Computer Networks 53 (2009) 1409–1430

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Epidemic-based reliable and adaptive multicast for mobile ad hoc networks

Oznur Ozkasap [a,*], Zulkuf Genc [b], Emre Atsan [a]

[a] *Department of Computer Engineering, Koc University, Istanbul, Turkey*
[b] *Department of Telecommunications, Delft University of Technology, The Netherlands*

ARTICLE INFO

Article history:
Received 29 December 2007
Received in revised form 8 January 2009
Accepted 14 January 2009
Available online 7 February 2009

Responsible Editor: M. van Steen

Keywords:
Mobile ad hoc networks
Reliable multicast
Adaptive
Epidemic
Bio-inspired
Protocol design and analysis

ABSTRACT

An emerging approach to distributed systems exploits the self-organization, autonomy and robustness of biological epidemics. In this article, we propose a novel bio-inspired protocol: EraMobile (*E*pidemic-based *R*eliable and *A*daptive *M*ulticast for M*obile* ad hoc networks). We also present extensive performance analysis results for it. EraMobile supports group applications that require high reliability. The protocol aims to deliver multicast data reliably with minimal network overhead, even under adverse network conditions. With an epidemic-based multicast method, it copes with dynamic and unpredictable topology changes due to mobility. Our epidemic mechanism does not require maintaining any tree- or mesh-like structure for multicasting. It requires neither a global nor a partial view of the network, nor does it require information about neighboring nodes and group members. In addition, it substantially lowers overhead by eliminating redundant data transmissions. Another distinguishing feature is its ability to adapt to varying node densities. This lets it deliver data reliably in both sparse networks (where network connectivity is prone to interruptions) and dense networks (where congestion is likely). We describe the working principles of the protocol and study its performance through comparative and extensive simulations in the ns-2 network simulator.

1. Introduction

Mobile ad hoc networks (MANET) have gained considerable interest and popularity in recent years since they have enormous potential in several fields of application. Lack of infrastructure, self-organization and mobility are the main reasons behind the popularity of MANETs. However, these features also introduce important challenges into the design of protocols and applications for ad hoc networks: for example, highly dynamic and unpredictable topological changes, low bandwidth, high error rates and limited power sources. Therefore, the design of network protocols, including multicasting, requires new approaches and distinctive changes.

Multicasting is an ideal communication paradigm for several application areas that require efficient support of group communication: for example, military operations, emergency and rescue missions, networked games, conferencing, and synchronization. Thus, many multicast protocols for ad hoc networks have been proposed such as [1,2] and their performance has been studied [3,4]. The simulation results in [3,4] reveal the interesting fact that the simplest dissemination method – flooding – attains better packet delivery ratios than the leading multicast protocols such as ODMRP and MAODV. The reason is that most of the existing solutions, including ODMRP and MAODV, deliver multicast data with mesh- or tree-based structures. Constructing and maintaining these structures constitutes overhead. Frequent topology changes arising from high mobility may impose prohibitive performance costs on multicast protocols in the presence of increasing overhead and link breakages.

In the dynamic environment of MANETs, information derived from the network topology can easily become outdated. Using this type of information, such as routing table

---

[*] Corresponding author. Tel.: +90 212 3381584; fax: +90 212 338 1548.
E-mail address: oozkasap@ku.edu.tr (O. Ozkasap).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.01.007

or group membership information, for the multicast structure may impose frequent updates on the protocol states. This update process can hinder the delivery of data packets. Furthermore, control messages triggered by the changes in the network can congest the wireless medium. Group communication paradigms may suffer from a wider spread of protocol state information that is required in order to deliver data scalably and robustly [5]. A potential solution should adapt to the dynamic environment of mobile ad hoc networks with a minimum of information about instantaneous network state.

In this article, we describe the design and performance evaluation of a novel reliable and adaptive multicast protocol for mobile ad hoc networks, EraMobile. Our solution is based on a bio-inspired epidemic method, since the stateless character of such methods fits well with the non-deterministic nature of MANETs (a non-determinism that arises from highly dynamic and unpredictable topology changes). The epidemic method of EraMobile is inspired by the anti-entropy technique described in [6]. This technique was proposed for dissemination of updates to replicated databases through gossip messages. Each node in the network periodically selects a random node and sends a gossip message containing the digest of database content to this node. If a pair detects any inconsistency between database contents, then it reconciles the inconsistency by exchanging necessary data messages. Similar probabilistic methods were used for multicasting in wired [7] and mobile ad hoc networks [8,9]. However, such methods require the nodes to have partial or global knowledge of the multicast group in order to select a random node from others for gossiping. The cost of obtaining such information in the dynamic environment of ad hoc networks is quite high. That is why, our solution is based on the idea of exploiting the broadcast nature of the wireless medium to send gossip messages. In our approach, nodes in a multicast group broadcast gossip messages locally, to neighboring nodes, instead of selecting a random node from a pre-defined list and sending the gossip message to this node as unicast. This slight conceptual change allows the multicast protocol to operate without any underlying routing protocol. Probabilistic behavior of EraMobile arises by each node gossiping its message history with one-hop neighbors (that is, a random subset of nodes). Note that, this random subset for each node changes dynamically due to node mobility characteristics, and also protocol's adaptivity mechanism based on node density levels. Furthermore, frequency of gossip messages that a node will send and probability of sending a request message are adjusted dynamically by the adaptivity mechanism. The protocol also requires neither a global nor a partial view of the network, nor does it requires information about neighboring nodes and group members for multicast delivery.

Our solution achieves high reliability with periodic gossip messages that let nodes recover missing data packets. This feature makes the protocol robust against transient adverse network conditions and delivery failures. Besides increasing reliability, the use of gossip messages also decreases bandwidth and energy requirements by reducing network overhead when compared to multicast flooding. Loss-sensitive network group applications can use our solution in order to achieve reliable multicast delivery with minimal overhead in the highly dynamic environment of MANETs. Considering that many routing protocols in ad hoc networks, e.g., ODMRP or MAODV, use beacon messages, to communicate changes in network topology, we believe that periodic small-sized gossip broadcasts do not burden the network unduly. We investigate major protocol parameters (namely, gossip interval and stability threshold) and study their effects in varying node densities. In addition, we have reinforced our model with an adaptivity mechanism. This mechanism can adjust protocol parameters dynamically according to node density levels. In sparse networks, the nodes can focus on delivering data reliably without concern about congestion. In dense networks, the nodes should consider the broadcast nature of the wireless medium, which is highly prone to congestion.

The article is organized as follows. In the next section, we give an overview of broadcasting and reliable multicasting in ad hoc networks, and then describe related work on epidemic-based approaches. In Section 3, we explain the system model, structure, algorithms and properties of EraMobile comprehensively. Section 4 describes the experimental setup, and Section 5 gives details of parameter analysis as well as density levels for adaptive behavior. In Section 6, we present and discuss performance analysis results. Section 7 includes concluding remarks and future directions.

2. Related work

In this section, we provide an overview of broadcasting and reliable multicasting in ad hoc networks. After that, we describe epidemic or gossip-based approaches in ad hoc networks and compare our solution with relevant studies.

2.1. Broadcasting in ad hoc networks

Broadcasting data to all nodes in a system is an essential service for several cooperative applications in ad hoc networks. Such a service should be designed to be efficient and reliable. In the simple broadcasting method of flooding, a source transmits a data message to all nodes in its wireless range. On receiving the data for the first time, each node forwards or rebroadcasts it to nodes in its range.

The work of Tseng et al. [47] studies broadcasting and related issues in a MANET. It shows that if blind flooding does the broadcasting, then problems of redundancy, contention, and collision emerge. These problems with flooding are known collectively as the broadcast storm problem. In order to deal with this problem, two directions were extensively studied. One is to reduce the probability of redundant broadcasts, and the other is to distinguish the timing of rebroadcasts. Based on these, probabilistic, counter-based, distance-based, location-based, and cluster-based schemes were developed for efficient broadcasting in MANETs and analyzed via simulations.

In probabilistic flooding, upon receiving a message for the first time, a node computes a fixed probability and rebroadcasts the message with that probability. It has been shown that plain flooding achieves the highest possible delivery ratio, particularly for sparse networks, when compared to probabilistic flooding. Furthermore, high delivery ratio levels in data broadcasting with probabilistic flooding have been demonstrated to be possible when the rebroadcasting probability is set to a rather high value [5,37]. For example, low density network simulations reported in [5] show that success rate (that is, delivery ratio) varies linearly with the rebroadcasting probability and high success rates are observed with probabilities greater than 0.9.

In order to reduce the number of redundant broadcasts, additional approaches such as counter-based, distance-based, and location-based rebroadcasting use local information at a node to decide whether to rebroadcast a message. For instance, the counter-based algorithm uses a counter to control the number of times a broadcast message is received at a node during the interval before rebroadcasting is performed. When the counter's value reaches a threshold for the message, the rebroadcast is suppressed. This scheme has been demonstrated to eliminate many redundant broadcasts in dense networks [47]. However, a drawback of such schemes is that they increase message transmission delays.

2.2. Reliable multicasting in ad hoc networks

Previous studies proposed several multicast protocols for MANET environments [1,2,10–15] and investigated their performance through extensive analysis [3,4,16–19]. Classifications of these multicast protocols are reviewed in [20,21]. For application areas that require high reliability, these best-effort multicast protocols do not provide sufficient packet delivery guarantees. In order to support reliability features in multicasting over MANETs, reliable multicast techniques have been proposed [22–25,8,9]. These protocols are classified as either deterministic or probabilistic, depending on their reliability guarantees [26], and as either ARQ (Automatic Retransmission reQuest)-based or gossip-based, depending on their recovery mechanisms [27]. Separately from these classifications, Rizzo and Vicisano [28] propose a FEC (Forward Error Correction) and ARQ-based hybrid reliable multicast protocol called Reliable Multicast Data Distribution Protocol (RMDP). Their technique uses erasure coding methods to tolerate some level of data loss at the receiver side in exchange for some redundancy.

RMA [22], a reliable multicast algorithm for MANETs, attains reliability with acknowledgment (ACK) messages from receivers to sources. This protocol represents a deterministic class of reliable multicast protocols that guarantee full packet delivery among group members. RMA assumes that senders have full group membership information that is maintained by flooding of join and leave messages throughout the network with no reliability guarantees. However, the unreliable join and leave mechanism can degrade the reliability of the protocol [26].

RALM (Reliable Adaptive Lightweight Multicast) [23,24] is a transport protocol built on ODMRP (On-Demand Multicast Routing Protocol) [1] to enhance its packet delivery ratio in small group operation scenarios. RALM achieves reliability with a congestion control mechanism using adjustable window sizes. It uses negative acknowledgment (NACK) feedback messages to adjust the congestion experienced by multicast receivers. Similarly to RMA, RALM assumes that senders have full group membership information. Simulation results [23,24] show that this deterministic protocol does well in static networks, where packet loss generally stems from congestion. In highly mobile networks, the congestion control mechanism may not solve the reliability problem [26,27]. Reliable, Adaptive, Congestion-Controlled Ad hoc Multicast Transport Protocol (ReACT) is an enhanced version of RALM with a local recovery mechanism [25]. In ReACT, receivers first try to recover packet losses from nearby nodes, such as upstream group members in the multicast structure. This local recovery mechanism improves the protocol's performance by preventing unnecessary back-offs and rate reductions.

2.3. Epidemic-based approaches and comparison

Epidemic communication was initially proposed for spreading updates in replicated databases [6] and then used for several purposes such as group membership tracking [29] and multicasting in wired networks [7,30–32], information dissemination [33–35], routing [36,37], broadcasting [38], and reliable multicasting [8,9] in MANETs. The most relevant works to our study are Anonymous Gossip (AG) [8] and Route Driven Gossip (RDG) [9,32].

AG [8] is one of the earlier protocols that use an epidemic-based approach to disseminate multicast data reliably in MANETs. It can be implemented on top of any tree- or mesh-based best-effort multicast protocol. The protocol has two phases. The first phase uses the underlying unreliable multicast protocol, MAODV [2], to disseminate the multicast data to the group. The concurrent second phase triggers an anonymous gossip mechanism to recover the missing messages, in order to guarantee that almost all reachable members receive the multicast packets. In the gossip mechanism, each node selects one of its neighbors randomly (selecting the nearest nodes with higher probability and distant nodes with lower probability) and sends a gossip message to it. When a member node receives a gossip message, it decides randomly to either accept the gossip message or forward it. A node accepting a gossip message compares the content of the 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 sees 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. 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 predict its probabilistic delivery ratio analytically [26].

The RDG protocol [9,32] uses a pure gossip scheme for reliable multicasting in ad hoc networks. Unlike AG, RDG does not rely on a multicast routing protocol. Instead, it uses an on-demand unicast routing protocol like Dynamic Source Routing (DSR) or Ad Hoc On-Demand Distance Vector (AODV) [39]. It disseminates 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 has no additional mechanism to form the list of group members. Instead, it collects group membership information with the underlying routing protocol. The gossip messages contain new data, the IDs of missing messages, the view of the membership and whether the node wishes to leave the group. A gossip receiver first updates its view by removing and adding nodes, then delivers the data and responds to the gossip sender by sending missing messages if the node has the data packet requested. RDG aims to achieve a probabilistic reliability that is expected to be predictable from simple information like the packet loss ratio. The paper presents an additional protocol called TA-RDG: topology-aware RDG. This variant applies similar heuristics to those in AG to partial topological information 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 scheme. 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 in addition to gossip content for predefined times by different members. This situation may cause redundant data transmissions, thus increasing overhead. It may also degrade the performance of the protocol in dense ad hoc networks, which are highly open to congestion.

Gossip-based ad hoc routing [37] proposes a generic gossiping method in which each node forwards a message with a probability, and applies this method to solve the problem of finding routes to nodes in ad hoc networks. It has been shown that gossiping can reduce the number of control messages considerably, hence lowering the overhead of the routing protocol. In particular, gossiping reduces the number of routing control messages by up to 35% compared with flooding, and also performs better. Flooding-based routing, even with various optimizations, still propagates large number of control messages needlessly. Our work differs from Haas et al. [37] since the latter tackles the generic gossiping in ad hoc networks and incorporates it in a routing protocol, whereas we focus on data multicasting without any routing support in MANETs. Although both studies use gossiping, we apply gossiping to the broadcast of a digest of message history to neighbors of a node. However, in [37], routing messages are forwarded via gossiping with a probability.

The recent study of Drabkin et al. [48] proposes RAPID, a reliable probabilistic dissemination protocol for MANETs. RAPID's key feature is probabilistic forwarding with deterministic corrective measures of gossiping and timer-based corrections. It sets the forwarding probability inversely proportionally to the number of observed one-hop neighbors of a node. In order to increase reliability, each node periodically gossips with its neighbors about which messages it has headers for. Moreover, RAPID offers timer-based corrections to resolve inconsistencies between the probabilistic assumptions and network conditions. Results show that these mechanisms achieve high delivery ratios with reduced overhead. In contrast with RAPID, EraMobile does not use forwarding probabilities. Usage of the number of one-hop neighbors observed around a node is similar to our way to determine density levels. RAPID uses the number of one-hop neighbors to calculate rebroadcast probability, whereas we use density level information to trigger adaptive protocol behavior. Each RAPID node needs to send a heartbeat message periodically (if it has not sent any other message in the periodic interval) in order to be aware of neighboring nodes. In contrast, our use of periodic gossip messages (broadcast to neighbors) lets nodes recover missing data packets. Furthermore, EraMobile calculates density levels with gossip and request message receipts at a node. Our solution integrates explicit buffering and gossiping mechanisms.

Overall, our solution is quite different from the approaches presented above in the following ways. EraMobile is not based on any underlying routing protocol. It disseminates and recovers data with a full gossip-based scheme. It requires neither routing nor multicast group information. Our solution exploits the broadcast nature of the wireless medium to disseminate gossip messages, instead of making information about possible gossip receivers available a priori. Furthermore, our protocol is supported by an adaptivity mechanism that reacts to changes in node densities for better performance.

3. Description of EraMobile

In this section, we describe the system model, structure, algorithms and properties of EraMobile.

3.1. System model

We consider a meso-scale to large-scale mobile ad hoc network consisting of a set of nodes denoted by $M = \{M_1, M_2, \ldots, M_N\}$ where N is the total number of nodes. Each node $M_i \in M$ has a unique host identifier and a message sent by a node M_i can be received by all nodes within M_i's wireless transmission range. Nodes are located in an area of finite size, and they may change their locations in the network (that is, they are mobile). New nodes may join, and a node may leave the system either voluntarily or as a result of a fail-stop failure, and then rejoin later. Node failures are assumed to be transient. We assume that there is no permanent network partitioning in the system.

Table 1
Abbreviations used in algorithms.

d	Data message
gm	Gossip message
rm	Request message
gc	Gossip count
st	Stability threshold
Id	Data message identifier
mgroupId	Multicast group identifier
missingId	Identifier(s) of missing *d*'(s)
myLastReceivedMid	*id* of the last received *d*
NeighborList	List of neighboring members
DataBuffer	Node's local data buffer
MissingDataBuffer	Buffer storing *missingId*'s
TempBuffer	Temporary buffer for *gm* and *rm*

3.2. Protocol structure and algorithms

Three functional units handle the operation of EraMobile. The *Data Dissemination* unit disseminates data and recovers missing data packets. The *Adaptivity* unit is responsible for setting the operating modes of a number of sub-components dynamically, based on network conditions such as node density. The *Buffer Management* unit administers the protocol's buffers and provides ordered data delivery to the application. This section describes the major parts of each protocol unit, along with the related algorithms. Abbreviations used in the algorithms are listed in Table 1.

3.2.1. Data dissemination unit

Gossip messages disseminate data, without any underlying multicast or unicast routing protocol. This unit can be divided into the following parts: *gossip digest construction and propagation*, *data generation and reception*, *gossip message reception* and *request message reception*.

3.2.1.1. Gossip digest construction and propagation. The protocol proceeds in rounds initiated by the gossip timer in which each group member broadcasts gossip messages to its neighboring nodes. The period of the gossip rounds is the *gossip interval* parameter of EraMobile.

A gossip message carries the digest of the sender's data buffer contents. In each gossip round, as shown in Algorithm 1, a node scans its data buffer and collects the IDs of the packets whose *gossip counts* (packet-specific variable incremented in each gossip round) are less than the *stability threshold* value. The stability threshold parameter determines the number of times the ID of a packet should be put into gossip messages: in other words, the number of rounds for which a packet should be kept in the data buffer.

Algorithm 1. EraMobile: Gossip round

```
algorithm executed periodically once per gossip round:
for each data message d in DataBuffer do
    if d.gc < st then
        put d.Id into TempBuffer
        d.gc = d.gc + 1
    end if
end for
if size of TempBuffer > 0 then
    compress all d.Id in TempBuffer and construct gm
    broadcast gm
end if
```

A gossip message is then propagated via broadcast to 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. The relatively small size of gossip messages and short random delays before sending gossip messages reduce the probability of collisions and packet losses that may arise from broadcasts to neighbors. Random subset of nodes (that is, one-hop neighbors) that each node gossips changes dynamically due to mobility features as well as EraMobile's adaptivity mechanism based on node density levels. Furthermore, gossip interval and probability of request sending are adjusted dynamically by the adaptivity mechanism.

For applications using EraMobile, we assume constant bit-rate multicast data generation from a source. This is why periodic gossiping, with the gossip interval determined based on the node density level, is sufficient to propagate multicast data. Otherwise, if the data traffic is not continuous, piggybacking of gossip messages on other messages could be performed as an improvement on the protocol.

The example in Fig. 1 illustrates the gossip digest construction process of a node for a single sender scenario. Note that, in the scenarios with multiple senders and multicast groups, sender and group IDs are also included in gossip messages.

3.2.1.2. Data generation and reception. A source node broadcasts a new data packet to its neighbors just once: when the packet is originated. The goal of this duplicate-free initial broadcast (that is, push model) is to increase the reliability of the protocol and the propagation speed of the data by letting more nodes get the new data packet in the gossiping stage. Then, data are distributed solely with the gossip mechanism through peer-to-peer communications. We believe that small-sized gossip broadcasts and request messages do not burden the network unduly. Furthermore, short random delays before sending gossips reduce the probability of collisions and packet losses. Note that a data packet enters the buffer of the source node and therefore is gossiped by that node as well. Data generation, reception and insertion are described in Algorithm 2.

Algorithm 2. EraMobile: Data generation, reception and insertion algorithms

```
generation of multicast data message d by the source:
DataInsertion(d)
broadcast d

upon reception of multicast data message d:
if (d.mgroupId = my mgroupId) & (d is not duplicate)
then
    DataInsertion(d)
      Deliver( )
endif

insertion of multicast data message d to the buffer:
DataInsertion(message d):
put d into DataBuffer
if d.Id < myLastReceivedMid then
    remove d.Id from MissingDataBuffer
endif

if d.Id > (myLastReceivedMid + 1) then
    put missingIds between d.Id and myLastReceivedMid
    into MissingDataBuffer
    myLastReceivedMid = d.Id
endif
if d.Id = (myLastReceivedMid + 1)
    myLastReceivedMid = d.Id
endif
```

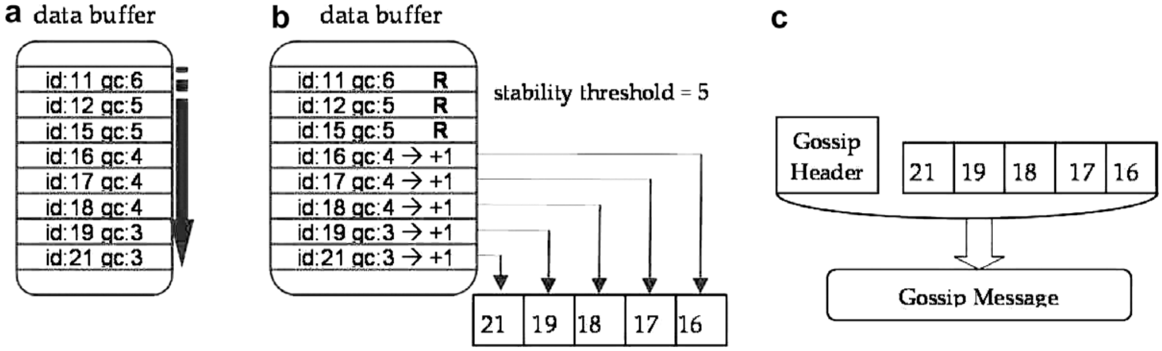

Fig. 1. Gossip digest construction: (a) The data buffer of the node is scanned to determine the status of the messages. The ID is the unique sequence number of the message, and the *gc* represents the gossip count variable, which is incremented in each gossip round. (b) The IDs of the messages whose gossip counts are less than the stability threshold (equal to 5 in this example), are put into the gossip message and their gossip counts (*gc*) are increased by one. Other messages are marked as removable (R) so they can be removed from the buffer. The missing messages (with ids 13 and 14) are declared as lost. (c) The gossip message is formed by adding the header to the gossip data.

Even if the source has few neighbors during the initial data broadcast, this would not cause a problem, since multicast data are then propagated with the gossiping mechanism. However, in order to propagate data even more quickly, the source could broadcast for the first few hops instead of one hop broadcast, if it has few neighbors.

3.2.1.3. Gossip message reception. Algorithm 3 describes the gossip reception phase. Upon receipt of a gossip message, a node compares the packet IDs placed in the gossip contents with the IDs of its missing data packets. If it notices one or more packet IDs that exist in the gossip contents but not in its data buffer, then it may request the missing packet(s) from the gossip sender. Messages carrying the missing packet ID(s) make these requests. The number of data packets that a node can request in one gossip round is limited by a parameter. Thus, a node that has missed many packets and has become aware of this situation upon receipt of a gossip message is not allowed to congest the network with a large amount of data requested. Additionally, a node cannot request a data packet that has been requested recently, for a pre-determined time in order to prevent redundant request and data transmission.

Algorithm 3. EraMobile: Gossip message reception

```
upon reception of gossip message gm from member n:
if (gm.mgroupId = my mgroupId) & (there is any d.Id in gm > myLastReceivedMid) then
    put this d.Id(s) into MissingDataBuffer
    for (each d.Id in gm = missingId in MissingDataBuffer) do
        if (requestLimit is not exceeded) & (d.Id is not recently requested) then
            put d.Id into TempBuffer
        end if
    end for
    put content of TempBuffer into rm
    send rm to n with RequestSendingProbability
end if
add n into NeighborList
```

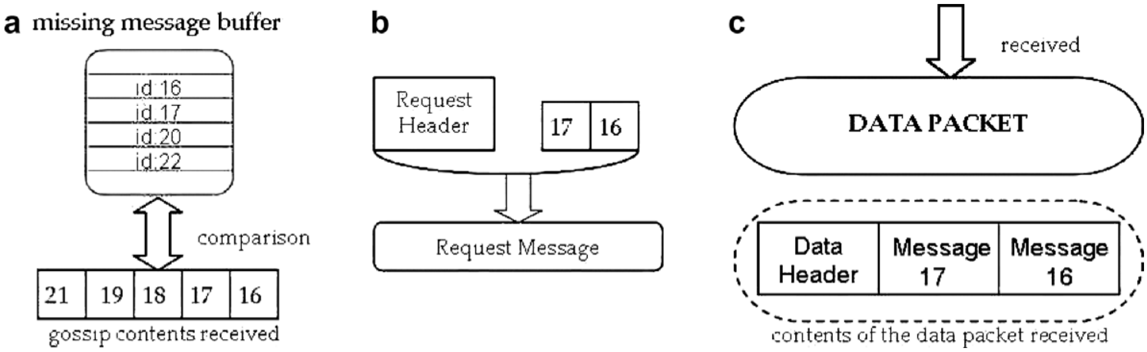

Fig. 2. Data distribution: (a) The node that has received a gossip message compares the message IDs placed in the gossip contents with the IDs of missing messages kept in the missing data buffer. (b) The node puts the IDs of the missing messages found in the received gossip contents into the request message and sends it to the gossip sender. (c) The node receives the data messages transmitted by the gossip sender.

3.2.1.4. Request message reception. A node that has received a request message may transmit the related data to the requester. These actions are described in Algorithm 4. In essence, the transmission decision is based on the number of data packets transmitted in one gossip round. A node does not answer the request messages if it has reached the transmission limit for that gossip round. Instead, different nodes can transmit messages over several gossip rounds, distributing the overhead spatially and temporally. Fig. 2 illustrates the data distribution phase.

Algorithm 4. EraMobile: Request message reception

```
upon reception of rm for data r from member n:
if (transmissionLimit is not exceeded for this round) & (r is in DataBuffer) then
    send r to n
    add n into NeighborList
end if
```

3.2.1.5. Joining and leaving a multicast group. Joining a multicast group or leaving it does not require a specific operation, since EraMobile does not maintain a structure for multicasting. We assume that multicast members initially know (e.g., through an advertising mechanism) the IDs of groups they are interested in. Nodes do not need to send or receive special messages to join or leave a multicast group. Instead, every node injects the multicast group IDs, together with the message IDs, into the gossip messages to indicate which data packets a particular gossip receiver is interested in. Gossip receivers may then ask for the missing data pertaining to the multicast group to which they belong. If a node receiving a gossip message is not in the same multicast group as the gossip sender, then it simply ignores that gossip message in favor of another multicast group.

If a node wants to leave a multicast group, then it simply ignores the gossip messages generated for that group and does not request them from the gossiper. It continues gossip broadcasting until all the data packets that belong to the multicast group it wishes to leave are removed from its data buffer.

3.2.2. Buffer management unit

This unit does tasks that are critical for the reliability and the efficiency of the protocol. It does garbage collection and data delivery in FIFO order as described next.

3.2.2.1. Garbage collection. The data buffer is maintained by means of a garbage collection mechanism that runs periodically. Algorithm 5 shows the garbage collection steps. This mechanism makes a decision about the status of every data packet based on its gossip count. If the gossip count of a packet is greater than or equal to the stability threshold then it is removed from the data buffer. In the worst case, a missing data packet that could not be recovered during *stability threshold* number of gossip rounds is declared as lost, and the protocol does not try to recover it any more.

Algorithm 5. EraMobile: Garbage collection and delivery algorithms

```
algorithm executed periodically for garbage collection:
for each data message d in DataBuffer do
    if $d.gc = st$ then
        if there is any $missingId < d.Id$ then
            declare missingId as lost
            remove missingId from MissingDataBuffer
            mark d as garbageCollectable
        end if
    end if
end for
Deliver()
remove d's marked garbageCollectable from DataBuffer
delivery of data messages to upper layer:
Deliver():
for each data message d in DataBuffer do
    if there is no gap between d.Ids then
        deliver data messages in order, to upper layer
    else
        if missingIds causing gap have been declared lost then
            deliver data messages in order, to upper layer
        end if
    end if
end for
```

The formula for the upper bound on the size of the data buffer of an EraMobile node is

$$st \times gi \times mr$$

data messages, where *st* is the stability threshold value, *gi* is the gossip interval and *mr* is the multicast data rate. For example, if *st* is 10, that means a data message stays in the data buffer for 10 gossip rounds. If the duration of the gossip round, *gi*, is 2 s, then that makes nodes buffer the data message for 20 s. If *mr* for the multicast group is 2 messages/s, then the data buffer of a node contains at most 40 data messages.

Besides the data buffer, EraMobile maintains a missing data buffer just for keeping missing packet IDs, in order to manage them easily. Upon receipt of a data packet, as described previously in Algorithm 2, if there is a gap between the sequence number of this packet and the sequence number of the last received one, then the protocol puts the ID(s) of the missing packet(s) into the missing packet buffer. This buffer is used for fast comparisons between missing packets and the list of packets in the contents of gossip messages received. The garbage collection mechanism also controls the missing packet buffer. When a packet is received or declared lost, the garbage collection mechanism removes that packet's ID from the buffer.

3.2.2.2. FIFO order delivery. The packets in the data buffer are queued in FIFO order. Upon receipt of a data packet, the buffer management unit does a delivery operation on the data buffer. The packets are delivered to the upper layer if they are in FIFO order and there is no gap between them, as specified by the Deliver( ) function given in Algorithm 5. Otherwise, they are held in the buffer until the missing packets are handled, as explained above.

3.2.3. Adaptivity unit

The mobility of the nodes in a MANET causes variable and unpredictable network conditions. The adaptivity unit lets our protocol adapt to change by adjusting the protocol parameters dynamically. The adaptivity unit considers node density (that is, the number of neighbors observed around a node) in its decision making, as shown in Algorithm 6. Note that the way density levels are defined is described in Section 5.2 and the parameter values used in performance analysis are described in Table 2.

Algorithm 6. EraMobile: Adaptivity

```
algorithm executed periodically:
calculate number of neighbors
calculate average node density
set parameters according to node density
reset NeighborList
```

Based on our analysis of protocol parameters through extensive simulation studies (explained in Section 5.1), we observed that node density affects the performance of our protocol significantly. The reason is that, since all members participate in data delivery during 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 stated previously, EraMobile does not have a separate mechanism to communicate neighbor information. Instead, it uses periodic gossip broadcasts and request messages received. A node considers the senders of both messages to be its 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*, during which node density can be calculated several times, to prevent oscillations. Then, it uses the average of the pre-calculated node density values to determine the density level. The details of density levels used for adaptive protocol behavior are described in Section 5.2.

Table 2
Values of parameters for different node density levels.

Parameters/density	Low	Normal	High
Gossip interval (s)	1.2	1.8	2.4
Stability threshold	180	150	120
Request and data transmission limits	28	16	4
Request sending probability	1	0.7	0.4
Gossip interval addition (s)	0.05	0.1	0.15
Gossip interval upper-limit (s)	4	8	12

3.3. Protocol properties

Briefly, we now explain important properties of EraMobile and the mechanisms to provide them.

- *Reliability*: A multicast protocol with reliability concerns should not be susceptible to loss of any broadcast or unicast packet. In EraMobile, the data are buffered in the nodes and advertised by gossip messages until the garbage collection mechanism of the buffer management unit removes them. A node that cannot receive the multicast data for some period of time can still recover the missing data in future gossip rounds. This feature makes the protocol robustness against transient adverse network conditions and delivery failures.
- *Scalability*: The epidemic characteristic of EraMobile can handle a growing network. As group size increases, the overhead of a single node almost remains constant, since both data dissemination and recovery of missing data are done in a fully distributed manner.
- *Fault tolerance*: EraMobile distributes the burden of data dissemination among nodes rather than leaving it completely on the source, with a peer-to-peer epidemic mechanism. Upon generation of a new multicast data packet, the source node just broadcasts the data to its one-hop neighbors. Then, all nodes participate in the data dissemination through periodic gossip rounds. Since the protocol depends only on local information and computations, this scheme avoids a single point of failure and bottlenecks.
- *Mobility friendliness*: Mobility is beneficial for EraMobile, in contrast with other multicast protocols. High mobility increases the chance that a node will encounter different nodes and this recovers missing messages. Since EraMobile nodes do not maintain a structure for multicasting, increasing mobility does not trigger any kind of control messages (such messages are required to update the multicast structure in structure-based protocols). This characteristic of EraMobile lets it operate without congesting the network by control messages in high-mobility scenarios.
- *Network friendliness*: EraMobile's gossip mechanism substantially reduces the network load by eliminating redundant data transmissions. The only control overhead that EraMobile adds to the network is that of periodic gossip broadcasts. We believe that small periodic gossip broadcasts, which only consist of some packet IDs, do not place an undue burden on the network, considering that many routing protocols in MANETs (such as ODMRP and MAODV) use beacon messages to announce network topology changes. As well, the adaptivity unit adjusts the frequency of gossip broadcasts according to the state of data propagation in the multicast group, in order to save energy and bandwidth. For example, the unit increases the period of gossip rounds when data insertion into the group slows down.

4. Experimental setup

We evaluate the performance of EraMobile in comparison with plain multicast flooding and Multicast ad hoc on-demand distance vector (MAODV) routing protocol [2]. We choose flooding because generally, it has the best packet delivery ratios, and it is commonly used in performance evaluation of multicast protocols proposed for ad hoc networks [3,4]. Therefore, comparing EraMobile with flooding gives a general idea of its packet delivery performance. We choose MAODV because it creates the lowest overhead out of the leading multicast protocols [3]. A comparison with MAODV sheds light on the performance overhead of EraMobile. We aim to explore the feasibility of our epidemic-based scheme along with an adaptivity mechanism for applications that require high reliability in MANETs under various network conditions.

The common simulation settings are as follows, unless stated otherwise. The random-waypoint model is used as the mobility pattern, and IEEE 802.11 is used as the MAC layer protocol at 2 Mbps with a wireless range of 250 m. Total simulation time is set to 1000 s. One sender begins to send data after 20 s, and it continues sending data for 870 s. The remaining 110 s are used to deliver the messages that are still in flight or in the data buffers and waiting for FIFO order delivery. The traffic rate is 2 packets of 512 bytes/s. We assume that the multicast members join to the multicast groups before data generation, in order to make a fair comparison with MAODV, in which join and leave operations add extra overhead (in contrast with EraMobile).

We implemented simulation models of EraMobile and of multicast flooding in ns-2 [42], and used the available MAODV implementation [43]. The flooding algorithm implemented for multicast and a brief overview of MAODV are described next.

4.1. Multicast flooding

The plain multicast flooding we implemented in ns-2 is given in Algorithm 7. When a node receives a new data message, it simply broadcasts it to its neighbors. Every node maintains a cache of previously received messages so that a node does not broadcast the same message twice. In order to react to the collision and contention problems in flooding, we implemented a basic collision avoidance technique. A node that has just received a new broadcast waits for a small random time interval before rebroadcasting it so that the timing of rebroadcasts is varied.

Algorithm 7. Multicast flooding

```
upon reception of a data message p:
if p.Id found in ReceivedPacketIdBuffer then
    discard p
else
    if p.mgroupId = my mgroupId then
        deliver p to application
    end if
    add p.Id into ReceivedPacketIdBuffer
    schedule p for broadcast at time
    (currentTime + random(0,jitter))
end if
```

4.2. A brief overview of MAODV

MAODV is the multicast-capable version of the AODV unicast routing protocol [39]. It multicasts 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 group connectivity by broadcasting group hello (GRPH) packets periodically. Nodes also broadcast hello packets with a time-to-live (TTL) value of 1 in order to maintain local connectivity. The group leader keeps and updates a sequence number, which is broadcast within GRPH packets. This sequence number ensures that the freshest route to the multicast groups is used.

Nodes that want to join a multicast group or to send data a multicast group without knowing a route to that group send a route request (RREQ) packet to the group leader if the node knows its address. Otherwise, the node broadcasts an RREQ packet. Both member and non-member nodes participate in broadcasting of the RREQ packets across the network and update their routing tables with information about the source node. A member of the multicast group answers the RREQ with a unicast route reply (RREP). A node may receive multiple RREP packets. In this case, it chooses the most recent and the shortest path from all the RREPs it has 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 that no other multicast members are reachable and declares itself the group leader.

5. Parameter analysis and density levels

In this section, we first analyze and discuss the main protocol parameters (gossip interval and stability threshold) through extensive simulations. After that, we describe density levels for adaptive protocol behavior in detail.

5.1. Effects of gossip interval and stability threshold

We study the effect of key EraMobile parameters in low, moderate and high node densities separately. It should be noted that these simulations exclude the adaptivity mechanism. The performance evaluation uses data packet delivery ratio and overhead metrics. The data packet *delivery ratio* is the ratio of number of data packets successfully delivered to the number of data packets generated. *Overhead* is calculated as the ratio of total control and the redundant data bytes transmitted to the original data bytes delivered. We now report the representative results that we obtained for three different average node densities. In the next subsection, we will describe comprehensively the way in which we define node density levels in the light of these results, as well as the adaptive behavior of EraMobile based on node density levels.

5.1.1. Low node density

In the low density network simulations, 10 hosts are simulated in an area of 1000 m × 1000 m with an average mobility of 5 m/s. The average number of neighbors observed around a node is nearly 2. The area of 1000 m × 1000 m is normally too large for 10 nodes to have all members of the network reachable, especially with low mobility. In such a case, nodes should keep the data for a reasonably long time in order to be able to answer request messages coming from nodes that missed some messages. Setting the stability threshold to a higher value is a direct way to increase the number of messages kept by the nodes. A greater stability threshold value lets the nodes keep more data in their buffers so that they can recover more missing messages, answering incoming requests. This is the reason for the increase in the delivery ratio with the stability threshold as shown in Fig. 3a. However, greater stability threshold values cause larger gossip messages, which imply more overhead, since gossip messages consist of message IDs in the data buffers of the nodes. Thus, the stability threshold value should be set with this trade-off in consideration.

Extending the gossip interval is another way to keep more data in the buffer. It may also help increase delivery ratio if the stability threshold value is too small to keep enough of messages in the buffer: for example, the delivery ratio rises while the gossip interval changes from 1 to 3 s for the stability threshold value of 30, as seen in Fig. 3a. This is because larger gossip intervals extend the lifetime of messages in the data buffers since their gossip counts (which increase in each gossip round) reach the stability threshold in a longer period of time to be removed from the buffer by the garbage collection mechanism. With greater stability threshold values (100 and 120 for the case in Fig. 3a), nodes keep enough data to answer requests for missing messages. Thus, extending gossip intervals cannot contribute to the delivery ratio. With longer gossip interval values, nodes send fewer gossip messages in total. This decline in the number of gossip messages reduces the amount of request and data traffic. Considering the limits for request and transmission packets, decreasing the number of transmission messages lowers the delivery ratio.

As shown in Fig. 3b, the overhead is proportional to the stability threshold. With higher stability threshold values, data messages are buffered longer, so that the number of messages in the buffers increases. Larger data buffers increase the size of gossip messages, thereby increasing network overhead. Long gossip intervals also increase the size of gossip messages. However, the overhead is reduced in long gossip intervals, since longer gossip intervals cause fewer gossip messages.

5.1.2. Moderate node density

In the moderate density network simulations, 60 hosts are simulated in the area of 1000 m × 1000 m. The average number of neighbors observed around a node is nearly 14. The node density obtained in the simulations of 60 nodes in the network area of 1000 m × 1000 m is sufficient to provide good network connectivity. Thus, the delivery ratio was observed in the range of 98% and 100% as seen in Fig. 4a. In this case, the influence of the protocol parameters on the delivery ratio was minimized. The correct values for the parameters should be determined considering performance overhead.

The overhead increases with the stability threshold, similarly to the former simulation study. However, as shown in Fig. 4b, 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 gossip broadcasts are the main source of overhead, higher node density causes more network overhead.

5.1.3. High node density

In the high node density network simulations, 120 hosts are simulated in the area of 1000 m × 1000 m. The average number of neighbors observed around a node is nearly 28. The node density may have a prohibitive effect on the delivery ratio after some point, in spite of perfect network connectivity in dense ad hoc networks. The increasing density means that increasing number of neighbors and higher network traffic load for the nodes. Such high traffic may congest the network and cause packet loss to occur more frequently. The decrease in delivery ratio compared to

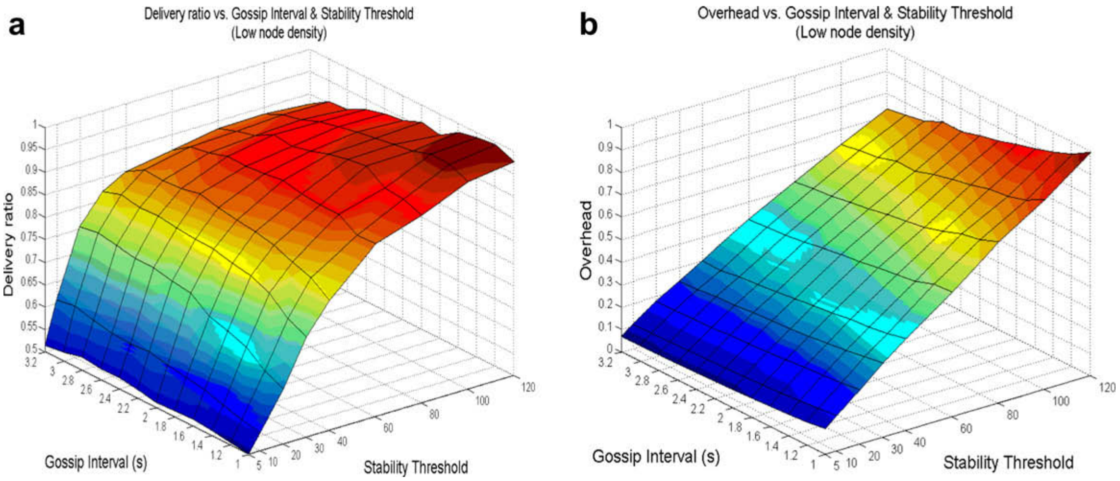

Fig. 3. The effect of the protocol parameters on (a) delivery ratio and (b) overhead in low node density.

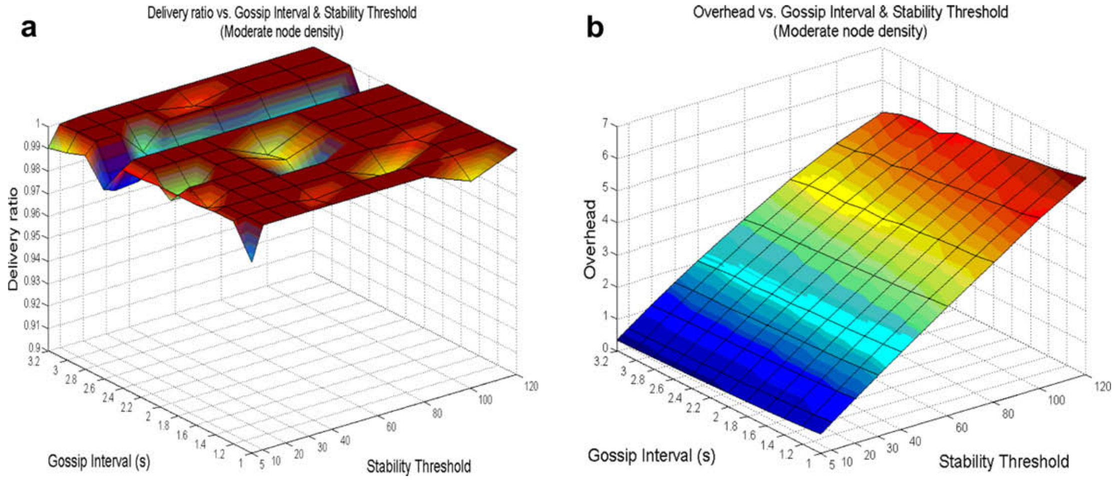

Fig. 4. The effect of the protocol parameters on (a) delivery ratio and (b) overhead in moderate node density.

the previous simulation results supports this assumption as depicted in Fig. 5a. The influence of the gossip interval on the delivery ratio becomes more apparent in the dense network simulations. The smaller gossip intervals make the nodes broadcast more gossip messages, thereby increasing the probability of congestion. The increase in congestion decreases the delivery ratio, as the figure shows. However, the delivery ratio tends to fall again in large gossip intervals like 4 s, since the number of gossip messages in such large intervals is not enough to deliver all the data. The stability threshold contributes positively to the delivery ratio, letting the nodes recover more missing messages. The delivery ratio declines, though, with higher stability threshold values such as 120, because it increases the size of gossip messages and so occupies the wireless medium longer.

The stability threshold increases overhead, as expected. However, the overhead also increases with the gossip interval as shown in Fig. 5b in contrast to the former simulation results. Additionally, the amount of overhead is less than in the case of moderate node density. The reason is that congestion decreases the probability of receiving packets, especially the gossip broadcasts. This situation causes a so-called improvement in the overhead performance of nodes exposed to congestion, since the overhead is calculated as the ratio of received bytes that do not belong to original data to delivered bytes of the original data.

As a result, the study of parameter analysis revealed that the gossip interval and stability threshold parameters strongly influence the performance of the protocol. This influence is especially apparent under adverse network conditions such as poor network connectivity and congestion. Static adjustment of the parameters may degrade the overall performance of the protocol under variable network conditions significantly. Hence, the parameter analysis confirmed the need for an adaptivity mechanism that adjusts the parameters dynamically according to variable node density levels.

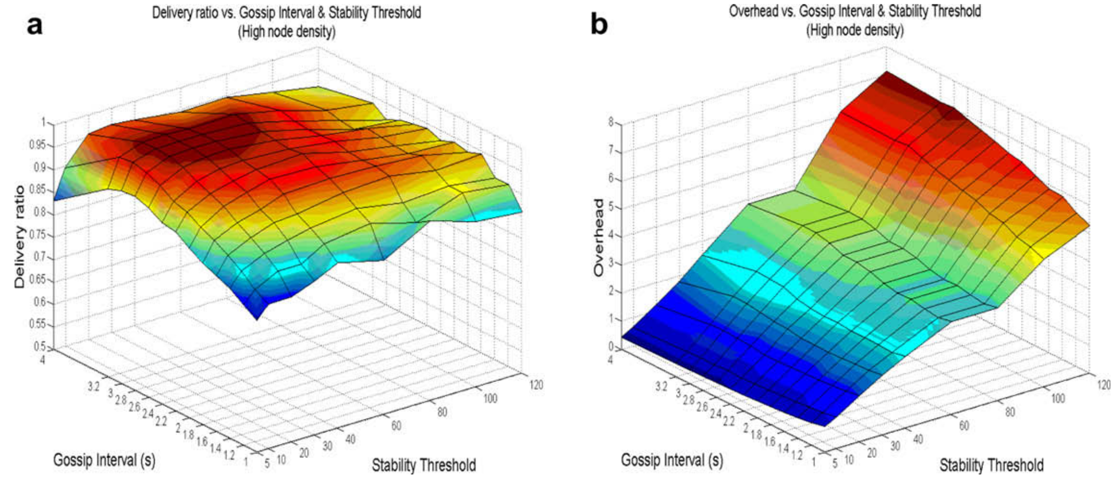

Fig. 5. The effect of the protocol parameters on (a) delivery ratio and (b) overhead in high node density.

5.2. Density levels for adaptive behavior

In the pioneering work of Kleinrock and Silvester [40], the capacity of packet radio networks consisting of nodes that are randomly located in an area has been studied. The system considered could represent a snapshot of a mobile network, or an arbitrary network. It has been shown that the optimal average node degree that maximizes throughput is approximately six: this is known as the magic number. In fact, an average degree less than the optimal six has been shown to cause a severe reduction in throughput, as well as network disconnection. On the other hand, an average degree larger than the optimum degrades performance gradually. Thus, network capacity is relatively insensitive to the case of larger degrees. The succeeding study of Takagi and Kleinrock [41] reconsiders the problem and proposes the optimal average degree (that is, the optimal number of nodes to be covered by the transmission range) as eight. Furthermore, the work of [5] considers average node degrees in the context of probabilistic broadcasting in MANETs.

Inspired by these prior studies [40,41,5] as well the results of our extensive parameter analysis reported in the previous subsection, our approach is to define three density levels: low, normal and high. In general, simulations of parameter analysis have shown that continuously adjusting the parameters under moderate node density conditions (that is, an average number of neighbors is between six and 20) does not influence overall performance considerably. The idea behind defining density levels is to keep the protocol resistant to density variations beyond the normal range. The density levels are based on the number of neighbors observed around a node. Nodes with less than six neighbors have a *low density level*. Nodes with between six and 20 neighbors have a *normal density level*. Nodes with more than 20 neighbors have a *high density level*.

Under *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 a real epidemic, the nodes should disseminate data on every contact opportunity. In MANETs, the time that two nodes will spend within the wireless range of each other varies. It may be too small for nodes to become aware of each other or to complete the data exchange. Thus, a node needs to send gossip messages more frequently in order 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. Under a *normal density level*, nodes maintain the protocol parameters at moderate values since they have suitable network conditions for communication. Under a *high density level*, even mild amounts of traffic generated by the nodes may make the network congested. Such a congested network causes many packet losses arising from collisions and ultimately wastes limited bandwidth. The adaptivity unit decreases the traffic rate of the nodes experiencing congestion in dense zones, suiting the heterogeneous environment of MANETs, thereby spreading 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 packets transmitted in a gossip round.

Extending the gossip intervals decreases the number of gossip broadcasts. Besides, it reduces the traffic of request and data packets indirectly. The request and transmission limits per round are also decreased in order to alleviate congestion. In addition to the request limit, the adaptivity unit defines a *request sending probability* that is reduced by the node density level, for request sending. The nodes use this to decide whether to send a request message. The request sending probability aims to prevent a request explosion when a gossip message (which includes the IDs of newly generated data messages) is broadcast in a dense zone.

EraMobile also has an adaptation mechanism to support energy conservation. Each node adds a small amount of time, the *gossip interval addition*, to its gossip interval value in every gossip round until it reaches a predefined upper-limit, the *gossip interval upper-limit*. Upon receipt of a data message, the extended value of the gossip interval is reset to its original value. This method aims to reduce energy consumption by eliminating redundant gossip broadcasts when there is no data traffic. The addition value and the upper-limit for the gossip intervals are determined from the node density level. In dense networks, the gossip interval is extended more quickly and the upper-limit is increased to assist the congestion control mechanism.

The values presented in Table 2 are used for the protocol parameters in the performance analysis simulations of the next section. As described in detail above, the adaptivity unit chooses the proper parameter values based on the density level observed around a node. In fact, these values of parameters for three density levels reflect an intuition that may not represent the most accurate values. As future work, the impact of the variation of the node density on the performance of the system can be analyzed in detail.

6. Performance evaluation

In order to explore the performance of EraMobile in comparison with other protocols, we use the metrics of packet delivery ratio, multicast reliability, receive-overhead, and send-overhead.

- *Data packet delivery ratio* is the ratio of number of data packets successfully delivered to the number of data packets generated. This metric indicates to what extent the protocol succeeds in delivering the originated packets.
- *Multicast reliability* is the ratio of the data packets delivered to *all* members to the number of data packets generated. That is, a data packet should be delivered to all members of the multicast group in order to be counted in this metric. It shows how successfully the protocol can deliver the data to all group members.
- *Receive-overhead* is the ratio of bytes belonging to control packets and redundant data packets received by a node to the bytes of data it has 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 in order to deliver one byte of data.

- *Send-overhead* is the ratio of bytes transmitted by a node to the bytes of data it has delivered. This metric indicates the network load generated to transmit one byte of data. However, it might be inadequate to show the actual network load in broadcast transmission, since multiple receivers would receive the broadcast packets. Thus, we examine two types of overhead and take into account both send-overhead and receive-overhead.

We investigate the effect of node mobility, node pause time, group mobility, node density, group size, number of multicast groups and senders, and traffic load on the performance of the protocols. Each simulation is repeated five times with different seed numbers and node mobility traces. The collected data are averaged over those runs.

6.1. Effect of mobility

The average speed of nodes is varied from 1 m/s (3.6 km/h) to 30 m/s (108 km/h) in steps of 5 m/s to examine the effect of mobility. The network modeled consists of 50 mobile nodes with moderate node density, where each node is set as a member of the multicast group. In moderate node density, there are enough nodes to get full network connectivity over multiple hops. Under these conditions, EraMobile achieves fully reliable packet delivery, measured by delivery ratio and multicast reliability metrics, without being affected by mobility as shown in Fig. 6a and b. Flooding also has performance close to that of EraMobile in terms of delivery ratio. However, the reliability of flooding is around 80%, since even a small number of packets missed due to link errors or collisions may degrade reliability performance significantly in the absence of a recovery mechanism. MAODV achieves a packet delivery ratio of over 95% for a mobility of 1 m/s. However, its delivery ratio decreases with mobility due to its mobility-fragile tree structure. The link breakages make it almost impossible for MAODV to achieve reliable data delivery over 10% at higher mobility values. The overhead of all protocols does not change considerably with mobility, as shown in Fig. 7a and b. The receive-overhead of flooding is higher than both MAODV and EraMobile due to the large number of duplicate data packets received through

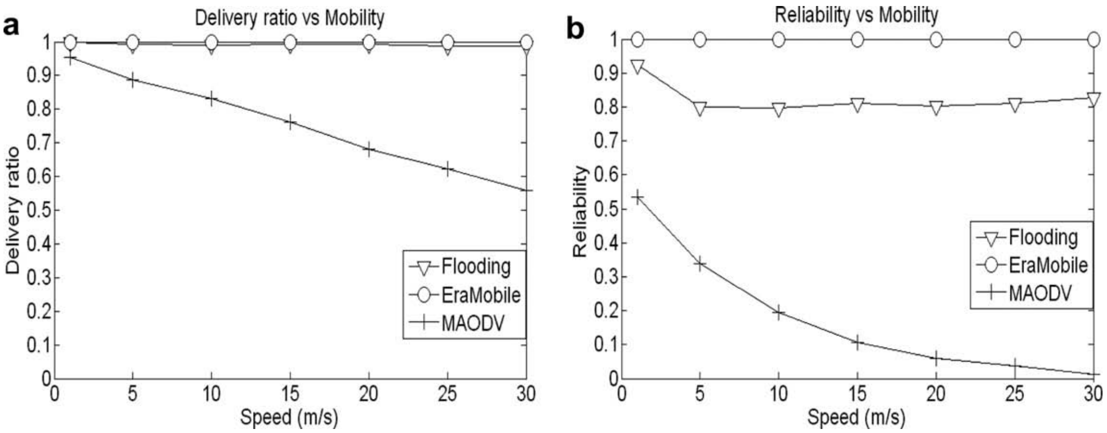

Fig. 6. Effect of mobility on (a) delivery ratio and (b) multicast reliability.

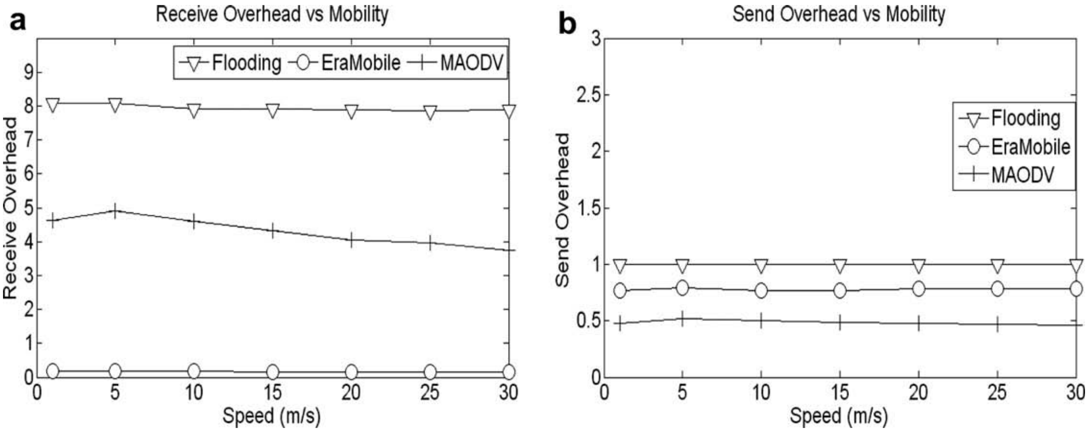

Fig. 7. Effect of mobility on (a) receive-overhead and (b) send-overhead.

broadcasts. The receive-overhead of EraMobile has the minimum value, since its gossip mechanism eliminates redundant data transmissions.

We also investigate low and high node density cases. In low density, some nodes may move to isolated areas, which are not covered by any node, and become unreachable. Throughout the unreachable state, such nodes miss the data packets disseminated in this period of time. In flooding and MAODV, nodes that missed data dissemination waves cannot recover the missing data. However, in EraMobile, nodes have the chance to recover lost data through consecutive gossip rounds. A node that is unreachable for a while, may compensate for the data it missed upon getting rid of isolation. The success probability of the recovery is dependent on the time passed while unreachable. Mobility acts as a positive factor by shortening this time. At high speeds, nodes can leave the isolated areas more quickly and enter the scope of up-to-date nodes before they remove the data messages from the buffers. Besides, higher mobility means a higher chance that a node will come across different nodes and recover missing messages.

The delivery ratio measurements of EraMobile verify this mobility-friendly characteristic of the protocol. It achieves fully reliable data delivery with increasing mobility. We observed relatively lower values (that is, 96% delivery ratio and 78% multicast reliability at the speed of 1 m/s) if the isolated nodes that recently became unreachable could not get rid of isolation till the end of the simulation. On the other hand, the packet delivery ratio of MAODV and flooding was measured at less than 60%. While the delivery ratio of flooding does not vary significantly with mobility, the MAODV delivery ratio decreases from 60% to 40% with increasing mobility. The reason of the drop in the delivery ratio of MAODV is that mobility causes link breakages on its fragile multicast tree and data cannot be delivered to the nodes beyond broken links until they are repaired. The multicast reliability performance of both flooding and MAODV is around 10% due to the isolated nodes. Under low node density, it is very likely that a node will become unreachable while others keep communication over multiple hops. In such situations, a data packet cannot be delivered to all group members and multicast reliability performance degrades. The overhead of all protocols does not change considerably with mobility. The receive-overhead of flooding is higher than that of both MAODV and EraMobile due to the large number of duplicate data packets received through broadcasts. Besides, 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. EraMobile's gossip mechanism lowers the receive-overhead substantially by eliminating redundant data transmissions. Neither the receive-overhead nor send-overhead of EraMobile is affected by mobility since EraMobile does not maintain any structure for multicast data delivery. EraMobile also achieves full reliable data delivery in high node density scenarios.

6.2. Effect of pause time and comparison with group mobility

In random waypoint mobility, as the most widely used model in simulations by the research community [49], a mobile node selects a random position (x,y) in the simulation area as a destination point and a velocity (v) from a uniformly distributed range. Then, the node starts to travel to the chosen destination point with the constant selected speed. When the node arrives to the destination point, it pauses for a specific time (*pause time*). After this time, it selects a new destination and speed, and repeats the process. In order to observe the effect of pause time of mobile nodes in the random-waypoint model, we did simulations for pause time values varying between 0 and 10 s, and three different speed values (1, 3 and 5 m/s).

The overhead of EraMobile does not change considerably with pause time, as shown in Fig. 8a for receive-over-

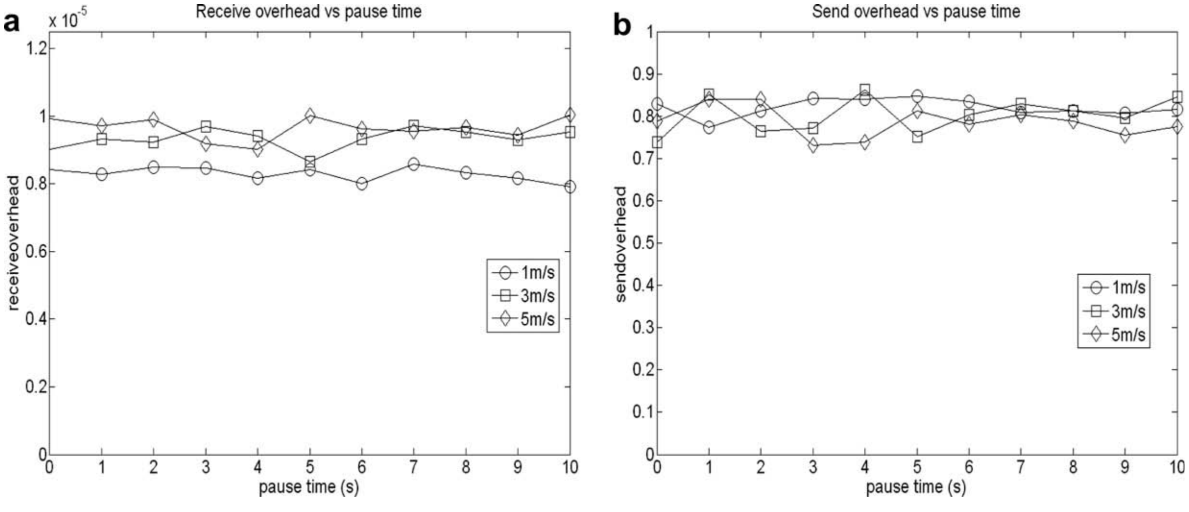

Fig. 8. Effect of node pause time on (a) receive-overhead and (b) send-overhead.

head and Fig. 8b for send-overhead. Thus, for a given speed value, overhead stays almost constant with varying pause times. This behavior is also valid across different node speed values. However, our analysis reveals some interesting findings. One is that the averages for send-overhead slightly decrease as node speed increases. For example, we measured the averages for send-overhead when the speed is 1, 3, and 5 m/s as 0.82, 0.80, and 0.78, respectively. This could be a symptom that as the node speed increases, the network load generated to transmit one byte of data by a typical node decreases independently from the pause time. On the other hand, the averages for receive-overhead slightly increase as node speed increases, and we measured the averages for receive-overhead when the speed is 1, 3, and 5 m/s as 0.0000082, 0.0000093, and 0.0000096, respectively. This could be an indication that as the node speed increases, the network load that a typical node exposes to deliver one byte of data increases independently from the pause time.

Overall, these findings indicate that although mobility helps disseminate multicast data, even in very low mobility with low speed and high pause times, the performance of EraMobile is not degraded. In the simulations, we measure the same delivery ratio and reliability levels for different pause time values.

We also compared random-waypoint (RW) model with reference point group mobility (RPGM) model. The RPGM model [50] defines the random motion of a group of nodes and the random motion of each node in the group. The group movements are based on the path traveled by a logical center. As discussed in [49], RPGM is the recommended model for group mobility. We used the BonnMotion tool [51] to generate RPGM scenarios for ns-2.

Results for RW mobility and RPGM as a function of node speed are shown for EraMobile in Fig. 9a for receive-overhead and Fig. 9b for send-overhead. As indicated earlier, both overheads in the case of RW mobility are stable across various node speeds. This behavior is also valid when nodes relocate according to the group mobility model RPGM as shown in the figure. In these simulations, EraMobile achieves fully reliable data delivery, measured by delivery ratio and multicast reliability metrics, for both mobility models. Analysis results show that send-overhead in the case of RPGM is much smaller compared to the case of RW mobility for all simulations. Thus, group mobility helps reduce the network load generated to transmit one byte of data by a typical node. On the other hand, receive-overhead in the case of RPGM is larger compared to RW mobility. This indicates that group mobility behavior increases the network load that a typical node exposes to deliver one byte of data. Further results on group mobility behavior for multiple groups are presented in Section 6.5.

6.3. Effect of node density

In order to analyze the impact of node density, the number of group members is varied from 10 to 140 in the simulation area of 1000 m × 1000 m. For these simulations, average node densities (that is, the number of neighbors around a node) observed for each group size are shown in Table 3. The node density values that EraMobile obtains through gossip and request messages were also

Table 3
Node densities obtained for each group size in the simulation area of 1000 m × 1000 m.

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

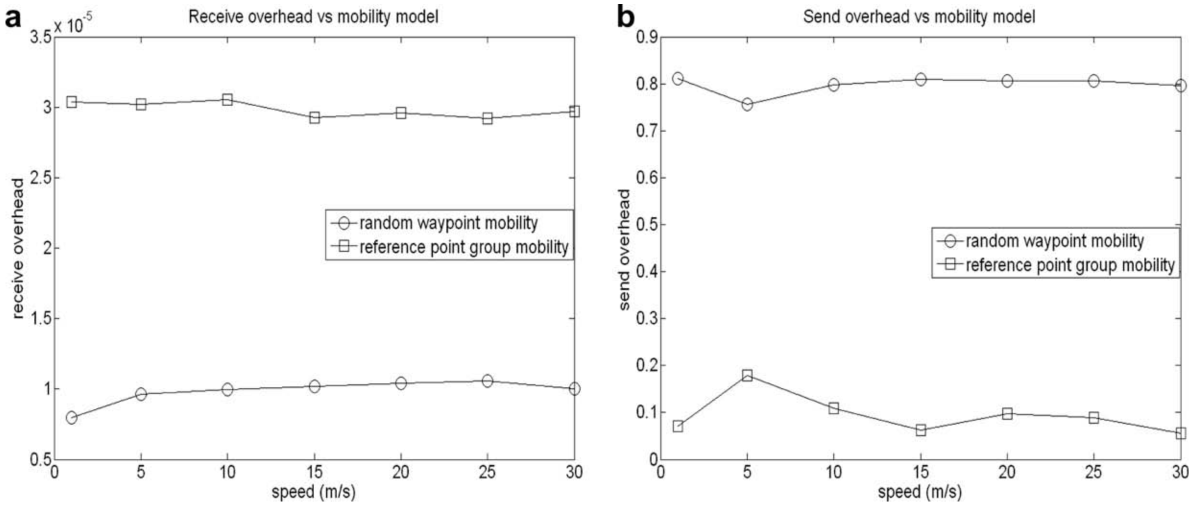

Fig. 9. Effect of mobility models on (a) receive-overhead and (b) send-overhead.

compared and found consistent with the exact values obtained from the ns-2 simulator.

For group sizes up to 50, EraMobile achieves fully reliable data delivery in all scenarios as shown in Fig. 10a and b. The delivery ratios of Flooding and MAODV increase with the number of nodes, since they have better-connected networks in higher node densities. The receive-overheads of all protocols increase with node density as seen in Fig. 11a since they all use broadcast communication in some way: for example, for data delivery in Flooding, for both data delivery and group maintenance in MAODV, and for gossip dissemination to neighbors in EraMobile. An increase in node density causes more broadcast packets to be received by nodes. This situation is the reason behind the increase in receive-overhead since both duplicate data and control packets such as link-updates and gossip messages are transmitted through broadcasting. However, the increase that EraMobile experiences in receive-overhead is negligible compared to other protocols. This result shows that EraMobile is scalable. Additionally, increasing node density causes more packet collisions, especially for broadcast packets. This fact increases the send-overhead of EraMobile slightly, as shown in Fig. 11b, since it requires retransmission of a packet that has been lost.

Simulations for group sizes between 60 nodes and 140 nodes are performed to investigate the behavior of EraMobile in extremely high node densities. The results show that EraMobile achieves fully reliable data delivery even in such dense network environments. Likewise, increase in the overhead of EraMobile as the group size scales up is negligible and almost constant. However, it needs more time to complete fully reliable delivery of data packets to all members since its adaptivity mechanism spreads overhead over time to prevent congestion and packet loss. This trade-off is further discussed in Section 6.7.

6.4. *Effect of group size*

In order to examine the scalability properties of EraMobile, we varied the number of nodes in the multicast groups from 20 to 200. When we increase the number of nodes, we also increase the simulation area proportionally while keeping the density of all areas constant. For

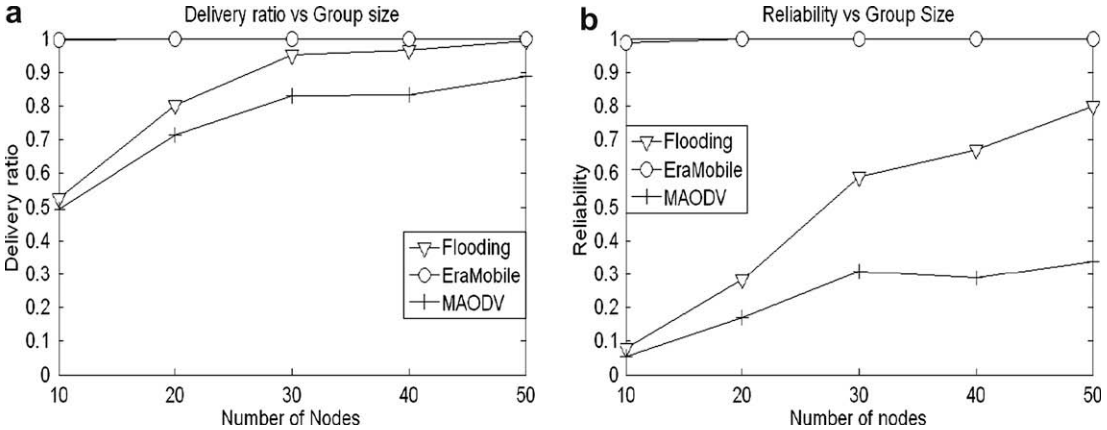

Fig. 10. Effect of node density on (a) delivery ratio and (b) multicast reliability.

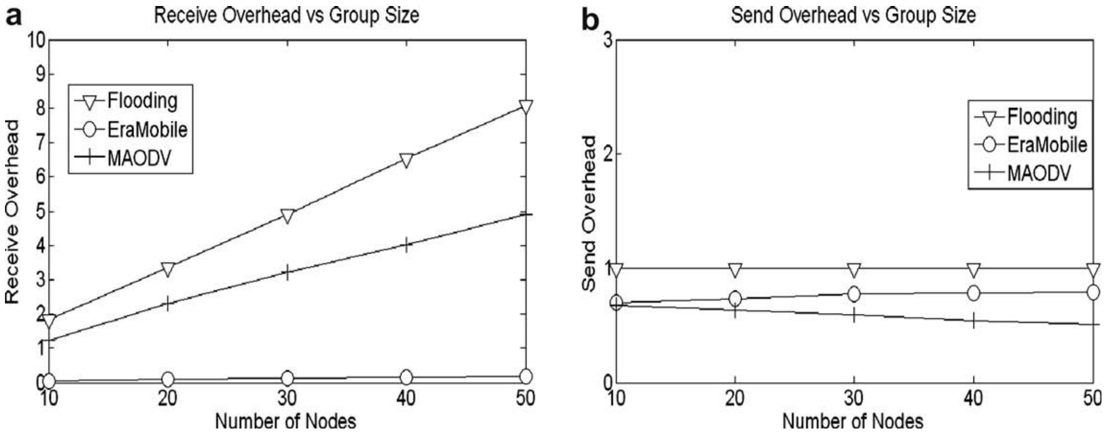

Fig. 11. Effect of node density on (a) receive-overhead and (b) send-overhead.

instance, when the number of nodes is 20, the simulation area is set to 632 m × 632 m. In the case of 200 nodes, the area is 2000 m × 2000 m. Our motivation for a constant-density network is to avoid the side effects of possible collision and contention on the shared wireless channel.

The receive-overhead of EraMobile increases slightly as a function of group size as shown in Fig. 12a. As the group size and hence the number of delivery destinations for multicast data scales up, an increase in the network load that a typical node exposes to deliver one byte of data is expected. But this increase is very low and below linear, which suggests that EraMobile is scalable in this respect. Furthermore, results for send-overhead and delivery ratio are shown in Fig. 12b. The network load generated to transmit one byte of data by a typical node proves to be scalable as the group size increases. It should be noted that, in these simulations, no data loss was observed and hence full packet delivery to group members is achieved.

6.5. Effect of multiple groups and senders

The number of multicast groups and senders is varied from 1 to 5, assigning one sender to each group. The total number of nodes in the network is fixed at 60 with a mobility of 5 m/s. MAODV performance could not be examined for multiple groups since its public ns-2 model was developed for single-group scenarios. In flooding, each node 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. 14a and b. Both the receive-overhead and send-overhead substantially increase, since nodes receive many data packets that they do not deliver to the application. Such a huge amount of overhead also lowers the delivery ratio of flooding as shown in Fig. 13a and b.

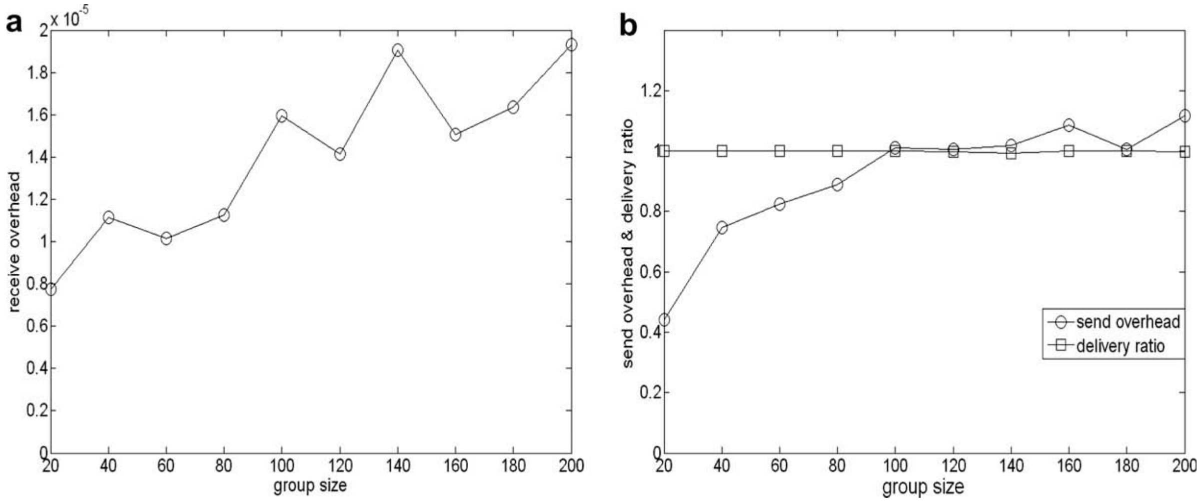

Fig. 12. Effect of group size on (a) receive-overhead (b) send-overhead and delivery ratio.

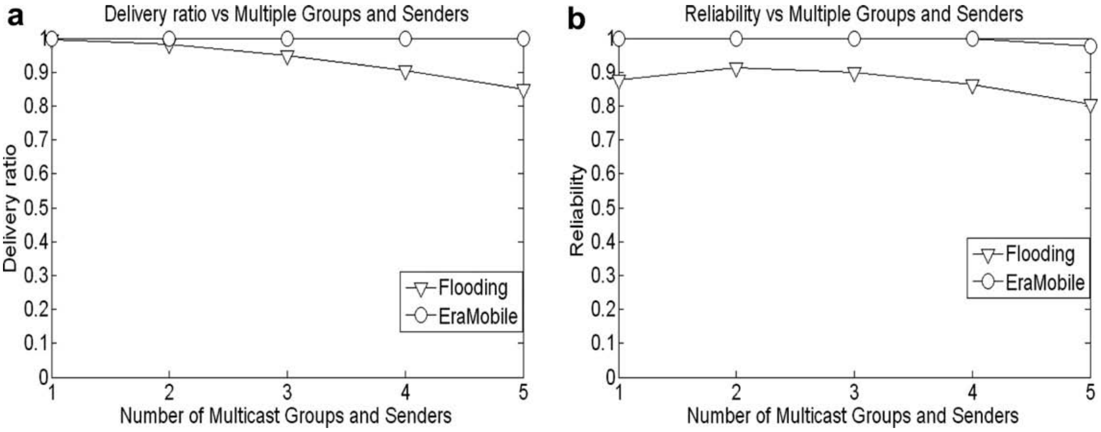

Fig. 13. Effect of multiple groups and senders on (a) delivery ratio and (b) multicast reliability.

However, EraMobile can deal with an increasing number of groups and senders by keeping the overhead at a minimum. In EraMobile, when a node receives a message from another multicast group, it just drops it after adding the size of the message to the receive-overhead. These packets coming from members of other multicast groups are the reason behind the increasing overhead of EraMobile.

We scrutinize the behavior of EraMobile further in the case of multiple groups for mobility models RW and RPGM when the node speed is 5 m/s and the network consists of 50 nodes. Different than the default settings, pause time of mobile nodes is 1 s and each sender generates 720 packets (with the rate 2 packets/s) in each simulation that lasts 500 s. Consistently with the findings we reported for mobility models as a function of node speed (in Section 6.2), the receive-overhead in the case of RPGM is also larger compared to RW mobility for multiple group scenarios. These results for the receive-overhead of EraMobile as the number of groups increases from 1 to 10 are shown in Fig. 15a. It should be noted that a node that receives and discards a message from a multicast group it is not a member of also counts it in the receive-overhead. That is the reason for the growth in receive-overhead as the number of groups increases. Overall, these results also show that group mobility behavior increases the network load that a typical node exposes to deliver one byte of data. On the other hand, the send-overhead in the case of RPGM is usually lower than in the case of RW mobility, as shown in Fig. 15b. In general, group mobility has a positive effect in reducing the network load generated to transmit one byte of data by a typical group member. Note that as the number of groups increases (with the fixed network size of 50 nodes) in these simulations, the size of each multicast group decreases proportionally. For example, when there is one multicast group its size is 50, and when number of groups is 10 each group has five members. Nodes are also randomly located in the area and each node is chosen randomly as the member of a multicast group. Send-overhead results show that RPGM has a positive effect in reducing overhead for small number of groups and larger group

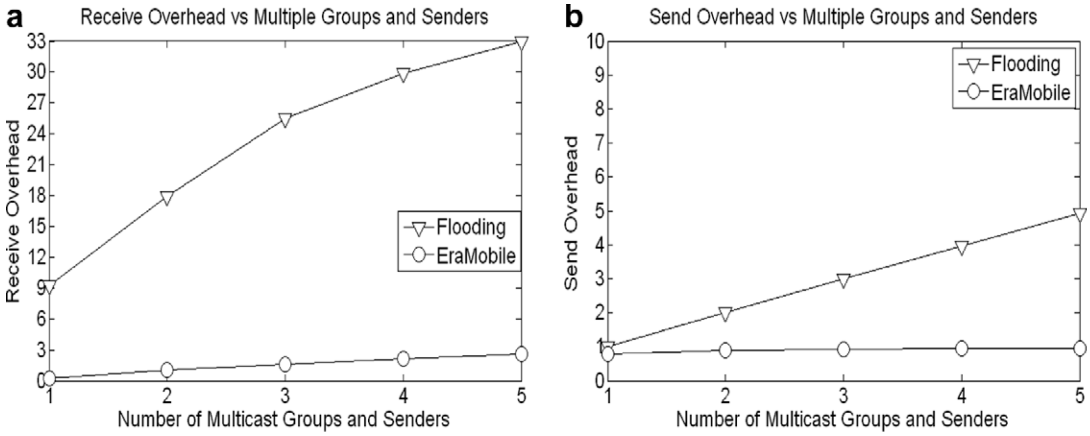

Fig. 14. Effect of multiple groups and senders on (a) receive-overhead and (b) send-overhead.

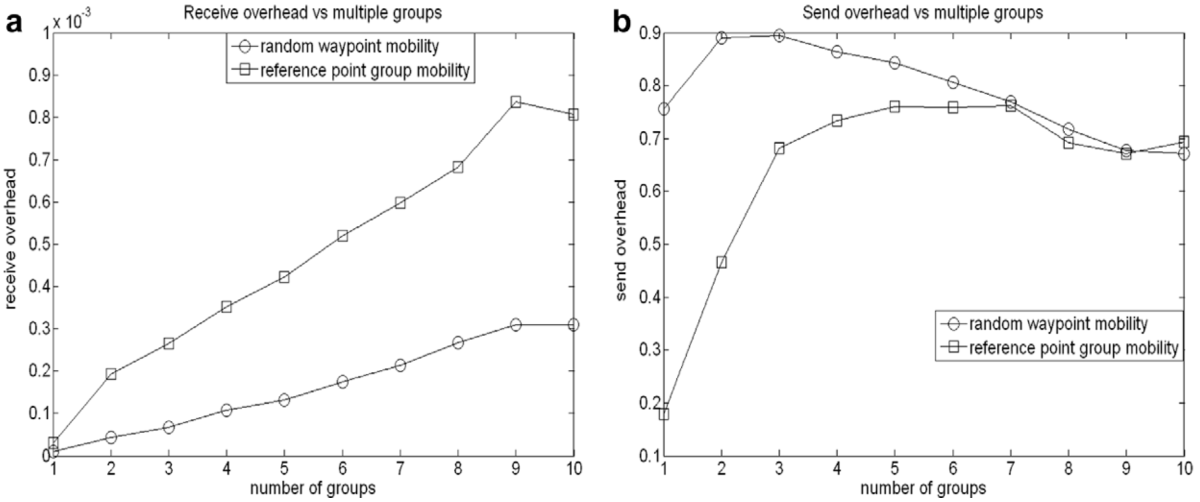

Fig. 15. Effect of multiple groups and mobility model on (a) receive-overhead and (b)send-overhead.

sizes. For RW mobility, this effect is not observed for send-overhead.

6.6. Effect of network load

We also investigate the impact of data rate. For this purpose, the number of data packets generated per second by the traffic source is varied from 1 to 16 with the payload size 512. The single multicast group consists of 20 nodes. While the traffic rate of 1 packet/s symbolizes the light traffic, the rate of 16 packets/s represents heavy traffic [44].

All protocols react similarly to increasing traffic. EraMobile achieves fully reliable delivery for all traffic rates except the heavy traffic scenario. Results for delivery ratio and multicast reliability as a function of data packet rate are shown in Fig. 16a and b, respectively. In heavy traffic scenario, the delivery ratio of the protocols tends to decrease due to the highly congested network environment. The main reason for the slight decrease in the delivery ratio 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 change considerably with the packet rate, as shown in Fig. 17a and b. 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 increases as well since a node requests and propagates more data with the rising packet rate in the higher traffic load.

6.7. Discussion

EraMobile achieves fully reliable data delivery in the dynamic environment of MANETs in several scenarios analyzed in this study. Mobility appeared to be a positive factor assisting the dissemination of data as in a real epidemic. The delivery ratio performance of MAODV at higher mobility values confirmed that structure-based

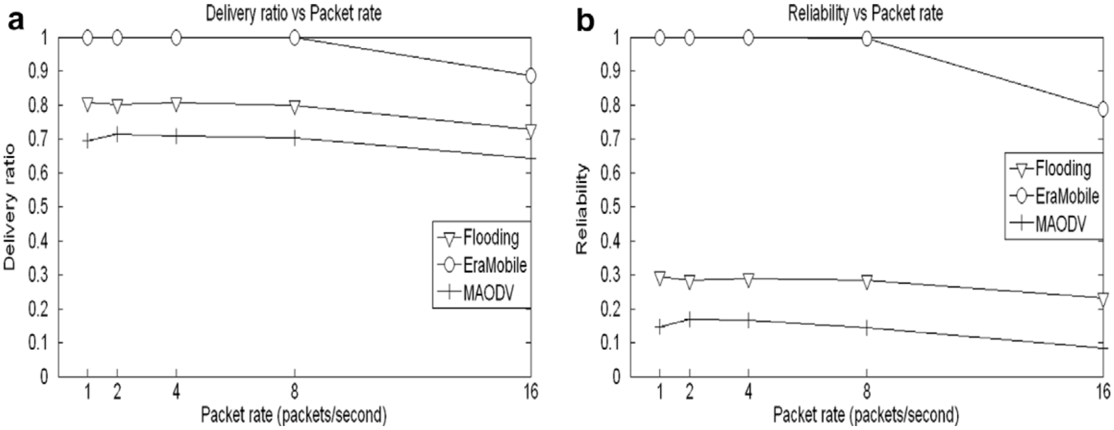

Fig. 16. Effect of traffic load on (a) delivery ratio and (b) multicast reliability.

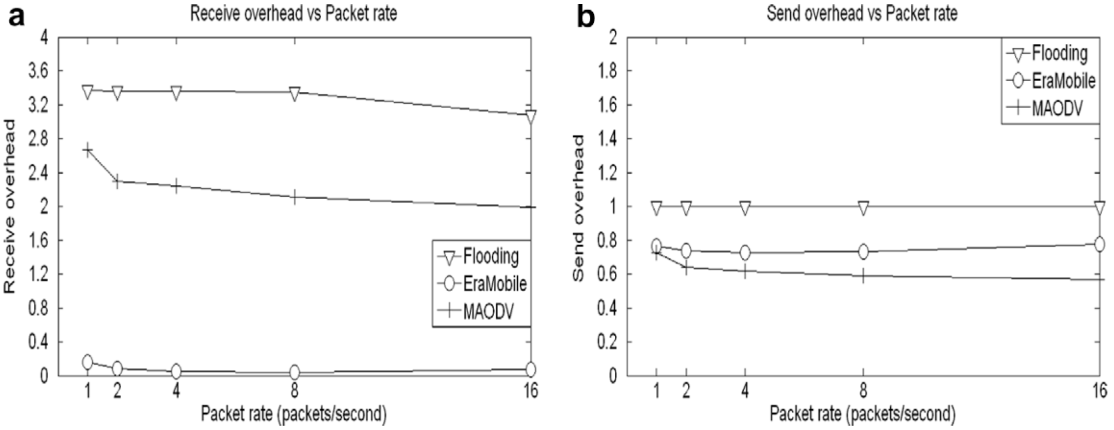

Fig. 17. Effect of traffic load on (a) receive-overhead and (b) send-overhead.

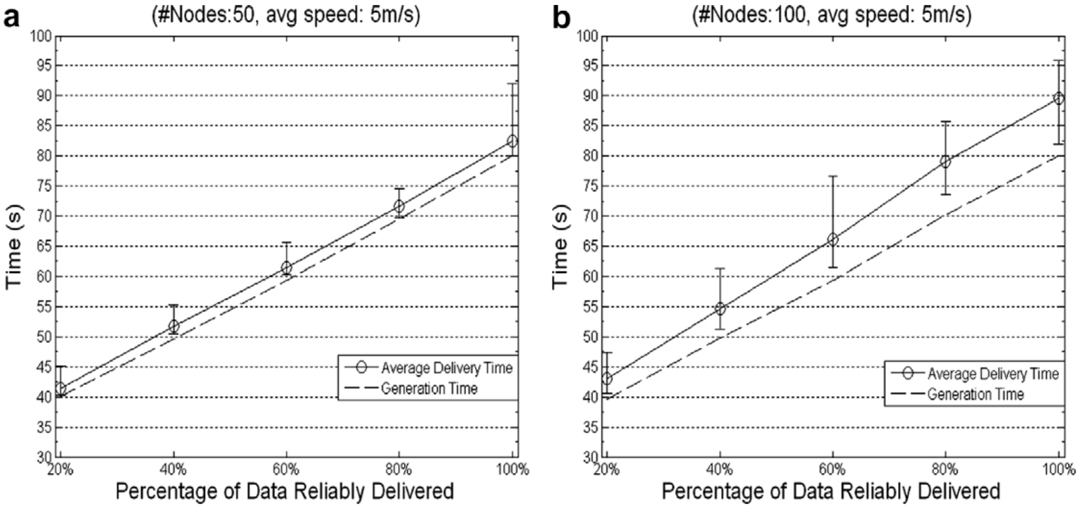

Fig. 18. Time for completion of data delivery versus percentage of data reliably delivered: (a) 50 nodes and (b) 100 nodes.

approaches cannot adapt to mobile ad hoc networks. The adaptivity mechanism of EraMobile also let it deliver data reliably in both sparse and highly dense networks without requiring any extrinsic adjustment. The delivery ratio performance of EraMobile showed slight decrease only in very high node density 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 cannot obtain fully reliable data delivery is that of removal of a data packet from the buffers of all the nodes while there are still some nodes that need it. The probability of such a situation can be lowered by increasing the stability threshold value and storage space of the mobile nodes.

Besides showing its superior delivery ratio performance, we also showed that EraMobile decreases the overhead that nodes induce. In multicast flooding and MAODV, the main source of receive-overhead is duplicate data packets. EraMobile eliminates redundant data transmissions by the gossip mechanism. The send-overhead of EraMobile remains stable in different scenarios.

The high performance of EraMobile in terms of delivery ratio, multicast reliability and overhead is partially based on a trade-off with delay. The current MANET conditions like limited bandwidth decreasing with node density [45], can hardly let most delay-sensitive group applications like real-time streaming operate properly in mobile ad hoc settings. Thus, we traded away delay for reliable and overhead-efficient multicast delivery. The time required for the completion of fully reliable data delivery is presented in Fig. 18. In these scenarios, the source node generates 2 packets/s in the time period from 30 to 80 s. 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. As shown in Fig. 18b, the higher node density with 100 nodes takes longer time due to the reduced traffic to prevent congestion.

7. Conclusions

We presented the design of EraMobile (an epidemic-based reliable multicast protocol for MANETs) and its performance evaluation as well as an adaptive mechanism as an enhancement for varying node densities. A 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 in both sparse and dense networks. Applications that are not delay-sensitive can use EraMobile to deliver multicast data reliably with minimal overhead in the highly dynamic environment of MANETs. The protocol copes with dynamic and unpredictable topology changes arising from mobility with an epidemic-based method for multicast. This method does not require the maintenance of any tree- or mesh-like structure for multicasting. It also needs neither a global nor a partial view of the network, nor information about neighboring nodes and group members. The only information that a multicast member should know is the identifier of its multicast group. The protocol substantially minimizes overhead by eliminating redundant data transmissions.

We modeled EraMobile and investigated its performance through comparative and extensive simulations on ns-2. We compared its delivery ratio achievement to that of multicast flooding and compared its overhead efficiency to that of MAODV. The performance evaluation studies have shown that the epidemic-based adaptive model has promising potential for a fully reliable multicast operation in MANETs. The mobility simulations verified the fact that its unstructured nature introduces a mobility-friendly characteristic into the protocol. The adaptivity mechanism let EraMobile achieve reliable multicast delivery with minimal overhead.

As future work, we intend to study the delay characteristic of EraMobile in conditions that vary in ways such as mobility and node density, and investigate improvements for lowering the delay in data delivery. One such improvement could be broadcasting of gossip messages immediately on reception of a data packet. Thus, neighboring nodes may receive a data packet without waiting for the next gossip round of the node. This is likely to lower the delay. However, its influence on the overall performance of EraMobile should be investigated carefully. The adaptivity mechanism should be capable of deciding in which conditions this improvement will be disabled. We also consider enhancing our protocol with a global congestion control mechanism to slow the data rate of senders when the network is highly loaded.

Acknowledgements

This study is supported in part by TUBITAK (The Scientific and Technical Research Council of Turkey) under CAREER Award Grant 104E064 and COST (European Cooperation in the field of Scientific and Technical Research) Action 279: Analysis and Design of Advanced Multiservice Networks supporting Mobility, Multimedia, and Internetworking. A preliminary version of this work appeared in [46]. We thank the anonymous reviewers for their valuable suggestions and comments.

References

[1] S.-J. Lee, Mario Gerla, C.-C. Chiang, On-demand multicast routing protocol, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), September 1999, pp. 1298–1304.

[2] E.M. Royer, C.E. Perkins, Multicast operation of the ad hoc on-demand distance vector routing protocol, in: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Seattle, WA, August 1999, pp. 207–218.

[3] K. Viswanath, K. Obraczka, G. Tsudik, Exploring mesh and tree-based multicast routing protocols for MANETs, IEEE Transactions on Mobile Computing 5 (1) (2006) 28–42.

[4] 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, March 2000, pp. 565–574.

[5] Y. Sasson, D. Cavin, A. Schiper, Probabilistic broadcast for flooding in wireless mobile ad hoc networks, in: IEEE Wireless Communications and Networking Conference (WCNC), 2003.

[6] A. Demers, D.Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, in: Proceedings of the Sixth ACM Symposium on Principles of Distributed Computing, 1987, pp. 1–12.

[7] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, Y. Minsky, Bimodal multicast, ACM Transactions on Computer Systems 17 (2) (1999) 41–88.

[8] R. Chandra, V. Ramasubramanian, K. Birman, Anonymous gossip: improving multicast reliability in mobile ad hoc networks, in: Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS), 2001, pp. 275–283.

[9] J. Luo, P.Th. Eugster, J.P. Hubaux, Route driven gossip: probabilistic reliable multicast in ad hoc networks, in: Proceedings of the INFOCOM, 2003, pp. 2229–2239.

[10] J.J. Garcia-Luna-Aceves, E.L. Madruga, The core-assisted mesh protocol, IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad hoc Networks 17(8) (1999) 1380–1394.

[11] C.W.Wu, Y.C. Tay, AMRIS: a multicast protocol for ad hoc wireless networks, in: Proceedings of the IEEE Military Communications Conference (MILCOM), November 1999, pp. 25–29.

[12] E. Bommaiah, M. Liu, A. McAuley, R. Talpade, AMRoute: adhoc multicast routing protocol, Internet-Draft (August) (1998).

[13] H. Zhou, S. Singh, Content based multicast (CBM) in ad hoc networks, in: Proceedings of the ACM/IEEE Workshop on Mobile Ad hoc Networking and Computing (MOBIHOC), Boston, MA, August 2000, pp. 51–60.

[14] J.G. Jetcheva, D.B. Johnson, Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks, in: Proceedings of the ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), October 2001, pp. 33–44.

[15] S.K.S. Gupta, P.K. Srimani, An adaptive protocol for reliable multicast in mobile multihop radio networks, in: Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, LA, February 1998, pp. 111–122.

[16] K. Obraczka, G. Tsudik, K. Viswanath, Pushing the limits of multicast in ad hoc networks, in: Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS), Phoenix, USA, 2001, pp. 719–722.

[17] S. Bunchua, 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.

[18] T. Kunz, E. Cheng, Multicasting in ad hoc networks: comparing MAODV and ODMRP, in: Proceedings of the Workshop on Ad hoc Communications, September 2001.

[19] M. Pandey, D. Zappala, A scenario-based performance evaluation of multicast routing protocols for ad hoc networks, in: Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), June 2005, pp. 31–41.

[20] P. Mohapatra, C. Gui, J. Li, Group communications in mobile ad hoc networks, Computer 37 (2) (2004) 52–59.

[21] C.M. Cordeiro, H. Gossain, D.P. Agrawal, Multicast over wireless mobile ad hoc networks: present and future directions, IEEE Network, Special Issue on Multicasting: An Enabling Technology (January/February) (2003).

[22] T. Gopalsamy, M. Singhal, D. Panda, P. Sadayappan, A reliable multicast algorithm for mobile ad hoc networks, in: 22nd IEEE International Conference on Distributed Computing Systems (ICDCS), 2002.

[23] K. Tang, K. Obraczka, S.-J. Lee, M. Gerla, Reliable adaptive lightweight multicast protocol, in: Proceedings of IEEE ICC, 2003.

[24] K. Tang, K. Obrazcka, S.-J. Lee, M. Gerla, Congestion controlled adaptive ligthweight multicast in wireless mobile ad hoc networks, in: Proceedings of IEEE ISCC, July 2002.

[25] 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, Networking (May) (2004).

[26] E. Vollset, P. Ezhilchelvan, A Survey of Reliable Broadcast Protocols for Mobile Ad hoc Networks, Technological Report, School of Computing Science, University of Newcastle, 2003.

[27] B. Ouyang, X. Hong, Y. Yi, A comparison of reliable multicast protocols for mobile ad hoc networks, in: IEEE SoutheastCon, Fort Lauderdale, FL, April 2005.

[28] L. Rizzo, L. Vicisano, RMDP: an FEC-based reliable multicast protocol for wireless environments, ACM Mobile Computing and Communications Review 2 (2) (1998) 23–31.

[29] R. Golding, K. Taylor, Group Membership in the Epidemic Style, Technical Report, UCSC-CRL-92-13, University of California at Santa Cruz, 1992.

[30] R. van Renesse, Y. Minsky, M. Hayden, A gossip-style failure detection service, in: Proceedings of Middleware, 1998, pp. 55–70.

[31] M.-J. Lin, K. Marzullo, Directional gossip: gossip in a wide area network, in: European Dependable Computing Conference, 1999, pp. 364–379.

[32] J. Luo, P. Eugster, J.P. Hubaux, Pilot: probabilistic lightweight group communication system for ad hoc networks, IEEE Transactions on Mobile Computing 3 (2) (2004) 164–179.

[33] A. Datta, S. Quarteroni, K. Aberer, Autonomous gossiping: a self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks, in: Proceedings of the International Conference on Semantics of a Networked World, 2004.

[34] A. Khelil, C. Becker, J. Tian, K. Rothermal, An epidemic model for information diffusion in MANETs, in: ACM MSWiM, September 2002.

[35] C. Lindemann, O.P. Waldhorst, Modeling epidemic information dissemination on mobile devices with finite buffers, in: ACM SIGMETRICS, 2005.

[36] A. Vahdat, D. Becker, Epidemic Routing for Partially Connected Ad-Hoc Networks, Technical Report, CS-2000-06, Duke University, April 2000.

[37] Z. Haas, J. Halpern, L. Li, Gossip-based ad hoc routing, IEEE/ACM Transactions on Networking 14 (3) (2006) 479–491.

[38] G. Wang, D. Lu, W. Jia, J. Cao, Reliable gossip-based broadcast protocol in mobile ad hoc networks, MSN, 2005.

[39] C. Perkins, Ad hoc on demand distance vector (AODV) routing, IETF Internet Draft, draft-ietf-manet-aodv-00.txt, November 1997.

[40] L. Kleinrock, J. Silvester, Optimum transmission radii for packet radio networks or why six is a magic number, in: IEEE National Telecommunications Conference, 1978.

[41] H. Takagi, L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Transactions on Communications 32 (3) (1984) 246–257.

[42] VINT Project, The UCB/LBNL/VINT network simulator-ns (Version 2), <http://www.isi.edu/nsnam/ns>.

[43] Y. Zhu, T. Kunz. MAODV Implementation for NS-2.26, Systems and Computing Engineering, Carleton University, Technical Report SCE-04-01, January 2004.

[44] T. Kunz, Reliable Multicasting in MANETs, Contractor Report, Communications Research Centre, Ottawa, Canada, July 2003.

[45] P. Gupta, P.R. Kumar, The capacity of wireless network, IEEE Transactions on Information Theory 46 (2) (2000) 388–404.

[46] Z. Genc, O. Ozkasap, EraMobile: epidemic-based reliable and adaptive multicast for MANETs, in: IEEE Wireless Communications and Networking Conference (WCNC), 2007.

[47] Y. Tseng, S. Ni, Y. Chen, J. Sheu, The broadcast storm problem in a mobile ad hoc network, Wireless Networks 8 (2/3) (2002) 153–167.

[48] V. Drabkin, R. Friedman, G. Kliot, M. Segal, RAPID: reliable probabilistic dissemination in wireless ad-hoc networks, in: 26th IEEE International Symposium on Reliable Distributed Systems (SRDS), October 2007.

[49] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Communications and Mobile Computing 2 (2002) 483–502.

[50] X. Hong, M. Gerla, G. Pei, C. Chiang, A group mobility model for ad hoc wireless networks, in: ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), August 1999.

[51] BonnMotion, A mobility scenario generation and analysis tool, <http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/>.

Oznur Ozkasap received her M.S. and Ph.D. degrees in Computer Engineering from Ege University in 1994 and 2000, respectively. From 1997 to 1999, she was a graduate research assistant at Cornell University, Department of Computer Science, where she completed her Ph.D. dissertation. Since 2000, she has been an assistant professor in the Department of Computer Engineering at Koc University. Her research interests include distributed computing systems, multicast protocols, peer-to-peer systems, bio-inspired distributed algorithms and computer networks.

Zulkuf Genc received a B.S. degree in Computer Engineering from Istanbul Technical University in 2003 and an M.S. degree in Electrical and Computer Engineering from Koc University in 2006. He is now working towards a Ph.D. in Wireless and Mobile Communications (WMC) Group at the Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS) of Delft University of Technology (TUDelft). His research interests include wireless and mobile communications.

Emre Atsan received a B.S. degree in Computer Engineering and an M.S. degree in Electrical and Computer Engineering from Koc University, in 2005 and 2007, respectively. He is currently pursuing graduate internship at EPFL. His research interests include mobile ad hoc networks and distributed systems.