FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Adnan Afsar Khan
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Parameterless Broadcasting in Static to Highly Mobile Wireless ad hoc, Sensor and Actuator Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Nejib Zaguia
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

N. Santoni

I. Stojmenovic

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Parameterless broadcasting in static to highly mobile wireless ad hoc, sensor and actuator networks

by

Adnan Afsar Khan

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Information Technology and Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-32457-8
Our file Notre référence
ISBN: 978-0-494-32457-8

Abstract

In a broadcasting task, source node wants to send the same message to all the other nodes in the network. Existing solutions range from connected dominating set (CDS) based for static networks, to blind flooding for moderate mobility within connected network, to hyperflooding for highly mobile and frequently partitioned networks. The only existing protocol for all scenarios is based on some threshold parameters to locally select between these three solution approaches. Here we propose a different protocol, which adjusts itself to any mobility scenario without using any parameter. Unlike existing methods for highly mobile scenarios, in proposed method, two nodes do not transmit every time they discover each other as new neighbors. Each node maintains a list of two hop neighbors by periodically exchanging 'hello' messages, and decides whether or not it is in CDS. Upon receipt of the first copy of message intended for broadcasting, it selects a waiting timeout and constructs two lists of neighbors: neighbors that received the same message and neighbors that did not receive it. Nodes not in CDS select longer timeouts than nodes in CDS. These lists are updated upon receipt of further copies of same packet. When timeout expires, node retransmits if the list of neighbors in need of message is non-empty. 'Hello' messages received while waiting, or after timeout expiration may revise all lists (and CDS status) and consequently the need to retransmit. This provides a seamless transition of protocol behavior from static to highly mobile scenarios. Our protocol is compared to existing solutions for all scenarios. It was shown to be superior to all of them in number of retransmissions and reliability.

Acknowledgements

My sincere gratitude goes to my supervisor Professor Nejib Zaguia, who both encouraged and funded my research. I would also like to thank Professor Ivan Stojmenovic deeply for his invaluable advice, encouragement, and guidance throughout my research activities. This thesis could not have been successful without their help.

Furthermore, I would like to acknowledge the University of Ottawa for providing me with the opportunity to undertake my Master's degree with financial assistance.

Finally, I would like to thank my parents for financing me and encouraging me to pursue higher studies at the University of Ottawa.

Table of Contents

List of Tables

List of Figures

List of Acronyms

BFS:	Breadth First Search
CDS:	Connected Dominating Set
CSMA / CA:	Carrier Sense Multiple Access with Collision Avoidance
DS:	Dominating Set
FIFO:	First In First Out
GPS:	Global Positioning System
MAC:	Medium ACcess
MANET:	Mobile Ad hoc wireless NETwork
N:	list of node containing Neighbors in need of message
NES:	Neighbor Elimination Scheme
PBSM:	Parameterless Broadcasting from Static to Mobile networks
R:	list of node containing other nodes that has Received message
TTL:	Time for data To Live
WSAN:	Wireless Sensor-Actuator Network
WSN:	Wireless Sensor Network

Chapter 1

Introduction

1.1 Background

A **mobile ad hoc wireless network (MANET)** is a collection of wireless mobile hosts that form a temporary network without any fixed infrastructure or centralized administration. The applications of ad hoc wireless networks range from military use in battlefields to interactive conferences. A **wireless sensor network (WSN)** consists of sensors that are spatially distributed over a geographical area. The sensors are normally static, but mobile sensors may also be envisioned. The sensors monitor the environment and report events (e.g. a high temperature or intrusion) to a sink, which is connected to a central server. **Wireless sensor-actuator networks (WSAN)** include both actuators and sensors and are an extension to WSN. Actuators are mobile and have a higher capability than sensors. Sensors report events to actuators, which may act on both the sensors and the environment. For example, when a sensor monitors a high temperature (e.g. a fire), it reports the event to the actuators, and actuators with water sprinklers are expected to arrive at the scene of the fire immediately in order to prevent it from spreading. Actuators may also replace failed sensors. They coordinate among themselves to decide which one will act. WSAN can be applied in a wide range of applications, including environmental monitoring, intrusion detection, etc. Mobile hosts, sensors, and actuators will be referred to simply as *nodes* here, following our generalization and describing a protocol that works for all three types of wireless networks. We assume a homogeneous network in which all nodes have the same transmission radius and thus the communication is assumed to be symmetrical. Thus, we consider broadcasting tasks in either ad hoc, sensor, or actuator networks, but not in any heterogeneous network containing several types of nodes.

Broadcasting is the task of sending a particular message from the source node to all other nodes in the network. Two mobile hosts are called neighbors if they are within transmission range of each other. Thus, they can communicate using a radio interface. We assume that there is a link (or edge) between two neighbors. Transmission range is limited due to signal strength attenuation with distance. Therefore, each mobile node acts as both a host and a router. A single transmission by one node reaches all its neighbors (one-to-all communication model). A source can send a message to a node that is not its neighbor via an intermediate node, which retransmits the packet received from the source (multi-hop communication). Broadcasting is an important phenomenon in this network and is widely applied for the transmission of important messages, route discovery, spreading alarm signals, paging, location updates, routing in highly mobile networks, etc. The **reliability** of a broadcasting task is a measure of the number of nodes receiving messages. Throughout this paper, a *one-to-all model* is considered for broadcasting in wireless networks. Thus, message overhead increases. Transmissions consume considerable network resources that are already scarce in wireless networks, e.g. battery power and bandwidth. Therefore, it is important and cost-effective to minimize the number of transmissions in a network.

We consider all network scenarios with respect to mobility. Each node can be *static*, *moderately mobile*, or *highly mobile*. The classification between moderately and highly mobile nodes depends on node velocity. Mobility may cause changes in network topology, since new neighbors may appear, and nodes may move away from existing neighbors. Roughly speaking, the distinction between moderate and high mobility is based on the percentage of neighbor changes of a node while broadcasting is in progress. In accordance with the behavior of our new protocol, we have the following scenarios in mind. In *static* networks, nodes move very slowly so that they are assumed to be constant, and the network topology does not change during the broadcasting task. A network is moderately mobile if all nodes within it are either static or moderately mobile.

Thus, even a single moderately mobile node among many static nodes classifies the network as moderately mobile. Finally, a network is considered highly mobile if at least one node within it is highly mobile. Therefore, a few high-speed vehicles among a lot of slower pedestrians along the road can classify the network as highly mobile.

A **network partition** is a division of a network into two or more components. In a highly mobile network, the rapid movement of a node may cause it to leave the transmission range of its neighbor and enter the transmission range of other nodes. This changes the network topology and may partition the network. Consider Figure 1(a). Suppose that, node *s* (source) transmits a message. The message is received by all of its neighbor *b, c,* and *d.* Before any neighbor of *s* can transmit, node *e* moves closer to the transmission region of node *s*. This partitions the graph into two disconnected components, (*s, b, c, d, e*) and (*g, h, i*). This is shown in Figure 1 (b). Therefore, nodes *g, h* and *i* will not receive the message transmitted from *s*.

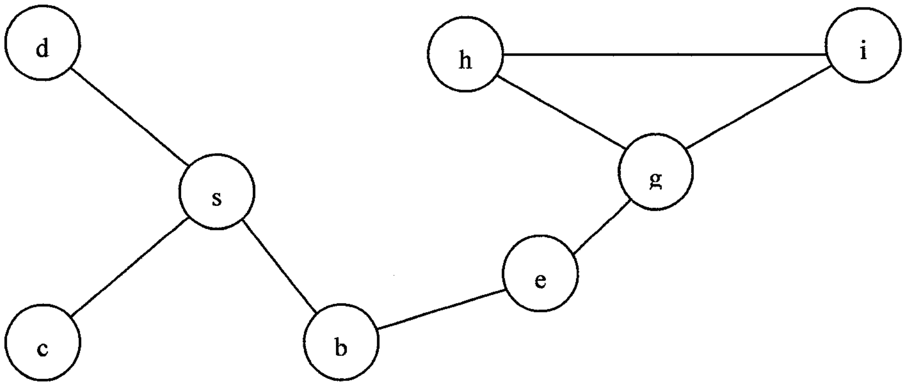


Figure 1 (a): Connected Network

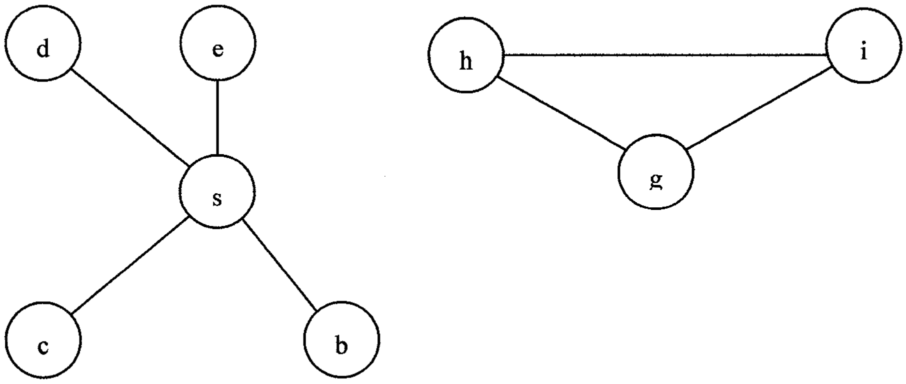


Figure 1 (b): Partitioned Network

In wireless networks, broadcasting and other related tasks can be classified into two types: *global* and *local*. In a *global* protocol, each node knows about all the other nodes and links in the network. The advantage is that a source can decide on the fastest route to deliver a message to a particular node. However, this introduces overhead since the node needs to store global information. Furthermore, any change in the network needs to be communicated to the other nodes. This communication is done locally and introduces communication overhead. For example, in Figure 1(a), if a link between *g* and *i* breaks, *g* will send this information to *e* which in turn will send it to *b*. Finally, *b* will send this information to source *s*. On the other hand, in a *local* protocol, a node knows only about its 1 or 2 hop neighbors. Consequently, if a link between two nodes changes, the nodes only need to convey this information to their neighbors. Thus, the local protocol reduces communication overhead.

In *centralized* broadcasting, the source node determines the complete broadcast process. For example, if a source node decides that it will employ plain flooding, then all nodes receiving the message will employ plain flooding. In *distributed* broadcasting, the broadcast process to be determined is distributed at each node. That is, a node receiving message may decide whether it will use scoped flooding or plain flooding depending on some condition. Any model (global or local) can be implemented in a distributed way where each node (or a subset of nodes) determines the type of flooding based on global or local information. Also, any model with local information can be implemented in a centralized way once a sufficient amount of information has been collected at the source through information exchanges.

1.2 Existing Solutions

Because of its limited transmission range, a source is normally unable to communicate directly with all nodes in a network. To fulfill a broadcasting task, many hosts thus have to act as routers by relaying the message to their physical neighbors. A simple way to broadcast is called *plain flooding*, where each node retransmits only when it receives a packet for the first time. However, this procedure generates too many redundant transmissions.

Scoped flooding [VO] is another flooding technique that uses neighbor elimination [PL, SSZ]. In this type of flooding, when a node receives a message for the first time, it compares the neighbor list of the transmitting node to its own neighbor list. If the receiving node's neighbor list is a subset of the transmitting node's neighbor list, then it does not retransmit the packet. Otherwise, it retransmits the packet.

Another type of flooding is *CDS based* broadcasting. A connected dominating set (CDS) [CS] consists of a set of nodes. In a connected network, the CDS is calculated in such a way that each node in the network either belongs to the CDS or has at least one neighboring node that belongs to CDS. In CDS based broadcasting, each node that receives the packet will retransmit it if and only if it belongs to the CDS. In this way, if only the CDS nodes transmit, all nodes will receive the message.

In a highly mobile network, the traditional methods of broadcasting (plain flooding, scoped, CDS-based) are inefficient since every node transmits at most once in these methods. These methods are unable to react if a node that has already transmitted discovers a new neighbor. The new neighbor may miss the message if it gets disconnected from the network when the original message was propagated. This is illustrated in Figure 1(b). Node *e* moves closer to node *s*, which has the message. Although, node *e* can receive the message from *s, s* will not transmit again since it has already transmitted at the beginning. This causes a reduction in reliability and an increase in the rate of failure.

Some broadcasting protocols have been developed to disseminate information across partitions. These protocols solve one side of the problem. Hyperflooding is a flooding technique similar to plain flooding. In hyperflooding, there are additional retransmissions whenever a node discovers a new neighbor. This increases reliability but at the cost of message overhead. Hyperflooding is described in [VO], [CEM], [HBR].

So far, only one protocol has been found that attempts to describe a single broadcasting protocol suitable for all mobility scenarios [VO]. However, it uses parameters (relative velocity or number of collisions) to detect network conditions and to change its characteristics. Thus, there is additional overhead in computing and storing parameters. Each node keeps track of relative movements in its neighborhood and compares them with low and high thresholds. Thus, different nodes can make different decisions and run different flooding modes. Nodes can decide to run scoped (restricted) flooding, blind flooding or hyperflooding based on the comparison. Measuring relative speeds poses additional problems as positional information or even special hardware is required.

1.3 Objective

Due to the limitations of existing solutions, it is important to develop a suitable method for broadcasting in both static and mobile networks that should be able to deal efficiently with changing network conditions and the discovery of new nodes. The goal of this thesis is to propose a new protocol for broadcasting messages in both static and mobile networks without using any parameters. That is, the protocol will adapt itself to any mobility scenario automatically. Furthermore, compared to some existing methods, this new method would have a smaller number of transmissions while maintaining good reliability. Mobile networks include wireless ad hoc networks and wireless sensor-actuator networks. The proposed technique uses connected dominating set (CDS) and does not need any parameters to detect network conditions or to change its characteristics. In a protocol proposed by [VO], relative velocity is used to detect network conditions. Computation and storage of this parameter introduces additional overhead. However, new method uses a variable T (time to leave) as input. T indicates the number of hello messages for which the broadcast message will be remembered by a node, after it receives the message. Variable like T is also used in existing methods. In the proposed protocol, each node keeps track of which neighbors receive a message and which do not. Variable T and lists used to keep track of which neighbors receive a message are easier to maintain and have negligible overhead.

1.4 Contribution

The proposed protocol is called "Parameterless Broadcasting from Static to Mobile networks" (PBSM). The main contribution is briefly described in this section. Each node maintains two lists: R (2-hop neighbors receiving the same message) and N (1-hop neighbors that have not received the message). After every periodic 'hello' message round, each node decides whether or not it is in CDS. The source transmits. If a node receives the message for the first time, it selects a waiting timeout (at random or inversely proportional to |N|). Nodes that are not in CDS select a longer timeout than nodes that are in CDS. Lists and timeouts are updated for every received copy of the message. When a timeout expires, the message is retransmitted if and only if N remains nonempty. Subsequent hello messages revise the N lists, revive timeouts if N is non-empty, and update CDS memberships. Thus, nodes may decide to retransmit regardless of previous silences or retransmit decisions between other hello intervals. This provides a seamless transition of protocol behavior from static to highly mobile scenarios. In static networks, number of transmissions is reduced as only CDS nodes transmit. In the case of mobile networks, unlike other existing methods, in proposed method, two nodes do not transmit every time they discover each other as new neighbors. When a node discovers a new neighbor, it first checks to see whether the new node is in R. If it is not in R, only then does the node consider transmitting. Otherwise, it does not transmit. Consequently, the new method is expected to have a smaller number of transmissions while maintaining good reliability in both static and mobile networks. Furthermore, we expect that this protocol, which does not use any thresholds (parameters), will perform better than the sole existing threshold based protocol. It should also be superior to all other existing broadcasting protocols since these addresses only single mobility scenario and perform poorly in other scenarios, while our protocol is adaptive and always remains competitive.

1.5 Publication

Adnan Afsar Khan, Ivan Stojmenovic, Nejib Zaguia, "Parameterless broadcasting in static to highly mobile wireless ad hoc, sensor and actuator networks", *32nd IEEE Conference on Local Computer Networks (LCN)*, 2007 (submitted).

1.6 Assumptions and Limitations

i) A unit disk graph is assumed. In this graph, all edges are less than or equal to a fixed threshold value. In a wireless network, a message sent by a node reaches all of its neighbors within transmission radius. If the transmission range of each node is *r*, then we can represent the wireless network by a unit disk graph in which all of its edges are at most *r*. An edge connects two nodes that are within communication range (i.e. distance between them is ≤ *r*). In the following figure, the centre of each circle represents a node (wireless host).

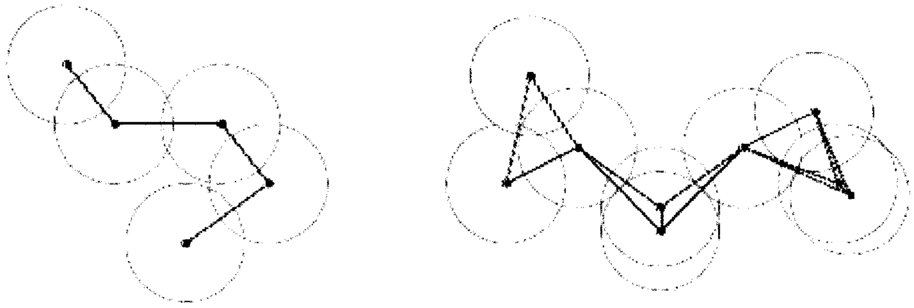


Figure 2: Unit Disk Graph

ii) It is assumed that all nodes have an equal transmission radius that is fixed.

iii) Ideal medium access (MAC) layer with no collisions is assumed in first part of simulation and simplified MAC layer with collision is assumed in second part.

iv) The proposed algorithm is localized, meaning that each node does not know about the global network configuration beyond its 1 or 2-hop neighbors. The nodes become aware of their 1 and 2-hop neighbors through hello messages. Each node periodically transmits hello messages and has two rounds. In the first round, each node sends its ID to all of its current neighbors. For example, in the following figure, if a node C receives a hello message from node B, D, and W, then node C adds B, D, and W to its (1-hop) neighbor list. After the first round, each node knows about all of its 1-hop neighbors. In the second round, each node transmits its neighbor list to all of its current neighbors. For example, in the following figure, if a node C receives neighbor-list from node B,D, and W, then node C will be aware that nodes U and A are its 2-hop neighbors as they are in the neighbor list of B. After the second round, each node learns about its current 2-hop neighbors.

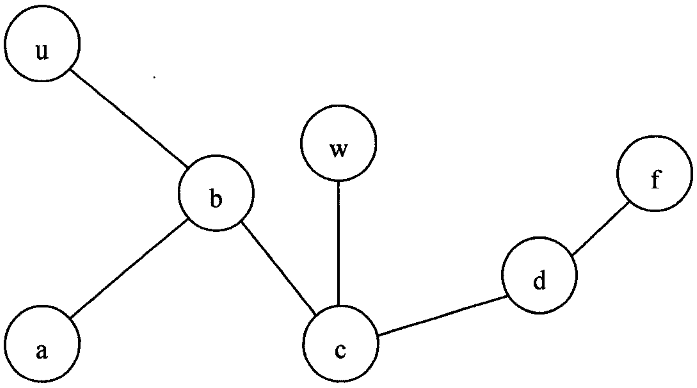


Figure 3: Example of hello message

1.7 Analysis and Simulation

The new protocol is first compared analytically to the existing methods described in [VO], [HBR], and [CEM]. The protocols are compared in both static and mobile scenarios. The analytical comparison is used to establish the claim that the proposed broadcasting protocol reduces the message overhead while maintaining good reliability for both static and mobile scenarios. The protocol is implemented and compared with some existing solutions by simulation. The simulation contains many scenarios including static, moderate, and high mobility. Moreover, the performance of the protocols is also compared based on D (a parameter used to define the speed of movement), density (d) (a parameter that indicates the average node degree of graph). The results of the simulation verified the claim that the proposed method has the lowest number of transmissions among the methods described, in both static and mobile networks. When ideal MAC layer is used, the proposed and existing methods have similar reliability. When simple MAC layer is used, the proposed method showed a significant gain in reliability.

1.8 Thesis Organization

In chapter 2, related work on broadcasting in wireless networks is presented. The proposed method is described in chapter 3. Chapter 4 presents the comparison of the proposed method with existing methods by example. Chapter 5 describes the implementation and simulation of broadcasting methods. The conclusion and future work is provided in chapter 6.

Chapter 2

Literature Review

In the past few years, there has been much related work on broadcasting in wireless networks. Therefore, there exists a plethora of proposed broadcasting protocols. A survey of these protocols is given in [SW, SSW]. The works that are important and relevant to our method are presented in this chapter.

2.1 Broadcasting in Static Networks

2.1.1 Scoped Flooding [VO]

Scoped flooding is a flooding technique that is suitable for static networks. In this type of flooding, when a node receives a message for the first time, it compares the neighbor list of the transmitting node to its own neighbor list. If the receiving node's neighbor list is a subset of the transmitting node's neighbor list, then it does not retransmit the packet. Otherwise, it retransmits. The advantage of scoped flooding is that it reduces retransmission. However, it is unable to react when a new neighbor is discovered.

2.1.2 Connected Dominating Set and Neighbor Elimination Scheme

In [ISGS], a connected dominating set (CDS) is defined as a set of nodes. In a connected network, the CDS is calculated in such a way that each node in the network either belongs to the CDS itself or has at least one neighboring node that belongs to CDS. The problem of deciding whether the calculated CDS is minimal is a NP-complete problem [APVH]. However, the calculation of the CDS itself has negligible overhead as each node uses only local (i.e. neighborhood 2-hop) information, without any additional messages, for CDS decisions. In CDS based broadcasting, each node that receives a packet, will retransmit it if and only if it belongs to the CDS. If only the CDS nodes transmit, all nodes will receive the message. Thus, the CDS based broadcasting reduces the number of transmissions.

In [GK], a method for calculating the CDS is provided and is as follows. Each node is initially colored as white. The densest node in the graph is then colored as black and all its neighbors as grey. Then, while there exists some white nodes, the grey node with the largest number of white neighbors is selected, colored as a black node, and all its white neighbors as grey nodes. Ties can be resolved by using certain keys (identifiers). At the end, the set of black nodes is a CDS. This algorithm provides a low number of nodes in the CDS but cannot be efficiently applied in a decentralized network. However, the size of the resulting set provides a good estimate of the limits one can reach with localized heuristics.

In [SSZ, S-ssz], a dominating set is calculated in the following way. A node is an *intermediate* node if it has two unconnected neighbors [WL]. A node *A* is covered by neighboring node *B* if each neighbor of *A* is also a neighbor of *B*, and *key(A)<key(B)*. Nodes not covered by any neighbor are *inter-gateway* nodes. A node *A* is covered by two connected neighboring nodes *B* and *C* if each neighbor of *A* is also a neighbor of either *B* or *C* (or both), *key(A)< key(B)*, and *key(A) < key(C)*. An intermediate node not covered by any neighbor becomes an *inter-gateway* node. An inter-gateway node not covered by any pair of connected neighboring nodes becomes a *gateway* node. Gateway nodes are also called *internal nodes* and they are in the dominating set. Both node degree and index are used in determining a node's key relative to its neighbor. A node with a higher degree is considered as a neighbor with a higher key. If the degrees of two nodes are equal, then the node with the higher index is considered as a neighbor with a higher key. A node can be covered by at most two of its neighbors.

In [DW], a more general rule for calculating CDS is proposed. It is called the *generalized self-pruning rule*, where coverage can be provided by an arbitrary number of connected 1-hop neighbors (instead of 1 or 2 as in [SSZ]). This definition was modified in [SW] to avoid similar message exchanges between neighbors, to the following form. Node *A* is covered by its 1-hop neighbors *B, C, D,...* if the neighbors *B, C, D, ...* are connected, any neighbor of *A* is a neighbor of at least one of nodes *B, C, D, ...* and *key(A) < min (key(B), key(C), key(D),...)*.

Carle and Symplot-Ryl [CS], computationally simplified the definition of determining a CDS by [SW]. First, each node checks whether it is an intermediate node. Then each intermediate node, **A**, determines whether it is in dominating set in the following way:

Construct subgraph G, consisting of only higher ***Id*** neighbors and existing edges between them.

- If G is empty or disconnected then *A* is in the CDS (G is connected if it has only one node).
- If G is connected but there exists any neighbor of *A* (not in G) that is not a neighbor of any node from G, then *A* is in the CDS.
- Otherwise A is covered and not in the CDS.
- Dijkstra's shortest path algorithm is used to test the connectivity of the subgraph of G.

This procedure is generalized since it allows coverage by any number of neighbors. It is computationally even less expensive than the two node coverage case. Each node compares the index (i.e. number or letter used to identify the node) of its neighbors with its own. Node *b*, with a smaller index, will consider node *a*, with a bigger index, as a higher id neighbor. Consequently, a node with higher index is more likely to be in the dominating set.

In *enhanced* CDS by Dai and Wu (elaborated in [ISS], [ISGS]), 2-hop neighbors can be used to cover 1-hop neighbors for smaller CDS. In this method, an intermediate node *u* is not in the CDS if there exists a connected set *A* of its 2-hop neighbors with higher priorities, such that each neighbor of u either belongs to A or is a neighbor of a node in A. Otherwise u is in the CDS.

In [PL, SSZ], broadcasting based on a *neighbor elimination scheme* (NES) is described. This scheme provides a simple way to eliminate redundant messages. In NES, a node does not need to rebroadcast a message if all of its neighbors have been covered by previous transmissions. In this method, when a node receives a message to broadcast, it sets a timeout period before transmitting. That is, nodes do not retransmit immediately, but wait for timeout duration while monitoring their neighborhood. If all neighbors become covered during the monitoring period then the node does not transmit. Otherwise, if the node's timeout expires and there are still uncovered neighbors, the node transmits. Timeout = (C / numberUncovered), where C is a constant and numberUncovered is the number of neighbors that have not received the packet, based on the node's knowledge.

In [SSZ], a broadcasting algorithm based on a dominating set and neighbor elimination [PL, SSZ] is proposed. In this algorithm, only nodes in the CDS will transmit. In this method, when a gateway node receives a message to broadcast for the first time, it initializes forwarding list (that consists of its neighbors) and sets a timeout period before transmitting. For each message received, the sender and any common neighbor of the sender and receiver are eliminated from the gateway node's forwarding list. When the gateway node's timeout expires, it transmits if its forwarding list is nonempty. Experimental results show that this algorithm is better than cluster-based and multipoint relay broadcasting.

CDS-based flooding [SSZ] does not allow multiple retransmissions by the same node. So, it is expected to perform poorly in mobile scenarios.

2.2 Broadcasting in Networks of Moderate Mobility

2.2.1 Plain Flooding

In *plain flooding*, when a node receives a message for the first time, it retransmits. When a node receives a copy of the same message, it doesn't retransmit. This is a traditional broadcasting protocol and its advantage is that it does not require neighbor knowledge. The drawback of blind flooding is that it generates many redundant transmissions and is unable to react when a new neighbor is discovered. This issue is studied in [NTCS] and is known as *broadcast storm problem.*

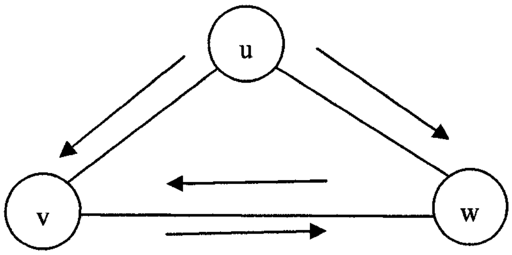


Figure 4: Redundant transmission in plain Flooding

The above figure illustrates redundant transmission in plain flooding. Node *u* first transmits. Nodes *v* and *w* receive the message for the first time and each of them transmits. The transmissions of nodes v and w are unnecessary as all nodes received the message from *u*.

Although connected dominating sets and neighbor elimination [SSZ] based flooding produce fewer transmissions than plain flooding, they require 2-hop neighbor knowledge. Furthermore, mobility makes the maintenance of such knowledge expensive, and therefore blind flooding is still considered to be the favorite protocol for scenarios with moderate mobility. Blind flooding may not suffice, however, in networks with temporary partitions and/or high mobility.

2.2.2 LW Protocol

In [LW], Lou and Wu study two environments to handle mobility. In the "static" environment, mobile hosts are allowed to roam freely in the working space. However, the broadcast process (including forward node selection and the broadcast process itself) is done quickly so that both 1-hop and 2-hop neighbor sets remain the same during the process for each host. In addition, each host has updated and consistent 1-hop and 2-hop neighbor sets when the broadcast process starts. Clearly, delivery of the broadcast packet is guaranteed as long as the selected forward nodes cover all hosts. In the "dynamic" environment, the broadcast process is still done quickly as in the static environment; so that both 1-hop and 2-hop neighbor sets remain the same during the process for each host. However, a host cannot update its 1-hop and 2-hop neighbor sets in a timely and consistent manner because the mobile hosts are moving at a fast speed. Under this model, the broadcast delivery rate is no longer 100 percent. A simulation result in [LW] shows that the broadcast delivery rate still remains high in an ad hoc network with slow- to moderate-speed mobile hosts (with respect to the transmission range) using an ideal MAC layer without contention and collision. This high delivery rate is partly due to the broadcast redundancy in selecting the forwarding nodes. Therefore, while excessive broadcast redundancy is harmful and will cause the broadcast storm problem, some degree of redundancy is useful for the purpose of reliability.

2.2.3 ZJ Protocol

Zhang and Jiang [ZJ] compare analytically and experimentally, the performance of several broadcasting algorithms in mobile scenarios. The selected algorithms are MPR (multipoint relay) and dominating set based on a generalized definition by Dai and Wu [DW]. Both schemes are improved by considering the relative stability levels of links in neighborhood and averaged over the network. This level may alter the decisions about forwarding. Thus, the modification adjusts the level of flooding based on mobility, but does not provide the opportunity for multiple retransmissions by the same node, which may be required in highly mobile scenarios.

Solutions [LW, ZJ] do not allow multiple retransmissions by the same node. As a result, they are expected to perform poorly in highly mobile scenarios.

2.3 Broadcasting in Highly Mobile Network

2.3.1 Hyperflooding

Hyperflooding is like plain flooding. The only difference is that when a node receives a data packet or hello message from a node that is not in the current neighbor list, it retransmits even though it has transmitted before. Additional retransmission ensures that a new node receives the message this time if it misses the original message. Thus, it increases reliability at the cost of transmission. Hyperflooding is described in [VO].

2.3.2 HBR Protocol

In highly mobile networks, when a new node is discovered, the protocol proposed by Hahner, Becker and Rothermel [HBR] behaves like hyperflooding [VO] but has two differences. The first difference is that instead of sending a full message, a short advertise message is sent. The second difference is that it uses a TTL (time for data to live) parameter. The protocol is described below.

The protocol uses data replication to increase the availability of data. Nodes maintain a local copy of the most recent state of all objects. Nodes also maintain a field called TTL (time for data to live). The TTL is decremented by each node that holds a copy of the data. A node u discovers new node v (v is not in u's current neighbor list) if it receives a hello or data message from v. Whenever a node discovers a new neighbor or receives a new message, it sends an advertise message as long as TTL has not expired. If TTL = 0, the data is deleted. Instead of sending a whole message, a node sends a short (advertise) message about its data. Upon receiving the advertised message, the nodes that need the data send a request message. If at least one neighbor requests data, the data is flooded;

otherwise it is not. When the full message is relatively short, this protocol is inefficient. Many advertising strategies are used. One strategy is to give data that has not been sent before a higher priority compared to data that has already been sent. Another strategy is to give data that has been sent few times a higher priority over data that has been sent even more times.

The advantage of this method is that it increases data reliability as data is advertised more than once. The downside is that there is message overhead due to some additional advertise and request message.

2.3.3 CEM Protocol

The method of Cooper, Ezhilchelvan, Mitrani [CEM] is an encounter based protocol. When a new node is discovered, this protocol behaves like hyperflooding [VO] but has two differences. The first difference is that, unlike hyperflooding, there are no additional transmissions when a node receives a data message from a node that is not in the current neighbor list. The second difference is that, in this protocol message expires after a certain number of encounters rather than after a certain time. It is described below.

Upon receiving a new message, node u does three things. First, it stores it and sets counter c to 0. Second, it adds the sender to its neighbor list if it is not already in the list. Third, if the current neighborhood contains nodes other than the one sending the message, then u retransmits the message and increases c by 1. An encounter is the discovery of a new neighbor that is not present in previous lists of neighbors (thus a node that has been a neighbor and disappears for just one 'hello' cycle is still considered as encountered upon return). Nodes do not maintain a history of neighbor list, but rather experience encounters at intervals. At every encounter, if $c< T$ (T is a parameter) then the node retransmits the message and increases counter c by 1. When $c=T$, the node deletes the message. The drawback is that it generates too many transmissions.

2.3.4 Other related work on broadcasting in highly mobile networks

R. O'Dell and R. Wattenhofer [OW], have designed flooding algorithms for dynamic networks with dynamic edges and fixed nodes. Among the assumptions they use are that the network stays connected at all times, and evolves slower than the message transit time between adjacent nodes. Their model, therefore, differs significantly from the model considered here. Their CounterFlooding algorithm requires one input *T* and intercepts two types of events: connectivity-driven events and message-driven events. When a message is first received at a node *u*, the counter *c* is reset to 0, and the message is retransmitted. If *u* is notified of a neighborhood change with the arrival of a connectivity-driven event, it broadcasts the message and increments the counter *c* by one until it reaches maximum *T*.

In [SE], Soedarmadji and McEliece relax the flooding algorithm's assumptions by removing the requirement that the network stays connected at all times, and extend the algorithm to solve the problem where dynamic nodes are also involved. This is achieved by introducing a delay between the actual connectivity change and its corresponding event. It is assumed that the nodes beacon their presence periodically to detect disconnections. The extended algorithm is reliable: it guarantees message-passing to all the destination nodes and terminates within a time bounded by a polynomial function of the maximum message transit time between adjacent nodes and the maximum number of nodes in the network.

Solutions [HBR, CEM, OW, SE] are based on hyperflooding where there are additional retransmissions whenever a new neighbor is found, up to a threshold number or a timeout. Thus, they will perform poorly for static networks since there are redundant transmissions. They may work well for moderate mobility, however, and are expected to work well for highly mobile scenarios.

2.4 Adaptive Broadcasting Method by VO

2.4.1 Description of Adaptive Protocol

Viswanath and Obraczka put forward an adaptive flooding method in which nodes, based on network conditions, can dynamically switch between different flooding techniques. The types of flooding techniques are *plain flooding, scoped flooding* and *hyperflooding.* This is the only protocol so far that attempts to describe a single broadcasting protocol suitable for all mobility scenarios [VO].

In *plain flooding*, when a node receives a message for the first time, it retransmits. When a node receives a copy of the same message, it doesn't retransmit. The drawback of blind flooding is that it generates many redundant transmissions and is unable to react when a new neighbor is discovered.

Scoped flooding is a flooding technique that is suitable for static networks. In this type of flooding, when a node receives a message for the first time, it compares the neighbor list of the transmitting node to its own neighbor list. If the receiving node's neighbor list is a subset of the transmitting node's neighbor list, then it does not retransmit the packet. Otherwise, it retransmits to the remaining nodes. The advantage of scoped flooding is that it reduces the amount of retransmissions. However, it is unable to react when a new neighbor is discovered.

Hyperflooding is like plain flooding. The only difference is that when a node receives a data packet or hello message from a node that is not in the current neighbor list, it retransmits even though it has transmitted before. Additional retransmission ensures that new node receives the message this time if it misses the original message. Thus, hyperflooding increases reliability at the cost of more transmissions.

Switching between flooding protocols takes place in two ways. One way is to use *network load* (MAC-Layer collisions) and the other way is to use *relative velocity*. The computation and storage of relative velocity or the number of collisions that a node requires for switching introduce additional overhead. Measuring relative speeds poses additional problems as positional information or special hardware is required.

Network load is indicated by MAC-layer collisions. In a network-load based switching method, each node computes the total number of collisions that have occurred in the current time window. Each node adaptively computes a low threshold and a high threshold value for the current time window. If the current number of collisions is lower than the low threshold, the node switches to hyperflooding mode. On the other hand, if the number of collisions is greater than the high threshold, then scoped flooding is selected.

The relative-velocity based switching criterion works as follows. Nodes send velocity (speed and direction) information as part of their hello messages. Each node is then able to compute its velocity relative to all of its immediate neighbors. Each node maintains a running average as well as the minimum and maximum value of relative velocity for the past five time windows. Based on the current value of relative velocity and its past history, each node adaptively chooses a low threshold and high threshold value for the current time window. If the current value of relative velocity is higher than the high threshold, the node switches to hyperflooding mode. If the relative velocity is below the low threshold, then scoped flooding is used. Otherwise, the node switches to plain flooding. Thus, different nodes can make different decisions and run different protocol modes. At lower speeds, adaptive flooding switches to the scoped flooding mode in an attempt to reduce redundant retransmissions. As the relative velocity increases, it switches to hyperflooding mode, thus increasing reliability and retransmission. Overall, a consistent message overhead is maintained.

2.4.2 Implementation of Adaptive Protocol

Scoped Flooding

There is just one difference between the implementation of scoped flooding and its description in section 2.4.1. When a node receives a message for the first time from a node that is not in the current neighbor list, it decides to transmit. This is because the receiver doesn't know the neighbor list of the new node.

Switching

Velocity is a vector since it has both magnitude (speed) and direction.

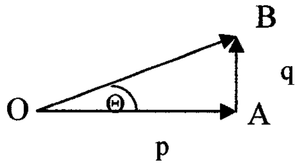


Figure 5: Vector Representation

The above diagram shows one way of representing a vector in a Cartesian co-ordinate system. Suppose that a node moves from point O to point B. Then its velocity is represented by the line OB. The vector OB can be decomposed into two components: OA (the horizontal component, parallel to x-axis) and AB (the vertical component, parallel to y-axis). Conventionally, **i** represents a unit vector in a positive x-direction and **j** represents a unit vector in a positive y-direction. If line OA is of length *p* and line AB is of length *q,* then the vector OB can be represented as **pi+qj.** The magnitude of velocity OB is $\sqrt{p^2 + q^2}$. The direction of velocity OB can be determined by, $\tan \Theta = q/p$.

Switching between various flooding modes is implemented by considering relative velocity. It is assumed that only nodes of this protocol are equipped with GPS. GPS is not used in implementation of proposed or CEM protocol. With the help of GPS, nodes can know their positions before and after movement. Therefore, a node can calculate its velocity using this information. Switching takes place only during hello messages. During hello message, nodes also send their velocity (speed and direction). Each node calculates its average relative velocity. That is, each node calculates its velocity relative to all of its neighbors and then finds the average of these relative velocities. For example, suppose that node *a* has three neighbors *b, c, d.* Node *a* has velocity $a_1\mathbf{i}+a_2\mathbf{j}$, node *b* has velocity $b_1\mathbf{i}+b_2\mathbf{j}$, node *c* has velocity $c_1\mathbf{i}+c_2\mathbf{j}$, and node *d* has velocity $d_1\mathbf{i}+d_2\mathbf{j}$. The velocity of node *a* relative to its neighbor *b* is $(a_1- b_1)\mathbf{i}+(a_2- b_2)\mathbf{j}$. Similarly, the velocity of node *a* relative to its neighbor *c* and *d* are $(a_1- c_1)\mathbf{i}+(a_2- c_2)\mathbf{j}$ and $(a_1- d_1)\mathbf{i}+(a_2- d_2)\mathbf{j}$ respectively. Therefore, the average relative velocity of node *a* is:

$[(a_1- b_1) + (a_1- c_1) + (a_1- d_1)]/3\ \mathbf{i} + [(a_2- b_2) + (a_2- c_2) + (a_2- d_2)]/3\ \mathbf{j}$

The value of average relative velocity of node *a* is:

$\surd\ \{[(a_1- b_1) + (a_1- c_1) + (a_1- d_1)]/3\}^2 + \{[(a_2- b_2) + (a_2- c_2) + (a_2- d_2)]/3\}^2$

Initially, nodes are moved for five time units. In each time unit, nodes exchange their velocities and each node calculates the average relative velocity. Each node keeps track of the minimum and maximum value of its average relative velocity attained in these five time units. As it is likely that the current average relative velocity of a node will be between this maximum and minimum value, the threshold values for the node are selected within this range. That is, low_threshold = min + (max-min)/3
high_threshold = max - (max-min)/3

Each node has different threshold values since each node calculates its own maximum and minimum value. Each node compares the value of its current average relative velocity with the high and low thresholds. If the current value of its average relative velocity is higher than the high threshold, the node switches to hyperflooding mode. If the average relative velocity is below the low threshold, scoped flooding is used. Otherwise, the node uses plain flooding. This ensures that a node, which has more relative movement in its neighborhood, is more likely to be in hyper mode.

After switching is done, the current value of the average relative velocity is compared with the minimum and maximum value. If the current value of the average relative velocity is less than the minimum value, then the minimum value is changed. Similarly, if the current value of the average relative velocity is higher than the maximum value, then the maximum value is changed. A change in maximum or minimum value will automatically change the threshold value. This is done so that the threshold value is updated during the hello message based on past history (max or min value), since what is current will become past in the next hello message. Switching is not clearly described in [VO] and there was no response from the authors when they were contacted. Therefore, the implementation of switching closely matches with the one stated in the article but may differ slightly.

Example showing operation of adaptive protocol:

The switching in this example is based on relative velocity.

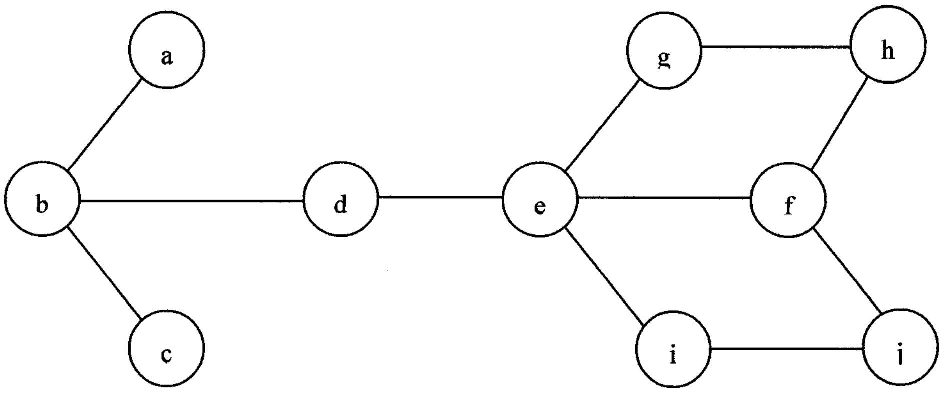


Figure 6: Example of adaptive protocol

Let l_u and h_u denote the low and high threshold values of node *u*. Let v_j be the velocity (vector) where $v_j = x\mathbf{i} + y\mathbf{j}$. Node *b* and *c* move from outside to their respective positions with high velocity v_1 and v_2 respectively. Node *h* moves with medium velocity v_3 but its neighborhood does not change. Node *j* moves with medium velocity v_4 but its neighborhood also does not change. The rest of the nodes are static. Hello message takes place and the nodes send their velocities as part of the hello message. Each node calculates its velocity relative to its neighbor and finds the average of these velocities. For example,

ave. rel. vel. of c = | vel of c – vel of b | = | v_2 - v_1 | (< l_c)

As the average relative velocity of *c* is less than l_c, node *c* will be in scoped mode.

$$\text{ave. rel. vel. of } b = \frac{(\text{vel of } b - \text{vel of } c) + (\text{vel of } b - \text{vel of } d) + (\text{vel of } b - \text{vel of } a))}{3}$$

$$= | v_1 - v_2 + v_1 - 0 + v_1 - 0 | / 3$$

As the average relative velocity of *b* is greater than h_b, node *b* will be in hyper mode.

$$\text{ave rel. vel. of } d = [(\text{vel of } d - \text{vel of } b) + (\text{vel of } d - \text{vel of } e)] / 2$$

$$= | 0 - v_1 \quad + \quad 0 | / 2$$

As the average relative velocity of *d* is greater than h_d, node *d* will be in hyper mode.

$$\text{ave rel. vel. of}_{,} f = \frac{(\text{vel of } f - \text{vel of } h) + (\text{vel of } f - \text{vel of } j) + (\text{vel of } f - \text{vel of } e)}{3}$$

$$= | 0 - v_3 \quad + \quad 0 - v_4 + 0 - 0 |/ 3$$

As the average relative velocity of *f* is greater than h_f, node *f* will be in hyper mode.

$$\text{ave. rel. vel. of } h = ((\text{vel of } h - \text{vel of } f) + (\text{vel of } h - \text{vel of } g)) / 2$$

$$= | v_3 - 0 + v_3 - 0 | / 2$$

As the average relative velocity of *h* is greater than h_h, node *h* will be in hyper mode.

$$\text{ave rel. vel. of } j = ((\text{vel of } j - \text{vel of } f) + (\text{vel of } j - \text{vel of } i)) / 2$$

$$= | v_4 - 0 + v_4 - 0 | / 2$$

As the average relative velocity of *j* is greater than l_j but less than h_j, node *j* will be in plain mode.

$$\text{ave. rel. vel. of } i = ((\text{vel of } i - \text{vel of } j) + (\text{vel of } i - \text{vel of } e)) / 2$$

$$= | 0 - v_4 + 0 - 0 | / 2$$

As the average relative velocity of *i* is greater than l_i but less than h_i, node *i* will be in plain mode.

Node *e* will be in scoped mode as *e* and its neighbor's velocity is zero.

The following table shows the nodes and their corresponding flooding mode

Node	Flooding Mode	Node	Flooding Mode
a	hyper	f	hyper
b	hyper	g	plain
c	scoped	h	hyper
d	hyper	i	plain
e	scoped	j	plain

Table 1: Flooding mode of nodes

From the above example we can see that a static node (e.g. *d* or *i*) can be in hyper or plain mode if there is movement in its neighborhood. On the contrary, a node that has moved with higher velocity (e.g. *c*) can end up in scoped mode if there is similar movement in its neighborhood as relative (difference) velocity decreases. Node *f* is static and there is little movement in its neighborhood. Since two neighbors of *f* move, *f* employs hyperflooding. Overall, the movement of a node and its neighbors influences the type of flooding exhibited by the node.

Node *b* will transmit as soon as it discovers its new neighbors since it is in hyper mode. Similarly, nodes *a* and *d* will transmit when they discover *b*. Node *c* will discover *d* but it will not transmit because it is in scoped mode.

Suppose that broadcasting starts from node *b*. Node *c* receives the message but it will not transmit as all of its neighbors are also neighbors of the sender (scoped mode). Nodes *a* and *d* immediately transmit since they are in hyper mode. Then node *e* (scoped mode) transmits because it has neighbors that are not neighbors of sender *d*. The remaining nodes (*f, g, h, i, j*) transmit as soon as they receive the message for the first time as they are in plain or hyper mode.

Chapter 3

Parameterless Broadcasting from Static to Mobile networks

3.1 Protocol Overview

The proposed protocol is called “Parameterless Broadcasting from Static to Mobile networks” (PBSM) and can be described as follows. Nodes periodically exchange hello messages in order to update local knowledge up to two hops. The CDS [CS] is calculated after each hello message round. A source node transmits the message. Upon receiving the message for the first time, each node initializes two lists: receiver list R containing all nodes (up to 2-hop distance) believed to have received the packet, and list N containing neighbors in need of the message. Furthermore, each node sets a timeout waiting period when it receives a message for the first time. If a node is not in the CDS then it selects a longer timeout than a node from the CDS so that the nodes in the CDS react first. Transmission by CDS nodes is likely to cover all neighbors of non-CDS nodes and therefore non-CDS nodes select a longer timeout.

For each further message received and own message sent, every node updates its R, N and timeout. At the end of its timeout it transmits if N is nonempty. The message is memorized until *T* hello messages are received. For each hello message received, N is updated. Nodes that are no longer 1-hop neighbors are eliminated from the list, while new neighbors not present in R, are added. Every node that has received a message starts or adjusts (if it is already running timeout) its timeout if the new N is non-empty.

3.2 Example showing the operation of PBSM

Before giving algorithmic details, we will illustrate the protocol's behavior.

3.2.1 Static Network

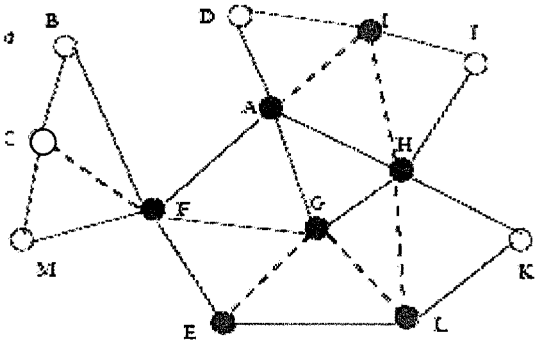
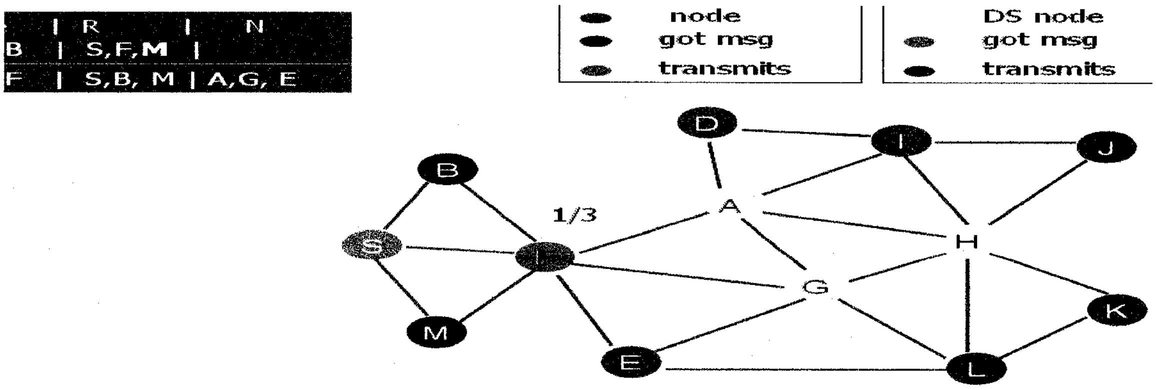


Figure 7: Source C and CDS nodes A, F, G, and H in a Static Connected Network

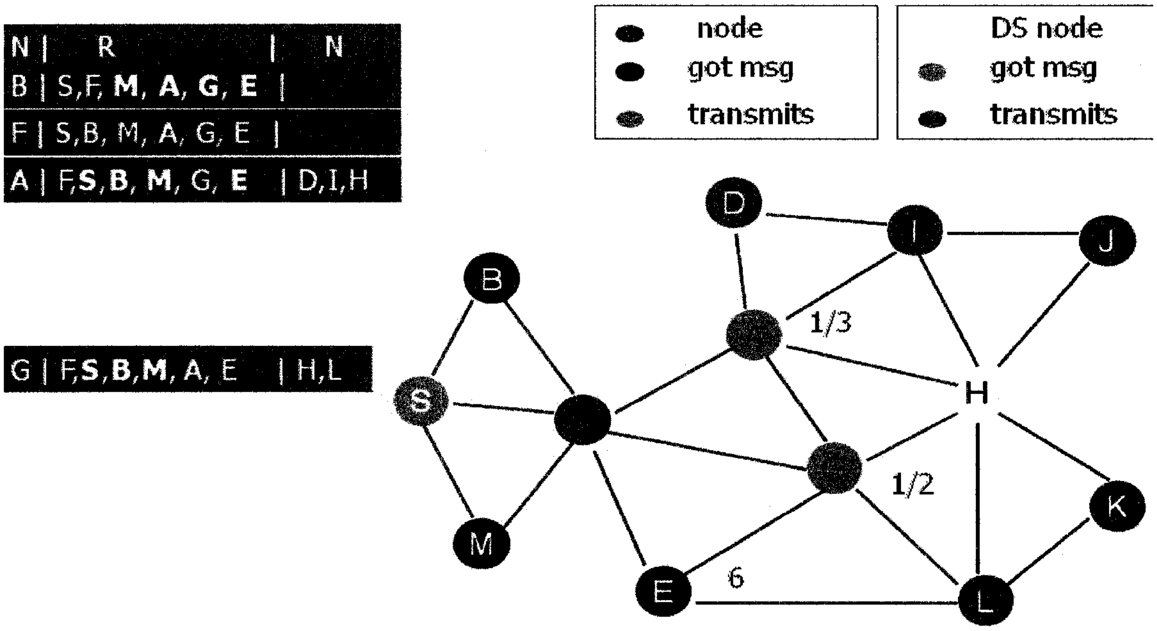
Consider first the behavior of PBSM in a static network (see Figure 7). Nodes B, M, D, J, and K are non-intermediate. The remaining nodes in the graph are intermediate. All intermediate nodes decide whether or not they are in the CDS [CS]. Nodes A, F, G, and H are in the dominating set. Node S (C) is the source and it transmits. Among the neighbors of S, only node F is in the CDS and thus it will retransmit first. F runs a timeout (= 1 / |N|). The following figure shows the timeout at F and the R, N lists of nodes B and F. The 2-hop neighbors in the R list are shown in bold.



	R	N
B	S,F,**M**	
F	S,B, M	A,G, E

Figure 8: (a) PBSM on a static network after S transmits.

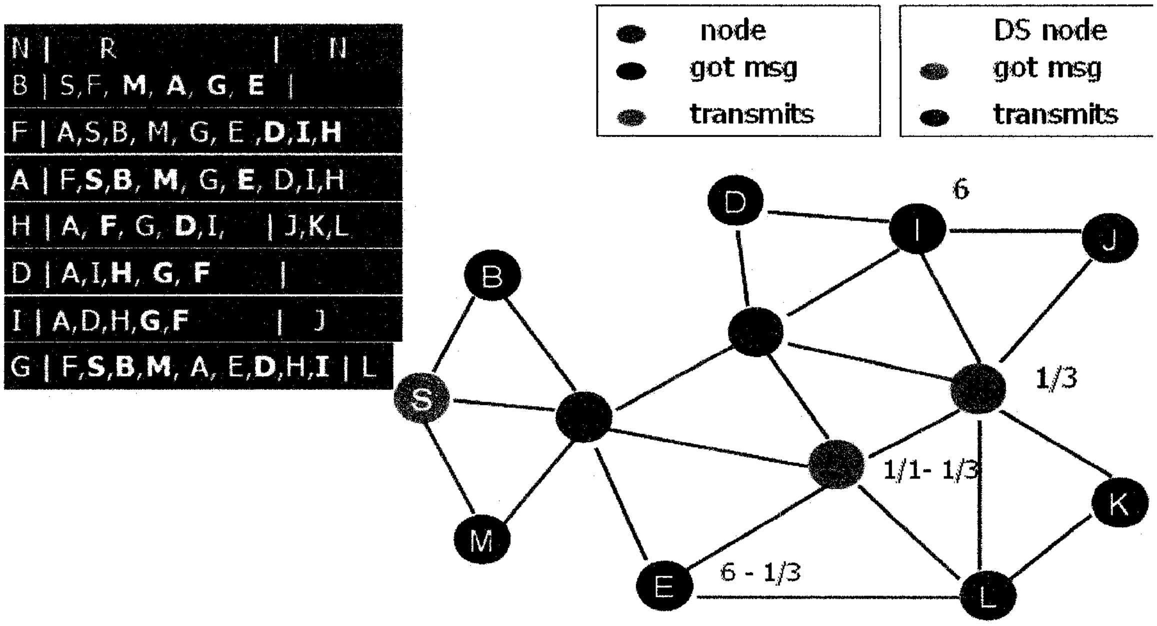
F transmits. Neighbors A, G, E of F receive the message for the first time. As A and G are in the CDS, they run timeout proportional to their number of uncovered neighbors, 1/3 (neighbors D, I and H of A) and ½ (neighbors H and L of G), respectively. As node E is not in CDS, it runs a longer timeout 6 (i.e. 5 + 1/1, one uncovered neighbor L). The following figure shows the R, N list and timeout after F transmits.



N	R	N
B	S,F, **M, A, G, E**	
F	S,B, M, A, G, E	
A	F,**S,B, M,** G, **E**	D,I,H
G	F,**S,B,M,** A, E	H,L

Figure 8: (b) PBSM on a static network after F transmits.

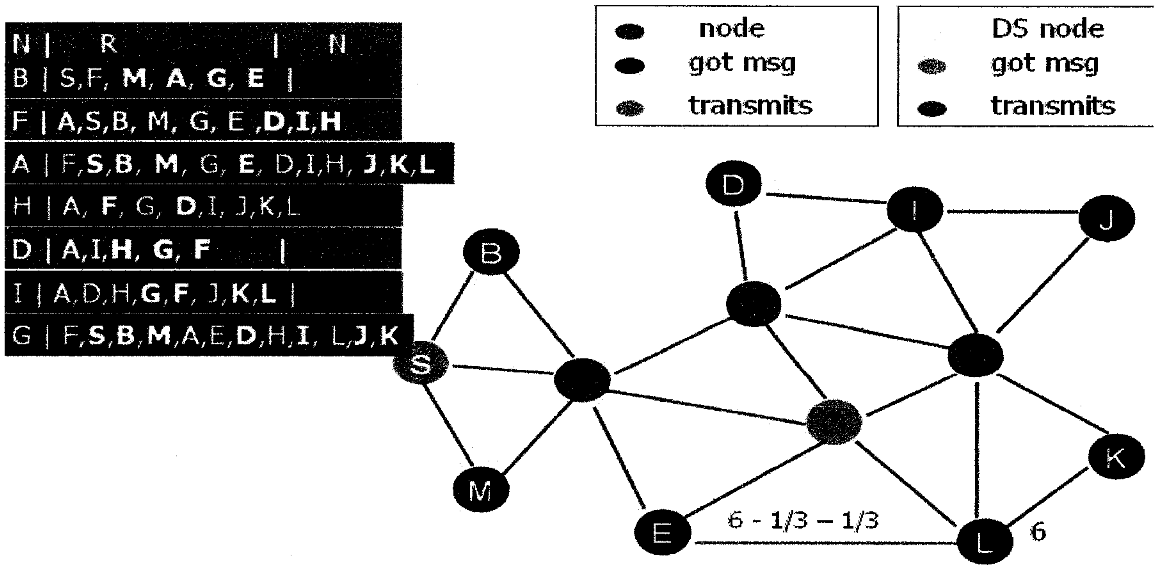
Since A has a smaller timeout, A transmits. Among the neighbors of A, only F, G and H are in CDS. F has already transmitted. At this moment G extends its timeout to 1/1-1/3 (one uncovered neighbor L left, 1/3 time elapsed). H receives the message for the first time and sets its timeout in proportion to 1/3 (neighbors I, J, K). E reduces its timeout to 6-1/3 (1/3 time elapsed). Node I (non-CDS) sets a longer timeout 6 (i.e. 5 + 1/1, one uncovered neighbor J).The following figure shows the R, N list and timeout after A transmits. Only B's R list doesn't change.



N	R	N
B	S,F, **M**, **A**, **G**, **E**	
F	A,S,B, M, G, E ,**D**,**I**,**H**	
A	F,**S**,**B**, **M**, G, **E**, D,I,H	
H	A, **F**, G, **D**,I,	J,K,L
D	A,I,**H**, **G**, **F**	
I	A,D,H,**G**,**F**	J
G	F,**S**,**B**,**M**, A, E,**D**,H,**I**	L

Figure 8: (c) PBSM on a static network after A transmits.

Since H has a smaller timeout than G, H transmits. Among the neighbors of H, only G and A are in the CDS. A has already transmitted. G and I's N lists now become empty and therefore their timeouts stop. E further reduces its timeout to 6-1/3-1/3 (1/3 more time elapsed). Node L (non-CDS) sets a longer timeout 6 (i.e. 5 + 1/1, one uncovered neighbor E). The following figure shows the R and N list after H transmits. B, F and D's R lists do not change.



N	R	N
B	S,F, **M**, **A**, **G**, **E**	
F	**A**,S,B, M, G, E ,**D**,**I**,**H**	
A	F,**S**,**B**, **M**, G, **E**, D,I,H, **J**,**K**,**L**	
H	A, **F**, G, **D**,I, J,K,L	
D	**A**,I,**H**, **G**, **F**	
I	A,D,H,**G**,**F**, J,**K**,**L**	
G	F,**S**,**B**,**M**,A,E,**D**,H,**I**, L,**J**,**K**	

Figure 8: (d) PBSM on a static network after H transmits.

Since E has a smaller timeout than L, E transmits. Among the neighbors of E, only G and F are in CDS. F has already transmitted and G's N list is empty. L's N list now becomes empty and therefore its timeout stops. The following figure shows the R and N list after E transmits.

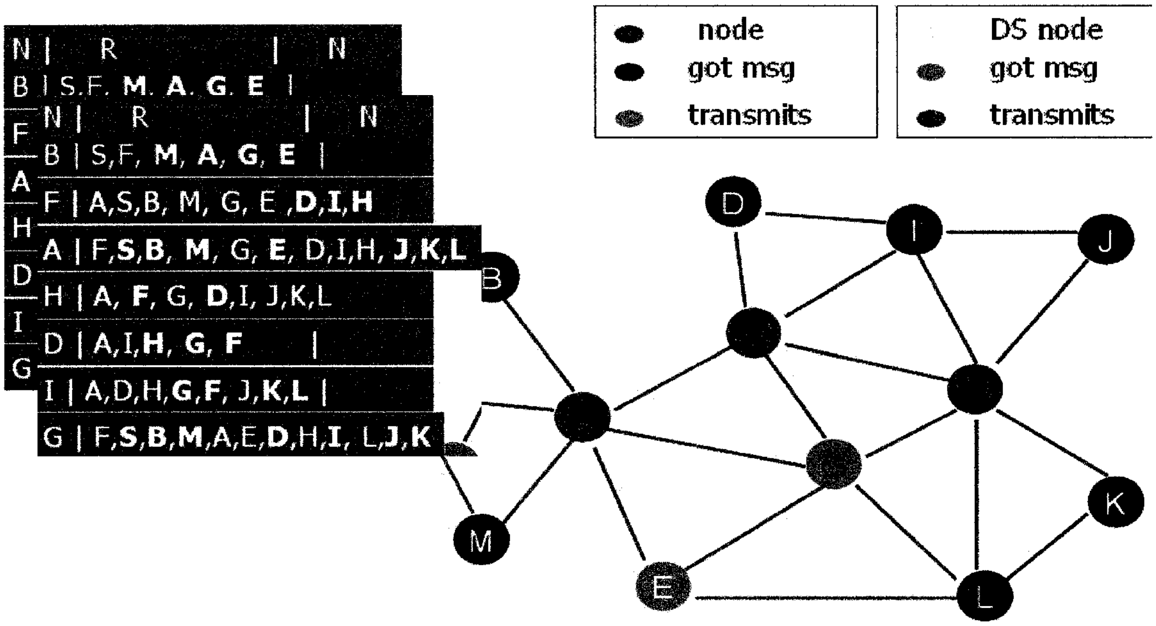


Figure 8: (e) PBSM on a static network after E transmits.

The proposed method broadcasts based on CDS and neighbor elimination scheme [PL, SSZ]. CDS nodes are selected [CS] in such a way that if only the CDS nodes transmit, all nodes in the static network will receive the message. Only four nodes (A, F, G, and H) are in the CDS. Although node G is in the CDS and initially decides to transmit (i.e. runs timeout), it does not transmit because all of its neighbors receive the message. On the contrary, the non-CDS node E transmits as its N list is non-empty according to its local knowledge. Consequently, all of the nodes in the network receive the message via the retransmissions of only four nodes. In a static network, the number of retransmissions is at most equal to the number of CDS nodes.

3.2.2 Mobile Network

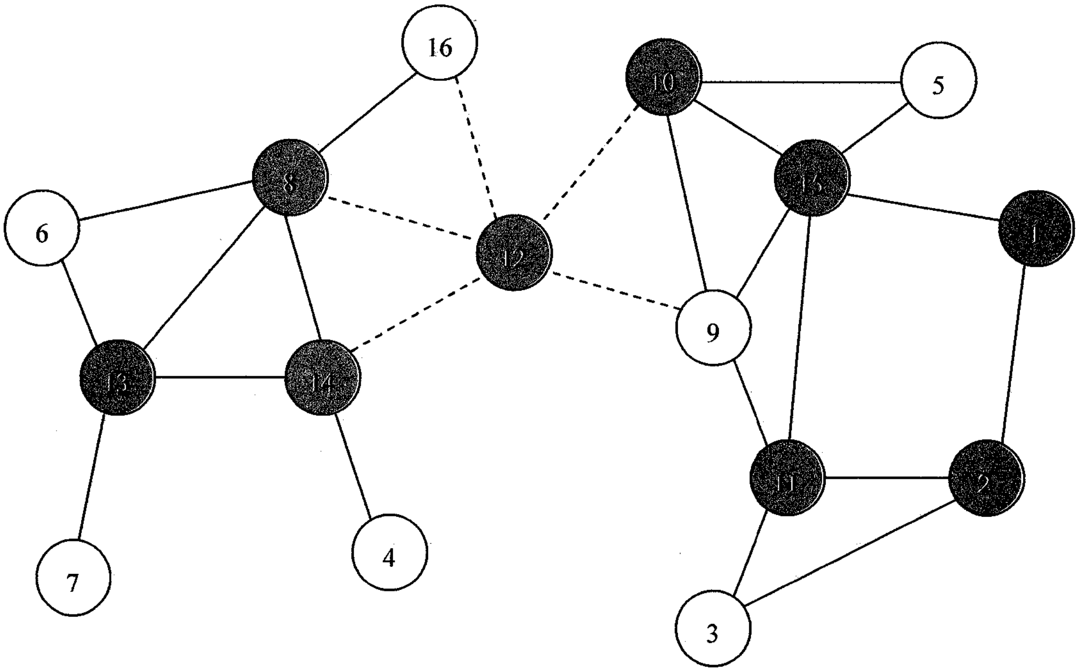


Figure 9: Node joining two disconnected network

PBSM is applied in Figure 9. Initially, the graph was divided into two disconnected components as node 12 was not there. Node 14 is the source and it transmits. All nodes on the left side of node 12 receive the message because they are directly or indirectly connected to the source. Nodes on the right side of node 12 do not receive the message. After sometime, node 12 arrives closer to source and other nodes (as shown above). Node 12 does not have the message. Hello message takes place and node 12 is discovered by its neighbors. Nodes 8, 16, and 14 decided to transmit because their N list contains 12, and they run timeout (1/1). Then CDS is calculated. Nodes 1, 2, 8, 10, 11, 12, 13, 14, and 15 are in the CDS. Node 8 transmits first. When nodes 14 and 16 receive the message, they remove 12 (12 is a neighbor of sender 8) from their N lists. Nodes 14 and 16 cancel their transmission as their N lists becomes empty. When node 12 receives the message from node 8, it removes 8, 16, and 14 (8 is the sender and 14 and 16 are its neighbors) from its N list. Node 12 is in CDS and it receives the message for the first time; therefore it runs timeout. The timeout of node 12 is ½ as it has 2 neighbors (9, 10) left in its N list. Please note that the timeout of a CDS-node = 1 / |N| whereas the timeout of a non-CDS node = 5 + 1 / |N| (longer timeout). The left column of the following table shows the nodes that transmit (in order) and the corresponding right column shows the nodes whose timeout changes due to the transmission (nodes in CDS are shown in bold).

Transmission	**Node-Timeout**
8	**12** - ½; **14** - cancels timeout; 16 - cancels timeout
12	9-(5+½); **10** - ½
10	9-(5+1/1-½);**15** - ½
15	**1** - 1/1; 9 - cancels timeout ;**11** – ½
11	**1** – (1/1-1/2); **2** – 1/1
1	**2** – cancels timeout

Table 2: Transmission order and timeout of PBSM

The CDS is recalculated after node 12 is discovered during the hello message. The CDS-nodes will transmit first since they have a shorter timeout. All nodes will receive the message by the retransmission of only CDS nodes.

3.3 Features of Protocol

The proposed technique uses connected dominating set (CDS) [CS], neighbor elimination scheme (NES) [PL, SSZ], and does not need any parameters to detect network conditions or to change its characteristics. The algorithm is localized, meaning that each node knows only about its 1 or 2-hop neighbors. It is suitable for broadcasting messages in both static and mobile networks. In static networks, the number of transmissions is reduced as only the CDS nodes transmit. Furthermore, due to NES, some nodes may not need to transmit. In mobile networks, two nodes discover each other when one node enters the transmission region of the other. Unlike other methods, in the proposed method, two nodes do not transmit every time they discover each other as new neighbors. When a node discovers a new neighbor, it first checks whether the new node is in R. If it is not in R, only then does the node consider transmitting. Otherwise, it does not transmit. Consequently, the new method is expected to have a lower number of transmissions while maintaining good reliability in static and mobile networks.

3.4 1-hop and 2-hop versions of protocol

The proposed method has two versions. The first version uses 1-hop positional information. 1-hop positional information means that a node knows the positions of its immediate neighbors. Because the node knows their positions, it can calculate the distance between its neighbors and thus find out whether its neighbors are neighbor themselves. The second version uses 2-hop topological information. This means that a node knows about the neighbors of its neighbors (2-hop) but doesn't know the position of its 2-hop neighbors. Thus, the node does not know whether its 2-hop neighbors are neighbor themselves.

The only difference between the two versions is the R list. For the 1-hop version, when a node X, receives a message, it adds the sender and any common neighbors of the sender and X into its R list. Therefore, the R list of node X consists of those nodes that have received the message (up to 1-hop). On the other hand, in the 2-hop version, when a node X receives a message it adds the sender and the neighbors of the sender into its R list. Therefore, the R list of node X, consists of those nodes that have received the message (up to 2-hops).

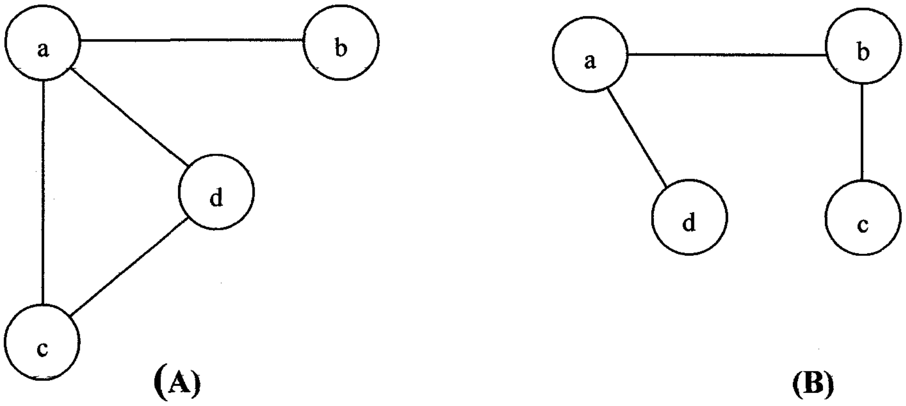


Figure 10: Example showing difference between 1-hop and 2-hop versions of protocol

1-hop version:

Consider Figure 10A. Suppose that the node *a* in Figure 10A transmits the data (broadcast) message. When node *b* receives the message, it includes the sender (node *a*) in *b*'s R list. Similarly, when node *c* receives the message, it includes the sender (node *a*) and their common neighbor (node *d*) in its R list.

Consider Figure 10B. Suppose that node *c* of Figure 10A moves into the vicinity of node *b*. During the hello message round, node *c* and node *b* discover each other as new neighbors. Node *c* does not contain node *b* in its R list and vice-versa. Therefore, one node (*b* or *c*) will transmit.

2-hop version:

Consider Figure 10A. Suppose that node *a* in Figure 10A transmits the data (broadcast) message. When node *b* receives the message, it includes the sender (node *a*) and its neighbors (nodes *c* and *d*) in node *b*'s R list. Similarly, when node *c* receives the message, it includes the sender (node *a*) and its neighbors (nodes *b* and *d*) in its R list.

Consider Figure 10B. Suppose that node *c* of Figure 10A moves to the vicinity of node *b*. During the hello message round, node *c* and node *b* discover each other as new neighbors. However, node *c* contains node *b* in its R list and vice-versa. Therefore, neither node *b* nor *c* transmits a data message in the 2-hop version, whereas in the 1-hop version one node (*b* or *c*) transmits. Consequently, the 2-hop version produces a lower number of transmissions than the 1-hop version.

3.5 Protocol Description

3.5.1 Hello message round and Connected Dominating Set (CDS)

Hello messages take place periodically and have two rounds. In the first round, each node sends its ID to all of its current neighbors. Upon receiving a hello message, a node adds the hello message originator to its neighbor list (if the list does not already contain that node). After the first round, each node knows about all of its neighbors (1-hop). In the second round, each node transmits the list of IDs of the nodes from whom the hello messages were just received in the first round. In this way, each node learns about its current 2-hop neighbors. Therefore, the proposed algorithm is localized, meaning that each node does not know about the global network configuration beyond its one or two-hop neighbors. Each node maintains an N list (neighbors who didn't receive messages) and R list (nodes that have received a message).

After every periodic hello message, the connected dominating set (CDS) is calculated as described in [CS]. Each node determines its ID relative to its neighbor in the following way. Each node compares the degree (size of the neighbor list) of its neighbors with its own. Node *b*, with a smaller degree, will consider node *a*, with a bigger degree, as a higher ID neighbor. If the degree of two nodes is equal, then the node with the higher index (i.e. number or letter that is used to identify a node) is considered as a higher ID neighbor. Consequently, a node with a higher degree or index is more likely to be in the dominating set.

3.5.2 Broadcasting process

The 1-hop version of the protocol requires position information. To obtain position information, nodes need to be equipped with GPS or be able to find their relative coordinates by measuring signal strengths (or time delays) in mutual communication. Therefore, there is an overhead in obtaining position information. For static networks, the 1-hop version of the protocol is similar to the one in [SSZ] but differs in the way that the CDS is calculated. The 2-hop version of the protocol produces a lower number of transmissions and it does not need any position information. As the 2-hop version of the protocol is more efficient, the 2-hop version is discussed, implemented, and compared with the other existing protocols. The proposed method is suitable for any mobility scenario. The algorithm will make use of a dominating set [CS] and neighbor elimination scheme [PL, SSZ]. The method is called PBSM (Parameterless Broadcasting from Static to Mobile networks) and is described below.

Initially, only the source has the message to be transmitted. Hello message takes place and each intermediate node determines whether it is in the dominating set or not. After this, the source transmits. The nodes react to two events:

a) receiving the message intended for broadcasting, and

b) sending/receiving hello messages at fixed intervals.

These activities proceed until a fixed number *T* of hello messages occurs after the initial source transmission. *T* is a parameter that determines the lifetime of the message being broadcasted. The remaining lifetime can be carried with the message if the synchronization of lifetime is desirable. It may be beneficial for each node to start its own *T* at the moment of receiving the first copy of the message, in order to reduce message size by not informing neighbors of the remaining lifetime.

a) Receiving and retransmitting broadcast messages between two hello messages:

The following steps are repeated until the next hello message time.

- [For any node (CDS or non-CDS)]. If the message is received for the first time, the node initializes its R and N lists as follows. All reported (at the last hello message time) 1-hop neighbors of the sender, and the sender itself, are first included in R, the receiver list. Therefore, the R list of receiver, consists of nodes that have received the message (up to 2-hop) based on the available and current local knowledge of the node. However, when the sender is not in the receiver's current neighbor list, the receiver adds only the sender to its R list. This is because the receiver doesn't know the neighbor list of the sender. After that, the N (neighbors in need of the message) list will only consist of the 1-hop neighbors of the receiver that are not in the R list.
- If the message is received for the first time and list N is non-empty, the node starts a timeout. If the node is in the CDS, it sets a timeout where Timeout = 1/ |N| (|N| is the number of nodes in list N). If the node is not in the CDS, it selects a longer timeout than a node from the CDS so that the nodes in the CDS react first. Timeout = 5 + 1/ |N| (timeout of a CDS node can be at most 1). Alternatively, a random timeout can be used.
- [For any node (CDS or non-CDS)]. Whenever a node receives a message (not just for the first time), it keeps updating the lists for N and R which might be needed after the next hello message round. If a node is running timeout:
  For a CDS-node, Timeout = 1/ |N| - any elapsed time.
  For a non-CDS-node, Timeout = 5 + 1/ |N| - any elapsed time.
  If N becomes empty, it cancels retransmission (timeout). Alternatively a 'fresh' random number for timeout is generated if a node is running timeout.
- If two nodes have the same timeout, then the timeout of the node with the smaller index (number or letter used to identify the node) will expire first. When the timeout expires, if there are still neighbors left that are believed not to have received the message (N is non-empty), the node retransmits. All of its neighbors are included in R and all neighbors are now removed from N. (N is now empty).

b) Sending/receiving hello messages and deciding to broadcast related updates

The following steps take place during hello message

- A new round of hello messages is sent, and each node updates its neighbor lists N.

- When a node receives a hello message from a neighbor, the sender is checked to see whether it is in the accumulated R list. If so, the node ignores it for the purpose of transmitting. If not, the node then adds it to N.

- All nodes that have already received broadcast packets at any time consider possible new, or very first transmissions by reevaluating decisions (including nodes that have previously decided not to transmit (non-CDS), nodes that have already retransmitted, and nodes running timeout, with nonempty N list) as follows.

  - Each node (CDS or non-CDS) that was not running timeout but now has a non-empty N list, starts timeout (1/|N|) fresh from the hello message point.
  - Each node (CDS or non-CDS) that was running timeout but now has an empty N list, cancels timeout.
  - Each node (CDS or non-CDS) that was already running timeout sets its new timeout as follows:

    New Timeout = 1 / |N| - elapsed time since start of previous timeout

    Note that 'refreshed' or restarted timeout can also be considered. However, we opted for a described version to avoid problems with relatively short hello message intervals that may delay retransmissions (due to restart) for quite a long time, leading even to their expiration.

After the hello message, each intermediate node again determines whether it is in the dominating set using the updated neighbor list.

The whole process (a and b) is repeated for *T* hello messages.

Pseudo-Code of the broadcasting process

The following notations are used in pseudo-code:

- Ne (x) means the set containing all 1-hop neighbors of node x
- N_X means the N list (neighbors who didn't receive message) of node x
- R_X means R list (nodes that have received message) of node x
- m means data message, h means hello message, # h means no. of hello msg.
- T_X means the timeout of node x, CDS means Connected Dominating Set
- eT_X means the elapsed time if node x is already running timeout

Initially, the hello message takes place and each intermediate node determines whether it is in the dominating set or not. The source transmits.
The following steps are performed by each node (x).

```
REPEAT
  Repeat
    On receiving data message (m) by node x from node y:
       $R_X \leftarrow R_X + y + Ne(y)$
       $N_X \leftarrow Ne(x) - R_X$
       If $T_X = 0$ (timeout not running)
           $eT_X = 0$
       If ( x receives msg for first time ) or ( $T_X \neq 0$ )
         If (x ε CDS )
           $T_X \leftarrow 1 / N_X - eT_X$   ( $T_X = 0$ if $N_X$ is empty )
         Else
           $T_X \leftarrow 5 + 1 / N_X - eT_X$   ( $T_X = 0$ if $N_X$ is empty )

    On expiration of $T_x$
       If $N_X \neq \emptyset$
           x transmits data_msg (m)
           $R_X \leftarrow R_X + Ne(x)$
           $N_X \leftarrow Ne(x) - R_X$   ( $N_X = \emptyset$ )
  Until next hello message time

  At HELLO message time (periodic time )
       #h ← #h + 1   (number of hello messages )
      On receiving hello message (h) by node x from node y:
            If y is not in $R_X$
                $N_X \leftarrow N_X + y$
      If node x has data message (m)
                $T_X \leftarrow 1 / N_X - eT_X$   ( $T_X = 0$ if $N_X$ is empty )
                If ( $T_X < 0$ )
                  $T_X \leftarrow 1 / N_X$
      After hello message, CDS is again calculated.
UNTIL #h = $T$
```

3.5.3 Importance of the transmissions of non-CDS nodes

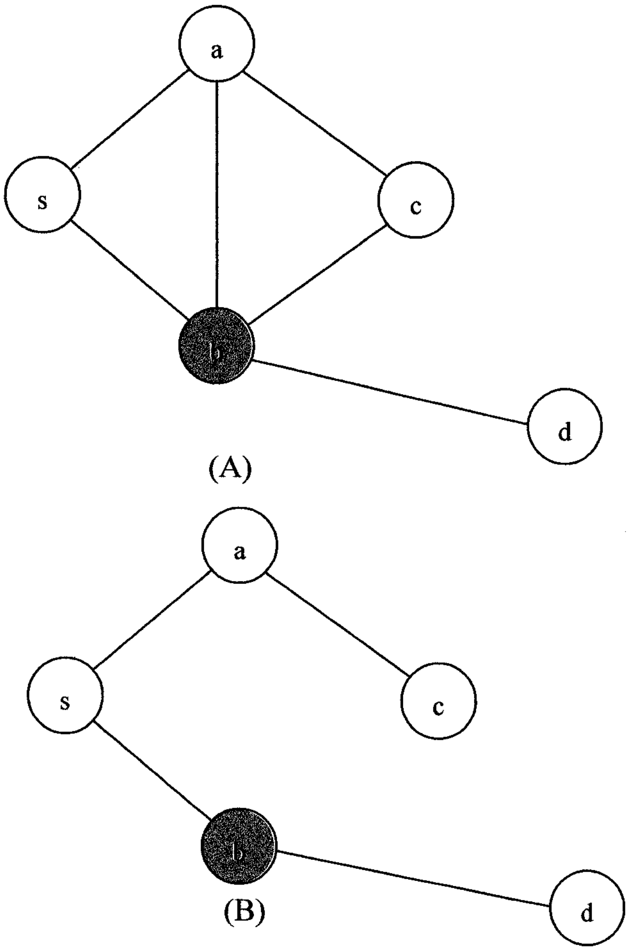


Figure 11: Importance of the transmissions of non-CDS nodes

The CDS is established right after the hello message. Consider the figure 11A. Node *s* is the source. Only node *b* is in the CDS as it has the highest degree. Suppose that after CDS is formed, node *b* moves down slightly such that the edges between *b* and *a,* and *b* and *c* break (see Figure 11 B). Source *s* transmits and the message is received by nodes *b* and *a*. Both nodes run timeout, and node *a* has a longer timeout as it is not in CDS. Node *b* transmits and only node *d* (also *s*) receives the message. Note that node *c,* which was supposed to get the message from node *b* (CDS) does not receive message as the edge between *b* and *c* has broken. However, node *a* (non-CDS), which is also running timeout, finds that its N list is still non-empty (contains *c*). Therefore, node *a* (non-CDS) will transmit and *c* will receive the message. This shows that the transmission of non-CDS nodes is necessary in mobile networks.

Chapter 4

New and Existing methods in different scenarios

After doing an extensive literature review, two existing protocols were selected for comparison. In this chapter, the PBSM protocol will be compared with the CEM protocol and adaptive flooding protocol by [VO]. These protocols are selected for comparison because they consider similar assumptions and are developed for highly mobile scenarios. Although they have some limitations, these protocols perform better than most of the other available protocols. Therefore, the protocols of [VO] and [CEM] are considered to be valid competitors.

4.1 Order of transmission of existing methods

All the neighbors of a sending node receive a message at the same time. The nodes make decisions about transmission as soon as they receive the message. If more than one neighbor decides to transmit, the neighbor with smallest ID will transmit first. The order of transmission proceeds according to a breadth-first search (BFS). That is, messages are sent first to the neighbors of the source. Each of these neighbors that decide to transmit will transmit. The node with the smallest ID will transmit first. Then the neighbor of the source's neighbor will make its own decision about transmission and so on.

4.2 Examples showing superiority of PBSM over other method

4.2.1 Static Network

The proposed method will be compared (by example) with methods described in [VO] and [CEM], considering the following static connected network.

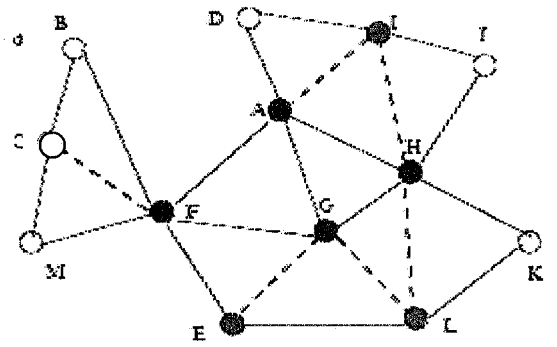


Figure 12: Source C and CDS nodes A, F, G, and H in a Static Connected Network

The details of the new method are described in Chapter 3. As discussed in section 3.2.1, all nodes in the network receive a message by the retransmission of only four nodes (A, F, H, E) when PBSM is applied to the above figure.

The adaptive broadcasting method proposed by [VO] uses scoped flooding when the relative velocity of each node is lower than a certain threshold value. In a static network, the relative velocity of each node is almost zero and therefore scoped flooding is used. Scoped flooding is described in section 2.1.1.

Scoped flooding is applied in figure 12. Node C is the source and it transmits. Neighbors B, M and F receive the message. Node B does not transmit because the neighbors of B (i.e. C and F) are a subset of the neighbors of sender C (i.e. B, F, and M). Similarly, Node M will not transmit. Node F transmits the data message because its neighbors (i.e. B, C, M, A, G, and E) are not a subset of the neighbors of sender C. Neighbors A, E, and G of F receive the message for the first time. Each of these neighbors (i.e. A, E, G) decides to transmit as each of them has neighbors that are not neighbors of sender F. Since A has a smaller ID than E and G, A will transmit first. Neighbors H and I of A, will decide to transmit. After that, E and G will transmit because they received the message earlier than H and I. Overall, the following nodes will retransmit in this order (F, A, E, G, H, I, L). That is, all nodes retransmit except those covered by a single neighbor.

The method proposed by [CEM] is described in section 2.3.3. The protocol of [CEM] is applied in figure 12. Node C is the source and it transmits. Neighbors B, M and F receive the message. Node B transmits as it has a neighbor F that is not the sender. Similarly, Node M will transmit. Node F will also transmit as it has neighbors (i.e. B, M, A, G, E), that are not the sender C. Overall, all other nodes in the network will retransmit. The order of transmission is: (B, F, M, A, E, G, D, H, I, L, J, and K).

The proposed and existing methods are applied to the same static network. In the adaptive protocol of [VO], 7 nodes retransmit while 12 nodes retransmit in the CEM protocol. The proposed method performs better than the existing protocols of [VO] and [CEM] as only four nodes need to retransmit in PBSM. We can claim that the proposed method has the lowest number of transmissions when a static network is considered because the proposed method uses CDS and neighbor elimination. All of the protocols discussed in this section are 100% reliable in the case of a static connected network. We need simulation containing many static networks in order to verify this claim.

4.2.2 Mobile Networks

The proposed method will be compared (by example) with methods described in [VO] and [CEM] considering mobile networks in which a node can enter or leave the transmission regions of other nodes.

In [VO], a node uses hyperflooding when the average of its velocity relative to all of its neighbors, is higher than a certain threshold value. Average relative velocity depends on relative movement in a node's neighborhood. Depending on the average relative velocity, some nodes may use hyperflooding or plain flooding whereas others may use scoped flooding. The method of [VO] is described in section 2.4.

Example 1

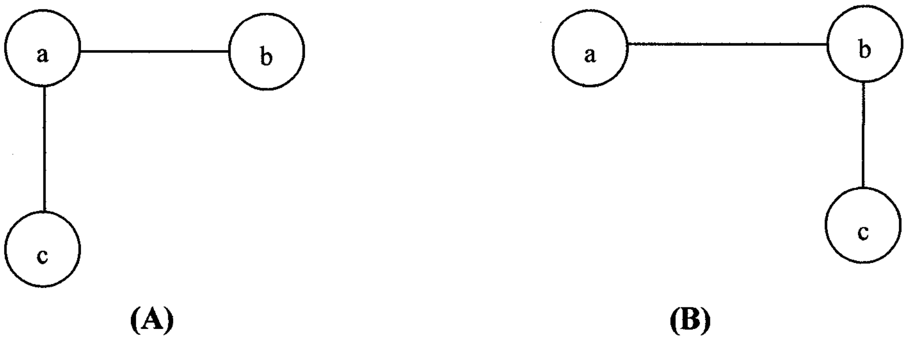


Figure 13: Reducing the number of transmission using the R list

PBSM is applied in Figure 13A. Suppose that the node *a* in Figure 13A transmits the data (broadcast) message. When node *b* receives the message, it includes the sender (node *a*) and its neighbor (node *c*) in node *b*'s R list. Similarly, when node *c* receives the message, it includes the sender (node *a*) and its neighbor (node *b*) in its R list.

Consider Figure 13B. Suppose that node *c* of Figure 13A moves into the vicinity of node *b*. During the hello message round, node *c* and node *b* discover each other as new neighbors. However, node *c* contains node *b* in its R list and vice-versa. Therefore, none of the node (*b* or *c*) will transmit data message in PBSM.

In [CEM] both node *b* and node *c* will transmit a broadcast message when they discover each other. Node *c* moves with a certain velocity and this velocity is sent as a part of hello message in the adaptive protocol of [VO]. If node *c* moves with such a high velocity that the relative velocity between *b* and *c* become greater than threshold of both *b* and *c*, then node *c* and its neighbor *b* will be in hyperflooding mode. Considering the protocol of [VO], both node *b* and node *c* will transmit when they discover each other during hello message. Thus, the proposed method reduces unnecessary transmissions. Reliability is preserved as node *c* initially receives the data message from node *a*.

Example 2

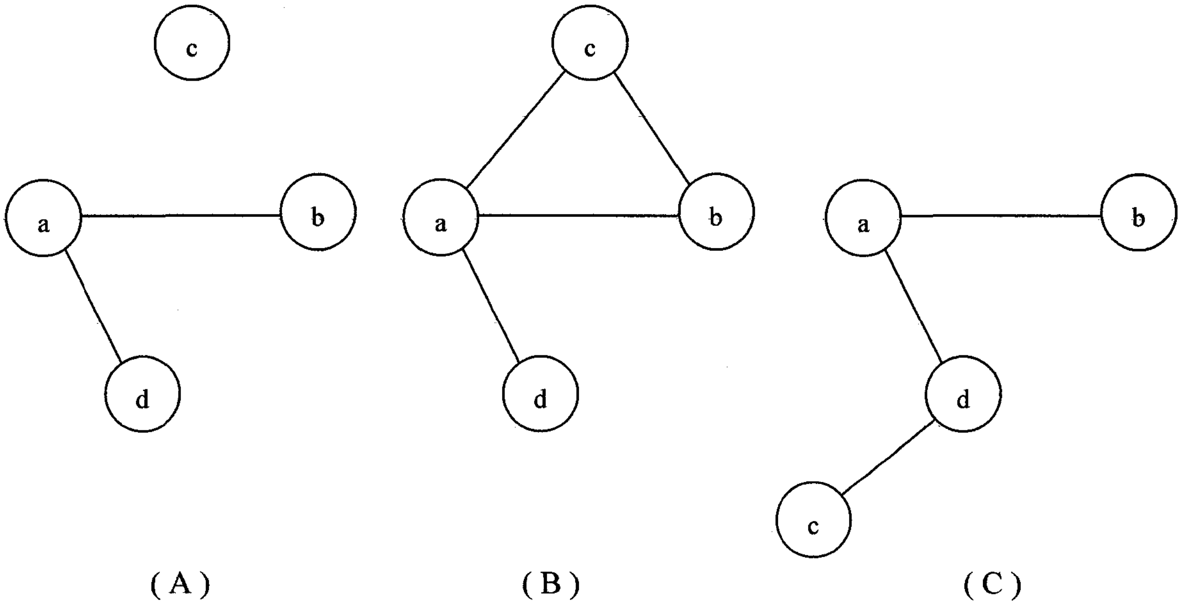


Figure 14: Reducing the number of transmissions using neighbor elimination

PBSM is applied in Figure 14B. Node *c* moves into the vicinity of both node *a* and node *b*. Node *c* does not have the data message and node *a* and node *b* have already received it. As both node *a* and node *b* discover the new node *c* during hello message, they will both decide to broadcast and run a timeout. Suppose that node *a*'s timeout expires first and it transmits. Nodes *b* and *d* will receive the transmitted data message. Node *b* will add node *c* to its R list and remove node *c* from its N list as node *c* is a neighbor of the sender (node *a*). Although node *b* decides to transmit at first, it will not transmit because its N list is now empty. This is an application of NES [PL, SSZ].

If node *c* moves with a high velocity such that relative velocity between *c* and *a,* and *c* and *b* is greater than threshold of both *a* and *b*, then nodes *a* and *b* will be in hyperflooding mode. Therefore, in the methods of [CEM, VO], both node *a* and node *b* will transmit when they discover the new node *c*. Thus, the proposed method reduces unnecessary transmission. Reliability is preserved as the new node *c* receives the data message from node *a*.

Consider Figure 14C. Node *c* now moves closer to node *d*. Now, although node *c* is more than 2 hops away from *b*, node *b* has node *c* in its R list because at some point node *c* moved closer to *b*. Therefore, the R list of node *x* also contains nodes that are now more than 2 hops away but at some point moved closer to *x*. After some time, if node *c* (currently more than 2 hops away) moves closer to *b*, *b* will not transmit as it already contains *c* in its R list from a previous encounter (see Figure 14B). This reduces the number of transmissions.

Example 3

PBSM is applied in Figure 14B. Node *c* moves into the vicinity of node *a* and node *b*. Nodes *b, d* and *c* do not have the message and node *a* has already received the data message. Node *a* and node *b* are not aware of node *c* as hello message has not yet take place. Suppose that node *a* transmits. Nodes *c*, *b* and *d* will receive the transmitted data message. Nodes *b* and *d* will add each other and node *a* to their respective R lists and then remove node *a* from their respective N lists. Note that node *c* is not added to the R list of *b* or *d* as node *a*'s neighbor list doesn't contain *c*. So, *c* is not added to *a*'s R list. Nodes *b* and *d* will not transmit as their N lists are now empty. Node *c* will add only node *a* to its R list because *c* doesn't know the neighbor list of new node *a*. Node *c*'s N and neighbor list are empty since it has arrived from a disconnected network. As a result, node *c* will not transmit.

In the hyperflooding-based methods of [VO], node *c* (considering that *c* is in hyper-mode as it is moving with high velocity) will transmit when it receives a message from node *a*, which is not in its current neighbor list. Thus, PBSM reduces unnecessary transmissions. Reliability is also preserved because new node *c* receives the data message from node *a*.

Example 4

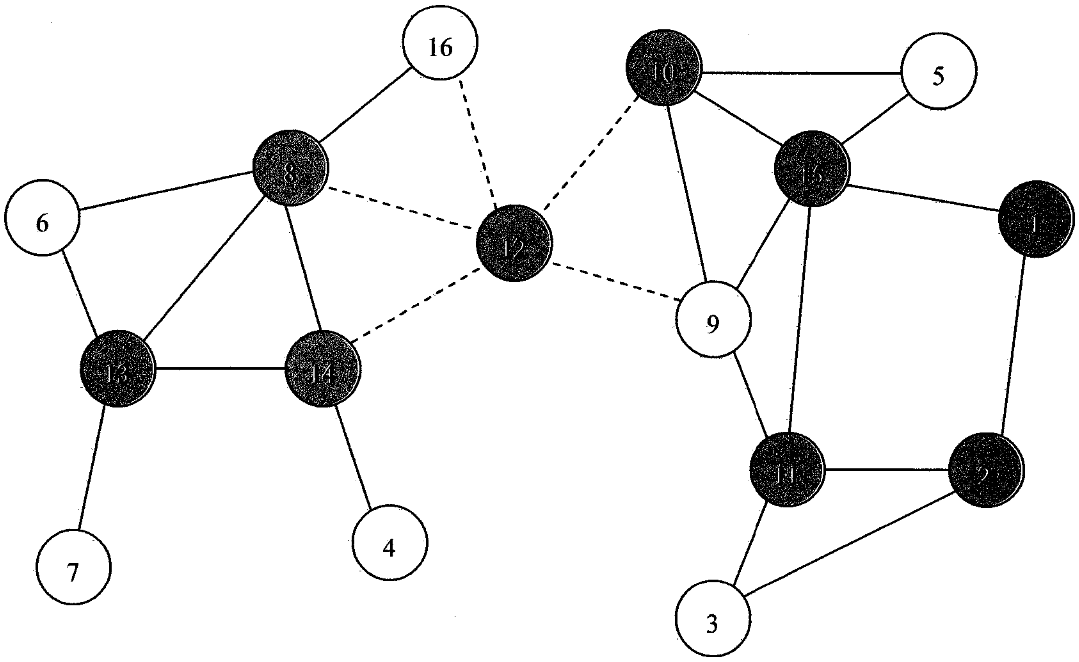


Figure 15: Reducing the number of transmissions using the periodic calculation of CDS

As discussed in section 3.2.2, the following nodes will transmit when PBSM is applied to the above figure.

Transmission	Node-Timeout
8	**12** - ½; **14** - cancels timeout; 16 - cancels timeout
12	9-(5+½); **10** - ½
10	9-(5+1/1-½);**15** - ½
15	**1** - 1/1; 9 - cancels timeout ;**11** – ½
11	**1** – (1/1-1/2); **2** – 1/1
1	**2** – cancels timeout

Table 3: Transmission order and timeout of PBSM

The method proposed by [VO] is applied to Figure 15 after node 12, which has not received the message, arrives closer to other nodes. All the nodes on the left side of node 12 have already received the message, whereas the nodes on the right side of node 12 have not received the message. Hello message takes place and node 12 is discovered by

its neighbors. Nodes 8, 9, 10, 12, 14, and 16 are most likely to be in hyperflooding mode since there is movement in their neighborhood (i.e. relative velocity > threshold). The remaining nodes are most likely to be in scoped mode. Nodes 8, 14 and 16 transmit since they have the message. Node 12 receives the message for the first time and it transmits. Then nodes 9 and 10 transmit when they receive the message from 12. Nodes 11 and 15 are in scoped mode and they receive the message for the first time from 9. The neighbors of node 11 are not a subset of sender 9, therefore node 11 transmits. Similarly, node 15 transmits. Node 2 is in scoped mode and its neighbors are not a subset of sender 11; therefore, node 2 transmits. Similarly, node 1 transmits. The order of transmission is:
8, 14, 16, 12, 9, 10, 11, 15, 1, 2.

The method proposed by [CEM] is applied to Figure 15 after node 12, which has not yet received the message, arrives closer to other nodes. All the nodes on the left side of node 12 have already received the message, whereas the nodes on the right side of node 12 have not received the message. Hello message takes place, and node 12 is discovered by its neighbors. Nodes 8, 14, and 16 transmit since they have the message. Node 12 receives the message for the first time and transmits since it has neighbors other than the sender. Nodes 9 and 10 receive the message for the first time from 12. Node 9 transmits since it has neighbors other than the sender 12. Similarly, node 10 transmits. The order of transmission is: 8, 14, 16, 12, 9, 10, 11, 15, 5, 2, 3, and 1.

The proposed method produces the lowest number of transmissions since it uses CDS and only CDS nodes transmitted. Due to neighbor elimination, the CDS node 2 did not transmit. Therefore, the recalculation of CDS after every hello message is justified. Every node receives the message in all the methods described.

Example 5

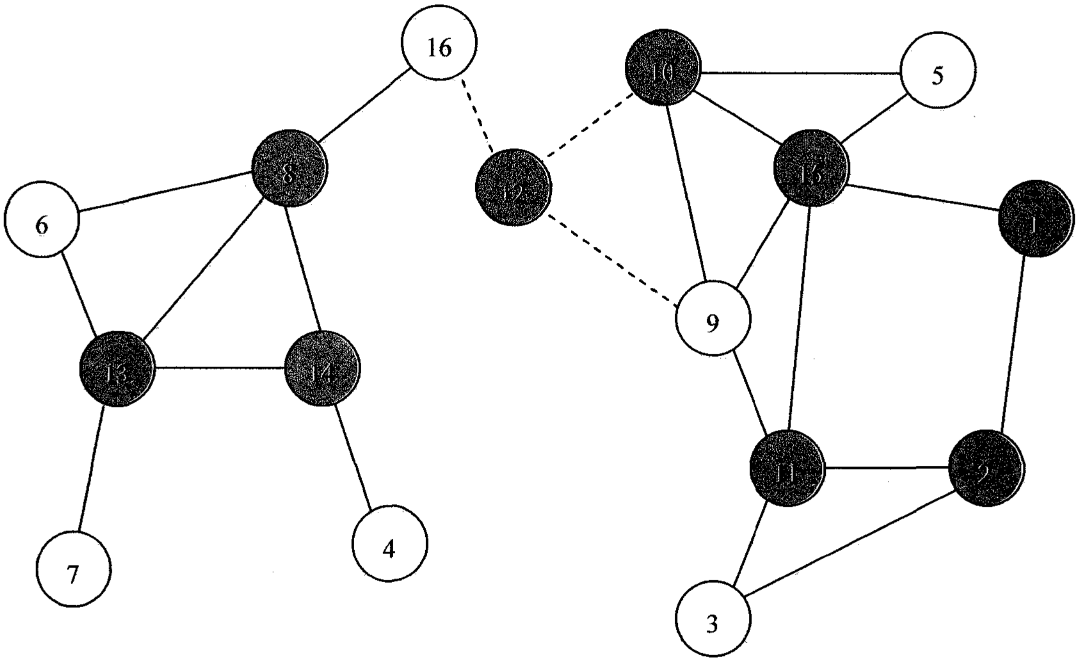


Figure 16: Increasing reliability when a node joins two disconnected network

The method of [VO] is applied to the above figure, where the network is initially partitioned into two components before the arrival of node 12. All nodes on the left side of node 12 have already received the message from node 14 (the source), whereas the nodes on the right side of node 12 have not yet received message. After one round of 'hello' messages, node 12, which has not yet received any message, is discovered as a new neighbor of nodes 16, 9, and 10, making the network connected. Nodes send their velocity as part of the hello message. Since node 12 is moving slowly, the average relative velocity of the neighbors of node 16 are below its threshold value. Thus, node 16 is either in scoped or plain mode and as a result does not transmit when it discovers node 12. Consequently, node 12 and all of the nodes on the right side of node 12 will not receive any message.

On the contrary, in PBSM, node 16 will transmit when it discovers node 12 as its N list is non-empty (i.e. contains node 12). On receiving the message from 16, node 12 will transmit since its N list is non-empty (i.e. contains 9 and 10). Therefore, the nodes on the right side of node 12 will receive the message. Thus, PBSM shows a greater reliability than [VO] in this case. The method of [CEM] showed an equal degree of reliability to PBSM but it generates more transmissions.

Results of comparison in mobile networks

We have presented some scenarios that are very likely to take place in a mobile network. We can claim that the proposed method has the lowest number of transmissions when a mobile network is considered. All of the protocols discussed in this section show an equal degree of reliability. However, in some cases, PBSM shows greater reliability than [VO]. The above claim is established by considering a few particular situations. We need simulation containing many mobile networks with many scenarios in order to verify this claim.

4.3 Examples showing the limitations of PBSM

4.3.1 Reducing reliability by avoiding transmissions when data is received from a new node

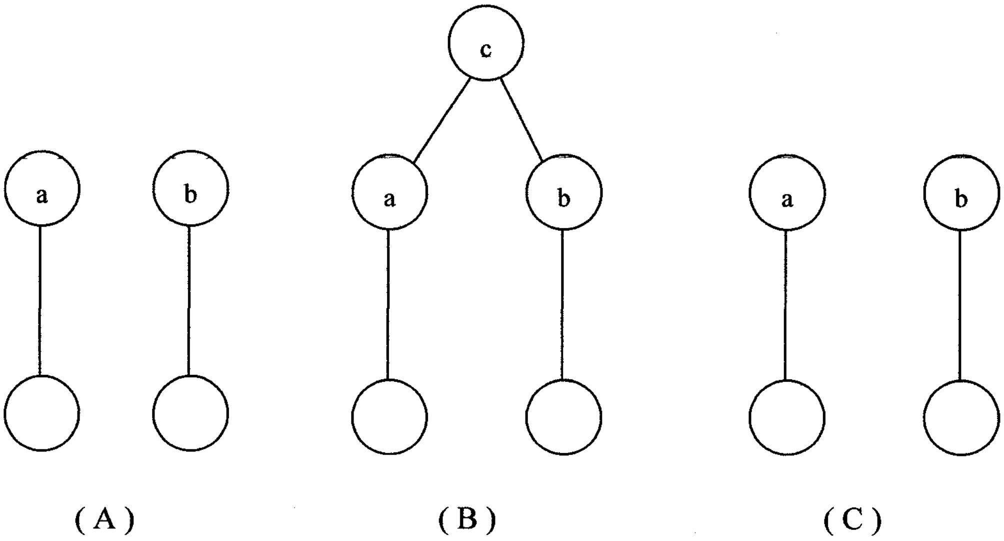


Figure 17: Reducing reliability by avoiding transmission when data is received from a new node

Consider Figure. 17A where a network contains two disconnected components. Node *a* receives the message whereas node *b* does not. In Figure 17B, node *c*, which has not received any message, comes into the transmission region of node *a* and node *b*. Nodes *a* and *b* are not aware of node *c* and vice versa since hello message has not taken place. Suppose that it is time for node *a* to transmit (normal transmission). Node *c* receives the message. In hyperflooding mode [VO], node *c* (considering that *c* is in hyper mode as it moves with high velocity) will immediately retransmit the data message as it receives the message from new node *a,* which is not in *c*'s neighbor list. Node *b* will receive the data message from *c*. Node *c* then moves away from the other nodes (as shown in Figure 17C) before the next hello message time. In the proposed method, when *c* receives the data message from the new node *a* (which is not in *c*'s neighbor list), it does not retransmit as *c*'s N list is empty (arrived from a disconnected network). It waits till the next hello message time, and if it discovers the new node, then it will transmit. However, *c* moves away from the network before the hello message time. Node *b* will not receive the data message because *c* does not transmit. Thus, the reliability of PBSM becomes lower than the methods of [VO] in this case. This situation seldom arises if the hello message time is not very long. Therefore, it is unlikely that a node will enter and leave a new transmission region in between two hello messages unless the node is moving at a very high speed.

To overcome this limitation, PBSM could be modified to allow a node to immediately retransmit whenever it receives a message from a new node that is not in its neighbor list. However, this would generate additional transmissions.

4.3.2 Reducing reliability by using the R list

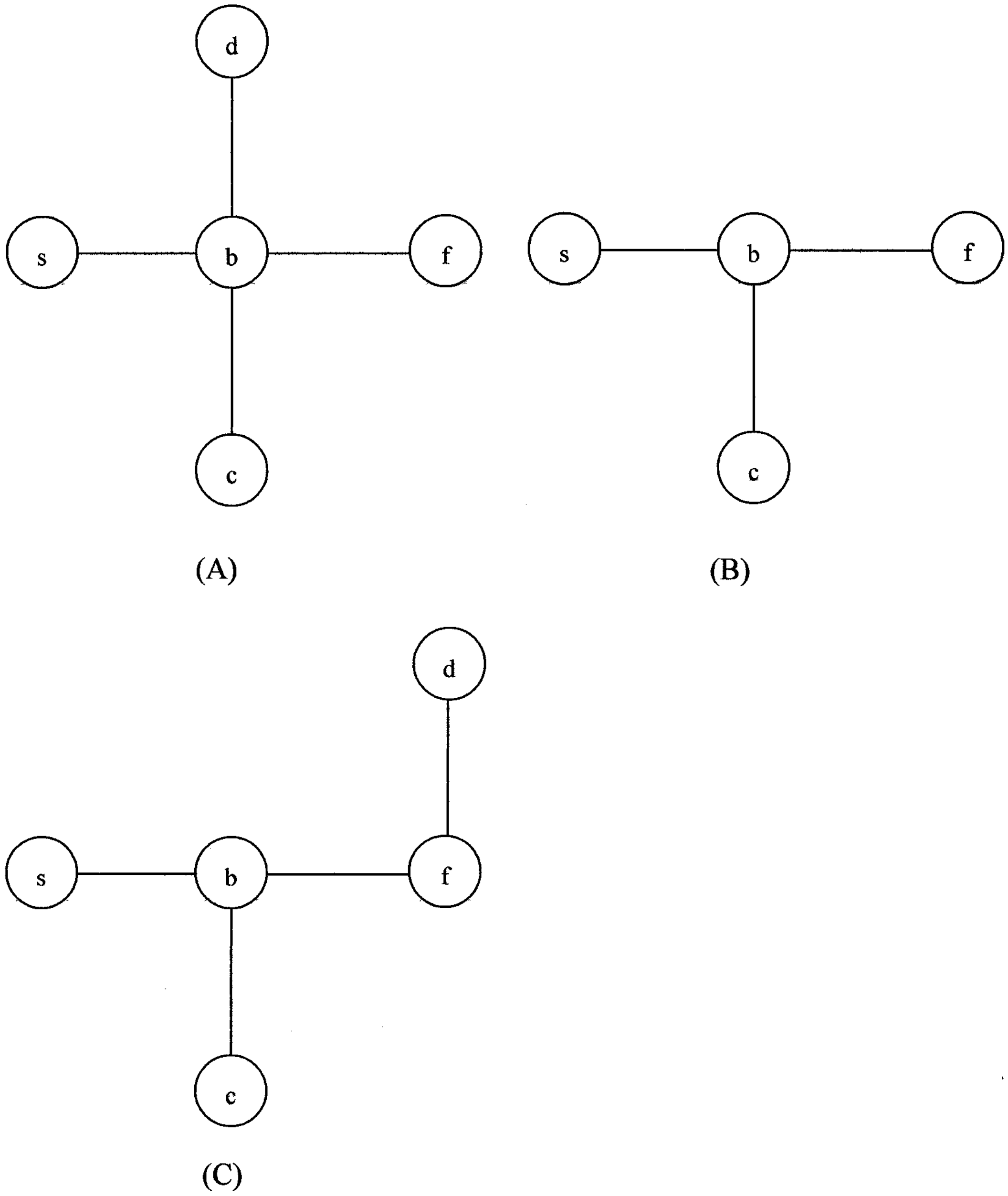


Figure 18: Reducing reliability by using the R list

Consider Figure 18A. Suppose that node *s* transmits and node *b* receives the message. Now node *d* moves away as shown in Figure 18B. Since the next hello message time has not yet arrived, node *b* still has *d* in its neighbor list. Now *b* transmits (*d* does not receive the message since it has moved away). Node *f* receives the message for the first time and includes *b* and its neighbor *c, d, s* (since neighbor list of *b* not updated) in *f*'s R list. Then, node *d* moves closer to *f* as shown in Figure 18C. Hello message takes place and *f* discovers *d*. Node *f* will not transmit since it has *d* in its R list. Thus, *d* will not receive the message. In other methods [VO, CEM], node *f* (considering that *f* is in hyper mode if node *d* moves with high velocity) will transmit when it discovers node *d* and node *d* will receive the message. Thus, the reliability of PBSM becomes less than other existing methods in this case.

To overcome this limitation, PBSM could be modified in the following way. Nodes could also include their geographic position, current speed, and direction of movement in hello messages. Each node can then estimate, for each neighbor, how long the current link will last. This estimation is possible since a node can predict its neighbor's position in subsequent times based on its current velocity and position [SLG]. Therefore, if a node estimates that a link with its neighbor is broken when a broadcasting message arrives, it will update its neighbor list and send a hello message. As shown in Figure 18A, *b* knows the velocity and position of *d* at hello message time. Then when *b* receives the message (Figure 18B), it can estimate that *d* is no longer its neighbor. At this point, *b* updates its neighbor list and sends a hello message to its neighbors. After that, *b* transmits and *f* receives the message. Node *f* includes *b* and its neighbor *c* and *s* (since *f* knows the updated neighbor list of *b*) in *f*'s R list. As shown in Figure 18C, *f* transmits when *d* is discovered as *d* is not in *f*'s R list. Node *d* receives the message. However, there is increased overhead in estimating link changes, velocity and obtaining position information.

4.3.3 Increasing the number of transmissions when a new node receives a message

Consider Figure. 17A. Node *a* receives the message whereas node *b* does not. In Figure 17B, node *c*, which has not received any message, comes into the transmission region of node *a*. Node *a* is not aware of node *c* and vice versa as no hello message has taken place. Suppose that it is time for node *a* to transmit (normal transmission). Node *a* does not include node *c* in its R list as *a* does not contain *c* in its neighbor list. Node *c* receives the message and it includes *a* in its R list. Hello message takes place and nodes *a* and *c* discover each other. Node *a* will transmit as it does not have *c* in its R list, whereas node *c* will not transmit as it has *a* in its R list. In the methods of [VO, CEM] both nodes *a* and *b* will transmit. Although PBSM performs better than the methods of [VO, CEM], there is an unnecessary transmission by node *a*.

To overcome this limitation, PBSM could be modified in the following way. When a node receives a data message from a node that is not in its current neighbor list, it will immediately send a hello message. If a node receives a hello message that is not periodic, and if the hello message is received shortly after transmitting, it will include the new node in its R list. As shown in Figure 17B, when node *c* receives the data message from node *a* that is not in its current neighbor list, it sends a hello message immediately. Node *a* receives a hello message, and since the hello message is not periodic and is received shortly after transmitting, *a* will include the new node *c* in its R list. Thus, the additional transmission by node *a* when it discovers *c* can be avoided. The drawback is that additional, non-periodic hello messages will be generated.

4.3.4 Other minor limitations of the protocol

- The R list of a node usually contains nodes that have received messages up to 2-hops. Therefore, if a node discovers a node that has received a message but was more than 2 hops away from it, it will transmit. This transmission is unnecessary.
- There may be an overhead due to the calculation of the CDS after every hello message. However, no additional message exchange is needed and overhead can thus be ignored.
- The maintenance of extra R and N lists by each node may have some overhead, especially if the R list becomes large.
- The second round of hello messages to gain 2-hop information does have some overhead in message size as the neighbor list is exchanged.

Chapter 5

Simulation

After performing an extensive literature review, two existing protocols have been selected for comparison. The protocols of [VO] and [CEM] are selected for comparison as they consider similar assumptions and are developed for highly mobile scenarios. Although they have some limitations, they perform better than many of the other available protocols. Therefore, they are considered to be valid competitors.

Another method, VO-CDS, is introduced. This method is the same as VO except that scoped flooding is replaced by following CDS-based flooding. In this type of flooding, when a node receives a message for first time, it starts a timeout if it is in the CDS and if it has a neighbor that is not a neighbor of the sender. Nodes that are not in the CDS select a longer timeout. The advantage of CDS-based flooding is that it reduces retransmission. However, it is unable to react when a new neighbor is discovered. This method is introduced so that we will know each time what the contribution of CDS is over scoped flooding, and what is the impact of using thresholds. The proposed method, the protocol of [VO and CEM], and VO-CDS are implemented. Each implementation is simulated in various mobility scenarios, changing parameters as required. The program was developed in both Visual C++ and Turbo C++.

5.1 Assumptions

The simulated environment is developed based on the following assumptions:

- A unit disk graph.
- All nodes have an equal transmission radius that is fixed.
- An ideal medium access (MAC) protocol with no collisions between messages is assumed in first part of simulation (sec 5.6) whereas simplified MAC protocol with collision is assumed in second part of simulation (sec 5.7).
- A one-to-all model for broadcasting is assumed. That is, when a node transmits, all of its neighbors will receive the message.
- There is a single message.
- There are no obstacles between any two nodes. A message is received if and only if the distance between the sender and receiver is less than transmission radius.
- There are no acknowledgements for any received message.
- Each node has a unique ID that is used to identify the node.

5.2 Environment

To generate the random unit disk graph, values for *d* and *n* are chosen. *d* represents the desired average node degree of the graph and *n* is the number of nodes. All nodes are randomly generated in a square of side *a*. That is, its *x*- and *y*- coordinates are chosen at random between 0 and *a*. We chose *a*=470 as the computer's viewport is 640 x 480. Next, the initial radius is calculated using the formula $r^2 = (d * a * a)/ ((n-1) * \pi)$. The generated graph is disconnected most of the time. A total of ten simulations are used to gather experimental data. Each simulation consists of a randomly generated graph. Each simulation consists of ten hello messages. Hello messages take place at fixed intervals.

5.3 Timing and Synchronization

Sending a broadcast message to an immediate neighbor requires fixed time *B'*. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit (i.e., *B'* = 1 unit or 1 sec). *H* is the fixed time between two hello messages. *B* is the number of messages that can be sent between two hello messages. That is, *B* = *H/B'*. Time is incremented by 1 during each transmission. A global variable *timer* is used to keep track of time. The node with the highest degree is selected as a source. Broadcasting starts randomly between two hello messages. The *timer* is set to 0 when the transmission starts from the source. In each time unit, a broadcasting method is called to perform a single transmission (if any) and then returns to the main program. The broadcasting methods store the processing data in a global variable before returning to the main program. This ensures that in the next time unit, when the method is called again, it can continue from the point where it stopped earlier. *T*, the number of hello messages, is 10. As hello messages take place periodically, it takes *10H* units of time to complete ten hello messages. Therefore, a node remembers a message for *10H* units of time. That is, a node will remember the message throughout the broadcasting process.

5.4 Mobility

The movement of nodes takes place in each time unit. Movement takes place either while broadcasting is in progress or when there is no broadcast. The nodes move in a way similar to the random waypoint mobility model [BMJ]. The co-ordinates of their destinations are chosen at random. A node move to its destination in straight line at random speeds between 0 and the maximum speed. Upon reaching the destination, the node pauses for some time (0 sec) and then select new random destination. A node moves to its new destination with different speed between 0 and the maximum speed. Under the maximal speed, a node can travel distance *D'* between two hello messages. Let *D'=D*r,* where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is *D*r /H* (*=D'/H*). The average velocity is maximum speed/2. Consequently, speed and *D'* depend on *D* and *D* is used as a parameter to define mobility. A movement can change the graph, and a broadcasting method acts with the changed graph. However, a node is not aware of the change until the next hello message time. Thus, some nodes may receive messages although their sender does not know about them (since the nodes have just arrived in the neighborhood), while some believed neighbor nodes may be gone.

Three types of mobility scenarios, namely static, moderate and high mobility are simulated.

- In the static mobility scenario, all nodes are static. That is, the graph does not change over time (D=0, so max speed = 0).
- In a moderate mobility scenario, some nodes move with moderate speed, whereas some nodes are static. [D=1/2 , so max speed = $r/ (2H)$].
- In the high mobility scenario, most of the nodes move. Some nodes are static or move with moderate speed, whereas some nodes move quickly. [D=1, so max speed = r/ H].

Therefore, the classification between moderately and highly mobile nodes depends on node velocity. Roughly speaking, the distinction between moderate and high mobility is based on the percentage of neighbor changes of a node while broadcasting is in progress. In *static* networks, nodes move very slowly so that they are assumed to be constant. Network topology does not change during the broadcasting task. A network is moderately mobile if all nodes in it are either static or moderately mobile. Thus, even a single moderately mobile node among many static nodes classifies the network as moderately mobile. Finally, a network is highly mobile if at least one node in it is highly mobile. Therefore, a few high-speed vehicles among a lot of pedestrians along the road can classify the network as highly mobile.

5.5 Order of transmission of existing protocols

In the protocols proposed by [VO] and [CEM], it is stated that all nodes delay for a random back-off time before transmitting. This is done to avoid simultaneous transmissions from several nodes. It is not mentioned how the back-off time is calculated and what will be the value of the back-off time of one node compared to another node. Therefore, some innovations are made in this regard and are applied to [VO and CEM] for uniformity. All the neighbors of a sending node will receive a message at the same time, and these nodes consider transmission as soon as they receive the message. All of the neighbors of a sender that decides to transmit initialize back-off time simultaneously. In first part of simulation (sec 5.6), the back-off time is a function of node ID (proportional to ID). That is, nodes with smaller IDs will have smaller back-off times than nodes with higher IDs. This is done so that there is a pattern. This pattern is followed by all existing methods so that there is uniformity and thus the comparison is fair. Furthermore, the back-off time of a node that receives the message earlier is always shorter than the back-off time of a node that receives the message later (regardless of ID). This can be achieved by making back-off time a function of time (proportional to time). Back-off time is relatively short compared to transmission time. Therefore, the flooding techniques of [VO, CEM] can be implemented using the concept of Breadth-First Search (BFS). That is, messages are sent first to the neighbors of the source. Each of these neighbors that decide to transmit is put into a queue. The node with the smallest ID is put into the queue first. The queue is First-In-First-Out (FIFO) and the neighbor with the smallest ID will be de-queued and it will transmit first. Each neighbor of the source then transmits the message to their neighbors and so on. The queue consists of nodes that are waiting to transmit. In the second part of simulation (sec 5.7), the back-off time of existing method is random. The timeout of proposed method is also random instead of 1/|N|.

5.6 Performance Evaluation using Ideal MAC layer

The methods of [VO and CEM], PBSM and VO-CDS are implemented and comparisons are done by simulation. An ideal MAC layer, with no collision between messages, is assumed. The simulation contains three main scenarios: static, moderate, and high mobility. Moreover, the performance of the protocols is also compared based on D (a parameter used to define the speed of movement), density (d) (a parameter that indicates the average node degree of graph) and B (the number of messages that can be sent between two hello messages. That is, $B = H/B'$ where $B' = 1$ and H is hello message time). Each scenario consists of ten different simulations. In order to obtain a smooth curve, an average of 10 or 20 run (where all parameters are constant) is used to gather each data. Each simulation consists of a randomly generated graph and ten hello messages. In each run (simulation), the hello message takes place first. After that, all the broadcasting methods are called. Broadcasting starts at time 0 with the transmission from the source. Hello messages take place at fixed intervals and movements can take place at any time. For each simulation and for each type of flooding, the following metrics are calculated in order to evaluate the method's performances.

$$\text{--}\textit{Total number of transmissions per node} = \frac{\text{total number of transmission by all node}}{\text{number of node}}$$

$$\text{--}\textit{\% Reliability} = \frac{\text{number of node that receive message}}{\text{total number of nodes that could have received message from source}} \text{ X } 100$$

The following method is applied to identify nodes that might have received the message from the source. Time is divided into time units (slots). Initially, the source node is colored black, and all the other nodes are colored white. In each time unit (slot), each node is checked to see whether it has at least one neighbor that was colored black in an earlier time slot. If it has at least one such neighbor, then the node is colored black. Recall that nodes can move anytime. When the simulation ends after a certain time, black nodes are exactly those that could have received the message from the source node. Then the reliability of a particular protocol is the percentage of black nodes that have received the message by the protocol. These nodes are used for the calculation of reliability. This gives more accurate results for reliability since it is impossible for nodes that are always disconnected to receive a message and therefore they are not considered. Moreover, it also considers nodes that may not be connected to the source at any given moment in time but could receive a message from the source. For instance, if another node moves between the areas where a source and destination node is located and carries the message.

5.6.1 Static Network

In a static mobility scenario all nodes are static. That is, there is no movement and the graph does not change over time (*D*=0, so max speed = *0*). In each simulation, the number of nodes is 25 and the average density is close to 4. *H* (fixed time between two hello messages) is 10 in each simulation. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit. Therefore, B (the number of messages that can be sent between two hello messages) is equal to *H*. Each simulation consists of a randomly generated graph and ten hello messages. The following graph is used in the first simulation:

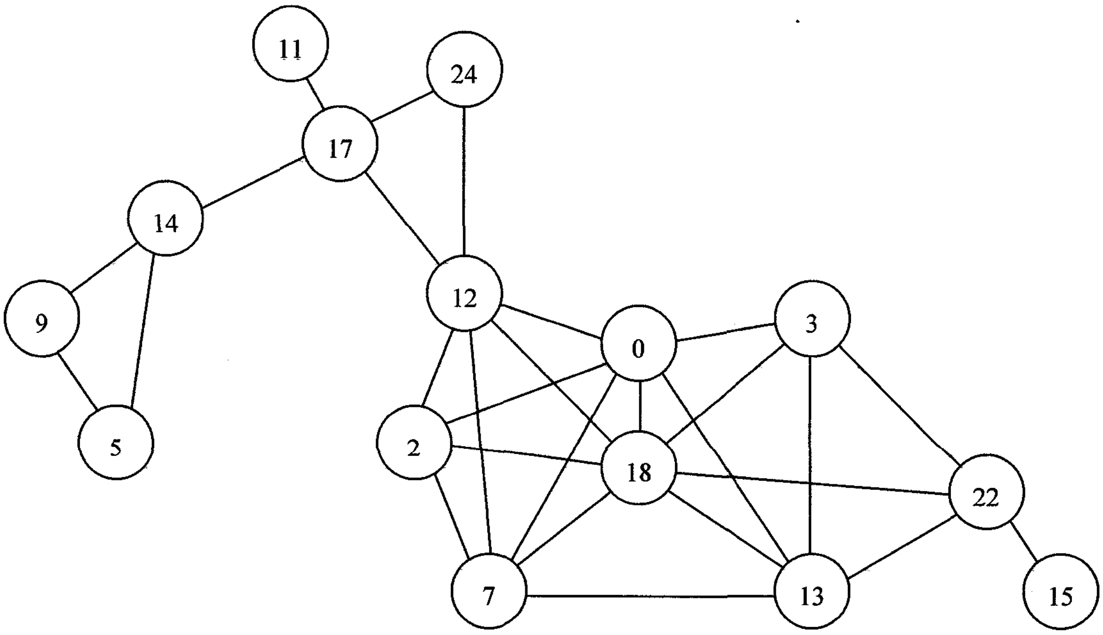
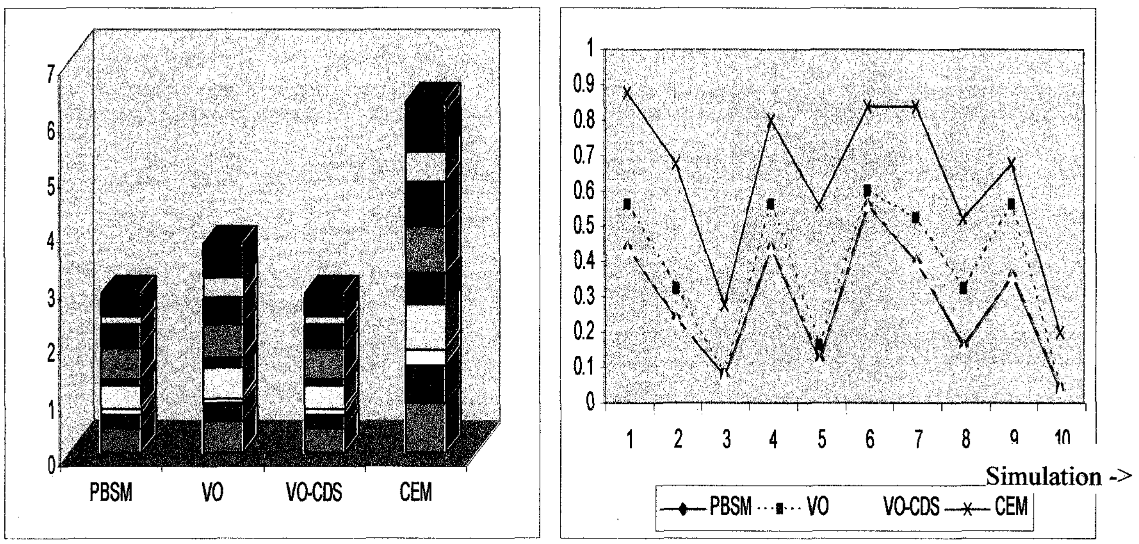


Figure 19: Static Graph in the first simulation (disconnected nodes are not shown)

Total number of transmissions per node



(A) (B)

Figure 20: Total number of transmissions per node in a static scenario (D=0, B=10, n=25, d=4)

	PBSM	VO	VO-CDS	CEM
Total	2.84	3.72	2.84	6.28
Average	0.284	0.372	0.284	0.628

Table 4: Summary of total number of transmissions per node in a static scenario

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than the methods of [VO] and [CEM]. There are two main reasons for this. First, PBSM uses CDS, and in a static network the number of transmissions is at most equal to the number of CDS nodes. All nodes in the network receive the message by transmissions from only CDS nodes, and therefore non-CDS nodes (which run a longer timeout) do not need to transmit. In the methods of [VO] and [CEM], every node considers retransmission when it receives a message for the first time. Second, as PBSM uses neighbor elimination, a node that initially decides to transmit might not transmit if its N list becomes empty. The performance of PBSM and VO-CDS is the same as both methods use CDS-based flooding in static networks.

The number of transmissions per node in PBSM is around 26% less than the method of [VO]. The methods of [VO] uses scoped flooding in static networks. Scoped flooding is based on neighbor elimination and therefore the performance of the adaptive protocol [VO] is close to that of PBSM. The method of [CEM] generates the highest number of transmissions since a node decides to transmit when it has any neighbor other than the sending node.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the number of simulations. The data tables are below:

	PBSM	**VO**	**VO-CDS**	**CEM**	Simulation
	0.44	0.56	0.44	0.88	1
	0.24	0.32	0.24	0.68	2
	0.08	0.08	0.08	0.28	3
	0.44	0.56	0.44	0.8	4
	0.12	0.16	0.12	0.56	5
	0.56	0.6	0.56	0.84	6
	0.4	0.52	0.4	0.84	7
	0.16	0.32	0.16	0.52	8
	0.36	0.56	0.36	0.68	9
	0.04	0.04	0.04	0.2	10
Total	2.84	3.72	2.84	6.28	
Average	0.284	0.372	0.284	0.628	

Table 5: Total Number of Transmissions per node in a static network (D=0, B=10, n=25, d=4)

	PBSM	VO	VO-CDS	CEM	Simulation
	11	14	11	22	1
	6	8	6	17	2
	2	2	2	7	3
	11	14	11	20	4
	3	4	3	14	5
	14	15	14	21	6
	10	13	10	21	7
	4	8	4	13	8
	9	14	9	17	9
	1	1	1	5	10
Total	71	93	71	157	
Average	7.1	9.3	7.1	15.7	

Table 6: Total number of transmissions in a static network (D=0, B=10, n=25, d=4)

Percentage of reliability

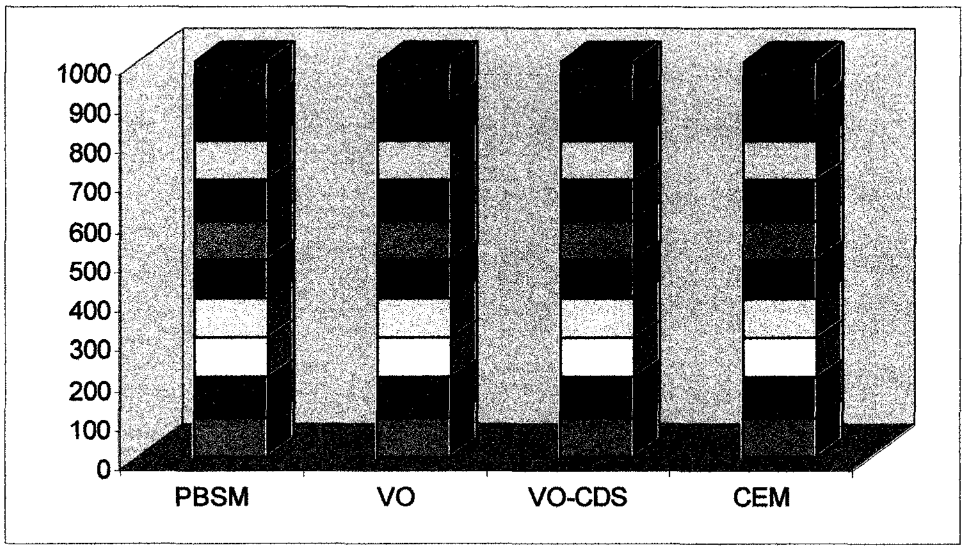


Figure 21: Percentage of reliability for a static network (D=0, B=10, n=25, d=4)

	PBSM	VO	VO-CDS	CEM
Total	1000	1000	1000	1000
Average	100	100	100	100

Table 7: Summary of percentage of reliability for a static network

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. In a static network, all protocols are 100% reliable. In each simulation, all of the nodes received the message for all protocols. This is depicted in the bar graph.

In the proposed PBSM method, the CDS nodes are selected in such a way that if only the CDS nodes transmit, all nodes in the network will receive message. The method of [VO] uses scoped flooding and VO-CDS uses CDS-based flooding in a static network.

5.6.2 Network of Moderate Mobility

Under maximal speed, a node can travel distance *D'* between two hello messages. Let *D'=D*r,* where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is *D*r /H* (=*D'/H*). Consequently, speed and *D'* depend on *D* and *D* is used as a parameter to define mobility.

In a moderate mobility scenario, half of the nodes move with moderate speed [*D*=1/2, so max speed = *r*/ (2*H*)]. The remaining nodes are static. In each simulation, the number of nodes (n) is 25 and the average density (d) is close to 4. The hello message time (*H*) is incremented by 2 in each simulation. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit. Therefore, B (the number of messages that can be sent between two hello messages) is equal to *H*. Each simulation consists of a randomly generated graph and ten hello messages. The graph of the first simulation is as follows:

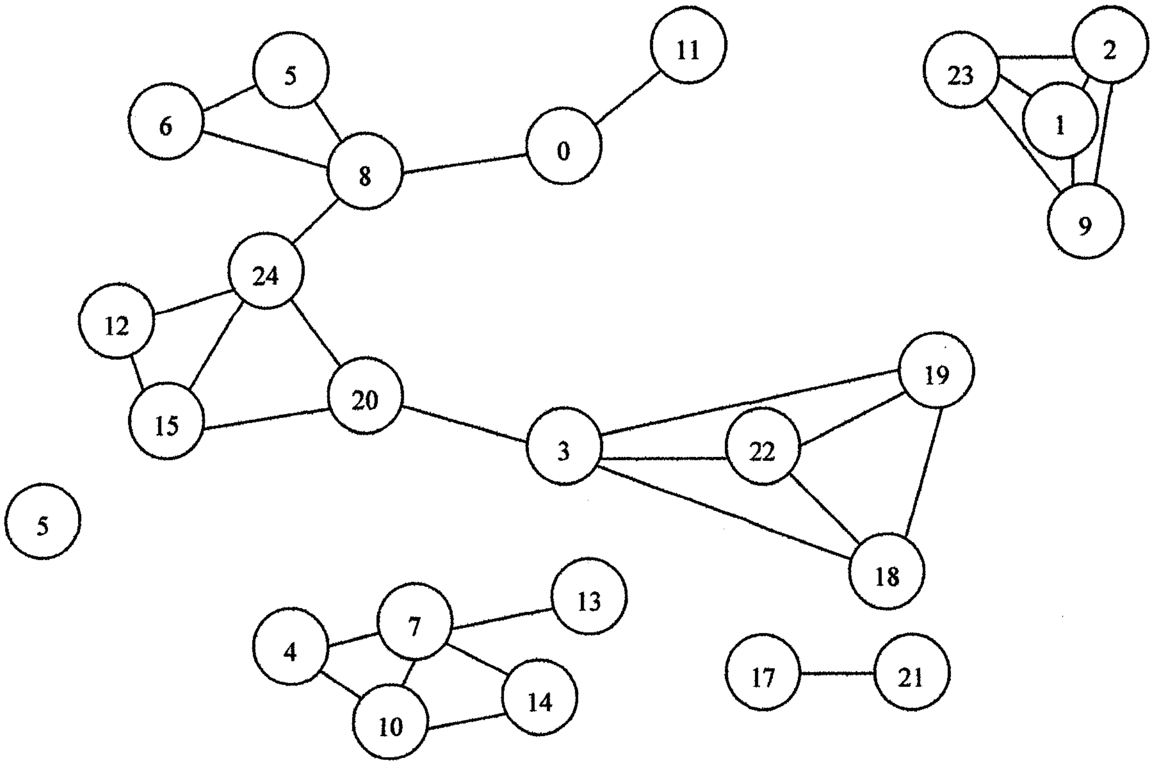
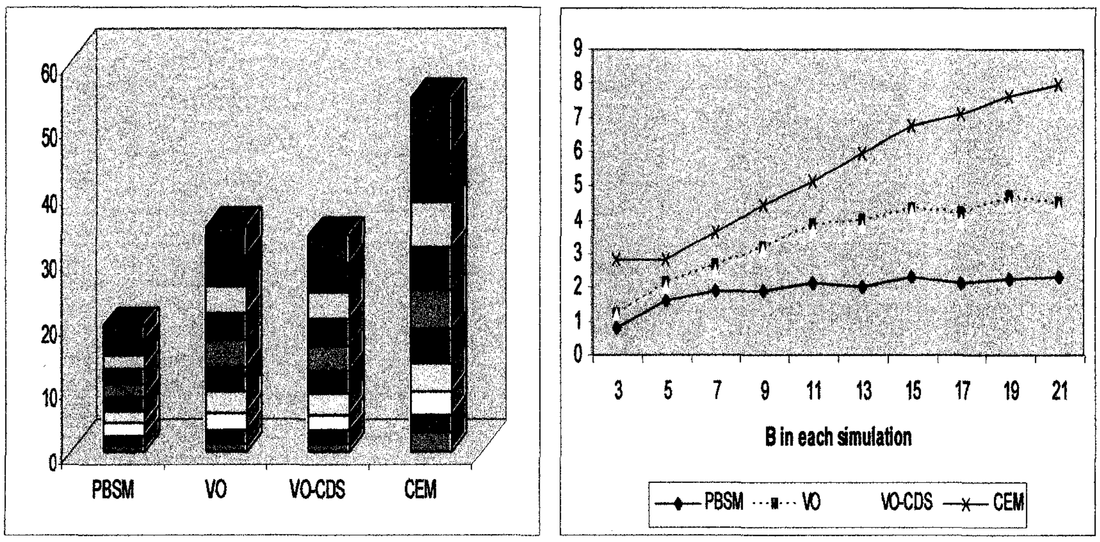


Figure 22: Graph of the first simulation of moderate mobility (nodes 0-12 are mobile)

Total number of transmissions per node



(A) (B)

Figure 23: Total number of transmissions per node in a moderately mobile network (D=1/2, n=25, d=4)

	PBSM	VO	VO-CDS	CEM
Total	19.4	34.76	33.4	54.28
Average	1.94	3.476	3.34	5.428

Table 8: Summary of total no. of transmissions per node in a moderately mobile network

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons for this difference. First, most nodes in the network receive the message via the transmissions of only CDS nodes, and therefore few non-CDS nodes (which run a longer timeout) need to transmit. In the methods of [VO] and [CEM], every node considers retransmission when it receives a message for the first time. Second, as PBSM uses neighbor elimination, a node that initially decides to transmit might not transmit if its N list becomes empty. Third, the proposed method does not consider broadcasting each time two nodes discover each other as new neighbors, which other approaches do. Each node maintains a receiver list (R) containing nodes that have received the message (up to 2-hops). When a node discovers a new neighbor, it checks whether the new node is in R. If it is not, then the node considers broadcasting. Otherwise, it does not broadcast. Finally, unlike the hyperflooding-based methods of [VO], there is no additional transmission when a node receives a message from new node that is not in its current neighbor list.

The number of transmissions per node in PBSM is around 44% lower than the methods of [VO, VO-CDS]. Since the nodes are moving with moderate speed, the new node discovered is expected to be a previous 2-hop neighbor most of the time. If it was formerly a 2-hop neighbor then it will be in the R list and there will therefore be no transmission in PBSM. Consequently, the differences between PBSM and the other methods are more than just those related to static networks. As explained in section 2.4., nodes use average relative velocity as a threshold to switch between plain, scoped and hyperflooding. On average, 42% of nodes are in scoped/CDS mode, 22% of nodes are in plain and 36% of nodes are in hyper mode. Some nodes in [VO, VO-CDS] are in plain, scoped or CDS-flooding mode even though there is a movement in their neighborhood. This is because the speed of the neighboring nodes (moderate mobility) might be less than the threshold value required to switch the type of flooding. As we know, in scoped, plain flooding [VO], or CDS-flooding [VO-CDS], a node will not transmit when it discovers a new neighbor. As a result, the methods of [VO, VO-CDS] perform better than the method of [CEM]. Furthermore, as CDS-based flooding saves more transmissions than scoped flooding, VO-CDS performs slightly better than VO. In addition, scoped/CDS flooding [VO, VO-CDS] saves more transmissions than [CEM], where a node decides to transmit if it has a neighbor other than the sending node. As we know, in hyperflooding, a node will transmit whenever it receives a message from a new neighbor. As a result, these protocols generate more transmissions in this network than in the static network.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there are more changes that occur between two hello messages. That is, the nodes will move by larger distances, there will be more broadcast messages, and there will be more discoveries of new nodes between subsequent hello messages. Compared to other methods, PBSM saves more transmissions since it does not consider broadcasting every time two nodes discover each other. Consequently, the difference between PBSM and the other methods increases as *B* increases. The data tables are given below:

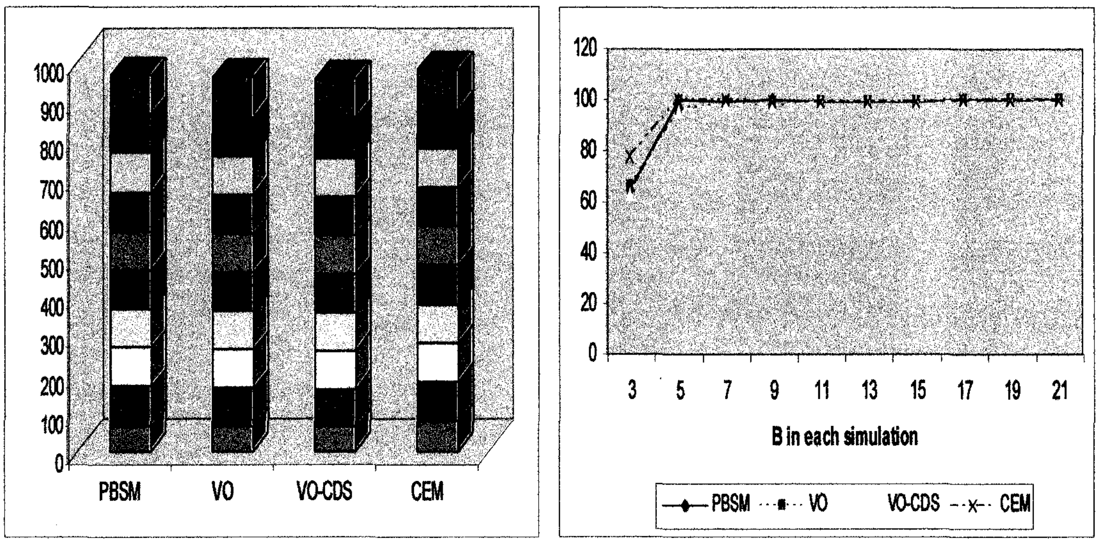
	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	0.8	1.24	1.2	2.8	3 (1)
	1.64	2.12	2	2.84	5 (2)
	1.88	2.68	2.48	3.64	7 (3)
	1.88	3.2	3.12	4.44	9 (4)
	2.16	3.84	3.68	5.16	11 (5)
	2.04	3.96	3.8	5.92	13 (6)
	2.32	4.32	4.28	6.76	15 (7)
	2.12	4.24	4	7.12	17 (8)
	2.24	4.68	4.48	7.64	19 (9)
	2.32	4.48	4.36	7.96	21 (10)
Total	**19.4**	**34.76**	**33.4**	**54.28**	
Average	**1.94**	**3.476**	**3.34**	**5.428**	
95%ConfidenceInterval	**1.94±0.28**	**3.48±0.71**	**3.34±0.69**	**5.43±1.21**	

Table 9: Total transmissions per node in a moderate network (D=1/2, n=25, d=4)

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	20	31	30	70	3 (1)
	41	53	50	71	5 (2)
	47	67	62	91	7 (3)
	47	80	78	111	9 (4)
	54	96	92	129	11 (5)
	51	99	95	148	13 (6)
	58	108	107	169	15 (7)
	53	106	100	178	17 (8)
	56	117	112	191	19 (9)
	58	112	109	199	21 (10)
Total	485	869	835	1357	
Average	48.5	86.9	83.5	135.7	

Table 10: Total number of transmissions in a moderately mobile network (D=1/2, n=25, d=4)

Percentage of reliability



(A) (B)

Figure 24: Percentage of reliability in a moderately mobile network (D=1/2, n=25, d=4)

	PBSM	VO	VO-CDS	CEM
Total	963.4	956.8	951.58	976.4
Average	96.34	95.68	95.158	97.64

Table 11: Summary of Percentage of reliability in a moderately mobile network

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. In a network of moderate mobility, PBSM and CEM are nearly 97% reliable. The VO and VO-CDS protocols are nearly 95% reliable. This is depicted in the bar graph. In the proposed PBSM method, the CDS nodes are selected in such a way that if only the CDS nodes transmit, then most of the nodes in the network will receive the message. The non-CDS nodes (that run a longer timeout) transmit if they have non-empty N lists and thus increase reliability. Moreover, if a node discovers a new neighbor, it will transmit if the new node is not in R list. The other methods are based on hyperflooding, where a node always transmits when it discovers a new node.

As explained in section 2.4, nodes use relative velocity as a threshold to switch between plain, scoped/CDS, and hyperflooding. The results show that, on average, 42% of nodes are in scoped/CDS mode, 22% of nodes are in plain mode and 36% of nodes are in hyper mode. Some nodes in [VO, VO-CDS] are in scoped/CDS or plain mode even though there is movement in their neighborhood. This is because the speed of the nodes (moderate mobility) might be less than the threshold value required to switch the type of flooding. As we know, in scoped/CDS or plain flooding [VO, VO-CDS], a node will not transmit when it discovers a new neighbor. So, reliability of method of [VO, VO-CDS] is slightly less than other methods.

The above line graph shows the reliability percentage (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there will be more changes occurring between the two hello messages. That is, the nodes will move by larger distances, there will be more broadcast messages and more disappearances of nodes between subsequent hello messages. However, the disappearance of nodes is compensated by the discoveries of new nodes. Consequently, the reliability of all protocols decreases very slightly as *B* increases. The data table is given below:

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	65	66	63	78	3 (1)
	100	96.4	95.58	100	5 (2)
	99.6	98.4	98.4	100	7 (3)
	100	97.6	97.6	99.6	9 (4)
	99.6	99.6	99.6	99.6	11 (5)
	99.6	99.6	99.6	99.6	13 (6)
	99.6	99.6	99.6	99.6	15 (7)
	100	99.6	99.2	100	17 (8)
	100	100	100	100	19 (9)
	100	100	99	100	21 (10)
Total	963.4	956.8	951.58	976.4	
Average	96.34	95.68	95.158	97.64	
Confidence Interval95%	96.3±6.83	95.7±6.5	95.2±7.05	97.6±4.28	

Table 12: Percentage of reliability in a moderately mobile network (D=1/2, n=25, d=4)

5.6.3 Highly Mobile Network

Under maximal speed, a node can travel distance *D'* between two hello messages. Let *D'=D*r,* where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is *D*r /H* (=*D'/H*). The average velocity is maximum speed/2. Consequently, both speed and *D'* depend on *D* and *D* is used as a parameter to define mobility.

In a high mobility scenario, all nodes move. [*D*=1, so max speed = *r*/ *H*]. In each simulation, the number of nodes (n) is 25 and the average density (d) is close to 4. The hello message time (*H*) is incremented by 2 in each simulation. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit. Therefore, B (the number of messages that can be sent between two hello messages) is equal to *H*. Each simulation consists of a randomly generated graph and ten hello messages. The following graph is used in the first simulation:

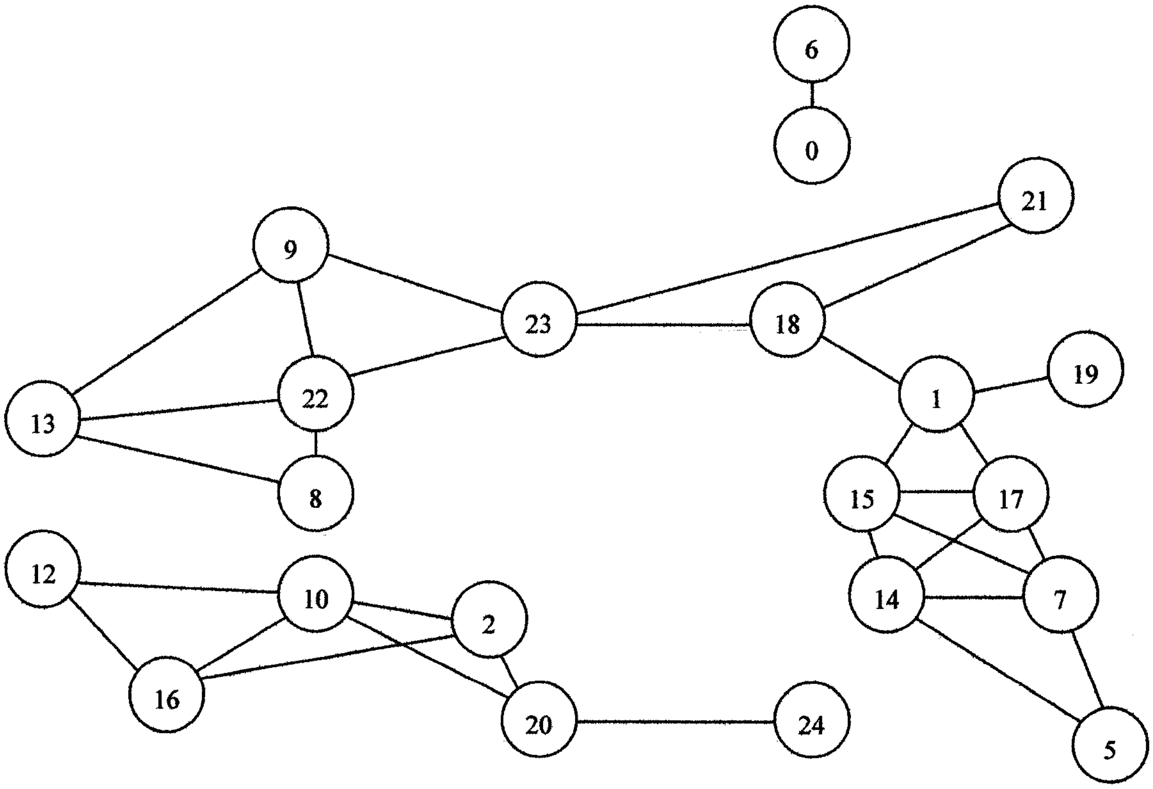
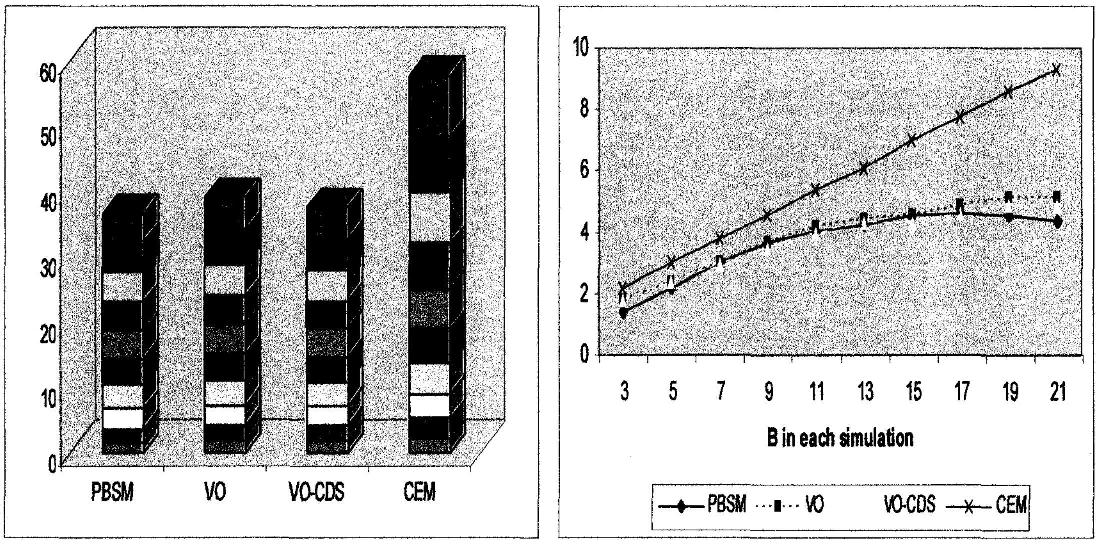


Figure 25: Graph of first simulation of a highly mobile scenario

Total number of transmissions per node



(A) (B)

Figure 26: Total number of transmissions per node in a highly mobile network (D=1, n=25, d=4)

	PBSM	VO	VOCDS	CEM
Total	36.56	39.12	37.56	57.56
Average	3.656	3.912	3.756	5.756

Table 13: Summary of total number of transmissions per node in a highly mobile network

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons for this which are stated in section 5.6.2.

The number of transmissions per node in PBSM is around 7% lower than the methods of [VO, VO-CDS]. In this network, some nodes are moving at a high speed. Therefore, the percentage of new nodes discovered in this network, which come from more than 2 hops away, is higher than that in a moderately mobile network. If a newly discovered node was more than 2-hops away, then it will not be in the R list even though it may have received the message. Therefore, there will be more transmissions in PBSM. Consequently, the difference between PBSM and the other protocols is less than that of a moderately mobile network. Furthermore, thresholds work better in highly mobile networks since there is more difference in the speed of nodes. This means that a node that discovers a new node will be in hyper mode as required, and nodes that do not need to transmit will be in other modes as required. As CDS-based flooding saves more transmissions than scoped flooding, VO-CDS performs slightly better than VO. As explained in section 2.4, nodes use relative velocity as a threshold to switch between plain, scoped, and hyperflooding. On average, 28% of nodes are in scoped/CDS mode, 32% of nodes are in plain mode, and 40% of nodes are in hyper mode. Most nodes in [VO] will switch to hyper mode since there will be frequent movements in their neighborhood. This is because the total speed of the neighboring nodes (high mobility) will be greater than the threshold value required to switch to hyperflooding. As we know, in hyperflooding [VO, VO-CDS, CEM], a node will transmit whenever it receives a message from a new neighbor. As a result, all of the protocols generate more transmissions in this network than in the static or moderately mobile networks.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there will be more changes occurring between the two hello messages. That is, the nodes will move by larger distances, there will be more broadcast messages and more discoveries of new nodes between subsequent hello messages. Compared to other methods, PBSM saves more transmissions as it does not consider broadcasting every time two nodes discover each other. Consequently, the difference between PBSM and other methods increases as *B* increases. The data tables are given below:

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	1.44	1.8	1.8	2.16	3 (1)
	2.2	2.4	2.4	3	5 (2)
	3	3	2.96	3.76	7 (3)
	3.64	3.68	3.48	4.56	9 (4)
	4.04	4.16	4	5.36	11 (5)
	4.24	4.4	4.32	6.12	13 (6)
	4.52	4.52	4.28	6.96	15 (7)
	4.6	4.88	4.6	7.76	17 (8)
	4.52	5.16	4.84	8.56	19 (9)
	4.36	5.12	4.88	9.32	21 (10)
Total	36.56	39.12	37.56	57.56	
Average	3.656	3.912	3.756	5.756	
ConfidenceInterval95%	3.66±0.68	3.91±0.72	3.76±0.66	5.76±1.49	

Table 14: Total number of transmissions per node in a highly mobile network (D=1, n=25, d=4)

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	36	45	45	54	3 (1)
	55	60	60	75	5 (2)
	75	75	74	94	7 (3)
	91	92	87	114	9 (4)
	101	104	100	134	11 (5)
	106	110	108	153	13 (6)
	113	113	107	174	15 (7)
	115	122	115	194	17 (8)
	113	129	121	214	19 (9)
	109	128	122	233	21 (10)
Total	914	978	939	1439	
Avg	91.4	97.8	93.9	143.9	

Table 15: Total number of transmissions in a highly mobile network (D=1, n=25, d=4)

Percentage of reliability

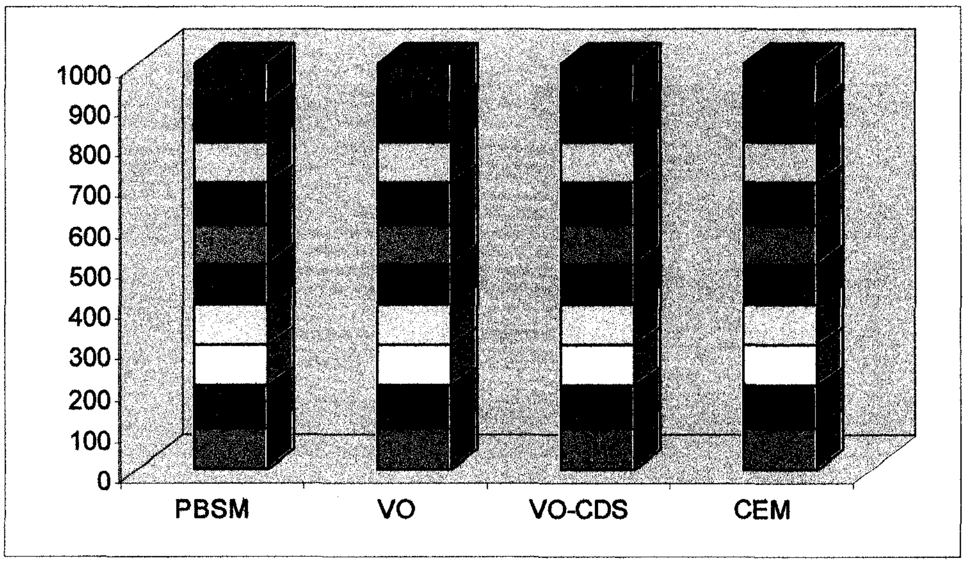


Figure 27: Percentage of reliability in a highly mobile network (D=1, n=25, d=4)

	PBSM	VO	VO-CDS	CEM
Total	990	990	990	990
Average	99	99	99	99

Table 16: Summary of Percentage of reliability in a highly mobile network

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. In a network of high mobility, all protocols are 99% reliable. This is depicted in the bar graph.

In this network, the speed of the mobile nodes is greater than that of the nodes in other network. Consequently, there are more disappearances of old nodes and discoveries of new nodes. The additional transmissions due to the discovery of new nodes make up for the reduction in reliability caused by the disappearance of old nodes. In [VO, VO-CDS], it is seen that on average 28% of nodes are in scoped/CDS mode, 32% of nodes are in plain mode, and 40% of nodes are in hyper mode. Since more nodes are in hyper mode, there will be more transmissions due to the discovery of new nodes. Furthermore, thresholds work better in highly mobile networks as there is more difference in the speed of nodes. Moreover, a node can connect two disconnected parts of a graph and can therefore increase reliability. Due to the increased movement, it is most likely that a disconnected node will be discovered by some other node and therefore receives the message from that node. Thus, reliability increases and becomes more than that of the moderately mobile network.

	PBSM	VO	VOCDS	CEM	B (No of simulation)
	95	95	95	95	3 (1)
	95	95	95	95	5 (2)
	100	100	100	100	7 (3)
	100	100	100	100	9 (4)
	100	100	100	100	11 (5)
	100	100	100	100	13 (6)
	100	100	100	100	15 (7)
	100	100	100	100	17 (8)
	100	100	100	100	19 (9)
	100	100	100	100	21 (10)
Total	990	990	990	990	
Average	99	99	99	99	
ConfidenceInterval95%	99±1.31	99±1.31	99±1.31	99±1.31	

Table 17: Percentage of reliability in a highly mobile network (D=1, n=25, d=4)

5.6.4 Comparing performance based on mobility (*D*)

Under maximal speed, a node can travel distance *D'* between two hello messages. Let *D'=D*r,* where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is *D*r /H* (*=D'/H*). The average velocity is maximum speed/2. Consequently, both speed and *D'* depend on *D* and *D* is used as a parameter to define mobility.

The performance of the protocols is compared based on mobility. In each simulation, the number of nodes (n) is 25, the hello message time (H) is 5, and the average density (d) is close to 4. Sending a broadcast message to an immediate neighbor requires fixed time B'. $B = H/B'$ (number of messages that can be sent between two hello messages where $B' = 1$ and H is 5). The parameter D indicates the maximum speed of the node. D is incremented by 0.15 in each simulation. When D is less than 1, half of the nodes move. When D >= 1 (high mobility), all nodes move. It can be seen that on average 20% of nodes are in scoped/CDS mode, 35% of nodes are in plain mode, and 45% of nodes are in hyper mode. Each simulation consists of a randomly generated graph and ten hello messages. The graph of the first simulation is:

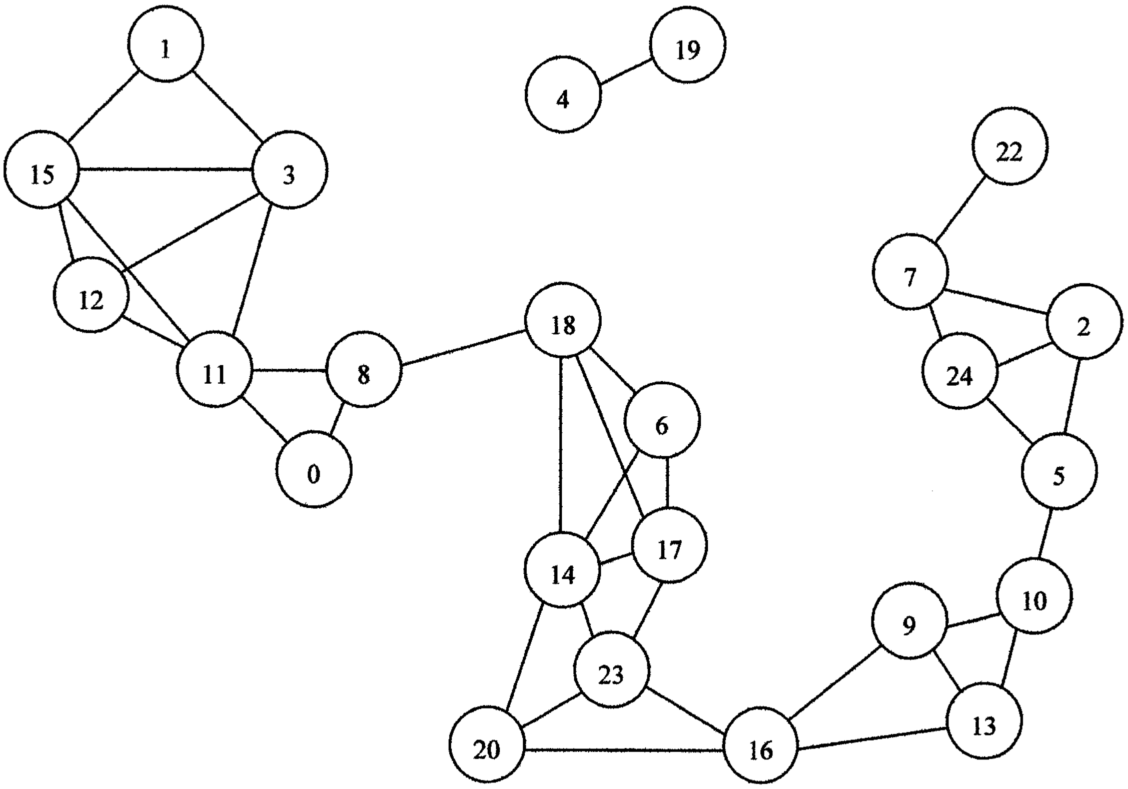
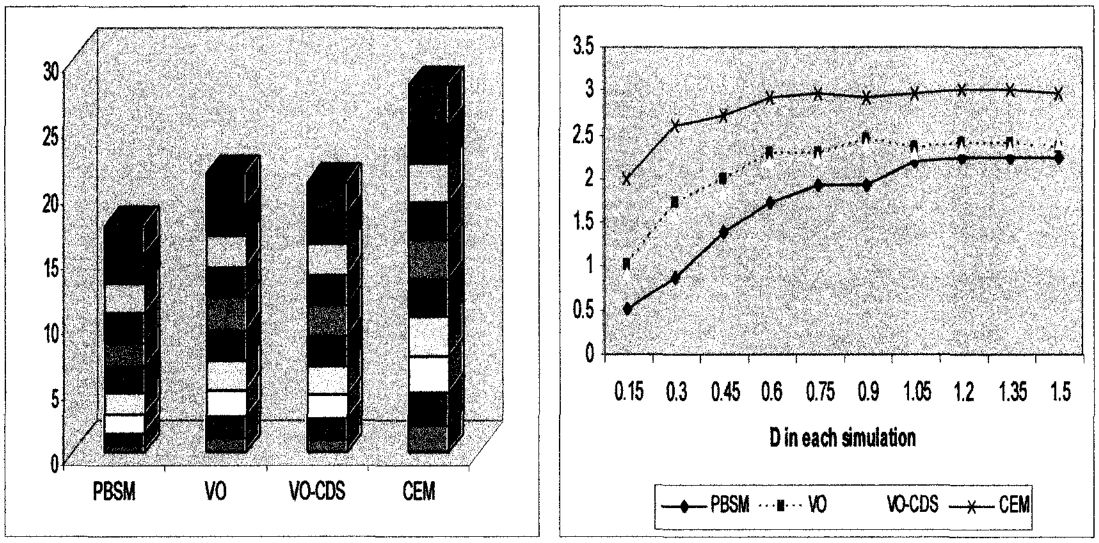


Figure 28: Graph used in first simulation of a mobility-based scenario (nodes 1-12 are mobile)

Total number of transmissions per node



(A) (B)

Figure 29: Total number of transmissions per node in a mobility-based scenario (B=5, n=25, d=4)

	PBSM	VO	VO-CDS	CEM
Total	17.28	21.24	20.6	28.04
Average	1.728	2.124	2.06	2.804

Table 18: Summary of total no. of transmissions per node in a mobility-based scenario

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 19% less than protocol of [VO, VO-CDS].

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *D* used in each simulation. As *D* increases, the disappearance of old nodes and the discoveries of new nodes increase. When nodes are moving at high speeds (*D* ≥ 1.05), the percentage of new nodes discovered that are coming from more than 2 hops away increases. If a newly discovered node was more than 2-hops away initially then it will not be in the R list even though it may have received the message. Therefore, there will be some extra transmissions in PBSM as well as in the other methods. Consequently, the difference in the number of transmissions between PBSM and the other methods decreases when D ≥ 1.05. Furthermore, thresholds work better as mobility increases since there is more difference in the speed of the nodes. This means that a node that discovers a new node will be in hyper mode as required, and that nodes that do not need to transmit will be in other modes as required. The data tables are given below:

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	0.52	1	0.92	2	0.15 (1)
	0.88	1.72	1.6	2.6	0.3 (2)
	1.4	2	1.84	2.72	0.45 (3)
	1.72	2.28	2.2	2.92	0.6 (4)
	1.92	2.28	2.24	2.96	0.75 (5)
	1.92	2.44	2.4	2.92	0.9 (6)
	2.2	2.36	2.28	2.96	1.05 (7)
	2.24	2.4	2.36	3	1.2 (8)
	2.24	2.4	2.36	3	1.35 (9)
	2.24	2.36	2.4	2.96	1.5 (10)
Total	17.28	21.24	20.6	28.04	
Average	1.728	2.124	2.06	2.804	
ConfidenceInterval95%	1.73±0.38	2.12±0.28	2.06±0.3	2.8±0.19	

Table 19: Total number of transmissions per node in a mobility-based scenario (B=5, n=25, d=4)

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	13	25	23	50	0.15 (1)
	22	43	40	65	0.3 (2)
	35	50	46	68	0.45 (3)
	43	57	55	73	0.6 (4)
	48	57	56	74	0.75 (5)
	48	61	60	73	0.9 (6)
	55	59	57	74	1.05 (7)
	56	60	59	75	1.2 (8)
	56	60	59	75	1.35 (9)
	56	59	60	74	1.5 (10)
Total	**432**	**531**	**515**	**701**	
Average	**43.2**	**53.1**	**51.5**	**70.1**	

Table 20: Total number of transmissions in a mobility-based scenario (B=5, n=25, d=4)

Percentage of reliability

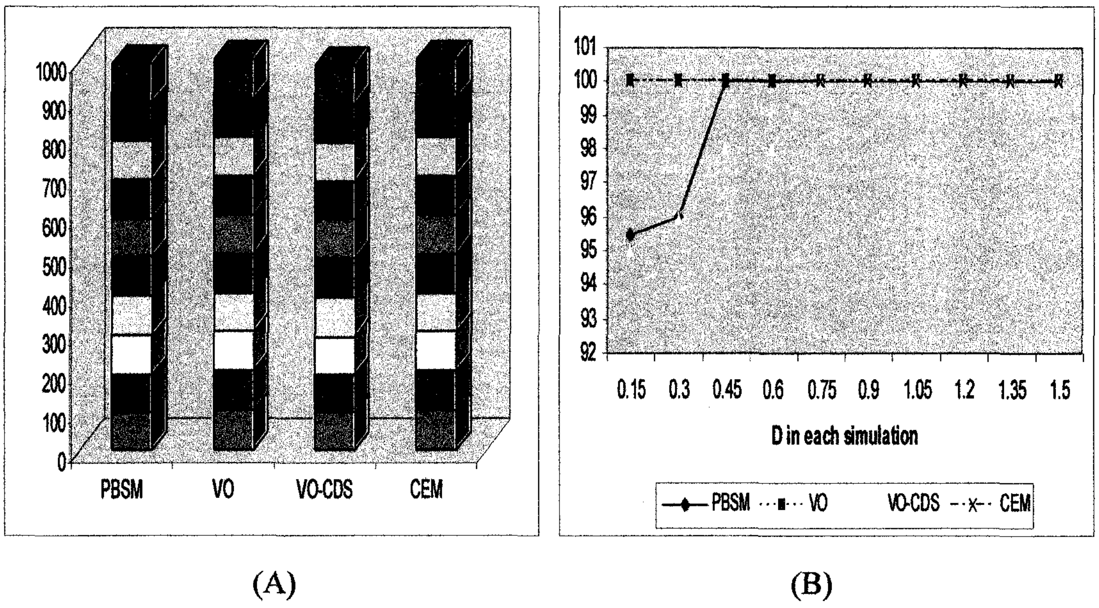


Figure 30: Percentage of reliability in mobility-based scenario (B=5, n=25, d=4)

	PBSM	VO	VO-CDS	CEM
Total	991.45	1000	987	1000
Average	99.145	100	98.7	100

Table 21: Summary of percentage of reliability in mobility-based scenario

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes that receive the message. Each color in the bar graph represents a simulation. All protocols are around 100% reliable. This is depicted in the bar graph.

The above line graph shows the percentage of reliability (y-axis) in each simulation (x-axis). The percentage of reliability is a measure of the number of nodes that receive the message. The numbers on the x-axis indicate the value of *D* used in each simulation. When nodes are moving at slow or moderate speed ($D \leq 0.45$), the newly discovered nodes are likely to come from 2 hops away (or less). If a newly discovered node was 2-hops away (or less) initially then it will be in the R list if it has received the message. Therefore, there will be no transmissions in PBSM. This transmission may be required if there is another node in the neighborhood that needs message. Consequently, the reliability of PBSM is less than other methods when $D \leq 0.45$. As *D* increases it is more likely that the newly discovered node will be more than 2 hops away. Consequently, there will be transmission in PBSM and its reliability will increase. At lower speed ($D \leq 0.75$), most nodes in VO-CDS employ CDS-based flooding where most of the time only nodes in CDS transmit. This reduces transmission at the cost of reliability. Therefore, the reliability of VO-CDS is less than other methods. The data table is given below:

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	95.45	100	95	100	0.15 (1)
	96	100	96	100	0.3 (2)
	100	100	98	100	0.45 (3)
	100	100	98	100	0.6 (4)
	100	100	100	100	0.75 (5)
	100	100	100	100	0.9 (6)
	100	100	100	100	1.05 (7)
	100	100	100	100	1.2 (8)
	100	100	100	100	1.35 (9)
	100	100	100	100	1.5 (10)
Total	991.45	1000	987	1000	
Average	99.145	100	98.7	100	

Table 22: Percentage of reliability in a mobility-based scenario (B=5, n=25, d=4)

5.6.5 Comparing performance based on density (*d*)

The performance of these protocols is compared based on density (*d*). In each simulation, the number of nodes (*n*) is 25, hello message time (*H*) is 10 and the value of mobility is between moderate and high mobility (i.e. *D*=0.75). Sending a broadcast message to immediate neighbor requires fixed time *B'*. *B* =*H/B'* (number of messages that can be sent between two hello messages where *B'* =1 and *H* is 10). Density (*d*) indicates the desired average node degree and is incremented by 1 in each simulation. The average node degree of the generated graph will be close to density (*d*). Each simulation consists of a randomly generated graph and ten hello messages. The following graph is used in the first simulation.

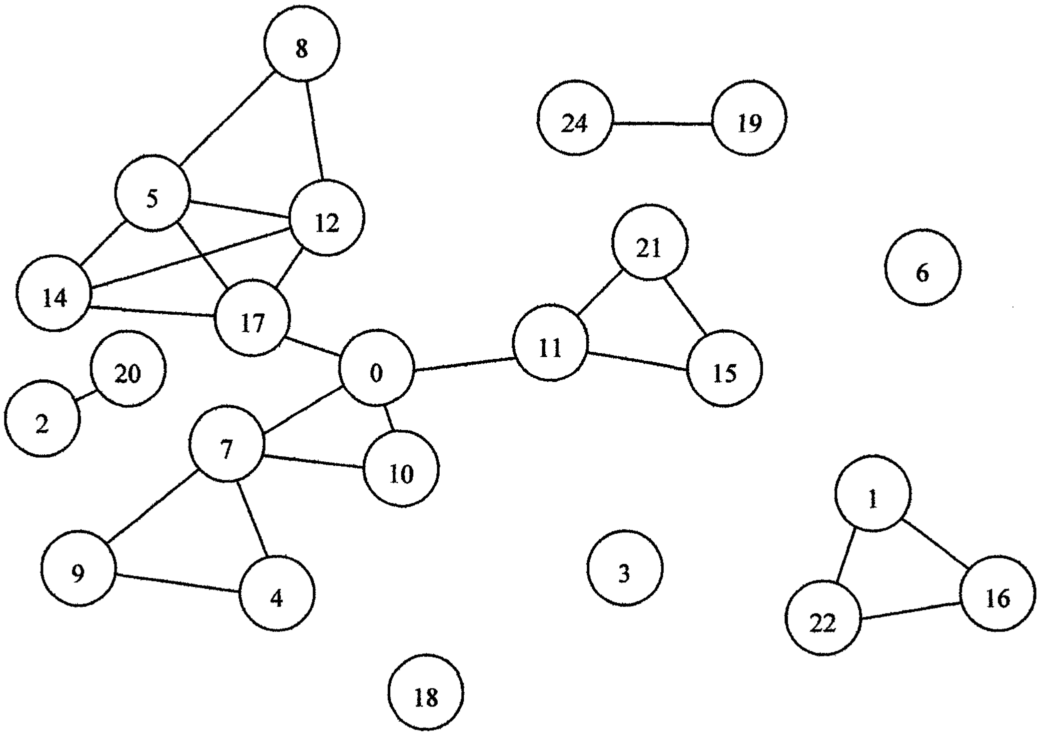
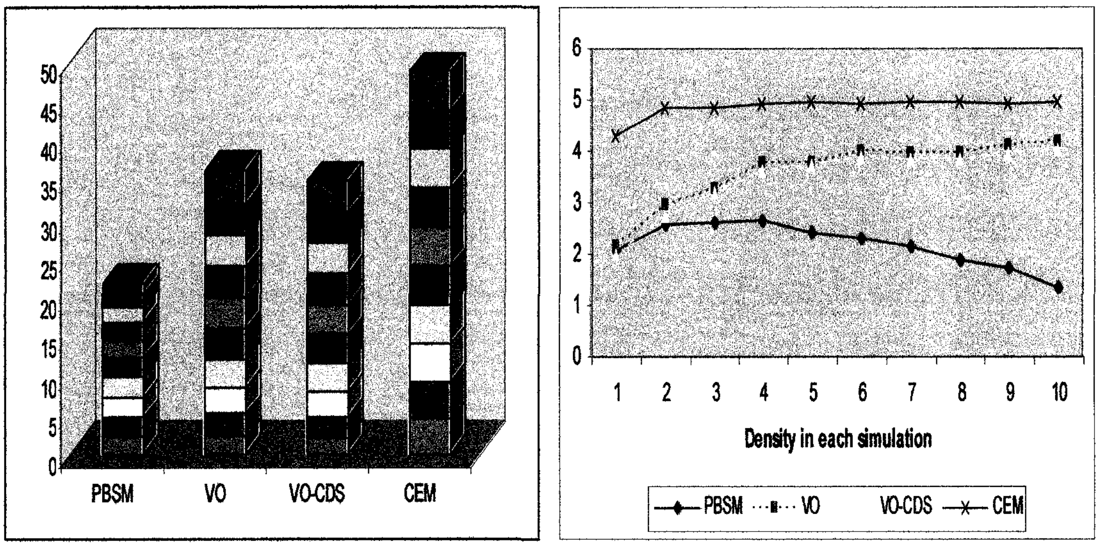


Figure 31: Graph used in the first simulation of a density-based scenario (nodes 1-12 are mobile)

Total number of transmissions per node



(A) (B)

Figure 32: Total number of transmissions per node in a density-based scenario (B=10, n=25, D=0.75)

	PBSM	VO	VO-CDS	CEM
Total	21.76	36.16	34.8	48.6
Average	2.176	3.616	3.48	4.86

Table 23: Summary of total number of transmissions per node in a density-based scenario

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons that are stated in section 5.6.2. The number of transmissions per node in PBSM is around 40% less than in the methods of [VO, VO-CDS].

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *d* used in each simulation. As *d* increases, the graph becomes denser, which means that more nodes will receive a message from a single transmission. The performance of [CEM] does not change with variations of density. As *d* increases, the percentage of CDS nodes compared to all nodes decreases. Furthermore, in a dense network there is a greater possibility that a new node discovered was previously a 2-hop neighbor. Therefore, if a new node has previously received the message, it will be in the R list. Unlike other methods, there will be no additional transmissions in PBSM. Consequently, the difference in the number of transmissions between PBSM and other methods increases as *d* increases. The data tables are given below:

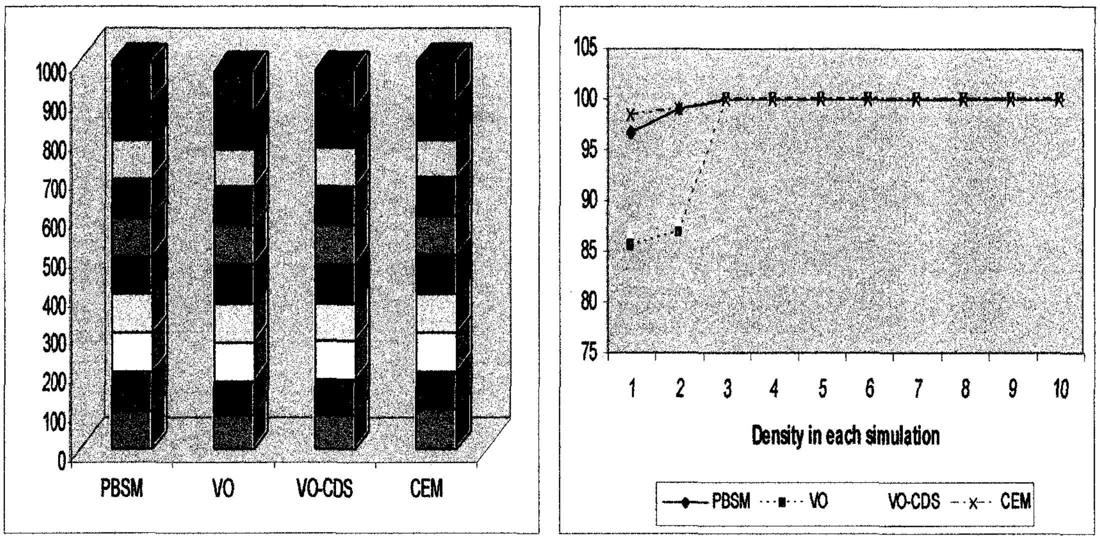
	PBSM	VO	VO-CDS	CEM	d (No of simulation)
	2.08	2.16	2	4.32	2 (1)
	2.56	2.96	2.72	4.84	3 (2)
	2.6	3.28	3.2	4.84	4 (3)
	2.64	3.76	3.6	4.92	5 (4)
	2.44	3.76	3.72	4.96	6 (5)
	2.32	4	3.88	4.92	7 (6)
	2.16	3.96	3.84	4.96	8 (7)
	1.88	3.96	3.84	4.96	9 (8)
	1.72	4.12	3.96	4.92	10 (9)
	1.36	4.2	4.04	4.96	11 (10)
Total	21.76	36.16	34.8	48.6	
Average	2.176	3.616	3.48	4.86	
Confidence Interval95%	2.18±0.26	3.62±0.4	3.48±0.41	4.86±0.12	

Table 24: Total number of transmissions per node in a density-based scenario (B=10, n=25, D=0.75)

	PBSM	VO	VO-CDS	CEM	d (No of simulation)
	52	54	50	108	2 (1)
	64	74	68	121	3 (2)
	65	82	80	121	4 (3)
	66	94	90	123	5 (4)
	61	94	93	124	6 (5)
	58	100	97	123	7 (6)
	54	99	96	124	8 (7)
	47	99	96	124	9 (8)
	43	103	99	123	10 (9)
	34	105	101	124	11 (10)
Total	544	904	870	1215	
Average	54.4	90.4	87	121.5	

Table 25: Total number of transmissions in a density-based scenario
(B=10, n=25, D=0.75)

Percentage of reliability



(A) (B)

Figure 33: Percentage of reliability in a density-based scenario (B=10, n=25, D=0.75)

	PBSM	VO	VO-CDS	CEM
Total	995.8	972.6	975	997.4
Average	99.58	97.26	97.5	99.74

Table 26: Summary of percentage of reliability in a density-based scenario

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. The PBSM and [CEM] protocols are around 98% reliable. The protocols of [VO and VO-CDS] are around 97% reliable. The reliability of [VO, VO-CDS] is lower since some nodes in [VO, VO-CDS] are in scoped or CDS flooding mode. These nodes do not transmit when they discover new node and therefore the percentage of reliability decreases.

The above line graph shows the percentage of reliability (y-axis) in each simulation (x-axis). The percentage of reliability is a measure of the number of nodes receiving a message. The numbers on the x-axis indicate the value of *d* used in each simulation. The graph indicates that the percentage of reliability remains fairly constant as *d* changes. The data table is given below:

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	96.8	85.6	87	98.4	2 (1)
	99	87	88	99	3 (2)
	100	100	100	100	4 (3)
	100	100	100	100	5 (4)
	100	100	100	100	6 (5)
	100	100	100	100	7 (6)
	100	100	100	100	8 (7)
	100	100	100	100	9 (8)
	100	100	100	100	10 (9)
	100	100	100	100	11 (10)
Total	995.8	972.6	975	997.4	
Average	99.58	97.26	97.5	99.74	
ConfidenceInterval95%	99.6±0.64	97.3±3.59	97.5±3.27	99.7±0.35	

Table 27: Percentage of reliability in a density-based scenario (B=10, n=25, D=0.75)

5.6.6 Comparing performance in a corridor scenario

A corridor scenario is simulated. In this scenario, half of the nodes lie in the left portion of the simulated area whereas the rest of the nodes lie in the right portion. There is another node, 12, that is connected to the left portion of the region but not connected to the right region. This is shown in the following graph:

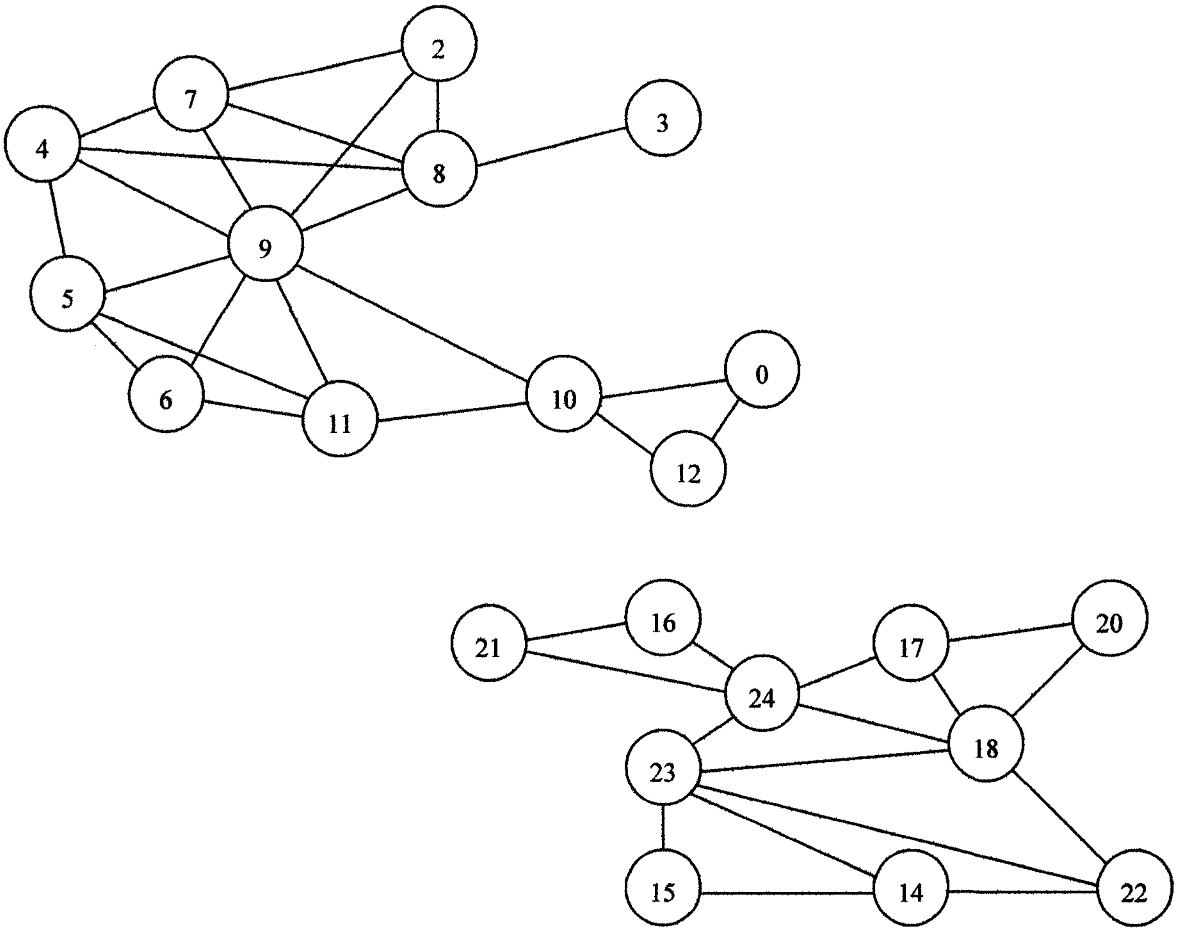


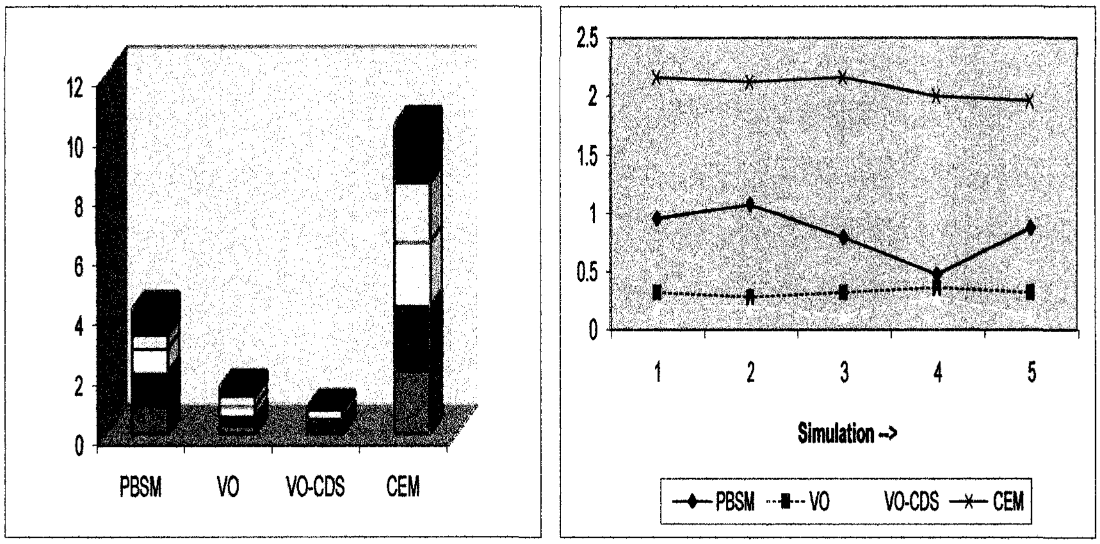
Figure 34: Graph used in the first simulation of a corridor scenario (nodes 12-24 are mobile)

The source lies on the left region and broadcasting starts from the source. All nodes remain static until node 12 receives the message. At this point, all nodes in the left region including node 12 will receive the message. Nodes in the right region do not yet receive any message. Node 12 is the only node that can connect the two regions. After node 12 receives the message, all nodes in the right region, including node 12, start moving. Nodes in the right region will move within their region. Node 12 moves between two regions so that it may come closer to nodes from the right region.

In each simulation, the number of nodes (*n*) is 25, average density (*d*) is close to 4, hello message time (*H*) is 4, and there is high mobility (i.e. D=1). Sending a broadcast message to an immediate neighbor requires fixed time *B'*. *B =H/B'* (number of messages that can be sent between two hello messages where *B'* =1 and *H* is 4). Each simulation consists of a randomly generated graph and ten hello messages.

Considering [VO, VO-CDS], node 12 moves slowly so that its speed is below the threshold and it will not be in hyper mode. Consequently, node 12 will not transmit even if it discovers any node from the left region. As a result, none of the nodes in the left region will receive the message. On the contrary, in PBSM and CEM, node 12 will transmit if it discovers any node from the left region. Therefore, the nodes in the left region will receive the message. For this reason, the reliability of PBSM and CEM is almost double that of [VO, VO-CDS]. The additional transmissions in PBSM and CEM are due to transmissions among nodes in the right region.

Total number of transmissions per node



(A) (B)

Figure 35: Total number of transmission per node in a corridor scenario (B=4, n=25, D=1, d=4)

	PBSM	VO	VO-CDS	CEM
Total	4.2	1.6	0.92	10.4
Average	0.84	0.32	0.184	2.08

Table 28: Summary of total number of transmission per node in a corridor scenario

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The data tables are given below:

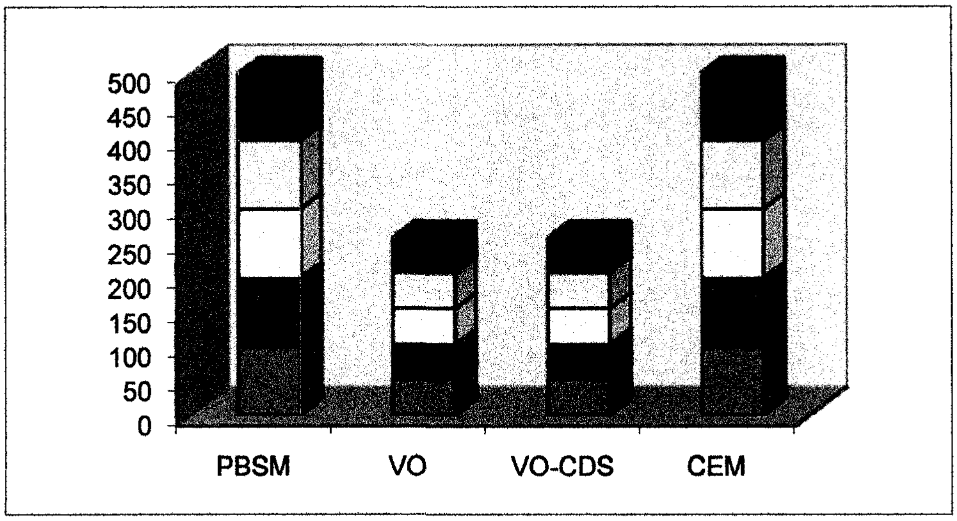
	PBSM	VO	VO-CDS	CEM	(No of simulation)
	0.96	0.32	0.2	2.16	(1)
	1.08	0.28	0.2	2.12	(2)
	0.8	0.32	0.12	2.16	(3)
	0.48	0.36	0.28	2	(4)
	0.88	0.32	0.12	1.96	(5)
Total	4.2	1.6	0.92	10.4	
Average	0.84	0.32	0.184	2.08	

Table 29: Total number of transmissions per node in a corridor scenario (B=4, n=25, D=1, d=4)

	PBSM	VO	VO-CDS	CEM	(No of simulation)
	24	8	5	54	(1)
	27	7	5	53	(2)
	20	8	3	54	(3)
	12	9	7	50	(4)
	22	8	3	49	(5)
Total	105	40	23	260	
Average	21	8	4.6	52	

Table 30: Total number of transmissions in a corridor scenario (B=4, n=25, D=1, d=4)

Percentage of reliability



(A) (B)

Figure 36: Percentage of reliability in a corridor scenario (B=4, n=25, D=1, d=4)

	PBSM	VO	VO-CDS	CEM
Total	500	260	260	500
Average	100	52	52	100

Table 31: Summary of percentage of reliability in a corridor scenario

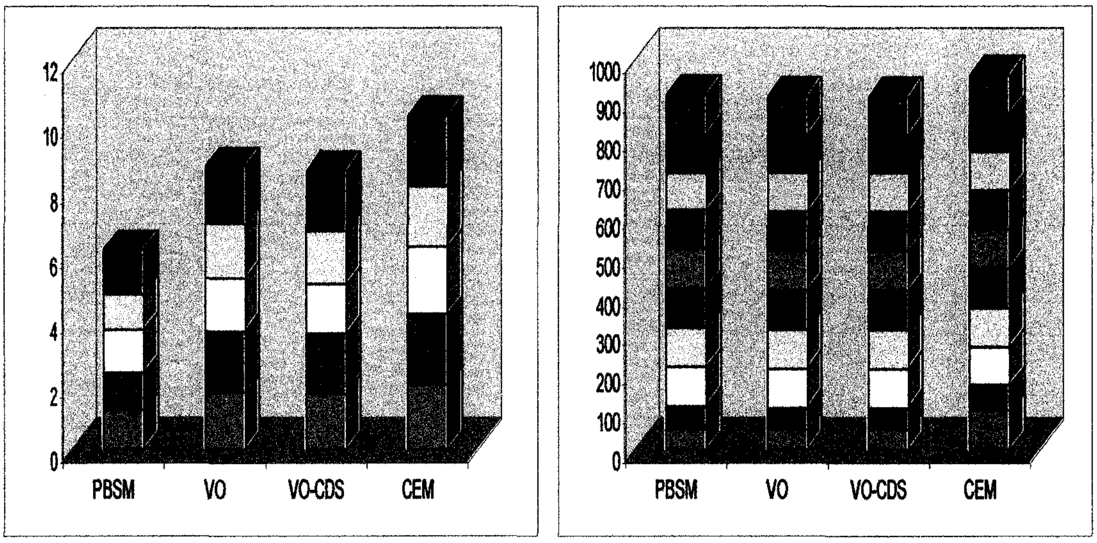
The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. Data table is given below:

	PBSM	VO	VO-CDS	CEM	No of simulation
	100	52	52	100	(1)
	100	52	52	100	(2)
	100	52	52	100	(3)
	100	52	52	100	(4)
	100	52	52	100	(5)
Total	500	260	260	500	
Average	100	52	52	100	

Table 32: Percentage of reliability in a corridor scenario (B=4, n=25, D=1, d=4)

5.6.7 Comparing performance in a disaster scenario

The performance of the protocols is compared in a disaster scenario. In a disaster scenario, all nodes increase speed in order to reach an exit quickly after hearing a fire alarm. That is, all nodes increase speed at the same time. In each simulation, the number of nodes (*n*) is 25, hello message time (*H*) is 3, and the value of mobility is between moderate and high mobility (i.e. *D*=0.75). Sending a broadcast message to an immediate neighbor requires fixed time *B'*. *B* =*H*/*B'* (number of messages that can be sent between two hello messages where *B'* =1 and *H* is 3). The speed of all nodes is increased in each time unit. Each simulation consists of a randomly generated graph and ten hello messages.



(A) (B)

Figure 37: (A) Total number of transmissions per node in a disaster scenario

(B) Percentage of reliability in a disaster scenario (B=3, n=25, D=0.75, d=4)

The bar graph in Figure 37A shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for proposed PBSM method is less than other methods. There are four main reasons that are stated in section 5.6.2. The number of transmissions per node in PBSM is around 30%

less than methods of [VO, VO-CDS]. As explained in section 2.4, nodes use relative velocity as a threshold to switch between plain, scoped/CDS, and hyperflooding. Results show that, on average, 20% of nodes are in scoped/CDS mode, 30% of nodes are in plain mode, and 50% of nodes are in hyper mode. As all nodes continue to increase speed, their average relative velocity becomes greater than the threshold. Consequently, most of the nodes are in hyper mode. As a result, there are more transmissions in [VO, VO-CDS] than in PBSM. However, some nodes are not in hyper mode. Although the speed of the nodes increases, their relative speed might decrease if their direction changes.

The bar graph in Figure 37B shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. All protocols are around 98% reliable. This shows that the additional transmissions in [VO, VO-CDS] do not necessarily increase reliability compared to PBSM. Data tables are given below:

	PBSM	**VO**	**VO-CDS**	**CEM**	**(No of simulation)**
	1.2	1.76	1.72	2	(1)
	1.16	1.88	1.84	2.16	(2)
	1.32	1.6	1.52	2.04	(3)
	1.12	1.68	1.64	1.92	(4)
	1.36	1.84	1.88	2.16	(5)
Total	6.16	8.76	8.6	10.28	
Average	1.232	1.752	1.72	2.056	

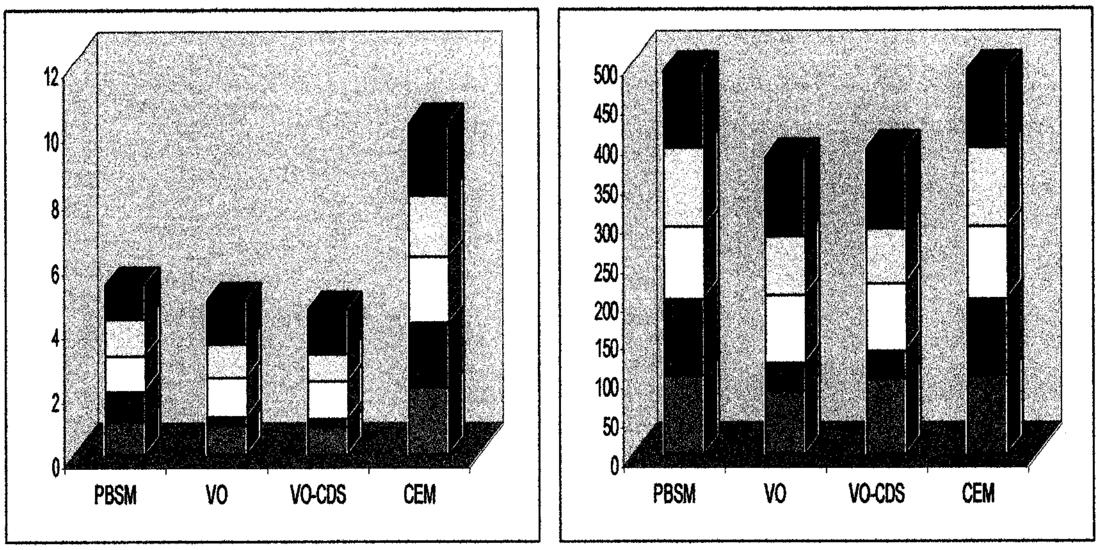
Table 33: Total number of transmissions per node in a disaster scenario
(B=3, n=25, D=0.75, d=4)

	PBSM	VO	VO-CDS	CEM	No of simulation
	100	100	100	100	(1)
	100	96	96	100	(2)
	84	76	76	88	(3)
	96	84	80	96	(4)
	100	88	100	100	(5)
Total	480	444	452	484	
Average	96	88.8	90.4	96.8	

Table 34: Percentage of reliability in a disaster scenario (B=3, n=25, D=0.75, d=4)

5.6.8 Comparing performance in traffic scenario

The performances of protocols are compared in a traffic scenario. In a traffic scenario, all nodes reduce speed in order to slow down when there is a red signal. That is, all nodes reduce speed at the same time. In each simulation, the number of nodes (*n*) is 25, hello message time (*H*) is 3, and the value of mobility is between moderate and high mobility (i.e. *D*=0.75). Sending a broadcast message to an immediate neighbor requires fixed time *B'*. *B* =*H/B'* (number of messages that can be sent between two hello messages where *B'* =1 and *H* is 3). The speed of all nodes decreases in each time unit. Each simulation consists of a randomly generated graph and ten hello messages.



(A) (B)

Figure 38: (A) Total number of transmissions per node in a traffic scenario

(B) Percentage of reliability in a traffic scenario (B=3, n=25, D=0.75, d=4)

The bar graph in Figure 38A shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than the other methods. There are four main reasons for this which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 15% more than in the methods of [VO, VO-CDS]. As explained in section 2.4, nodes use relative velocity as a threshold to switch between plain, scoped/CDS, and hyperflooding. It can be seen that, on average, 60% of nodes are in scoped/CDS mode, 28% of nodes are in plain mode, and 12% of nodes are in hyper mode. As all nodes continuously reduce speed, their average relative velocity becomes lower than the threshold value. Consequently, most of the nodes are in scoped mode. As a result, there are less transmissions in [VO, VO-CDS] than in PBSM. However, some nodes are not in scoped mode. Although the speed of the nodes decreases, the relative speed might increase if the node's directions are opposite.

The bar graph in Figure 38B shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. The reliability of [VO, VO-CDS] is much less than other methods as most nodes in [VO, VO-CDS] are in scoped or CDS flooding mode. These nodes do not transmit when they discover new nodes and therefore their percentage of reliability decreases. Data tables are given below:

	PBSM	VO	VO-CDS	CEM	(No of simulation)
	1	0.88	0.84	2.04	(1)
	0.96	0.24	0.24	2	(2)
	1.04	1.24	1.16	2.04	(3)
	1.16	1.04	0.84	1.92	(4)
	1.08	1.36	1.36	2.16	(5)
Total	5.24	4.76	4.44	10.16	
Average	1.048	0.952	0.888	2.032	

Table 35: Total number of transmissions per node in a traffic scenario (B=3, n=25, D=0.75, d=4)

	PBSM	VO	VO-CDS	CEM	No of simulation
	100	80	96	100	(1)
	100	36	36	100	(2)
	92	88	88	92	(3)
	100	75	70	100	(4)
	100	100	100	100	(5)
Total	492	379	390	492	
Average	98.4	75.8	78	98.4	

Table 36: Percentage of reliability in a traffic scenario (B=3, n=25, D=0.75, d=4)

5.7 Performance Evaluation using simplified MAC layer

Experiments with ideal MAC layer (sec 5.6) showed that PBSM have significant savings in number of transmissions but not significant gains in reliability. If simplified MAC layer is used then more messages will produce more collisions and therefore fewer messages will be received. This means that PBSM, which produces less transmission, is expected to have significant gain in reliability when simplified MAC layer is used.

The methods of [VO and CEM], PBSM and VO-CDS are implemented and comparisons are done by simulation. In this case, a simplified MAC layer, with collision between messages, is assumed. The simulation contains three main scenarios: static, moderate, and high mobility. Moreover, the performance of the protocols is also compared based on D (a parameter used to define the speed of movement), density (d) (a parameter that indicates the average node degree of graph) and B (the number of non-overlapping messages that can be sent between two hello messages. That is, $B = H/B'$ where $B' = 1$ and H is hello message time). Each scenario consists of ten different simulations. In order to obtain a smooth curve, an average of 10 run (where all parameters are constant) is used to gather each data. Each simulation consists of a randomly generated graph and ten hello messages. In each run (simulation), the hello message takes place first. After that, all the broadcasting methods are called. Broadcasting starts at time 0 with the transmission from the source. Hello messages take place at fixed intervals and movements can take place at any time. For each simulation and for each type of flooding, the *total number of transmissions per node* (as in sec 5.6) and *percentage of reliability* (as in sec 5.6) are calculated in order to evaluate the method's performances.

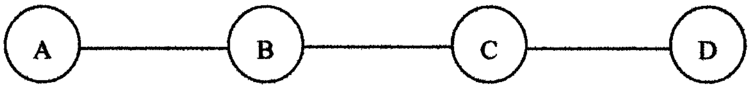


Figure 39: Hidden and Exposed node problem

A simplified IEEE 802.11 MAC protocol has been implemented. It is based on *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) scheme. In simplified MAC layer, a node can listen to the transmission from a neighboring node. The time taken to transmit a message from a node to its immediate neighbor is divided into p slots. Before transmitting, a node listens to the channel. The node will sense the channel busy if at least one of its neighbors is transmitting and it will sense the channel idle if none of its neighbors is transmitting. After receiving message, a node will wait a random amount of time (W). That is, a node will listen in each slot and waits for W (not necessarily consecutive) idle slots. After W such slots, node starts transmission in next slot. Once node started transmission, it continues transmitting for p next slots without listening to anything. There is no collision detection, meaning that a node can't listen while transmitting. A node will receive message correctly if it receives unique messages for p consecutive slots from same neighbor.

In simulation of Ideal MAC layer (section 5.3), it is mentioned that the time taken to transmit a message from a node to its immediate neighbor is one unit (i.e., $B' = 1$ unit or 1 sec). Furthermore, it is also mentioned that H is the fixed time between two hello messages and B is the number of non-overlapping messages that can be sent between two hello messages. That is, $B = H/B'$. In simulation of simplified MAC layer, the following changes are made. B' is divided into 128 slots (i.e. p=128), each of duration 1/128. Thus there are $B*p$ slots between two hello messages. Timer is incremented after p slots. Movement is prorated at smaller intervals, while preserving existing scenarios. That is, movement speeds remain same with respect to B but one move is now divided into p smaller (incremental) moves. The waiting timeout (W) is a random integer between 0 and 25 for all protocols. In PBSM and VO-CDS, the non-CDS nodes add 50 to this timeout.

In simplified MAC layer, reliability reduces mainly due to the following cases. Messages collide when two or more neighboring nodes start transmission simultaneously. *Hidden node problem* occurs when two or more messages arrive simultaneously at a receiving node. The messages collide and as a result the receiving node doesn't receive any message. For example in above figure, if both node A and node C sends a message to neighboring node B at the same time, then the messages will collide at node B. Node B will not receive any message. Thus, reliability reduces. *Exposed node problem* occurs when a node is prevented from sending messages to other nodes due to transmission of a neighboring node. For example in above figure, suppose node B is transmitting a message to node A. Now, node C wants to send a message to node D. Node C senses that the channel is busy due to the transmission of neighbor B. However, *D* could have received the message from *C* because transmission from *C* won't cause any collision at *A* or *D* (*A* is outside the range of *C* and *D* is outside the range of *B*). Thus, there is a loss of throughput as transmission from *C* has to be delayed. Meanwhile, if *D* moves away from *C*, then *D* will not receive the message. Overall, in simplified MAC layer, a node can't receive and transmit at the same time. Moreover, a node cannot receive more than one message at the same time. A node cannot transmit and listen at the same time. A node cannot start transmission if at least one of its neighbors is transmitting at the moment.

5.7.1 Static Network

In a static mobility scenario all nodes are static. That is, there is no movement and the graph does not change over time (*D*=0, so max speed = *0*). In each simulation, the number of nodes is 50. *H* (fixed time between two hello messages) is incremented by 2 in each simulation. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit. Therefore, *B* (the number of non-overlapping messages that can be sent between two hello messages) is equal to *H*.

A. Considering Medium Density (*d*=4) in static scenario
Total number of transmissions per node

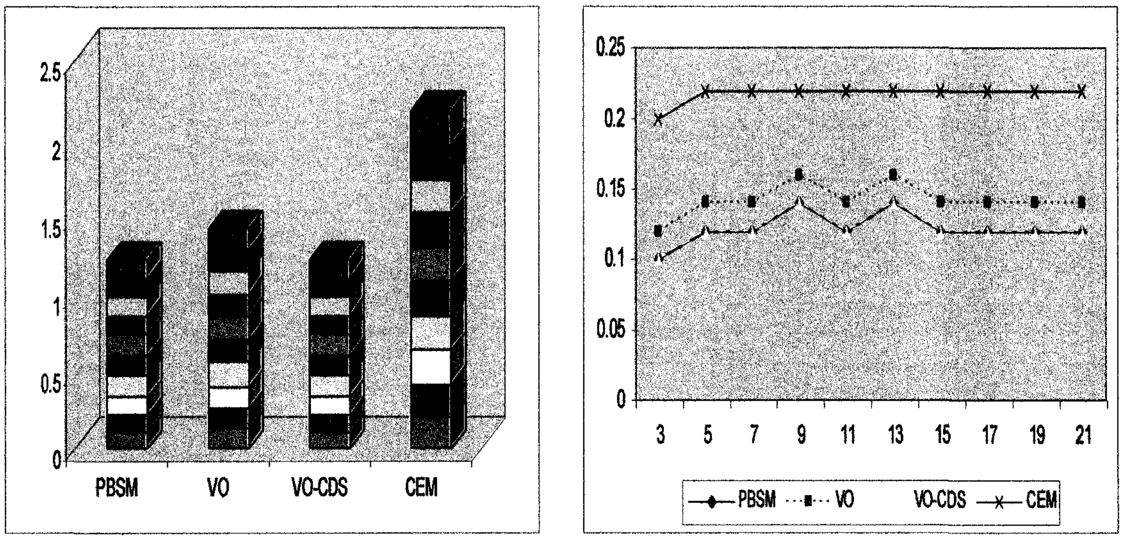


Figure 40: Total number of transmissions per node in a static scenario

(D=0, B=10, n=50, d=4)

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than the methods of [VO] and [CEM]. There are two main reasons which are stated in section 5.6.1. The performance of PBSM and VO-CDS is same as both methods use CDS-based flooding in static networks. The number of transmissions per node in PBSM is around 14% less than the method of [VO]. The methods of [VO] uses scoped flooding in static networks. CEM protocol generates the highest number of transmissions. Under similar condition (i.e. same parameter values), all the methods generated more transmissions when ideal MAC layer is used. This is because in simplified MAC layer, a node has to delay its transmission if at least one of its neighbors is currently transmitting (exposed node problem) and therefore it might not have the opportunity to transmit if broadcast time is over. Furthermore, more nodes decide to transmit (i.e. runs timeout) in methods of [VO, CEM] than PBSM but some of these nodes may not have time to transmit if these nodes have to wait for their neighbors to finish transmission (exposed node problem). Consequently, the difference in number of transmissions between PBSM and methods of [VO, CEM] is smaller than similar scenario (sec 5.6.1) where ideal MAC layer is used.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, the graph won't change as nodes are static. Consequently, the number of transmissions remains fairly constant as *B* increases. The data tables are below:

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	0.1	0.12	0.1	0.2	3 (1)
	0.12	0.14	0.12	0.22	5 (2)
	0.12	0.14	0.12	0.22	7 (3)
	0.14	0.16	0.14	0.22	9 (4)
	0.12	0.14	0.12	0.22	11 (5)
	0.14	0.16	0.14	0.22	13 (6)
	0.12	0.14	0.12	0.22	15 (7)
	0.12	0.14	0.12	0.22	17 (8)
	0.12	0.14	0.12	0.22	19 (9)
	0.12	0.14	0.12	0.22	21 (10)
Total	1.22	1.42	1.22	2.18	
Average	0.122	0.142	0.122	0.218	
ConfidenceInterval95%	0.12±0.01	0.14±0.01	0.12±0.01	0.22±0	

Table 37: Total Number of Transmissions per node in a static network (D=0, B=10, n=50, d=4)

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	5	6	5	10	3 (1)
	6	7	6	11	5 (2)
	6	7	6	11	7 (3)
	7	8	7	11	9 (4)
	6	7	6	11	11 (5)
	7	8	7	11	13 (6)
	6	7	6	11	15 (7)
	6	7	6	11	17 (8)
	6	7	6	11	19 (9)
	6	7	6	11	21 (10)
Total	61	71	61	109	
Average	6.1	7.1	6.1	10.9	

Table 38: Total number of transmissions in a static network (D=0, B=10, n=50, d=4)

Percentage of reliability

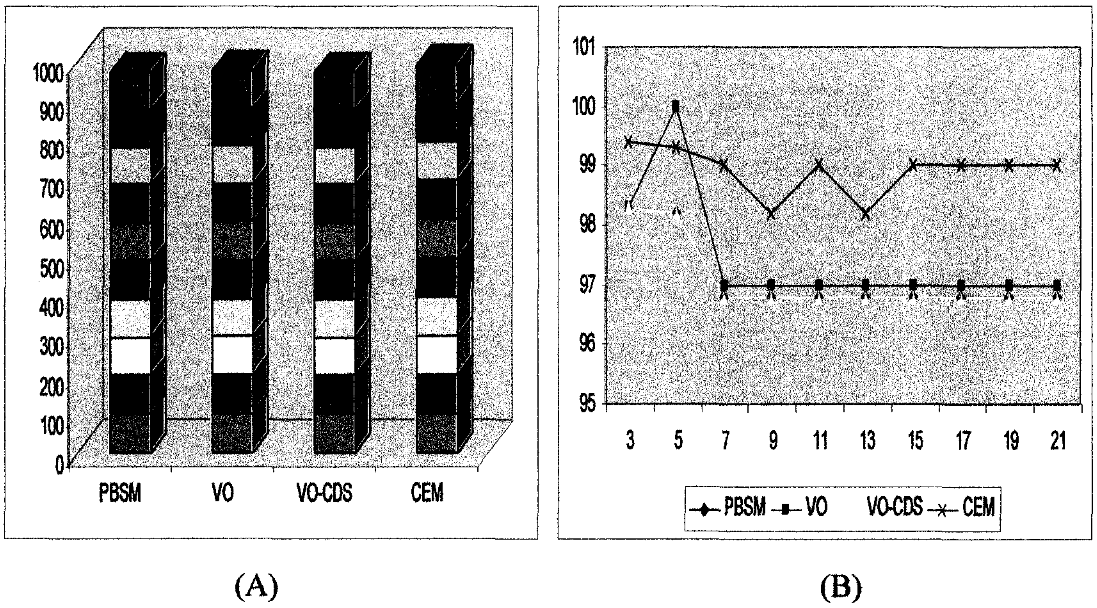


Figure 41: Percentage of reliability for a static network (D=0, B=10, n=50, d=4)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. In a static network, all protocols are around 97% reliable. This is depicted in the bar graph.

In the proposed PBSM method, the CDS nodes are selected in such a way that if only the CDS nodes transmit, all nodes in the network will receive message. The method of [VO-CDS & PBSM] has same reliability because they both use CDS-based flooding in a static network. The method of [VO] uses scoped flooding. Under similar condition (i.e. same parameter values), all the methods show a higher reliability when ideal MAC layer is used. In this case the reliability reduces due to collision as simplified MAC layer is used. Although the methods of [VO, CEM] generates more number of transmission and encounter more collision than PBSM, their reliability is close to PBSM because the increased number of transmission make up for loss of reliability due to collision.

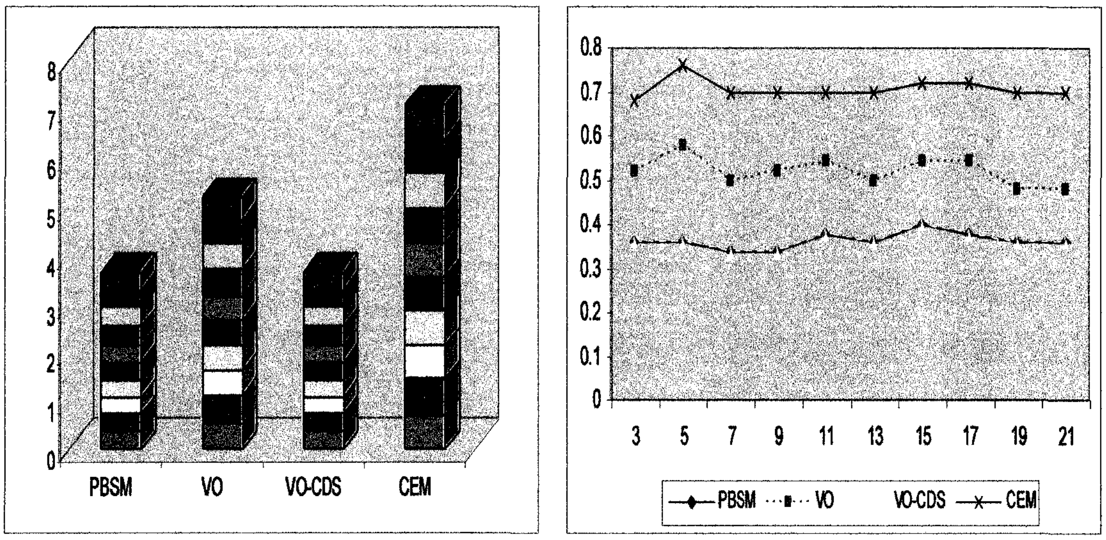
The above line graph shows the percentage of reliability (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, the graph won't change as nodes are static. Consequently, the percentage of reliability remains fairly constant as *B* increases. The data table is shown below:

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	98.3	98.3	98.3	99.4	3 (1)
	98.2	100	98.2	99.3	5 (2)
	96.8	97	96.8	99	7 (3)
	96.8	97	96.8	98.2	9 (4)
	96.8	97	96.8	99	11 (5)
	96.8	97	96.8	98.2	13 (6)
	96.8	97	96.8	99	15 (7)
	96.8	97	96.8	99	17 (8)
	96.8	97	96.8	99	19 (9)
	96.8	97	96.8	99	21 (10)
Total	970.9	974.3	970.9	989.1	
Average	97.09	97.43	97.09	98.91	
ConfidenceInterval95%	97.1±0.38	97.4±0.61	97.1±0.38	98.9±0.25	

Table 39: Percentage of reliability in a static network (D=0, B=10, n=50, d=4)

B. Considering High Density (*d*=10) in static scenario

Total number of transmissions per node



(A)

(B)

Figure 42: Total number of transmissions per node in a static scenario
(D=0, B=10, n=50, d=10)

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than the methods of [VO] and [CEM]. There are two main reasons which are stated in section 5.6.1. The performance of PBSM and VO-CDS is same as both methods use CDS-based flooding in static networks. The number of transmissions per node in PBSM is around 30% less than the method of [VO]. At higher density there are more transmissions as more nodes receive message from a single transmission. In static network, PBSM only uses CDS-based flooding and the percentage of CDS nodes compared to all nodes decreases as density increases. Since in PBSM (also VO-CDS), the CDS nodes transmit most of the time, the difference between number of transmissions between PBSM (also VO-CDS) and other methods increases at higher density.

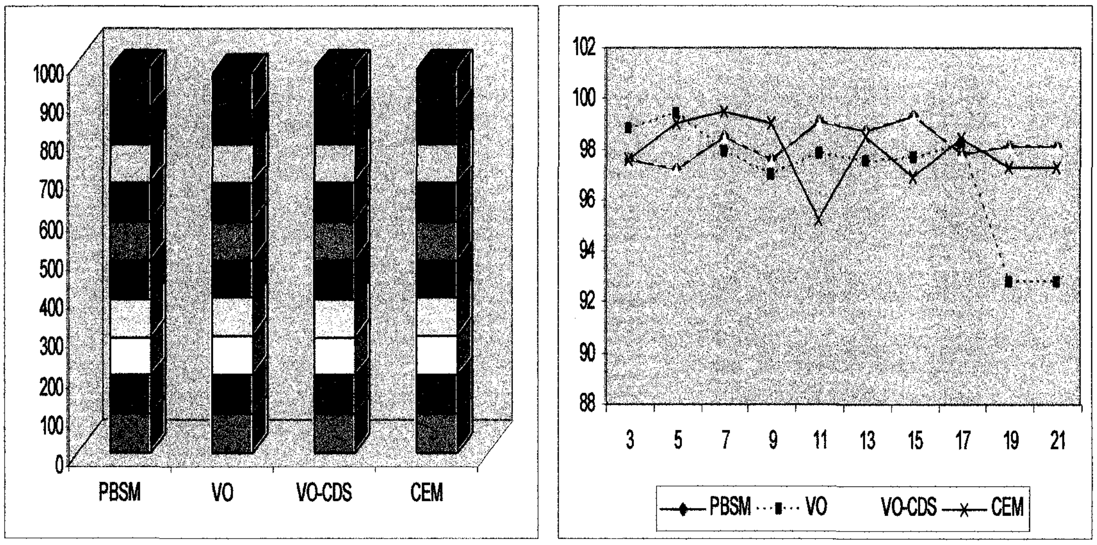
The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, the graph won't change as nodes are static. Consequently, the number of transmissions remains fairly constant as *B* increases. The data table is shown below:

	PBSM	**VO**	**VO-CDS**	**CEM**	**B (No. of Simulation)**
	0.36	0.52	0.36	0.68	3 (1)
	0.36	0.58	0.36	0.76	5 (2)
	0.34	0.5	0.34	0.7	7 (3)
	0.34	0.52	0.34	0.7	9 (4)
	0.38	0.54	0.38	0.7	11 (5)
	0.36	0.5	0.36	0.7	13 (6)
	0.4	0.54	0.4	0.72	15 (7)
	0.38	0.54	0.38	0.72	17 (8)
	0.36	0.48	0.36	0.7	19 (9)
	0.36	0.48	0.36	0.7	21 (10)
Total	3.64	5.2	3.64	7.08	
Average	0.364	0.52	0.364	0.708	
ConfidenceInterval95%	0.36±0.01	0.52±0.02	0.36±0.01	0.71±0.01	

Table 40: Total Number of Transmissions per node in a static network

(D=0, B=10, n=50, d=10)

Percentage of reliability



(A) (B)

Figure 43: Percentage of reliability for a static network (D=0, B=10, n=50, d=10)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). In a static network, all protocols are around 98% reliable.
In the proposed PBSM method, the CDS nodes are selected in such a way that if only the CDS nodes transmit, all nodes in the network will receive message. The method of [VO-CDS & PBSM] has same reliability because they both use CDS-based flooding in a static network. The method of [VO] uses scoped flooding. At high density, a node has more neighbors. Therefore, even if there is collision among transmission from few neighbors, a node can receive transmission from other neighbors. Consequently, reliability of all methods increases slightly.

The above line graph shows the percentage of reliability (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, the graph won't change as nodes are static. Consequently, the percentage of reliability remains fairly constant as *B* increases. The data table is shown below:

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	97.55	98.8	97.55	97.6	3 (1)
	97.2	99.4	97.2	99	5 (2)
	98.5	97.9	98.5	99.5	7 (3)
	97.6	97	97.6	99	9 (4)
	99.1	97.8	99.1	95.2	11 (5)
	98.7	97.5	98.7	98.5	13 (6)
	99.3	97.7	99.3	96.9	15 (7)
	97.8	98.2	97.8	98.4	17 (8)
	98.1	92.8	98.1	97.3	19 (9)
	98.1	92.8	98.1	97.3	21 (10)
Total	981.95	969.9	981.95	978.7	
Average	98.195	96.99	98.195	97.87	
ConfidenceInterval95%	98.2±0.43	97±1.43	98.2±0.43	97.9±0.79	

Table 41: Percentage of reliability in a static network (D=0, B=10, n=50, d=10)

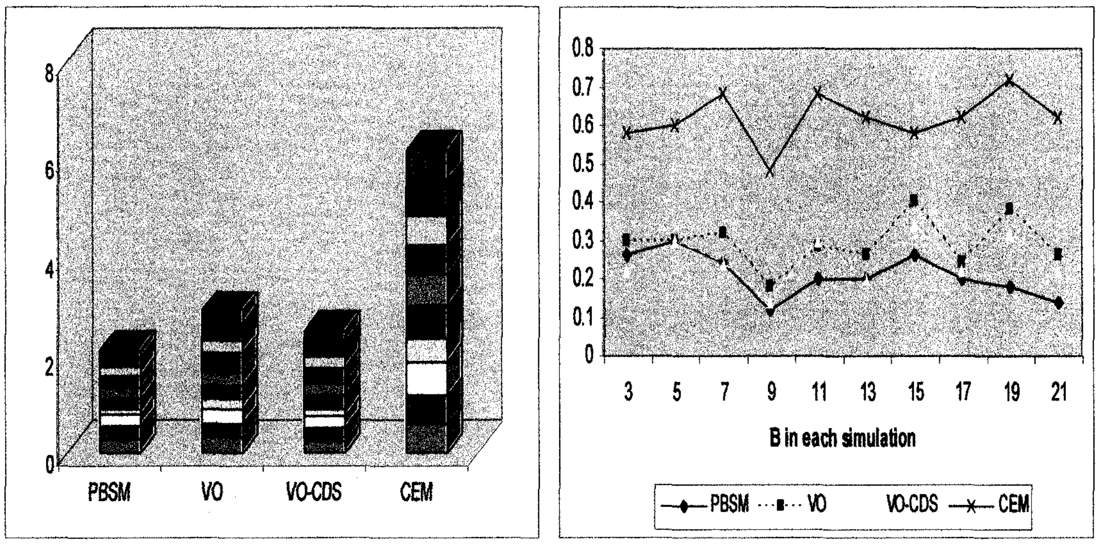
5.7.2 Network of Moderate Mobility

Under maximal speed, a node can travel distance *D'* between two hello messages. Let *D'=D*r,* where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is *D*r /H* (=*D'/H*). Consequently, speed and *D'* depend on *D* and *D* is used as a parameter to define mobility.

In a moderate mobility scenario, half of the nodes move with moderate speed [*D*=1/2, so max speed = *r*/ (2*H*)]. The remaining nodes are static. In each simulation, the number of nodes (n) is 50. The hello message time (*H*) is incremented by 2 in each simulation. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit. Therefore, *B* (the number of non-overlapping messages that can be sent between two hello messages) is equal to *H*. Each simulation consists of a randomly generated graph and ten hello messages.

A. Considering Medium Density (*d*=4) in moderately mobile scenario

Total number of transmissions per node



(A) (B)

Figure 44: Total no. of transmissions per node in a moderate network (D=1/2, n=50, d=4)

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 28% and 16% lower than the methods of VO and VO-CDS respectively. The general reasons for the above data are explained in section 5.6.2. Under similar condition (i.e. same parameter values), all the methods generated more transmissions when ideal MAC layer is used. This is because in simplified MAC layer, a node has to delay its transmission if at least one of its neighbors is currently transmitting (exposed node problem) and therefore it might not have the opportunity to transmit if broadcast time is over.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there are more changes that occur between two hello messages. That is, the nodes will move by larger distances, there will be more broadcast messages, and there will be more discoveries of new nodes between subsequent hello messages. Compared to other methods, PBSM saves more transmissions since it does not consider broadcasting every time two nodes discover each other. Consequently, the difference between PBSM and the other methods increases as *B* increases. The data tables are given below:

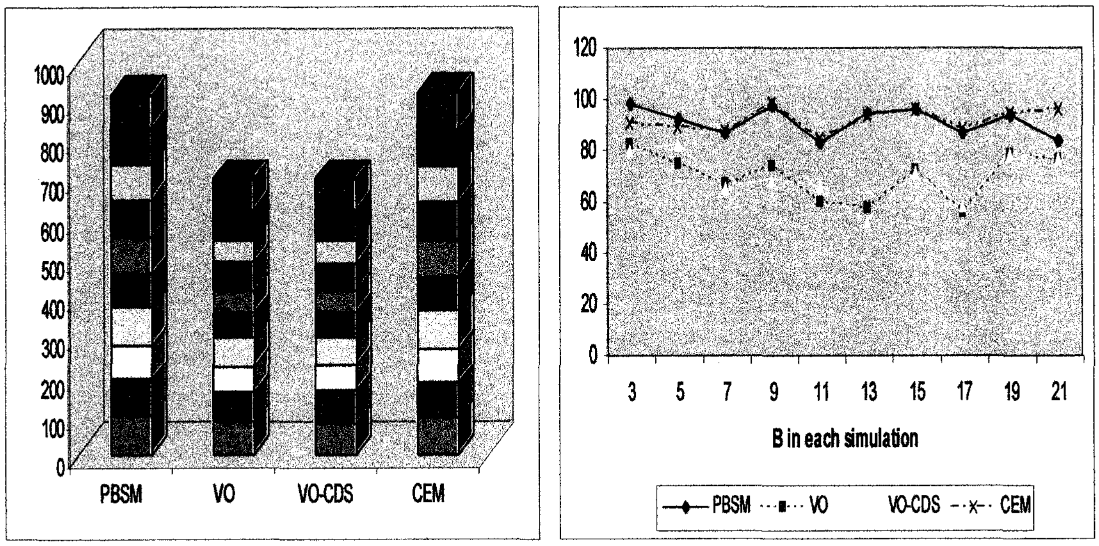
	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	0.26	0.3	0.22	0.58	3 (1)
	0.3	0.3	0.3	0.6	5 (2)
	0.24	0.32	0.24	0.68	7 (3)
	0.12	0.18	0.14	0.48	9 (4)
	0.2	0.28	0.3	0.68	11 (5)
	0.2	0.26	0.2	0.62	13 (6)
	0.26	0.4	0.34	0.58	15 (7)
	0.2	0.24	0.22	0.62	17 (8)
	0.18	0.38	0.32	0.72	19 (9)
	0.14	0.26	0.22	0.62	21 (10)
Total	2.1	2.92	2.5	6.18	
Average	0.21	0.292	0.25	0.618	
95%ConfidenceInterval	0.21±0.03	0.29±0.04	0.25±0.04	0.62±0.04	

Table 42: Total transmissions per node in a moderate network (D=1/2, n=50, d=4)

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	13	15	11	29	3 (1)
	15	15	15	30	5 (2)
	12	16	12	34	7 (3)
	6	9	7	24	9 (4)
	10	14	15	34	11 (5)
	10	13	10	31	13 (6)
	13	20	17	29	15 (7)
	10	12	11	31	17 (8)
	9	19	16	36	19 (9)
	7	13	11	31	21 (10)
Total	**105**	**146**	**125**	**309**	
Average	**10.5**	**14.6**	**12.5**	**30.9**	

Table 43: Total number of transmissions in a moderate network (D=1/2, n=50, d=4)

Percentage of reliability



(A) (B)

Figure 45: Percentage of reliability in a moderately mobile network (D=1/2, n=50, d=4)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. Each color in the bar graph represents a simulation. In a network of moderate mobility, both PBSM and CEM protocol are nearly 91% reliable. The VO and VO-CDS protocols are nearly 70% reliable. This is depicted in the bar graph. The general reasons for the above data are explained in section 5.6.2.

Under similar condition (i.e. same parameter values), all the methods show a higher reliability when ideal MAC layer is used. In this case the reliability reduces due to collision as simplified MAC layer is used. Furthermore, the other methods encounter more collision than PBSM because they generate more number of transmissions. Consequently, the difference in reliability between PBSM and other methods is greater than similar scenario (sec 5.6.2) where ideal MAC layer is used. Although the method CEM generates much more number of transmission and encounter more collision than other methods, its reliability is close to PBSM because the increased number of transmission make up for loss of reliability due to collision.

The above line graph shows the reliability percentage (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there will be more changes occurring between the two hello messages. That is, the nodes will move by larger distances, there will be more broadcast messages and more discoveries of new nodes between subsequent hello messages. The discoveries of new node are compensated by the disappearance of old nodes. Thus, the reliability of all protocols remains fairly constant as *B* increases. The data table is given below:

	PBSM	**VO**	**VO-CDS**	**CEM**	**B (No of simulation)**
	98.2	82	80	91	3 (1)
	92	75	82	89	5 (2)
	87.1	67.2	65.4	87.8	7 (3)
	97.8	74.1	68.7	98.4	9 (4)
	82.9	60	66.4	84.9	11 (5)
	94.7	57.7	52.2	94.1	13 (6)
	96.2	72.2	71.6	95.9	15 (7)
	87.1	56	58.6	88.67	17 (8)
	94.2	78.8	78.5	94.8	19 (9)
	84.1	76.5	76.9	96.1	21 (10)
Total	914.3	699.5	700.3	920.67	
Average	91.43	69.95	70.03	92.067	
ConfidenceInterval95%	91.4±3.53	70±5.72	70±6.02	92.1±2.73	

Table 44: Percentage of reliability in a moderately mobile network (D=1/2, n=50, d=4)

B. Considering High Density (*d*=10) in moderately mobile scenario

Total number of transmissions per node

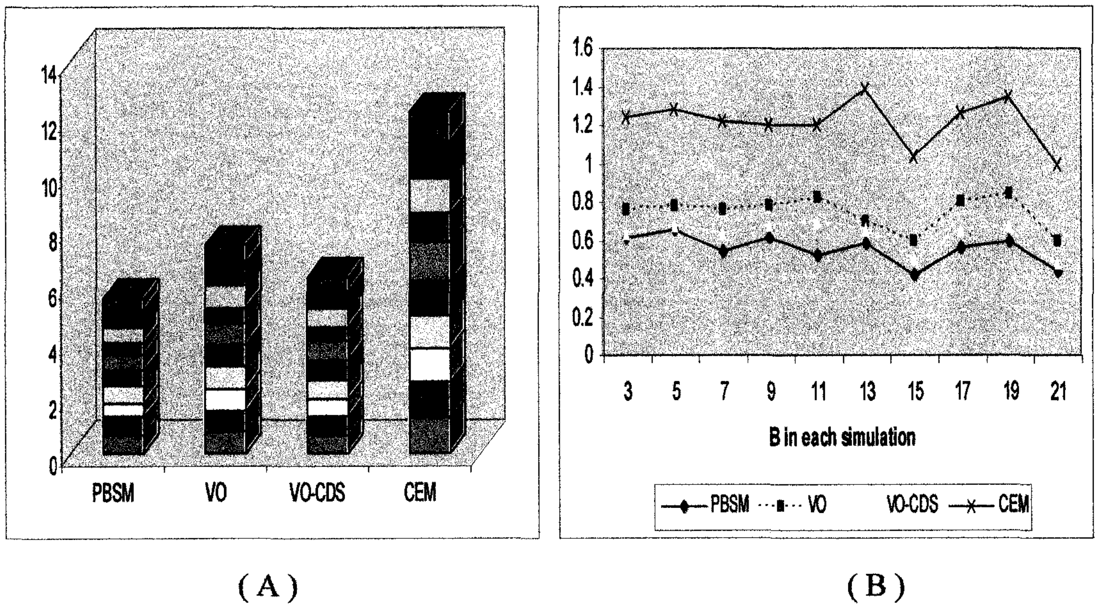


Figure 46: Total no. of transmissions per node in a moderate network (D=1/2,n=50,d=10)

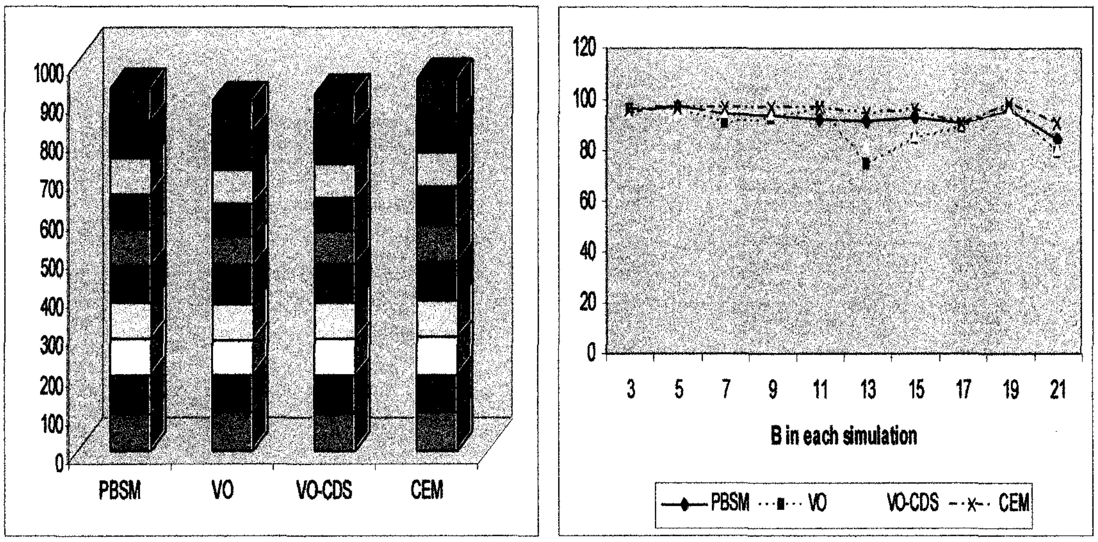
The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 25% and 12% lower than the methods of VO and VO-CDS respectively. The general reasons for the above data are explained in section 5.6.2. At higher density there are more transmissions as more nodes receive message from a single transmission.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. At high density, as *B* increases there will be more discoveries of new node. Therefore, more nodes will decide to transmit and some nodes might not have the opportunity to transmit due to exposed node problem. Consequently, the number of transmissions of all methods decreases slightly as *B* increases. The data table is given below:

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	0.62	0.76	0.64	1.24	3 (1)
	0.66	0.78	0.68	1.28	5 (2)
	0.54	0.76	0.64	1.22	7 (3)
	0.62	0.78	0.68	1.2	9 (4)
	0.52	0.82	0.7	1.2	11 (5)
	0.58	0.7	0.66	1.38	13 (6)
	0.42	0.6	0.5	1.04	15 (7)
	0.56	0.8	0.66	1.26	17 (8)
	0.6	0.84	0.68	1.34	19 (9)
	0.44	0.6	0.48	1	21 (10)
Total	5.56	7.44	6.32	12.16	
Average	0.556	0.744	0.632	1.216	
95%ConfidenceInterval	0.56±0.05	0.74±0.05	0.63±0.05	1.22±0.07	

Table 45: Total transmissions per node in a moderate network (D=1/2, n=50, d=10)

Percentage of reliability



(A) (B)

Figure 47: Percentage of reliability in a moderately mobile network (D=1/2, n=50, d=10)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). In a network of moderate mobility, all protocols are nearly

92% reliable. The general reasons for the above data are explained in section 5.6.2. At high density, a node has more neighbors. Therefore, even if there is collision among transmission from few neighbors, a node can receive transmission from other neighbors. Consequently, reliability of all methods increases and the difference between PBSM and other method decreases.

The above line graph shows the reliability percentage (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As graph is dense and the nodes are moderately mobile, the neighborhood of a node does not change much as *B* increases. Thus, the reliability of all protocols is fairly constant as *B* increases.

	PBSM	**VO**	**VO-CDS**	**CEM**	**B (No of simulation)**
	100	100	100	100	3 (1)
	90	90	90	90	5 (2)
	80	80	89	98	7 (3)
	100	100	100	100	9 (4)
	70	93	91	98	11 (5)
	100	100	100	100	13 (6)
	100	100	100	100	15 (7)
	98	95	90	90	17 (8)
	77.5	80	100	100	19 (9)
	93.6	68	68	100	21 (10)
Total	909.1	906	928	976	
Average	90.91	90.6	92.8	97.6	
ConfidenceInterval95%	90.9±6.93	90.6±6.9	92.8±6.22	97.6±2.53	

Table 46: Percentage of reliability in a moderately mobile network (D=1/2, n=50, d=10)

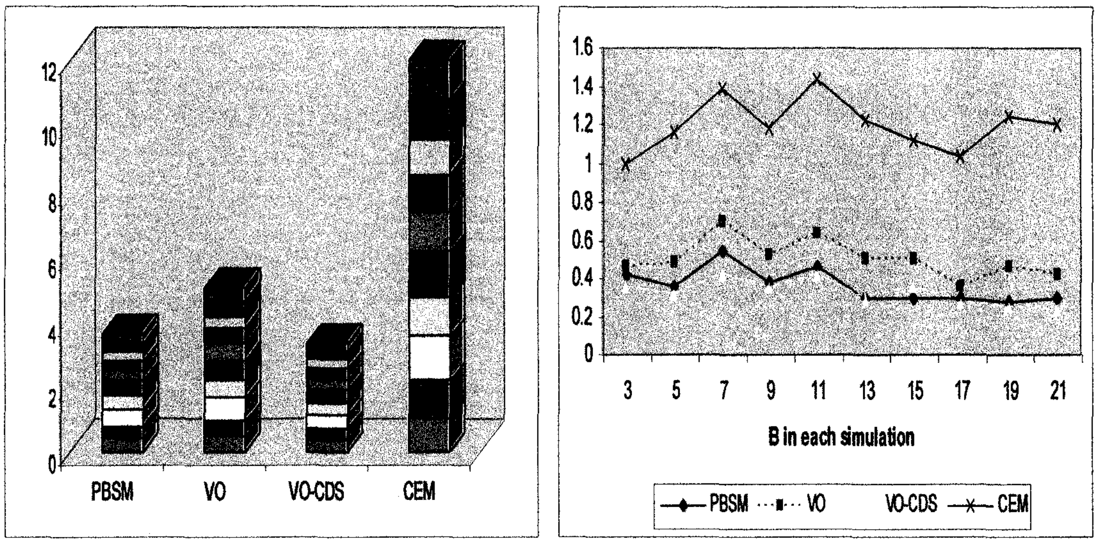
5.7.3 Highly Mobile Network

Under maximal speed, a node can travel distance *D'* between two hello messages. Let *D'=D*r,* where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is *D*r /H (=D'/H)*. The average velocity is maximum speed/2. Consequently, both speed and *D'* depend on *D* and *D* is used as a parameter to define mobility.

In a high mobility scenario, all nodes move. [*D*=1, so max speed = *r*/ *H*]. In each simulation, the number of nodes (n) is 50. The hello message time (*H*) is incremented by 2 in each simulation. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit. Therefore, *B* (the number of non-overlapping messages that can be sent between two hello messages) is equal to *H*. Each simulation consists of a randomly generated graph and ten hello messages.

A. Considering Medium Density (*d*=4) in highly mobile scenario

Total number of transmissions per node



(A) (B)

Figure 48: Total number of transmissions per node in a highly mobile network (D=1, n=50, d=4)

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 28% lower than the method of VO. The general reasons for the above data are explained in section 5.6.3. Under similar condition (i.e. same parameter values), all the methods generated more transmissions when ideal MAC layer is used. This is because in simplified MAC layer, a node has to delay its transmission if at least one of its neighbors is currently transmitting (exposed node problem) and therefore it might not have the opportunity to transmit if broadcast time is over.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there will be more discoveries of new nodes between subsequent hello messages. Therefore, more nodes will decide to transmit and some nodes might not have the opportunity to transmit due to exposed node problem. Consequently, the number of transmissions of all methods decreases slightly as *B* increases. The data tables are given below:

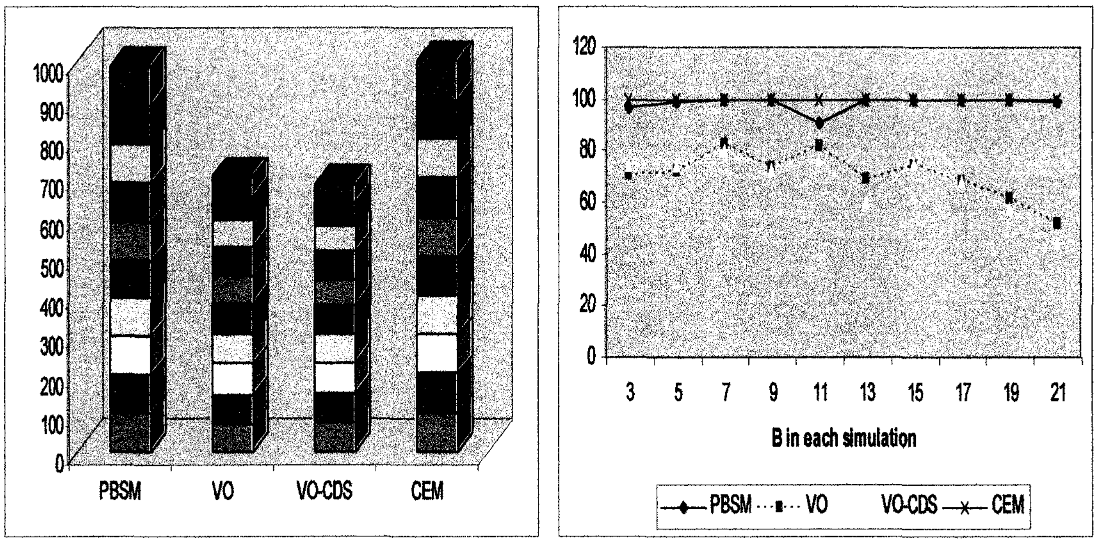
	PBSM	**VO**	**VO-CDS**	**CEM**	**B (No of simulation)**
	0.42	0.46	0.36	1	3 (1)
	0.36	0.48	0.32	1.16	5 (2)
	0.54	0.7	0.42	1.38	7 (3)
	0.38	0.52	0.36	1.18	9 (4)
	0.46	0.64	0.42	1.44	11 (5)
	0.3	0.5	0.3	1.22	13 (6)
	0.3	0.5	0.36	1.12	15 (7)
	0.3	0.36	0.28	1.04	17 (8)
	0.28	0.46	0.26	1.24	19 (9)
	0.3	0.42	0.24	1.2	21 (10)
Total	3.64	5.04	3.32	11.98	
Average	0.364	0.504	0.332	1.198	
ConfidenceInterval95%	0.36±0.05	0.5±0.06	0.33±0.04	1.2±0.08	

Table 47: Total number of transmissions per node in a highly mobile network (D=1, n=50, d=4)

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	21	23	18	50	3 (1)
	18	24	16	58	5 (2)
	27	35	21	69	7 (3)
	19	26	18	59	9 (4)
	23	32	21	72	11 (5)
	15	25	15	61	13 (6)
	15	25	18	56	15 (7)
	15	18	14	52	17 (8)
	14	23	13	62	19 (9)
	15	21	12	60	21 (10)
Total	182	252	166	599	
Avg	18.2	25.2	16.6	59.9	

Table 48: Total number of transmissions in a highly mobile network (D=1, n=50, d=4)

Percentage of reliability



(A) (B)

Figure 49: Percentage of reliability in a highly mobile network (D=1, n=50, d=4)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes receiving the message. In a network of high mobility, PBSM and CEM are about 99% reliable. The VO and VO-CDS protocols are nearly 70% reliable. This is depicted in the bar graph. The general reasons for the above data are mentioned in section 5.6.3. In this network, the nodes are moving at higher speed and therefore there are more disappearances and discoveries of nodes. The discoveries of node will lead to more transmission and collision. Consequently, the difference in reliability between PBSM and other methods is more than the moderately mobile network.

Under similar condition (i.e. same parameter values), the methods [VO, VO-CDS] show a higher reliability when ideal MAC layer is used. In this case the reliability reduces due to collision as simplified MAC layer is used. The other methods encounter more collision than PBSM because they generate more number of transmissions. Consequently, the difference in reliability between PBSM and the other methods is greater than similar scenario (sec 5.6.3) where ideal MAC layer is used. Although the method CEM generates much more number of transmission and encounter more collision than other methods, its reliability is better than [VO, VO-CDS] because the increased number of transmission make up for loss of reliability due to collision.

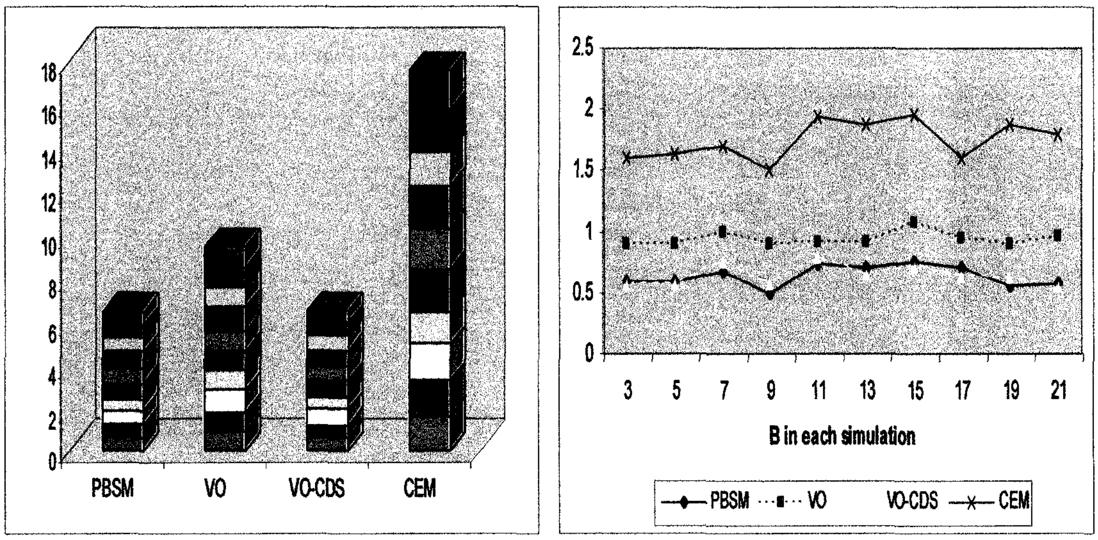
The above line graph shows the reliability percentage (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there will be more discoveries of new nodes between subsequent hello messages. Therefore, more nodes will decide to transmit in other methods. Due to exposed node problem, an important transmission might not occur as it has to wait for some redundant transmissions. Thus, the reliability of [VO, VO-CDS] decreases as *B* increases. Unlike other methods, in PBSM nodes do not transmit every time they discover each other and the reliability of PBSM remains fairly constant. The data table is given below:

	PBSM	VO	VOCDS	CEM	B (No of simulation)
	96.8	70.6	74.1	100	3 (1)
	99.3	72.2	74.3	100	5 (2)
	99.8	82.6	80	99.4	7 (3)
	100	73.4	72.7	100	9 (4)
	90.67	81.67	76.4	100	11 (5)
	100	69	60	100	13 (6)
	100	74.3	74.7	100	15 (7)
	100	68.1	68.1	100	17 (8)
	99.8	61.8	57.5	100	19 (9)
	99.3	51.9	44.3	99.7	21 (10)
Total	985.67	705.57	682.1	999.1	
Average	98.567	70.557	68.21	99.91	
ConfidenceInterval95%	98.6±1.82	70.6±5.57	68.2±6.83	99.9±0.13	

Table 49: Percentage of reliability in a highly mobile network (D=1, n=50, d=4)

B. Considering High Density (*d*=10) in highly mobile scenario

Total number of transmissions per node



(A) (B)

Figure 50: Total number of transmissions per node in a highly mobile network (D=1, n=50, d=10)

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 32% lower than the method of VO. The general reasons for the above data are explained in section 5.6.3. At higher density there are more transmissions as more nodes receive message from a single transmission.

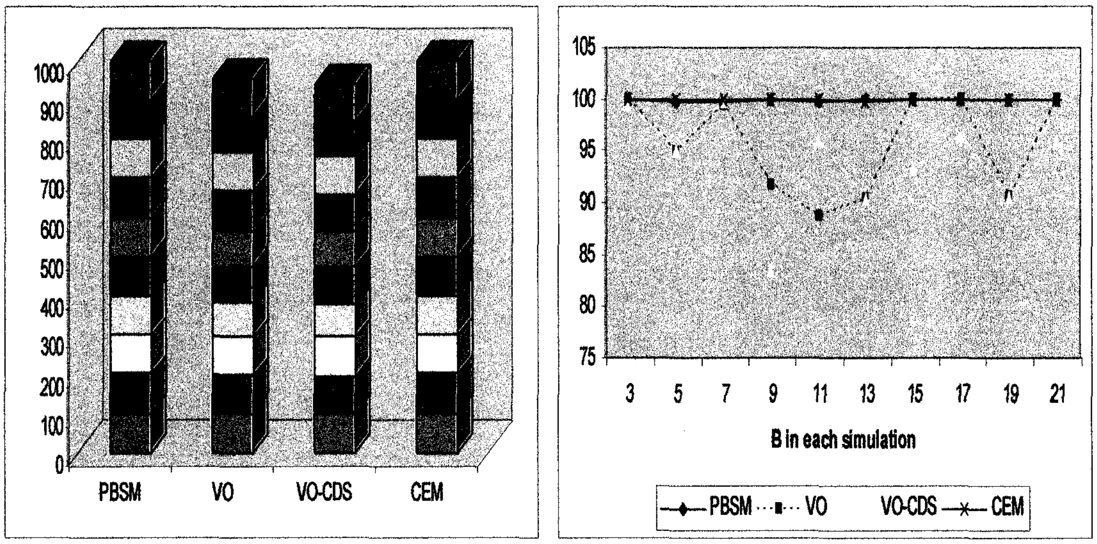
The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there will be more discoveries of new nodes between subsequent hello messages. Therefore, more nodes will decide to transmit. However, as density is high, a node has many neighbors and nodes (especially with longer timeout) might not have time to transmit due to exposed node problem. Consequently, the number of transmissions remains fairly constant as *B* increases. The data table is shown below:

	PBSM	VO	VO-CDS	CEM	B (No of simulation)
	0.6	0.9	0.58	1.6	3 (1)
	0.6	0.9	0.58	1.64	5 (2)
	0.68	1	0.76	1.7	7 (3)
	0.5	0.9	0.58	1.5	9 (4)
	0.74	0.92	0.78	1.94	11 (5)
	0.7	0.92	0.66	1.88	13 (6)
	0.76	1.08	0.72	1.96	15 (7)
	0.7	0.94	0.64	1.6	17 (8)
	0.56	0.9	0.62	1.88	19 (9)
	0.58	0.96	0.56	1.8	21 (10)
Total	6.42	9.42	6.48	17.5	
Average	0.642	0.942	0.648	1.75	
ConfidenceInterval95%	0.64±0.05	0.94±0.04	0.65±0.05	1.75±0.1	

Table 50: Total number of transmissions per node in a highly mobile network

(D=1, n=50, d=10)

Percentage of reliability



(A)

(B)

Figure 51: Percentage of reliability in a highly mobile network (D=1, n=50, d=10)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). In a network of high mobility, PBSM and CEM protocols are about 99% reliable whereas VO and VO-CDS protocols are about 95% reliable. The general reasons for the above data are mentioned in section 5.6.3. At high density, a node has more neighbors. Therefore, even if there is collision among transmission from few neighbors, a node can receive transmission from other neighbors. Consequently, reliability of all methods increases and the difference between PBSM and other method decreases.

The above line graph shows the reliability percentage (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *B* used in each simulation. As *B* increases, there will be more discoveries of new nodes between subsequent hello messages. The discoveries of new nodes are compensated by the disappearance of nodes. Thus, the reliability of all protocols remains fairly constant as *B* increases. The data table is given below:

	PBSM	VO	VOCDS	CEM	B (No of simulation)
	100	100	99.4	100	3 (1)
	99.8	94.8	95	100	5 (2)
	99.8	99.4	99.8	100	7 (3)
	100	91.6	83.5	100	9 (4)
	99.8	88.6	96	100	11 (5)
	100	90.4	90.4	99.8	13 (6)
	100	100	93.3	100	15 (7)
	100	100	96.4	100	17 (8)
	100	90.6	90.6	100	19 (9)
	100	100	96	100	21 (10)
Total	999.4	955.4	940.4	999.8	
Average	99.94	95.54	94.04	99.98	
ConfidenceInterval95%	99.9±0.06	95.5±2.99	94±3.01	100±0.04	

Table 51: Percentage of reliability in a highly mobile network (D=1, n=50, d=10)

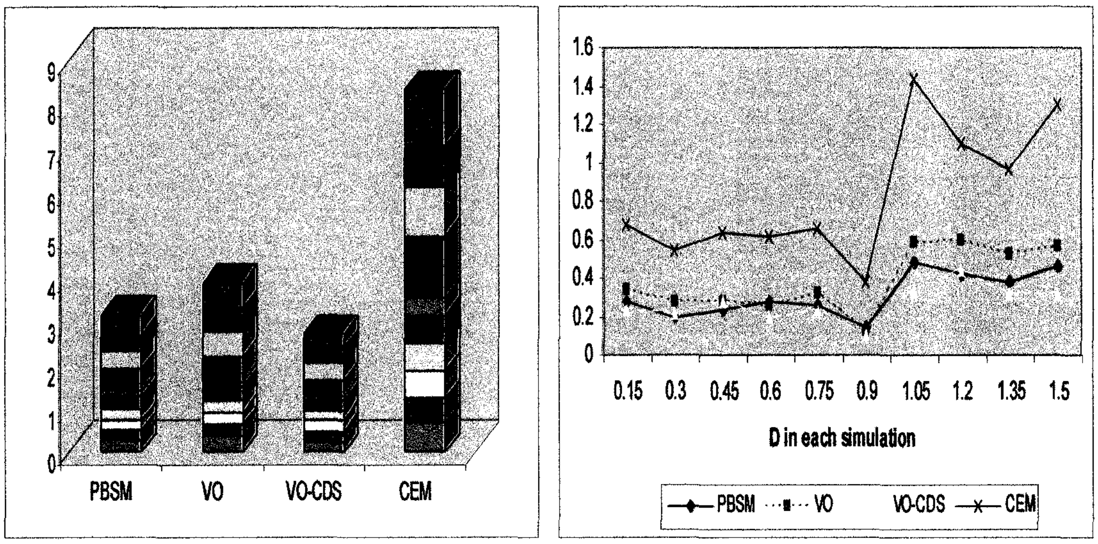
5.7.4 Comparing performance based on mobility (*D*)

Under maximal speed, a node can travel distance *D'* between two hello messages. Let $D'=D*r$, where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is $D*r\ /H$ ($=D'/H$). The average velocity is maximum speed/2. Consequently, both speed and *D'* depend on *D* and *D* is used as a parameter to define mobility.

The performance of the protocols is compared based on mobility. In each simulation, the number of nodes (*n*) is 50 and the hello message time (*H*) is 5. Sending a broadcast message to an immediate neighbor requires fixed time *B'*. $B = H/B'$ (number of non-overlapping messages that can be sent between two hello messages where $B' = 1$ and *H* is 5). The parameter *D* indicates the maximum speed of the node. *D* is incremented by 0.15 in each simulation. When *D* is less than 1, half of the nodes move. When *D* >= 1 (high mobility), all nodes move. Each simulation consists of a randomly generated graph and ten hello messages.

A. Considering Medium Density (*d*=4) in mobility based scenario

Total number of transmissions per node



(A) (B)

Figure 52: Total number of transmissions per node in a mobility-based scenario (B=5, n=50, d=4)

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 18% less than protocol of VO. Under similar condition (i.e. same parameter values), all the methods generated more transmissions when ideal MAC layer is used. This is because in simplified MAC layer, a node has to delay its transmission if at least one of its neighbors is currently transmitting (exposed node problem) and therefore it might not have the opportunity to transmit if broadcast time is over.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *D* used in each simulation. As *D* increases, the discoveries of new node increases and therefore the number of transmission increases. The data tables are given below:

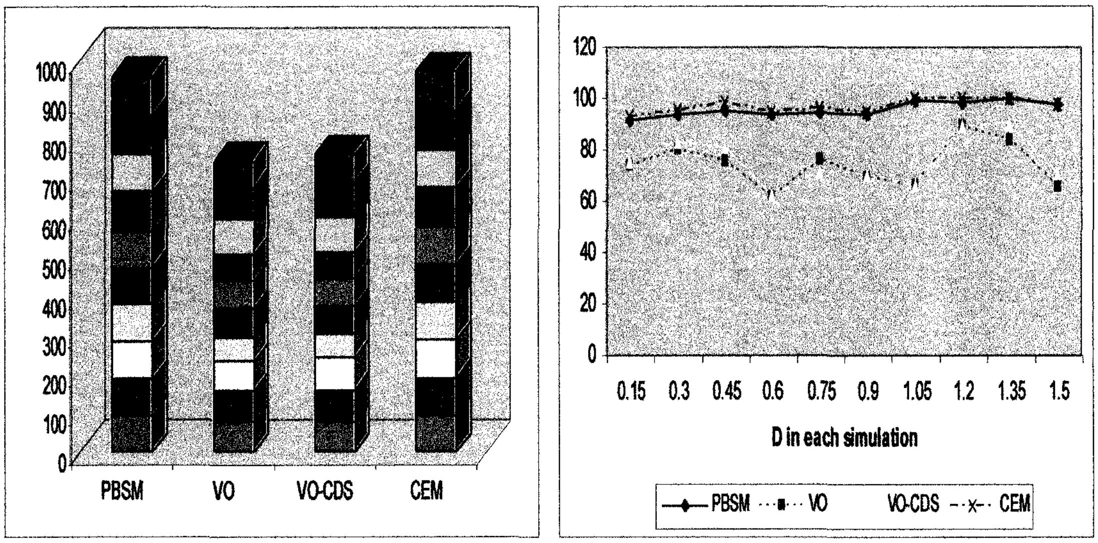
	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	0.28	0.34	0.24	0.68	0.15 (1)
	0.2	0.28	0.22	0.54	0.3 (2)
	0.24	0.28	0.28	0.64	0.45 (3)
	0.28	0.26	0.2	0.62	0.6 (4)
	0.26	0.32	0.24	0.66	0.75 (5)
	0.14	0.12	0.1	0.38	0.9 (6)
	0.48	0.58	0.34	1.44	1.05 (7)
	0.42	0.6	0.44	1.1	1.2 (8)
	0.38	0.52	0.32	0.96	1.35 (9)
	0.46	0.56	0.32	1.3	1.5 (10)
Total	3.14	3.86	2.7	8.32	
Average	0.314	0.386	0.27	0.832	
ConfidenceInterval95%	0.31±0.07	0.39±0.1	0.27±0.06	0.83±0.22	

Table 52: Total number of transmissions per node in a mobility-based scenario (B=5, n=50, d=4)

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	14	17	12	34	0.15 (1)
	10	14	11	27	0.3 (2)
	12	14	14	32	0.45 (3)
	14	13	10	31	0.6 (4)
	13	16	12	33	0.75 (5)
	7	6	5	19	0.9 (6)
	24	29	17	72	1.05 (7)
	21	30	22	55	1.2 (8)
	19	26	16	48	1.35 (9)
	23	28	16	65	1.5 (10)
Total	157	193	135	416	
Average	15.7	19.3	13.5	41.6	

Table 53: Total number of transmissions in a mobility-based scenario (B=5, n=50, d=4)

Percentage of reliability



(A) (B)

Figure 53: Percentage of reliability in mobility-based scenario (B=5, n=50, d=4)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). It is a measure of the number of nodes that receive the message. Each color in the bar graph represents a simulation. PBSM and CEM are about 96% reliable whereas VO and VO-CDS protocols are nearly 75% reliable.

Under similar condition (i.e. same parameter values), all the methods show a higher reliability when ideal MAC layer is used. In this case the reliability reduces due to collision as simplified MAC layer is used. The other methods encounter more collision than PBSM because they generate more number of transmissions. Consequently, the difference in reliability between PBSM and the other methods is greater than similar scenario (sec 5.6.4) where ideal MAC layer is used. Although the method CEM generates much more number of transmission and encounter more collision than PBSM, its reliability is close to PBSM because the increased number of transmission make up for loss of reliability due to collision.

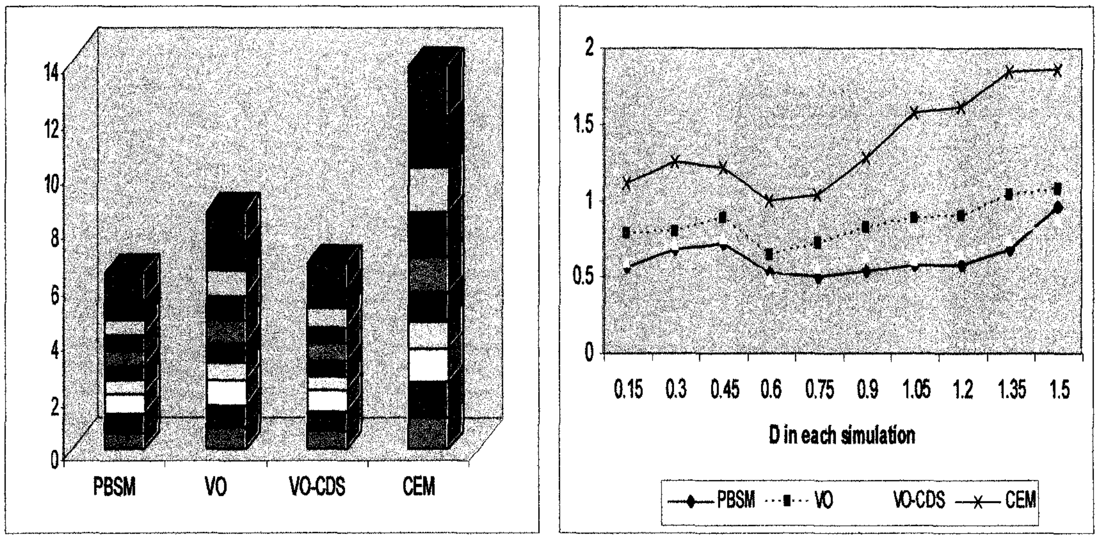
The above line graph shows the percentage of reliability (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *D* used in each simulation. As *D* increases there are more discoveries of new node but it is compensated by the disappearance of old nodes. Thus, the reliability of all protocols remains constant.

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	91.6	74.1	75.4	92.8	0.15 (1)
	94.1	79.7	82.9	95.6	0.3 (2)
	95.4	75.58	80.85	98.46	0.45 (3)
	94	61.3	62.1	94.39	0.6 (4)
	95	76.5	71.2	96.5	0.75 (5)
	93.8	69.6	70	94.6	0.9 (6)
	99.3	65.89	65.89	100	1.05 (7)
	98.1	89.4	89.4	100	1.2 (8)
	100	83.7	89.7	100	1.35 (9)
	97.5	65.1	70.7	97.5	1.5 (10)
Total	958.8	740.87	758.14	969.85	
Average	95.88	74.087	75.814	96.985	
ConfidenceInterval95%	95.9±1.69	74.1±5.46	75.8±5.92	97±1.62	

Table 54: Percentage of reliability in a mobility-based scenario (B=5, n=50, d=4)

B. Considering High Density (*d*=10) in mobility based scenario

Total number of transmissions per node



(A) (B)

Figure 54: Total number of transmissions per node in a mobility-based scenario (B=5, n=50, d=10)

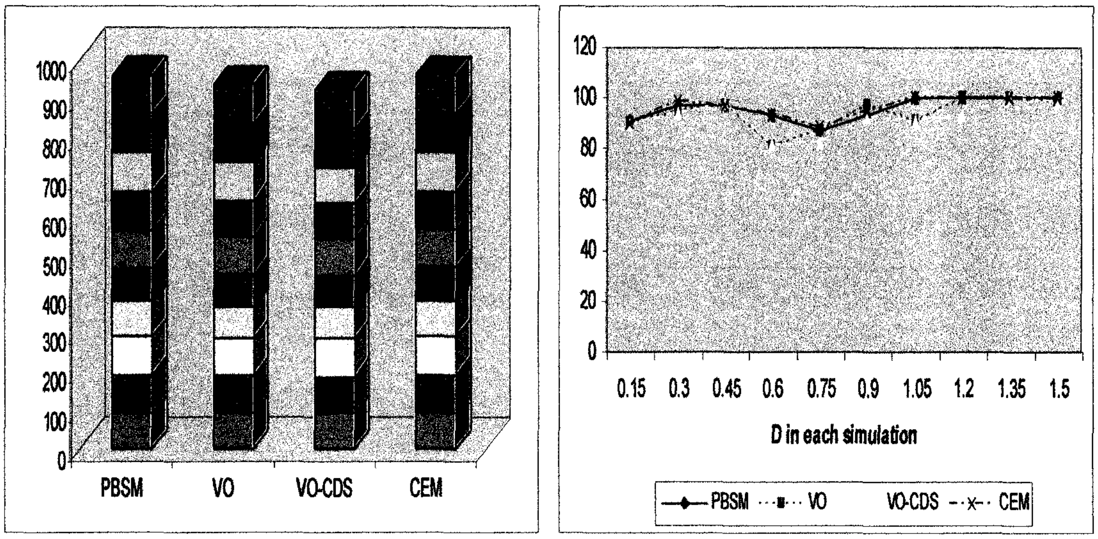
The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 26% less than protocol of VO. At higher density there are more transmissions as more nodes receive message from a single transmission. In PBSM, a node will not decide to transmit if all its neighbors are covered. The percentage of CDS nodes compared to all nodes decreases as density increases. Since in PBSM (also VO-CDS), the CDS nodes transmit most of the time, the difference between number of transmissions between PBSM (also VO-CDS) and other methods increases at higher density.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *D* used in each simulation. As *D* increases, there will be more discoveries of new nodes between subsequent hello messages. Therefore, more nodes will decide to transmit in other methods. However, in PBSM (also VO-CDS), the newly discovered node will be in R list most of the time as density is high. Thus, the number of transmissions in PBSM and VO-CDS does not increase appreciably as *D* increases. Consequently, the difference in number of transmissions between PBSM and the methods of [VO, VO-CDS] increases as *D* increases. The data table is shown below:

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	0.56	0.78	0.6	1.12	0.15 (1)
	0.68	0.8	0.72	1.26	0.3 (2)
	0.72	0.88	0.76	1.22	0.45 (3)
	0.52	0.64	0.5	1	0.6 (4)
	0.5	0.72	0.56	1.04	0.75 (5)
	0.54	0.82	0.64	1.28	0.9 (6)
	0.58	0.88	0.62	1.58	1.05 (7)
	0.58	0.9	0.66	1.62	1.2 (8)
	0.68	1.04	0.74	1.84	1.35 (9)
	0.96	1.08	0.88	1.86	1.5 (10)
Total	6.32	8.54	6.68	13.82	
Average	0.632	0.854	0.668	1.382	
ConfidenceInterval95%	0.63±0.08	0.85±0.08	0.67±0.07	1.38±0.2	

Table 55: Total number of transmissions per node in a mobility-based scenario (B=5, n=50, d=10)

Percentage of reliability



(A) (B)

Figure 55: Percentage of reliability in mobility-based scenario (B=5, n=50, d=10)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). PBSM and CEM are nearly 96% reliable. VO and VO-CDS protocols are nearly 92% reliable. At high density, a node has more neighbors. Therefore, even if there is collision among transmission from few neighbors, a node can receive transmission from other neighbors. Consequently, reliability of all methods increases and the difference between PBSM and other method decreases.

The above line graph shows the percentage of reliability (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *D* used in each simulation. As *D* increases, there will be more discoveries of new nodes. The additional transmissions will lead to more collisions but overall more nodes will receive message as density is high. Consequently, the reliability of all methods increases slightly as *D* increases. The data table is given below:

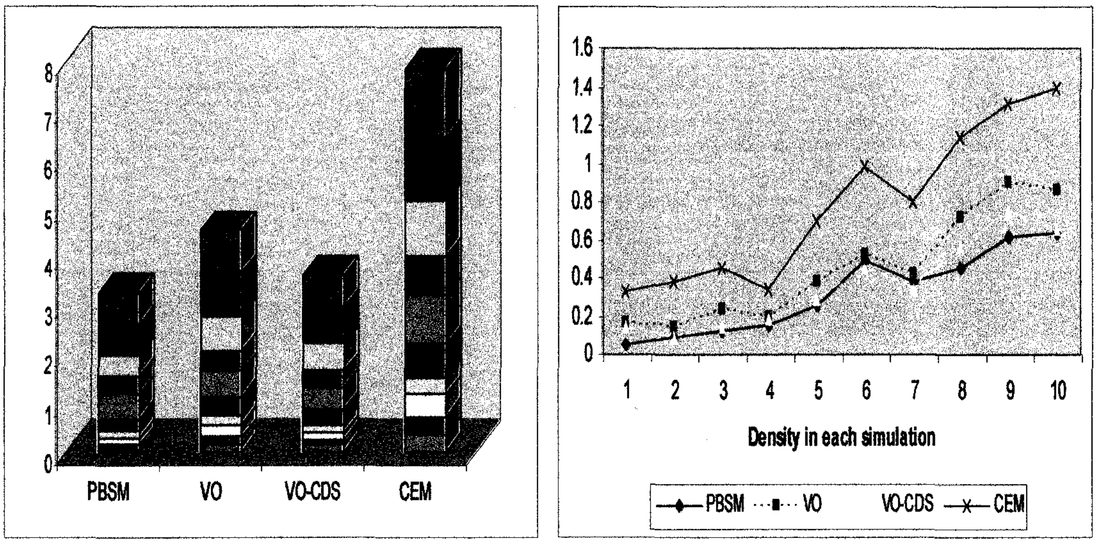
	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	91	90.3	89.3	89.9	0.15 (1)
	97.1	94.4	94	98.7	0.3 (2)
	97.3	96.7	97.4	96.7	0.45 (3)
	92.86	80.4	81.2	93.3	0.6 (4)
	86.87	86.84	81.94	88.4	0.75 (5)
	93.4	96.8	92.4	94.3	0.9 (6)
	100	91	91	100	1.05 (7)
	100	100	93.3	100	1.2 (8)
	100	100	100	100	1.35 (9)
	100	98.67	98.67	100	1.5 (10)
Total	958.53	935.11	919.21	961.3	
Average	95.853	93.511	91.921	96.13	
ConfidenceInterval95%	95.9±2.86	93.5±3.95	91.9±3.97	96.1±2.75	

Table 56: Percentage of reliability in a mobility-based scenario (B=5, n=50, d=10)

5.7.5 Comparing performance based on density (*d*)

The performance of these protocols is compared based on density (*d*). In each simulation, the number of nodes (*n*) is 50, hello message time (*H*) is 10 and the value of mobility is between moderate and high mobility (i.e. *D*=0.75). Sending a broadcast message to immediate neighbor requires fixed time *B'*. *B* =*H*/*B'* (number of non-overlapping messages that can be sent between two hello messages where *B'* =1 and *H* is 10). Density (*d*) indicates the desired average node degree and is incremented by 1 in each simulation. The average node degree of the generated graph will be close to density (*d*). Each simulation consists of a randomly generated graph and ten hello messages.

Total number of transmissions per node



(A) (B)

Figure 56: Total number of transmissions per node in a density-based scenario (B=10, n=50, D=0.75)

The above bar graph shows the total transmissions per node (y-axis) against four different flooding techniques (x-axis). Each color in the bar graph represents a simulation. The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than other methods. There are four main reasons that are stated in section 5.6.2. The number of transmissions per node in PBSM is 28% and 10% less than [VO], [VO-CDS] respectively. Under similar condition (i.e. same parameter values), all the methods generated more transmissions when ideal MAC layer is used. This is because in simplified MAC layer, a node has to delay its transmission if at least one of its neighbors is currently transmitting (exposed node problem) and therefore it might not have the opportunity to transmit if broadcast time is over. Furthermore, more nodes decide to transmit (i.e. runs timeout) in methods of [VO, CEM] than PBSM but some of these nodes may not have time to transmit if these nodes have to wait for their neighbors to finish transmission (exposed node problem). Consequently, the difference in number of transmissions between PBSM and methods of [VO, CEM] is smaller than similar scenario (sec 5.6.1) where ideal MAC layer is used.

The above line graph shows the total transmissions per node (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *d* used in each simulation. As *d* increases, the graph becomes denser, which means that more nodes will receive a message from a single transmission. Therefore, more nodes will decide to retransmit and the number of transmissions of all methods increases as *d* increases. As *d* increases, the percentage of CDS nodes compared to all nodes decreases. Furthermore, in a dense network there is a greater possibility that a new node discovered was previously a 2-hop neighbor. Therefore, if a new node has previously received the message, it will be in the R list. Unlike other methods, there will be no additional transmissions in PBSM. Consequently, the difference in the number of transmissions between PBSM and other methods increases as *d* increases. The data tables are given below:

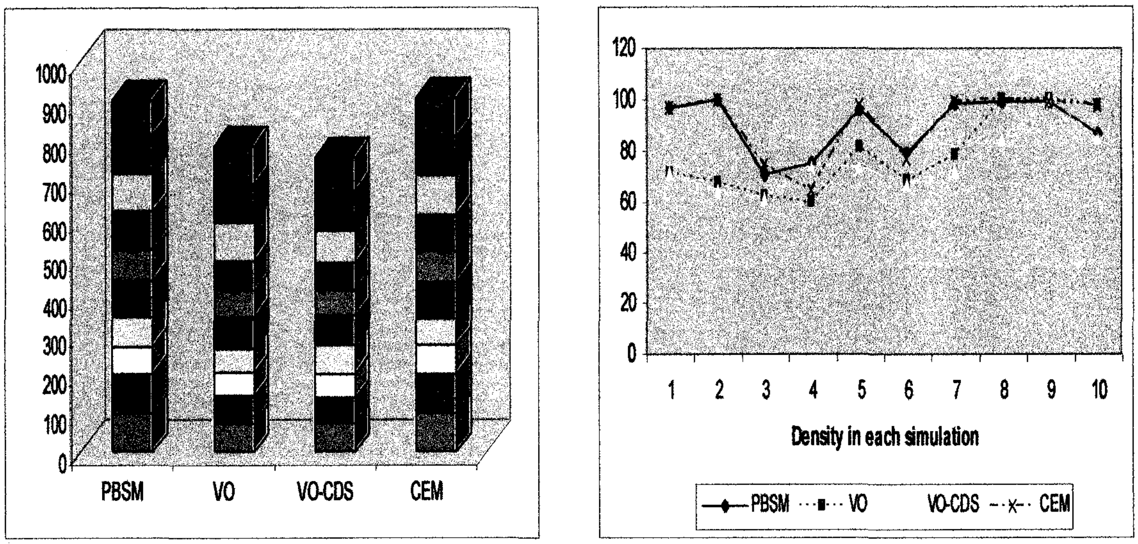
	PBSM	VO	VO-CDS	CEM	d (No of simulation)
	0.048	0.16	0.148	0.332	2 (1)
	0.096	0.148	0.092	0.38	3 (2)
	0.128	0.24	0.14	0.456	4 (3)
	0.152	0.192	0.184	0.336	5 (4)
	0.252	0.376	0.288	0.696	6 (5)
	0.488	0.524	0.448	0.988	7 (6)
	0.38	0.416	0.344	0.804	8 (7)
	0.448	0.716	0.564	1.136	9 (8)
	0.62	0.9	0.748	1.308	10 (9)
	0.64	0.86	0.652	1.392	11 (10)
Total	3.252	4.532	3.608	7.828	
Average	0.3252	0.4532	0.3608	0.7828	
ConfidenceInterval95%	0.33±0.14	0.45±0.18	0.36±0.14	0.78±0.25	

Table 57: Total number of transmissions per node in a density-based scenario (B=10, n=50, D=0.75)

	PBSM	VO	VO-CDS	CEM	d (No of simulation)
	2.4	8	7.4	16.6	2 (1)
	4.8	7.4	4.6	19	3 (2)
	6.4	12	7	22.8	4 (3)
	7.6	9.6	9.2	16.8	5 (4)
	12.6	18.8	14.4	34.8	6 (5)
	24.4	26.2	22.4	49.4	7 (6)
	19	20.8	17.2	40.2	8 (7)
	22.4	35.8	28.2	56.8	9 (8)
	31	45	37.4	65.4	10 (9)
	32	43	32.6	69.6	11 (10)
Total	162.6	226.6	180.4	391.4	
Average	16.26	22.66	18.04	39.14	

Table 58: Total number of transmissions in a density-based scenario (B=10, n=50, D=0.75)

Percentage of reliability



(A) (B)

Figure 57: Percentage of reliability in a density-based scenario (B=10, n=50, D=0.75)

The above bar graph shows the percentage of reliability (y-axis) against four different flooding techniques (x-axis). The PBSM and [CEM] protocols are around 90% reliable. The protocols of [VO and VO-CDS] are around 76% reliable. The reliability of [VO, VO-CDS] is lower since some nodes in [VO, VO-CDS] are in scoped or CDS flooding mode. These nodes do not transmit when they discover new node and therefore the percentage of reliability decreases.

Under similar condition (i.e. same parameter values), all the methods show a higher reliability when ideal MAC layer is used. In this case the reliability reduces due to collision as simplified MAC layer is used. The other methods encounter more collision than PBSM because they generate more number of transmissions. Consequently, the difference in reliability between PBSM and the other methods is greater than similar scenario (sec 5.6.5) where ideal MAC layer is used. Although the method CEM generates most number of transmission and encounters most number of collision, its reliability is close to PBSM because the increased number of transmission make up for loss of reliability due to collision.

The above line graph shows the percentage of reliability (y-axis) in each simulation (x-axis). The numbers on the x-axis indicate the value of *d* used in each simulation. At lower density, a node has few neighbors and therefore it will receive message from transmission of few neighbors. If there is collision among these transmissions, then a node might not receive message. Compared to PBSM, the methods of [VO, VO-CDS] generate more transmission and more collision. Therefore, the reliability of [VO, VO-CDS] is much less than PBSM at lower densities. As *d* increases, a node will have more neighbors. Therefore, even if there is collision among transmission from few neighbors, a node can receive transmission from other neighbors. Consequently, the reliability of [VO, VO-CDS] increases and the difference between these methods and PBSM decreases. The percentage of reliability of PBSM remains fairly constant. At lower density, PBSM produces very few collisions. As *d* increases, the number of collision in PBSM will increase. However, as *d* increases, a CDS node will cover more nodes. Therefore, a successful transmission from CDS node will cover many nodes. This makes up for the loss of reliability due to collision. For similar reason, the percentage of reliability of CEM is also constant as successful transmissions from few higher degree nodes will make up for the increased number of collisions as *d* increases. The data table is given below:

	PBSM	VO	VO-CDS	CEM	D (No of simulation)
	97.2	71.4	71.4	97.2	2 (1)
	100	68	65	100	3 (2)
	70.8	62	61.6	74.6	4 (3)
	75.6	60	74.8	64.8	5 (4)
	95.8	81.8	74	97.6	6 (5)
	79.2	68.6	66.8	77	7 (6)
	98.6	78.4	72	99.2	8 (7)
	99.2	100	85	100	9 (8)
	99.2	100	100	99.4	10 (9)
	87.2	97.4	85.4	97.4	11 (10)
Total	902.8	787.6	756	907.2	
Average	90.28	78.76	75.6	90.72	
Confidencelnterval95%	90.3+-6.94	78.8+-9.61	75.6+-7.15	90.7+-8.19	

Table 59: Percentage of reliability in a density-based scenario (B=10, n=50, D=0.75)

For all scenarios using simplified MAC layer, PBSM could have made a more significant gain in reliability due to collision if the following problem could be avoided. In PBSM, when a node *X* transmits, it adds all its immediate neighbors to R list. If a neighbor, *Y*, did not receive any message due to collision, then *Y* will be mistakenly put in *X*'s R list. Furthermore, when other neighbors of *X* (except *Y*) receives message from *X*, they will mistakenly put *Y* (neighbor of *X*) in their respective R list. Therefore, if node *X* or its current neighbors later discover *Y*, they will not transmit as *Y* is in their respective R list. Hence, *Y* will not receive any message. This problem does not occur in other methods because in other methods, a node transmits whenever it discovers a new neighbor.

In addition, nodes remove their neighbors from N list if they are in R list. As R list may mistakenly contain some nodes, N list of some of the neighbors of *X* may become mistakenly empty and then these neighbors will cancel transmission. Thus, reliability reduces. This problem does not occur in other methods because in other methods (except VO-CDS), nodes do not cancel transmission based on N list.

For each scenario (except density) the data are measured both at medium (d=4) and high (d=10) density. It is seen that the difference in reliability is more appreciable at medium density. There are two main reasons which are described below. The above mentioned problem in PBSM increases with density. That is, at higher density, more nodes will be mistakenly put in R list and more nodes will be mistakenly removed from N list. On the other hand, at lower density, a node has few neighbors and therefore it will receive message from transmission of few neighbors. If there is collision among these transmissions, then a node might not receive message. Compared to PBSM, the methods of [VO, VO-CDS] generate more transmission and more collision. Therefore, the reliability of [VO, VO-CDS] is much less than PBSM at lower densities. As d increases, a node will have more neighbors. Therefore, even if there is collision among transmission from few neighbors, a node can receive transmission from other neighbors. Furthermore, the number of transmissions (also collision) does not increase by a large amount as density increases due to exposed node problem. Overall, the number of collision increases slightly at higher density but the effect of collision (which leads to less reliability) is more significant at lower densities.

In all of the above mentioned scenario, PBSM showed significant gain against the methods of [VO, VO-CDS]. PBSM has negligible gain in reliability when compared with CEM protocol. Therefore, a new specific (classroom) scenario is simulated so that PBSM has an appreciable gain in reliability when compared with CEM protocol.

5.7.6 Comparing performance in a classroom scenario

In classroom scenario, there are *n* students in a class. These students can talk to each other and all of them know the information (e.g. course outline). Then a new group of *n* students enters the room. The task of the old student is to pass the information to nearby new student. An example is given below:

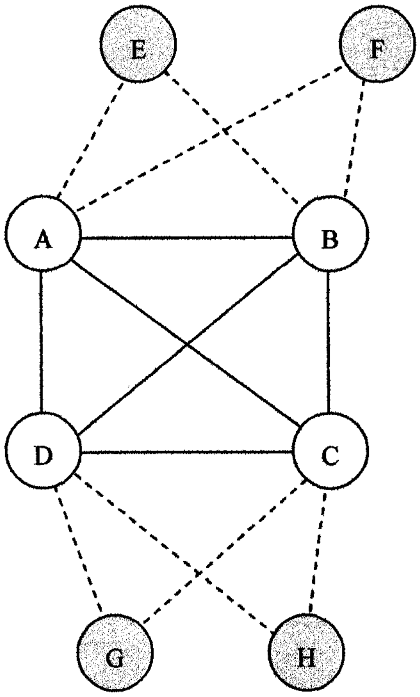


Figure 58: Example of classroom scenario

In the above figure, nodes *A, B, C,* and *D* are neighbors of each other and all of them has the message. Nodes *A, B, C,* and *D* form a complete graph. After sometimes, nodes *E, F, G* and *H* moved into the vicinity of the nodes *A, B, C* and *D.* None of the nodes *E, F, G,* and *H* have the message. Hello message takes place and the nodes discover each other. Nodes *A, B, C* and *D* decide to transmit as they have the message and therefore they run timeout. Suppose each transmission takes *p* slots.

CEM Protocol: Suppose node *A* transmits first. This transmission will take *p* slots. Node *E* and *F* will decide to transmit as both these nodes has neighbor other than the sender.

Nodes *B*, *C*, *D*, *E* and *F* have to wait for these *p* slots. If node *B*, *E* and *F* has shorter timeout than *C* and *D*, then they will transmit first. This means that node *C* and *D* has to wait for at least *3p* more slots. Nodes *C* and *D* might not have time to transmit and nodes *G* and *H* will not receive the message.

PBSM Protocol: Suppose node *A* transmits first. This transmission will take *p* slots. Node *B*, *E* and *F* will not decide to transmit as their neighbor received message from the sender (empty N list). Nodes *C* and *D* have to wait for *p* slots. In PBSM, node *C* and *D* has to wait for about *p* slots whereas in CEM node *C* and *D* has to wait for more than 3p slots. Therefore, nodes *C* and *D* will have time to transmit and nodes *G* and *H* will receive the message.

Therefore, in this case PBSM will have more reliability than CEM because in CEM an important transmission might not occur as it has to wait too long for other redundant transmissions. The example has 8 nodes and the reliability gain will be more if more nodes are used.

A classroom scenario is simulated. The purpose of this scenario is to show that PBSM has significant gain in reliability over CEM in certain circumstances. The main idea of this scenario is to simulate a situation where an important transmission cannot take place as it has to wait for other redundant transmissions (exposed node problem). Some nodes may not receive message if this transmission couldn't take place and hence there will be loss of reliability. PBSM will exhibit higher reliability if the important transmission takes place in PBSM but not in CEM. We know that CEM generates too many redundant transmissions. Broadcast time is made short so that important transmission could not occur in CEM as redundant transmissions takes up the broadcast time.

20 nodes are randomly generated in a square area, *S*. Each node is a neighbor of all other nodes (complete graph). It is assumed that every one of these 20 nodes has the message. A complete graph is used so that some node cannot start transmission as it has to wait for other neighbors to complete their transmission (exposed node problem). Therefore, a node could not transmit if it waits too long and broadcast time is over. Furthermore, CEM generates many redundant transmissions when graph is complete because a node in CEM protocol will decide to transmit after receiving message if it has neighbors other than

sender. However, PBSM will generate very few redundant transmissions as a node in PBSM will not transmit after receiving message if its entire neighbor is also a neighbor of sender. Now, some new set of 20 nodes are generated around the square area *S* (outside *S*). The old nodes are static and the new nodes are mobile. The new nodes are moved in such a way so that they are always outside *S* and maintain a minimum distance with the boundary of *S*. This is done so that the new nodes will have few old nodes (usually the nodes that are close to boundary of *S*) as neighbors. Therefore, new nodes can receive message from transmission of few old nodes and there will be loss of reliability if some of the old nodes, that has new nodes as neighbors, cannot transmit (exposed node problem). The graph containing the initial 20 nodes is complete but the whole graph (containing all 40 nodes) is not complete.

Computer screen area is 480 X 480. The top left corner has co-ordinate (0, 0) and the bottom right corner has co-ordinate (480,480). Square *S* is at the middle of screen area and its area is one-fourth of the total screen area. That is, square *S* extends from co-ordinate (120, 120) to (360,360). Transmission radius, *r*, is calculated using the formula $r^2 = (d * a * a) / ((n-1) * \pi)$, where average node degree (*d*) is 16, *n* is 20 and *a* (=230) represents the side of square *S*. Two nodes are considered connected if the distance between them is at most *r*. 20 nodes are generated in square *S*. A complete graph is obtained by first generating one node. Then, when another node is added, it is made sure that it is connected to all the nodes that are generated earlier. Otherwise, the node is generated again. In this way, a complete graph of 20 nodes is generated. The x and y co-ordinates are calculated using the formula: x = 120 + rand () % (360 - 120).

That is, x and y co-ordinates are between 120 and 360 (within *S*). Now, some new set of 20 nodes are generated around the square area *S* (outside *S*). The *x* and *y* co-ordinates of the additional nodes are calculated using the formula:

x = 0 + rand() % (90 - 0) or x = 390 + rand() % (470 - 390).

That is, x and y co-ordinates are between 0 and 90 or between 390 and 470 (computers viewport is 480 X 480). This is done so that the new sets of nodes initially don't have any old nodes as neighbors. The additional nodes move in a way similar to the random waypoint mobility model [BMJ]. The co-ordinates of their destinations are chosen at random. A node moves to its destination in straight line. The co-ordinates of the destination are calculated using the formula:

newx = 0 + rand() % (105 - 0) or newx = 375 + rand() % (470 - 375);

That is, the x and y co-ordinates are outside square *S* and there is at least some difference between the edge of the square and the co-ordinates of destination. This ensures that the new nodes will have few old nodes (usually the nodes that are close to boundary of *S*) as neighbors.

Under maximal speed, a node can travel distance *D'* between two hello messages. Let *D'=D*r*, where *r* is the transmission radius and *H* is the time between two hello messages. Therefore, the maximal speed is *D*r /H* (*=D'/H*). The average velocity is maximum speed/2. Consequently, both speed and *D'* depend on *D* and *D* is used as a parameter to define mobility. In this scenario, half of the nodes move. [*D*=1.35, so max speed = 1.35* *r*/ *H*]. In each simulation, the number of nodes (n) is 40. The hello message time (*H*) is 1. The time taken to transmit a message from a node to its immediate neighbor is taken as one unit. Therefore, *B* (the number of non-overlapping messages that can be sent between two hello messages) is equal to *H*. Each simulation consists of a randomly generated graph and ten hello messages.

Total number of transmissions per node

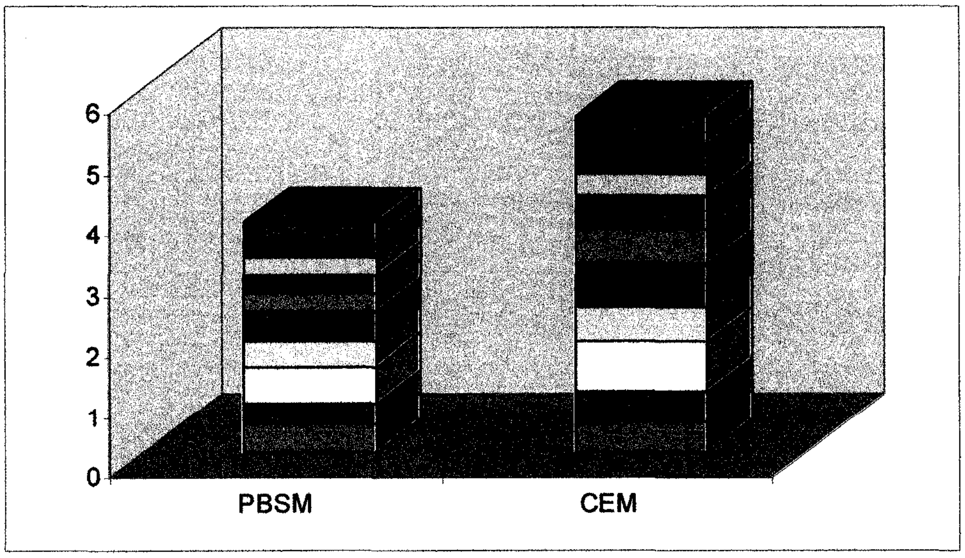


Figure 59: Total number of transmissions per node in classroom scenario

(D=1.35, n=40, B=1)

The above bar graph shows the total transmissions per node (y-axis) against PBSM and CEM protocol (x-axis). The bar graph shows that the total number of transmissions per node for the proposed PBSM method is less than CEM. There are four main reasons which are stated in section 5.6.2. The number of transmissions per node in PBSM is around 30% lower than the method of CEM. The difference in number of transmissions between PBSM and CEM is less than other scenarios as many nodes in CEM did not have time to transmit as broadcast time was over (exposed node problem). Since all old nodes are neighbors of each other, some old node has to wait a long time for their neighbors to finish transmission. The data table is given below:

			No of
	PBSM	**CEM**	**simulation**
	0.45	0.5	1
	0.375	0.525	2
	0.6	0.8	3
	0.4	0.575	4
	0.5	0.725	5
	0.3	0.55	6
	0.3	0.55	7
	0.3	0.375	8
	0.275	0.5	9
	0.3	0.45	10
Total	3.8	5.55	
Average	0.38	0.555	
ConfidenceInterval95%	0.38±0.07	0.56±0.08	

Table 60: Total number of transmissions per node in a classroom scenario

(D=1.35, n=40, B=1)

Percentage of reliability

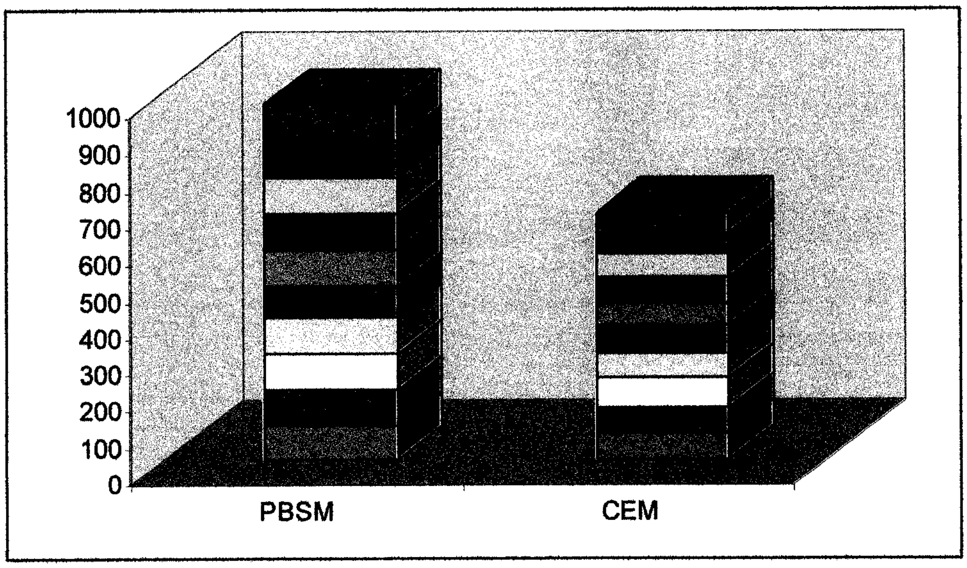


Figure 60: Percentage of reliability in a classroom scenario (D=1.35, n=40, B=1)

The above bar graph shows the percentage of reliability (y-axis) against PBSM and CEM protocol (x-axis). It is a measure of the number of nodes receiving the message. PBSM and CEM are about 96% and 67% reliable respectively. Suppose an old node discovers a new node and it transmits (first transmission). Other old node will wait as they can hear this transmission. There will be no collision in this case and all old nodes will receive this transmission. In CEM, all other old node will decide to transmit after receiving message (receiver has neighbor other than sender). Redundant transmissions are transmissions by old nodes that don't have any new nodes (this happens only in CEM). In PBSM, only those old nodes, that have new neighbors, will decide to transmit after receiving message because in PBSM a node will not decide to transmit if all of its neighbors are also neighbor of sender (NES). Furthermore, in PBSM, a node will cancel transmission if its N list becomes empty. The redundant transmissions are thus eliminated. The nodes that initially contain message can hear each other's transmission (i.e. neighbors) and therefore there are negligible collisions at the receiver (hidden terminal problem). Therefore, in PBSM, the problem of mistakenly putting nodes in R list is minimal. On the contrary, in CEM nodes do not cancel transmission and more nodes runs timeout. The nodes that initially contain message have many neighbors. Therefore, a node cannot start transmission as it has to wait for the transmission of its neighbors to end (exposed node problem). An important transmission might not occur because broadcast time is over (hello message time is short) and it has to wait for other redundant transmissions. An important transmission by an old node (usually the nodes that are near the boundary of *S*) is the transmission where one or more new node can only receive message from this old node. Consequently, PBSM showed a significant gain in reliability in this scenario. The data table is given below:

	PBSM	**CEM**	**No of simulation**
	94.4	72	1
	97.3	72	2
	97.8	78	3
	98	70.7	4
	88.7	74.4	5
	98	58	6
	98	72.7	7
	98	66	8
	97.8	55.5	9
	97.3	56	10
Total	965.3	675.3	
Average	96.53	67.53	
ConfidenceInterval95%	96.5±1.84	67.5±5.08	

Table 61: Percentage of reliability in a classroom scenario (D=1.35, n=40, B=1)

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this paper, we have proposed a new method for broadcasting in both static and mobile wireless ad hoc networks. The features of the protocol are the following. It uses connected dominating set (CDS) [CS], neighbor elimination scheme (NES) [PL, SSZ], and does not need any parameters to detect network conditions or to change its characteristics. It is suitable for broadcasting messages in both static and mobile networks. In static networks, the number of transmissions is reduced as total transmission number is at most equal to the number of CDS nodes. Furthermore, due to NES, some nodes may not need to transmit. In the case of mobile networks, a node *v* is discovered by node *u* if it enters the transmission region of node *u*. Unlike other methods, in the proposed method, two nodes do not transmit every time they discover each other as new neighbors. When a node discovers a new neighbor, it first checks whether the new node is in R. If it is not in R, only then does the node consider transmitting. Otherwise, it does not transmit.

The protocols of [VO] and [CEM] are selected for comparison since they consider similar assumptions and are developed for highly mobile scenarios. Although they have some limitations, they perform better than some of the other available protocols. Therefore, they are considered valid competitors. Furthermore, another method, VO-CDS, is also introduced to monitor the contribution of CDS over scoped flooding and the impact of using thresholds. The proposed method is first compared with these methods by example considering both static and mobile networks. After that, the proposed and existing methods are implemented and comparison is done by simulation. The simulation is performed in two parts. In the first part (sec 5.6) an ideal MAC layer without collision is assumed whereas in the second part (sec 5.7) a simplified MAC layer with collision is considered. The simulation contains many scenarios including static, moderate, and high mobility. Moreover, the performance of protocols is also compared based on *D* (a parameter used to define the speed of movement), density (*d*) (a parameter that indicates the average node degree of graph) and *B* (the number of messages that can be sent between two hello messages. $B = H/B'$ where $B' = 1$ and *H* is hello message time). The results of the simulation verified the claim that the proposed method has the minimum number of transmissions among the methods described, in both static and mobile networks. However, the difference between PBSM and the adaptive protocol decreases in highly mobile networks, as the thresholds work better due to greater differences in the speed of nodes. The proposed method also maintains an equal degree of reliability compared to the other methods when ideal MAC layer is used. However, PBSM showed a significant gain in reliability over the methods of [VO, VO-CDS] when simplified MAC layer is considered. This is because less transmission in PBSM leads to lower number of collision and therefore a higher degree of reliability than other methods. Furthermore, a classroom scenario is simulated to show cases where PBSM has appreciable gain in reliability when compared with CEM protocol.

The protocols of [VO, CEM] were not very efficient since they increase reliability at the expense of the number of transmissions. The flooding methods of [CEM] and hyperflooding [VO] produce extra transmissions. Scoped flooding [VO] often fails to transmit when a new node is discovered. Moreover, the method of [VO] needs calculation and the storage of extra parameters (e.g. relative velocity) in order to switch between different flooding techniques. Furthermore, there are a few cases where the performance of PBSM is much better than that of the adaptive protocol [VO]. For example, in a corridor scenario (see sec 5.6.6), the reliability of PBSM is almost double that of the adaptive protocol. In a disaster scenario (see sec 5.6.7), where all nodes increase speed, the adaptive protocol has many more transmissions than PBSM.

Moreover, in a traffic scenario (see sec 5.6.8), where all nodes reduce speed, the adaptive protocol has much less reliability than PBSM.

Some limitations of the protocol are stated in Chapter 4. The minor limitations are as follows

- The R list of a node usually contains nodes that have received the message up to 2-hops. Therefore, if a node discovers a node that has received message but was more than 2 hops away from it, it will transmit. This transmission is unnecessary.
- There may be an overhead due to the calculation of CDS after every hello message. However, only local information (up to 2 hops) without any message exchange is needed to calculate CDS and the overhead can be ignored.
- The maintenance of extra R and N lists by each node may have some overhead especially if the R list becomes large.

Although the 2-hop version of the protocol is more efficient, there is some overhead as it uses 2-hop information. The second round of hello messages to gain 2-hop information in fact has some overhead in message size as neighbor lists are exchanged. The 1-hop version of the protocol can avoid this overhead if position information is used. There are other position based broadcasting schemes based on area coverage which may be applied as an alternative.

Some assumptions are made (as stated in section 5.1) which makes the simulated environment slightly different from the real network. The implementation of the existing methods differs slightly from the actual protocol but it is implemented in such a way that the experimental data gathered will not differ from the actual protocol. The main concept of each protocol is used in the implementation. The simulated environment slightly differs from the real network but, since all the protocols are examined in the same environment; the gathered data shows the comparisons fairly correctly.

This research enabled me to learn about different broadcasting algorithms in wireless networks and the principles behind these methods. It also improved my simulation skills. We are glad to propose a protocol that is more efficient than some of the existing protocols.

6.2 Future Work

The proposed protocol could be modified, for better performance and to overcome limitations, in the following ways:

- To avoid the limitation mentioned in section 4.3.1, a node can transmit a message whenever it receives a message from a node that is not in its current neighbor list. The drawback is that this generates additional transmission.
- Nodes can also include their geographic position, their current speed, and their direction of movement in hello message. Each node can estimate, for each neighbor, how long the current link will last. This estimation is possible as a node can predict its neighbor's position in subsequent times based on its current velocity and position [SLG]. Therefore, if a node estimates that a link with its neighbor is broken when a broadcasting message arrives, it updates its neighbor list and sends a hello message. Consequently, the limitation mentioned in section 4.3.2 can be avoided using this information. The drawback is that there is an overhead in estimating link duration and exchanging or calculating velocity. Also, GPS or the calculation of signal strength is needed to obtain position information.
- When a node receives a data message from a node that is not in its current neighbor list, it will immediately send a hello message. If a node receives a hello message that is not periodic and if the hello message is received shortly after transmitting, it will include the new node in its R list. Consequently, the limitation mentioned in section 4.3.3 can be avoided using additional hello messages. The drawback is that additional, non-periodic hello messages will be generated.
- When a node receives broadcast message from a node that is not in its current neighbor list, it will immediately re-calculate CDS. This is especially helpful in 1-hop version of the protocol. This can reduce the number of transmissions.
- A node can know the R list of its neighbors if R lists become part of hello messages. Therefore, a node could know whether other nodes that are more than 2-hops away receive the message. This reduces the number of transmissions.

However, it introduces overhead in hello message size as R list could become big

- A variable transmission radius could be considered to simulate more realistic network.

One of the protocols from [HBR] considers short advertisements, sent whenever a new node is found, followed by a full message if any neighbor responds requesting it. The approach is justified when the full message is much longer than the short one. The protocol for disseminating short messages alone can be considered a broadcasting task in itself. We could modify our proposed protocol to handle short messages like those of [HBR]. However, there would be two variations from the protocol [HBR]. The first one is to introduce waiting periods before sending advertisements. The second is that it is not necessary for all nodes to send advertisements. Only those that are in CDS, and with nonempty N lists, can do so. Thus, we can apply our proposed protocol to disseminate short messages.

Although the modification mentioned above will overcome the limitation and will improve the protocol for some scenarios, it may include additional overhead or transmissions. Therefore a simulation is required to verify the efficiency of the modifications. The future work needed is to do another simulation using the modified protocol. Furthermore, we can change the simulated environment by considering a larger number of nodes and more mobility scenarios.

Bibliography

- [APVH] A. Amis, R. Prakash, T. Vuong, and D. Huynh, "Max-min D-cluster formation in wireless ad hoc networks", in *Proceedings of the IEEE Infocom*, Anchorage, Alaska, April 2001.
- [BMJ] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking* (MobiCom'98), 1998.
- [CEM] Cooper, D.E., Ezhilchelvan, P. and Mitrani, I., "High Coverage Broadcasting for Mobile Ad Hoc Networks", *3rd International IFIP-TC6 Networking Conference Networking 2004*, Athens, Greece, May 9 - 14, 2004, Lecture Notes in Computer Science Volume 3042, 100-111, 2004.
- [CS] J. Carle and D. Simplot-Ryl, "Energy efficient area monitoring for sensor networks", *IEEE Computer Magazine*, vol. 37, no. 2, pp. 40 – 46, 2004.
- [DW] F. Dai and J. Wu, "Distributed dominant pruning in ad hoc networks", in *Proceedings of the IEEE International Conference on Communications* (ICC'03), Anchorage, Alaska, (2003).
- [GK] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets". Algorithmica, 20(4): 374 - 387. (1998)
- [HBR] J. Hahner, C. Becker, K. Rothermel, "A protocol for data dissemination in frequently partitioned mobile ad hoc networks", *IEEE Int. Symp. Computers and Communications ISCC*, Turkey, July 2003, 633-640.
- [ISGS] François Ingelrest, David Simplot-Ryl, Hong Guo and Ivan Stojmenovic, "Broadcasting in ad hoc and sensor networks", in: **The Handbook of Computer Networks**, 3 Volume Set (Hossein Bidgoli, EIC), Wiley, to appear.
- [ISS] Franois Ingelrest, David Simplot-Ryl, Ivan Stojmenovic, "Smaller Connected Dominating Sets in Ad Hoc and Sensor Networks based on Coverage by Two-Hop Neighbors".

- [LW] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks", *IEEE Transactions on Mobile Computing*, 1, 2, 111-122.
- [NTCS] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network", in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, USA, August 1999.
- [OW] Regina O'Dell, Roger Wattenhofer, "Information Dissemination in Highly Dynamic Graphs", *DIALMPOMC' 05,* September 2, 2005, Cologne, Germany.
- [PL] W. Peng and X. Lu, " On the reduction of broadcast redundancy in mobile ad hoc networks", in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*, Boston, USA, 2000.
- [SE] Edwin Soedarmadji, Robert J. McEliece, "A Dynamic Graph Algorithm for the Highly Dynamic Network Problem", *Proc FAWN at IEEE PerCom*, March 2006.
- [SLG] W. Su, S. J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management,* vol. 6, no. 4, pp. 351–360, January 2001.
- [SSW] David Simplot-Ryl, Ivan Stojmenovic, and Jie Wu, "Energy efficient backbone construction, broadcasting, and area coverage in sensor networks", in: **Handbook of Sensor Networks: Algorithms and Architectures** (I. Stojmenovic, ed.), Wiley, 2005, 343-379.
- [SSZ] I. Stojmenovic, M. Seddigh, J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks", *IEEE Transactions on Parallel and Distributed Systems*, 13(1): 14 – 25. 2001
- *[S-ssz].* I. Stojmenovic, Comments and corrections to "Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, 15, 11, Nov. 2004, 1054-1055.
- [SW] I. Stojmenovic and J. Wu, "Broadcasting and activity scheduling in ad hoc networks", in: **Mobile Ad Hoc Networking** (S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, eds.), IEEE Press, 2004, 205-229.

- [VO]$_1$ K. Viswanath and K. Obraczka, “An adaptive approach to group communications in multi-hop ad hoc networks”, *International Conference on Networking (ICN 2002)*, Atlanta, Georgia, August 2002.
- [VO]$_2$ K. Viswanath and K. Obraczka, “An adaptive approach to group communications in multi-hop ad hoc networks”, *IEEE Int. Symp. Computers and Communications ISCC*, Taormina, Italy, July 2002, 559-566.
- [WL] J. Wu and H. Li, “A dominating set based routing scheme in ad hoc wireless networks”, *Proc. DIAL M*, Seattle, Aug. 1999, 7-14; Telecommunication Systems, 18, 1-2, 2001, 13-36.
- [ZJ] Hao Zhang and Zhong-Ping Jiang, “Performance evaluation and improvement of broadcasting algorithms in highly mobile ad hoc networks”, *Workshop on Performance of Wireless Networks and Communication Systems (WiNCS)*, Philadelphia, PA, July 23-28, 2005.