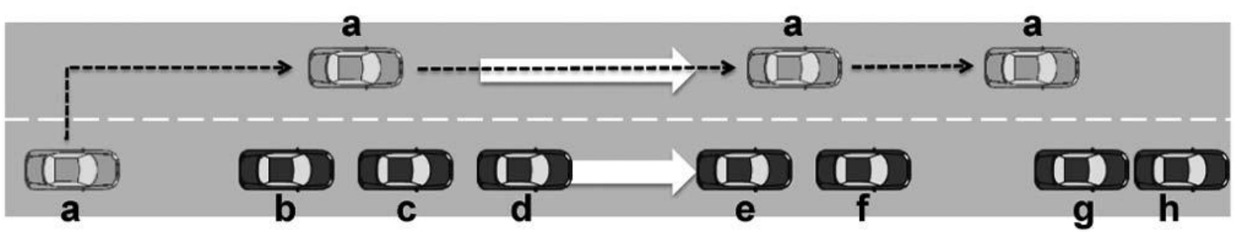


Figure 3.3: Vehicular Scenario.

According to Figure 3.3, Vehicle ‘*a’,* transmits a message that is initially buffered by ‘*a*’ and then received by b, c, d. Receivers set up a shorter waiting time out if the vehicle is part of the computed CDS. Let node d is in the CDS, so it primarily transmits. Vehicles b and c abandon transmission because most of their neighbors were covered by d's transmission. Both e and f vehicles obtain the message. None of them, however, has uncovered neighbors, so there is no retransmission. Vehicle ‘a’ speeds up and surpasses vehicle b_f. In the instance of PBSM protocol, fresh transmissions would occur because a (and vice versa) would have to cover new neighbors e and f. They are redundant, however, because all the vehicles have received the message. ABSM protocol saves these redundant broadcasts because the beacons contain the message acknowledgement and the newly discovered neighbors are no longer covered.

Algorithm-1: Protocol Design

```
Initialize
T← neighbor set; rc← communication range;
X ←∅; Y ← ∅;
Event: copy received or generated from h
Insertion of message id in consequent beacons;
X← ∪{h}; Y← \∪{h};
for each m ϵ B do
    if dist (m, h) ≤ rc then
        X← ∪{m}; Y← Y \ {m};
        schedule tack for m;
    else
        if m ∉ X then
    Y← Y ∪ {m};
if h= source then
    802.11 protocol forwards message ;
else
    if Y = ∅ then
        cancel to_evl;
    else
        schedule to_evl;
Event beacon received from neighbor h
Add h to neighbor set and Calculate CDS;
if beacon encloses ack then
    cancel tack for h;
    X← X ∪{ h }; Y← Y \ { h};
else
    if h ∉ X then
        if h ∉ Y then
            schedule to_evl;
        Y← Y ∪ {h};
Event tack deceases
if Y ≠∅ then
```

33 X← X ∪ Y;
34 **for each** *h* ∉ Y **do**
35 schedule *tack* for *h*;
36 Y← ∅;
37 forward message using 802.11;
38 **Event** *tack deceases for neighbor h and ack from h never received*
39 X← X\ {*h*};
40 **if** h ∉ Y **then**
41 schedule *to_evl*;
42 Y← Y ∪ {*h*};
43 **Event** beacon from h not received for last beacon hold time
44 **if** Y = {*h*} **then**
45 cancel *to_evl*;
46 Y ← Y\ {h};
47 Eliminate *h* from neighbor set; calculate CDS set

3.3 Protocol Description

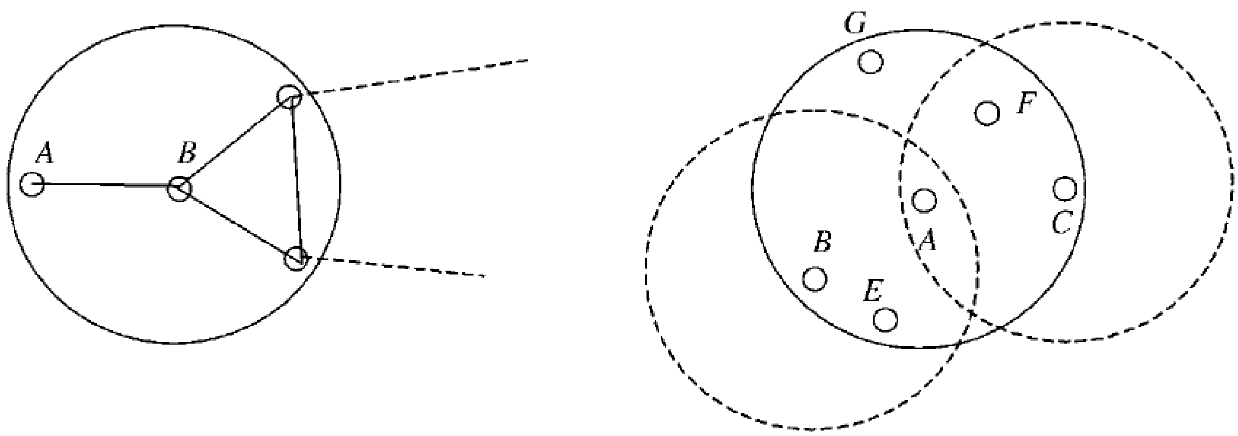
When the broadcast message is received, vehicle x includes the source and all its known neighbors in X (and starts to acknowledge timers) because they have probably received the message (lines 4 - 10). These vehicles are therefore removed from Y (lines 6 - 9). In Y nodes are inserted the remaining x neighbors that are not coupled to the sender (their distance is superior to that of r). There is a time-out to_evl function, which assigns each vehicle a waiting time before it can be transmitted. to_evl is relative to 1=jYj, where jYj is the number of elements in Y, and be subject to on whether the CDS presently contains the node. The rationale is to offer vehicles with priority for retransmitting messages to more neighbors. If some neighbors take the similar status and the amount of neighbors that need the message, the same to_evl value will be obtained. This does not mean, however, an increased number of collisions, as it runs on the network layer and these messages must still contend to access the link layer medium.

Whenever a new neighbor is injected into R, x initialize a time-out tack attached to that neighbor (line 10). It is used to wait for the acceptance. Allocate set_tack to the beacon holding time approximately, which is the maximum time a node pauses without getting beacons from a neighbor before removing them from its list of neighbors (lines 43-47).This permits nodes to obtain recognition after it gets extra one beacon interval, if the original message was not established initially but received later from other re transmitters.

In other words, it saves some extra transmissions by waiting a little longer for these acknowledgments. If the tack deceases and the acknowledgement is not yet received, the neighbor is transferred from X to Y (lines 39, 42) or if the expected beacons have not been received from the lists. If Y is empty and an original element is injected, to evl is revitalized if it is not running already (lines 40-41). If to_evl is running, it is reorganized to the new value of jYj and the time elapsed from the last schedule (lines 40-41). If N is empty (jYj ¼ 0), x abandons to_evl and chooses not to transmit it again (lines 17-18). When to_evl demises and Y is not empty, x node transmits the message and moves Y to X (lines 31-37). For individual message scheduled in a neighboring b beacon, x abandons the related tack (lines 23-24) and confirms b is in X (when removed from Y) (lines 25). Note that certain acknowledgments can be acknowledged before the message, so that X cannot be empty when the message is first received.

3.4 Improved Broadcasting Algorithm by Adding Neighbor Elimination Scheme

Figure 3.4(A) Demonstrates a potential problem with all described methods (line 18). Node A was broadcast from its only neighbor, B. According to counter, probabilistic methods, location or distance based methods, A retransmits the message even though there is no other neighbor who needs the message. A large supplementary coverage zone can therefore be frequently empty. Even the lowest id clustering can treat A as a cluster- head and impose retransmission on A. Note that A is not an internal node or a relay point for any other point, but not always. We will now suggest an enhancement for each of the broadcasting algorithms deliberated. This enhancement is founded on the reflection in Figure 3.4(A). A node will simply retransmit the message if there is a neighbor who may need the message. Some neighbors are therefore eliminated for retransmission.



(A) (B)

Figure 3.4: (A) Cluster nodes formation for receiving message from its only neighbor node. (B) Node formation scenario for eliminating neighbor nodes from its broadcast list.

First, each node not supposed to be retransmitted by the clustering method is assigned to its possibly retransmitting neighbors. This neighbor is the appropriate cluster head in a clustered structure. Next, we propose to assign the neighbor internal node structure as follows: every non-internal node A is assigned to the neighboring internal node B, which has the highest degree. Use the lowest id between candidate neighbors in case of ties occurred. This rule assigns additional neighbors to greater degree nodes, so that the allocated table of low degree internal nodes can be "emptied."

Secondly, neighbors who acknowledged with one of the copies of the messages arriving at node A. Next they are removed from list of A neighbors, who may need the message. According to Figure 3.4(B), this received the message transmitted by neighbors B and C two times. Neighbors E and F are eliminated from the broadcast list, since neighbors B and C respectively received the same broadcast message. Though, node A will still retransmit the message in this instance because neighbor G is not "covered "by node B or node C. E, F, and G nodes are either internal nodes (that is, the dominant set) or non-internal nodes assigned to A.

This scheme will reduce retransmissions in a dissemination task further. MAC scheme determines the efficiency of this pattern. The dominant set (most powerful concept of the gateway node) and the neighbor elimination-based broadcasting structure are be described below:

For each node j triggers the succeeding procedure broadcast-receive (j, k, and l) upon getting broadcast data from a neighboring node k.

Procedure *broadcast-receive (j, k, l)*;

If *gateway* (j) **then** {

    If k *received* for the first time, **then**

        {

            Time slot decides the rebroadcasting;

            Forward-neighbors (j) = set of all neighbors of j

        };

    For each neighbor w of j **do**

        If d (l, w) <R **then**

            forward-neighbors (j) = forward-neighbors (j) — w;

            Wait until time slot for rebroadcasting;

        If *forward-neighbors (j)* is nonempty and m not

            already rebroadcasted by j

        then rebroadcast the message;

Neighborhood elimination schemes and internal nodes need to distinguish the meticulous position of all its neighbors (if GPS or any other location technique is used) or to distinguish the neighborhood list for each neighbor. This significantly reduces the overhead communication in the presence of mobility node compared to the multi-point relay method [21], since the multi-point relay method wants node neighbors that triggered topological modification to respond by informing their entire neighbor about their fresh relay status. In the overhead communication, where d is the average node degree in the network, a reduction of about d times is achieved. Furthermore, the fixed selection of internal nodes can be used as a virtual backbone for routing through reduced routing tables. This virtual backbone is not provided by the multipoint relay method.

3.5 Solving Intersection Problem

In order to reduce many transmissions while maintaining reliability, we develop a broadcasting backbone based on heuristic CDS. A node in the CDS selects the shortest time-out and gives it the maximum precedence for retransmission. Furthermore, NES is used to diminish various redundant transmissions. This approach applies to vehicle scenarios with intersections in urban layouts proposed by [47]. Vehicles placed at junctions that are the only vehicles connected to other vehicles on the roads are designated as the dominant nodes. The dominant nodes will to spread the message on those streets. In forward selection for VANET specific protocols, the transmitter concept is applied in [26],[48] (Even for safety related applications).Other approaches, as suggested by [47], [49], are required to address the case of intersections explicitly by launching new directional broadcasts.

In the implementation model of the unit disk graph (UDG), two nodes s and t are neighbors and can communicate directly if physical distance (s, t) <= cr. ('cr' - is the communication range radius). The current definition of CDS in a realistic physical layer is complicated; the connection among two cars is probabilistic, so it is not even clear when their neighbors should be declared. Indeed, UDG-based approximation is used to define CDS here, but then we demonstrate that such simple approximate CDS use is sufficient for acceptable performance in realistic VANET physics. Manipulating a CDS in a VANET environment is periodically triggered from the permitted beacons with geographical information. Using acknowledgement means that the protocol is suitable to the VANET atmosphere. If a message is not acknowledged via a theoretical neighbor, the latter will not announce that it will be received in the subsequent beacons and the nodes with the data will deliver a fresh message.

When message is acknowledged from a theoretical non-neighbor, no retransmission will take place later if the node unexpectedly converted as neighbor. The supremacy of our design over the ad hoc PBSM wireless protocol is achieved by correcting the initial assessment of the UDG model. PBSM apprises the X and Y list discreetly on the basis of the UDG model.

3.6 Resolving Broadcast Storm Problem

Most of the effort involved in the development of new forwarding procedures is built on: local topology, statistical distance, position, map and timer proposed by [15], SRD and AID, used trajectory-based approaches. Problems caused by the flooding of message in the network are redundancy, collision and dispute. As discussed in the literature review there are some methods used to solve these issues, still there is possibility to achieve more efficiency in term of message overhead. We use the region on interest method. Our main objective is to disseminate data in an area of interest with high rate of delivery, low message overhead and short message delay. Approach namely, broadcast conquest and delay de-synchronization is used to do this. We use a preference zone to cut the broadcast storm issue and a delay de-synchronization mechanism to diminish the synchronization issues affected by the 802.11p.

The preference area is assigned as a region where its nodes are best used to endure the dissemination of data. The nodes from the neighbor eliminating table selects the clusters and the cluster head from the preference zone region. Vehicles within preference zones are most probably spread the data and reach a larger amount of neighbors that the earlier transmitter could not reach. The concept is that a node within the preference zone has a highest precedence to be transmitted, so that the vehicles within the zone obtain inferior delays than those outside the zone. The delay de-synchronization approach obtains a waiting time calculated by the conquest mechanism of broadcasting and, if essential, recalculates a new waiting time based on channel-switching regime. Algorithm 1 inputs are the set of vehicles inside the area of interest (AI), the road side component initiates dissemination, the vehicle coordinates that transmit the data (m_x, m_y) and the vehicle coordinates that obtain this data (n_i, n_v). With these inputs, we can calculate the time to strategy messages to endure the process of data dissemination.

When a node n within the zone of interest obtains a new message, the node computes its distances to the transmitter (distToSender) and the typical delay based on the distToSender value and the broadcast range. N subsequently checks whether or not it is in the preference zone. In the preference zone, nodes are given smaller delays compared to nodes outside to have a higher precedence for retransmitting the data. In any case, n calculates the waiting time to circulate the constraints to the de-synchronization technique (shown in Algorithm 2), which gauges additional time (Td) to schedule the message. Though node n is waiting for retransmission, if n eavesdrops a replica of the message from a detached node, it abandons its retransmission, else the retransmission takes place. Such procedure goes on till the entire area of interest is covered.

Algorithm :1

Input: AI // area of interest of set of vehicles

Source // RSU that starts the data dissemination

(sx, sy) // Emitting vehicle coordinates

(rx, ry) // Receiving vehicle coordinates

Output: time compound Scheduled transmission

foreach n $\in$ AI **do**

if new message **then**

distToSender = $\sqrt{(m_x - n_x)^2 + (m_y - n_y)^2}$;

Assign defaultDelay = 0.01 x (*distToSender/communicationRadious*);

if (in *preference zone)* **then**

Delay = *random* (0, 0.01)+ *defaultDelay*;

end

else

Delay = random (0.02, 0.05)+ *defaultDelay*;

end

T_d = desynchronize(*Delay*);

n. Schedule-Message(T_d);

end

else

if *Scheduled message* **then**

if *Dist (n, Source)* < *Dist (m, Source)* **then**

Cancel message scheduled;

end

end

Discards the received message;

end

end

Algorithm- Desynchronization Approach

```
Input: T
    Tc // each channel assigned with 50ms
    Ts/ / channel swapping time remain
if CCH is live then
    cch_cycles = ⌊T/Tc⌋;
    Ta ← Ts + (cch_cycles * Tc );
    Td ← T + Ta ;
    end
    else
        Td ← T;
        Ttmp ← T-Ts;
        if Ttmp > 0 then
            chh_cycles ← ⌈Ttmp/Tc⌉;
            Ta ← cch_cydes x Tc;
            Td ← T +Ta;
        end
    end
    return Td
```

Next, we have employed a desynchronization technique that obtains the original delay computed from Algorithm I and a new Td delay is computed by engaging the switching channel regime if necessary. Algorithm 2 demonstrates how this technique works. The core concept of Algorithm 2 is to append an extra time T to the original delay (Delay). This additional time Ta, is the amount of time that m would observe the CCH (Control Channel) as live if m was directly transferred to the MAC layer and then pause for seconds to be transmitted to the SCH. As a sample, a node obtains a message at the time of Ti, in Figure 3.4. The algorithm determines a delay, T= 60ms to send the message again. For reference, the CCH is active in time T1 and remains for a time T_s= 5ms.

3.7 Route Selection and Data Forwarding In Opposite Direction

Data forwarding and route selection algorithms mostly use the Greedy forwarding technique. Greedy Perimeter Stateless Routing protocol uses packet forwarding and perimeter forwarding. The right hand rule is applied for routing when some void is present in the network. According to right hand rule, when movement between any two neighbors say node x, node y, the next sequential anticlockwise edge will be traversed first. The shortcomings of greedy forwarding:

- The accuracy decreases if the nodes move due to mobility and packet will loss. It results into packet loss. It is also possible if a neighbor table entry is outdated which cause excessive data re-sending.
- Link-layer transmit of the beacons will result in missing of recovery from failure. This leads to failure in transmission, because nodes being close to each other are not recognized as such.
- Network load will boosted due to the beacons.

TrafficView mobility model computes the on-road relative position of other vehicles. Two types of broadcast data being used: generated data and relayed data.

- Generated Data: vehicle's own data, i.e., ID, speed, and location.
- Relayed data: stored data about the other vehicles ahead and it is propagated backward.

When a vehicle broadcasts a data packet, only vehicles moving in the same direction are responsible for packet propagation, which were used in the previous woks. Our proposed protocol uses opposite direction also for packet forwarding. Figure 3.5 shows the traffic scenario in urban highway. 2lane highway with a median consists of many vehicles moving in specified directions. Due to obstacle traffic in one direction is blocked.

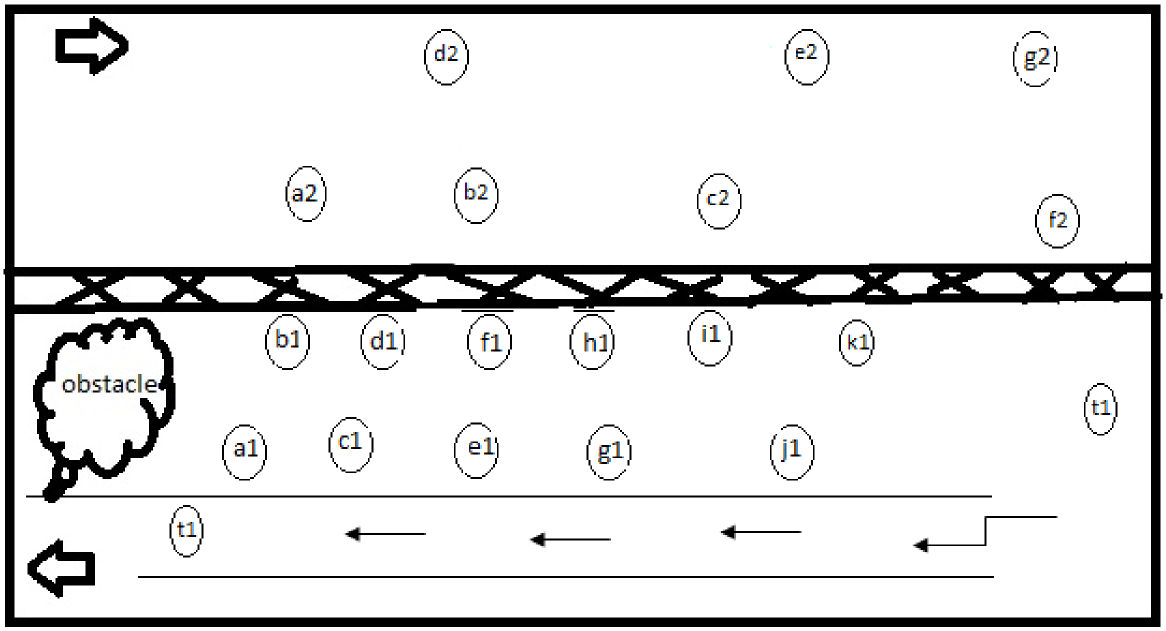


Figure 3.5: Urban 2Lane highways.

Vehicles in that lane namely a1, b1, c1… etc is blocked. Vehicles in other lane namely a2, b2, c2…etc is moving freely. Route selection and forwarding of packets are performed according to the newly proposed algorithm. Here the routing table consists of source node, neighbor list, and current status. Node T1 moving in direction of the traffic jam or obstacle, via beacon signals node t1 receives its neighbor information. After applying the neighbor elimination scheme, t1's neighbors are node f2 and k1.

The proposed algorithm works in three phases. In phase1 required variables and values are initialized. Collision occurrence and avoidance procedure is implemented in phase2. Route selection and data forwarding is implemented in the last phase.

Proposed algorithm works as below mentioned steps:

Input:

1. Source node S
2. Neighbor List N

Auxiliary Variables:

Current status (ideal/moving);

Output:

Done // if Greedy forwarding is successful

Error // if Greedy forwarding is not successful.

Begin:

Phase 1: Table analysis

Step 1: Apply Neighbor elimination scheme;

Step 2: Update the table;

Phase 2: collision verification

Step 1: if

Node t1 moves towards the obstacle direction, collision occurs go to phase 1;

Else end;

Step 2: Analysis the relay information obtained in the opposite direction.

Step 3: if current status – ideal collision occurs;

Step 4: if current status: – moving, select new route go to phase-3;

Phase 3: Route Selection and Forwarding

Step 1: Get the geographic location of all nodes in route.

Step 2: Find out the distance of destination node from source node with intermediate nodes using Dijkstra algorithm.

Step 3: Select a node in the route of destination node with shortest path data towards the destination node.

Step 4: Forward packet to the destination node. If reply or acknowledgment of the packet is not coming from destination node from a long period of time then next phase will start again to regenerate Route after reinitialize parameter in phase 1.

3.8 Chapter Summary

This chapter described in detail the implementation portion of the proposed protocols. The protocol design discusses about the formation of connected dominated sets and its correctness. With the identified dominated sets, the neighbor elimination scheme is applied to get intended neighbor for transmissions. The improvement of broadcasting protocol been achieved by adding the neighbor elimination scheme. In the next part, the intersection problem is resolved with the new approach. Lastly, the broadcasting storm issue has been resolved using new preference zone approach. Route Selection and Data Forwarding in Opposite Direction is addressed with new algorithm. The next chapter describes the simulation setup used for experimentation and the results achieved.

CHAPTER-4 Implementation and Results Discussions

4.1 Theoretical implementation and result discussion of Connected Dominating set formation

A VANET can be demonstrated using graph concepts with help of E-edges and V-vertices. Two graph nodes are only connected if there is a connection among them. When the network is denoted as a graph, the subsequent problem is whether some graph property affects the VANET. For instance, a dominating set D of a graph is the set of vertices where a vertex $v \in V$ is both in D (and in neighboring set) or neighboring to a vertex in D. When dominating set vertices are connected, the dominating set is named as a connected dominating set (CDS) and creating a CDS in the VANET graph model provides a backbone for routing purposes in the real mobile network.

Since, discovering a minimum DS or a CDS is a NP-complete problem in graph theory and therefore approximation algorithms for problems wherever suboptimal answers using certain heuristics is typically the solitary option. Nevertheless, manipulative an approximation algorithm with a promising approximation proportion to the problem is not enough because you have no global information about a real network. A distributed algorithm is run by all adhoc nodes, exchanges data through its neighbor nodes by passing data only and ultimately leads to a certain network status.

4.2 Design of Network Models

For the design and implementation of connected dominating sets, first the network models are developed. Then the topology control models are constructed.

4.2.1 Implementation of Unit Disk Graph (UDG)

A UDG is a special graph instance in which each node is recognized by a unit radius disk *rd*=1, accept that there is an edge among two nodes u and v if and only if the space among u and v is not greater than 1. The model is shown in Figure 4.1(A). The communication range of each node is drawn as a dotted circle. The edges that link nodes are drawn in straight lines. The node u neighbors are node v, node w, node y and node z, in the simplified graph shown in Figure 4.1(B). Even though UDG is comprehensively used network model, its simplicity causes disadvantages. In actual configurations, even minor barriers among collaborating parties can interfere with wireless transmission, so UDG is not a realistic model for ad hoc networks on heterogeneous object areas. It does not model the signal excellence among nodes, therefore the topology for multi-hop communication can be poor. In addition to that, node weights containing of node energy, mobility, etc. are not demonstrated. This makes UDG unsuitable for selecting high-weight node routes.

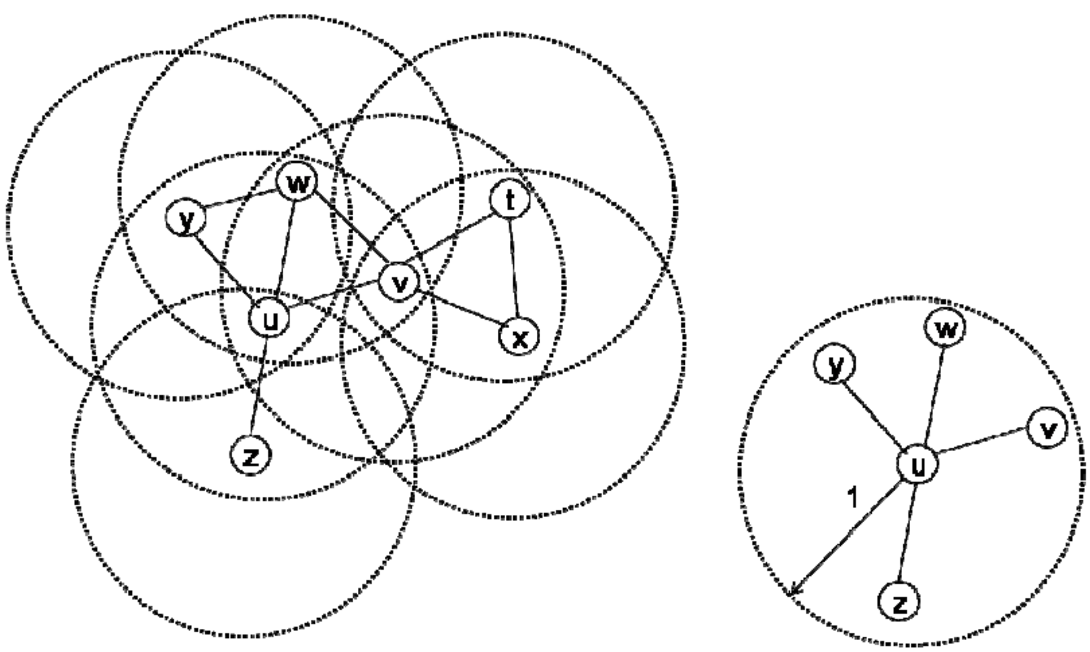


Figure 4.1 (A) 4.1 (B)

Figure 4. 1 :(A) Unit Disk Graph Model (B) Node *u*'s Neighbors

4.2.2 Undirected Graph (UG)

An undirected graph is designated as F= (V, E) where V is the set of vertices or nodes (V= {V_1, V_2, V_3… V_N}), and E is the set of edges among them (E= {E_{12}, E_{21}…}). E_{xy} is an edge that starts at vertices x and ends at vertices y. Subsequently the graph is undirected, links are presumed on both sides as starting and ending. An illustration network with 10 UG-model nodes is shown in figure 4.2. The set of vertices in this model is V= {V_1, V_2, V_3…, V_{10}}, the set of edges is E= {E_{16}, E_{61}, E_{26}, E_{62}… E_{910}, E_{109}}.

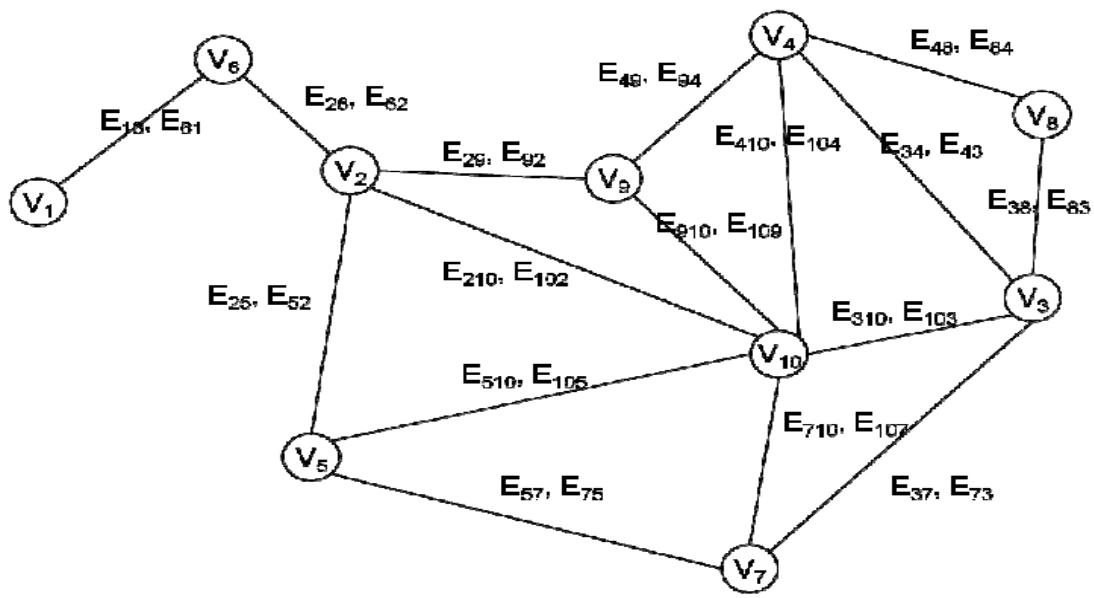


Figure 4.2: Undirected Graph Model Construction

4.3 Topology Control Models

4.3.1 Independent Set (IS)

IS is a set of nodes that are not adjacent to any of the nodes. If this set cannot be prolonged with the addition of a different node, the maximum IS is named. The maximum IS is called, the IS with the biggest number of nodes. Figure 4.3(A) shows that six gray nodes are the maximum ARE elements.

Nevertheless, this set cannot be prolonged by tallying a new node, eliminating some nodes from this set and adding other nodes may escalate the size. Figure 4.3 (B) shows the maximum IS of 8 nodes. The weighted form of this problem allocates a weight to each node and aims to maximize the total weight of this set. Maximum weighted IS with a whole weight of 57 is shown in Figure 4.3(C). The construction of the IS is used in ad hoc systems to locate facilities and form backbones. Facility location problem shields the optimum location of facilities to minimize the associated costs. The designated IS nodes can link with each other by increasing their broadcast range to distribute and route data to non-IS nodes. If the network is clustered, the selected nodes can be cluster heads. The maximum IS problem and its weighted version is NP hard.

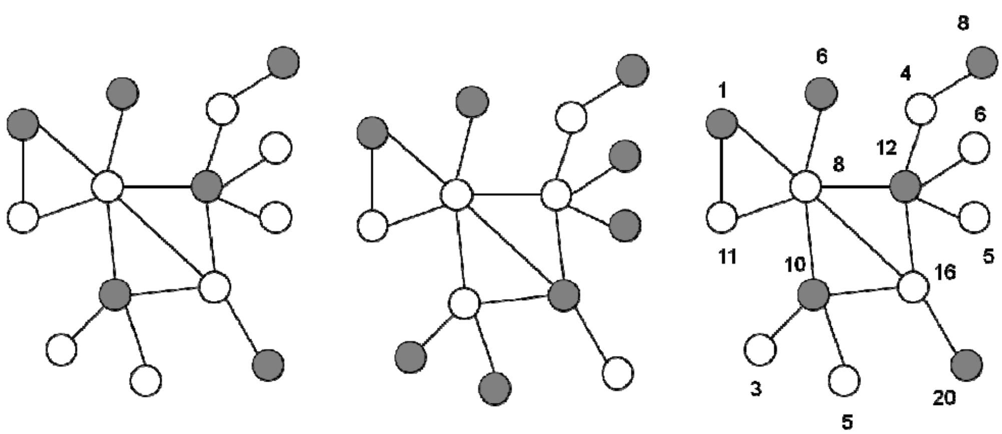


Figure 4.3 (A) (B) (C)

Figure 4.3: (A) Maximal Independent Set (B) Maximum Independent Set (C) Maximum Weighted Independent Set.

Thus maximal IS algorithms and its complexities are deliberate in the literature. A practical distributed procedure proposed in [1] that finds maximum IS and discovers maximum weighted IS if node weights instead of node ids are used. All nodes are in the *WHITE* state at the initial stage of the algorithm and are all candidates aimed at the IS. If node's id is larger than its neighbors, it directs a message *IamInTheSet* and becomes an IS component. Once a node obtains a message from *IamInTheSet*, it directs a message from *NotInTheSet* and does not become a candidate. After a candidate node accepts a *NotInTheSet* message, it first removes the source node from its neighborhood and then directs an *IamInTheSet* message and becomes a component of IS if a node's id is larger than its neighboring are.

4.3.2Dominating Set (DS)

A dominant set is a subset S in graph F so that each F vertex is either in S or adjacent to S vertex. Minimum dominating set issues are NP-complete. A maximal independent set is assumed to be a dominating set. Dominant sets are broadly applied for topology control, where elements are selected as cluster heads in dominant sets. Dominant sets can be categorized into three key classes, Independent Dominating Sets (IDS), Weakly Connected Dominating Sets (WCDS) and Connected Dominating Sets (CDS) shown in Figure4.4.

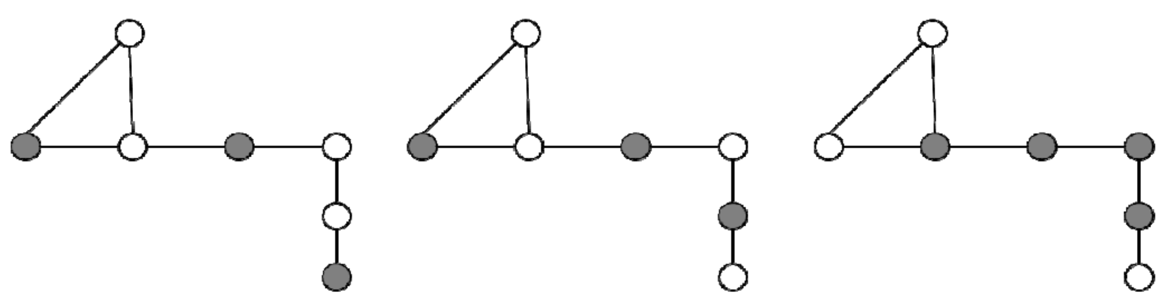


Figure 4.4 (A) (B) (C)

Figure 4.4: (A) Example of Independent Dominating Set (B) Weakly Connected Dominating Set (C) Connected Dominating Set

Independent Dominating Sets (IDS): IDS is a dominating set *S* of a graph *G* where no adjacent vertices exist. Figure 4.4(A) shows a gray-filled independent dominant set. By means of independent dominating sets, this can make sure that no adjacent cluster heads are present in the entire graph. This reduces the number of unwise clusters within the network.

Weakly Connected Dominating Sets (WCDS): A weakly made subgraph *Sw* is a subset *S* of a graph *F* with the vertices of *S*, their neighbors and all edges of the original graph *G* with at least one endpoint in *S*. A subset *S* is a weakly connected dominating set, if *S* is dominating and S_w is associated. Gray nodes show an example of the WCDS in Figure 4.4(B). Even though independent dominant sets are appropriate for optimal sizes, they have some shortcomings, such as a lack of traditional communication among cluster heads. WCDS can be employed to build clusters to achieve connectivity among cluster heads.

Connected Dominating Sets (CDS): A connected dominant set (CDS) is a subset S in graph F so that S forms a dominant set and is connected. Figure 4.4(C) displays the CDS sample. The principal problem in building connected dominating sets is the minimum connected dominating decision problem.

Below are the phases of the Wu's algorithm:

1. Each node *u* discoveries the neighborhood set г (u)
2. Each node *u* transmits г (*u*) and receives г (*v*) of all neighbors.
3. If node *u* has two neighbors *v*, *w* and *w* is not in г (*v*) then *u* sees itself in the set CDS.

In Figure 4.5, shows the output of this algorithm.

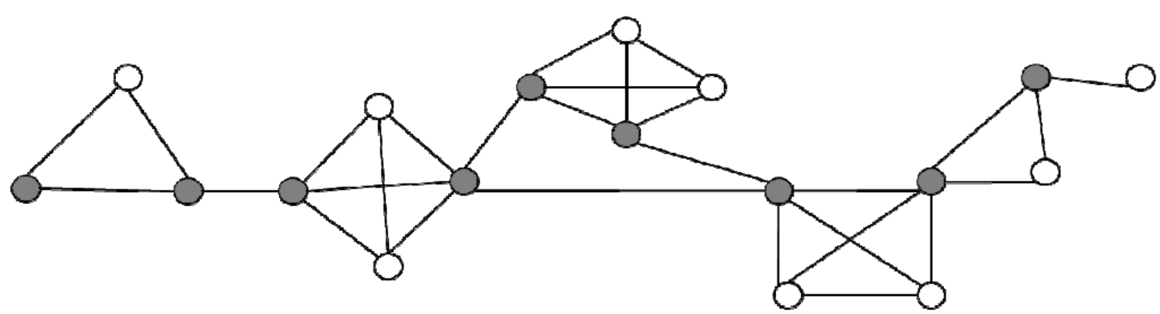


Figure 4.5: Construction of Connected Dominating Set using Wu's Algorithm.

4.4 Simulation

In communication, wireless technology has become an indispensable structure for day-to-day communication of messages. The researchers are interested to modify the current technology to suit the user requirements. The modern wireless devices are mostly made up of existing and limited infrastructure. Ad - hoc networking is a popular approach to non-fixed infrastructure. It has an answer to the wireless application's infrastructure limitations. Specifically, in an adhoc network, the node movements are dynamic and with no-boundary's. Vehicular adhoc network is sub class of mobile adhoc network. The real time environment requires very intensive experiments and evaluation methods, which are very hard to carry out. Among all the available types applications, the least possible to implement in real time are distributed applications, in which testing can be carried out with the predictable fixed set of data.

Due to pragmatic considerations, the rest of the application execution and testing in the real-time environment could not be valid at all. To deal with such environments requires simulators for possible implementations. The simulator generates almost possible topology and scenarios. The topology is kept in (2D) two - dimensional arrays containing the adjacent node status. With reasonable modifications in neighbor matrix gives the anticipated mobility scenarios.

4.4.1 NS-2 Simulator

NS2 was developed at ISI, California, as a discrete event simulator. NS2 offers significant provisions for the simulation of many routing and multicast protocols. In 1989 ns2 was developed as variant for REAL network simulator. Over the years, NS has been developed with support from projects such as VINT by DARPA. NS2 has included significant contributions from other researchers like Sun Microsystems, CMU Monarch and UCB Daedelus. The NS2 is a powerful open source tool for simulating and testing wired and wireless system applications. Its early versions are designed only for wired networks; with the addition of various extensions, it supports the wireless networks also. It closely follows the Open Systems Interconnection (OSI) model. NS2 is developed in C++ as a front end with an OTcl interpreter. User applications are implemented in C++, and replication scenarios and set-up are developed using OTcl. Currently, the simulator is an event-driven single-threaded program, only one event at a given time can be executed. If additional events are needed to run simultaneously, the built-in scheduler will execute them in a first-come first - served fashion (FCFS manner) and the next earliest event in the queue will be completed.

Creation of nodes is carried out with the help of OTcl classes. A node contains address or id, a routing module and a node type. Nodes id is assigned initially to zero then increased monotonically. The prolonged class is titled mobile-node. The stack structure of these nodes built around an ARP module, a link layer (LL), next with a border priority queue, then with a MAC layer and a network interface (netIF). OTcl used to create these components and built a strong channel for transmission. NS2 supports both unicast and multicast simulation, in which unicast is a default mode.

4.4.2 Protocol Implementations scenarios

Variety of protocol is implemented in NS2 according to the user personification. Simulation is accomplished in three major phases. The first step consists of, implementing codes using C++ and OTcl to the ns2 source base. With OTcl, script simulation is described in the second step. In final step, ns2 runs in command mode which results in generation of trace files. The make files are written in C++ under the same folder of the protocol, and then the OTcl scripts run along the make files generates simulation output. Nodes in the scripts are initialized with network parameters such as interface queue type, antenna model, radio propagation model, channel type, MAC layer type, and link layer type. The physical parameter is locations, movement scenarios, start-end times and moving directions.

Once the nodes are molded in the OTcl file, individual node can be linked to different (or the same) protocols by means of loops individually. Now, theses parameters are executed along with the make files to find the trace files. Then, the trace files are displayed directly or stored in a file for data analysis. Scenario Generations: (3D) three-dimensional topology is designed to capture the mobile node movements. In first alternative method: speed, start-end postposition is assigned randomly. With the fixed time slot, nodes are moved from the initial position to its destination at a defined speed. These movements are stored in a trace file. In the second method, speed is varying dynamically. Here nodes' starting positions are generated initially. During the simulation the user, dynamically assign nodes destination and speed values. Various network traffic patterns also can be generated in ns2. The likes of constraint bit rate and TCP connection scenarios are generated in wireless mobile scenarios using these traffic patterns.

4.5 Construction of Connected dominating set using simulation:

For the construction of the connected dominating sets, ns2 simulator is used with OTcl as scripts include in the Appendix A.

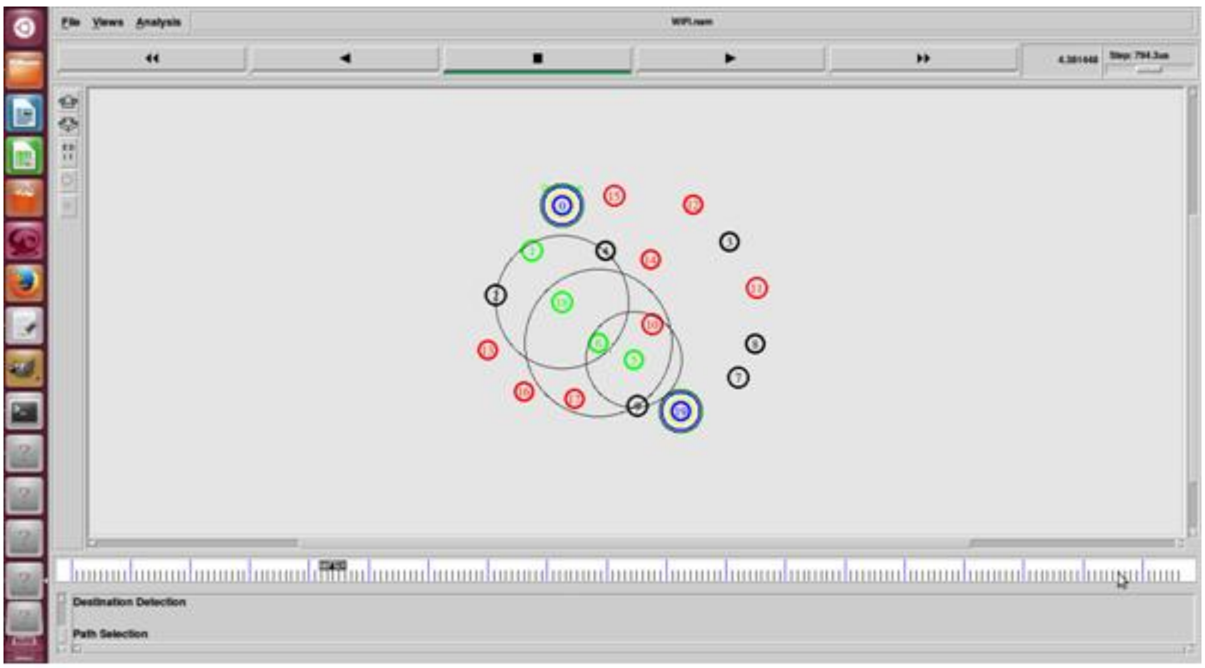


Figure 4.6: Sample CDS construction.

Algorithm is implemented as CDS heuristic approach shown in Figure 4.6.

The time-out parameter to_ev is calculated as,

$$to\_ev = \begin{cases} W/|N|, \text{ if in CDS,} \\ W.\ (1+ (1/|N|)), \text{ otherwise} \end{cases}$$

While *to_ack* is fixed to a constant value.

SUMO microscopic road simulation package is used to replicate the highway and suburban scenarios. The common situation likes vehicles over taking and waiting in intersections are easily simulated here. Which resembles the intermittent connectivity and uneven distribution of vehicles for various scenarios? Based on the assigned traffic rate the cars are injected on the road. The traffic injection rates various from 1=75 to 1=5 vehicles per second, also it depends on the network density.

Considering the suburban scenario, at a lower rate (1=15) the density is lower than the highway setup at a higher rate (1=5). The table 4.1 describes the various parameters of the simulation scenario for analysis.

Table 4.1: Simulation parameters for the vehicular Scenarios

Simulation time	120 secs
Beacon interval	05 secs
Beacon hold time	1.5 sec
Waiting time (W)	(0.10,0.25,0.50) sec
Acknowledgment time (To_ack)	(0.60,1.10,1.60,2.10) sec
Contention Window	802.11p
Traffic rate	(1/75,1/60,1/45,1/30) veh/sec/route

Every cycle involves of a transmission task initiated by an arbitrary source selected from a subset of nodes. During the subsequent iterations, the vehicle must be active when the steady state is reached. With payload data of 500 bytes and a lifetime of 120 sec are assigned for each cycle, afterward it is discarded according to the section 4.4.2.

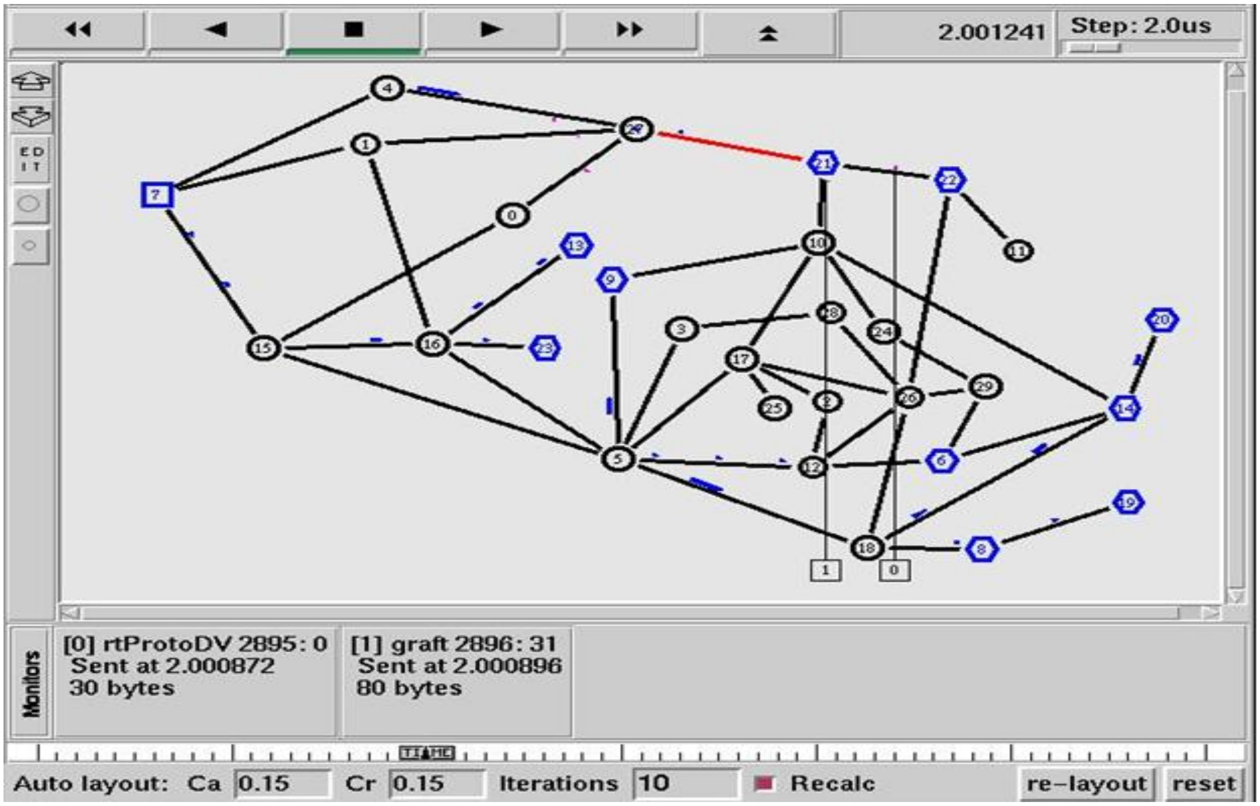


Figure 4.7: Construction of CDS with unit disk graph model (UDG).

Performance metrics used for evaluation are Packet Deliver Ratio, End-to-End Delay and the Packet Routing overhead. The simulation contains the following per-setup parameters.

- Numbers of nodes.
- Transmission range.
- Packet size is assigned as 1400 bytes.
- Node speed is assigned as 10 m/s
- Mobility model is Random way Point

4.5.1 Packet Delivery Ratio

The analysis of Figure 4.8 shows that AODV protocol has performed better than the other protocols when the number of nodes increases, when the number of nodes becomes more stable, a stable path is obtained. In DSDV protocol link breaks occurs when there is a increase in the number of nodes, which leads to more packets drops; hence, it gives lesser performance than the other protocols. Our proposed protocol i.e. broadcasting protocol is better than DSDV protocol, since it finds the new route when link break occurs.

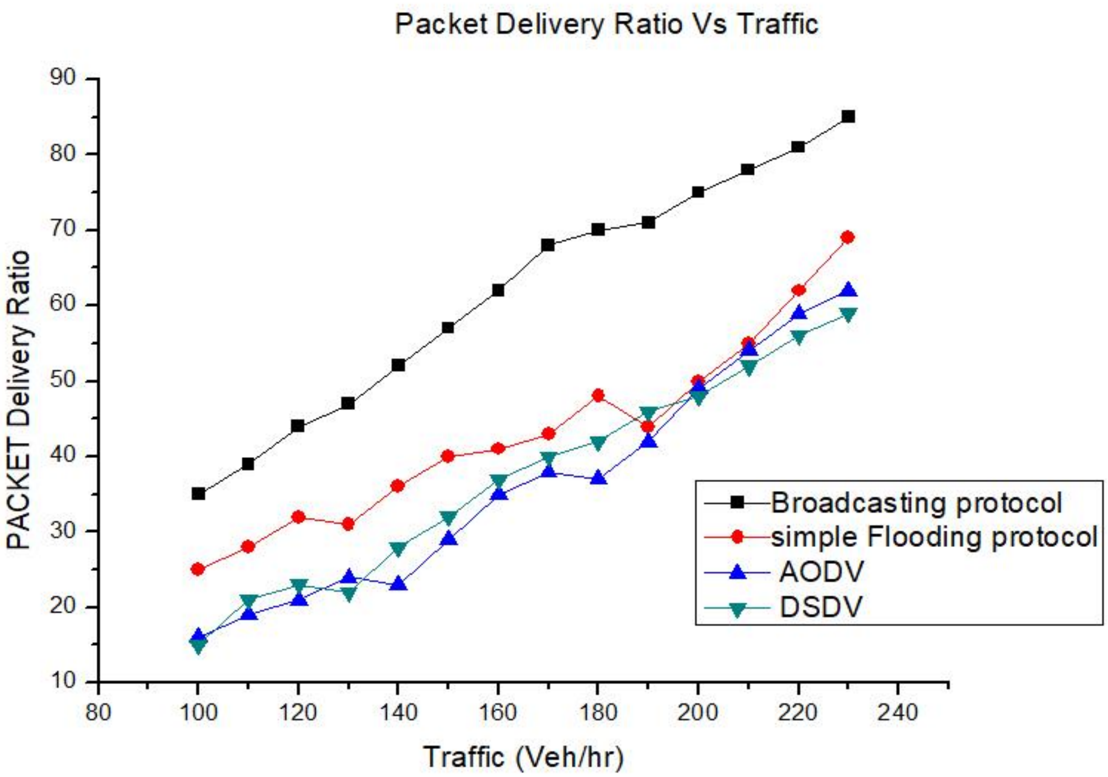


Figure 4.8: Packet Delivery ratio

4.5.2 Routing Overhead

From the Figure 4.9, it is inferred that the DSDV protocol less prone to route stability factor when compared to AODV protocol. Broadcasting protocol makes fewer routing overhead factor, when compared to DSDV protocol, since it does not affect by the number of nodes. For AODV, routing overhead is not as affected as in DSDV.

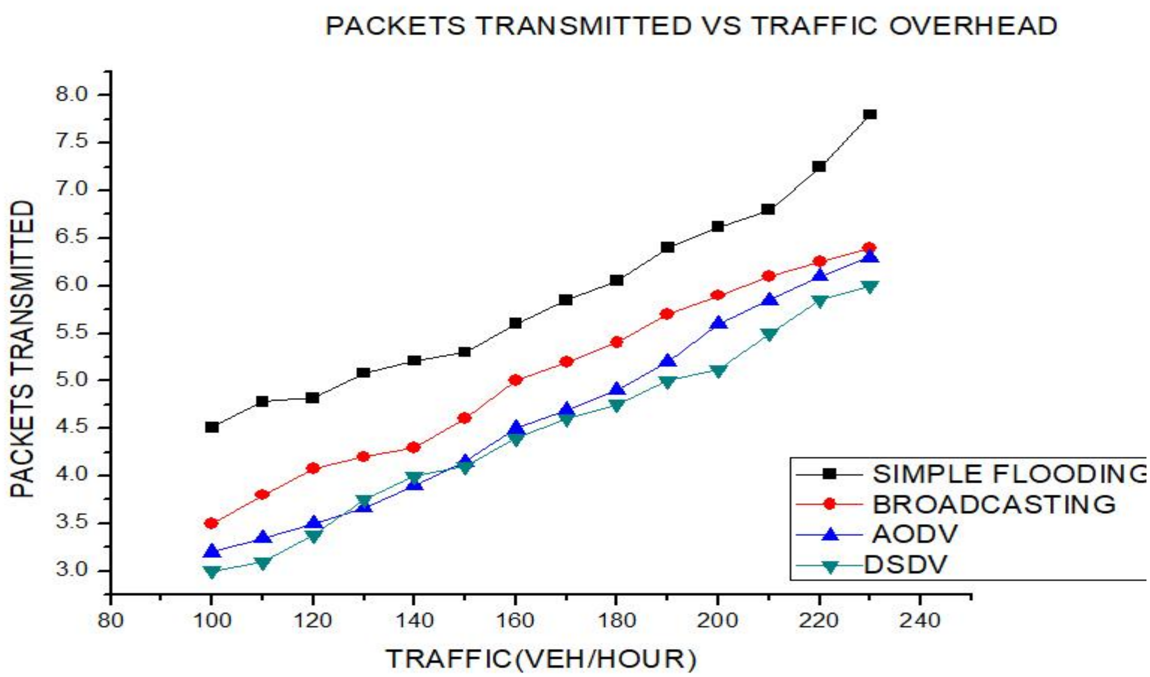


Figure 4.9: Routing Overhead

4.5.3 End-to-End-delay

In Broadcasting protocol, even when the number of nodes increased it does not produce much delay as shown in figure 4.10. Hence, the proposed protocol is better than the other three protocols.

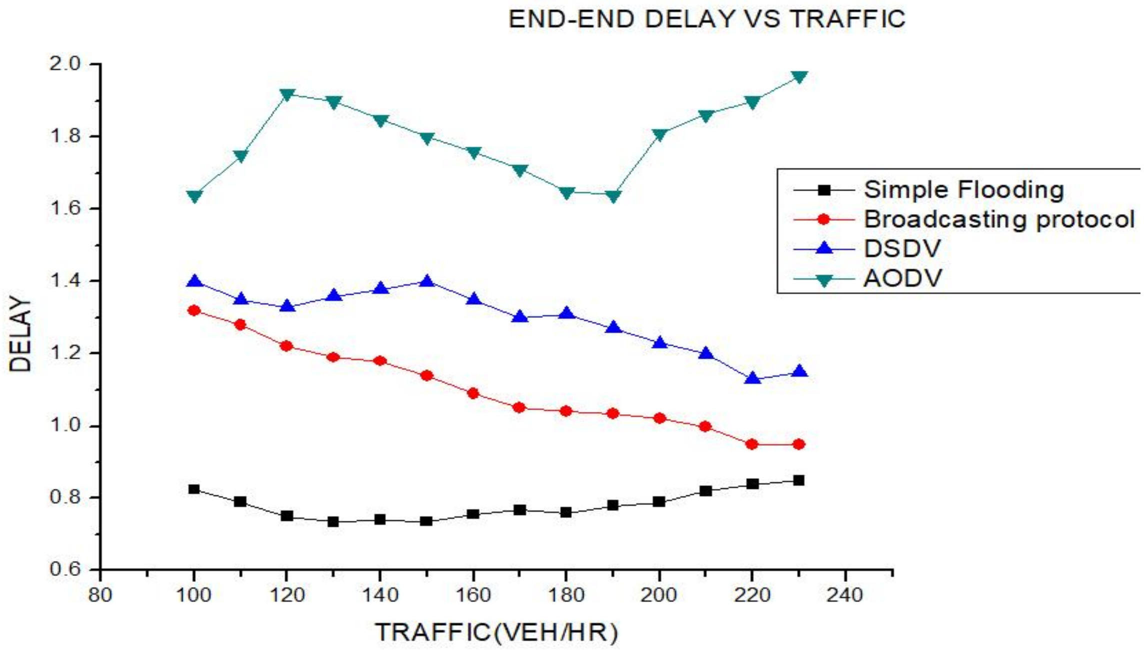


Figure 4.10: End-to-End-delay

The efficiency of the broadcasting algorithm in VANETs depends on how the connected dominating sets are constructed. In addition, internal node concepts have significantly reduced the maintenance cost compared to cluster head maintenance. Performance of our broadcasting protocol been compared with the existing protocols such as AODV, DSDV and simple flooding by using metrics as Packet Deliver Ratio, End-to-End Delay and the Routing overhead. The simulation results show that our protocol is superior to the DSDV and AODV with respective to Packet Deliver Ration and the End-to-End Delay when the node presence are increased more. In Routing overhead aspect, it consumes more computation overhead rather than DSDV protocol in the case of mobility factor; it yields modest performance when compared to AODV protocol.

4.6 Simulation evaluation of broadcasting storm issue

A problem caused by flooding of a message over a network resulting from redundancy, contention and collision factors is considered as broadcast storm. The region on interest method is applied here. Our main aim is to do data dissemination approach within an area of interest with the consideration of some of factors such as low overhead, packet delivery rate, and short delay. Approach namely, broadcast conquest and delay de-synchronization used to do this. A method called preference zone is applied to cut these issues, which are prompt, by the 802.11p protocol. The preference zone is identified as an area best utilized to achieving data dissemination continually within their visionary. Vehicles settled inside the preference zones area are more likely to spread successfully. By this process it accumulate more neighbors, prior transmitter cannot do this. This leads to achieve higher priority to transmit data in preference zone. With this higher priority, nodes receive lower delay compared to the nodes present outside of this area.

The broadcast conquest mechanism is implemented in initial stages to calculate waiting delay. During the iterations, the channel-switching regime is applied to recalculate a new waiting delay. As explained in the chapter -3, the broadcasting storm issue addressed according to the data dissemination algorithm using ns2 simulator.

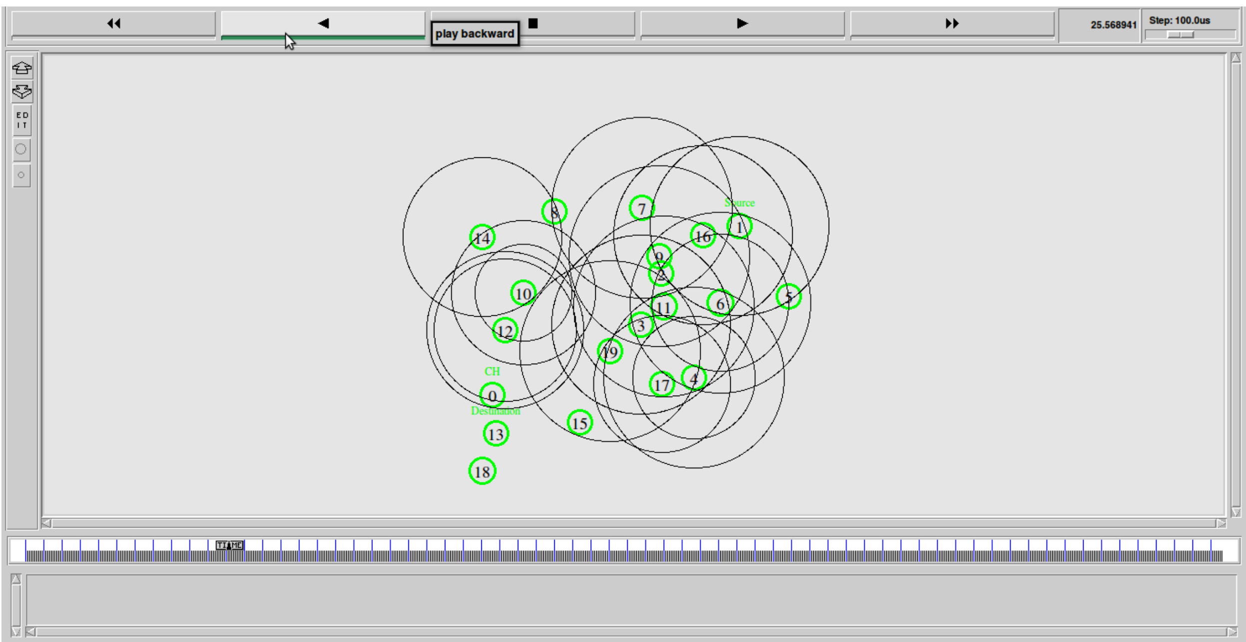


Figure 4.11: Data Dissemination.

To appraise the performance of the proposed protocol along with ns2 simulator, SUMO simulator is used to replicate the highway scenario. For evaluation the traffic conditions are replicate with high, normal and low traffic scenario and compared with simple flooding protocol, SRD [50] , AID [51] protocol. TCL scripts implementation included in Appendix B. Reliability, efficiency, and scalability of the protocol are further evaluated with following aspects:

• Delivery ratio- The percentage of data messages received by intended recipients, which is generated by the roadside unit. A better dissemination protocols are expected to get near by 100%.

• Transmitted messages: An accumulation of data messages transmitted during the dissemination process by all vehicles in the network. Achieving of a high number is the indication of data redundancy during disseminated, which may resulting the broadcast storm.

• Collisions- The mean number of packet collisions at the protocols MAC layer per vehicle to disseminate all data messages. Lower the collisions indicate lower of broadcasting storm present in the protocol.

4.6.1 Evaluation setup

The simulation scenario contains two roads with opposite directions and each road has three lanes with a length of 1 km. During the simulation process, the vehicles enter the highway at one edge of each road, travel entire distance and reach the opposite edge in single iteration. To get the variety of data for analysis, number of vehicles in road varying from 1000 vehicles/hour to 3000 vehicles /hour is generated. Different types of vehicles are deliberated to emulate the overtaking process. The first type of vehicles can reach the maximum speed of 35 m/s, while the other group of vehicles can reach a maximum speed of 15 m/s. Among the vehicles, it may contain passenger cars and heavy trucks. SUMO mobility simulators to simulate realistic vehicles movements are simulated using SUMO mobility simulator, the intercommunication among the vehicles are simulated using NS2 platform.

The message dissemination is achieved by positioning the Road Side Unit(RSU) 500m away from left edge of the highway, then nodes generates 100 messages of 2048 bytes, the dissemination rate is fixed as 1.5 Mbit/s to all neighbor vehicles. Here, 0.5 km of extension is assumed as area of interest. Multi-hop communication method is used to disseminate emergency warning message with right-bound direction to alert the drivers approaching the RSU. The purpose of using RSU is solely to generate emergency warning message only, it is least used in dissemination process.

As described in the earlier algorithm of desynchronization mechanism, an extra time is added in the broadcasts to eradicate the synchronization consequence triggered by 802.11p, which eventually eases the collisions. This formulation indicates a lower delay for proposed protocol than SRD and AID generates. In Appendix C implementation scripts are included. The following result vindicates that it gives a suitable solution for warning message dissemination kind of application.

Figure 4.12 demonstrate that the average number of collisions at the MAC layer per vehicle. Simple Flooding protocol has highest number of collisions, it is a direct pro-potion to increase of traffic.

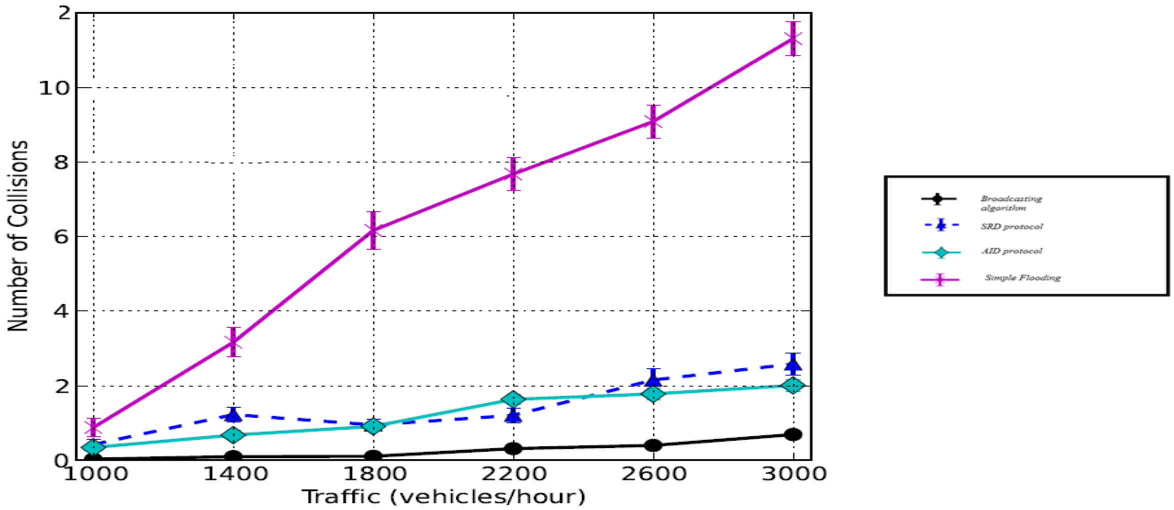


Figure 4.12: Packet Collision

This is happened in the packet retransmission process because of lack of coordination, i.e., many vehicles try to access the channel simultaneously. Clearly, the results indicate that the proposed broadcasting protocol has lowest number of collisions. For further justification, the literature shows ADD has lowest collision rate. Here, proposed broadcasting protocol generates about nearly less than 40%. The above analysis shows that the broadcast protocol can avoid the broadcast storm issues and use the available bandwidth efficiently. With the above analysis shows that, broadcasting protocol is able to avoid the broadcast storm problem and efficiently utilizes the available bandwidth.

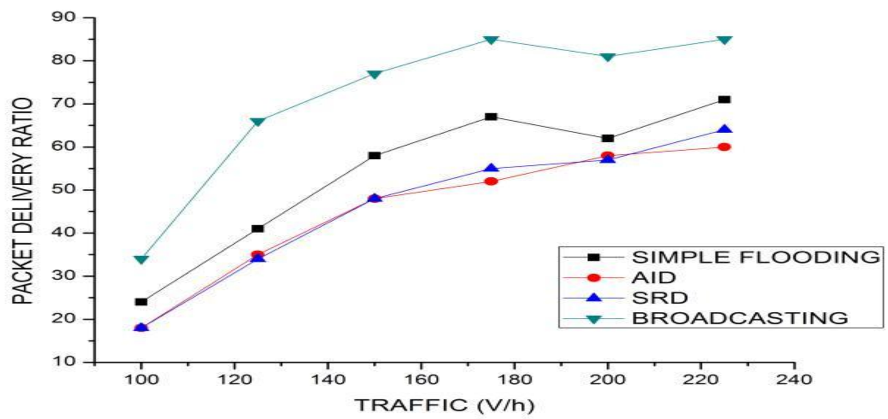


Figure 4.13: Packet Delivery ratio

The Figure 4.14 represents the total packet transmission aspect in detail for different traffic conditions. Broadcast storm for dense scenarios is evaluated with consideration in area of interest region. Broadcasting protocol gives more transmission number compared with other three protocols. In final analysis, the Simple Flooding protocol has high overhead indicator, due to lack of any kind of broadcast suppression mechanism. The examination of packet delivery ratio aspect indicates in Figure 4.13, even the AID protocol, and SRD protocols take fewer transmissions to attain data dissemination, but they present low delivery ratio. Meanwhile, the delivery ratio aspect of other protocols is lesser than the proposed broadcasting protocol in all the simulation scenarios.

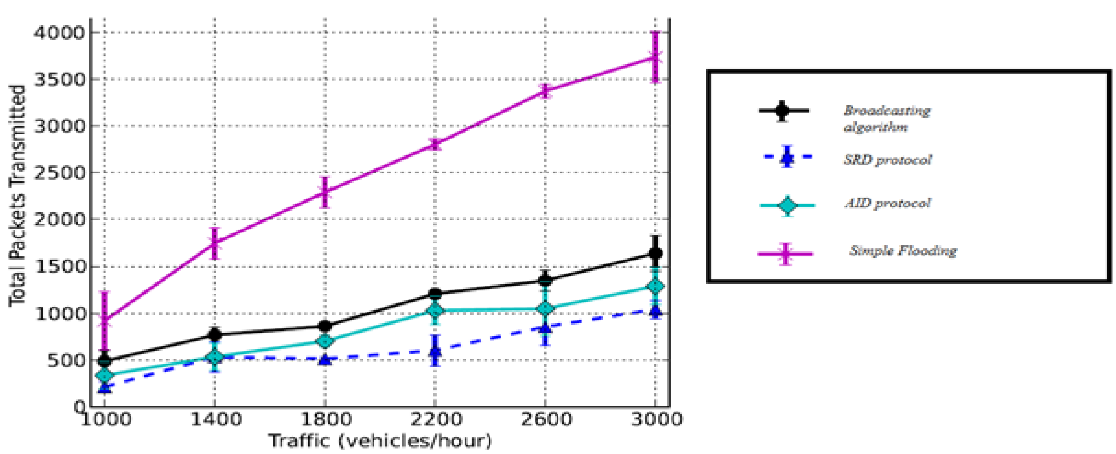


Figure 4.14: Total packets transmitted

4.7 Simulation evaluation Route Selection and Data Forwarding in Opposite Direction of Data approach

NS-2 and SUMO simulator used for basic evaluation of this section. The Comparison is performed against GPSR, GPCR and AODV protocol in Highway, rural and urban scenario. Performance is measured for packet delivery ratio and throughput. Initial location of nodes is obtained using unified location. Sender and receiver nodes are selected primarily for simulation. Nodes mobility model used is Random Way Point.

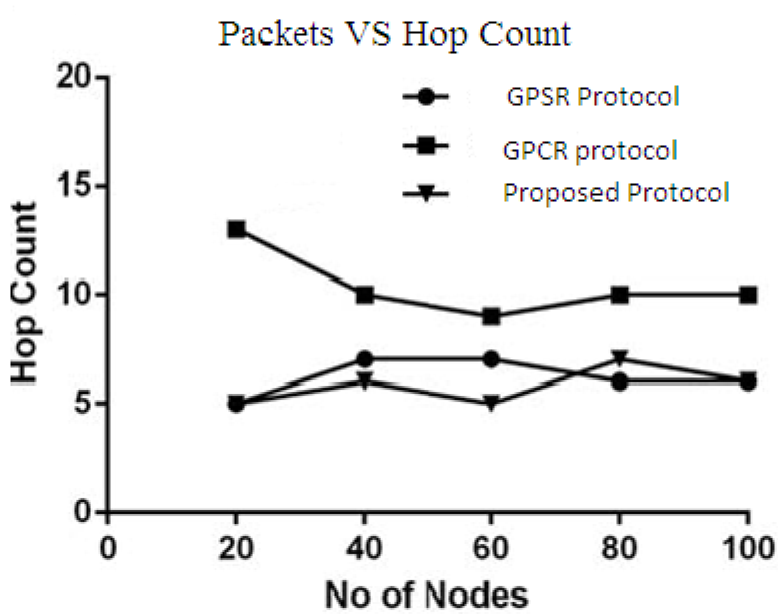


Figure 4.15: Packets vs Hop Count

Performance is measured for packet delivery ratio and throughput. We compare the performance of the proposed protocol to several like GPSR, GPCR. Our Protocol consistently delivers packets at rate of around 85% on various traffic scenarios.

4.8 Chapter summary

This chapter describes the simulation scenarios and the results of all the objectives. For the simulation purpose NS2 and SUMO simulation scenarios are discussed next. With network models and unit disk graph connected dominating sets are constructed. To evaluate the implemented protocol, the performance metrics such as Packet Delivery Ratio, Routing Overhead and End-to-End-delay are simulated and compared with the existing protocols. Next, broadcasting storm issues is evaluated based on the performance metrics. The resultant graph exhibits that the proposed protocols is superior to the existing protocols. Finally the route selection issue has been evaluated against the existing protocols. Next chapter describes conclusion for the project and the future works.

CHAPTER 5- Conclusion and Future Work

The real time implementation for this kind of networks involves huge amount of money and labor, so the latest simulation software is considered as the platform for testing and implementation of new protocol. In this work, several improved methods are developed and the integration of these models provides an improved protocol for VANET.

5.1 Concluding Remarks

In field of VANET in terms of safety related classification aspects, the various routing protocols, their classifications, advantages and drawbacks are discussed in introduction section. Next, broadcasting protocols and selection of dominating sets are studied in specific. Then one of the internal design aspects called broadcasting storm issue, the impact it creates in delay of the message forwarding and the various ways to resolve this issue is elaborated. Next, the intersection section issue in the urban area is studied and their drawbacks are identified as research gap.

Chapter-3 describes the details of the implementation portion of the proposed protocols. The protocol design discussed about the formation of connected dominated sets and the correctness of the algorithm. With the identified dominated sets, the neighbor elimination scheme is applied to get intended neighbor for transmissions. The unit graph models are implemented to construct the connected dominated set as described in Section 4.5. In the next part, the intersection problem is resolved with the new approach. The resultant dominating nodes formation with neighbor elimination schemes shows that, our protocol for making better decisions in finding the available paths without the need of recognizing the intersections.

To evaluate the above implemented scenarios, Packet Delivery Ratio, Routing Overhead and End-to-End-delay are taken as metrics to compare with simple flooding, AODV and DSDV protocols. The simulation results show that our protocol is superior to the DSDV and AODV with respective to Packet Deliver Ratio and the End-to-End Delay when the node is high. In Routing overhead aspect, it consumes much computation overhead for DSDV protocol in the presence of mobility factor, yielding modest performance when compared to AODV protocol.

In next section, the broadcasting storm issue has been resolved using new preference zone approach. The efficiency of the broadcasting algorithm in VANETs seems to be, linked to the creation of a connected dominating set. In addition, internal nodes concepts have reduced the maintenance communication cost compared to cluster structure maintenance. The enhanced Broadcast Conquest and Delay De-synchronization mechanism address the broadcasting storm issues. Section 4.6 elaborates the simulation evaluation for this approach with Delivery ratio, Transmitted messages and Packet Collisions as metrics. The results indicate a lower delay in case of the proposed protocol than the delay generated by the SRD and AID protocols.

Third objective, Route Selection and Data Forwarding in Opposite Direction is addressed with new algorithm implemented in the section 3.7. Performance is measured in terms of packet delivery ratio and throughput. We compare the performance of the proposed protocol to several other protocols like GPSR, GPCR. Our Protocol consistently delivers packets at rate of around 85% on various traffic scenarios. Simulation results show that the proposed iterative algorithms outperform the existing protocol, which exhibits in the section 4.7.

5.2 Future Works

This section presents the open issues that deserve further research for each of the aspects carried out in this research work. There remain some limitations in our work that we seek to address in the future, such as; Link failure in finding next route in intersections can be improved. Advertisement and multimedia Messages transmission for broadcasting storms is not considered. Transmission of security related message passing. Energy Efficiency and Quality of service in route selection and data forwarding can also be a part of our future work.

Appendix A: CDS Unit disk graph model construction

Number of nodes	10, 25
Channel	Wireless channel
Propagation model	Two ray ground mobility model
Packet size	Uto140 bytes
Routing protocol	DSDV
Node speed	10ms maximum

Table A-1 CDS Construction Parameters.

A-1: TCL pseudo code for CDS Construction.

```
#======================================
#     Simulation parameters setup

set val(chan)    Channel/WirelessChannel     ;# channel type
set val(prop)    Propagation/TwoRayGround    ;# radio-propagation model
set val(netif)   Phy/WirelessPhy             ;# network interface type
set val(mac)     Mac/802_11                  ;# MAC type
set val(ifq)     Queue/DropTail/PriQueue     ;# interface queue type
set val(ll)      LL                          ;# link layer type
set val(ant)     Antenna/OmniAntenna         ;# antenna model
set val(ifqlen) 50                           ;# max packet in ifq
set val(nn)      10                          ;# number of mobilenodes
set val(rp)      DSDV                        ;# routing protocol
set val(x)       881                         ;# X dimension of topography
set val(y)       652                         ;# Y dimension of topography
set val(stop)    250.0                       ;# time of simulation end
#======================================
#        Initialization
#======================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out1.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out1.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel
```

```
#=====================================
#     Mobile node parameter setup
#=====================================
$ns node-config -adhocRouting   $val(rp) \
            -llType        $val(ll) \
            -macType        $val(mac) \
            -ifqType        $val(ifq) \
            -ifqLen        $val(ifqlen) \
            -antType        $val(ant) \
            -propType        $val(prop) \
            -phyType        $val(netif) \
            -channel        $chan \
            -topoInstance   $topo \
            -agentTrace     OFF \
            -routerTrace    ON \
            -macTrace       ON \
            -movementTrace ON

#=====================================
#       Nodes Definition
#=====================================
#Create 16 nodes
set n0 [$ns node]
$n0 set X_ 437
$n0 set Y_ 301
$n0 set Z_ 0.0
$n0 color orange
$ns at 0.0 "$n0 color orange"
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 324
$n1 set Y_ 337
$n1 set Z_ 0.0
$n1 color orange
$ns at 0.0 "$n1 color orange"
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 274
$n2 set Y_ 257
$n2 set Z_ 0.0
$n2 color orange
$ns at 0.0 "$n2 color orange"
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 291 $n3 set Y_ 456 $n3 set Z_ 0.0
$n3 color orange $ns at 0.0 "$n3 color orange

$ns initial_node_pos $n3 20
set n4 [$ns node]
```

```
$ns node-config -adhocRouting   $val(rp) \
$n4 set X_ 478   $n4 set Y_ 390 $n4 set Z_ 0.0
$n4 color orange $ns at 0.0 "$n4 color orange"
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 150   $n5 set Y_ 357   $n5 set Z_ 0.0 $n5 color orange
$ns at 0.0 "$n5 color orange"
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 436   $n6 set Y_ 153   $n6 set Z_ 0.0
$n6 color orange
$ns at 0.0 "$n6 color orange"
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 451   $n7 set Y_ 509   $n7 set Z_ 0.0
$n7 color orange
$ns at 0.0 "$n7 color orange"
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 150   $n8 set Y_ 166   $n8 set Z_ 0.0
$n8 color orange
$ns at 0.0 "$n8 color orange"
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 566   $n9 set Y_ 305   $n9 set Z_ 0.0
$n9 color orange
$ns at 0.0 "$n9 color orange"
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 611   $n10 set Y_ 462   $n10 set Z_ 0.0
$n10 color orange
$ns at 0.0 "$n10 color orange"
$ns initial_node_pos $n10 20


$ns at 0.5 "$n0 setdest 741.222 219.713 15.0"
$ns at 0.5 "$n1 setdest 661.911 293.578 15.0"
$ns at 0.5 "$n2 setdest 720.195 368.71 15.0"
$ns at 0.5 "$n3 setdest 841.832 311.792 15.0"
$ns at 0.5 "$n4 setdest 846.279 231.381 15.0"
$ns at 0.5 "$n5 setdest 154.962 76.1411 15.0"
$ns at 0.5 "$n6 setdest 277.55 146.965 15.0"
$ns at 0.5 "$n7 setdest 218.432 229.934 15.0"
$ns at 0.5 "$n8 setdest 58.4876 147.719 15.0"
$ns at 0.5 "$n9 setdest 113.938 245.951 15.0"
```



```
$ns at 70.0 "$n0 setdest 546.22 499.713 15.0"
```

```
$ns at 75.5 "$n4 setdest 720.195 368.71 15.0"
$ns at 75.5 "$n2 setdest 846.279 231.381 15.0"
$ns at 80.5 "$n9 setdest 58.4876 147.719 15.0"
$ns at 80.5 "$n8 setdest 113.938 245.951 15.0"
#$ns at 138.0 "$n10 setdest 656.26 181.573 15.0"
#$ns at 162.0 "$n10 add-mark N2 white circle"
#$n10 color darkgreen
#$ns at 162.0 "$n10 color darkgreen"
#$ns at 39.0 "$n1 label High-Power-Cluster"
#$ns at 39.0 "$n6 label High-Power-Cluster"
#$ns at 39.0 "$n10 label High-Power-Cluster"
 #$ns at 108.0 "$n10 label   ordinary node"
 #$ns at 99.1 "$n10 color   orange"
#$ns at 99.1 "$n14 color   darkgreen"
 #$ns at 118.0 "$n1 color   orange"
#$ns at 118.0 "$n0 color   darkgreen"
$ns at 64.1 "$n1 color darkgreen"
$n6 color darkgreen
$ns at 47.1 "$n6 color darkgreen"
$n10 color darkgreen
$ns at 56.1 "$n10 color darkgreen"
$n1 color darkgreen
$ns at 134.1 "$n1 color darkgreen"
$n6 color darkgreen
$ns at 140.1 "$n6 color darkgreen"
$n10 color darkgreen
$ns at 146.1 "$n10 color darkgreen"

$ns at 64.1 "$n1 add-mark N2 white circle"
$ns at 47.1 "$n6 add-mark N2 white circle"
$ns at 56.1 "$n10 add-mark N2 white circle"
$ns at 130.1 "$n1 add-mark N2 white circle"
$ns at 137.1 "$n6 add-mark N2 white circle"
$ns at 142.1 "$n10 add-mark N2 white circle"
$ns at 120.1 "$n10 color darkgreen"
$ns at 115.1 "$n1 add-mark N2 white circle"
$ns at 109.1 "$n6 add-mark N2 white circle"
$ns at 190.1 "$n10 add-mark N2 white circle"
$ns at 199.1 "$n1 add-mark N2 white circle"
$ns at 210.1 "$n6 add-mark N2 white circle"
$ns at 220.1 "$n10 add-mark N2 white circle"
#$ns at 99.1 "$n10 delete-mark N2"

#$ns at 118.0 "$n1 delete-mark N2"
#$ns at 118.0 "$n0 add-mark N2 white circle"
#$ns at 157.0 "$n0 delete-mark N2"
#$ns at 157.0 "$n0 add-mark N2 pink circle"
#$ns at 185.1 "$n10 delete-mark N2"
#$ns at 168.0 "$n1 delete-mark N2"
$ns at 70.0 "$n0 setdest 546.22 499.713 15.0"
```

```
 #$ns at 177.0 "$n0 delete-mark N2"
  #$ns at 177.0 "$n0 add-mark N2 pink circle"

proc create_cluster_head_node {

} {


global val ns_ node_ topo contador_nodos rng

Phy/WirelessPhy set Pt_ $val(pt_cluster_head)
$ns_ node-config -sensorNode ON \
-adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-energyModel $val(en) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
 -agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-rxPower 0.3 \
-txPower 0.6 \
-initialEnergy 100.0 \
-movementTrace OFF
set node_($contador_nodos) [$ns_ node]
$node_($contador_nodos) random-motion 0
set x [$rng uniform 0.0 $val(x)]
set y [$rng uniform 0.0 $val(y)]
$node_($contador_nodos) set X_ $x
$node_($contador_nodos) set Y_ $y
$node_($contador_nodos) set Z_ 0.0
set interval [$rng uniform 0.0 1.0]
Node/MobileNode/SensorNode set processingPower 0.36
Node/MobileNode/SensorNode set instructionsPerSecond_ 150000000
Phy/WirelessPhy set   bandwidth_ 1000000.0

set udp_($contador_nodos) [new Agent/UDP]

set app_($contador_nodos) [create_cluster_head_app [$node_(1) node-addr]
$val(disseminating_type) $val(cluster_head_disseminating_interval)]
$node_($contador_nodos) attach $udp_($contador_nodos) $val(port)
$node_($contador_nodos) add-app $app_($contador_nodos)
set processing_($contador_nodos) [new Processing/AggregateProcessing]
```

```
$app_($contador_nodos) node $node_($contador_nodos)
$app_($contador_nodos) attach-agent $udp_($contador_nodos)

$app_($contador_nodos) attach-processing $processing_($contador_nodos)
$processing_($contador_nodos) node $node_($contador_nodos)

$ns_ at [expr $val(start) + 1 + $interval] "$app_($contador_nodos) start"
$ns_ at $val(stop) "$app_($contador_nodos) stop"

incr contador_nodos
 }

set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns   attach-agent $n0 $tcp
$ns attach-agent $n1 $sink
$ns   connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
#$ftp set rate_ 1.0Mb
#$ftp set random_ null
#$ftp   set interval_ 0.4
$ns   at 64.2 "$ftp start"
$ns   at 68.0 "$ftp stop"

set tcp1 [new Agent/TCP]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
$ns   attach-agent $n2 $tcp1
$ns attach-agent $n1 $sink1
$ns   connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
#$ftp1 set rate_ 1.0Mb
#$ftp1 set random_ null
#$ftp1   set interval_ 0.4
$ns   at 68.2 "$ftp1 start"
$ns   at 72.0 "$ftp1 stop"

set tcp2 [new Agent/TCP]
$tcp2 set class_ 2
set sink2 [new Agent/TCPSink]
$ns   attach-agent $n3 $tcp2
$ns attach-agent $n1 $sink2
$ns   connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
#$ftp2 set rate_ 1.0Mb
```

```
#$ftp2   set interval_ 0.4
$ns   at 72.2 "$ftp2 start"
$ns   at 76.0 "$ftp2 stop"

set tcp3 [new Agent/TCP]
$tcp3 set class_ 2
set sink3 [new Agent/TCPSink]
$ns   attach-agent $n4 $tcp3
$ns attach-agent $n1 $sink3
$ns   connect $tcp3 $sink3
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3
#$ftp3 set rate_ 1.0Mb
#$ftp3 set random_ null
#$ftp3   set interval_ 0.4
$ns   at 76.2 "$ftp3 start"
$ns   at 80.0 "$ftp3 stop"

set tcp4 [new Agent/TCP]
$tcp4 set class_ 2
set sink4 [new Agent/TCPSink]
$ns   attach-agent $n1 $tcp4
$ns attach-agent $n15 $sink4
$ns   connect $tcp4 $sink4
set ftp4 [new Application/FTP]
$ftp4 attach-agent $tcp4
#$ftp4 set rate_ 1.0Mb
#$ftp4 set random_ null
#$ftp4   set interval_ 0.4
$ns   at 80.2 "$ftp4 start"
$ns   at 84.0 "$ftp4 stop"

set tcp5 [new Agent/TCP]
$tcp5 set class_ 2
set sink5 [new Agent/TCPSink]
$ns   attach-agent $n5 $tcp5
$ns attach-agent $n6 $sink5
$ns   connect $tcp5 $sink5
set ftp5 [new Application/FTP]
$ftp5 attach-agent $tcp5
#$ftp5 set rate_ 1.0Mb
#$ftp5 set random_ null
#$ftp5   set interval_ 0.4
$ns   at 84.2 "$ftp5 start"
$ns   at 88.0 "$ftp5 stop"

set tcp6 [new Agent/TCP]
$tcp6 set class_ 2
set sink6 [new Agent/TCPSink]
```

```
$ns attach-agent $n6 $sink6
$ns   connect $tcp6 $sink6
set ftp6 [new Application/FTP]
$ftp6 attach-agent $tcp6
#$ftp6 set rate_ 1.0Mb
#$ftp6 set random_ null
#$ftp6   set interval_ 0.4
$ns   at 88.2 "$ftp6 start"
$ns   at 92.0 "$ftp6 stop"

set tcp7 [new Agent/TCP]
$tcp7 set class_ 2
set sink7 [new Agent/TCPSink]
$ns   attach-agent $n8 $tcp7
$ns attach-agent $n6 $sink7
$ns   connect $tcp7 $sink7
set ftp7 [new Application/FTP]
$ftp7 attach-agent $tcp7
#$ftp7 set rate_ 1.0Mb
#$ftp7 set random_ null
#$ftp7   set interval_ 0.4
$ns   at 92.2 "$ftp7 start"
$ns   at 96.0 "$ftp7 stop"

set tcp8 [new Agent/TCP]
$tcp8 set class_ 2
set sink8 [new Agent/TCPSink]
$ns   attach-agent $n9 $tcp8
$ns attach-agent $n6 $sink8
$ns   connect $tcp8 $sink8
set ftp8 [new Application/FTP]
$ftp8 attach-agent $tcp8
#$ftp8 set rate_ 1.0Mb
#$ftp8 set random_ null
#$ftp8   set interval_ 0.4
$ns   at 96.2 "$ftp8 start"
$ns   at 100.0 "$ftp8 stop"

set tcp9 [new Agent/TCP]
$tcp9 set class_ 2
set sink9 [new Agent/TCPSink]
$ns   attach-agent $n6 $tcp9
$ns attach-agent $n15 $sink9
$ns   connect $tcp9 $sink9
set ftp9 [new Application/FTP]
$ftp9 attach-agent $tcp9
#$ftp9 set rate_ 1.0Mb
#$ftp9 set random_ null
#$ftp9   set interval_ 0.4
```

```
$ns   at 104.0 "$ftp9 stop"

set tcp10 [new Agent/TCP]
$tcp10 set class_ 2
set sink10 [new Agent/TCPSink]
$ns   attach-agent $n11 $tcp10
$ns attach-agent $n10 $sink10
$ns   connect $tcp10 $sink10
set ftp10 [new Application/FTP]
$ftp10 attach-agent $tcp10
#$ftp10 set rate_ 1.0Mb
#$ftp10 set random_ null
#$ftp10   set interval_ 0.4
$ns   at 104.2 "$ftp10 start"
$ns   at 108.0 "$ftp10 stop"

set tcp11 [new Agent/TCP]
$tcp11 set class_ 2
set sink11 [new Agent/TCPSink]
$ns   attach-agent $n12 $tcp11
$ns attach-agent $n10 $sink11
$ns   connect $tcp11 $sink11
set ftp11 [new Application/FTP]
$ftp11 attach-agent $tcp11
#$ftp11 set rate_ 1.0Mb
#$ftp11 set random_ null
#$ftp11   set interval_ 0.4
$ns   at 108.2 "$ftp11 start"
$ns   at 112.0 "$ftp11 stop"


#Define a 'finish' procedure
proc finish {

} {

    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out1.nam &
    exit 0

}
for {
set i 0
} {
$i < $val(nn)
} {
```

```
} {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

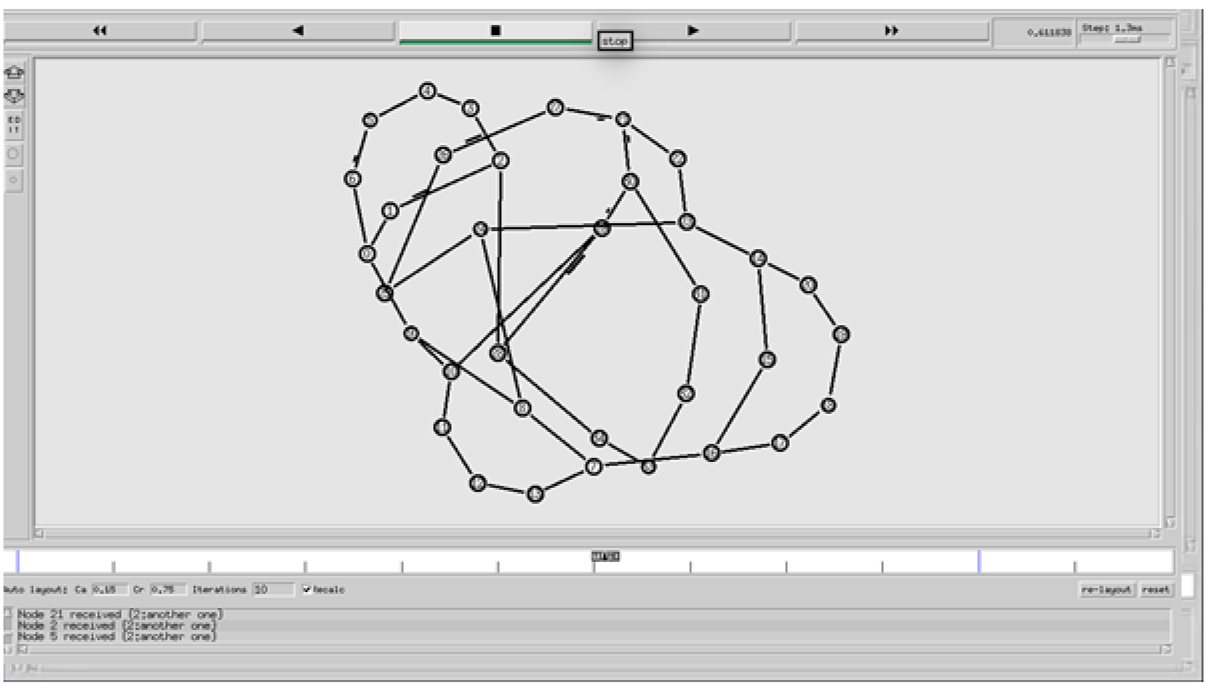


Figure A-1: Unit graph construction.

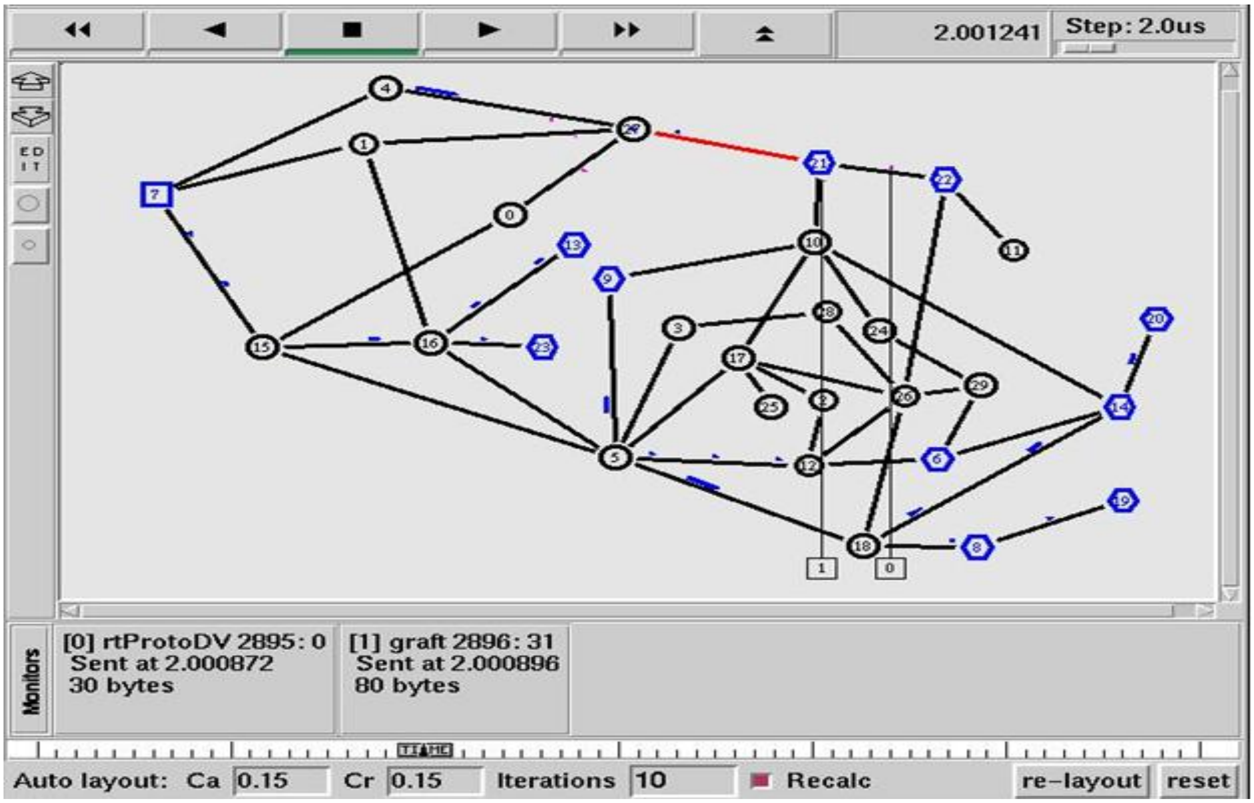


Figure A-2: Construction of CDS with unit disk graph model (UDG).

Summary

Unit disk graph model is constructed to implement the connected dominating sets for data dissemination. The no of nodes size taken here is 10-25, with random way form mobility models. The node movements and data transfers are exhibited with different colors. The trace file exhibits the connected dominating sets with mobility speed around 10ms for different mobility scenarios were considered.

Appendix B: Addressing the Broadcasting storm issues

Number of nodes	10, 25
Channel	Wireless channel
Propagation model	Two ray ground mobility model
Packet size	Upto140 bytes
Routing protocol	AODV,SRD , AID
Node speed	10ms maximum

Table B-1: Illustration of Broadcasting Storm.

```
set val(chan)      Channel/WirelessChannel  ;# channel type
set val(prop)      Propagation/TwoRayGround ;# radio-propagation model
set val(ant)       Antenna/OmniAntenna      ;# Antenna type
set val(ll)        LL                       ;# Link layer type
set val(ifq)       Queue/DropTail/PriQueue  ;# Interface queue type
set val(ifqlen)    50                       ;# max packet in ifq
set val(netif)     Phy/WirelessPhy          ;# network interface type
set val(mac)       Mac/802_11               ;# MAC type
set val(nn)        6                        ;# number of mobilenodes
set val(rp)        AODV                     ;# routing protocol
set val(x)         800
set val(y)         800

set ns [new Simulator]
#ns-random 0
set ns [new Simulator]
#ns-random 0
set f [open 1_out.tr w]
$ns trace-all $f
set namtrace [open 1_out.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
set f0 [open packets_received.tr w]
set f1 [open packets_lost.tr w]
set f2 [open proj_out2.tr w]
set f3 [open proj_out3.tr w]

set topo [new Topography]
$topo load_flatgrid 800 800
create-god $val(nn)
```

```
set chan_1 [new $val(chan)]
set chan_2 [new $val(chan)]
set chan_3 [new $val(chan)]
set chan_4 [new $val(chan)]
set chan_5 [new $val(chan)]
set chan_6 [new $val(chan)]

#CONFIGURE AND CREATE NODES

$ns node-config  -adhocRouting $val(rp) \
        -llType $val(ll) \
             -macType $val(mac) \
             -ifqType $val(ifq) \
             -ifqLen $val(ifqlen) \
             -antType $val(ant) \
             -propType $val(prop) \
             -phyType $val(netif) \
             #-channelType $val(chan) \
             -topoInstance $topo \
             -agentTrace OFF \
             -routerTrace ON \
             -macTrace ON \
             -movementTrace OFF \

             -channel $chan_1  # \
             #-channel $chan_2   \
             #-channel $chan_3   \
             #-channel $chan_4   \
             #-channel $chan_5   \
             #-channel $chan_6
```

```
proc finish {}
 {
    global ns f f0 f1 f2 f3 namtrace
    Sns flush-trace
    close Snamtrace
    close Sf0
    close Sf1
     close Sf2
     close Sf3
      exec xgraph packets_received.tr packets_lost.tr
                                # proj_out2.tr proj_out3.tr
      xec nam -r 5m 1_out.nam &
    exit 0
}
proc record {}
{
  global sink0 sink1 sink2 sink3 sink4 sink5 f0 f1 f2 f3
   #Get An Instance Of The Simulator
   set ns [Simulator instance]
    #Set The Time After Which The Procedure Should Be Called Again
   set time 0.05
   #How Many Bytes Have Been Received By The Traffic Sinks?
   set bw0 [Ssink5 set npkts_]
   set bw1 [Ssink5 set nlost_]
   #set bw2 [Ssink2 set npkts_]
   #set bw3 [Ssink3 set npkts_]
    #Get The Current Time
   set now [Sns now]
   #Save Data To The Files
   puts Sf0 "Snow [expr Sbw0]"
```

```
  #Re-Schedule The Procedure
  $ns at [expr $now+$time] "record"
 }
# define color index
$ns color 0 blue
$ns color 1 red
$ns color 2 chocolate
$ns color 3 red
$ns color 4 brown
$ns color 5 tan
$ns color 6 gold
$ns color 7 black
set n(0) [$ns node]
#$ns at 0.0 "$n(0) color red"
$n(0) color "0"
$n(0) shape "circle"

set n(1) [$ns node]
$n(1) color "blue"
$n(1) shape "circle"
set n(2) [$ns node]
$n(2) color "tan"
$n(2) shape "circle"
set n(3) [$ns node]

$n(3) color "red"
$n(3) shape "circle"
set n(4) [$ns node]
$n(4) color "tan"
$n(4) shape "circle"
```

```
set n(5) [$ns node]
$n(5) color "red"
$n(5) shape "circle"

for {set i 0} {$i < $val(nn)} {incr i} {
   $ns initial_node_pos $n($i) 30+i*100
}
$n(0) set X_ 0.0
$n(0) set Y_ 0.0
$n(0) set Z_ 0.0
$n(1) set X_ 0.0
$n(1) set Y_ 0.0
$n(1) set Z_ 0.0
$n(2) set X_ 0.0
$n(2) set Y_ 0.0
$n(2) set Z_ 0.0
$n(3) set X_ 0.0
$n(3) set Y_ 0.0
$n(3) set Z_ 0.0
$n(4) set X_ 0.0
$n(4) set Y_ 0.0
$n(4) set Z_ 0.0
$n(5) set X_ 0.0
$n(5) set Y_ 0.0
$n(5) set Z_ 0.0
$ns at 0.0 "$n(0) setdest 100.0 100.0 3000.0"
$ns at 0.0 "$n(1) setdest 200.0 200.0 3000.0"
$ns at 0.0 "$n(2) setdest 300.0 200.0 3000.0"
$ns at 0.0 "$n(3) setdest 400.0 300.0 3000.0"
```

```
$ns at 0.0 "$n(4) setdest 500.0 300.0 3000.0"
$ns at 0.0 "$n(5) setdest 600.0 400.0 3000.0"

$ns at 2.0 "$n(5) setdest 100.0 400.0 500.0"
#$ns at 1.5 "$n(3) setdest 450.0 150.0 500.0"


# CONFIGURE AND SET UP A FLOW


set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]
set sink3 [new Agent/LossMonitor]
set sink4 [new Agent/LossMonitor]
set sink5 [new Agent/LossMonitor]
$ns attach-agent $n(0) $sink0
$ns attach-agent $n(1) $sink1
$ns attach-agent $n(2) $sink2
$ns attach-agent $n(3) $sink3
$ns attach-agent $n(4) $sink4
$ns attach-agent $n(5) $sink5
#$ns attach-agent $sink2 $sink3
set tcp0 [new Agent/TCP]
$ns attach-agent $n(0) $tcp0
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
set tcp2 [new Agent/TCP]
$ns attach-agent $n(2) $tcp2
set tcp3 [new Agent/TCP]
```

```
$ns attach-agent $n(3) $tcp3
set tcp4 [new Agent/TCP]
$ns attach-agent $n(4) $tcp4
set tcp5 [new Agent/TCP]
$ns attach-agent $n(5) $tcp5
proc attach-CBR-traffic { node sink size interval }
 { set ns [Simulator instance]
  set cbr [new Agent/CBR]
  $ns attach-agent $node $cbr
  $cbr set packetSize_ $size
  $cbr set interval_ $interval
  $ns connect $cbr $sink
  return $cbr
  }
set cbr0 [attach-CBR-traffic $n(0) $sink5 1000 .015]
#set cbr1 [attach-CBR-traffic $n(1) $sink2 1000 .015]
#set cbr2 [attach-CBR-traffic $n(2) $sink3 1000 .015]
#set cbr3 [attach-CBR-traffic $n(3) $sink0 1000 .015]
#set cbr4 [attach-CBR-traffic $n(4) $sink3 1000 .015]
#set cbr5 [attach-CBR-traffic $n(5) $sink0 1000 .015]
$ns at 0.0 "record"
#$ns at 0.5 "$cbr0 start"
#$ns at 0.5 "$cbr2 start"
#$ns at 2.0 "$cbr0 stop"
#$ns at 2.0 "$cbr2 stop"
$ns at 1.0 "$cbr0 start"
#$ns at 4.0 "$cbr3 stop"
$ns at 10.0 "finish"
puts "Start of simulation.."
$ns run
```

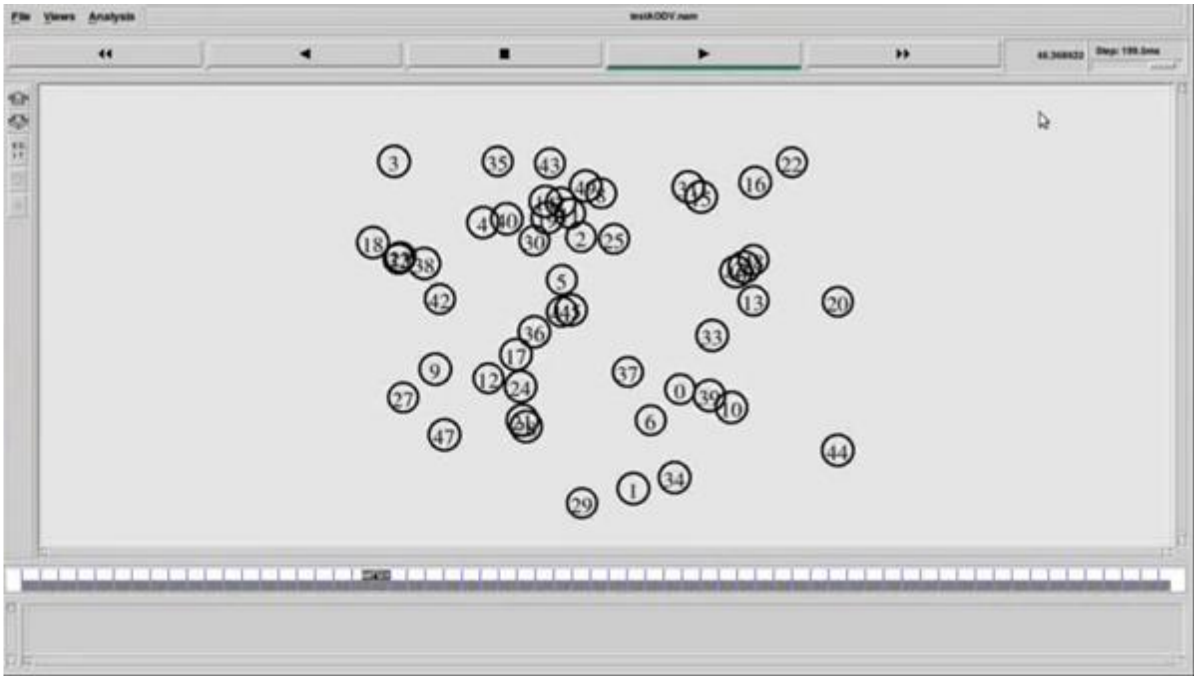


Figure B-1: Node movement

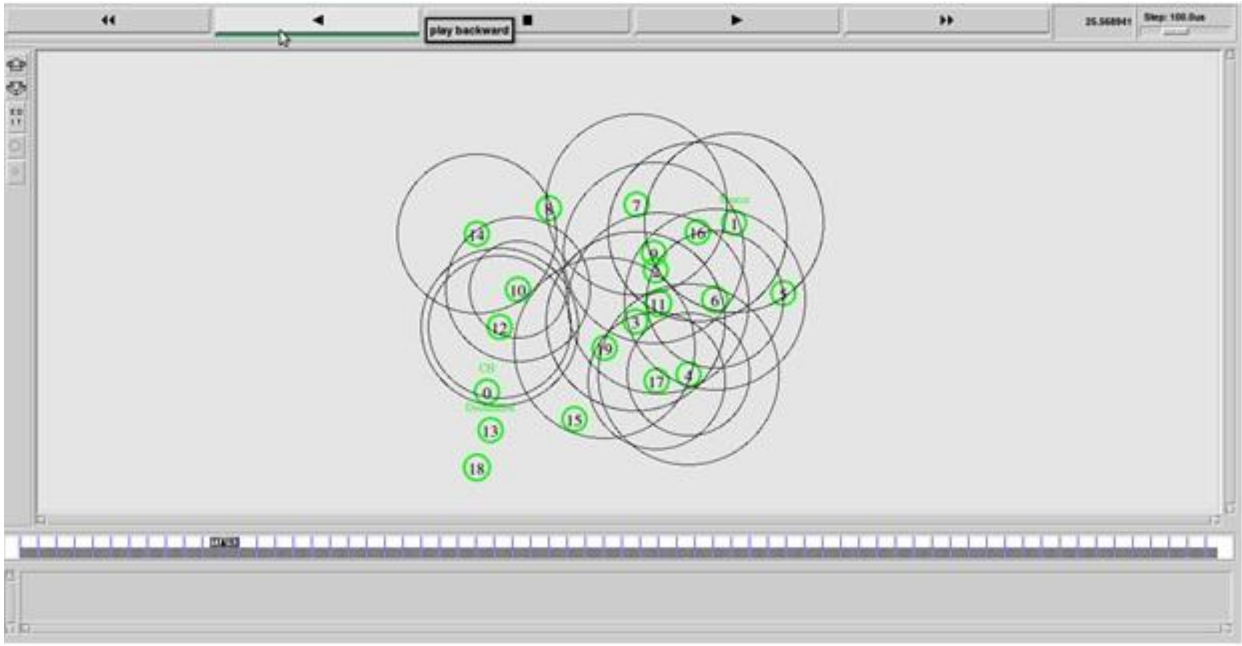


Figure B-2: Data Dissemination

Summary

This sample code implements the simple flooding and addressing the broadcasting storm issues. The configuration of the implementation been, Two ray ground mobility model, Packet size -Upto140 bytes, protocol for implementation - AODV,SRD , AID; Node speed- 10ms maximum. In the first phase, simple node movements are simulated for the given conditions. In the next phase data dissemination for broadcasting storm issue been simulated for various mobility scenarios. The sample resultant graph exhibits data dissemination with 20 nodes.

Appendix C: Comparison with existing protocols

AODV-DSDV-BDP Protocol comparison

Number of nodes	50
Channel	Wireless channel
Propagation model	Two ray ground mobility model
CBR	50
Routing protocol	DSDV, AODV,BDP
Node speed	10ms maximum

Table C-1: AODV-DSDV-BDP Protocol comparison Parameters

```
if {$argc !=3}
{
        puts "Usage: ns adhoc.tcl  Routing_Protocol Traffic_Pattern
Scene_Pattern "
        puts "Example:ns adhoc.tcl DSDV cbr-50-10-8 scene-50-0-20"
        exit
}

set par1 [lindex $argv 0]
set par2 [lindex $argv 1]
set par3 [lindex $argv 2]


set val(chan)          Channel/WirelessChannel    ;# channel type
set val(prop)          Propagation/TwoRayGround   ;# radio-propagationmodel
set val(netif)         Phy/WirelessPhy          ;# network interface type
set val(mac)           Mac/802_11               ;# MAC type
```

```
if { $par1=="BDP"}
{
 set val(ifq)          CMUPriQueue
}
 else {
 set val(ifq)         Queue/DropTail/PriQueue    ;# interface queue type
}
set val(ll)           LL ;# link layer type
set val(ant)          Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)       50                    ;# max packet in ifq
set val(rp)           $par1                   ;# routing protocol
set val(x)            500
set val(y)            500
set val(seed)         0.0
set val(tr)           temp.tr
set val(nn)           50
set val(cp)           $par2
set val(sc)           $par3
set val(stop)         100.0

set ns_           [new Simulator]

set tracefd     [open $val(tr) w]
$ns_ trace-all $tracefd
$ns_ use-newtrace
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
```

```
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -channel $chan_1_ \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace OFF

    for {set i 0} {$i < $val(nn) } {incr i} {
        set node_($i) [$ns_ node]
        $node_($i) random-motion 0          ;# disable random motion
    }

puts "Loading connection pattern..."
source $val(cp)

puts "Loading scenario file..."
source $val(sc)

for {set i 0} {$i < $val(nn) } {incr i} {
  $ns_ initial_node_pos $node_($i) 20
}

for {set i 0} {$i < $val(nn) } {incr i}
```

```
{
    $ns_ at $val (stop).00000001 "$node_ ($i) reset";
}

$ns_ at $val(stop).000000001 "puts \"NS EXITING...\"; $ns_ halt"
puts " Simulation Start ...   NS_2"
$ns_ run
```

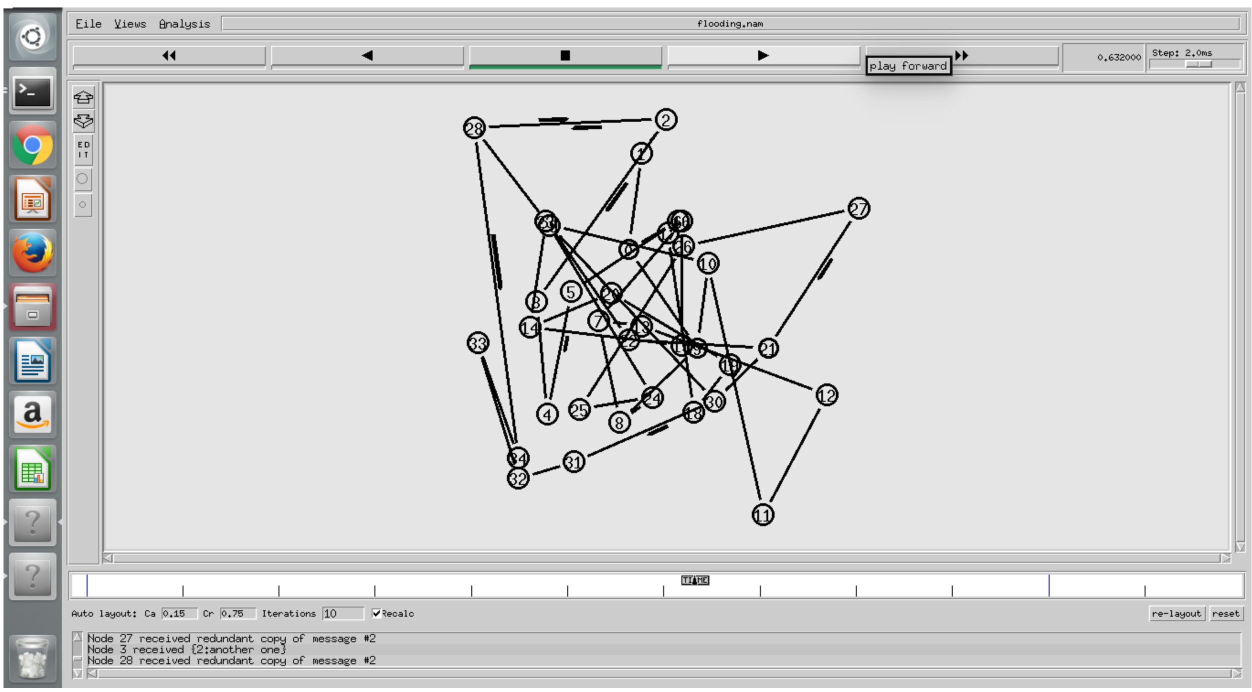


Figure C-1: Packet dropping in flooding scenario

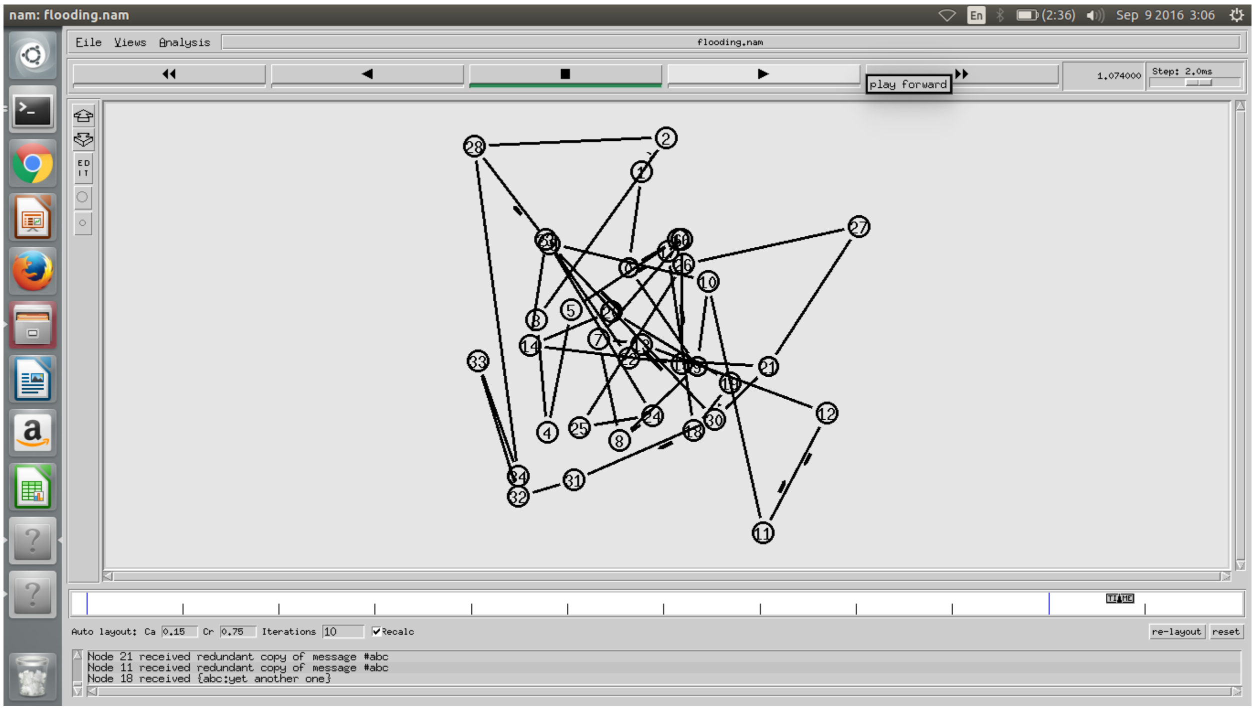


Figure C-2: Packet dropping Calculation in various paths

```
/********End to End Delay, Routing Load and Packet Density Fraction ********/
/**************************************************************************/
BEGIN {
      sends=0;
      recvs=0;
      routing_packets=0.0;
      droppedBytes=0;
      droppedPackets=0;
      highest_packet_id =0;
      sum=0;
      recvnum=0;
   }
   { time = $3;
 packet_id = $41;
 # CALCULATE PACKET DELIVERY FRACTION
  if (( $1 == "s") &&  ( $35 == "cbr" ) && ( $19=="AGT" ))
 {  sends++; }

  if (( $1 == "r") &&  ( $35 == "cbr" ) && ( $19=="AGT" ))
  { recvs++; }

  # CALCULATE DELAY
  if ( start_time[packet_id] == 0 )
 start_time[packet_id] = time;
  if (( $1 == "r") &&  ( $35 == "cbr" ) && ( $19=="AGT" ))
 {  end_time[packet_id] = time;
 }

else
{  end_time[packet_id] = -1;  }
```

```
 # CALCULATE TOTAL AODV OVERHEAD


 if (($1 == "s" || $1 == "f") && $19 == "RTR" && ($35 =="AODV" ||$35
=="AOMDV"))
routing_packets++;

 # DROPPED AODV PACKETS
 if (( $1 == "d" ) && ( $35 == "cbr" )  && ( $3 > 0 ))
     {
          droppedBytes=droppedBytes+$37;
          droppedPackets=droppedPackets+1;
     }

     #find the number of packets in the
simulation
        if (packet_id > highest_packet_id)
          highest_packet_id = packet_id;
 }
 END {
 for ( i in end_time )
 {
 start = start_time[i];
 end = end_time[i];
 packet_duration = end - start;
 if ( packet_duration > 0 )
 {    sum += packet_duration;
     recvnum++;
 }
 }
```

```
    delay=sum/recvnum;
    NRL = routing_packets/recvs;  #normalized
routing load
    PDF = (recvs/sends)*100;  #packet delivery
ratio[fraction]
    printf("Send Packets = %.2f\n",sends);
    printf("Received Packets = %.2f\n",recvs);
    printf("Roting Packets = %.2f\n",routing_packets++);
    printf("Packet Delivery Function = %.2f\n",PDF);
    printf("Normalised Routing Load = %.2f\n",NRL);
    printf("Average end to end delay(ms)= %.2f\n",delay*1000);
    printf("No. of dropped data (packets) = %d\n",droppedPackets);
    printf("No. of dropped data (bytes)  = %d\n",droppedBytes);
  }
```

Summary

In this section broadcasting protocol been compared with the existing protocol. To evaluate the protocol, end-end delay, packet delivery ratio and data overhead are taken as performance metrics. The sample output exhibits the packet dropping in flooding scenario. The section 4.5- 4.7 discussed the details of theses metrics and comparisons with existing protocols.

Bibliography

[1]. Panichpapiboon, S. and W. Pattara-Atikom, *A review of information dissemination protocols for vehicular ad hoc networks.* Communications Surveys & Tutorials, IEEE, 2012. **14**(3): p. 784-798.

[2]. Parno, B. and A. Perrig. *Challenges in securing vehicular networks*. in *Workshop on hot topics in networks (HotNets-IV)*. 2005.

[3]. Faezipour, M., et al., *Progress and challenges in intelligent vehicle area networks.* Communications of the ACM, 2012. **55**(2): p. 90-100.

[4]. Ji, Y.-b., Q.-h. Tian, and C. Zhang, *Design and Implementation of GPS in Vehicle Location and Monitor.* Journal of Petrochemical Universities, 2002. **15**(3): p. 61-63.

[5]. Zeadally, S., et al., *Vehicular ad hoc networks (VANETS): status, results, and challenges.* Telecommunication Systems, 2012. **50**(4): p. 217-241.

[6]. Toor, Y., P. Muhlethaler, and A. Laouiti, *Vehicle ad hoc networks: Applications and related technical issues.* Communications Surveys & Tutorials, IEEE, 2008. **10**(3): p. 74-88.

[7]. Miller, J. *Vehicle-to-vehicle-to-infrastructure (V2V2I) intelligent transportation system architecture*. in *Intelligent Vehicles Symposium, 2008 IEEE*. 2008. IEEE.

[8]. Al-Sultan, S., et al., *A comprehensive survey on vehicular Ad Hoc network.* Journal of network and computer applications, 2014. **37**: p. 380-392.

[9]. Hartenstein, H. and L. Laberteaux, *A tutorial survey on vehicular ad hoc networks.* IEEE Communications magazine, 2008. **46**(6).

[10]. van Eenennaam, M., et al. *Exploring the solution space of beaconing in VANETs*. in *Vehicular Networking Conference (VNC), 2009 IEEE*. 2009. IEEE.

[11]. Gerasenko, S., et al., *Beacon signals: what, why, how, and where?* Computer, 2001. 34(10): p. 108-110.

[12]. Li, F. and Y. Wang, *Routing in vehicular ad hoc networks: A survey.* Vehicular Technology Magazine, IEEE, 2007. **2**(2): p. 12-22.

[13]. Yoon, J., M. Liu, and B. Noble. *Random waypoint considered harmful*. in *INFOCOM 2003. twenty-second annual joint conference of the IEEE computer and communications. IEEE societies*. 2003. IEEE.

[14]. Martinez, F.J., et al., *A survey and comparative study of simulators for vehicular ad hoc networks (VANETs).* Wireless Communications and Mobile Computing, 2011. **11**(7): p. 813-828.

[15]. Tonguz, O.K., N. Wisitpongphan, and F. Bai, *DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks.* Wireless Communications, IEEE, 2010. **17**(2): p. 47-57.

[16]. Lee, K.C., U. Lee, and M. Gerla, *Survey of routing protocols in vehicular ad hoc networks.* Advances in vehicular ad-hoc networks: Developments and challenges, 2010: p. 149-170.

[17]. Lee, K.C., P.-C. Cheng, and M. Gerla, *GeoCross: A geographic routing protocol in the presence of loops in urban scenarios.* Ad Hoc Networks, 2010. **8**(5): p. 474-488.

[18]. Tseng, Y.-C., et al., *The broadcast storm problem in a mobile ad hoc network.* Wireless networks, 2002. **8**(2-3): p. 153-167.

[19]. Tseng, Y.-C., S.-Y. Ni, and E.-Y. Shih, *Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network.* IEEE transactions on computers, 2003. **52**(5): p. 545-557.

[20]. Lloyd, E.L., *Broadcast scheduling for TDMA in wireless multihop networks.* Handbook of wireless networks and mobile computing, 2002: p. 347-370.

[21]. Qayyum, A., L. Viennot, and A. Laouiti. *Multipoint relaying for flooding broadcast messages in mobile wireless networks*. in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*. 2002. IEEE.

[22]. Pagani, E. and G.P. Rossi. *Reliable broadcast in mobile multihop packet networks*. in *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*. 1997. ACM.

[23]. Lauer, G. *Address servers in hierarchical networks*. in *Communications, 1988. ICC'88. Digital Technology-Spanning the Universe. Conference Record., IEEE International Conference on*. 1988. IEEE.

[24]. Lin, C.R. and M. Gerla, *Adaptive clustering for mobile wireless networks.* IEEE Journal on Selected areas in Communications, 1997. **15**(7): p. 1265-1275.

[25]. Briesemeister, L. and G. Hommel. *Role-based multicast in highly mobile but sparsely connected ad hoc networks*. in *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. 2000. IEEE Press.

[26]. Sun, M.-T., et al. *GPS-based message broadcast for adaptive inter-vehicle communications*. in *Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd*. 2000. IEEE.

[27]. Chen, W.-T. and N.-F. Huang, *The strongly connecting problem on multihop packet radio networks.* IEEE Transactions on Communications, 1989. **37**(3): p. 293-295.

[28]. Wieselthier, J.E., G.D. Nguyen, and A. Ephremides. *On the construction of energy-efficient broadcast and multicast trees in wireless networks*. in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. 2000. IEEE.

[29]. Stojmenovic, I., *Handbook of wireless networks and mobile computing*. Vol. 27. 2003: John Wiley & Sons.

[30]. Wu, J. and H. Li. *On calculating connected dominating set for efficient routing in ad hoc wireless networks*. in *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*. 1999. ACM.

[31]. Guha, S. and S. Khuller, *Approximation algorithms for connected dominating sets.* Algorithmica, 1998. **20**(4): p. 374-387.

[32]. Wan, P.-J., K.M. Alzoubi, and O. Frieder. *Distributed construction of connected dominating set in wireless ad hoc networks*. in *INFOCOM 2002. Twenty-First annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*. 2002. IEEE.

[33]. Alzoubi, K., et al., *Geometric spanners for wireless ad hoc networks.* IEEE Transactions on Parallel and Distributed Systems, 2003. **14**(4): p. 408-421.

[34]. Zhou, D., M.-T. Sun, and T.-H. Lai. *A timer-based protocol for connected dominating set construction in ieee 802.11 multihop mobile ad hoc networks*. in *The 2005 Symposium on Applications and the Internet*. 2005. IEEE.

[35]. Sakai, K., M.-T. Sun, and W.-S. Ku. *Fast connected dominating set construction in mobile ad hoc networks*. in *2009 IEEE International Conference on Communications*. 2009. IEEE.

[36]. Chen, Y.P. and A.L. Liestman. *Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks*. in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. 2002. ACM.

[37]. Bao, L. and J.J. Garcia-Luna-Aceves. *Topology management in ad hoc networks*. in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. 2003. ACM.

[38]. Das, B. and V. Bharghavan. *Routing in ad-hoc networks using minimum connected dominating sets*. in *Communications, 1997. ICC'97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*. 1997. IEEE.

[39]. Tonguz, O., et al. *Broadcasting in VANET*. in *2007 mobile networking for vehicular environments*. 2007. IEEE.

[40]. Vegni, A.M., A. Stramacci, and E. Natalizio. *Opportunistic clusters selection in a reliable enhanced broadcast protocol for vehicular ad hoc networks*. in *Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on*. 2013. IEEE.

[41]. Wisitpongphan, N., et al., *Broadcast storm mitigation techniques in vehicular ad hoc networks.* Wireless Communications, IEEE, 2007. **14**(6): p. 84-94.

[42]. Seet, B.-C., et al. *A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications*. in *Networking 2004*. 2004. Springer.

[43]. Karp, B. and H.-T. Kung. *GPSR: Greedy perimeter stateless routing for wireless networks*. in *Proceedings of the 6th annual international conference on Mobile computing and networking*. 2000. ACM.

[44]. Kim, Y.-J., et al. *Geographic routing made practical*. in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. 2005. USENIX Association.

[45]. Govindan, Y.-J.K.R., B. Karp, and S. Shenker. *Lazy cross-link removal for geographic routing*. in *Proceedings of the 4th international conference on Embedded networked sensor systems*. 2006. ACM.

[46]. Fomin, F.V., F. Grandoni, and D. Kratsch. *Measure and conquer: domination–a case study*. in *International Colloquium on Automata, Languages, and Programming*. 2005. Springer.

[47]. Korkmaz, G., et al. *Urban multi-hop broadcast protocol for inter-vehicle communication systems*. in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. 2004. ACM.

[48]. Biswas, S., R. Tatchikou, and F. Dion, *Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety.* Communications Magazine, IEEE, 2006. **44**(1): p. 74-82.

[49]. Korkmaz, G., E. Ekici, and F. Ozguner. *An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems*. in *Communications, 2006. ICC'06. IEEE International Conference on*. 2006. IEEE.

[50]. Schwartz, R.S., et al. *A simple and robust dissemination protocol for VANETs*. in *Wireless Conference (EW), 2010 European*. 2010. IEEE.

[51]. Bakhouya, M., J. Gaber, and P. Lorenz, *An adaptive approach for information dissemination in Vehicular Ad hoc Networks.* Journal of Network and Computer Applications, 2011. **34**(6): p. 1971-1978.