

Epidemic-based Reliable and Adaptive Multicast for Mobile Ad hoc Networks

by

Zülküf Genç

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Electrical & Computer Engineering

Koç University

March 2006

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Zülküf Genç

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assist. Prof. Öznur Özkasap (Advisor)

Prof. Reha Civanlar

Assist. Prof. Sibel Salman

Date:

To My Mother Sabiye

ABSTRACT

The characteristic of mobile ad hoc networks demands new set of network strategies in order to provide reliable group communication. The techniques used by majority of the existing solutions are inherited from wired networks and can not adapt well to the dynamic environment of mobile ad hoc networks. On the other hand, the stateless character of epidemic-based approaches is well suited to the non-deterministic nature of such networks.

This thesis presents a novel protocol, namely EraMobile, offering Epidemic-based Reliable and Adaptive Multicast for Mobile ad hoc networks. EraMobile's target is group applications requiring high-level of reliability, and the protocol aims to provide fully reliable multicast data delivery with minimal network overhead even in the adverse network conditions. EraMobile utilizes an epidemic-based method in multicast operation to cope with dynamic and unpredictable topology changes arising from the mobility. In contrast to prior studies, this method carries out both multicast delivery and recovery of missing data by use of gossip messages without needing any underlying routing protocol. The gossip messages are periodically disseminated by exploiting the broadcast nature of wireless medium. The multicast data delivery is then performed through peer-to-peer communications succeeding the gossip broadcasts. Our epidemic mechanism does not require the maintenance of any tree- or mesh-like structure for multicasting. It also needs neither having global or partial view of the network nor having information of neighboring nodes and group members. Besides, it substantially minimizes the overhead incurred by eliminating redundant data transmissions. Another distinguishing feature of EraMobile is its capability of adapting to varying node densities in order to provide reliable data delivery in both sparse networks, where the network connectivity is prone to interruptions, and dense networks, where congestion is likely to occur. EraMobile is shown to achieve fully reliable multicast data delivery for most of the scenarios studied through extensive simulations on ns-2 network simulator by outperforming the other protocols compared, especially in terms of both packet delivery ratio and overhead efficiency.

ÖZETÇE

Devingen plansız ağların (mobile ad hoc networks) kendine özgü karakteristiği güvenilir (reliable) grup iletişimini gerçeklemede yeni yaklaşımlar gerektirmektedir. Mevcut yaklaşımlar genelde kablolu ağlara özgü tekniklerle çözüm yoluna gittiklerinden plansız ağların devingen ortamına uyum sağlayamamaktadırlar. Diğer yandan, yaygın-tabanlı (epidemic-based) yaklaşımların durumsuz (stateless) yapısı plansız ağların kararsız ortamına oldukça uyum sağlamaktadır.

Bu tez çalışması, EraMobile (Epidemic-based Reliable and Adaptive Multicast for Mobile ad hoc networks) adında, devingen plansız ağlar için yeni bir çoğa gönderim (multicast) protokolü sunmaktadır. EraMobile'ın hedefi yüksek düzeyde güvenilirlik gerektiren grup uygulamalarında güvenilir çoğa gönderimi en düşük seviyede ağ ek yükü (overhead) ile elverişsiz ağ şartlarında dahi gerçekleştirmektir. EraMobile, devingenliğin neden olduğu dinamik ve kestirilemez ağ düzeni değişimleri ile başa çıkabilmek için yaygın-tabanlı bir teknik kullanmaktadır. Önceki çalışmaların aksine bu teknik hem çoğa gönderim dağıtımını hem de kayıp verilerin telafisini *gossip* adı verilen özel mesajlarla alt katmanda herhangi bir yol atama (routing) protokolüne gerek duymadan yapabilmektedir. Gossip mesajlar periyodik olarak telsiz kapsama alanı içindeki diğer katılımcılara telsiz ortamın doğal yayın özelliğinden yararlanılarak yayınlanır (broadcast). Ardından çoğa gönderim veri dağıtımı eşler arası (peer-to-peer) iletişimlerle gerçekleştirilir. Kullandığımız yaygın-tabanlı teknik çoğa gönderim için herhangi bir ağaç benzeri yapı kullanmamaktadır. Ayrıca, yöntemimiz ağın genel veya özel bilgisi, telsiz kapsama alanında olan diğer katılımcıların bilgisi ve çoğa dağıtım grup üyelerinin bilgisine gereksinim duymamaktadır. Buna ek olarak, kullanmış olduğumuz teknik fazladan veri iletimini engelleyerek ağın üzerindeki ek yükü önemli ölçüde azaltmaktadır. EraMobile'ın önemli diğer özelliği de değişken katılımcı yoğunluğuna uyarlanarak hem ağ ulaşılabilirliğinin zayıf olduğu seyrek ortamlarda hem de sıkışıklığa açık yoğun ortamlarda güvenilir çoğa gönderim dağıtımını gerçekleştirebilmesidir. Ns-2 ağ benzetici ile gerçekleştirilen geniş kapsamlı benzetim çalışmalarının sonucunda, EraMobile diğer çoğa gönderim dağıtım protokollerinden daha iyi paket dağıtımı, güvenilirlik ve ek yük verimliliği sağlayarak denenen çoğu senaryo için tam güvenilir çoğa gönderim dağıtımını başarmıştır.

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my advisor Assist. Prof. Öznur Özkasap for her guidance, understanding and patience throughout my graduate study in Koç University. I would like to thank Prof. Reha Civanlar and Assist. Prof. Sibel Salman for their valuable comments and time in reviewing this thesis as my committee members. I would also like to give thanks to my former advisor, Assist. Prof. Feza Buzluca, in Istanbul Technical University for his support. I was fortunate to have such great friends close to me during my graduate education; Ozan Sönmez my roommate, Bengi Mizrahi and Tayfun Elmas my neighbors in the dorm, Gökçe Görbil my officemate and Utkan Ögmen my classmate. I am grateful to my parents: Harun and Sabiye, my brothers: Aziz and Yalçın for their continuous love and support during my entire life. Finally, I owe thanks to founders, administrators and engineering faculty members of Koç University for the high-class research, education and living facilities provided in such wonderful campus environment.

This research was supported in part by TUBITAK (The Scientific and Technological Research Council of Turkey) - COST Action: 279, Analysis and Design of Multiservice Networks Supporting Mobility, Multimedia, Internetworking project.

List of Tables	ix
List of Figures	x
Nomenclature	xiii
Chapter 1: Introduction	1
1.1 Motivation.....	2
1.2 Solution	5
1.3 Application Scenarios	10
1.4 Contributions	11
Chapter 2: Background and Related Work	12
2.1 Ad hoc Networks	12
2.1.1 Characteristic	13
2.1.2 Applications.....	15
2.2 Epidemic Communication	17
2.3 Related Work	18
Chapter 3: EraMobile	23
3.1 Overview	23
3.2 EraMobile Units	24
3.2.1 Data Dissemination Unit	24
3.2.2 Buffer Management Unit.....	27
3.2.3 Adaptivity Unit.....	29
3.3 Optimizations.....	32
3.4 Properties and Algorithm.....	34

Chapter 4: Experimental Setup	40
4.1 Simulation Methodology	40
4.2 Simulation Environment.....	42
4.2.1 Radio Model	43
4.2.2 Medium Access Control Protocol	44
4.2.3 Mobility Pattern.....	45
4.2.4 Traffic Pattern.....	46
Chapter 5: Parameter Analysis	47
5.1 Low Node Density	47
5.2 Moderate Node Density	51
5.3 High Node Density	54
5.4 Result	58
Chapter 6: Performance Evaluation	59
6.1 Performance Metrics	59
6.2 Results	60
6.2.1 Effect of Mobility	61
6.2.2 Effect of Group Size	71
6.2.3 Effect of Network Load.....	77
6.2.4 Effect of Multiple Groups and Senders	80
6.3 Discussion	83
Chapter 7: Conclusion and Future Work	86
Bibliography	90
Vita	96

LIST OF TABLES

Table 2.1. Mobile ad hoc network applications	16
Table 3.1. The events used in the pseudo code of EraMobile	35
Table 3.2. The data structures used in the pseudo code of EraMobile	35
Table 3.3. The special variables used in the pseudo code of EraMobile	35
Table 5.1. The throughput vs. gossip interval (GI) and stability threshold (ST) in low node density	50
Table 5.2. The overhead vs. gossip interval (GI) and stability threshold (ST) in low node density	51
Table 5.3. The throughput vs. gossip interval (GI) and stability threshold (ST) in moderate node density	52
Table 5.4. The overhead vs. gossip interval (GI) and stability threshold (ST) in moderate node density	53
Table 5.5. The throughput vs. gossip interval (GI) and stability threshold (ST) in high node density	56
Table 5.6. The data packet loss ratio vs. gossip interval (GI) and stability threshold (ST) in high node density	57
Table 5.7. The overhead vs. gossip interval (GI) and stability threshold (ST) in high node density	58
Table 6.1. The values of the protocol parameters for different node density levels	60
Table 6.2. The node densities obtained for each group size in the simulation area of 1000m x 1000m	71
Table 6.3. Multicast groups, senders and size of groups	80

LIST OF FIGURES

Figure 1.1. A mobile ad-hoc network scenario.....	9
Figure 2.1. Relaying the packets by the intermediate node	14
Figure 3.1. Gossip digest construction.....	25
Figure 3.2. Propagation of gossip messages	26
Figure 3.3. Data distribution	28
Figure 3.4. Gossip Size Reduction.....	33
Figure 4.1. Join process in MAODV	42
Figure 4.2. Omni-directional antenna	44
Figure 5.1. The effect of the protocol parameters on the throughput in low node density	49
Figure 5.2. The effect of the protocol parameters on the overhead in low node density...	50
Figure 5.3. The effect of the protocol parameters on the throughput in moderate node density	52
Figure 5.4. The effect of the protocol parameters on the overhead in moderate node density	53
Figure 5.5. The effect of the protocol parameters on the throughput in high node density	55
Figure 5.6. The effect of the protocol parameters on the losses of the data packets in high node density	56
Figure 5.7. The effect of the protocol parameters on the overhead in high node density..	57
Figure 6.1. The effect of mobility on the throughput of the protocols in low node density	63
Figure 6.2. The effect of mobility on the reliable-throughput of the protocols in low node density	63

Figure 6.3. The effect of mobility on the receive-overhead of the protocols in low node density	64
Figure 6.4. The effect of mobility on the send-overhead of the protocols in low node density	64
Figure 6.5. The effect of mobility on the throughput of the protocols in moderate node density	66
Figure 6.6. The effect of mobility on the reliable-throughput of the protocols in moderate node density	67
Figure 6.7. The effect of mobility on the receive-overhead of the protocols in moderate node density	67
Figure 6.8. The effect of mobility on the send-overhead of the protocols in moderate node density	68
Figure 6.9. The effect of mobility on the throughput of the protocols in high node density	69
Figure 6.10. The effect of mobility on the reliable-throughput of the protocols in high node density	69
Figure 6.11. The effect of mobility on the receive-overhead of the protocols in high node density	70
Figure 6.12. The effect of mobility on the send-overhead of the protocols in high node density	70

Figure 6.13. The effect of group size on the throughput of the protocols	73
Figure 6.14. The effect of group size on the reliable-throughput of the protocols	73
Figure 6.15. The effect of group size on the receive-overhead of the protocols	74
Figure 6.16. The effect of group size on the send-overhead of the protocols.....	74
Figure 6.17. The effect of group size on the throughput of the protocols	75
Figure 6.18. The effect of group size on the reliable-throughput of the protocols	75
Figure 6.19. The effect of group size on the receive-overhead of the protocols	76
Figure 6.20. The effect of group size on the send-overhead of the protocols.....	76
Figure 6.21. The effect of traffic load on the throughput of the protocols	78
Figure 6.22. The effect of traffic load on the reliable-throughput of the protocols.....	78
Figure 6.23. The effect of traffic load on the receive-overhead of the protocols	79
Figure 6.24. The effect of traffic load on the send-overhead of the protocols	79
Figure 6.25. The effect of multiple groups and senders on the throughput of the protocols.....	81
Figure 6.26. The effect of multiple groups and senders on the reliable-throughput of the protocols	81
Figure 6.27. The effect of multiple groups and senders on the receive-overhead of the protocols	82
Figure 6.28. The effect of multiple groups and senders on the send-overhead of the protocols	82
Figure 6.29. Time for completion of data delivery	85

NOMENCLATURE

MANET	Mobile Ad hoc Network
PDA	Personal Digital Assistant
ARQ	Automatic Repeat reQuest
UDP	User Datagram Protocol
FIFO	First In/First Out buffer
IEEE	Institute of Electrical and Electronics Engineers
MAODV	Multicast Ad hoc On-Demand Distance Vector
AODV	The Ad hoc On-demand Distance Vector
ODMRP	On-Demand Multicast Routing Protocol
RALM	Reliable Adaptive Lightweight Multicast Protocol
AG	Anonymous Gossip
RDG	Route Driven Gossip
RMA	Reliable Multicast Algorithm for Mobile Ad hoc Networks
ReAct	Reliable, Adaptive, Congestion-Controlled Ad hoc Multicast Transport Protocol
MAC	Medium Access Control
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
RTS/CTS	Request to Send/Clear to Send

Chapter 1

INTRODUCTION

Mobile ad hoc networks (MANETs) have gained considerable interest and popularity in recent years as they have enormous military and commercial potential. The latest advances in computer and wireless communication technologies contribute to this potential enabling portable devices such as cell phones, laptops, PDAs, media players to become smaller, affordable and equipped with short range wireless interface. These devices increasingly populate the world and extend the coverage of mobile ad hoc networks in daily life. Besides the current wide array of applications ranging from scientific to industrial to military operations and to rescue missions, intense research and development activities focusing on this area will open up new exciting possibilities that are not presently realized or envisioned.

A mobile ad hoc network is a collection of mobile computing devices using wireless links for communication without relying on any fixed infrastructure. This infrastructure-less and mobile nature is the main reason behind the popularity of ad hoc networks. However, it also imposes important challenges, e.g. highly dynamic and unpredictable topology changes, low bandwidth, high error rates and limited power sources, to the protocols and applications for mobile ad hoc networks. Therefore, new approaches and distinctive changes are required in the network protocol designs including multicasting.

In this thesis, we propose a reliable and adaptive multicasting solution which is well suited to wireless medium and characteristics of mobile ad hoc networks thanks to its epidemic-based nature.

1.1 Motivation

Group communication is an essential requirement for ad hoc networks since nodes forming these temporary networks generally come together for common purposes and they need to collaborate. Multicast routing protocols play a significant role in ad hoc networks to provide this communication. Multicasting is the transmission of data packets to a group of hosts identified by a single destination address [1]. In multicasting, the data can be delivered to several destinations by a single transmission instead of multiple transmissions as in unicast communication. Therefore, it minimizes the bandwidth consumption, sender and forwarder node processing, and delivery delay [2]. In ad hoc networks, the role of multicasting is even more important because of bandwidth and energy savings [3]. The use of multicast rather than multiple unicast in ad hoc settings improves the utilization of network sources by exploiting the inherent broadcast nature of wireless medium.

This efficiency of multicasting promotes it as an ideal communication paradigm for many possible applications requiring efficient support of group communication from underlying network protocols, e.g. military operations, emergency and rescue missions, networked games, conferencing, synchronization needs. Thus, many multicast protocols have been proposed [4, 5, 6, 7, 8, 9, 10 and 11] and their performance has been studied [12, 13, 14, 15, 16 and 17].

On-Demand Multicast Routing Protocol (ODMRP) [4] and Multicast Ad hoc On-Demand Distance Vector (MAODV) [5] are leading [12] best effort multicast routing protocols proposed for ad hoc networks. Both protocols are classified in the reactive routing protocols category. That is, the routes to the multicast receivers are built on demand

when multicast source has packet(s) to send but no fixed routes to receivers. They broadcast the periodic beacon messages to update the routes whereby they are called *soft state* protocols. The difference between them is the nature of multicast topology. ODMRP maintains mesh-based structure while MAODV utilizes a tree-based approach for multicast operation. A mesh structure allows the protocol to form multiple routes between source and receiver, whereas there is only a single path between source and receiver in tree based protocols. This structural difference affects the performance of the protocols. According to the simulation results obtained in [12], ODMRP performs better than MAODV in terms of throughput. The reason is that availability of multiple routes in mesh structure can enable the data delivery under high mobility at the cost of more network overhead. However, data packets can not be delivered to the receiver when the single path between source and receiver is broken due to mobility in tree-based protocols such as MAODV.

On the other hand, constructing and maintaining mesh and tree structures require certain amount of overhead. Frequent topology changes arising from high mobility may cause the negative effect of increasing overhead to turn out a prohibitive factor on the performance of multicast protocols. The simulation results in [12 and 14] reveal this interesting fact that flooding, which is the simplest dissemination method, performs better packet delivery ratios than the ODMRP and MAODV. However, at high mobility, even flooding results in packet deliver ratio of 70% [18].

In order to overcome the packet loss problem, some reliable multicast protocols have been proposed in recent years [19, 20, 21, 22, 23 and 24]. These protocols are classified in specific categories according to their reliability mechanisms such as deterministic-probabilistic [26] or ARQ-Gossip based [25]. The brief descriptions of main features of these protocols are presented below.

A Reliable Multicast Algorithm for Mobile Ad hoc Networks (RMA) [19] provides reliability employing acknowledgment (ACK) messages from receivers to sources. In the

case of unknown routes to the receivers, the source broadcasts the data packets and waits for ACK messages with the route information from nodes. If the sender does not receive ACK message from all destinations then it re-broadcasts the same data packets for pre-determined number of times. Thus, it is highly possible to go through a feedback implosion problem [27] when the group size grows [25]. ACK based approach requires a sender to know all the nodes in its multicast group. The group membership information is maintained by flooding of JOIN and LEAVE messages throughout the network without any reliability guarantees. This unreliable joining and leaving mechanism can degrade the reliability degree of the protocol [26].

Reliable Adaptive Lightweight Multicast Protocol (RALM) [20, 21] is a transport protocol built on ODMRP to enhance its packet delivery ratio in small group operation scenarios. RALM tries to achieve reliability employing a congestion control mechanism. It uses negative acknowledgment (NACK) feedback messages to adjust the congestion experienced by multicast receivers. However, to avoid the feedback implosion problem, a node is selected as a “feedback receiver” from the group of multicast receivers and only this node replies to the source. The simulation results show that RALM can slightly outperform UDP, unreliable transport protocol proposed for wired networks, for the traffic rates over 200ms. Furthermore, the difference in packet delivery ratio is less than 5% between UDP and RALM under increasing mobility. It seems that this protocol can operate well in static networks where the packet losses generally stem from congestion. In the highly mobile networks, the congestion control mechanism may not solve the reliability problem [25 and 26].

Reliable, Adaptive, Congestion-Controlled Ad hoc Multicast Transport Protocol (ReAct) [22] is the enhanced version of RALM with a local recovery mechanism. In ReAct, receivers first attempt to recover packet losses from the nearby nodes such as an upstream group member in the multicast structure. This local recovery mechanism improves

the performance of protocol preventing unnecessary back-offs and rate reductions. In the failure of local recovery, lost packets are recovered from source by use of NACK messages. If this happens frequently, protocol may subject to long delays and low delivery ratios [25].

Anonymous Gossip (AG) [23] is a reliability extension which can be added on any best-effort multicast protocol. AG is the first protocol utilizing an epidemic-based approach to provide reliable data dissemination in mobile ad hoc networks. The details of AG are explained in section 2.3 since it is highly related to the solution proposed in this thesis. The simulation results in [22] show that AG is far from providing reliable data delivery.

Route Driven Gossip (RDG) [24] protocol also uses an epidemic-based approach for reliable multicasting in ad hoc networks. However, it doesn't rely on any multicast protocol. Instead, RDG uses a pure gossip schema on the top of a unicast protocol for both data delivery and reliability. The details of this protocol can be found in section 2.3. The results in [24] show that the packet delivery ratio of RDG tends to decrease with increasing mobility and group size. Additionally, [28] claims RDG presents considerably lower performance than flooding according to the simulation based comparisons.

As it is presented above, there is not a comprehensive solution for reliable multicasting in mobile ad hoc networks. Frequent topology changes arising from mobility and constrained bandwidth limit the performance of MANET protocols. Thus, a new approach which can adapt variable and adverse network conditions of MANETs is needed to make the reliable group communication paradigm achievable in such networks.

1.2 Solution

The characteristics of mobile ad hoc networks demands new set of network strategies in order to provide reliable group communication. These strategies should be devised

inclusively analyzing the challenges and the advantages of ad hoc settings. The optimum solution can be constructed taking into account each of the challenges presented below.

1- Frequent and unpredictable topology changes due to mobile nodes

In the dynamic environment of mobile ad hoc networks, information derived from the network topology can easily be invalidated. Using this type of information, e.g. network information for routing table, group membership information for multicast structure may impose frequent updates on the protocol states. The delivery of data packets can be hindered during this update process. Furthermore, wireless medium can be congested by the control messages triggered by the changing network state. Thus, group communication paradigms may suffer from wider spread of protocol state information in providing scalable and robust data delivery [29]. A possible solution should minimize the need for information about instantaneous network state to adapt to dynamic environment of mobile ad hoc networks.

We utilized an epidemic technique which is inspired from Anti-Entropy method described in [30] for our solution. In this method, each node in the network periodically selects a random node and recovers differences in database by exchanging data contents with that node. [23, 24 and 31] also used very similar probabilistic methods for multicasting. These probabilistic methods require the nodes have partial or global information of network to select a node randomly from other nodes.

We adapted this method to ad hoc settings exploiting the broadcast nature of wireless medium. In our approach, nodes periodically broadcast gossip messages to their neighbors instead of selecting a random node and sending it the gossip message as unicast. The gossip messages carry the digest of message buffers, e.g. message IDs of the data messages in the buffers of nodes, so size of gossip messages is significantly less than size of data messages. If the gossip receiver discovers that it missed some data then missing data might be

requested by a request message. Then, gossip sender may transmit the data requested back to requester node.

In this scheme, data delivery can be performed without requiring any type of network information, routing information, multicast group information and information of neighboring nodes, which means ids of the nodes in the wireless range. Thus, the problem of maintaining protocol state information is eliminated. Considering many routing protocols, e.g. ODMRP, MAODV, in ad hoc networks use beacon messages, periodic short broadcasts, to be aware of changes in network topology, we believe that use of periodic small-sized gossip broadcasts does not place an undue burden on the network.

2-Congestion and collisions

The broadcast nature of shared wireless medium increases the occurrence probability of collisions. The collision avoidance mechanisms of media access control (MAC) protocols do not support broadcast transmissions as explained in section 4.2.2. For this reason, any solution having reliability concerns should not be susceptible to loss of any broadcast or unicast packet. In our model, nodes can recover their missing data messages by use of periodic gossip broadcasts. Even though a node can not receive gossip broadcasts or transmitted data packets for a long period of time, it has still chance to recover missing data in consecutive gossip rounds.

The wireless medium of ad hoc networks can easily be congested and nodes may contend with each other for the access right of wireless medium to transmit their packets. Especially in dense networks, data delivery may substantially degrade because of high contention in accessing the shared channel. Increasing node density causes longer delays in MAC layer and makes it harder for MAC protocols to provide fair channel access. The long delays in MAC layer may trigger the buffer overflows and hence packet losses. Thus, a multicast protocol should not completely leave the congestion control process to the underlying MAC protocol. Our solution comprises an adaptivity mechanism. This

mechanism controls congestion arising from high node density by adjusting the protocol parameters and preventing transmissions which are not vital.

3-Energy and bandwidth constraints

The efficient consumption of limited energy in all layers is an important design constraint for mobile ad hoc networks to enhance the lifetimes of the battery-powered mobile nodes. The multicast protocols should provide the data delivery with the minimum level of power use eliminating the avoidable transmissions.

The bandwidth is another restrictive factor in ad hoc environment. The available bandwidth for a single node decreases by the increasing number of nodes in the same broadcast domain because of the shared channel. This scarce bandwidth demands minimal control overhead and no redundant data transmissions.

Our solution reduces energy and bandwidth consumption since its stateless nature minimizes network overheads while the use of gossip messages eliminates the redundant data transmissions.

4-Variable network conditions

Mobility of the nodes leads the formation of variable network conditions in mobile ad hoc networks. Some nodes may be isolated from the rest of the network as some others may be congested in a different region of the network as in fig. 1.1.

Our solution can adapt to different network conditions by adjusting the protocol parameters dynamically and bring the isolated nodes up to date by periodic gossip messages when they re-enter into the network coverage.

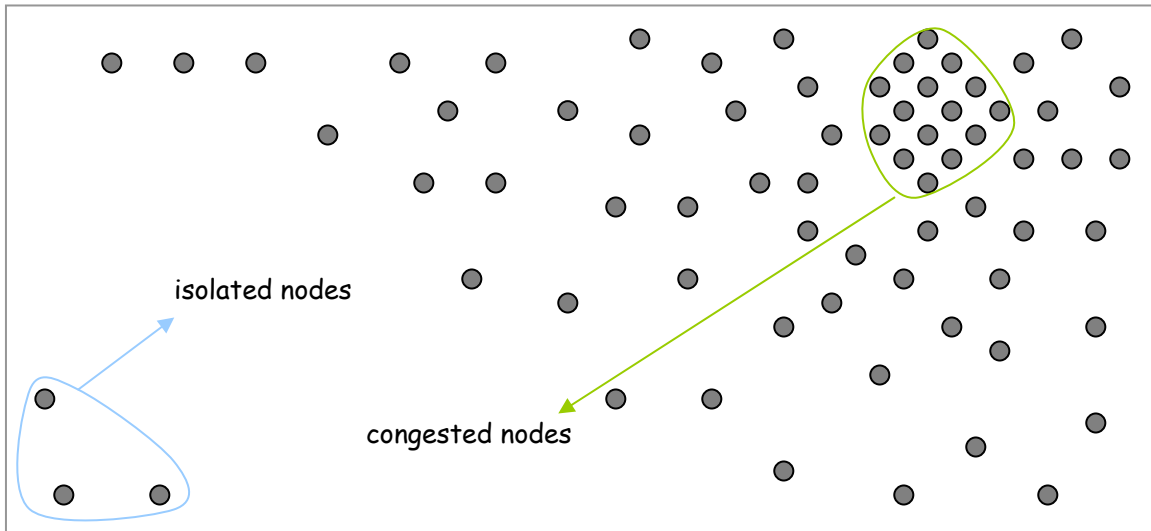


Figure 1.1. A mobile ad hoc network scenario

Solution in brief

In our solution, we have utilized an epidemic method since its stateless character is well matched with the non-deterministic nature, which arises from highly dynamic and unpredictable topology changes, of mobile ad hoc networks. However, we have changed the usual use of gossip messages in this epidemic method. The nodes locally broadcast the gossip messages to the neighboring nodes instead of selecting a random node from a pre-defined list and sending the gossip message to this node as unicast. Thus, our model requires neither having global or partial view of the network nor even information of neighboring nodes for data delivery.

Our solution also provides strong reliability by use of periodic gossip messages which enable the nodes to recover missing data messages. This feature provides the robustness to the protocol against transient adverse network conditions and delivery failures. Besides the reliability, the use of gossip messages improves the bandwidth and energy savings reducing the network overhead.

We have additionally reinforced our model with an adaptivity mechanism. This mechanism can adjust the protocol parameters dynamically according to variable network conditions. It can also make decisions about whether or not to send request messages for preventing congestions.

Consequently, we propose a reliable and adaptive multicasting solution which is well suited to wireless medium and characteristics of mobile ad hoc networks. The drawback of this solution is transmission delay since data is delivered through periodic gossip rounds. Therefore, we do not aim time-sensitive applications such as real-time streaming data.

1.3 Application Scenarios

The first scenario is the campus settings. Assume all students and officers in a university have an application, which enables data sharing, in their mobile devices. The university administration may want to attract the attention of students to an event which will happen in a short time, e.g. talk, concert, or to announce the rescheduled dates of an event like a conference. Furthermore, there may be some students who seek a study friend or a housemate. For all those cases, utilizing a multicast mechanism would be very efficient solution.

The second scenario is planet exploration by mobile sensors. In the near future, thousands of mobile devices may be deployed over a large area to monitor the environment on a planet beyond earth. These mobile sensors can track the activities related to potential life. In such an enormous network, data gathered should be shared with other nodes and relayed outside of the network through end-user access points. Such a network should also be controlled and changed over those access points. Additionally, the nodes stranded for a while should be brought up to date when they rejoin the network. Considering this scenario is on the way [32], there is an urgent need for an efficient reliable group communication paradigm to accommodate the requirements of such networks.

1.4 Contributions

- A new reliable and adaptive multicast protocol, EraMobile, has been defined for mobile ad hoc networks.
- An efficient data dissemination mechanism has been developed by inspiring from the epidemic methods for the new protocol. This mechanism does not demand network information, multicast group information, routing information and neighborhood information for data delivery.
- The protocol has been modeled and implemented on *ns-2* network simulator.
- The comprehensive performance evaluation of the protocol has been performed by a large set of simulations for a wide range of MANET scenarios.
- The effect of the protocol parameters on the performance of the protocol has been analyzed by the simulation studies.
- An adaptivity mechanism has been developed for the protocol to maintain the efficiency in varying node densities.
- The performance comparison has been conducted among our protocol, flooding and MAODV over extensive network conditions. EraMobile outperformed those protocols in terms of throughput, reliable multicast data delivery and overhead efficiency.
- It has been shown that fully reliable multicast delivery is possible, even under challenging conditions of mobile ad hoc networks, with minimal overhead.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Ad hoc Networks

The word of “ad hoc” comes from the Latin phrase meaning “for this [purpose]”. It generally represents a solution adapted to a specific purpose, as opposed to general solutions [33]. In computer networking, ad hoc is a network connection technique used to establish temporary networks among wireless hosts without relying on a fixed infrastructure. The hosts are often mobile and capable of communicating over multiple nodes.

The root of ad hoc networking can be traced back to 1970, when the ALOHAnet [34] computer networking system was developed at the University of Hawaii. The ALOHAnet was built for linking together the universities of the Hawaiian Islands wirelessly. Even if a fixed single-hop network was aimed, the ALOHAnet used a single shared medium for transmission requiring a distributed channel access management. Thus, it led the development of next distributed channel access schemas that were suitable for ad hoc networking [35]. The success and novelty of the ALOHAnet inspired the development of Ethernet and packet radio network (PRnet) in 1973. In early 1990s, the Institute of Electrical and Electronic Engineering (IEEE) renamed the term of packet-radio network as ad hoc network in the development stage of IEEE 802.11 wireless local area network standards [36]. The growing interest revealed the necessity of open standards in this new

area and hence the Mobile Ad hoc Networking (MANET) working group was formed within the Internet Engineering Task Force (IETF) to standardize the protocols and specifications. In 1994, Ericsson Mobile Communications proposed Bluetooth standard to provide short-range, low-power and low-cost wireless connectivity between mobile phones and their accessories. In 1998, five companies Ericsson, Nokia, IBM, Intel and Toshiba formed a Special Interest Group (SIG) to develop the Bluetooth standard providing the interoperability between different vendors.

2.1.1 Characteristics

Infrastructureless Network: The nodes in an ad hoc network do not rely on an infrastructure opposite to cellular networks. The control and management of the network is distributed among the nodes forming the ad hoc network. In other words, all the network functions like routing and security should be handled by the participating nodes in a uniform manner. The absence of central administration also requires the nodes to be organized dynamically for taking part in the network.

Multi-hop Routing: The nodes in the wireless range of each other can communicate directly in an ad hoc network. In order to provide the communication between nodes that are not in the wireless range of each other, the intermediate nodes act as routers relaying the packets toward destination node as it is shown in fig. 2.1.

Dynamic Topology: Nodes are free to move arbitrarily in ad hoc networks. The mobility of nodes may cause randomly and rapidly topology changes at unpredictable times [37]. In order to prevent disruptions in the operation of applications and services, the network connectivity should be maintained even in high mobility.

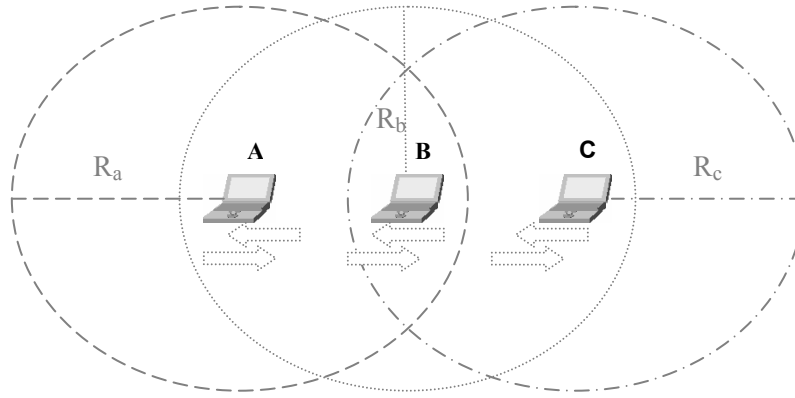


Figure 2.1. Relaying the packets by the intermediate node: In the scenario above, there are three nodes and the dashed lines around the nodes represent their wireless ranges. R is the length of wireless range. Node A and node C are not within each other's wireless range but node B is within the wireless range of both A and C. Thus, node B plays a router role between A and C relaying the messages from the source node to the destination node.

Bandwidth Constraint: The wireless channel is shared by all nodes within the same broadcast domain. Thus, available bandwidth provided to each node varies with the number of nodes in the environment and the amount of traffic load generated.

Resource Constraints: The nodes forming ad hoc networks are often battery powered devices. Considering most of handheld devices like PDAs and cell phones have short range wireless interfaces, there are also constraints on computing power and buffer storage.

High Error Rates: The bit error rate in wireless medium is very high (the order of 10^{-5} – 10^{-3}) compared to the error rate in wired networks (the order of 10^{-12} – 10^{-9}) [38]. Additionally, other inherent problems of wireless medium such as noise, interference, fading and

asymmetric links make the realized throughput of wireless communications often much less than a radio's maximum transmission rate [37].

Limited Security: Additional to the shared wireless medium, the absence of central administration brings even more security challenges to ad hoc networks. Considering the multi-hop and energy constrained nature of ad hoc networks, it is obvious that denial of service attacks can cause more serious problems in ad hoc environment compared to wired counterparts.

2.1.2 Applications

Ad hoc networks have wide range of application areas because of the important advantages they offer. Their infrastructureless character highlights them as an appropriate solution for the cases where infrastructure for the communication is not available, too expensive or unreliable. They can be rapidly deployed and maintained by the self-organizing capability of the participating nodes without requiring a central configuration mechanism. They can also be used for extending the coverage of infrastructure-based networks like cellular communication networks. Moreover, ad hoc networks can integrate many different devices and make their services available to other nodes of the networks. In the future, the ad hoc networks are expected to play more important roles as part of the coming 4G architecture by enabling the users to access information and communication anytime, anywhere and from any device [39]. Table 2.1 presents an overview of current and possible ad hoc network applications based on [39].

Application	Possible scenarios/services
Tactical networks	• Military communication and operations • Automated battlefields
Emergency services	• Search and rescue operations • Disaster recovery • Replacement of fixed infrastructure in case of environmental disasters • Policing and fire fighting • Supporting doctors and nurses in hospitals
Commercial and civilian environments	• E-commerce: electronic payments anytime and anywhere • Business: dynamic database access, mobile offices • Vehicular services: road or accident guidance, transmission of road and weather conditions, taxi cab network, inter-vehicle networks • Sports stadiums, trade fairs, shopping malls • Networks of visitors at airports
Home and enterprise networking	• Home/office wireless networking • Conferences, meeting rooms • Personal area networks (PAN), Personal networks (PN) • Networks at construction sites
Education	• Universities and campus settings • Virtual classrooms • Ad hoc communications during meetings or lectures
Entertainment	• Multi-user games • Wireless P2P networking • Outdoor Internet access • Robotic pets • Theme parks
Sensor networks	• Home applications: smart sensors and actuators embedded in consumer electronics • Body area networks (BAN) • Data tracking of environmental conditions, animal movements, chemical/biological detection
Context aware services	• Follow-on services: call-forwarding, mobile workspace • Information services: location specific services, time dependent services • Infotainment: touristic information
Coverage extension	• Extending cellular network access • Linking up with the Internet, intranets, etc.

Table 2.1. Mobile ad hoc network applications [39].

2.2 Epidemic Communication

An epidemic is usually a widespread disease that affects many individuals in a population [40]. In biological epidemics, the disease spreads to a wider population very rapidly making minimal contact with the originally infected peer. Once a few peers have the infection, the epidemic spreads like a gossip from peer to peer, until the whole population is infected. After some threshold level, the spread can not be stopped even if the original source of the infection is isolated or extinguished from the population [40].

Epidemic algorithms imitate this behavior in order to achieve the scalability and robustness of natural epidemics in the propagation of data through a computer network. In general, the infective sources, the peers that have new information, send the update message to randomly chosen members. Then, those members forward the message to other randomly chosen nodes and the information spreads exponentially fast through the network. In order to decide which members need for updates, the nodes first gossip about the data before transmission. This procedure is performed by exchanging the digest, in other words snapshot, of their data. The messages carrying digest of data are called “gossip messages” and so epidemic protocols employing this gossip mechanism can be referenced as gossip-based protocols. After gossiping, the nodes may request some messages they missed from gossip sender (*pull-based* approach) or they may send the messages missed by gossip sender to it (*push-based* approach). The combination of both methods is also possible.

Epidemic communication mechanisms were first proposed for spreading updates in a replicated database. Anti-entropy is an epidemic communication strategy introduced for achieving and maintaining consistency among the sites of a widely replicated database. Compared to deterministic algorithms for replicated database consistency, this strategy also reduces network traffic [30]. Anti-entropy has been proposed as a mechanism that runs in background for recovering errors of direct mail in large network, as well [41].

Epidemic or gossip style of communication has been used for several purposes. Examples include use of epidemic communication techniques for group membership tracking [42], for support of replicated services [43] and for failure detection. [31, 44, 45 and 46] used epidemic algorithms to address multicasting in wired networks. The epidemic methods were also utilized by [23 and 24] to provide reliable multicasting in mobile ad hoc networks.

2.3 Related Work

A route discovery mechanism is proposed by use of gossiping [47]. Instead of flooding the network, each node broadcasts a route request message with some probability. When a node having route information or the destination node itself receives the route request message, it sends this information back to the source node. They present the behavior of gossiping by different parameters. The p , probability of broadcasting the request to the neighbors, and k , number of hops broadcasting the request with probability 1 before continuing to gossip with probability p , are the parameters mainly investigated. They ensure that all nodes get the message with the probability about 0.65-0.75 while k is 4 causing 35-25% fewer message load than flooding.

[48] presents a set of data dissemination protocols for wireless sensor networks. These protocols aim to eliminate redundant data transfers using meta-data negotiations before transmitting original data. The nodes employ three types of messages, ADV, REQ, and DATA, to communicate. Once a SPIN node has new data to share, it first advertises this fact by transmitting an ADV message to its neighbors. This ADV message carries the meta-data about original data. The node that has received an ADV message sends a REQ message back if it wishes to receive some actual data. Actual data is transmitted within DATA message with a meta-data header to the requester(s). Also, SPIN nodes can poll the system resources to learn the amount of available energy to them. They use this

information for making decision about using their resources efficiently. The protocols of SPIN family are SPIN-PP, SPIN-EC, SPIN-BC and SPIN-RL. SPIN-PP and SPIN-EC use the point-to-point transmission media while the others use the broadcast media. SPIN-PP works in three stages as explained above. SPIN-EC employs energy-conservation heuristic additional to SPIN-PP. SPIN-EC behaves just like SPIN-PP when there is not an energy shortage. However, when energy level of a node approaches to a low-energy threshold, the participation of this node in the network is limited with the current energy state. That is, this node can only participate in the protocol stages if its energy level allows it without going below the low-energy threshold. SPIN-BC improves upon SPIN-PP by exploiting the broadcast nature of wireless medium. When the interference-free broadcast communication is possible, SPIN-BC nodes use cheap one-to-many communication method. SPIN-RL, the reliable version of SPIN-BC, is developed for maintaining data dissemination even when nodes are subject to packet losses and asymmetric communication links.

The negotiation approach and the resource-awareness nature of SPIN protocols allow very efficient data dissemination in wireless sensor networks compared to flooding and classic gossiping. However, there is a lack of mechanism which can enable the nodes to choose and change protocol type observing the network conditions in an adaptive manner. Therefore, SPIN nodes can not adapt to changing conditions of mobile ad hoc networks. For instance, the packet delivery ratio of a node which executes SPIN-BC protocol falls below 50% on a lossy network while SPIN-RL protocol makes the 100% packet delivery achievable in the same network settings. Additionally, SPIN protocols except SPIN-RL can not guarantee delivery of data if the destination nodes are far away from the source node and the intermediate nodes are not interested in the data of source node [49]. The behavior of SPIN protocols should also be investigated for mobile networks in variable node densities.

Anonymous Gossip (AG) is one of the former protocols utilizing the gossip technique for reliable multicasting in ad hoc networks [23]. Indeed, it is a reliability improvement rather than a multicast delivery protocol. AG can be implemented on any on-demand multicast routing protocol that provides best-effort data delivery. AG operates in two phases. In the first phase, the underlying multicast protocol, MAODV in the implementation, is used to disseminate the multicast data to the group without having a reliability concern. AG, in the concurrent phase, recovers the missing messages running on top of this multicast protocol. The recovery phase is performed by use of gossip mechanism. Each node randomly selects one of its neighbors, nearest nodes with higher probability and distant nodes with lower probability, and sends a gossip message to it. If the receiver node is not a member of multicast group then it forwards the message one of its neighbors, which is a member of the multicast group addressed by the gossip message. When a member node receives a gossip message, it randomly decides to either accept the gossip message or forward it. A node accepting a gossip message compares the content of gossip message, which includes the ids of message(s) missed by the sender, and the sequence number of the next expected message with its history of messages received. If it realizes a message sought by the gossip initiator in its history then it unicasts this data message back to the gossip initiator as the gossip reply.

AG proposes a new way to provide reliable multicasting in ad hoc networks. The simulation results show that it greatly improves the packet delivery of pure MAODV. However, its performance is highly dependent on MAODV and this makes it impossible to analytically predict its probabilistic delivery ratio [26].

Route Driven Gossip (RDG) [24] protocol uses a pure gossip schema for reliable multicasting in ad hoc networks. Unlike AG, RDG does not rely on a multicast routing protocol but on an on-demand unicast routing protocol like DSR. It disseminates the data in the periodic gossip messages together with membership information and negative

acknowledgements. The gossip receivers are randomly selected from the partial list of group members. RDG does not employ an additional mechanism to form the list of group members. Instead, it utilizes the underlying routing protocol to collect group membership information together with routing information. The gossip messages contain new data, the ids of missing messages, the view on the membership and whether the node wishes to leave the group. The gossip receiver first updates its view by removing and adding the nodes then delivers the data and responds to gossip sender by sending missing messages if the node has the data packet requested. RDG aims to achieve a probabilistic reliability which is expected to be predictable based on simple information like packet loss ratio. The paper presents an additional protocol named TA-RDG, topology aware RDG. This variant uses partial topological information to utilize a similar heuristics to AG in order to pick a closer member with higher probability for gossiping. Therefore, different weights are assigned to the members in an active view according to the length of the routing path to them.

The major novelty of RDG is its analytical performance estimation schema. The simulation results follow the trend of these analytical predictions very well. However, there is little information about the overhead associated with the protocol [26]. The new data packets are transmitted within gossip messages additional to gossip content for pre-defined times by different members. This situation may result in redundant data transmissions and increase the overhead. It may also degrade the performance of the protocol in dense ad hoc networks which are highly open to congestion.

In the end, if one compares our solution with the previous approaches, it is seen that our model is quite different than the approaches proposed above. Even if its gossip method is similar to the negotiation technique of SPIN protocols, SPIN protocols were designed for wireless sensor networks and do not include an adaptivity mechanism for variable network conditions. They also employ neither periodic negotiation messages nor buffer management policy for recovering missing packets. EraMobile is also different than AG

and RDG although they all use gossip method for reliable data dissemination. Our solution is not based on any underlying routing protocol. It disseminates and recovers the messages by use of a full gossip based schema. It requires neither routing information nor neighbor awareness for data dissemination. However, AG requires the information about neighboring nodes while RDG needs the partial view of the group member nodes. Our solution exploits the broadcast nature of wireless medium to disseminate the gossip messages, instead of having information about the possible gossip receivers apriori. The performance of these protocols is also limited by underlying routing protocols. Additionally, these approaches claim they provide probabilistic reliability while our solution aims to provide full reliable packet delivery.

Chapter 3

EraMobile

3.1 Overview

EraMobile aims to provide reliable multicasting in mobile ad hoc networks even under adverse network conditions. It distributes the burden of data dissemination among the nodes instead of leaving it completely on the source by utilizing a peer-to-peer epidemic mechanism. The source node just broadcasts the new data messages once generated. Then, all nodes participate in the data dissemination through periodic gossip rounds. This scheme prevents the single point-of-failures and bottlenecks. The nodes of EraMobile have the ability to adapt to variable node densities by setting the protocol parameters to appropriate values. In sparse networks, the nodes can just focus on delivering the data in a reliable manner without having a congestion concern. In dense networks, the nodes should take into account the broadcast nature of wireless medium which is highly prone to congestion. EraMobile reduces the traffic only generated by the nodes subject to congestion in dense zones suiting the heterogeneous environment of mobile ad hoc networks. This technique spreads the overhead in space and time in order to avoid congestion. Since EraMobile performs the data delivery by equal participation of the nodes, such a reduction in the traffic rate increases delays. However, EraMobile has not already been developed for delay-sensitive applications like real-time streaming data.

3.2 EraMobile Units

The operation of EraMobile is handled by three main units. The data dissemination and recovery of missing data messages are performed by *Data Dissemination* unit. The *Adaptivity* unit is responsible for setting the operating modes of a number of sub-components dynamically based on network conditions. The *Buffer Management* unit administrates the buffers of the protocol.

3.2.1 Data Dissemination Unit

Data dissemination is performed by use of gossip messages. A gossip message carries the digest of the sender's data buffer contents. Every node in the network broadcasts gossip messages to their neighboring nodes at a rate determined by the *gossip interval* parameter. Upon receipt of a gossip message, a gossip receiver compares the list of messages in the digest of the gossip message with the list of its own missing messages. If it notices its missing message(s) in the digest, the node may request the related packet(s) sending a request message to the gossip sender. Then, the gossip sender may transmit the data message(s) to the requester.

The main schema mentioned above can be divided into three phases: Gossip digest construction, gossip propagation and data distribution.

a-) Gossip Digest Construction

The protocol proceeds in gossip rounds initiated by the gossip timer. The period of the gossip rounds is a parameter of the protocol. In each gossip round, a node scans its data buffer and collects the ids of the messages whose *gossip counts*, message-specific variables incremented in each gossip round, are less than the *stability-threshold* value. The *stability-*

threshold parameter is used to determine how many times id of a message should be put into gossip messages, in other words, how many rounds a message should be kept in the data buffer. The example in fig.3.1 illustrates the gossip digest construction process of a node for a single sender scenario. In the scenarios having multiple senders and multicast groups, sender and group ids are also put into gossip messages.

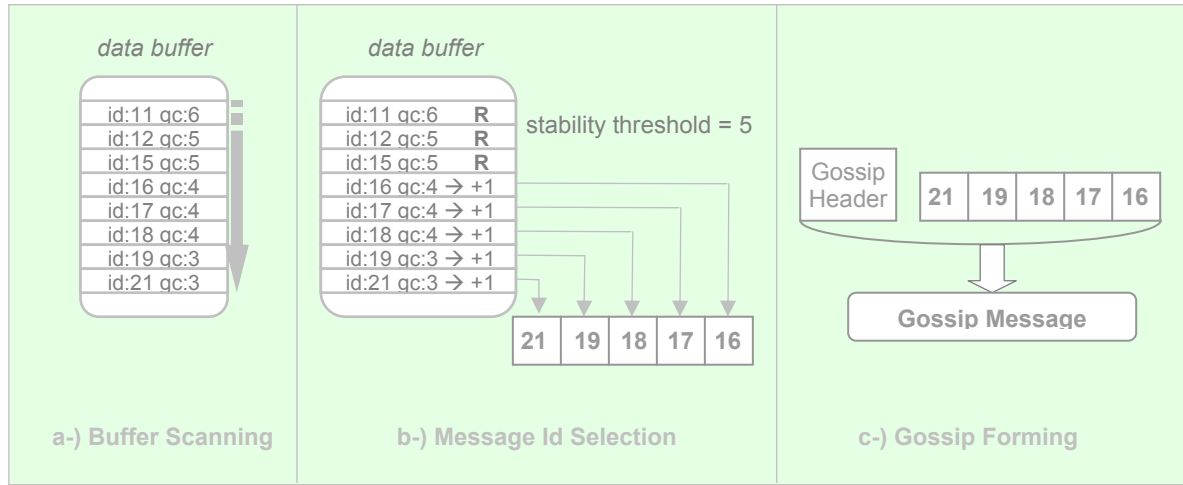


Figure 3.1. Gossip digest construction: a-) The data buffer of the node is scanned to determine the status of the messages. The id means unique sequence number of the message and the gc represents gossip count variable which is incremented in each gossip round. b-) The ids of the messages whose gossip counts are less than the stability-threshold, which equals to 5, are put into gossip message and their gossip counts (gc) are increased by one. Other messages are marked as removable (R) to be removed from the buffer. The missing messages placed between message 12 and message 15 are declared as lost. c-) Gossip message is formed by adding header to gossip data.

b-) Gossip Propagation

Gossip messages are propagated to all nodes in the wireless range of the gossip sender. This method does not require having knowledge of which nodes are in the wireless range before transmission since it exploits the broadcast nature of the wireless medium. A node,

which has gossip message to send, just puts the message on air and its neighbors, nodes in the wireless range of this node, receive the gossip message. The small size of gossip messages and random delays on gossip sending times reduce the probability of collisions and packet losses that may arise from broadcasts to neighbors. Fig. 3.2 illustrates the propagation of gossip messages.

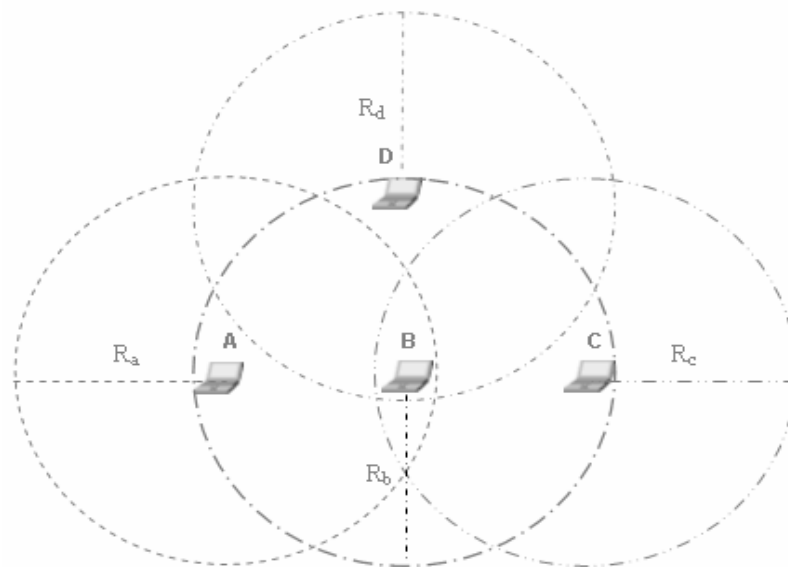


Figure 3.2. Propagation of gossip messages: In the scenario above, there are four nodes and the dashed lines around the nodes show their wireless ranges. R represents the radius of each wireless range. Node B can send a gossip message to all other nodes by a single broadcast since they are all in the wireless coverage of B although B does not need to be aware of other nodes for gossiping. Node A, C and D can only send gossip message to B since they do not cover each other but B.

c-) Data Distribution

A new data message is broadcasted by the source node to its neighbors just once when the data is originated. The goal of this process is to increase the reliability of the protocol

and propagation speed of the data by enabling more nodes to have the new data message during gossiping. Then, data distribution is completely performed by use of gossip mechanism through peer-to-peer communications. Upon receipt of a gossip message, nodes compare the message ids placed in the gossip contents with the ids of their missing messages, which are determined by comparing the id of the last received data message with the greatest id in gossip contents. If nodes notice one or more message ids which exist in the gossip contents but in their data buffer then they may request the missing message(s) from the gossip sender. This requesting process is performed by use of request messages which carry the missing message id(s). The number of messages that a node can request in one gossip round is limited by a parameter. Thus, a node, which has missed many packets and realized this situation upon receipt of a gossip message, is not allowed to congest the network by high amount of request messages.

A node that has received a request message may transmit the related data to the requester. The transmission decision is based on the number of data packets transmitted in one gossip round. The maximum number of data packets which can be transmitted in a gossip round is determined by a parameter. A node does not answer the request messages if it has reached the transmission limit for one gossip round. Instead, the transmission can be performed by different nodes over several gossip rounds, spreading the overhead in space and time. The fig. 3.3 illustrates the data distribution.

3.2.2 Buffer Management Unit

The buffer management has an important role on the performance of the protocol. The nodes should have sufficient number of data messages within their buffers in order to contribute to recovery of missing messages of the other nodes. However, mobile nodes may have memory constraints so the size of buffers should not exceed the limits. Additionally, considering the fact that gossip messages carry the digest of data buffers, a larger data

buffer leads to bigger-size gossip messages, thereby increasing the overhead and the congestion probability in dense networks.

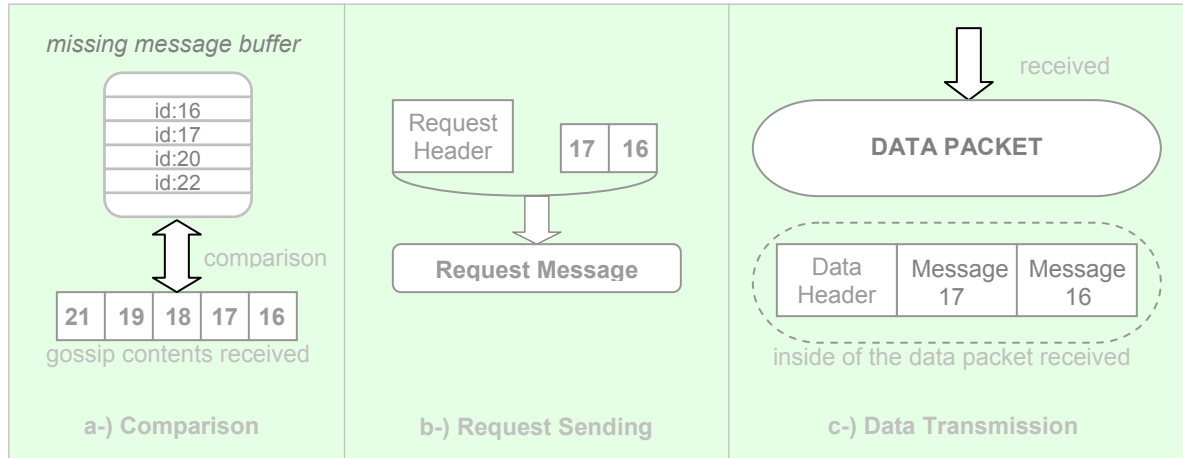


Figure 3.3. Data distribution: a-) The node which has received the gossip message compares the message ids placed in the gossip contents with the ids of missing messages kept in the *missing message buffer* b-) The node puts the ids of the missing messages found in the received gossip contents into the request message and sends it to the gossip sender c-) The node receives the data messages transmitted by the gossip sender

Buffer management unit carries out critical tasks for the reliability and the efficiency of the protocol. It maintains both buffers of the protocol and performs data delivery in FIFO order as explained below.

- *Buffer Maintenance:* The data buffer of the protocol is maintained by use of a garbage collection mechanism which is operated periodically at a pre-defined rate. This mechanism makes a decision about the status of every message based on its gossip count, which is a message-specific variable incremented in each gossip round. If the gossip count of a message is greater than or equal to the stability threshold then it is removed from the data buffer. A missing message which could not be recovered during *stability threshold* times gossip rounds is declared as *lost* and the protocol does not try to recover it any more.

Besides the data buffer, EraMobile maintains a second buffer for missing messages. Upon receipt of a data message, if there is a gap between the sequence number of this message and the sequence number of the last received one, the protocol puts the id(s) of the missing message(s) into the missing message buffer. This buffer is used for making fast comparisons between the missing messages and the list of messages in the contents of gossip messages received. The garbage collection mechanism also controls the missing message buffer. It removes a message from this buffer when the missing messages is received or declared as *lost*.

- *FIFO Order Delivery*: The messages in the data buffer are queued in FIFO order. Upon receipt of a data message, the buffer management unit performs a delivery operation on the data buffer. The messages are delivered to upper layer if they are in FIFO order and there is not a gap between them. Otherwise, they are hold in the buffer until the missing messages causing the gap are received or declared as *lost*.

3.2.3 Adaptivity Unit

The mobility of the nodes in a mobile ad hoc network causes the occurrence of variable and unpredictable network conditions. The adaptivity unit enables the protocol to adapt to the changes by adjusting the parameters of the protocol dynamically. The node density observed is taken into account by the adaptivity unit in decision making. It was chosen as the adaptation criterion based on the preliminary result analysis of substantial number of simulation studies. We observed that node density has significant influence on the performance of our protocol. The reason is that, since the data delivery is performed by the participation of all members through gossip rounds, high density increases the traffic load that nodes expose while low density results in poor network connectivity and hence inadequate data delivery.

As it is stated previously, EraMobile does not employ a separate mechanism to have the information of neighbors around a node. Instead, it utilizes the periodic gossip broadcasts and request messages received. The senders of both messages are considered as neighboring nodes. EraMobile counts those senders during pre-defined gossip rounds to make sure every neighboring node has broadcast at least one gossip message. Then, it calculates the average number of neighbors. However, the adaptivity unit does not set the protocol parameters each time the node density is calculated. Instead, it periodically operates waiting for an *adaptivity period*, in which node density can be calculated several times, to prevent oscillations. Then, it uses the average of pre-calculated node density values to determine the density level.

We defined three separate density levels, low, normal and high, according to the results obtained through ns simulations and taking into account the related work [50, 51 and 52]. The density levels are based on the number of neighbors observed around a node. The low density level comprises the nodes having less than 6 neighbors. The normal density level is defined for the nodes having neighbors between 6 and 20. The nodes having more than 20 neighbors are accepted in the high density level.

In low density level, the spread of data all over the network is likely to be a challenging task because of the poor network connectivity. As in an epidemic, the nodes should utilize each contact opportunity to disseminate the data. In mobile ad hoc networks, the time that two nodes will spend within the wireless range of each other is variable. It may be too small to be aware of each other or to complete the data exchange. Thus, a node needs to broadcast the gossip messages more frequently to catch another mobile node as soon as their wireless ranges cover each other. The *gossip interval* parameter of the protocol should be decreased for more frequent gossip broadcasts. Additionally, the *request limit* and the *transmission limit* for a gossip round should be broadened for allowing the nodes to perform data transmission as much as need in the congestion-free environment of sparse

networks. The low node density may also cause a number of nodes to be isolated from the rest of the network for some time. Those nodes should be brought up to date when they get rid of isolation. Therefore, the messages should be kept for a longer time in the data buffers of the nodes which are able to receive data messages. The isolated nodes should also wait for a longer time to declare the missing messages *lost*. The adaptivity unit increases the *stability threshold* value to deal with such situations. A higher stability threshold value retards the removal of the messages from the data buffer and declaration of missing messages *lost*.

In normal density level, nodes maintain the protocol parameters in moderate values since they have suitable network conditions for communication.

In high density level, even mild traffic rates generated by the nodes may make the network congested. Such a congested network causes high number of packet losses arising from collisions and ultimately wastes limited bandwidth. The adaptivity unit decreases the traffic rate of the nodes experiencing congestion, thereby spreads the overhead in space and time. The reduction in traffic rate is carried out in two dimensions: extending the gossip intervals and limiting the number of request and data messages transmitted in a gossip round. Extending the gossip intervals decreases the number of gossip broadcasts. Besides, it reduces the traffic of request and data messages indirectly. The request and transmission limits per round are also decreased in order to alleviate the congestion. Additional to request limit, the adaptivity unit defines a probability value, *request sending probability*, which is reduced by the node density level, for request sending. This probability is used by the nodes to make a decision on whether to send a request message. *Request sending probability* aims to prevent a request explosion when a gossip message, which includes the ids of newly generated data messages, broadcasted in a dense zone.

The final congestion avoidance technique is putting random *jitters* based on the node density onto message transmission times. The adaptivity unit puts random *jitter* onto transmission time of each request and gossip message according to the formula below.

$$jitter(sec) = k * (\# neighbors around the node) / 100;$$

(*k* is a constant randomly chosen from a uniform distribution on the range 0 to 1)

EraMobile also has an adaptation mechanism for energy conservation. Each node adds a small amount of time, *gossip interval addition*, to its gossip interval value in every gossip round until it reaches a pre-defined upper limit, *gossip interval upper limit*. Upon receipt of a data message, the extended value of gossip interval is reset to its original value. This method aims to reduce energy consumption by eliminating redundant gossip broadcasts when the data traffic is stopped. The addition value and the upper-limit for the gossip intervals are determined based on the node density level. In dense networks the gossip interval is extended more quickly and the upper-limit is chosen larger to assist the congestion control mechanism.

3.3 Optimizations

We enhanced our protocol with the following optimizations in order to reduce the overhead by shrinking size of the gossip messages and eliminating redundant transmissions.

Optimization 1: Gossip Size Reduction. In the investigation of the simulation results, we observed that message ids included in the gossip messages are generally sequential. Thus, instead of putting them one-by-one into the gossip messages, we developed a technique to combine them in a size-reduced form. In this technique, when a node faces a sequential

message ordering within its data buffer in the gossip preparing phase, it puts the last element of this ordering into the gossip message by multiplying “-1” and then places the first element of the ordering right after this one. If there is not a sequential ordering, the ids are put into gossip message one-by-one. A gossip receiver decompresses the ids when it notices a minus sign by using the id preceded by the minus sign and the next id. This operation is illustrated in fig. 3.4.

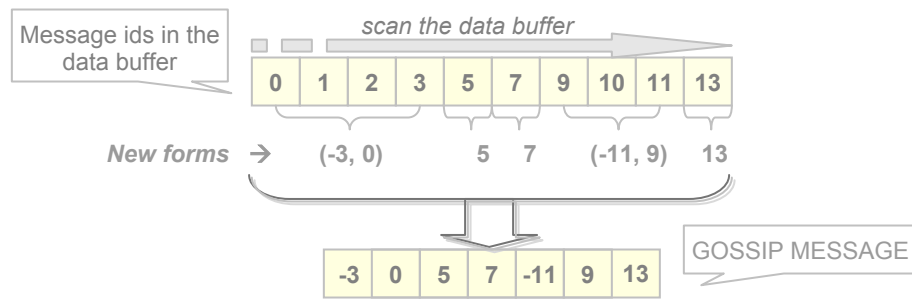


Figure 3.4. Gossip Size Reduction: The sequential id orderings found through the data buffer scanning are transformed into new form and put into the gossip message. In this scenario, size of the gossip message is reduced to 7 from 14.

Optimization 2: Request Time Limit. When a node sends a request message for a data packet, it may receive, before the reception of the requested data packet, some more gossip messages from different nodes having the id of the requested packet. In this case, it may request the same message for multiple times. We use a parameter, *request-time-limit*, in order to prevent such redundant request and data transmissions. This parameter holds a time value and a node can not request the message, which has been requested at *request-sent-time*, during the interval of *request-time-limit*.

3.4 Properties and Algorithm

EraMobile performs the multicast operation satisfying following properties:

- Atomicity: Indeed, EraMobile does not provide atomicity guarantee but it satisfies this property by obtaining fully reliable data delivery in most of the scenarios.
- Ordering: Each participant in the group delivers the data messages in FIFO order. In other words, multicast messages originated from a sender are delivered by each member in the order of generation at the sender.
- Scalability: As the group size increases, overheads of the protocol remain almost constant or grow slowly compared to other multicast protocols.
- Throughput stability: In the scenarios where 100% data delivery can not be achieved, the throughput variation observed at the participants of a group is considerably low.
- Multicast stability detection: EraMobile detects the stability of multicast messages. That is, each message is kept in the data buffer for a pre-defined time till the protocol makes sure this data message is not need by any member. Then, such messages can be safely garbage collected. If needed, the application can be informed as well.

The message formats and the algorithm of EraMobile: The events, for which a specific piece of code is executed, main data structures and special variables of EraMobile are presented in table 3.1, 3.2 and 3.3 respectively.

Event	Description
<i>Data Reception</i>	Reception of a data message
<i>Request Message Reception</i>	Reception of a request message
<i>Gossip Round</i>	Gossip dissemination procedure operated periodically
<i>Gossip Message Reception</i>	Reception of a gossip message
<i>Garbage Collection Run</i>	Garbage collection procedure operated periodically
<i>Adaptivity Run</i>	Adaptivity procedure operated periodically
<i>Data Generation</i>	Generation of multicast data by the source node
<i>Deliver</i>	Delivery of data messages to upper layer
<i>Data Insertion</i>	Insertion of a data message into the data buffer

Table 3.1. The events of EraMobile

Data Structure	Description
<i>DATA_MESSAGE</i>	Data message received
<i>GOSSIP_MESSAGE</i>	Gossip message to be send
<i>REQUEST_MESSAGE</i>	Request message to be send
<i>DATA_BUFFER</i>	Data buffer of the current node
<i>MISSING_DATA_BUFFER</i>	Missing data buffer storing ids of missing messages
<i>NEIGHBOR_LIST</i>	List of neighboring nodes
<i>TEMP_BUFFER</i>	Temporary storage space

Table 3.2. The data structures of EraMobile

Special Variable	Description
<i>Message_Id</i>	Unique id of each data message
<i>Missing_Id</i>	Id of missing data messages
<i>Message_MGroup_Id</i>	Id of the targeted multicast group by the data message
<i>My_MGroup_Id</i>	Multicast group id of the current node
<i>My_LastReceived_Mid</i>	Id of the last received data message by the current node
<i>Gossip_Count</i>	Gossip count value of each data message
<i>Adaptivity_Period</i>	Period of adaptivity mechanism
<i>Gossip_Round</i>	Gossip round counter increased in each gossip round
<i>Stability_Threshold</i>	Stability threshold value
<i>Request_Send_Probability</i>	Probability value assigned by the adaptivity unit

Table 3.3. The special variables of EraMobile

Message Formats:*DATA_MESSAGE:*

Multicast Group Id 2 octets	Message Type 1 octet	Size of Message 2 octets	Data Contents Max. 2048 octets
-----------------------------------	----------------------------	--------------------------------	-----------------------------------

GOSSIP_MESSAGE:

Message Type 1 octet	Size of Message 2 octets	Gossip Contents [(group id, sender id, message ids), (group id, sender id, message ids) ...] Max. 2048 octets
----------------------------	--------------------------------	---

REQUEST_MESSAGE:

Message Type 1 octet	Size of Message 2 octets	Request Contents Max. 2048 octets
----------------------------	--------------------------------	--------------------------------------

Multicast Group Id: The id of multicast group to which the sender of message belongs.

Message Type: The type of EraMobile specific message (*DATA*, *GOSSIP*, *REQUEST*)

Size of Message: The size of the payload

Data Contents: The application data

Gossip Contents: The data message ids with the sender and group ids of these messages

Request Contents: The ids of requested data messages with their sender and group ids

Pseudo-code of the algorithm for each event:

<i>Data Generation:</i>
-Data Insertion (<i>DATA_MESSAGE</i>) -Broadcast <i>DATA_MESSAGE</i>

Data Reception:

```

If ( Message_MGroup_Id = My_MGroup_Id ) then;
  If ( DATA_MESSAGE is not duplicate ) then;
    -Data Insertion ( DATA_MESSAGE )
  endif
endif

```

Request Message Reception:

```

If ( transmission limit is not exceeded for this round ) then;
  If ( DATA_MESSAGE requested is still in DATA_BUFFER )
    -Send DATA_MESSAGE requested to the request sender
    -Add request sender id into NEIGHBOR_LIST
  endif
endif

```

Gossip Round:

```

-Scan DATA_BUFFER
If ( Gossip_Count attached to DATA_MESSAGE < Stability_Threshold ) then;
  -Put Message_Id of DATA_MESSAGE into TEMP_BUFFER
  -Increase Gossip_Count attached to DATA_MESSAGE
endif
If ( size of TEMP_BUFFER > 0 ) then;
  -Compress Message_Ids ( Optimization 1 )
  -Put compressed Message_Ids into GOSSIP_MESSAGE
  -Broadcast GOSSIP_MESSAGE
endif
-Increase Gossip_Round

```

Adaptivity Run:

```

-Calculate number of neighbors
If ( ( Gossip_Round % Adaptivity_Period ) = 0 ) then;
  -Calculate average node density
  -Set parameters to proper values according to the node density
endif
-Reset NEIGHBOR_LIST

```

Gossip Reception:

```

If ( Message_MGroup_Id = My_MGroup_Id ) then;
-Decompress Message_Ids in the gossip message received ( Optimization 1 )
  If ( there is any Message_Id in the gossip contents > My_LastReceived_Mid ) then;
    -Put Message_Id into MISSING_DATA_BUFFER
  endif
  If ( size of MISSING_DATA_BUFFER > 0 ) then;
    -Compare Message_Ids in the gossip contents with Missing_Ids in MISSING_DATA_BUFFER
    If ( Message_Id = Missing_Id ) then;
      If ( request limit is not exceeded ) then;
        If ( Message_Id is not recently requested ) then;           ( Optimization 2 )
          -Put Message_Id into TEMP_BUFFER
        endif
      endif
    endif
  endif
  If ( size of TEMP_BUFFER > 0 ) then;
    -Put content of TEMP_BUFFER into REQUEST_MESSAGE
    If ( Request_Send_Probability > randomly uniformly selected number from (0,1) ) then;
      -Send REQUEST_MESSAGE to the gossip sender
    endif
  endif
endif
endif

```

Deliver:

```

-Scan DATA_BUFFER
If ( there is no gap between Message_Ids ) then;
  -Deliver messages in order to the upper layer
else
  If ( Missing_Ids causing gap have been previously declared lost ) then;
    -Deliver messages in order to the upper layer
  endif
endif
endif

```

Garbage Collection:

```

-Scan DATA_BUFFER
If ( Gossip_Count attached to DATA_MESSAGE = Stability_Threshold ) then;
  If ( there is any Missing_Id < Message_Id of DATA_MESSAGE ) then;
    -Declare Missing_Id as lost
    -Remove Missing_Id from MISSING_DATA_BUFFER
  endif
  -Remove DATA_MESSAGE from DATA_BUFFER
endif
-Deliver()

```

Data Insertion (DATA_MESSAGE)

```

-Put DATA_MESSAGE into DATA_BUFFER
If ( Message_Id < My_LastReceived_Mid ) then;
  -Remove Message_Id from MISSING_DATA_BUFFER
endif
If ( Message_Id > ( My_LastReceived_Mid + 1 ) ) then;
  -Put Missing_Ids between Message_Id and My_LastReceived_Mid into MISSING_DATA_BUFFER
  -Assign Message_Id to My_LastReceived_Mid
endif
If ( Message_Id = ( My_LastReceived_Mid + 1 ) ) then;
  -Assign Message_Id to My_LastReceived_Mid
endif

```

Chapter 4

EXPERIMENTAL SETUP

4.1 Simulation Methodology

The performance of the EraMobile was evaluated in comparison with plain flooding and Multicast ad hoc on-demand distance vector (MAODV) routing protocol [5]. The flooding was chosen because it is like a common currency in performance evaluation of multicast and broadcast protocols proposed for ad hoc networks. It is involved in most of the performance comparisons and generally shows best packet deliver ratios [12] [14]. Therefore, comparing EraMobile with flooding could give a general idea about the performance, especially packet delivery performance, of EraMobile. MAODV was chosen since it causes the lowest overhead among the leading multicast protocols [12]. A comparison with MAODV could be helpful to investigate the overhead efficiency of EraMobile.

In flooding, data is disseminated by flood waves. When a node receives a new data message, it simply broadcasts it to all the neighbors. Every node keeps a cache of previously received messages so that a node does not broadcast the same message twice. In order to reduce the probability of collisions, we utilized a simple collision avoidance technique in our flooding implementation. A node which has just received a new message waits for a small random time interval before broadcasting it.

MAODV is the multicast capable version of the AODV unicast routing protocol [53]. It performs the multicast operation by locating the nodes of a multicast group in a common tree structure. Every multicast group has a group leader which is responsible for maintaining the group connectivity by periodically broadcasting group hello (GRPH) packets. Nodes also broadcast hello packets with the time-to-live (TTL) value of 1 to maintain the local connectivity. The group leader keeps and updates a sequence number which is broadcast within GRPH packets. This sequence number is utilized to ensure that the freshest route to the multicast groups is used. Nodes, willing to join the multicast group or having data to send to a multicast group without the knowledge of a route to that group, send a route request (RREQ) packet to the group leader if its address is known by the nodes; otherwise, the RREQ packet is broadcast. Both member and non-member nodes participate in broadcasting of the RREQ packets across the network and update their routing tables with the information of the source node. The RREQ is answered with a unicast route reply (RREP) by a member of the multicast group. A node may receive multiple RREP packets. In this case, it chooses the most recent and the shortest path from all the RREPs received and sends a multicast activation (MACT) packet back. The MACT packet activates the route from the source node to the replier node informing the intermediate nodes. If a source node does not receive a RREP packet in spite of several route request attempts, it assumes there are not any other multicast members reachable and declares itself the group leader. Fig. 4.1 illustrates the join process in a multicast tree.

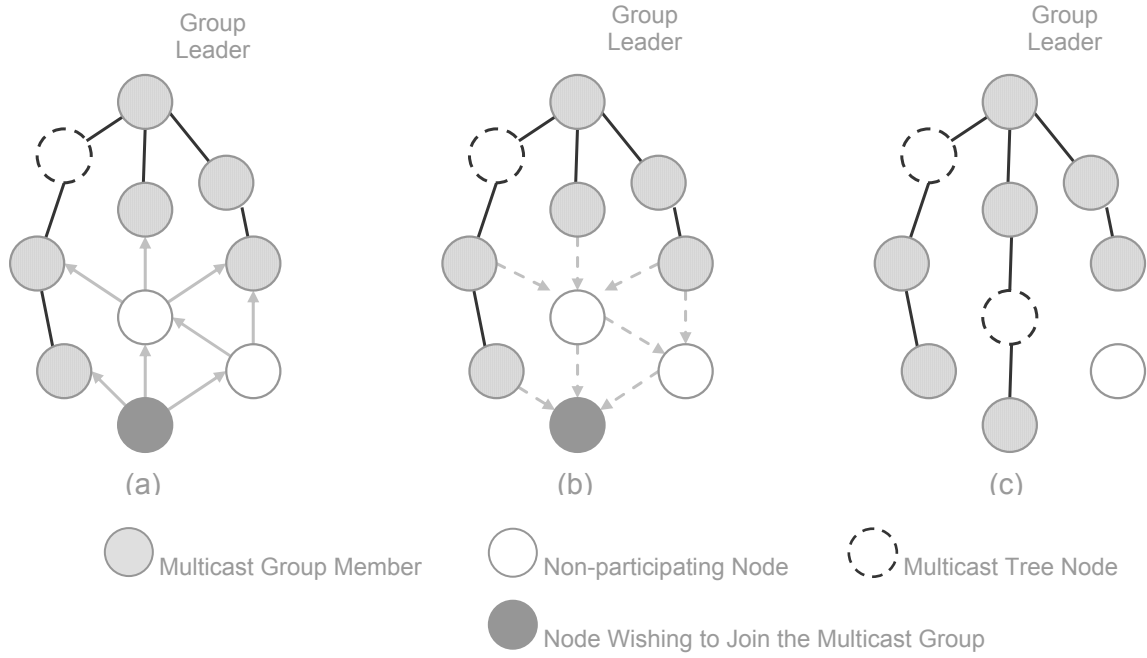


Figure 4.1. Join process in MAODV: (a) The node broadcasts a RREQ packet in order to find a route to multicast group. Non-participating nodes also broadcast the RREQ packet. (b) The multicast tree members send the RREP packets back to the node through the same path used for the RREQ packets. (c) The node selects the most recent and the shortest route to join the multicast tree.

4.2 Simulation Environment

We used *network simulator version 2* (ns-2) [54] as the simulation platform. It is a widely-used open-source discrete event simulator developed for networking research. The ad hoc networking extensions of ns-2 were developed by the MONARCH research group in CMU [55].

We implemented EraMobile and flooding on ns-2 and used *tcl* and *perl* scripts in order to run the protocol simulations. The MAODV implementation for ns-2 was taken from [56]. It is only publicly available MAODV implementation to the best of our knowledge.

We could not use it for the simulations in which the number of nodes is greater than 50 since it can simulate at most 50 nodes. It also can not simulate multiple multicast groups.

4.2.1 Radio Model

The radio model is used to predict the received signal power of each packet. N_s nodes have *receiving threshold* at physical layer and the packets with signal power lower than this threshold are marked as error and dropped by the MAC layer. In our simulations, we used *two-ray ground reflection* model as the radio propagation model. The two-ray ground reflection model considers both a direct path and a ground reflection path. The received power at distance d is estimated by [57]

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (4.1)$$

(G_t and G_r are the antenna gains of the transmitters and receivers respectively, L is the system loss, h_t and h_r are the heights of the transmit and receive antennas respectively, P_t is the transmitted signal power. G_t and $G_r = 1$ and $L=1$ in *ns* simulations)

In the short range, the *two-ray ground reflection* model does not perform well because of the oscillation caused by the constructive and destructive combination of the two rays [57]. Thus, *ns-2* calculates a cross-over distance, dc as it is seen in equation 4.2, in order to change the equation according to the distance. The equation of the *two-ray ground reflection* model, equation 4.1, is used when $d > dc$, otherwise equation of the *free space* model, equation 4.3, is used.

$$d_c = (4\pi h_t h_r) / \lambda \quad \lambda \text{ is the wave length} \quad (4.2)$$

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (4.3)$$

Ns-2 uses omni-directional antenna model. The omni-directional antennas receive and transmit equally well in all directions. The antenna pattern in fig. 4.2 receives four signals equally well. The signals radiated in these directions have also the same strength.

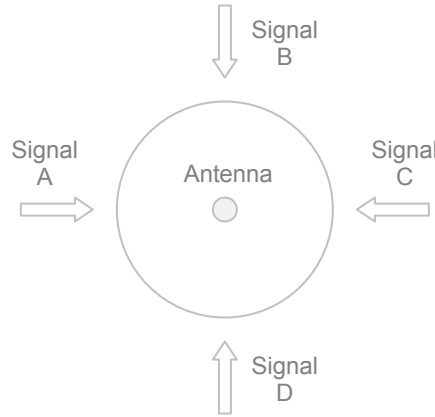


Figure 4.2. Omni-directional antenna

4.2.2 Medium Access Control Protocol

The IEEE 802.11 [36] was used as the medium access control protocol (MAC). The radio propagation range and the channel capacity offered by IEEE 802.11 are 250 meters and 2Mb/s respectively. The IEEE 802.11 uses carrier sense with multiple access and

collision avoidance (CSMA/CA) MAC layer mechanism to prevent the collisions which occur when two or more nodes attempt to send a signal along the same channel at the same time and generally result in garbled message. Each unicast packet transmission is preceded by a handshake operation between the sender and the receiver to ensure that the medium is free. This handshake operation is performed by request-to-send/clear-to-send (RTS/CTS) packets. The node wishing to send data first sends a RTS packet to the destination node for reserving the channel. The destination node, if it is available, replies with a CTS packet. Upon reception of the CTS packet, sender can start transmission process. Other nodes receiving the CTS packet, do not attempt to access the channel for some time announced in both RTS and CTS packets. Once the transmission of a data packet is completed, it is acknowledged by the receiver sending back an ACK packet to the sender. The RTS/CTS and ACK scheme can not be used for broadcast transmissions since there is not a unique receiver known by the sender for handshaking and acknowledgement. This fact substantially decreases the reliability of broadcast packets, especially, in dense ad hoc networks which are naturally open to frequent collisions.

4.2.3 Mobility Pattern

As the mobility pattern, we used the *random waypoint* model, which is the most commonly used model, for the simulations. In this model, each node randomly picks a destination and moves towards it with a velocity chosen uniformly randomly from the interval $[V_{min}, V_{max}]$, where V_{min} and V_{max} are the minimum and maximum speed values allowed. When a node arrives at the destination, it stops for some time defined by the *pause time* parameter. Then, it picks another random destination and repeats the process during simulation time.

In the simulations, we defined the difference between V_{min} and V_{max} as 0.2 m/sec in order to prevent any possible effect of the variation at node speeds. The *pause time* was

assigned to zero to create more rigorous mobile environment. The size of the network area where the mobile nodes travel is 1000m x 1000m. The mobility scenarios were generated by the *setdest* tool included in *ns-2*.

4.2.4 Traffic Pattern

The constant bit rate traffic is generated in the simulations with the payload data size of 512 bytes. The traffic generator attached the source generates data messages with a rate of 2 messages per second. Each simulation is run for 1000 seconds. Data generation starts 20 seconds after a simulation begins to execute, and continues for 870 seconds generating total of 1740 data messages. The remaining 110 seconds are used to deliver the messages which are still in flight or in the data buffers and waiting for FIFO order delivery.

The remaining 110 seconds are used to deliver the packets which are still in flight or in the data buffers and waiting for FIFO order delivery.

Chapter 5

PARAMETER ANALYSIS

The protocol parameters have decisive influence on the performance of the protocol. The inclusive analysis of the protocol parameters is an important requirement for obtaining the maximum performance. It can also be helpful for understanding the protocol response to different conditions and hence making possible improvements on the protocol.

This section presents the analysis and discussion of the main protocol parameters, gossip interval and stability threshold, through extensive simulations on *ns-2*. The effect of the parameters has been studied in low, moderate and high node densities separately. The adaptivity mechanism and optimizations are excluded in the simulations in order to view the effect of the parameters clearly. Thus, the results of the simulations in this section do not reflect the real performance of EraMobile. Instead, they basically demonstrate the effect of the protocol parameters. Throughput, data packet delivery ratio, and overhead metrics were used for the performance evaluation. Overhead was calculated as the ratio of total control and the redundant data bytes transmitted to the original data bytes delivered.

5.1 Low Node Density

In the low density network simulations, 10 hosts were simulated in the area of 1000m x 1000m with the mobility of 5 m/s. The average number of neighbors observed around a node was nearly 2. The area of 1000m x 1000m is normally too large for 10 nodes to have all members of the network reachable, especially in low mobility. In such a case, nodes

should keep the data for a reasonable long time to be able to answer the request messages coming from the nodes that missed some messages. Setting the stability threshold to higher values is a direct way to increase the number of messages kept by the nodes. Bigger stability threshold value enables the nodes to keep more data in their buffers so that they can recover more missing messages answering the coming requests. This is the reason of increase in the throughput with the stability threshold as shown in fig. 5.1 and table 5.1. However, bigger stability threshold values cause larger gossip messages, which mean larger overhead, since gossip messages consist of message ids in the data buffers of the nodes. Thus, the stability threshold value should be set taking this trade-off into consideration.

Extending the gossip interval is another way to keep more data in the buffer. It may also assist to increase throughput if the stability threshold value is too small to keep enough number of messages in the buffer, e.g. throughput rises while gossip interval changes from 1s to 3s for the stability threshold value of 30 as seen in fig. 5.1. Because, larger gossip intervals extend the lifetime of the messages in the data buffers since their gossip counts, which are increased in each gossip round, reach the stability threshold in longer period of time to be removed from the buffer by garbage collection mechanism. In bigger stability threshold values, 100 and 120 for the case in fig. 5.1, nodes keep sufficient amount of data to answer the requests for missing messages. Thus, extending gossip intervals can not contribute to throughput. In longer gossip interval values, nodes send less number of gossip messages in total. This decline in number of gossip messages brings the reduction in request and data traffic. Considering the limits for request and transmission packets, less number of transmission messages results in lower throughput.

As shown in fig. 5.2 and table 5.2, the overhead is proportional to the stability threshold. In higher stability threshold values, data messages are kept in the buffers for longer times so that the number of messages in the buffers is increased. The larger data

buffers cause the gossip messages to become bigger-sized, thereby, increasing the overhead in the network. Long gossip intervals also increase the size of gossip messages. However, overhead is reduced in long gossip intervals since the longer gossip intervals cause the less number of gossip messages.

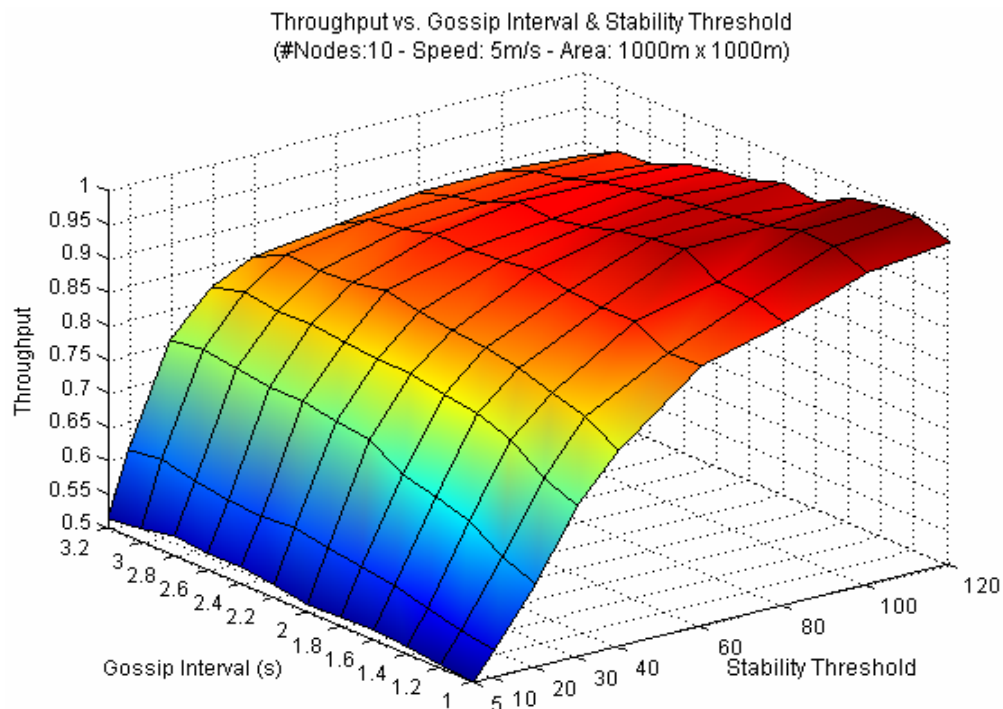


Figure 5.1. The effect of the protocol parameters on the throughput in low node density

GI \ ST	5	10	20	30	40	60	80	100	120
1	0,5	0,54	0,622	0,723	0,789	0,884	0,917	0,961	0,975
1,2	0,505	0,548	0,642	0,743	0,819	0,882	0,932	0,979	0,992
1,4	0,511	0,56	0,674	0,779	0,833	0,908	0,936	0,981	0,987
1,6	0,515	0,573	0,689	0,797	0,849	0,908	0,964	0,967	0,978
1,8	0,513	0,58	0,701	0,804	0,858	0,912	0,958	0,961	0,951
2	0,511	0,587	0,732	0,814	0,863	0,921	0,949	0,954	0,959
2,2	0,519	0,596	0,738	0,815	0,87	0,915	0,935	0,947	0,942
2,4	0,525	0,592	0,747	0,819	0,88	0,911	0,925	0,938	0,936
2,6	0,524	0,595	0,747	0,825	0,871	0,912	0,93	0,928	0,925
2,8	0,529	0,603	0,754	0,823	0,868	0,899	0,907	0,912	0,904
3	0,519	0,614	0,761	0,831	0,871	0,892	0,896	0,902	0,901
3,2	0,513	0,607	0,754	0,817	0,846	0,863	0,881	0,883	0,875

Table 5.1. The throughput vs. gossip interval (GI) and stability threshold (ST) in low node density

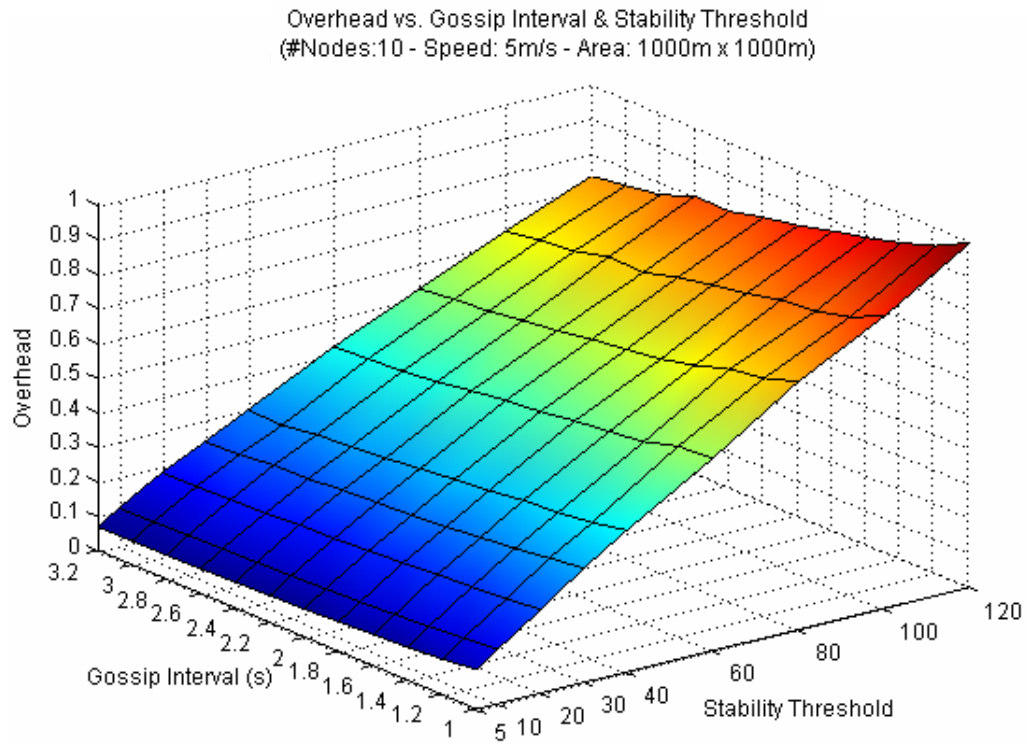


Figure 5.2. The effect of the protocol parameters on the overhead in low node density

GI \ ST	5	10	20	30	40	60	80	100	120
1	0,114	0,157	0,241	0,33	0,408	0,554	0,715	0,842	0,993
1,2	0,102	0,147	0,23	0,317	0,389	0,548	0,683	0,798	0,942
1,4	0,095	0,14	0,223	0,303	0,376	0,519	0,667	0,773	0,909
1,6	0,09	0,132	0,217	0,293	0,37	0,507	0,637	0,764	0,882
1,8	0,085	0,127	0,209	0,282	0,356	0,496	0,613	0,741	0,86
2	0,08	0,123	0,202	0,274	0,347	0,481	0,599	0,724	0,834
2,2	0,077	0,119	0,196	0,269	0,336	0,472	0,587	0,704	0,816
2,4	0,074	0,116	0,194	0,263	0,325	0,458	0,57	0,677	0,792
2,6	0,071	0,114	0,185	0,254	0,316	0,442	0,558	0,68	0,794
2,8	0,069	0,111	0,183	0,252	0,309	0,431	0,543	0,655	0,756
3	0,068	0,108	0,179	0,241	0,295	0,425	0,536	0,645	0,741
3,2	0,067	0,106	0,178	0,236	0,297	0,418	0,524	0,629	0,727

Table 5.2. The overhead vs. gossip interval (GI) and stability threshold (ST) in low node density

5.2 Moderate Node Density

In the moderate density network simulations, 60 hosts were simulated in the area of 1000m x 1000m with the mobility of 5 m/s. The average number of neighbors observed around a node was nearly 14. The node density obtained in the simulations of 60 nodes in the network area of 1000m x 1000m was sufficient to provide good network connectivity. Thus, the throughput was observed in the range of 98% and 100% as seen in fig. 5.3 and table 5.3. In this case, the influence of the protocol parameters on the throughput was minimized. The correct values for the parameters should be determined considering the overhead performances.

The overhead increases with the stability threshold similar to the former simulation study. However, as shown in fig. 5.4 and table 5.4, the amount of overhead is greater than the overhead obtained in the low network density simulations because of the higher node density. The number of gossip broadcasts received by a node increases with the number of neighboring nodes. Since the gossip broadcasts are the main factor behind the overhead, higher node density brings larger overhead to the network.

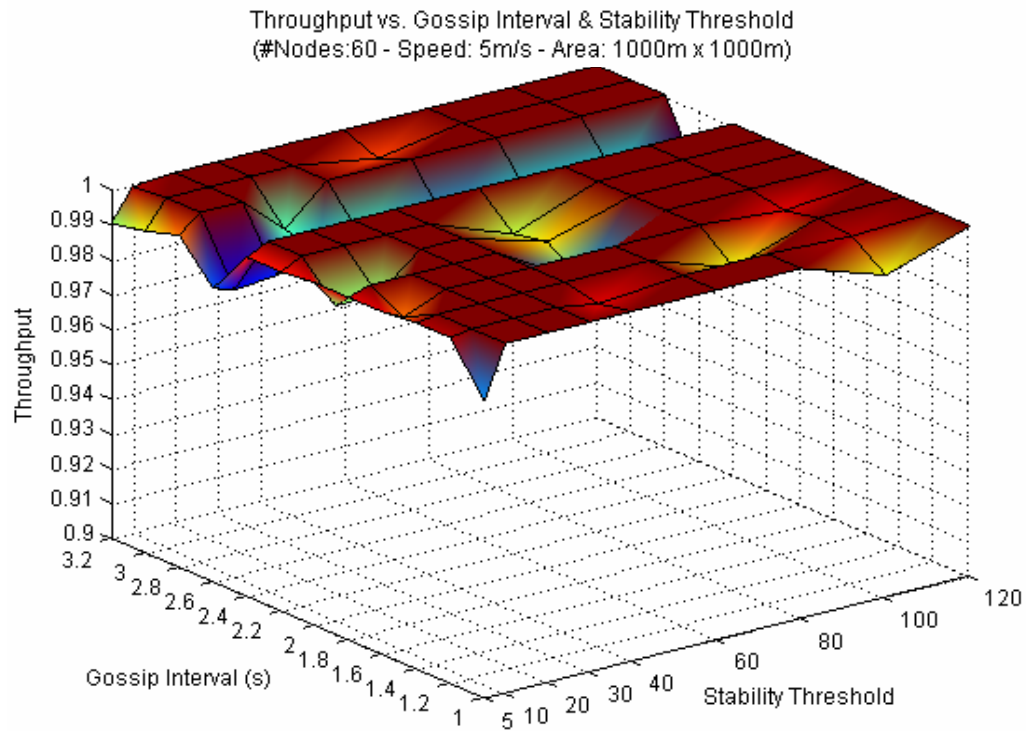


Figure 5.3. The effect of the protocol parameters on the throughput in moderate node density

GI \ ST	5	10	20	30	40	60	80	100	120
1	0,985	1	1	1	1	1	1	0,992	1
1,2	0,999	1	1	1	0,998	1	1	0,999	1
1,4	0,999	1	1	1	1	1	0,992	1	1
1,6	0,998	0,995	1	1	1	1	1	0,998	1
1,8	0,998	1	1	1	1	0,986	1	1	1
2	0,998	0,99	0,99	1	1	0,993	1	1	1
2,2	0,997	1	1	1	1	0,991	1	1	1
2,4	0,997	1	1	1	1	1	1	1	1
2,6	0,984	0,982	0,983	0,987	0,988	0,986	0,987	0,987	0,98
2,8	0,995	1	1	0,989	1	1	1	1	1
3	0,992	1	1	1	1	0,996	1	1	1
3,2	0,99	0,999	1	1	1	1	1	1	1

Table 5.3. The throughput vs. gossip interval (GI) and stability threshold (ST) in moderate node density

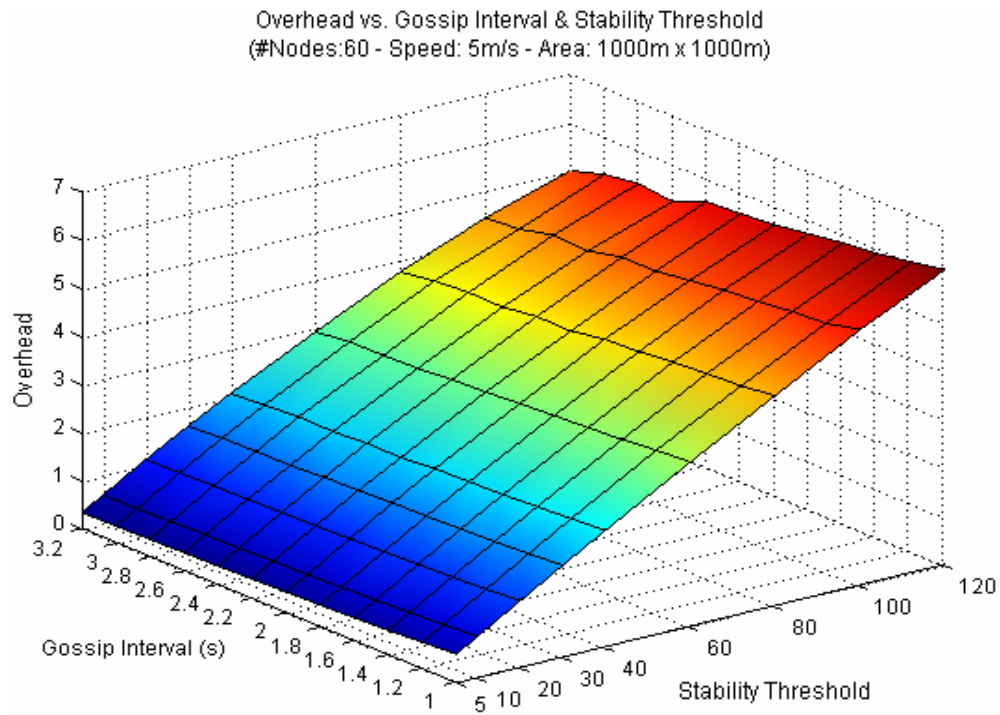


Figure 5.4. The effect of the protocol parameters on the overhead in moderate node density

GI \ ST	5	10	20	30	40	60	80	100	120
1	0,595	0,86	1,383	1,902	2,41	3,4	4,352	5,295	6,111
1,2	0,548	0,807	1,331	1,851	2,364	3,356	4,26	5,132	5,977
1,4	0,504	0,77	1,293	1,811	2,323	3,286	4,227	5,088	5,865
1,6	0,473	0,733	1,263	1,781	2,282	3,242	4,173	5,001	5,788
1,8	0,446	0,712	1,238	1,759	2,258	3,202	4,098	4,95	5,688
2	0,426	0,679	1,21	1,742	2,238	3,158	4,067	4,856	5,581
2,2	0,403	0,67	1,187	1,71	2,176	3,107	3,942	4,758	5,52
2,4	0,38	0,646	1,173	1,688	2,179	3,072	3,953	4,755	5,478
2,6	0,366	0,636	1,153	1,642	2,133	3,013	3,859	4,595	5,185
2,8	0,353	0,616	1,143	1,643	2,145	2,99	3,869	4,63	5,27
3	0,336	0,589	1,122	1,627	2,097	2,983	3,819	4,463	5,169
3,2	0,324	0,583	1,101	1,582	2,044	2,894	3,714	4,406	4,955

Table 5.4. The overhead vs. gossip interval (GI) and stability threshold (ST) in moderate node density

5.3 High Node Density

In the high node density network simulations, 120 hosts were simulated in the area of 1000m x 1000m with the mobility of 5 m/s. The average number of neighbors observed around a node was nearly 28. The node density may turn out a prohibitive factor on the throughput after some point in spite of perfect network connectivity in dense ad hoc networks. The increasing density means increasing number of neighbors and higher network traffic load for the nodes. Such high traffic may congest the network and cause the packet losses to occur more frequently. The decrease in throughput compared to the previous simulation results supports this assumption as depicted in fig. 5.5 and table 5.5. The influence of the gossip interval on the throughput becomes more apparent in the dense network simulations. The smaller gossip intervals cause the nodes to broadcast higher number of gossip messages, thereby, raising the probability of congestions. The increase in congestions decreases the throughput as it can be seen in the figure. However, the throughput tends to fall again in large gossip intervals like 4s since the number of gossip messages in such large intervals is not adequate to provide full delivery of the data. The stability threshold contributes the throughput in a positive way enabling the nodes to recover more missing messages. The throughput declines, though, in the bigger stability threshold values such as 120 because it increases the size of gossip messages and so occupies the wireless medium longer.

Fig. 5.6 and table 5.6 show the packet loss ratio experienced by the nodes due to congestions. Considering the average loss ratio is under 1% in moderate node density, the packet loss ratios, which are over 20% in the gossip interval of 1s, illustrate the effect of the congestions. The packet loss ratio rises in small gossip intervals and large stability threshold values due to the highly congested wireless medium.

The overhead performance of the protocol was also shaped by the congestions. The stability threshold increased the overhead as expected. However, the overhead also

increased with the gossip interval as shown in fig. 5.7 and table 5.7 contrary to the former simulation results. Additionally, the amount of overhead was less than the results of moderate node density. The reason of those unexpected results is that congestions decrease the reception of any type of packets but especially gossip broadcasts. This situation cause a so-called improvement in the overhead performance of the nodes exposed to congestions since the overhead is calculated as the ratio of received bytes which do not belong to original data to delivered bytes of the original data.

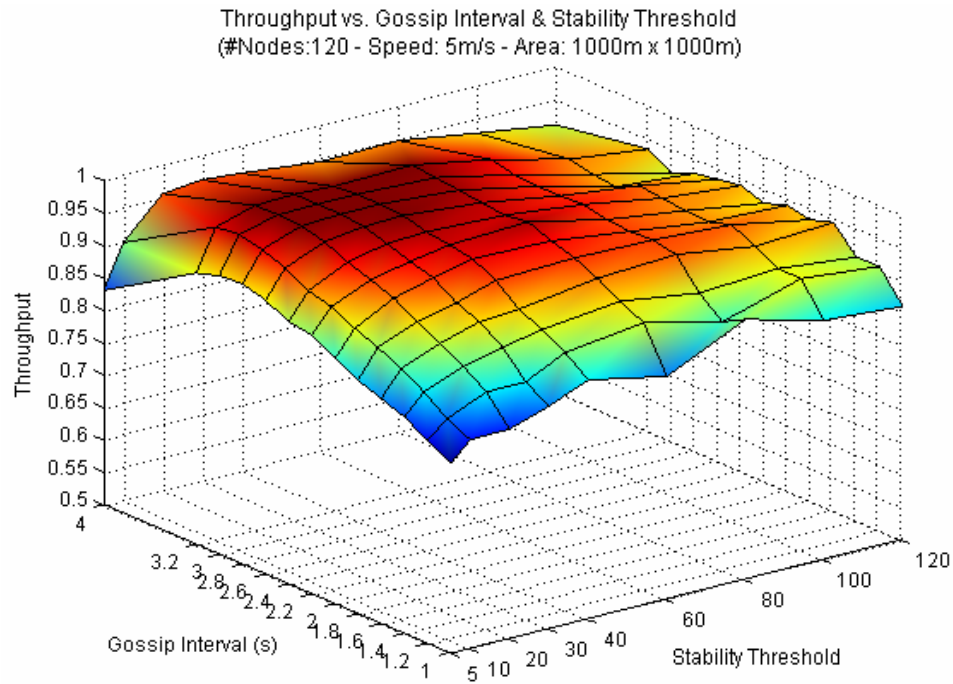


Figure 5.5. The effect of the protocol parameters on the throughput in high node density

GI \ ST	5	10	20	30	40	60	80	100	120
1	0,792	0,821	0,822	0,842	0,867	0,842	0,899	0,867	0,858
1,2	0,812	0,838	0,864	0,864	0,894	0,909	0,88	0,916	0,904
1,4	0,833	0,864	0,888	0,902	0,914	0,927	0,909	0,916	0,903
1,6	0,846	0,884	0,908	0,926	0,93	0,947	0,953	0,935	0,941
1,8	0,859	0,902	0,926	0,94	0,959	0,963	0,96	0,953	0,93
2	0,877	0,916	0,947	0,963	0,977	0,974	0,958	0,942	0,937
2,2	0,898	0,934	0,96	0,973	0,983	0,988	0,961	0,944	0,925
2,4	0,903	0,943	0,972	0,985	0,988	0,983	0,969	0,948	0,937
2,6	0,921	0,963	0,988	0,996	0,998	0,987	0,967	0,952	0,931
2,8	0,931	0,976	0,997	1	1	0,991	0,966	0,935	0,925
3	0,928	0,984	0,999	1	1	0,995	0,969	0,939	0,91
3,2	0,917	0,98	0,998	1	0,997	0,999	0,977	0,947	0,932
4	0,829	0,895	0,955	0,961	0,956	0,943	0,945	0,926	0,908

Table 5.5. The throughput vs. gossip interval (GI) and stability threshold (ST) in high node density

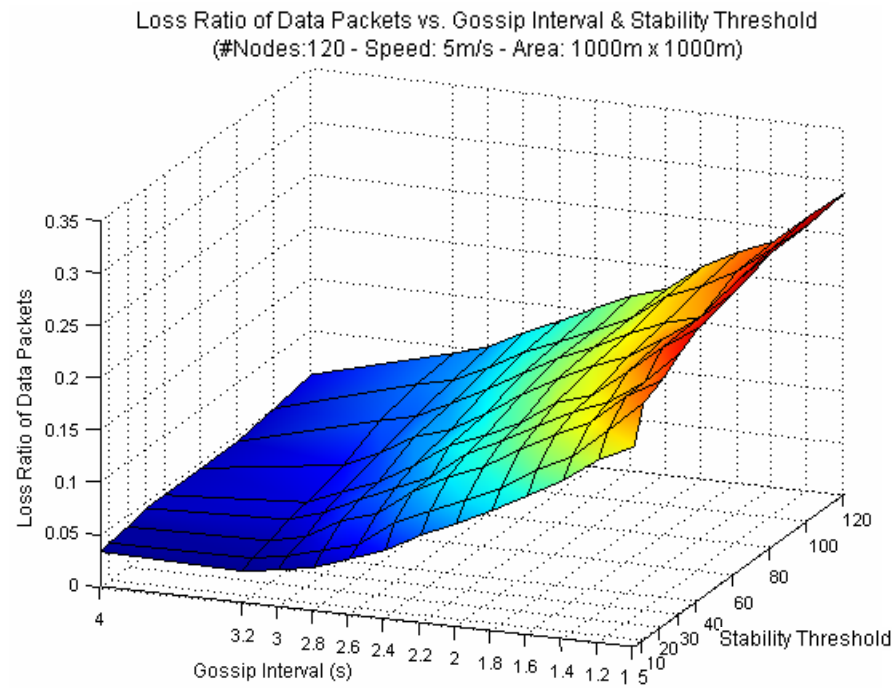


Figure 5.6. The effect of the protocol parameters on the losses of the data packets in high node density

GI \ ST	5	10	20	30	40	60	80	100	120
1	0,191	0,227	0,235	0,249	0,260	0,269	0,281	0,282	0,287
1,2	0,175	0,196	0,216	0,217	0,238	0,240	0,249	0,256	0,261
1,4	0,151	0,168	0,188	0,194	0,210	0,217	0,212	0,230	0,233
1,6	0,133	0,151	0,172	0,181	0,184	0,196	0,202	0,212	0,219
1,8	0,115	0,130	0,154	0,164	0,173	0,183	0,188	0,198	0,202
2	0,098	0,114	0,134	0,146	0,150	0,166	0,171	0,173	0,176
2,2	0,084	0,100	0,120	0,129	0,134	0,148	0,157	0,158	0,165
2,4	0,065	0,085	0,102	0,103	0,121	0,126	0,139	0,144	0,150
2,6	0,053	0,061	0,077	0,089	0,098	0,106	0,120	0,129	0,134
2,8	0,041	0,051	0,067	0,074	0,078	0,096	0,102	0,111	0,119
3	0,035	0,042	0,048	0,058	0,065	0,076	0,086	0,095	0,102
3,2	0,030	0,036	0,040	0,047	0,051	0,061	0,078	0,088	0,092
4	0,033	0,035	0,038	0,043	0,043	0,043	0,043	0,049	0,056

Table 5.6. The data packet loss ratio vs. gossip interval (GI) and stability threshold (ST) in high node density

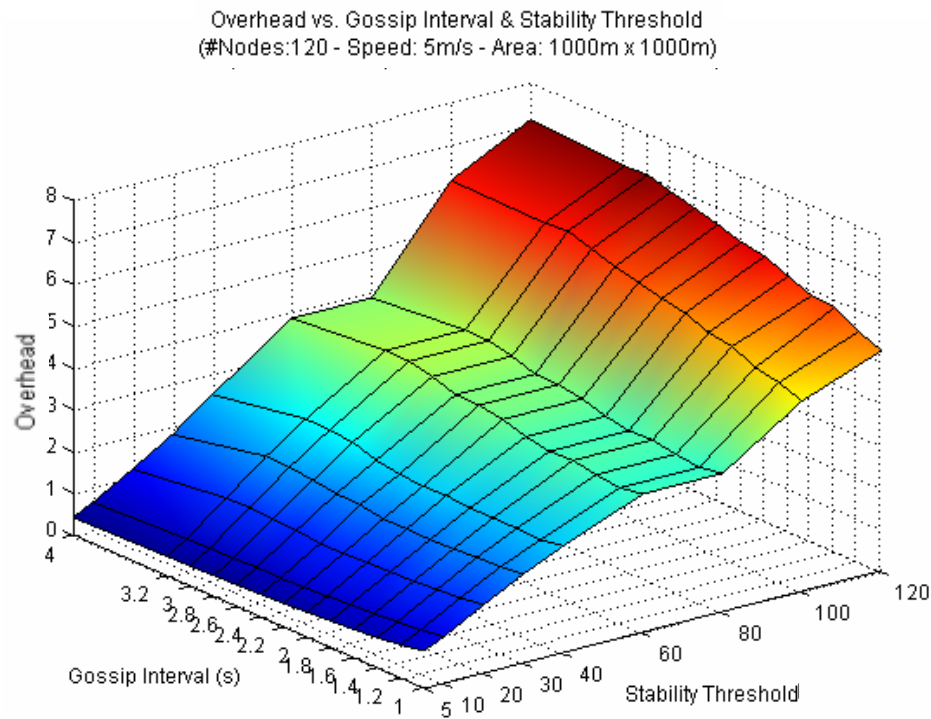


Figure 5.7. The effect of the protocol parameters on the overhead in high node density

GI \ ST	5	10	20	30	40	60	80	100	120
1	0,898	1,114	1,63	2,12	2,543	3,292	3,292	4,531	5,244
1,2	0,792	1,031	1,549	2,043	2,436	3,191	3,191	4,726	5,447
1,4	0,723	0,978	1,5	2,025	2,469	3,262	3,262	4,841	5,786
1,6	0,67	0,927	1,457	1,983	2,48	3,357	3,357	5,097	5,817
1,8	0,632	0,903	1,436	1,959	2,467	3,33	3,33	5,185	6,039
2	0,594	0,857	1,413	1,97	2,523	3,433	3,433	5,402	6,27
2,2	0,558	0,83	1,386	1,968	2,5	3,602	3,602	5,509	6,361
2,4	0,535	0,797	1,376	2,052	2,54	3,698	3,698	5,645	6,533
2,6	0,515	0,816	1,448	2,077	2,633	3,702	3,702	5,779	6,681
2,8	0,492	0,776	1,473	2,153	2,818	3,889	3,889	5,955	6,827
3	0,476	0,784	1,507	2,219	2,855	4,011	4,011	6,137	6,987
3,2	0,462	0,758	1,481	2,235	2,881	4,023	4,023	6,082	6,95
4	0,413	0,649	1,197	1,779	2,481	3,812	3,812	6,121	7,083

Table 5.7. The overhead vs. gossip interval (GI) and stability threshold (ST) in high node density

5.4 Result

The study of parameter analysis revealed the fact that the gossip interval and the stability threshold parameters have great influence on the performance of the protocol. This influence especially becomes more apparent in adverse network conditions such as poor network connectivity and congestion. The static adjustment of the parameters may significantly degrade the overall performance of the protocol against variable network conditions. For instance, in order to achieve 99% throughput in the low node density scenarios, the stability threshold should be 120 and the gossip interval should be 1.2s shown in table 5.1. If the same values are used in moderate node density, the overhead becomes 5.97 as given in table 5.4. However, it is also possible to achieve 99% throughput with the overhead of 0.548, ten times less than previous overhead value, setting the stability threshold to 5 and the gossip interval to 1.2s in moderate node density.

Consequently, the study of parameter analysis confirmed the need for an adaptivity mechanism in order to adjust the parameters dynamically according to variable network conditions of mobile ad hoc networks.

Chapter 6

PERFORMANCE EVALUATION

6.1 Performance Metrics

In order to investigate the packet delivery, reliability and overhead performance of the protocol, we used the following metrics: *throughput*, *reliable-throughput*, *receive-overhead* and *send-overhead*.

- *Throughput*, in other words packet delivery ratio, is calculated as the ratio of total number of data packets delivered by the receivers to the number of data packets supposed to be delivered. It indicates in what extent the protocol succeeds the delivery of the packets originated.
- *Reliable-throughput* is the fraction of the data packets delivered to **all** receivers over the total number of data packets supposed to be delivered. That is, a packet should be delivered to all members of a multicast group in order to be counted in *reliable-throughput*. It shows how successfully the protocol can deliver the data to all receivers.
- *Receive-overhead* is the ratio of bytes belonging to control packets and duplicate data packets received by a node to the bytes of data it delivered. That is, all the bytes received, except the bytes of data packets received for the first time, are counted in the *receive-overhead* portion of the protocol. *Receive-overhead* illustrates the network load that a node exposes to deliver one byte of data. In lossy networks, *receive-overhead* may be misleading since the nodes in such networks can hardly receive the packets. This

situation triggers a drop in the network load noticed by the nodes and so causes a *receive-overhead* lower than it actually is.

- *Send-overhead* is the ratio of total bytes sent by a node to the bytes of data it delivered. *Send-overhead* shows the network load generated to transmit one byte of data. However, it may be inadequate to show the actual network load in broadcast transmission since multiple receivers may receive the packet. Thus, it should be taken into account together with *receive-overhead*.

6.2 Results

In the simulations, we examine the effect of mobility, group size, traffic load and number of multicast groups and senders on the performance of the protocols. Each simulation is repeated 5 times with different seed numbers and node movement scenarios, which are different than the scenarios used in parameter analysis, and the collected data are averaged over those runs. The values presented in table 6.1 are used for the protocol parameters. The adaptivity unit chooses the proper parameter values based on the node density observed. In order to enable data delivery in extremely dense networks, in which number of neighbors around a node is greater than 30, the *gossip interval* is gradually increased, i.e, 0.03 s. for each increment in the node density.

Parameters	Low Density Level	Normal Density Level	High Density Level
<i>Gossip Interval</i>	1.2 sec	1.8 sec	2.4 sec
<i>Request & Data Transmission Limits for a Round</i>	28	16	4
<i>Stability Threshold</i>	180	150	120
<i>Request Sending Probability</i>	1	0.7	0.4
<i>Gossip Interval Addition</i>	0.05 sec	0.1 sec	0.15 sec
<i>Gossip Interval Upper Limit</i>	4 sec	8 sec	12 sec

Table 6.1 The values of the protocol parameters for different node density levels

6.2.1 Effect of Mobility

The average speed of nodes was varied from 1 m/s (3.6 km/hr) to 30 m/s (108 km/hr) in steps of 5 m/s. The mobility variation was investigated in low, moderate and high node densities separately to show, if there is any, the correlational effect of mobility and node density on the performance of the protocols. The network modeled consists of 10, 50 and 100 mobile nodes in a single group for low, moderate and high node density scenarios respectively.

a-) Effect of Mobility in Low Node Density

In low density, some nodes may move to isolated areas, which are not covered by any node, and become unreachable. Throughout the unreachable state, they miss the data packets disseminated in this period of time. Flooding and MAODV nodes that missed data dissemination waves can not recover the missing data. However, in EraMobile, nodes have the recovery chance for the missing data packet through consecutive gossip rounds. A node, which has been unreachable for a while, may compensate the data it missed upon getting rid of isolation. The success probability of the recovery is dependent on the time passed being unreachable. Mobility acts as a positive factor by shortening this time. At high speeds, nodes can leave the isolated areas more quickly and come into the coverage of the nodes up-to-date before they remove the data messages from the buffers. Besides, higher mobility means higher chance for a node to come across different nodes and to recover missing messages. The throughput performance shown in fig. 6.1 and 6.2 verifies this mobility-friendly characteristic of our protocol. EraMobile achieves 100% reliable data delivery with increasing mobility. The reason of relatively lower throughput performance in the beginning as shown in fig. 6.1 and 6.2, i.e. 96% *throughput* and 78% *reliable-throughput* at the speed of 1m/s, is the isolated nodes that recently became unreachable and could not get rid of isolation till end of the simulation. The packet delivery ratio of MAODV and flooding can not exceed 60% as shown in fig 6.1. While the throughput of

flooding does not significantly vary with mobility, MAODV throughput decreases from 60% to 40% with increasing mobility. The reason of the drop in the throughput of MAODV is that mobility causes link breakages on its fragile multicast tree and data can not be delivered to the nodes beyond broken links till they are repaired. The reliable throughput performance of both flooding and MAODV goes around 10% as shown in fig. 6.2 due to the isolated nodes. In low node density, it is very likely that a node becomes unreachable while others keep communication over multiple hops. In such situations, a data packet can not be delivered to all the nodes of the group and reliable throughput performance is degraded. The overhead of all protocols does not change considerably with mobility as shown in fig. 6.3 and 6.4. The *receive-overhead* of flooding is higher than both MAODV and EraMobile due to large number of duplicate data packets received through broadcasts. Its *send-overhead* is always 1 for single multicast group because nodes broadcast all the data packets they deliver, that is, the number of transmitted bytes is equal to the number of delivered bytes. The *receive-overhead* of MAODV exhibits a slight decrease as seen in fig. 6.3. Indeed, there is a huge increase in control packets of MAODV with higher mobility since increasing mobility triggers more frequent updates on the multicast tree structure of MAODV. However, *receive-overhead* does not increase because of the drop in the number of duplicate packets received. The *send-overhead* of MAODV remains almost constant irrespective of mobility due to the same reason. EraMobile lowers the *receive-overhead* substantially by eliminating redundant data transmissions thanks to its gossip mechanism. The only factors increasing the receive-overhead of the protocol are gossip and request messages. However, the number and the size of those packets are highly reasonable, i.e. 239 request messages with size of 30.4 and 1475 gossip messages with size of 25.3 were received in average during the simulations at the speed of 30m/s. Both *receive-overhead* and *send-overhead* of EraMobile are also not affected by mobility since it does not maintain any structure for multicast data delivery.

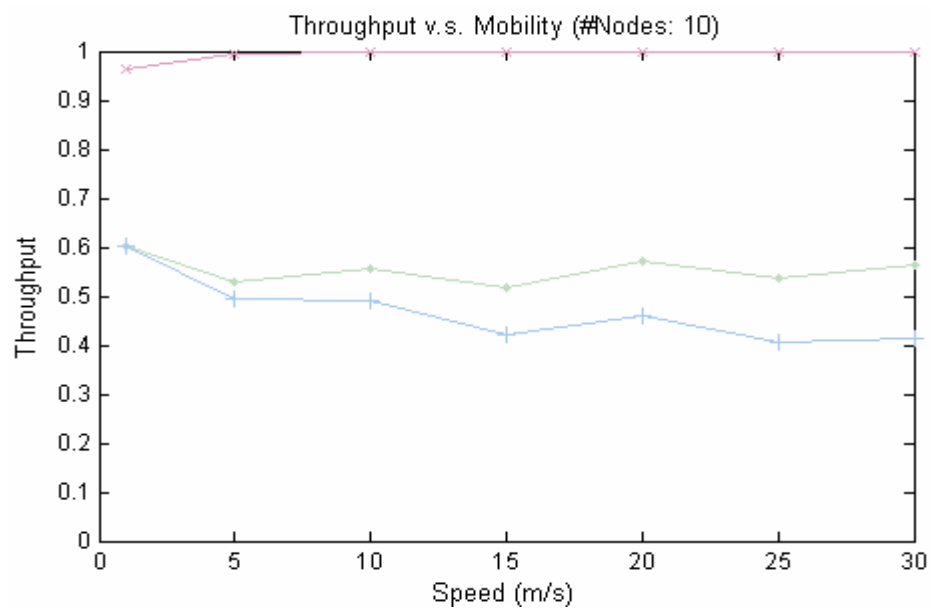


Figure 6.1. The effect of mobility on the throughput of the protocols in low node density

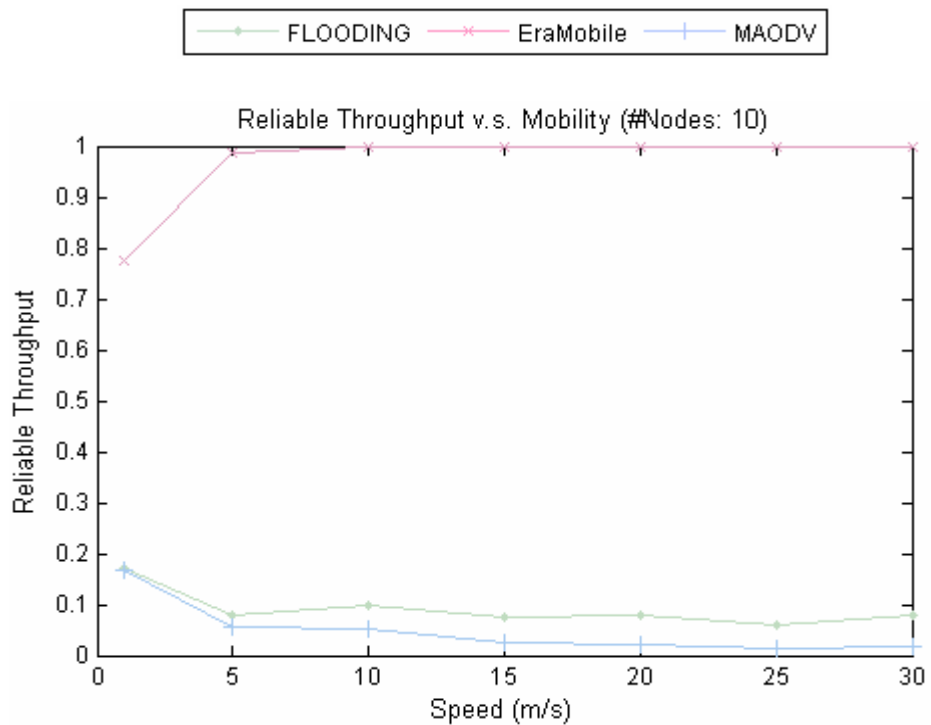


Figure 6.2. The effect of mobility on the reliable-throughput of the protocols in low node density

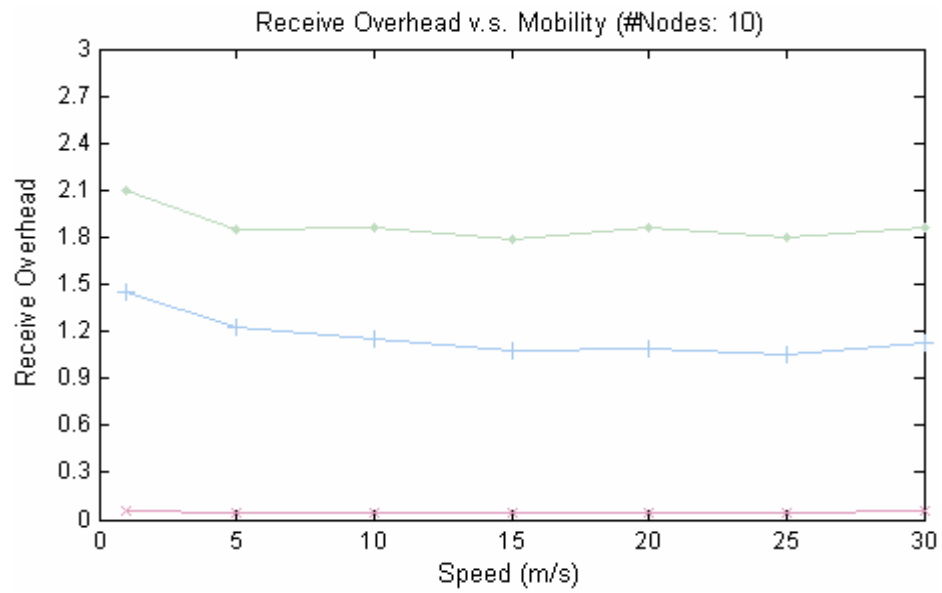


Figure 6.3. The effect of mobility on the receive-overhead of the protocols in low node density

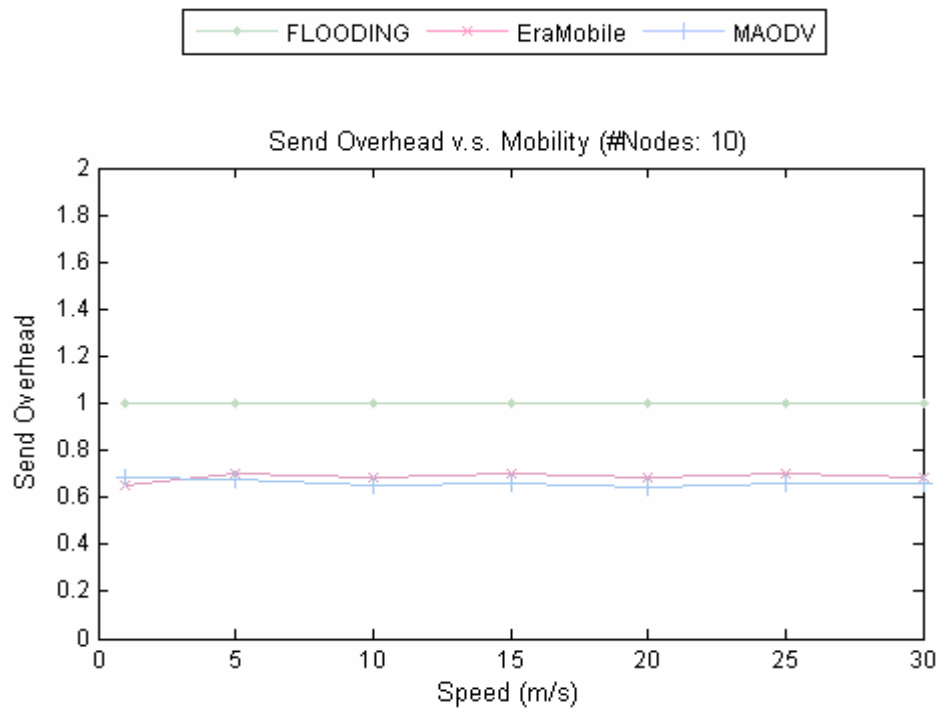


Figure 6.4. The effect of mobility on the send-overhead of the protocols in low node density

b-) Effect of Mobility in Moderate Node Density

In moderate node density simulations, protocols have sufficient number of nodes to obtain the full network connectivity over multiple hops. The effect of mobility can be observed independent of node density since the number of unreachable nodes is minimized in moderate node density. The suitable network conditions enable all the protocol to reach excellent throughput values. EraMobile performs 100% reliable packet delivery without being affected by mobility as shown in fig. 6.5 and 6.6. Flooding also exhibits close performance to EraMobile. However, *reliable-throughput* of flooding goes around 80% since even small number of packets missed due to link errors or collisions may significantly degrade the reliability performance in absence of a recovery mechanism. The only method to achieve 100% *reliable-throughput* is to have 100% data delivery ratio. Otherwise, even in the case of reaching 99% packet delivery ratio, the *reliable-throughput* may remain zero if each node missed a different data packet. This situation can be observed on the throughput performance of MAODV in fig. 6.5 and 6.6. MAODV achieves packet delivery ratio over 95% for the mobility of 1m/s. However, its *throughput* decreases with mobility due to its mobility-fragile tree structure. The situation is more dramatic for *reliable-throughput* performance of MAODV. While its *throughput* decreases linearly with mobility, its *reliable-throughput* declines exponentially due to the reason explained above. The link breakages make it almost impossible for MAODV to achieve reliable data delivery over 10% at higher mobility values. The overhead characteristic of the protocol does not change in moderate node density as seen in fig. 6.7 and 6.8. The only difference is the increase on the *receive-overhead* values of all protocols because the broadcast packets are received by more nodes when the number of neighbors increases. This situation can best be seen on the *receive-overhead* of flooding. Since the broadcast transmission is most used by flooding, the increasing node density affects its *receive-overhead* most by inflating

the duplicate data packets received. The *receive-overhead* of EraMobile also increases due to the higher number of nodes receiving gossip broadcasts. However, the amount of this increase is reasonable thanks to small size of gossip messages.

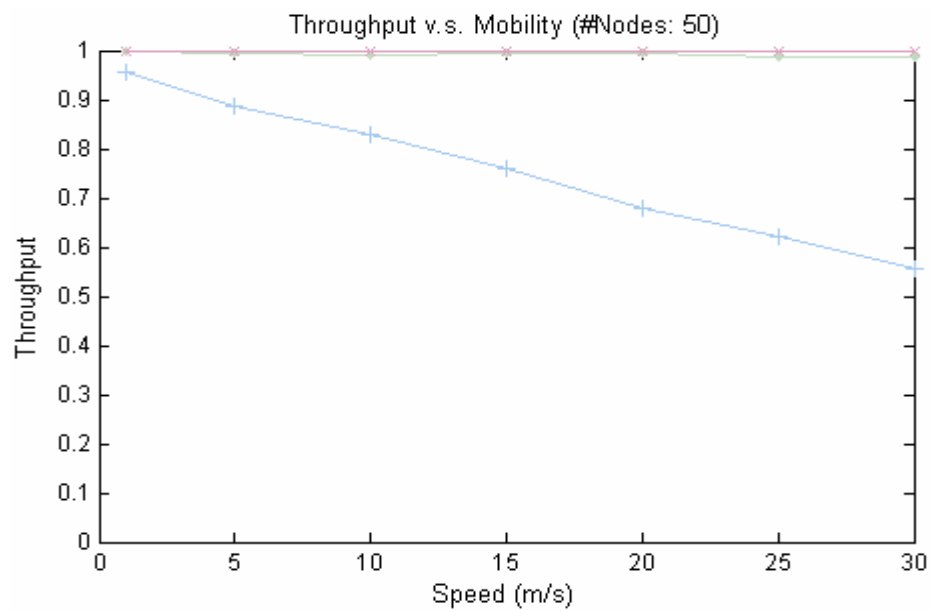


Figure 6.5. The effect of mobility on the throughput of the protocols in moderate node density



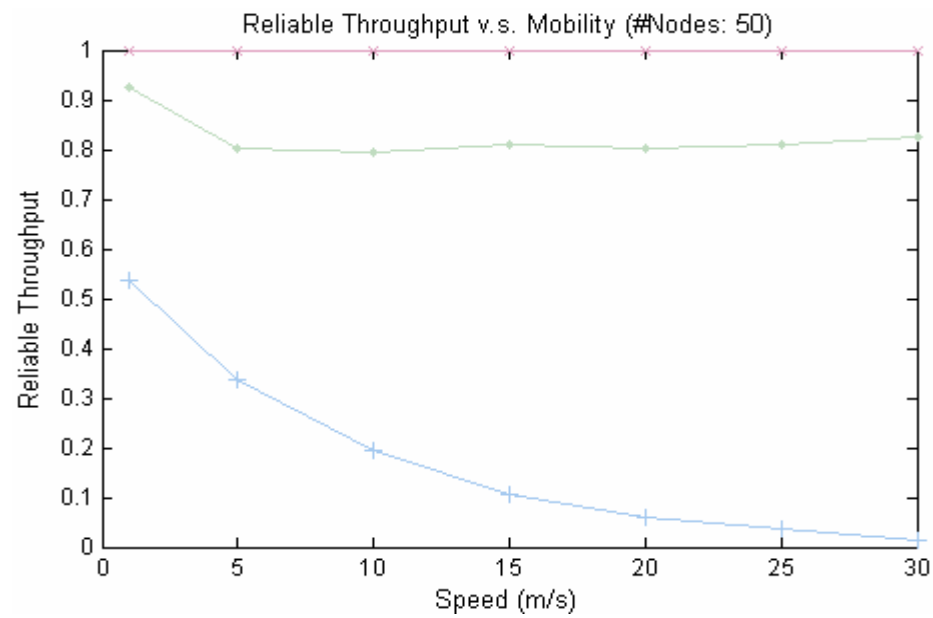


Figure 6.6. The effect of mobility on the reliable-throughput of the protocols in moderate node density

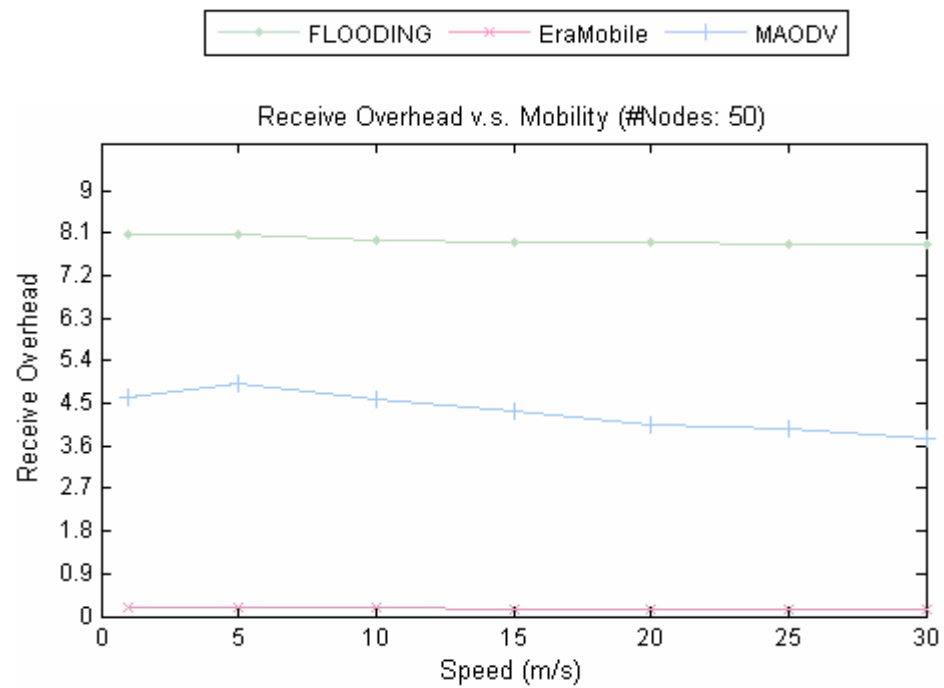


Figure 6.7. The effect of mobility on the receive-overhead of the protocols in moderate node density

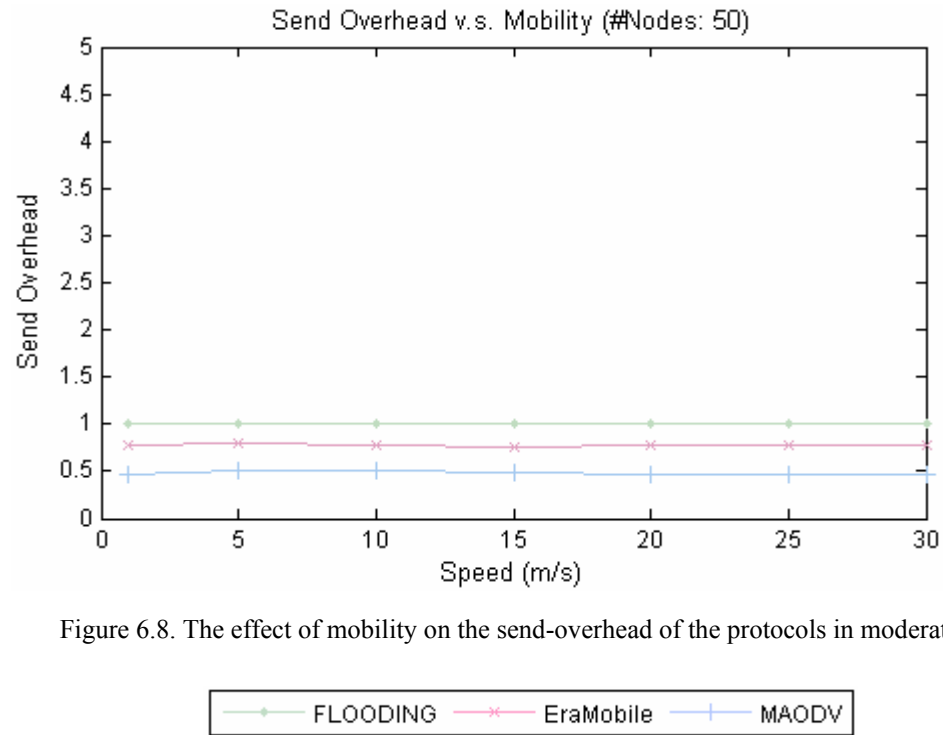


Figure 6.8. The effect of mobility on the send-overhead of the protocols in moderate node density

c-) Effect of Mobility in High Node Density

EraMobile achieves full reliable data delivery for almost all mobility values in high node density as shown in fig. 6.9 and 6.10. Flooding also exhibits very similar performance to EraMobile with one difference: its *reliable-throughput* varies between 99% and 96% according to mobility value. The *receive-overhead* of both protocols increased with higher node density, as shown in fig. 6.11, due to the reason explained previously. The *send-overhead* of both protocols remains without change as seen in fig. 6.12.

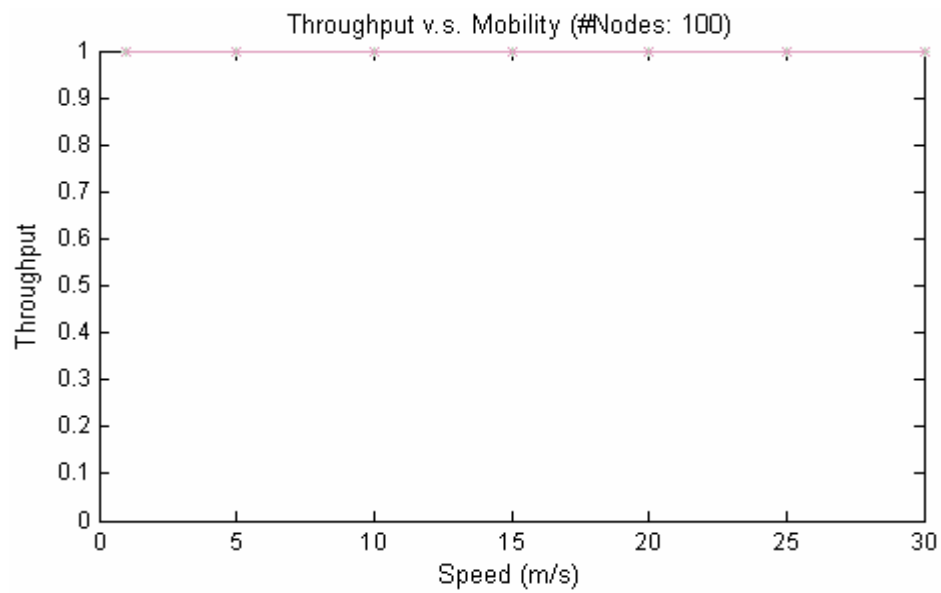


Figure 6.9. The effect of mobility on the throughput of the protocols in high node density

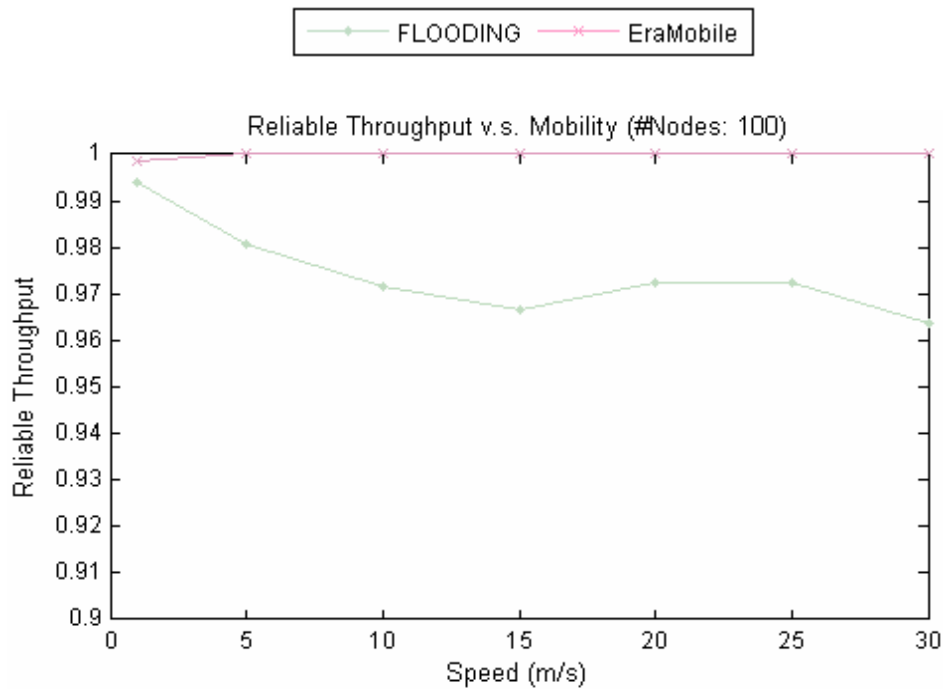


Figure 6.10. The effect of mobility on the reliable-throughput of the protocols in high node density

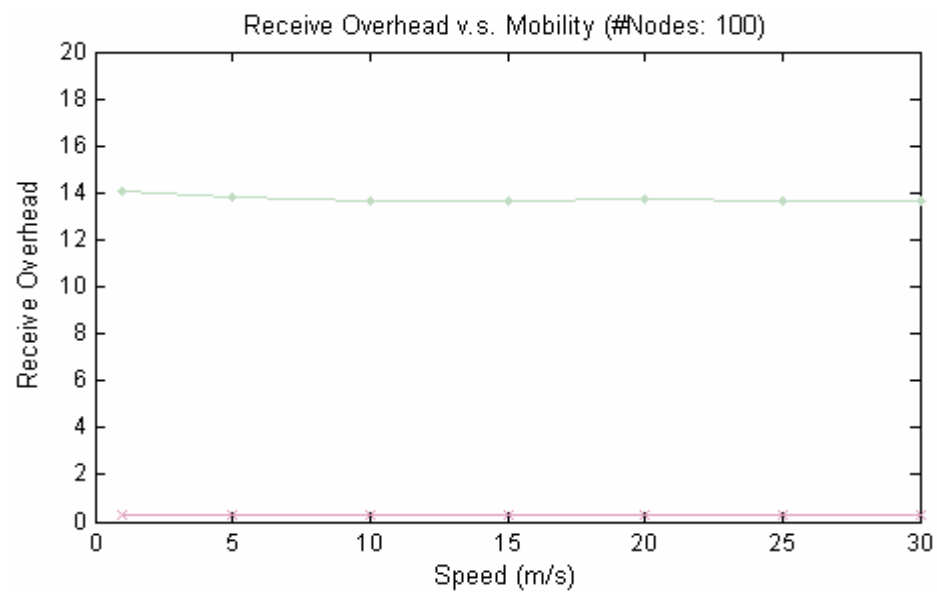


Figure 6.11. The effect of mobility on the receive-overhead of the protocols in high node density

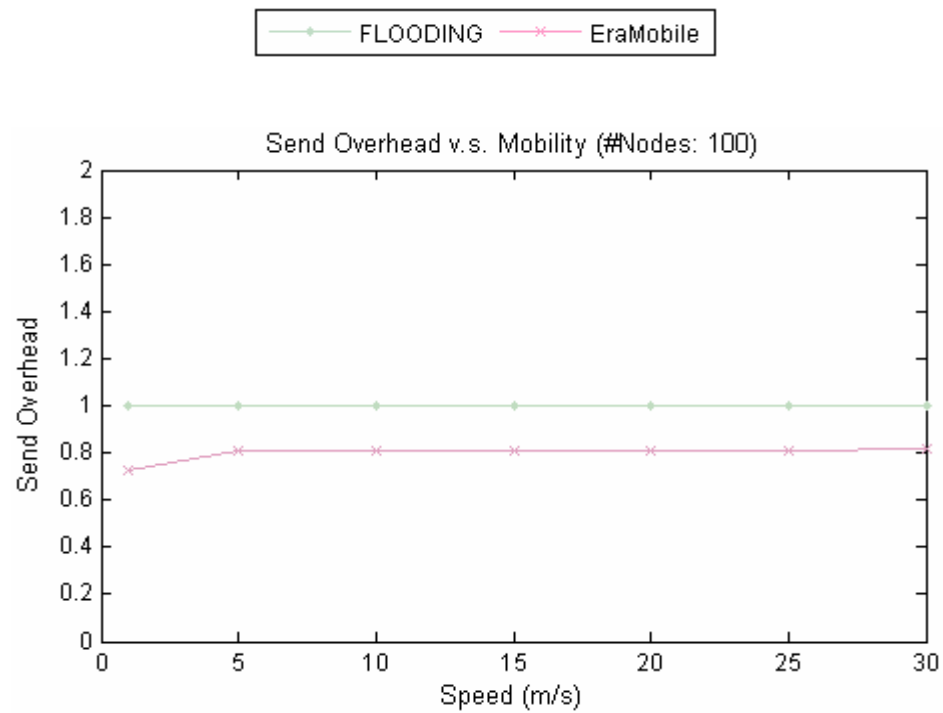


Figure 6.12. The effect of mobility on the send-overhead of the protocols in high node density

6.2.2 Effect of Group Size

The number of nodes in a single group was varied from 10 to 50 for flooding, EraMobile and MAODV and from 60 to 140 for flooding and EraMobile since the MAODV implementation on *ns-2* can not simulate the scenarios having more than 50 nodes. The speed of the nodes was fixed at 5m/s. The node densities, number of neighbors around a node, observed for each group size is shown in table 6.1. The node density values that EraMobile obtained through gossip and request message were also compared with the exact values obtained from *ns* simulator. The difference was always less than 5%.

Number of nodes	Node density	Number of nodes	Node density
10	2.2	60	13.33
20	4.28	80	18.14
30	6.45	100	23.05
40	8.82	120	27.6
50	11.39	140	32

Table 6.1. The node densities obtained for each group size in the simulation area of 1000m x 1000m

In the simulations up to 50 nodes, EraMobile achieves 100% reliable data delivery for almost all scenarios as shown in fig.6.13 and 6.14. The *throughput* of flooding and MAODV is increasing with number of nodes since they have better-connected network in higher node densities. The mobility of 5m/s prevents MAODV to catch a packet delivery ratio over 90% and a *reliable-throughput* over 40%. The *receive-overhead* of all protocols is increasing with the node density since all of them use broadcast communication, i.e. for data delivery in flooding, for both data delivery and group maintenance in MAODV, for gossip dissemination in EraMobile. An increase in the number of neighbors around a node means that a broadcast packet will be received by more nodes. This situation is the reason behind the rise of the *receive-overhead* since both duplicate data and control packets such

as link-updates and gossip messages are transmitted through broadcasting. Additionally, increasing node density causes more packets collisions, especially for broadcast packets. This fact causes a very slight increase in the send-overhead of EraMobile since it requires re-transmission of a packet which has been lost.

In the simulations from 60 nodes to 140 nodes, both flooding and EraMobile exhibit great throughput performance as shown in fig. 6.17 and 6.18. However, flooding outperforms EraMobile in very high node density. The reason of this situation is the limited simulation time. Since the adaptivity mechanism of EraMobile spreads the overhead in time to prevent congestions and packet losses, EraMobile needs more time to deliver the data packets to all members in the group. On the other hand, increasing amount of broadcasts with higher number of neighbors improved the *reliable-throughput* performance of flooding. In higher node density, nodes have greater chance to receive all the packets in spite of link failures and collisions due to large amount of duplicate data broadcasts. For instance, in the scenario with 140 nodes, a node, which has nearly 32 neighbors as shown in table 6.1, may have 32 copies of the same data packet since its all neighbors broadcast the data packets they delivered. Even if this situation increases the throughput of flooding, it costs a huge amount of *receive-overhead* as shown in fig. 6.19. The *receive-overhead* of flooding is 25.5 times more than the *receive-overhead* of EraMobile in the 140-node scenario. The *sent-overhead* of EraMobile shows very slight increase due to the increase in packet losses despite the high node density as seen in fig. 6.20.

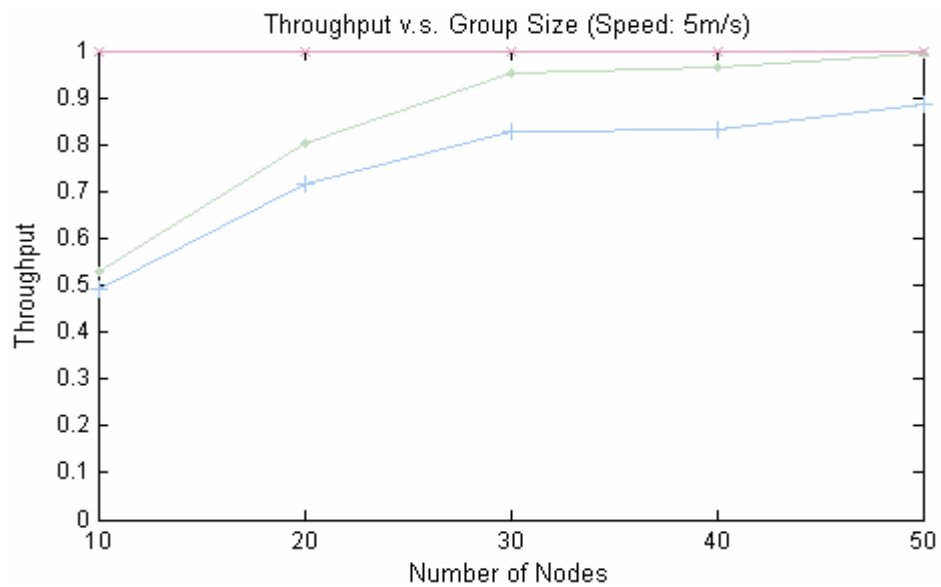


Figure 6.13. The effect of group size on the throughput of the protocols

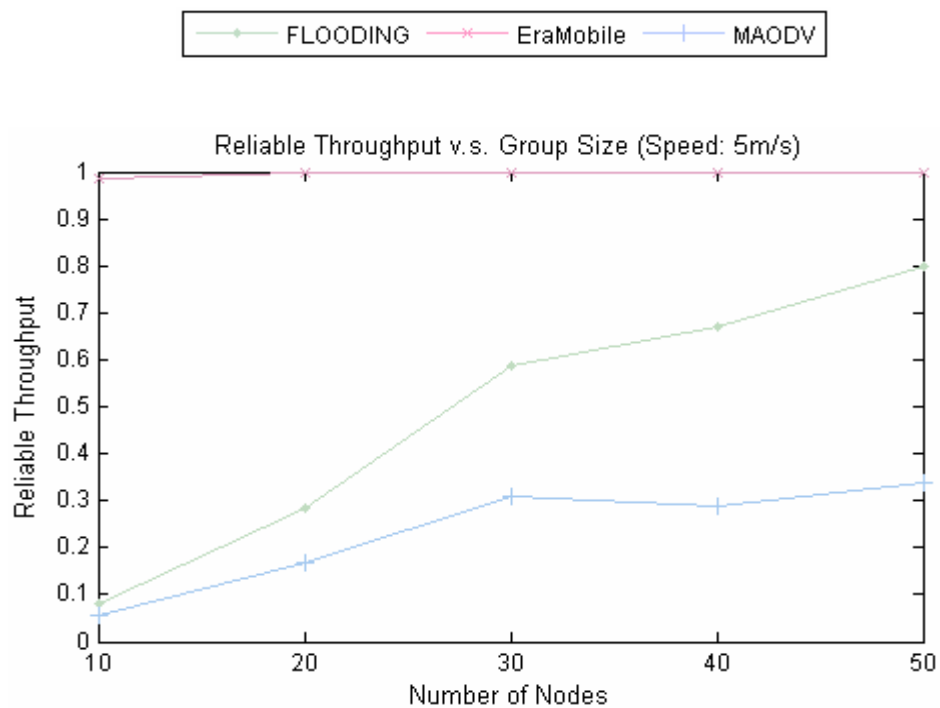


Figure 6.14. The effect of group size on the reliable-throughput of the protocols

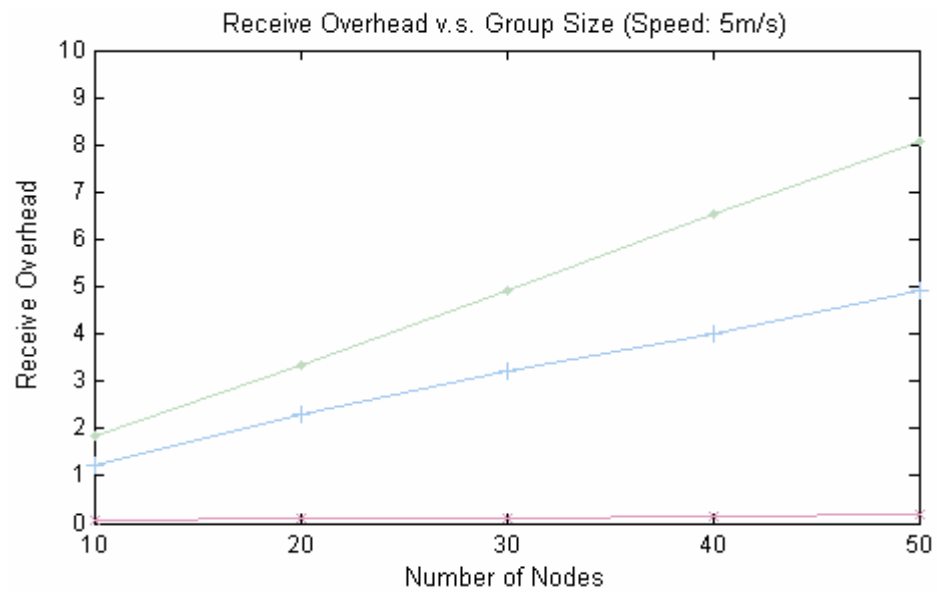


Figure 6.15. The effect of group size on the receive-overhead of the protocols

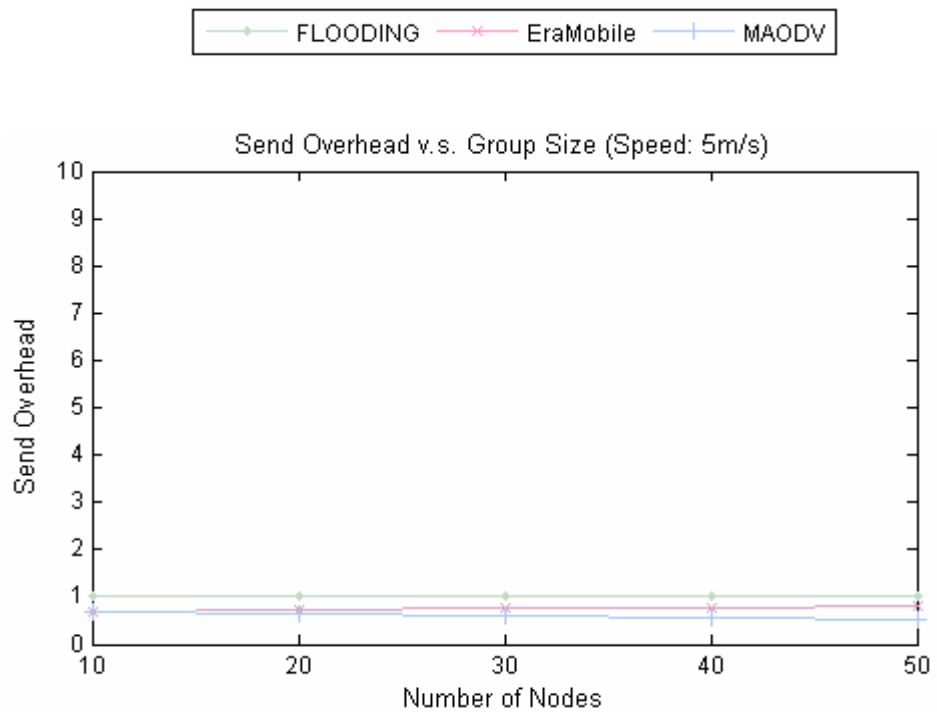


Figure 6.16. The effect of group size on the send-overhead of the protocols

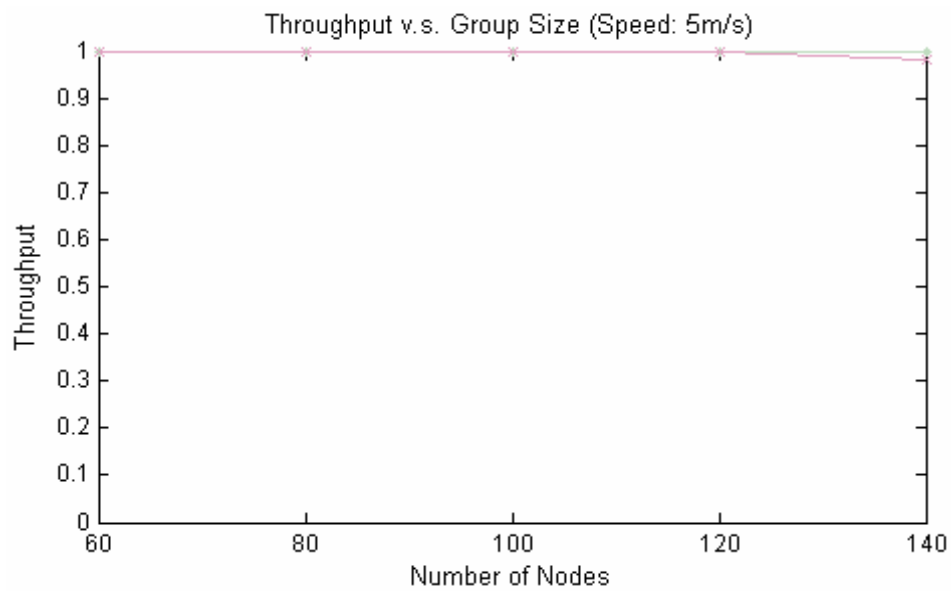


Figure 6.17. The effect of group size on the throughput of the protocols

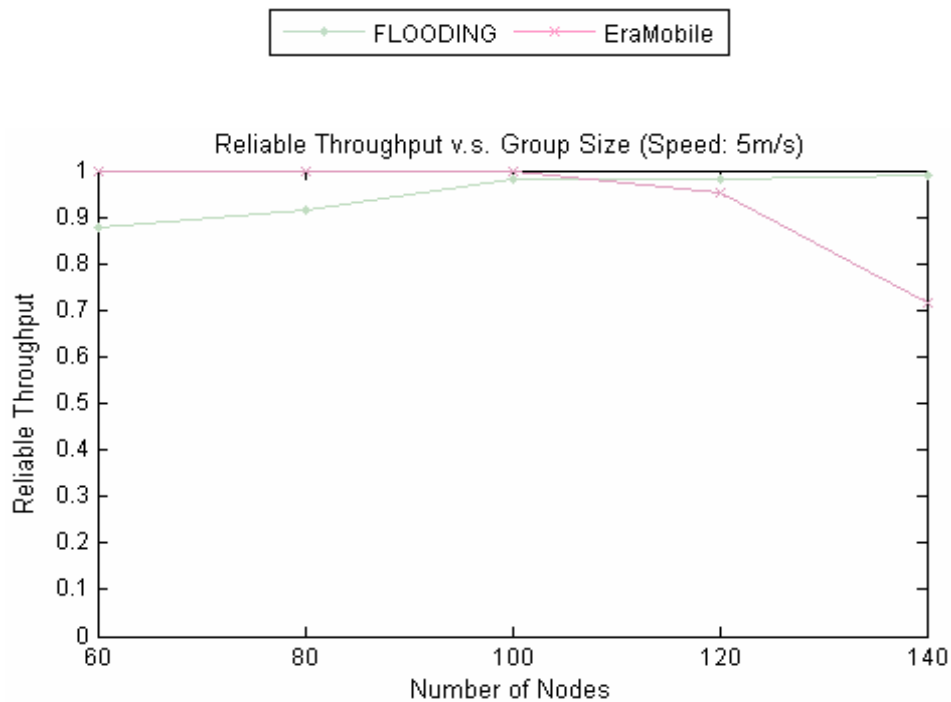


Figure 6.18. The effect of group size on the reliable-throughput of the protocols

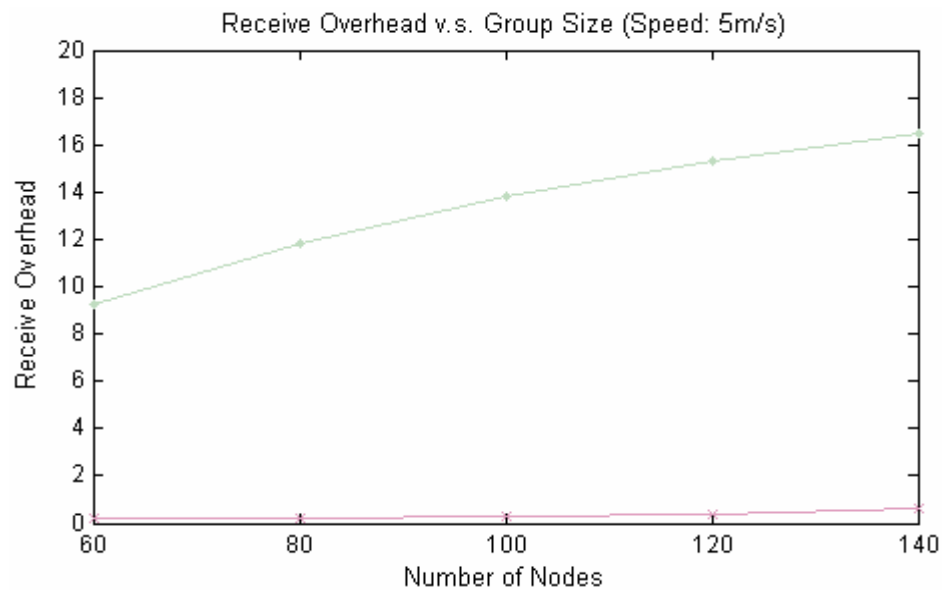


Figure 6.19. The effect of group size on the receive-overhead of the protocols

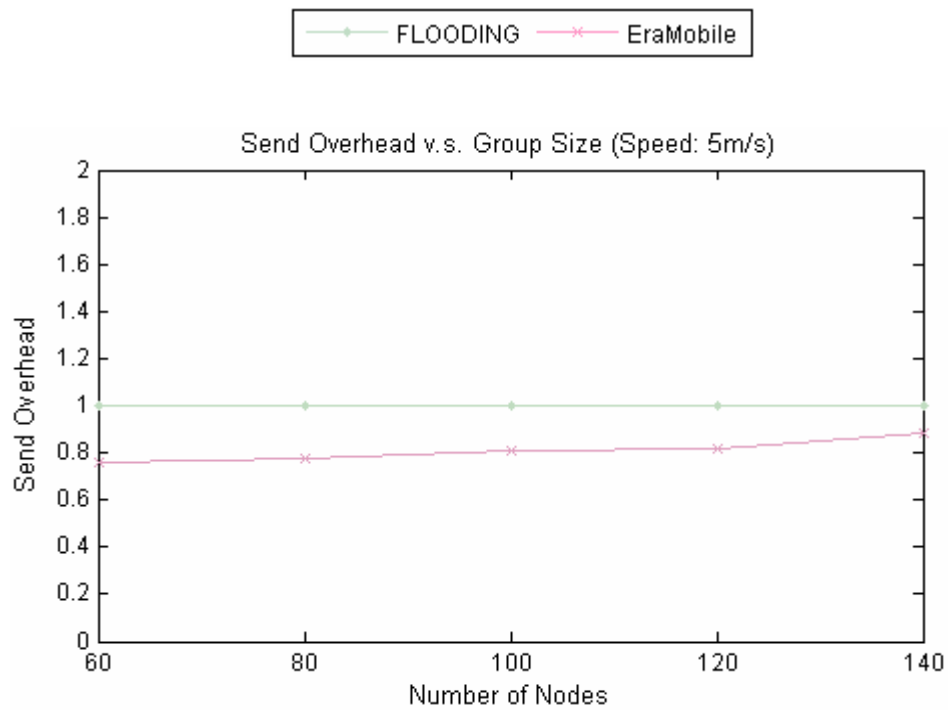


Figure 6.20. The effect of group size on the send-overhead of the protocols

6.2.3 Effect of Network Load

The number of data packets generated by the traffic source per second was varied from 1 to 16 with the payload size of 512. The single multicast group consists of 20 nodes moving at the speed of 5m/s. While the traffic rate of 1 packet/sec symbolizes the light traffic, the rate of 16 packets/sec represents heavy traffic [58].

The reaction of all protocols to increasing traffic is similar to each other. EraMobile achieves 100% reliable delivery for all traffic rates except the heavy traffic scenario as shown in fig. 6.21 and 6.22. In heavy traffic scenario, the throughput performance of the protocols tends to decrease due to the highly congested network environment. The throughput drop ratios of the protocols are 4%, 7% and 11% for MAODV, flooding and EraMobile respectively. The main reason of the decrease in the throughput of EraMobile is the lack of time in order to deliver such large amount of data packets with limited-size request and transmission messages. The adaptivity mechanism is also inadequate to deal with congestion arising from packet generation rate since it operates based on node density. The overhead of the protocols does not considerably change with the packet rate as shown in fig. 6.23 and 6.24. The higher traffic rates increase the number of data packets delivered by the nodes in all protocols. It also increases the duplicate data packets in flooding and MAODV. This situation causes the *receive-overhead* to remain constant with the traffic load. The total size of request and gossip messages in EraMobile is increased as well since a node requests and propagates more data with the rising packet in the higher traffic load.

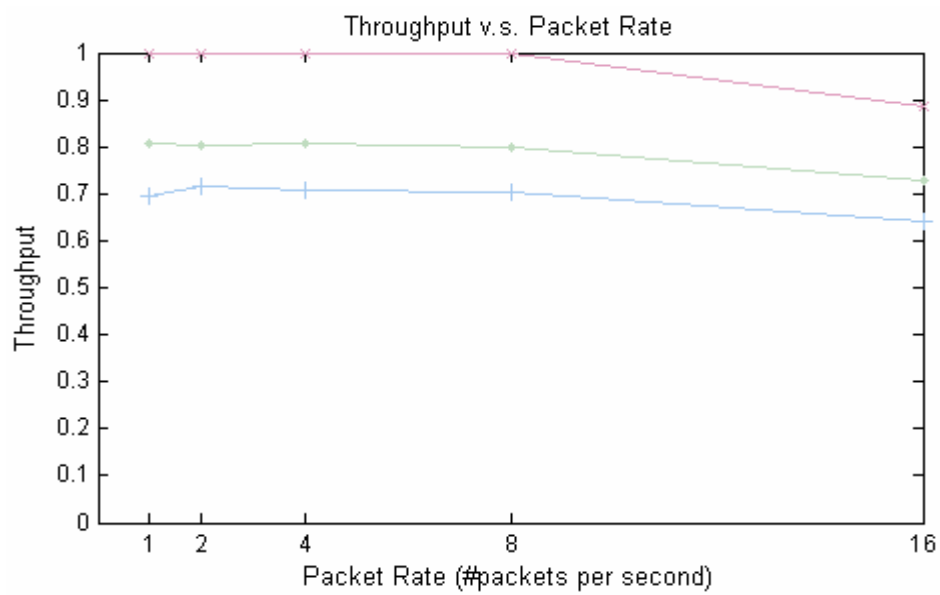


Figure 6.21. The effect of traffic load on the throughput of the protocols

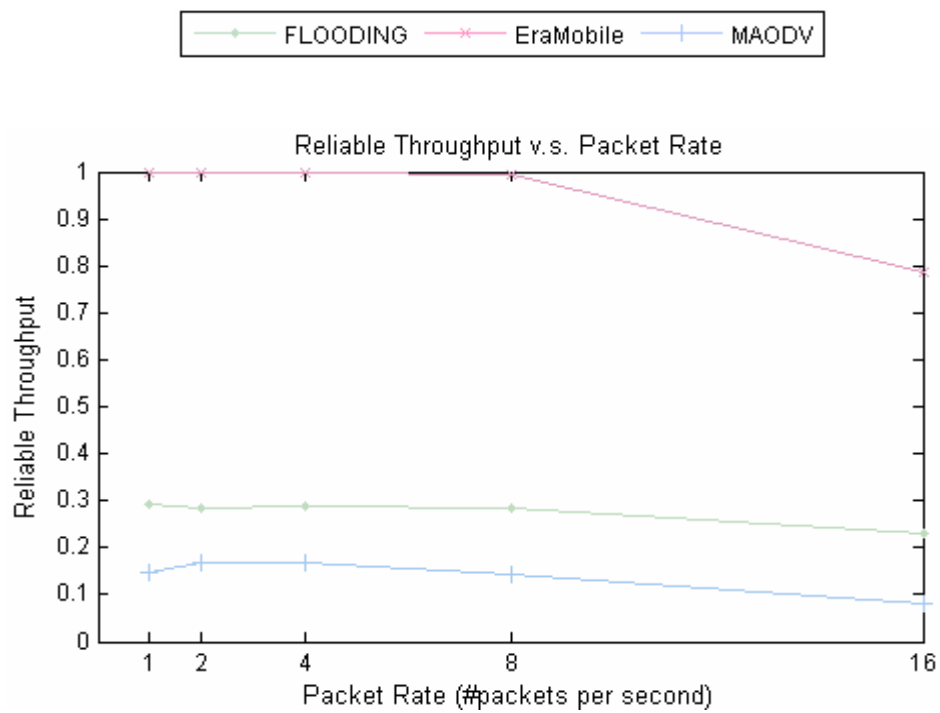


Figure 6.22. The effect of traffic load on the reliable-throughput of the protocols

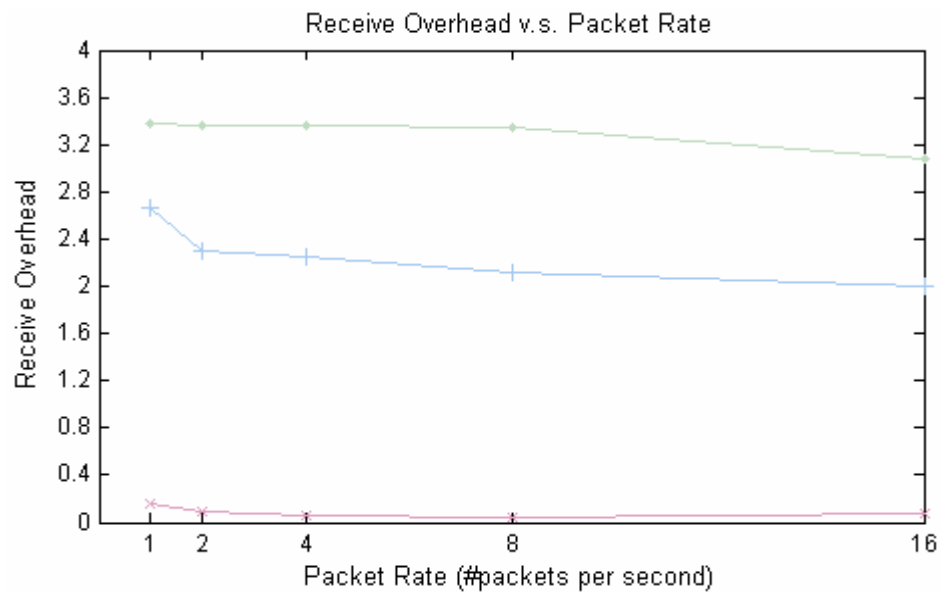


Figure 6.23. The effect of traffic load on the receive-overhead of the protocols

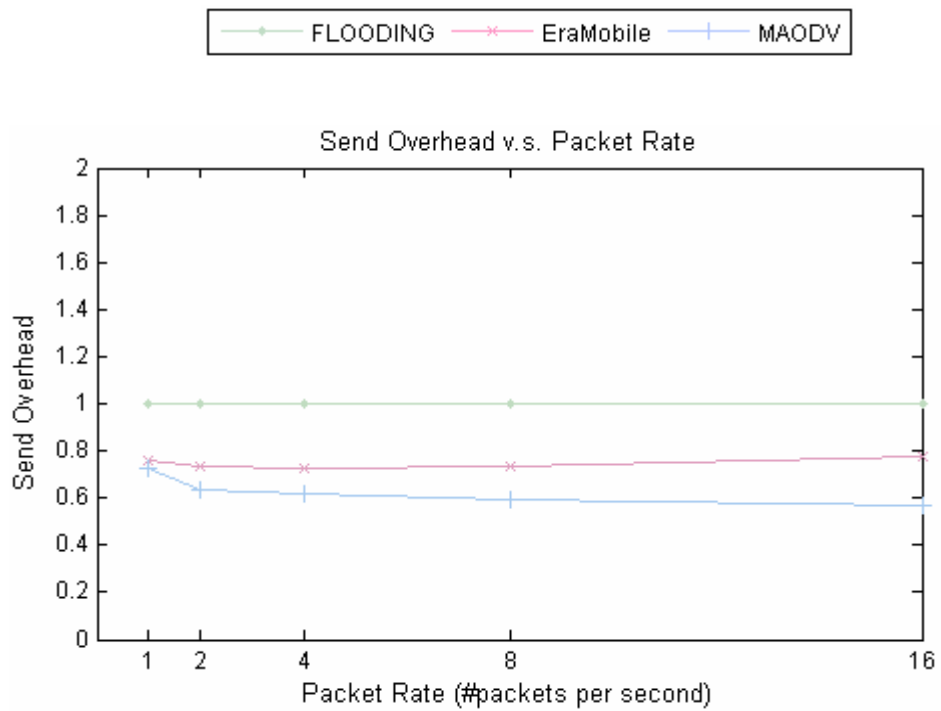


Figure 6.24. The effect of traffic load on the send-overhead of the protocols

6.2.4 Effect of Multiple Groups and Senders

The number of both multicast groups and senders was varied from 1 to 5 assigning one sender to each group as shown in table 6.2. Total number of nodes in the network was fixed at 60 with the mobility of 5m/s. MAODV performance could not be examined for multiple groups since its implementation was developed for single group scenarios.

Number of Groups	Number of Senders	Number of Members in Each Group
1	1	60
2	2	30
3	3	20
4	4	15
5	5	12

Table 6.2. Multicast groups, senders and size of groups

Each node in flooding broadcasts all the data packets it received without regard to its multicast group. This behavior enables the dissemination of flooding waves across the network over multiple hops, which may belong to different multicast groups. The cost of obtaining good level of network connectivity by this operation is huge amount of overhead as shown in fig. 6.27 and 6.28. Both the *receive-overhead* and the *send-overhead* substantially increase since nodes receive and send many data packets that they do not deliver. Such huge amount of overhead also decreases the throughput of flooding as shown in fig. 6.25 and 6.26. However, EraMobile can deal with increasing number of groups and senders by keeping the overhead at minimum. Thus, it achieves 100% reliable data delivery for all scenarios except the 5-group scenario, where its *throughput* is 99.9% and *reliable-throughput* is 97.5%. In EraMobile, when a node receives a packet from another multicast group, it just drops it after adding the size of the packet to the *receive-overhead*. These packets coming from members of other multicast groups are the reason behind the increasing overhead of EraMobile.

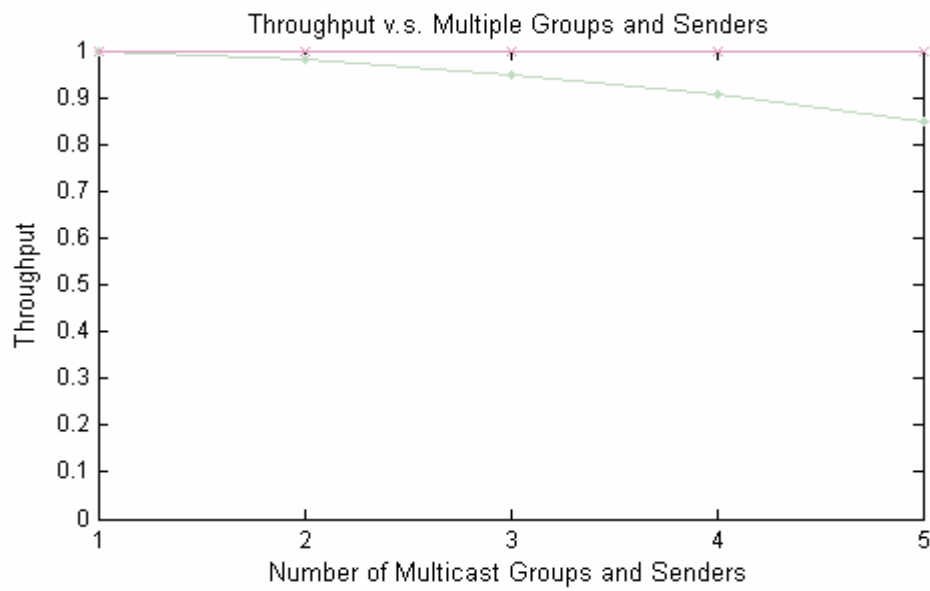


Figure 6.25. The effect of multiple groups and senders on the throughput of the protocols

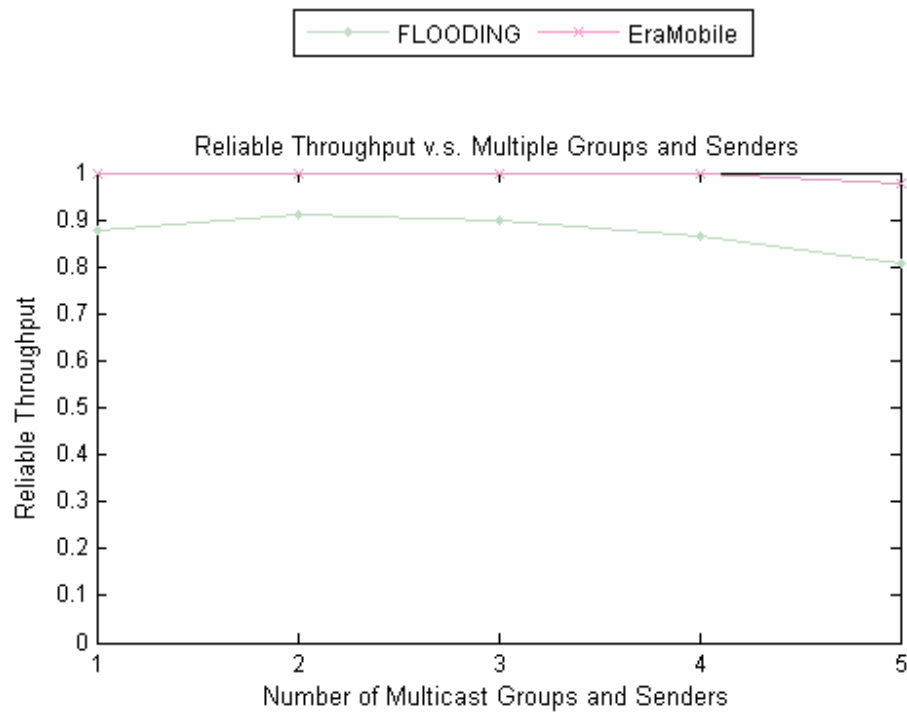


Figure 6.26. The effect of multiple groups and senders on the reliable-throughput of the protocols

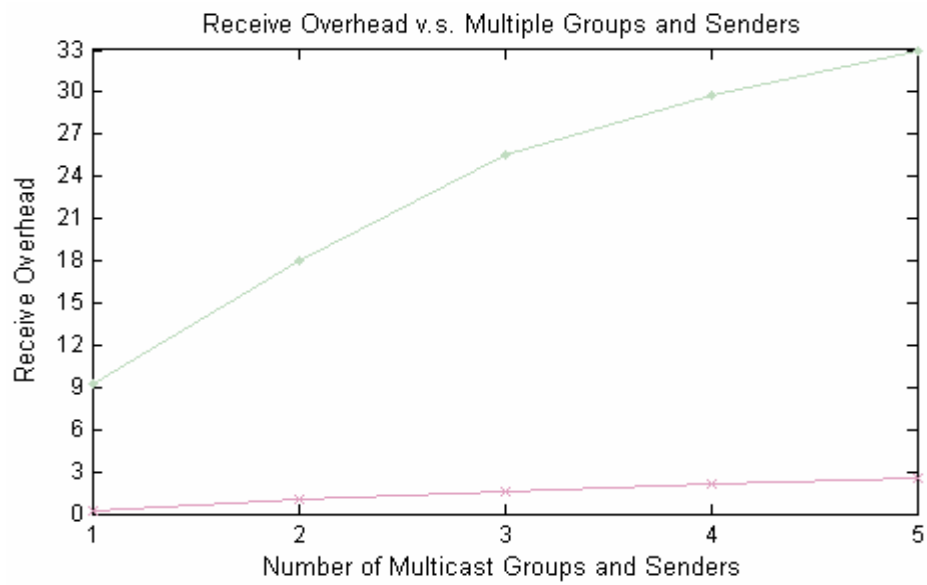


Figure 6.27. The effect of multiple groups and senders on the receive-overhead of the protocols

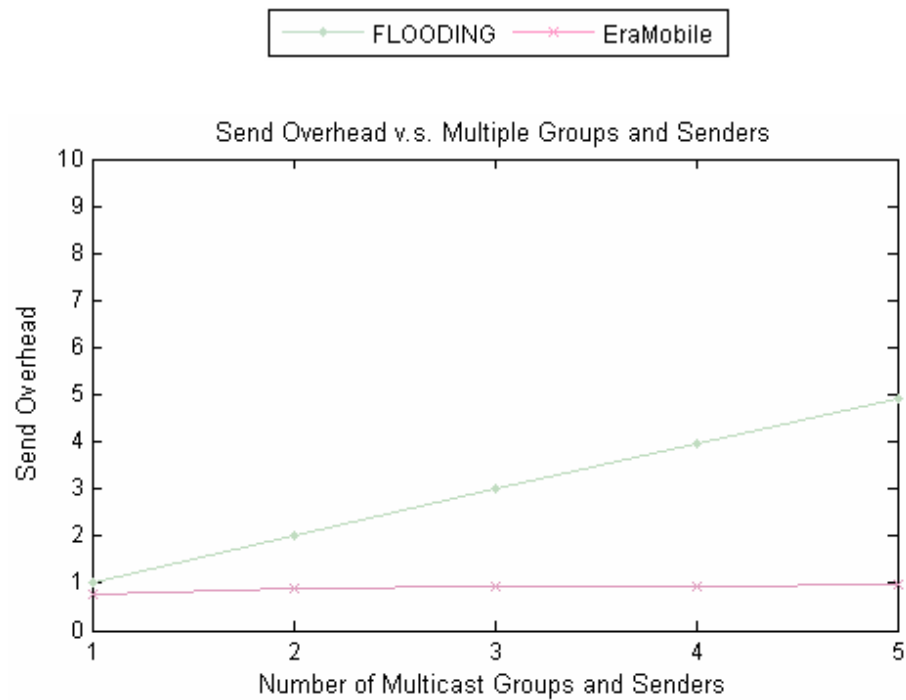


Figure 6.28. The effect of multiple groups and senders on the send-overhead of the protocols

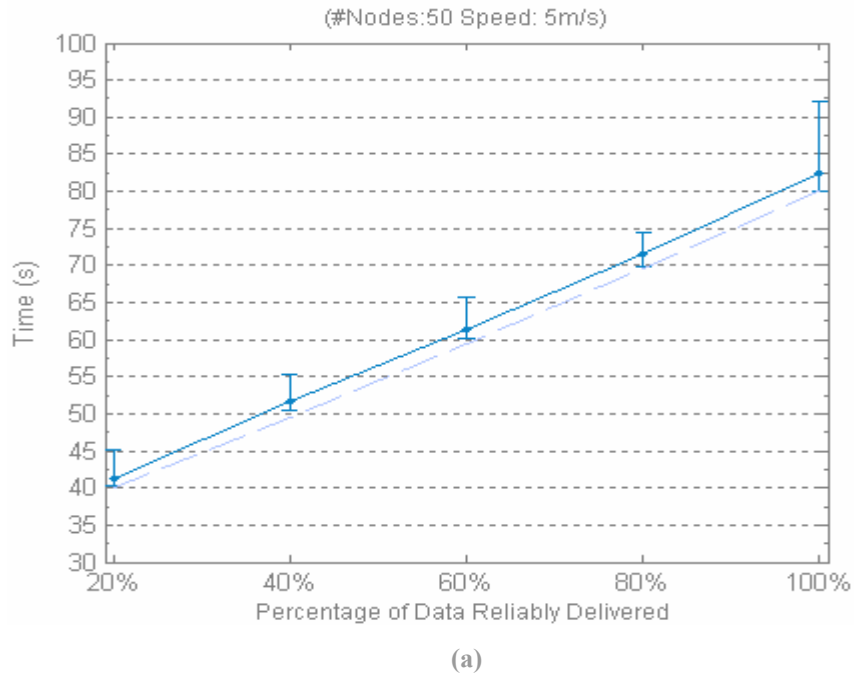
6.3 Discussion

EraMobile achieved fully reliable data delivery in the dynamic environment of mobile ad hoc networks for most of the scenarios presented above. The mobility came out as a positive factor assisting the dissemination of data as in real epidemic. The throughput performance of MAODV at higher mobility values showed that structure-based approaches can not suit to mobile ad hoc networks. The adaptivity mechanism of EraMobile also enabled it to exhibit great performance in both sparse and highly dense networks without requiring any exterior adjustment. The throughput performance of EraMobile showed slight decrease only in very high node density and heavy traffic scenarios since it could not complete the full data delivery till end of the simulations. However, it can eventually achieve fully reliable data delivery over time if the network conditions are not extremely adverse. The only case in which EraMobile can not obtain fully reliable data delivery is removal of a data packet from the buffers of all the nodes while there are still some nodes needing it. The probability of such a situation can be reduced by increasing the stability threshold value and storage space.

Besides its superior throughput performance, EraMobile also decreases the overhead exposed by the nodes very substantially. In flooding and MAODV, the main reason of the *receive-overhead* is duplicate data packets. The severity of the situation can better be understood when one considers the total size of gossip and request messages received by a node is 149774 bytes for the 50-node scenario in which number of generated data packets is 1740 and the mobility is 5m/s. This amount of overhead corresponds to only 292.5 duplicate data packets out of 1740 generated data packets with the payload size of 512 byte. However, in flooding, all the neighbors broadcast the data packets, i.e. $1740 \times (\#neighbors - 1)$ duplicate data packets are received by a node. EraMobile eliminates the redundant data transmissions by the gossip mechanism. The *send-overhead* of EraMobile remains always stable in variable scenarios. It may be expected that *send-overhead* should

become over one since the data transmission is performed by peer-to-peer communications. However, the first dissemination of data packets is carried out through broadcast transmission by the source node once the data is generated. Thus, 20-30% of the nodes, number of neighbors around the source node as seen in table 6.1, receive the data packets with one single broadcast transmission. It also can not cause duplicate data packets since none of nodes have a data packet before the sender broadcasts it.

High performance of EraMobile in terms of throughput, reliability and overhead is partially based on a trade-off with delay. The current MANET conditions, i.e. limited bandwidth decreasing with node density [59], can hardly enable most of the delay sensitive applications like real-time streaming to operate properly in mobile ad hoc settings. Thus, we traded off delay for the reliable and overhead-efficient multicast delivery. The time required for the completion of fully reliable data delivery is presented in fig. 6.29. The higher node density takes longer time as seen in fig. 6.29.b due to the reduced traffic to prevent congestion.



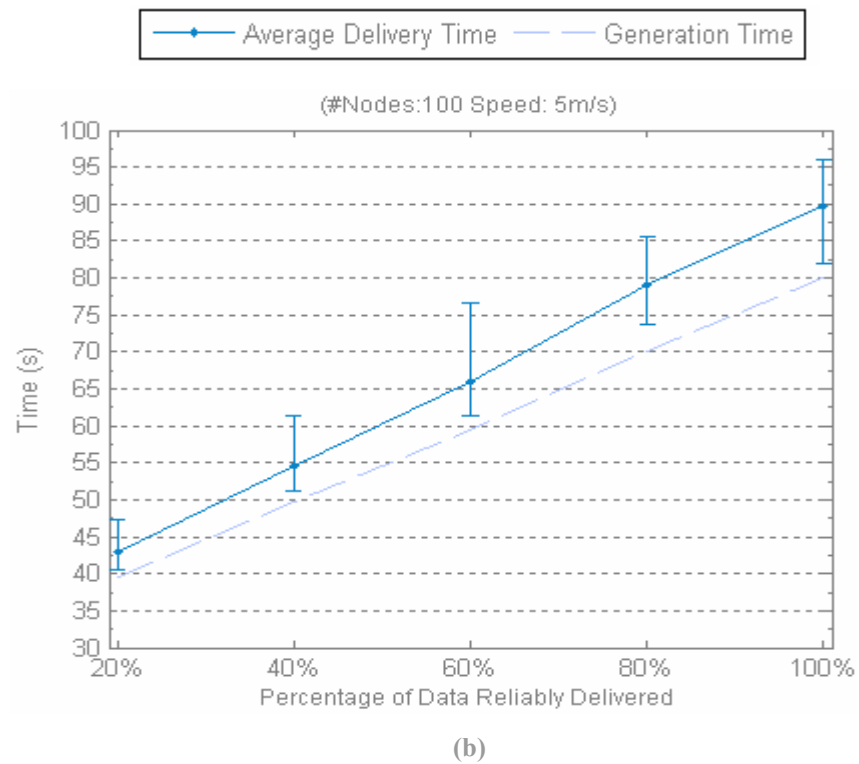


Figure 6.29. Time for completion of data delivery. The source node generates 2 packets per second between 30s. and 80s. The dashed lines represent the sending time of the data packets by the source node. The error bars show the difference between maximum and minimum delivery times of the nodes.

Chapter 7

CONCLUSION AND FUTURE WORK

This thesis presents a new protocol, Epidemic-based Reliable and Adaptive Multicast protocol for Mobile ad hoc networks (EraMobile), for reliable multicast delivery in mobile ad hoc networks. The proposed protocol utilizes an epidemic-based method to cope with dynamic and unpredictable topology changes arising from the mobility. This method does not employ any tree- or mesh-like structure for multicast operation. Instead, both data delivery and recovery of missing data packets are carried out by gossip messages without relying on any underlying routing protocol.

The gossip messages are disseminated by exploiting the broadcast nature of wireless medium. The data delivery is performed through peer-to-peer communications succeeding the gossip broadcasts. This model enables the protocol to operate without having any information related to network topology, neighboring nodes and multicast group members. The only information that a multicast member should know is the id of its multicast group. The gossip mechanism also substantially minimizes the overhead incurred by eliminating redundant data transmissions.

Another distinguishing feature of EraMobile is that it can adapt to varying node densities by adjusting the protocol parameters dynamically. Thus, it can achieve reliable data delivery both in sparse and dense networks.

We modeled and implemented EraMobile on *ns-2* and evaluated its performance through extensive simulations for a wide range of MANET scenarios. We compared it with flooding, due to its throughput achievement, and with MAODV, due to its overhead efficiency. The simulation results showed that EraMobile can provide 100% reliable data delivery for most of the scenarios outperforming the other protocols in terms of data delivery and overhead efficiency. The mobility simulations verified the fact that its unstructured nature brings in mobility-friendly characteristic to the protocol. The adaptivity mechanism enabled EraMobile to obtain fully reliable multicast delivery for almost all node densities by minimal overhead.

Consequently, we designed an epidemic-based reliable multicast protocol for MANETs and enhanced it with an adaptivity mechanism for varying node densities. We also demonstrated the efficiency of this protocol through comprehensive simulation studies. EraMobile can be utilized by all the applications, which are not delay sensitive, aiming reliable multicast delivery with minimal overhead in highly dynamic environment of mobile ad hoc networks.

In the near future, we will study the delay characteristic of EraMobile in varying conditions such as mobility and node density. We also consider enhancing our protocol with a global congestion control mechanism to slow the data rate of the senders when the network is highly loaded. Additionally, we will investigate the reaction of EraMobile to the improvements presented below:

1. *Quick Gossip*: This improvement proposes broadcasting of gossip messages right upon reception of a data message. Thus, neighboring nodes may receive a data message without waiting for the next gossip round of the node, which has recently had the message missed by neighbors. This improvement is likely to lower the delay in suitable network conditions. However, its influence on the overall performance of EraMobile should be carefully investigated. It may

degrade the protocol performance in dense or heavily loaded networks increasing the congestion. The adaptivity mechanism should be capable of deciding in which conditions this improvement will be disabled.

2. *Initial Flooding*: This improvement proposes to increase number of nodes participating in initial broadcasting of newly generated data packets. In the current implementation, only the source node broadcasts the data packets once they are generated. The dissemination of data can be accelerated by allowing the nodes, which are in the neighborhood of the source node, to broadcast the data packets they received from the source node. This improvement can be performed by setting the TTL values of broadcast packets, i.e. only neighbors of the source node can broadcast a data packet if its TTL value has been set to 2 by the sender. Such a behavior can accelerate the dissemination of data at the cost of duplicate data overhead.
3. *Broadcast Threshold*: The MAC protocol, presently IEEE 802.11, delivers *broadcast* packets without employing a congestion avoidance technique and an acknowledgement mechanism. Thus, broadcast transmission is more prone to packet losses compared to unicast transmission. Additionally, redundant data broadcasts may cause queue overflows at the MAC layer unnecessarily. EraMobile mainly carries out the data dissemination by peer-to-peer communications avoiding the negative factors mentioned above. However, broadcast transmission can be utilized, especially in dense networks, to perform multicast operation in a shorter time without occupying the wireless medium with request-send unicast communications. In order to implement such a scheme, EraMobile nodes may wait for a small time after each gossip broadcast and count request messages received. If the number of request messages received for the same data is greater than a pre-defined threshold then related data can be

broadcast. This improvement may contribute to throughput and delay performance of the protocol positively degrading its overhead efficiency.

In the long term, we plan to modify our protocol for the settings employing directional-antennas since the current MANET environment put important limitations onto performance of multicast protocols.

BIBLIOGRAPHY

- [1] S. Deering, Host extensions for IP multicasting, RFC 1112, August 1989, available at <http://www.ietf.org/rfc/rfc1112.txt>.
- [2] S. Paul, 1998 Multicasting on the Internet and Its Applications. Kluwer Academic Publishers.
- [3] I. Chlamtac, M. Conti, J. Liu, Mobile Ad hoc Networking: Imperatives and Challenges Ad hoc Network Journal, Vol.1 N.1 January-February-March, 2003.
- [4] S.J. Lee, M. Gerla, C.C. Chiang, On-Demand Multicast Routing Protocol. In Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC '99, pages 1298–1304, September 1999.
- [5] E.M. Royer and C.E. Perkins, Multicast Operation of the Ad hoc On-Demand Distance Vector Routing Protocol, Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Seattle, WA, August 1999, pp. 207-218.
- [6] J.J. Garcia-Luna-Aceves and E.L. Madruga, The Core-Assisted Mesh Protocol, IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad hoc Networks, vol. 17, no. 8, August 1999, pp. 1380-1394.
- [7] C.W. Wu and Y.C. Tay, AMRIS: A Multicast Protocol for Ad hoc Wireless Networks, Proceedings of the IEEE Military Communications Conference (MILCOM), Atlantic City, NJ, November 1999, pp. 25-29.
- [8] E. Bommaiah, M. Liu, A. McAuley, R. Talpade, AMRoute: Adhoc Multicast Routing Protocol, Internet-Draft, draft-talpade-manet-amroute-00.txt, August 1998, Work in progress.

-
- [9] H. Zhou and S. Singh, Content Based Multicast (CBM) in Ad hoc Networks, Proceedings of the ACM/IEEE Workshop on Mobile Ad hoc Networking and Computing (MOBIHOC), Boston, MA, August 2000, pp. 51-60.
 - [10] J.G. Jetcheva and D.B. Johnson, Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad hoc Networks. In Proceedings of the 2001 ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc 2001), pages 33–44, October 2001.
 - [11] S.K.S. Gupta and P.K. Srimani, An Adaptive Protocol for Reliable Multicast in Mobile Multihop Radio Networks, Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, LA, February 1998, pp. 111-122.
 - [12] K. Viswanath, K. Obraczka, G. Tsudik, Exploring Mesh and Tree-Based Multicast Routing Protocols for MANETs, IEEE Transactions on Mobile Computing, vol. 5, no. 1, pp. 28-42, January, 2006.
 - [13] S.J. Lee, W. Su, J. Hsu, M. Gerla, R. Bagrodia, A Performance Comparison Study of Ad hoc Wireless Multicast Protocols. In Proceedings of IEEE INFOCOM 2000, pages 565–574, March 2000.
 - [14] K. Obraczka, G. Tsudik, K. Viswanath, Pushing the Limits of Multicast in Ad Hoc Networks, in Proc. of the 21st Int. Conference on Distributed Computing Systems, pp. 719-722, Phoenix, USA, 2001.
 - [15] S. Bunchua and C.K. Toh, Performance Evaluation of Flooding-Based and Associativity-Based Ad hoc Mobile Multicast Routing Protocols. In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), September 2000.
 - [16] T. Kunz and E. Cheng. Multicasting in ad hoc networks: Comparing MAODV and ODMRP. In Proceedings of the Workshop on Ad hoc Communications, September 2001.

-
- [17] M. Pandey and D. Zappala, A Scenario-Based Performance Evaluation of Multicast Routing Protocols for Ad hoc Networks. Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05), pages: 31- 41, June 2005.
- [18] K. Obraczka, K. Viswanat, G. Tsudik, Flooding for Reliable Multicast in Multi-Hop Ad hoc Networks, *Wireless Networks*, pp. 627-634, Vol. 7, No. 6, 2001.
- [19] T. Gopalsamy, M. Singhal, D. Panda, P. Sadayappan, A Reliable Multicast Algorithm for Mobile Ad hoc Networks, *icdcs*, p. 563, 22nd IEEE International Conference on Distributed Computing Systems (ICDCS'02), 2002.,
- [20] K. Tang, K. Obraczka, S.J. Lee, M. Gerla, Reliable Adaptive Lightweight Multicast Protocol. In Proceedings of IEEE ICC 2003, May 2003.
- [21] K. Tang, K. Obraczka, S.J. Lee, M. Gerla, Congestion Controlled Adaptive Lightweight Multicast In Wireless Mobile Ad hoc Networks. In Proceedings of IEEE ISCC, July 2002.
- [22] V. Rajendran, K. Obraczka, Y. Yi, S.J. Lee, K. Tang, M. Gerla, Combining Source- and Localized Recovery to Achieve Reliable Multicast in Multi-Hop Ad hoc Networks, to appear in Proceedings of the Networking' 04, May 2004.
- [23] R. Chandra, V. Ramasubramanian, K. Birman, Anonymous gossip: Improving multicast reliability in mobile ad hoc networks. In Proc. 21st International Conference on Distributed Computing Systems (ICDCS), pages 275–283, 2001.
- [24] J. Luo, P. Th Eugster, J.P. Hubaux, Route Driven Gossip: Probabilistic Reliable Multicast in Ad hoc Networks, *Proc. INFOCOM*, pp. 2229-2239, 2003.
- [25] B. Ouyang, X. Hong , Y. Yi, A Comparison of Reliable Multicast Protocols for Mobile Ad hoc Networks, *IEEE SoutheastCon 05*, Fort Lauderdale, FL, April, 2005.

-
- [26] E. Vollset, and P. Ezhilchelvan, A Survey of Reliable Broadcast Protocols for Mobile Ad hoc Networks, Technological Report, School of Computing Science, University of Newcastle, Jun 2003.
- [27] P.B. Danzig, Optimally Selecting the Parameters of Adaptive Backoff Algorithms for Computer Networks and Multiprocessors, PhD thesis, University of California, Berkeley, December 1989.
- [28] M. Sirivianos, Epidemic Overlays for Multicast in Ad hoc Networks, Masters Thesis, University of California, San Diego, May, 2004.
- [29] P. Mohapatra, C. Gui, J Li, Group Communications in Mobile Ad hoc Networks, Computer, vol. 37, no. 2, pp. 52-59, February, 2004.
- [30] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, Epidemic algorithms for replicated database maintenance. Proceedings of the sixth annual ACM Symposium on Principles of distributed computing. Pages 1-12. 1987.
- [31] K. Birman, M. Hayden, O. Ozkasap, Z. Xaio, M. Bidiu, Y. Minsky, Bimodal multicast. ACM Transactions on Computer Systems. 17, 2 (May 1999), Pages 41-88.
- [32] Sensor Webs Project: <http://sensorwebs.jpl.nasa.gov/>.
- [33] http://en.wikipedia.org/wiki/Ad_hoc.
- [34] N. Abramson, The ALOHA system in Computer Networks, N. Abramson and F. Kuo, Eds. Englewood NJ: Prentice-Hall, 1973, pp. 501-518.
- [35] M. Frodigh, P. Johansson, and P. Larsson, Wireless ad hoc networking: The art of networking without a network, Ericsson Review, 2000.
- [36] <http://grouper.ieee.org/groups/802/11/>.
- [37] S. Corson, and J. Macker, Mobile Ad hoc Networking(MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, January 1999.
- [38] C. S. R. Murthy and B. S. Manoj, Ad hoc Wireless Networks: Architectures and Protocols, 2004.

-
- [39] J. Hoebeke, I. Moerman, B. Dhoedt, P. Demeester, An overview of mobile ad hoc networks: applications and challenges. *The Journal of The Communications Network*. Vol. 3: (3) 60-66; 2004.
- [40] <http://en.wikipedia.org/wiki/Epidemics>.
- [41] A. Birrell, R. Levin, R. Needham, M. Schroeder, Grapevine: an exercise in distributed computing. *CACM*, 25(4):260--274, April 1982.
- [42] A. Golding and K. Taylor, Group membership in the epidemic style. Technical report, University of California, Santa Cruz, Jack Baskin School of Engineering, 1992. UCSC-CRL-92-13.
- [43] R. Ladin, B. Liskov, L. Shirira, S. Ghemawat, Providing Availability Using Lazy Replication. *ACM Transactions on Computer Systems*, 10(4):360-391, 1992.
- [44] Y. Renesse and M. Hayden, A gossip-style failure detection service, in *Proceedings of Middleware'98*, pages 55–70. IFIP, September 1998.
- [45] M.J. Lin and K. Marzullo, Directional gossip: Gossip in a wide area network. In *European Dependable Computing Conference*, pages 364-379, 1999.
- [46] P. Eugster, R. Guerraoui, S. Handrukande, A.M. Kermarrec, P. Kouznetsov, Lightweight probabilistic broadcast. In *Proc. of the Int. Conf. on Dependable Systems and Networks (DSN2001)*, pages 443-452, 2001.
- [47] L. Li, J. Halpern, Z. Haas, Gossip-based ad hoc routing. In *Proceedings of the 21st Conference of the IEEE Communications Society (INFOCOM'02)*, 2002.
- [48] J. Kulik, W. Heinzelman, H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks. *Wirel. Netw.* 8, 2/3 (Mar. 2002), 169-185.
- [49] J.N. Al-Karaki, and A.E. Kamal, A Taxonomy of Routing Techniques in Wireless Sensor Networks, in *Sensor Networks Handbook*, I. Mahgoub and M. Ilyas (eds.), CRC Publishers, 2004.

-
- [50] Y. Sasson, D. Cavin, A. Schiper, Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks,' in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2003), 2003.
 - [51] L. Kleinrock and J. Silvester, Optimum transmission radii for packet radio networks or why six is a magic number. In Proceedings of the IEEE National Telecommunications Conference, pages 4.3.1–4.3.5, December 1978.
 - [52] H. Takagi and L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Trans. Commun. 32 (1984) 246-257.
 - [53] C. Perkins, Ad hoc On Demand Distance Vector (AODV) Routing IETF, Internet Draft, draft-ietf-manet-aodv-00.txt, November 1997.
 - [54] VINT Project. The UCB/LBNL/VINT network simulator-ns (Version 2). <http://www.isi.edu/nsnam/ns>.
 - [55] CMU Monarch Project. Mobility Extensions to ns-2, 1999. Available from <http://www.monarch.cs.cmu.edu/>.
 - [56] Y. Zhu and T. Kunz, MAODV Implementation for NS-2.26. Systems and Computing Engineering, Carleton University, Technical Report SCE-04-01, January 2004.
 - [57] Ns Documentation. <http://www.isi.edu/nsnam/ns/doc/>.
 - [58] T. Kunz, Reliable Multicasting in MANETs, Contractor Report, Communications Research Centre, Ottawa, Canada, July 2003.
 - [59] P. Gupta, P. R. Kumar, The Capacity of Wireless Network, IEEE Trans. On Information Theory, Vol.46, No.2, pp.388-404, Mar. 2000.

VITA

Zülküf Genç was born in Istanbul, Turkey on April 15, 1981. He received his B.Sc. degree in Computer Engineering from Istanbul Technical University, Istanbul, in 2003. From September 2003 to August 2005, he worked as a teaching and research assistant in Koç University, Istanbul, Turkey and participated in the “TUBITAK-COST Action: 279, Analysis and Design of Multiservice Networks Supporting Mobility, Multimedia, Internetworking” project. He presented a technical report at COST-279 Action Meeting'04 (Ghent, Belgium) and a conference paper at ICIT'04 (Istanbul, Turkey). He has currently two conference and a journal submissions. At the time of press, he had a Ph.D. admission from the Department of Telecommunications of the Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS) at Delft University of Technology for the Wireless and Mobile Communications (WMC) Group.