GRIDROUTE: A MULTI-LAYERED GRID BASED ROUTING PROTOCOL FOR DELAY TOLERANT MOBILE NETWORKS

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Emin Yiğit Köksal
May, 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. İbrahim Körpeoğlu (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Özgür Ulusoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ezhan Karasan

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

GRIDROUTE: A MULTI-LAYERED GRID BASED ROUTING PROTOCOL FOR DELAY TOLERANT MOBILE NETWORKS

Emin Yiğit Köksal
M.S. in Computer Engineering
Supervisor: Assoc. Prof. Dr. İbrahim Körpeoğlu
May, 2012

This work proposes a new routing protocol for delay-tolerant mobile networks (DTMNs) called GridRoute. The proposed protocol can be adopted considering network requirements such as low message delay or low resource usage. GridRoute is a probabilistic routing protocol that takes advantage of mobility and location information of nodes. It uses a multi-layered grid for contact probability maximization. It requires almost no memory storage of contact or location probabilities for intelligent routing decisions. GridRoute also minimizes the number of redundant messages throughout the network with feasible delay on message delivery, and provides some security advantages like identity secrecy. Our simulation results show that GridRoute outperforms existing routing protocols in terms of memory requirement. It also achieves high delivery ratio, reasonable end-to-end delay and significantly lower message overhead.

Keywords: Routing, Delay tolerant, Delay tolerant mobile network, Intermittently connected.

ÖZET

GRIDROUTE: GECİKME TOLERANSLI AĞLAR İÇİN ÇOK KATMANLI AĞ DİZGE TABANLI YÖNLENDİRME PROTOKOLÜ

Emin Yiğit Köksal
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Assoc. Prof. Dr. İbrahim Körpeoğlu
Mayis, 2012

Bu çalışma Gecikme Toleranslı Ağlar (GTA) üzerinde yönlendirme sorununa odaklanmaktadır. GridRoute ağ elemanlarının hareket ve pozisyon bilgilerinden yaralanarak her yönlendirme adımında mesajın hedefine ulaşma şansını artıran olasılıksal bir yönledirme protokolüdür. Bu çalışmanın literatüre en büyük katkısı GTA'larda akıllı yönlendirme adımları uygulayabilmek için ağ elemanları üzerinde diğer elemanlarla alakalı hiç bir bilgi depolaması gerektirmeyen bir protokolün sunumudur. Aynı zamanda, GridRoute gereksiz mesaj trafiğini var olan protokoller arasında en aza indirgemekte, kabul edilebilir mesaj gecikme sürelerine ulaşabilmekte ve kimlik gizliliği gibi bazı güvenlik avantajları sağlamaktadır. Benzeştirim sonuçlarına göre, GridRoute var olan yönlendirme protokollerinin hepsinden hafıza gereksinimi açısından üstündür. Bununla birlikte GridRoute, yüksek ulaştırma oranlarına, başarılı gönderici-hedef arası gecikmeye ve önemli derecede az gereksiz mesaj trafiğine ulaşmıştır.

Anahtar sözcükler: Yönlendirme, Gecikme Toleranslı, Gecikme Toleranslı Mobil Ağ, Kesintili Bağlantı .

Acknowledgement

In the first place I would like to express my gratitude to my supervisor, Assoc. Prof. Dr. İbrahim Körpeoğlu, for his guidance and support in supervision of this thesis. I would like to thank espically for his patience, understanding and constant encouragement in the process. Working with him has always been a pleasure.

I would also like to thank to Prof. Dr. Özgür Ulusoy and Assoc. Prof. Dr. Ezhan Karasan for accepting to spend their valuable time for evaluating my thesis. I am thankful to academic staff of Computer Science Department of Bilkent University and Assoc. Prof. Dr. Oya Karasan whom I learned very much about computer science and engineering. I also especially would like to thank to Assist. Prof. Dr. Ali Aydin Selçuk for his remarkable contribution to security aspects of GridRoute.

I can not thank enough to my office friends, especially those in Systems and Networking group for their support and feedbacks during my master study. Their presence made this study more enjoyable.

I would like to thank to Çağla Irmak Rumelili for her encouragement and support. She has been an indispensable part of my life while I was working on this thesis.

Last but not least, I am greateful to my family for their continuous love, trust and for their sincere and endless support at every moment througout my life and providing me every opportunity they can, especially for my education. They did their best to raise me well and gave everything they had. Their encouragement made me believe that I can achieve anything. This thesis would be a dream without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This section provides a broad explanation for Delay Tolerant Mobile Networking. The application domain of DTMN is explained and common characteristics and challenges are provided in this section. Then the chapter focuses on routing aspect of DTMN and finalized with the organization of the rest of the paper.

Wireless technologies and portable devices such as PDA's and tablet computers have extended the communication opportunities. Effective wireless communication and maintenance without an existing network infrastructure is the main aim in Delay Tolerant Mobile Networks (DTMNs). Delay-Tolerant Mobile Networking (DTMN) is an approach to computer network architecture and it aims to address the technical issues like routing or energy efficiency in heterogeneous mobile networks with certain characteristics which will be explained in detail later in this thesis. Well designed protocols on DTMNs can be used for establishing communication in extreme environments. Marine life monitoring [3], [29], space explorations [13], establishing communication in rural villages [8] are just a few possible DTMN applications. However, typical features of DTMNs make it challenging to achieve effective communication in such environments.

DTMNs have relatively limited application space. Some of important applications that require communication in real time or near real time are not suitable for DTMNs. Instant messaging, multimedia streaming or connection oriented applications like SSH are some important examples to this kind of applications. However, considerable portion of applications can be used in delay tolerant architecture. E-mail or web together with file transfer applications can be named as just a few significant examples of these.

Ad-hoc and sensor networks are other important application domains for DTMNs. Any portable device with wireless communication capability can be used in a DTMN architecture for domain specific networking. People can exchange information with each other in campuses or corporates without the limitations of service providers or service costs by using these devices and DTMN routing protocols. Additionally, mobile sensor network applications can take advantage of DTMN routing protocols to increase their data delivery ratio and to decrease redundant message traffic or average delay of messages.

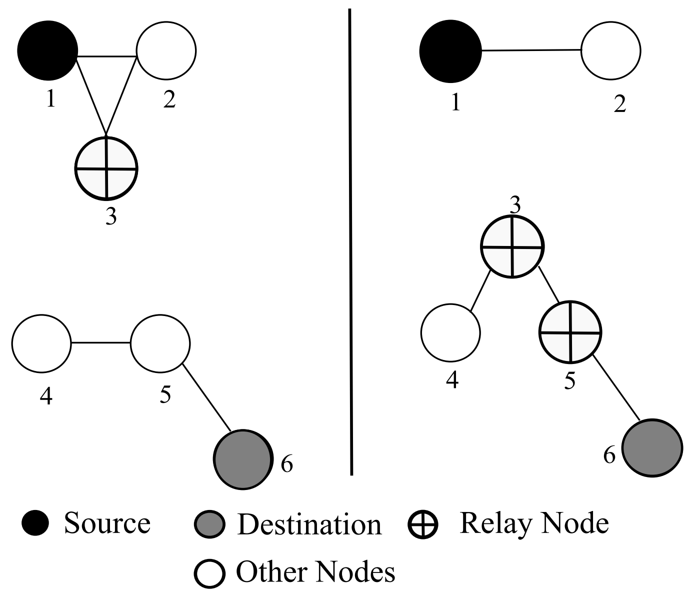

Figure 1.1: A sample DTMN message transfer on two-dimensional space.

Low node density, intermittent connectivity, lack of end-to-end path, high data loss, high latency and limited resources are common characteristics of a DTMN [9]. These limitations make almost impossible to use existing network protocols on DTMN architecture. For example, routing protocols such as AODV [25] or DSDV [24] cannot operate properly on a DTMN due to possible lack of an end-to-end path between a sender and receiver.

Routing on DTMNs is one of the most challenging topics in DTMN research. Figure 1.1 demonstrates an example message transfer in an intermittently connected environment. On the left hand side of Figure 1.1, there are two connected sub-networks. Assume that node 1 tries to send a message to node 6. Although there is not a path between node 1 to node 6 in classical sense, mobility in DTMN allows eventual message transfer between unconnected parts of the network. Assume node 1 guesses that there is a reasonable probability for node 3 to deliver the message to node 6. Node 1 sends the message to node 3 and eventually, node 3 is connected with the sub-graph of node 6 as in the right hand side of Figure 1.1. Although there is no direct communication, node 3 can forward the message to node 6 through node 5. Selecting node 3 rather than node 2 as a relay node is an important decision for a DTMN routing protocol. In order to make successful decisions, routing protocols for DTMNs try to estimate the mobility patterns and contact opportunities of hosts in order to maximize message delivery rate with high-reliability and low message overhead [17][7][18].

Contact information and clustering are two of main tools that are used in state of the art DTMN routing protocols. Broadly speaking, nodes in the network capture their contact probabilities with the other nodes and use this information in different ways. Contact information based protocols mainly forward the message to a node that has higher contact chance with the destination. On the other hand, clustering based protocols separates nodes into different groups and provide ways to carry messages in and between these clusters.

Another important information about nodes, namely the location information of nodes can be used in DTMNs to make intelligent message forwarding decisions. Most of the location information based DTMN routing protocols that are proposed so far have unrealistic assumptions or poor performance. Hence, in DTMNs this information is not fully utilized when it is compared to location information usage in mobile ad-hoc networks.

Location information is an important data that is commonly used in ad-hoc routing protocols. It is also promising for DTMNs to use in routing decisions. As it is evaluated in related work section, human mobility has a simple pattern. Mobile nodes that are able to operate according to human mobility pattern can increase the performance of DTMN routing protocols. Capturing the human mobility information can be challenging in DTMNs in which nodes may have low resources. A DTMN routing protocol that operates based on human mobility must consider this in order to provide an efficient and usable protocol.

In this paper, GridRoute routing protocol is presented to attack routing problem in DTMNs. Rather than focusing on contact probabilities or clustering, spatial information, i.e., locations of nodes are used to maximize the delivery probability of a message to its destination. This approach of GridRoute enables nodes to make intelligent routing decisions without storing any information of the other agents in the network. GridRoute also minimizes number of redundant messages throughout the network with feasible delay on message delivery, and provides some security advantages like identity secrecy. Our simulation results show that GridRoute outperforms existing routing protocols in terms of memory requirement. It also achieves high delivery ratio, reasonable end-to-end delay and significantly lower message overhead.

Rest of the thesis is organized as follows. Section II gives information about some related work and in Section III, GridRoute is presented. Simulation setups are described in Section IV, and simulation results are presented and discussed in Section V. Finally, the thesis is concluded in Section VI.

Chapter 2

Related Work

This section explains some of the important and efficient routing protocols that exist in the literature. Apart from explaining the protocols, analysis on their performance are also provided with short evalutions and reasoning. In this section, also the protocols are classified according to some parameters that are indicated below.

2.1 Epidemic Routing

Epidemic Routing [35] [36] is one of the first routing protocols that is designed for DTMNs. In this protocol nodes store messages in their buffers. A message is forwarded to all of the nodes in the communication range, if that particular message is not in the buffer of the receiver already. After forwarding the messages, nodes still keep those messages in their buffers. This is valid until the message is overwritten by some other message; messages are stored in buffers in round-robin fashion, and try to find other nodes that have not received that message up to that time. No knowledge of network or mobility of the nodes are required so there is no maintenance or synchronization requirement. However, spreading the message to all possible receivers requires lots of buffer space and redundant message transmission. In order to overcome these problems, time to live (TTL) info is attached to each message and messages are forwarded until the TTL limit. However, even this limit cannot reduce the redundant message traffic and excessive resource requirement as discussed in Section V.

Epidemic Routing is based on opportunistic contacts. As discussed in [17], opportunistic based protocols rely on eventual contact of two nodes without any prior estimation on node contacts or success probability of message delivery. There are two other main contact categories in DTMNs, namely predicted and scheduled contacts. Prediction based protocols such as [1] estimate the future contacts of nodes based on the previous information on the network. On the other hand, scheduled based protocols like [12] have the exact contact information of the nodes for the future. However, this type of protocols have limited application space as exact information on the future contacts is not always possible. Most of the protocols and GridRoute use predictive contact information for routing decisions.

2.2 PROPHET

PROPHET [18] is an important DTMN routing protocol that uses predicted contact probabilities. PROPHET continuously sends the message to nodes that have more contact probability with the destination. The nodes in PROPHET hold local contact information for all other nodes in the network. This feature of PROPHET causes a total of $O(N^2)$ contact history memory requirement network-wise where N is the node count in the network. If hundreds of low capacity nodes are used in the network, PROPHET may not scale well due to this memory obligation. Whenever a message is generated in PROPHET, broadly speaking, nodes ask their neighbors about their contact probability with the destination. If a node has higher contact probability and the message has enough replication limit, the limit is decreased and the message is forwarded to that node. Note that multiple copies of the same message exist in a network that uses PROPHET. This feature wastes buffer spaces and causes redundant message traffic. However PROPHET achieves high data delivery and relatively low redundant message traffic when it is compared with Epidemic Routing.

2.3 Scheduled Contact Based Protocols

Scheduled contact based DTMN routing protocols like [12] work in a relatively easier domain when they are compared with other two contact type based protocols. However, the global schedule may not always be available for all nodes in the network. The main challenge in this type of network is to optimize the message delivery using existing information. Generally there is no message replication and naturally there is no contact estimation but calculating the optimal path based on local or global information may be computationally expensive. As a result, these kind of DTMN routing protocols focus on reducing computational complexity of their routing algorithms and aim to provide heuristics to reduce the complexity while achieving the most optimal routing path that is possible. DTMNs in space and inter-satellite communication are some important application domains of such protocols. Also these types of protocols can be used in DTMNs with predicted mobility like in [22].

In [22], the trajectory of nodes can be calculated with a deterministic function of time. Using this information, nodes are able construct a graph using location and time information. The links on the graph connect different nodes in different times so paths for message delivery can be calculated by taking the message transmission opportunities of timely links into consideration. [22] minimizes the message delivery time by applying a distance function on the links of the graph. In this work no contact prediction or mobility patterns estimation are used but rather the existed periodic connection information is used to deliver the messages. In lots of cases this periodicity in the network does not appear.

2.4 Clustering Approach in DTMNs

Clustering is another popular technique for routing in DTMNs. Contact histories of users or their location information are used to produce clusters or communities. First an inter-cluster routing protocol delivers the message to the appropriate cluster and intra-cluster routing leads the message to the receiver. Various information like mobility patterns or contact information in the network can be used to divide the nodes into clusters. According to information that is used, people that have close similarity metrics are grouped in a cluster. However in most of the cases a node in the network must have all of the other nodes' cluster information. This feature requires periodic information transfer between nodes with broadcast-like message transmissions. Effective broadcast procedure in DTMNs is a challenging problem on its own. Apart from complicating the routing process, using clustering technique increases the total message traffic in the network. [19] and [7] are important examples of such protocols. However, synchronizing and updating the cluster information bring huge overhead to the network.

2.5 Message Replication in DTMNs

2.5.1 Multi-Copy Approaches

Replication strategy of a DTMN routing protocol is another important classification metric. Both single-copy and multi-copy routing approaches are common in the literature. Epidemic Routing can be considered as an extreme case of multi-copy DTMN routing where a message is replicated to all encountered nodes. In general multi-copy DTMN routing protocols use more conservative replication techniques to reduce the message overhead. PROPHET is a good example for such protocols. In PROPHET a message is replicated only if the candidate receiver has higher probability to deliver the message to the destination.

Spray and Wait [32] is another important example of conservative multi-copy DTMN routing protocols. In [32] nodes replicates the messages up to a given threshold. If this threshold is n for a given message, this message is in spray stage and a node transfer the message to a neighbor with threshold of $(n/2)$ and halves the threshold of its own message. Once the threshold is 1, that particular message is moved to wait state and awaits for the encounter with the destination. Conservative replication as opposed to epidemic like message forwarding significantly reduces the message traffic in the network.

2.5.2 Single Copy Approaches

Even if the conservative approaches reduces the message overhead, multi-copy routing algorithms cause considerable unnecessary message transmissions. Although this feature increases the message delivery ratio, multi-copy protocols are not very suitable to networks that have nodes with limited power source, limited buffer space or high message generation rate. To overcome these restrictions, several single-copy DTMN routing protocols are proposed with acceptable delivery ratios. In [31] authors present a number of different single-copy routing protocols that can be used in DTMNs. At the base case, a message is transferred with direct transmission. In other words, a node that generates a message holds the message in its buffer until it contacts with the destination node. Not surprisingly, this approach has a low delivery ratio and high message delays.

Authors improve the single-copy routing by presenting a randomized forwarding strategy. In this case a node forwards the message to a neighbor with some probability and deletes the message. Utility based message forwarding is another routing protocol that is suggested in [31]. In this case location and last encounter time are used by each node to calculate a utility function for all destinations of messages that a node has in its buffer. If a node's neighbor has higher utility value to deliver a message, then that message is forwarded to the neighbor and the message holder deletes it from its buffer. Single-copy routing algorithms achieve low redundant message traffic but most of them have relatively low message delivery ratio when they are compared with multi-copy routing algorithms. In this work a single copy routing protocol that achieves similar message delivery ratio with multi-copy techniques will be presented.

2.6 Graph Based DTMN Routing Protocols

Producing a graph based on network information is another important method in DTMN routing protocols. In this case, contact, mobility or location data together with time are used to construct a connectivity graph among nodes. Note that the difference of this technique with scheduled contact based routing protocols like [22] is that, scheduled contact based protocols use the existing information and generally do not collect any data from the network. However graph based protocols continuously collect data and construct and update the graph accordingly. These are actually predicated contact based routing algorithms that use graphs as data structures. Using graph allows using existed shortest path or some other existed routing algorithms on DTMNs.

In this type of protocols constructing the graph and modifying the desired classical routing protocol accordingly are the main challenges. Most of these type of protocols modify existing algorithms as routing is extensively studied for classical networks in literature and as they work effectively. Constructing the connectivity graph in classical sense in DTMNs is not very effective. Low node density in DTMNs results in lots of subnetworks that are unconnected with each other. In such a case routing is impossible with classical graph based routing algorithms. To overcome this problem, generally connectivity graph is constructed using time information. All possible estimated contacts are represented by edges that are augmented with time information. Adding the time information allows constructing a connected graph and based on constructed graph DTMN routing protocols try to find the best path between sender and receiver.

[21] is a good example for this kind of protocols. In this work connectivity graphs are generated based on scheduled and scheduled periodic connections. Also the contact duration is added to the graph in order to extend the efficiency. Different from scheduled based contacts, the graphs can be updated when there is a change in the mobility of the nodes. On these graphs modified shortest path algorithm of Dijkstra [6] is executed and a path that provides shortest delivery time or shortest path is selected as the route to destination. This method works very effectively but the network must contain periodic connections in order to apply the method.

2.7 Location Information in DTMNs

Apart from contact probabilities, location information of nodes is a promising data that can be used in routing decisions for DTMNs. Some studies like [17] use position information to construct the mobility pattern of nodes but not directly operate based on locations. Just a few protocols like in [37] and [34] directly use location information on DTMNs but they include unrealistic assumptions about the network like assuming the social sets of people (like friends or family) are known globally by all nodes in the network. Moreover these approaches can not achieve efficient communication in DTMNs. As a result using location information has a high potential to increase the efficiency of routing protocols in DTMNs.

Studies such as [5], [20] show that people in communities like campuses or corporations [2] have simple and similar mobility patterns. These works indicate that human mobility can be captured by power law distribution almost perfectly. This means that people visit just a few locations very frequently and spend lots of time in these locations. On the other hand, majority of the locations are visited a few times and for short durations.

In [10] it is observed that most of the people spend more than 60% of their time in a single location. The information of human locations can be very useful for DTMN routing protocols. Predicted locations of nodes in network can be used for eventual delivery of message to the destination.

Using geographical information is common in a lot of network protocols. [23] uses Cartesian space for localization of Internet hosts and [16] uses a multi-layered grid for content management in mobile ad-hoc network. In [16] nodes with GPS devices collect the location data and replicate the content among grids to reduce the latency. Similar approach with [16] is used in this work but GridRoute uses location information to forward messages to predefined grid cells in fastest possible way with minimal redundancy rather than focusing on distribution and replication of contents to desired network locations or maintaining the contents.

Chapter 3

GridRoute

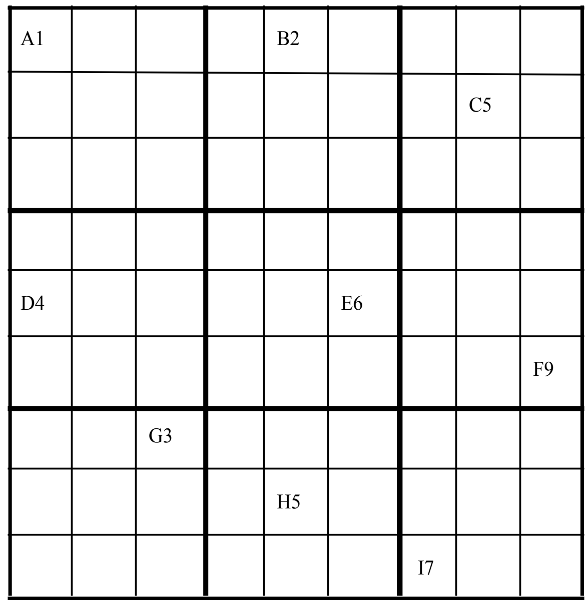

Figure 3.1: Sample multi-layered grid that GridRoute operates on.

In this section the proposed protocols are presented. 5 different DTMN routing protocol is suggested by considering some different network requirements. Detailed analysis and explanations are provided with pros. and cons. of the suggested protocols. The suggested protocols are clerified with psuedo-codes on some parts of the suggested protocols.

3.1 GridRoute

The first requirement of GridRoute is to divide the network area into a multi-layered grid. Decisions on the grid size and number of layers can be made in several ways. One possible way is to use a linear programming optimization technique as described in [16]. Another possible way to construct the grid is to set the diagonal of the lowest level cell to the communication range and triple or double the diameter for higher levels of grid recursively until less than 10 grid cells remain. Figure 3.1 is an example for a 2-layered grid. Tick lines represent the Layer 0 grid and smaller squares are in the level 1 grid. There are 9 layer 0

Symbol	Definition
σ	Current node that executes GridRoute
α	Exponentially Weighted Moving Average constant
N	Set of nodes
C	Communication range
M	Set of messages in buffer
G	Multi-layered grid
G^i	Set of grid-cells on layer i
G^i_n	Current grid-cell of node n on layer i
F_n	Favorite grid-cell of node n
L	Total number of grid layers
R_m	Receiver of message m
S_m	Sender of message m
D^i_m	Destination grid-cell of message m on layer i
P	Visiting probability
$P^{g^i}_n$	Node n's visiting probability of grid-cell g on layer i
P_{ij}	Node i's visiting probability of grid-cell j
$P^{g^i_t}_n$	Node n's visiting probability of grid-cell g on layer i at period t
K	Total number of periods in GridRoute_TA
CS^g_{ij}	Cosine similarity of existing probabilities of node i and j on grid-cell g
$RSSI_n$	Received Signal Strength Indicator from node n

Table 3.1: Definitions of symbols that are used in this work.

grid cells from A to I, and each of these has 9 sub-cells from 1 to 9.

In GridRoute it is assumed that each node in the network has a GPS module to calculate its location in the Cartesian space. It is also possible to use

localization algorithms or manual setup of loation information. A method for nodes without GPS devices will be explained later in this section.

All nodes are responsible for maintaining their visiting probabilities for each cell in each layer of the grid. Algorithm 2 provides the actual process of grid-cell probability calculation. As described in [7], Exponentially Weighted Moving Average (EWMA) is a simple and effective method to calculate and update the probabilities. It is also proved in [7] that this method produces the real probability values in the long run, regardless of the value of the α. In every sampling period nodes update their grid cell probabilities for each layer (P_{ij}, node i's visiting probability of grid-cell j) as follows:

$$P_{ij} = \begin{cases} (1-\alpha)P_{ij} + \alpha & \text{if } i \text{ is in cell } j \\ (1-\alpha)P_{ij} & \text{Otherwise} \end{cases} \tag{3.1}$$

It is proved in [7] that (3.1) produces real probability values which can be proved as follows:

Theorem 1. *If nodes i has probability of ξ_{ij} to be in grid-cell j in each time slot, EWMA yields P_{ij}, whose mean conerges to ξ_{ij}.*

Proof. Consider a sequence of time slots and let $P_{ij}(t)$ denote P_{ij} in time t. Clearly, the mean of $P_{ij}(1)$ is

$$E(P_{ij}(1)) = (1-\alpha)P_{ij} + \xi_{ij}\alpha.$$

Similarly, we have

$$E(P_{ij}(2)) = (1-\alpha)^2 P_{ij}(0) + \xi_{ij}\alpha[1 + (1-\alpha)],$$

and

$$E(P_{ij}(t)) = (1-\alpha)^t P_{ij}(0) + \xi_{ij}\alpha[1 + (1-\alpha) + \cdots + (1-\alpha)^{t-1}].$$

Let $t \to \infty$, it is arrived at

$$\lim_{t\to\infty} E(P_{ij}(t)) = \alpha\,\xi_{ij}\frac{1}{\alpha} = \xi_{ij}$$

$\square$

For simplicity, assume that all nodes maintain their grid probabilities, and they network-wide broadcast only their most likely grid cells for each layer of grid. For example, if node n spends most of its time in H5 in Figure 3.1, it only broadcasts H5 as its favorite grid and does not send any information about other grid probabilities. Note that a node can retrieve the favorite grid-cells of higher grid layers from the favorite grid-cell of the lowest layer grid. A simple encoding scheme is enough to retrieve this data. For example, a node can understand from the information H5 that the higher layer favorite grid-cell of this node is H. Later, GridRoute_NM, which does not require this information will be explained.

Whenever a message m is generated by node n, GridRoute proceeds as follows. As probability information is broadcasted, node n knows the most likely grid cells of destination. Hence, destination of message is set hierarchically for each layer of the grid. First, message m is stored at the buffer (FIFO queue; see Algorithm 4). Each node in the network runs forwarding operation periodically. The actual time period depends on network parameters like mobility and node density, however, as described in [38] increasing transmission or receiving operations does not increase the overall energy consumption significantly. Hence, a small interval like 10 seconds between periodic forwarding operations is enough to capture promising nodes that are in the communication range to forward the message for regular human mobility. This interval can be increased for highly mobile networks and vice versa for networks with low mobility.

In each periodic forwarding operation, the node n checks for other nodes in the communication range. Unique message IDs and destination information of the messages in the buffer are exchanged with the nodes in the communication range. Assume node n at A1 in Figure 2 exchanges this information with node k in A2 (right neighbor of A1). Further assume that destination of m is E6. As A2 is not in Grid E, it will compare its probability of being on grid E with the probability of n. If node k has higher probability, it requests the message from node n, and node n deletes the message after it sends the message to k if the single copy option is preferred. When message m reaches to grid E, then the probability of being in E6 will be compared between nodes and message will be relayed if one has higher probability. Algorithm 1 provides a pseudo-implementation of the forwarding operation. This process increases the contact probability of message with the destination as message is sent to nodes that spend more and more of their time in the favorite place of destination.

The complexity of GridRoute depends on four parameters, namely M, N, G and L. As it can be seen in Algorithm 1, forward operation requires checking each message for each node in communication range, and it has to find the correct grid layer for each combination of former two parameters. At the worst case it takes $O(|N| * |M| * |L|)$ time to complete forward operation for all messages in the buffer. The time requirement of updating grid-cell probabilities only depends on the total number of grid-cells linearly. More formally, it takes $O(\sum_{i=0}^{L} |G^L|)$ time. Finally, send and receive operations costs $O(1)$ time for operation on single message. Holistically, GridRoute requires $O(|N| * |M| * |L|)$ time, as forwarding is the most costly operation.

Parameters N, M and G effect the memory requirement of GridRoute. A node in GridRoute needs to maintain its own grid-cell probabilities, which requires a total of $O(\sum_{i=0}^{L} |G^L|)$ memory space. Furthermore, it has to be able to store $O(|M|)$ messages and $O(L * (|N| - 1))$ favorite grid-cell positions of other nodes on the network. However, with a proper encoding scheme, favorite grid-cells of each layer can be retrieved from the lower favorite grid-cell information, so it is enough to hold only lowest-layer favorite grid-cell data. This property reduces the memory requirement of favorite grid-cell positions to $O(|N| - 1)$. Thus, in GridRoute a node needs $O(\sum_{i=0}^{L} |G^L|) + |M| + |N| - 1)$ memory space, which results in $O(|N| * ((\sum_{i=0}^{L} |G^L|) + |M| + |N| - 1))$ memory in the whole network. In the next section, GridRoute_NM, which does not require to store $O(|N| - 1)$ favorite grid-cell positions, will be explained.

The total number of message traffic in the whole network that is generated by GridRoute is related with the parameters N, K, T and H. In the worst case, all messages will be delivered using the whole hop limit, and no messages will be overwritten from the buffers. Depending on the message generation rate $O(|N| * K)$ messages will be generated in each second in the network. Each of these messages can be forwarded H times so a total of $O(|N| * K * H)$ message traffic may be required for the messages that are generated in one second. Assuming that network will be up for T amount of time, during the life-time of the network $O(|N| * K * H * T)$ message traffic can be generated by the GridRoute in the worst case.

Algorithm 1 GridRoute Forward

procedure FORWARD
  for all n in N st. $Distance(\sigma, n) \leq C$ **do**
    for all m in M of σ **do**
      if $R_m = n$ **then**
        Forward m to n
        Delete m
      else if $HL_m \geq 0$ **then**
        $i \leftarrow 0$
        while $i \leq L$ and $G^i_{D_m} = G^i_\sigma$ **do**
          $i \leftarrow i + 1$
        end while
        if $P_n^{D^i_m} > P_\sigma^{D^i_m}$ **then**
          Forward m to n
          Delete m
        end if
      end if
    end for
  end for
  UpdateGridProbabilities(σ, α);
end procedure

3.2 GridRoute_NM

GridRoute_NM (GridRoute_NoMemory) is not very different from GridRoute, however, in GridRoute_NM nodes do not have any information about the destination. They do not store any contact or grid probability of other nodes. This feature can be crucial for devices with very small memory capacity, and, other than epidemic-like routing, there is no routing protocol that does not require information storage.

Although this routing process is similar to GridRoute in Section A, destination

Algorithm 2 GridRoute Update Grid-Cell Probabilities

procedure UPDATEGRIDPROBABILITIES($Node\ n$, α)
    for all l in L **do**
        for all g of n on G^l **do**
            if $g = G^l_n$ **then**
                $P^g_n \leftarrow (1-\alpha)P^g_n + \alpha$
            else
                $P^g_n \leftarrow (1-\alpha)P^g_n$
            end if
        end for
    end for
end procedure

Algorithm 3 GridRoute Receive

procedure RECEIVE
    for all Incoming messages m **do**
        if $R_m = \sigma$ **then**
            Receive Message
        else
            Send(m, R_m, F_{R_m}) ▷ Put to buffer to forward
        end if
    end for
end procedure

Algorithm 4 GridRoute Send

procedure SEND(m, R_m, F_{R_m})
    Enqueue(m to R_m at F_{R_m}) ▷ Put to buffer
end procedure

Algorithm 5 GridRoute_NM Forward

procedure FORWARD
  for all n in N st. $Distance(\sigma, n) \leq C$ **do**
    for all m in M of σ **do**
      if m is grid-cell info request **then** ▷ Epidemic
        if M of n !contain m and $HL_m \geq 0$ **then**
          Forward m to n
        end if
      else if $R_m = n$ **then**
        Forward m to n
        Delete m
      else
        $i \leftarrow 0$
        while $i \leq L$ and $G^i_{D_m} = G^i_\sigma$ **do**
          $i \leftarrow i + 1$
        end while
        if $P_n^{D^i_m} > P_\sigma^{D^i_m}$ **then**
          Forward m to n
          Delete m
        end if
      end if
    end for
  end for
  UpdateGridProbabilities(σ, α);
end procedure

Algorithm 6 GridRoute_NM Receive

procedure RECEIVE
  for all Incoming messages m **do**
    if $R_m = \sigma$ **then**
      if m is grid-cell info request **then**
        Send(F_σ to S_m at F_{S_m})
      else if m is grid-cell info for σ **then**
        Send(actual message to S_m at F_{S_m})
      else
        Receive Message
      end if
    else ▷ σ is a forward node
      Send(m, R_m, F_{R_m}) ▷ Put to buffer to forward
    end if
  end for
end procedure

Algorithm 7 GridRoute_NM Send

procedure SEND(m, R_m, F_{R_m})
    if $FavoriteCellOfRec$!= $NULL$ **then**
        Enqueue(m to R_m at F_{R_m}) ▷ Put to buffer
    else
        Enqueue("grid-cell request" to R_m epidemically)
        Store actual message to send when the grid-cell
        response is received
    end if
end procedure

information must be obtained in GridRoute_NM before the actual routing procedure. As indicated in Algorithm 5, information request is spread to the network with Epidemic Routing. Once the destination receives this request (see Algorithm 6), it sends its favorite grid cell information to the sender with GridRoute, and when the sender receives this information it can send the message to the destination with GridRoute (see Algorithm 7). Its only alternative is to use Epidemic Routing directly. However, rather than spreading kilobytes of information to the network, just a few bytes are transferred in GridRoute_NM epidemically. Although this process increases the average delay more than two times, high delivery ratio with reduced overhead on network can be achieved.

Gateways in the network is another scenario in which it is very suitable to use GridRoute_NM. Gateways are basically stationary nodes that provide Internet connection to nodes that are in communication range. Using GridRoute, the nodes out of the communication range of the gateway can get Internet connection. In this case, locations of gateways are known by the nodes in the network, and no epidemic message transfer is required between nodes to get the favorite grid-cell data. This feature allows GridRoute_NM to make intelligent routing decisions without storing any information of other hosts. Note that the only information stored at nodes in GridRoute is the favorite grid-cells of other nodes in the network. However, in this scenario, GridRoute_NM only needs to memorize the location information of just a few gateways. There is no need to hold information about other nodes. For example, assume that E6 is the grid-cell of the gateway in Figure 2 and assume that node m in I7 wants to send a message. As m knows the gateway location, it can ask the existence probability of other encountered nodes for cell E6 and can relay a message to another node that has bigger probability to appear on E6 in the future. Other than epidemic-like routing algorithms, all other algorithms such as [17], [12], [19], [7] need to hold contact information in order to operate efficiently even in this case. This requires n^2 memory where n is the number of nodes in the network. In this scenario GridRoute_NM continuously relays messages to the nodes that spend most of their time around gateways without holding any data about other nodes. As a result, high delivery ratio can be achieved as described in Section 5.

GridRoute can also be used directly without memory requirement with a reasonable assumption. If there is a systematic addressing protocol for nodes in the network, there is no need to obtain favorite grid-cell information using the three stage approach in GridRoute_NM. For example, e-mail addresses can be used to deliver messages. The address xyz@cs.univ.edu contains adequate information to retrieve the favorite grid-cell data. In this case, the receiver id is set to xyz, and the location of the computer science building in the university is set as the favorite grid-cell of the destination. In the simulations of GridRoute_NM, the performance of ordinary GridRoute is also added to graphs in order to cover this case.

3.3 GridRoute_NoGPS

GridRoute requires GPS modules on nodes in order to be able to operate. However, this restriction can be relaxed by GridRoute_NoGPS protocol. GridRoute_NoGPS allows for the existence of some number of nodes (actual proportion depends on the localization technique that is used) without GPS chips on the network. It uses Received Signal Strength Indicator (RSSI) to calculate the approximate position of a node without a GPS chip. Note that some nodes must contain a GPS device even in GridRoute_NoGPS. It only allows for participation of nodes without GPS devices to the network. GridRoute_NoGPS is very similar to GridRoute. It includes slight modification in Exponentially Weighted

Moving Average for the nodes without GPS devices. Nodes with GPS devices use the formula (1). Others must use the method below to calculate their grid-cell probability periodically:

$$P_{ij} = \begin{cases} (1-\alpha)P_{ij} + \alpha\left(\frac{RSSI}{RSSI_{MAX}}\right) & \text{if } i \text{ is in cell } j \\ (1-\alpha)P_{ij} & \text{Otherwise} \end{cases} \tag{3.2}$$

Algorithm 8 GridRoute_NoGPS Update Grid-Cell Probabilities

procedure UPDATEGRIDPROBABILITIES($Node\ n$, α)
    $arr[\,][\,] \leftarrow NULL$
    for all n in N st. $Distance(\sigma, n) \leq C$ **do**
        $arr[n][RSSI] \leftarrow RSSI_n$
        $arr[n][position] \leftarrow G_n$
    end for
    $trustedNode \leftarrow n\ with\ max.\ RSSI$
    $estimatedPosition \leftarrow arr[trustedNode][position]$
    for all l in L **do**
        for all g of n on G^l **do**
            if $g = estimatedPosition$ **then**
                $P_n^g \leftarrow (1-\alpha)P_n^g + \alpha$
            else
                $P_n^g \leftarrow (1-\alpha)P_n^g$
            end if
        end for
    end for
end procedure

Algorithm 8 presents the main steps in maintaining grid-cell probabilities in GridRoute_NoGPS. In every sampling period, a node without a GPS chip checks for other nodes in its communication range. It requests position information from the neighbors and assumes that it is in the same lower layer grid-cell with the node that it receives the signal with maximum RSSI value. It updates the probability values as described in (2). Other than this, routing is the same with the GridRoute.

Other than the RSSI technique, GridRoute_NM allows using more complex and accurate localization techniques such as [11] and [26]. In this case just a few anchor nodes with GPS devices are enough to use GridRoute_NoGPS effectively.

3.4 Security Advantage of GridRoute

Security is not the main focus of this paper, however, GridRoute has a major security advantage in terms of anonymity. Classical PKI-based solutions for security are not applicable for DTMNs due to its unconnected nature. Lots of effort is put on the security aspect of DTMNs and Identity Based Encryption (IBE) [28] is the common ground to provide

Algorithm 9 GridRoute_Secure Forward

```
procedure FORWARD
    for all n in N st. Distance(σ, n) ≤ C do
        for all m in M of σ do
            if D_m^0 = G_n^0 then
                Forward m to n
            else if H_m ≥ 0 then
                i ← 0
                while i ≤ L and G_{D_m}^i = G_σ^i do
                    i ← i + 1
                end while
                if P_n^{D_m^i} > P_σ^{D_m^i} then
                    Forward m to n
                    Delete m
                end if
            end if
        end for
    end for
    UpdateGridProbabilities(σ, α);
end procedure
```

secrecy, anonymity or authentication. However, other than Epidemic Routing, DTMN routing protocols that use contact probability or clustering cannot use IBE cryptography directly to provide anonymity as relay nodes must know the identity of the receiver to forward the message. There are several ways to provide security [27] and anonymity [14] in these protocols, however, they require lots of message traffic between two hosts for single successful message transfer.

GridRoute is able to achieve anonymity and secrecy by using IBE without any redundant message transmission. IBE is public key cryptography technique that

Algorithm 10 GridRoute_Secure Receive

```
procedure Receive
    for all Incoming messages m do
        if message can be decrypted correctly then
            if Signature is verified then
                Decrypt (m, PublicKeyOfSender)
            else
                Counterfeit message ▷ An adversary try to impersonate a user, discard the message
            end if
        else
            Send(m, D^0_m)                          ▷ Put to buffer to forward
        end if
    end for
end procedure
```

Algorithm 11 GridRoute_Secure Send

```
procedure Send(m, D^0_m)
    Enqueue(m to D^0_m)                             ▷ Put to buffer
end procedure
```

uses unique identity of users like email addresses as public keys. Private keys for decryption are generally distributed by a trusted server or authority. Assuming that private keys are distributed, GridRoute provides secrecy and anonymity as follows:

Sender appends a signature and the receiver ID to the message encrypted by its private key. Then this message with signature and ID is encrypted by the public key of the receiver. Different from the classical GridRoute packet, this packet does not contain the receiver ID at the header part but, rather, the favorite grid cell info of receiver is added like in Figure 3.2.

Routing is performed as in the classical GridRoute protocol until the message is transferred into the favorite grid cell of destination. As indicated in Algorithm 9, once the message is transferred into this cell, message holder relays the message to each user whose favorite cell is the same as the receiver of the message. However message holder does not delete the message from the buffer as it needs to remember that it received the message previously in order to remember the

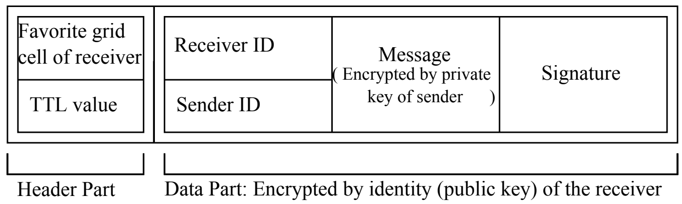

Figure 3.2: Sample packet format.

secure messages it has. The receiver of the message tries to decrypt the data part with its own private key. If it can be decrypted correctly then it means that the message is at the destination. If not, it means that it is a relay node, and it tries to send the message to a node that has the same favorite grid-cell as the message and has not received the message previously.

After a node identifies itself as the actual receiver of the message, one more step should be applied in order to guarantee that the sender of the message is actually the node that is indicated with the Sender ID. An adversary can encrypt a message with the public key of the receiver and he/she can try to impersonate another node by simply overwriting the SenderID field. In order to overcome this problem, once the message is decrypted successfully, the signature that is appended to the message must be verified by the receiver. As the original signature must be encrypted by the private key of the sender, only a real sender can encrypt it correctly. As a result, authenticity of the message can be guaranteed using the GridRoute. While in the favorite grid cell of the message, if the node leaves the cell before it can relay, it tries to forward the message to any neighbor in the cell. If this is not possible, it tries to resend the message to that grid cell using classical GridRoute protocol.

One important concern in this protocol is to prevent the loops in message transfer. Note that the nodes do not delete the secure messages that they relay from their buffers. By this way, they will not accept retransmission of the messages they receive earlier. However, this method does not guarantee a loopless protocol. Nodes may receive lots of messages, and the secure message in the buffer may get overwritten, which will result in forgetting the secure message and accepting possible future retransmissions. In order to overcome this problem, a hop limit value is added to secure messages in order to limit the possible number of transmissions of secure messages in the favorite grid cell of the destination.

In this protocol, intermediate nodes cannot learn the identity of the sender or receiver during this message transmission. The only information they gain is the favorite grid cell of the receiver. Assuming that more than one node has the same favorite grid cell, the actual receiver is hidden from the intermediate nodes. A node that receives sender x's message directly cannot distinguish whether x is the sender or it just relays the message. Also, the node that sends the message to receiver y cannot know whether y can decrypt the message or it is just a relay node. The receiver can also verify that the message has come from the actual sender. As the signature is encrypted by the private key of the sender, and it is assumed that the public key of the sender is known, the receiver can decrypt the signature with the identity of the sender that is appended to the data part.

3.5 GridRoute_IR

GridRoute_IR (GridRoute Inactive Replication) aims to gain the advantage of multi-copy routing without increasing the redundant message traffic. Existance of multiple copies of a message on the network significantly increases the probability of delivering the message to the destination. However increasing the number of message replicas causes an avalanche effect even if the replication is limited by some conditions. For example, PROPHET is a conditional replication based DTMN routing protocol. As it can be seen in Section V, even if the replication of messages is constrained, lots of redundant messages are spread to network. Other than increasing the traffic on the network, this feature wastes essential buffer spaces of the nodes. As a result, lack of adequate buffer space decreases the delivery ratio of the replication based routing protocols.

GridRoute_IR eliminates excessive message traffic generation of replicas by differentiating messages into two categories, namely active and passive messages.

Active messages are the ones that are forwarded to nodes exactly same to ordinary GridRoute. However passive messages are forwarded only if the forwarded node is the destination of the message.

In GridRoute_IR a message is marked as active when it is generated. As it can be seen in Algorithm 12, similar to GridRoute, a message holder tries to find a more promising node that has higher existing probability on the destination grid-cell of the message. Once a node with this property is found, message is forwarded to it as an active message however, different from ordinary GridRoute, message holder does not delete the message from its buffer but rather marks it as an inactive message. An inactive message can not be forwarded to another node until the message holder directly contacts with the destination of the message. In other words, only direct message transmission between the message holder and the destination is allowed for inactive messages.

This allows replication of messages with minimal extra message overhead. When compared with ordinary GridRoute, extra message transmission can occur only if the multiple message holders meet with the destination and if the destination deletes the received message from its buffer between this multiple reception. (Destination receives one copy of the message, after some time it deletes the reception information and then another copy of the message is forwarded to the destination.) However, as no indirect forwarding operations can be performed for the inactive messages, extra message overhead can be reduced significantly while gaining limited increase in message delivery probability.

Inactive messages are deleted from node buffers in time. As buffers of nodes operate in FIFO fashion, newly generated or received messages causes deleting inactive messages. Comparing with ordinary GridRoute, not deleting forwarded messages from the buffers directly can cause unintended message drops. For example, assume that a node has buffer of 10 messages and smaller indices are assigned for newly generated or received messages. Further assume that, buffer is full and two active messages x and y are in indices 9 and 10 of the buffer respectively. If x is forwarded to a more promising node, in GridRoute it is removed from the buffer and after that a new message can be stored in the buffer without effecting the message y. However in such a case in GridRoute_IR, message x would be marked as an inactive node but it would not be deleted from the buffer. So after this time, if a a new message is arrived from that node, due to FIFO queue implementation, the message y which is active would be deleted from the buffer in GridRoute_IR.

Algorithm 12 GridRoute_IR Forward

```
procedure FORWARD
    for all n in N st. Distance(σ, n) ≤ C do
        for all m in M of σ do
            if R_m = n then
                Forward m to n
                Delete m
            else if HL_m ≥ 0 then
                i ← 0
                while i ≤ L and G^i_{D_m} = G^i_σ do
                    i ← i + 1
                end while
                if m is active then
                    if P_n^{D^i_m} > P_σ^{D^i_m} then
                        Forward m to n
                        Mark m as inactive
                    end if
                end if
            end if
        end for
    end for
    UpdateGridProbabilities(σ, α);
end procedure
```

In order to mitigate this problem, FIFO implementation of the queue can be altered slightly. If the buffer of a node is full, first the inactive messages can be dropped from the buffer in FIFO fashion if there are any. As a result inactive messages in GridRoute_IR wastes buffer space only if there is enough free space and they do not effect the active messages. However especially in scenarios with high message generation rates, deleting inactive messages first rather than pure FIFO cancels the gain from inactive replication. As there are lots of newly generated messages, inactive messages are constantly deleted from buffers and GridRoute_IR becomes nearly identical to ordinary GridRoute. Hence, pure FIFO is implemented for GridRoute_IR even it causes to delete some active messages.

As it can be seen in Section V, this opportunistic replication significantly decreases the average message delay especially in scenarios with nodes that have big buffer spaces.

3.6 GridRoute_TA

The average delay of ordinary GridRoute is slightly higher than compared protocols as it is depicted in evaluation section. However, average delay can be decreased by augmenting time information to the GridRoute protocol. In order to improve the message delivery time, GridRoute_TA (Time Augmented GridRoute) is proposed in this section.

In ordinary GridRoute even if the messages are forwarded to nodes that spend lots of time in the favorite grid-cell of the destinations, there is no guarantee that the forwarded node and destination will be in the same grid-cell at the same time. For example, assume node x appears on grid-cell g between 01:00am to 11:00am and node y appears on g from 01:00pm to 11:00pm. Further assume that the destination of message m is y. As x and y spends lots of time in g probably both will have g as their favorite grid-cells. Whenever holder of m meets with x, it will see that x has a high existence probability in the favorite grid-cell of destination and the message will be forwarded to x. However, x and y actually never meets in g.

GridRoute_TA solves this problem by adding the time information to the location probabilities. In order to store the timely location information, a day is divided into periods. For example, periods of 1 hour may be sufficient to gather enough information but the actual period time again depends on the network parameters. Assuming that the period of 1 hour is used, a node will store an array of 24 existing probabilities for each grid-cell in each grid layer. Whenever it tries to update its existing probabilities, it finds the correct period according to current time and only updates that particular probabilities of the grid-cells.

The other 23 existing probabilities remain the same until that particular period has arrived.

This feature of GridRoute_TA allows selecting promising nodes that spends lots of time in the favorite grid-cell of destination of the message at similar times. However, as a node has multiple existing probabilities in one grid-cell, comparing the existing probabilities can not be done in one step. Nodes has an array of existing probabilities and cosine angle separation is an effective method to calculate the similarity between two vectors. When these multiple existing probabilities is thought as a vector, cosine angle separation [33] can effectively compare the existing probabilities of nodes for a given grid-cell.

Cosine angle separation can be calculated as follows:

$$CS_{ij} = \frac{\sum_{k=1}^{K} x_{ik}.x_{jk}}{\sqrt{\sum_{k=1}^{K} x_{ik}^2 . \sum_{r=1}^{K} x_{jr}^2}} \tag{3.3}$$

Other than this difference, the forwarding operation is very similar to ordinary GridRoute. In GridRoute_TA, nodes periodically check their neighbors and try to find a more promising node for each message in their buffers by calculating the cosine similarity between the destination grid-cell of the message and the neighbors on predefined subset of their array of existing probabilities for one cell. Starting from current period a subset of 12 existing probabilities with assuming 1 hour of periods is used in the simulations of GridRoute_TA in cosine angle separation calculation.

This feature allows focusing on closer time domain and selection of promising nodes that can deliver the message in the near future rather than nodes that can deliver message later. If a neighbor has higher similarity, that node probably spends more time in the destination grid-cell at similar times with the destination node. By this way, a node that spends lots of time in the destination grid-cell in different times than the destination or a node that will be in the same grid-cell at the same time with the destination, much later than the current time will not be selected to be forwarded. As a result the average message delay can significantly be decreased as it can be seen in Section V. A more formal description of GridRoute_TA can be seen in Algorithm 13.

Assume that 3 nodes have the following existing probabilities in grid g when the period is 1 hour. Existing probabilities for 24 hours are presented below:

D :	0	0	0	0	0	0	0	0.1	0.1	0.1	0.1	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0
X :	0.1	0.1	0.1	0.1	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y :	0	0	0	0	0	0	0.1	0.1	0.1	0.1	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0

In classical GridRoute there is no difference between node X and Y, as both of them has 0.6 existing probability on g. However, X actually never meets with destination D in g. On the other hand, Y is in g at similar times with D. Assume message is generated at the first period. Cosine angle separation metric between D and Y, as opposed to D and X is much higher when the first 12 values are used in calculation. Hence GridRoute_TA efficiently handles this situation and forwards the message to node Y.

A possible improvement to GridRoute_TA can be thought as relaxing the home grid-cell condition while selecting a node to forward a message. In all of the variations of GridRoute above, only the similarity between favorite grid-cell of destination is considered. However, a node that spends its time in any grid-cell at the same time with the destination node of the message can be thought to be a good candidate to forward the message. However existing probabilities and durations in grid-cells other than favorite position are not enough for effective message transfer with the destination. In an DTMN environment precise assumptions about message delivery times is not possible in general.

For example, assume that node x holds a message that its destination spends most if its time in grid-cell g from 11:00am to 11:30 like node y and current time 10:00am. If the message is forwarded to y, both nodes must meet in g in that period of 30 minutes. However this small time interval is not adequate to constitute a good probability to deliver the message. On contrary, human simulation studies like [2] and [4] indicates that, humans spend more than 60% of their (more than 14 hours) time in their favorite grid-cell. As a result messages that are sent to favorite grid-cell of destination do not need to be very precise in terms of delivery time on contrary with the suggested method above. Hence, this possible improvement is not added to GridRoute_TA and only the favorite grid-cells of destinations are considered.

Algorithm 13 GridRoute_TA Forward

procedure FORWARD
    for all n in N st. $Distance(\sigma, n) \leq C$ **do**
        for all m in M of σ **do**
            if $R_m = n$ **then**
                Forward m to n
                Delete m
            else if $HL_m \geq 0$ **then**
                $i \leftarrow 0$
                while $i \leq L$ and $G^i_{D_m} = G^i_\sigma$ **do**
                    $i \leftarrow i + 1$
                end while
                if $CS_{P_n^{D_m^i}} > CS_{P_\sigma^{D_m^i}}$ **then**
                    Forward m to n
                    Delete m
                end if
            end if
        end for
    end for
    UpdateGridProbabilities(σ, α);
end procedure

Algorithm 14 GridRoute_TA Update Grid-Cell Probabilities

procedure UpdateGridProbabilities($Node\ n$, α)
    $t \leftarrow$ Current time period
    for all l in L **do**
        for all g of n on G^l **do**
            if $g = G^l_n$ **then**
                $P^{g_t}_n \leftarrow (1-\alpha)P^{g_t}_n + \alpha$
            else
                $P^{g_t}_n \leftarrow (1-\alpha)P^{g_t}_n$
            end if
        end for
    end for
end procedure

Chapter 4

Simulation Experiments and Results

This section first explains the simulation environment and presents various performance analysis with detailed graphs. Detailed evaluations on suggested protocols that are compared with Epidemic routing PROPHET are also presented.

4.1 Simulation Environment

A stand-alone simulator in C++ is implemented to evaluate the performance of GridRoute like in [18] and [35]. This simulator makes simple assumptions about the underlying network layers. For example it assumes that there is 5% packet drop rate and nodes can transmit messages to each other no matter what, if they are in the in the communication range. The effect of transmission and computation delay is omitted in average delay calculations as they are negligable when compared to routing delay.

In a DTN simulation, it is important to establish a realistic mobility scenario and environment. In this thesis, GridRoute is compared with Epidemic Routing [35] and PROPHET [18]. Both of these two routing protocols uses random way-point mobility model [4] and this model is very popular in the evaluation of DTN routing protocols. Also this model is important as it includes almost no information about the node movements so, it can give an insight on how the protocols can work with limited knowledge on the network.

Random way-point mobility model is used in the first simulation model. In this model, 50 nodes are placed on 400 m x 1600 m sized area randomly. Both original Epidemic Routing and PROPHET papers uses network area of 500m x 1500m In order to be able divide the network are evenly in to grid-cells, the sizes are changed slightly. During the simulations, nodes choose a random destination in the area and move there with speed of 0 to 20 m/s that they decide randomly. Once they are in the destination, they wait there 0-10 seconds which is again chosen randomly and then they determine a new destination. In this simulation multi-copy GridRoute is implemented. In this case transmitter does not delete the message after the transmission until it is removed from the FIFO queue in time. The reason to implement the multi-copy GirdRoute is to provide a more even handed simulation environment as compared protocols use multi-copy approach. Single-copy GridRoute is implemented in the following simulation scenario.

Another mobility pattern is used in order to test GridRoute in a more realistic environment. Similar with [7], [18] and [30], community based mobility model (CBMM) is implemented. In this model, 5 grid cells are chosen randomly as hot-spots at the beginning of the simulations as it is observed in real mobility traces such as [15]. Also each node has a random home grid cell where it spends most of its time. In this model, 400 m x 1600 m network area is divided into disjoint subsets as home, hot and cold spots for each grid cell. A node chooses a destination probabilistically. A node has 0.7 probability to choose its destination as home and 0.2 probability to choose one of the hot spots. Finally with the probability of 0.1 it chooses one of the cold spots as destination randomly. When a node reaches its destination, it rests there 0 to 10 seconds which is determined randomly and again chooses a new destination probabilistically. In this simulation single copy GridRoute is implemented. Once the message is transmitted, transmitter deletes the message from the buffer.

In order to test GridRoute_NM (single copy), community based model is altered slightly. In this environment, a gateway is placed in a random central grid cell and each node tries to send message to it. Other than this change, all other parameters are remained same. A more general approach for GridRoute_NM is also presented. In this case again CBMM is used but nodes choose a random node as destination of messages rather than a gateway. As explained in the previous section, first favorite grid-cell information is requested by epidemic routing and the answer is sent by GridRoute. Finally, actual message is delivered using GridRoute again.

GridRoute_IR is tested using CBMM, however, in order to stress out the efficiency of GridRoute_TA the CBMM is altered slightly as CBMM_TA. 240 seconds are considered as a duration of a day and one day is divided into two as day and night which has total of 120 seconds. Each node in the network has different home and hot spot grid-cells during day and night so two random home grid-cells and 10 hot spots are assigned to each node, half of which is used during day times and the rest is for the nights. For GridRoute_TA, a day is divided into 24 for periods. Each node finds the correct time period and updates that periods existing probabilities as explained in previous section. Other than this difference nodes move like in the CBMM. However, if it is day time they choose home grid-cell or hot spot as destination from the grid-cells that are assigned for them as night cells and vice versa.

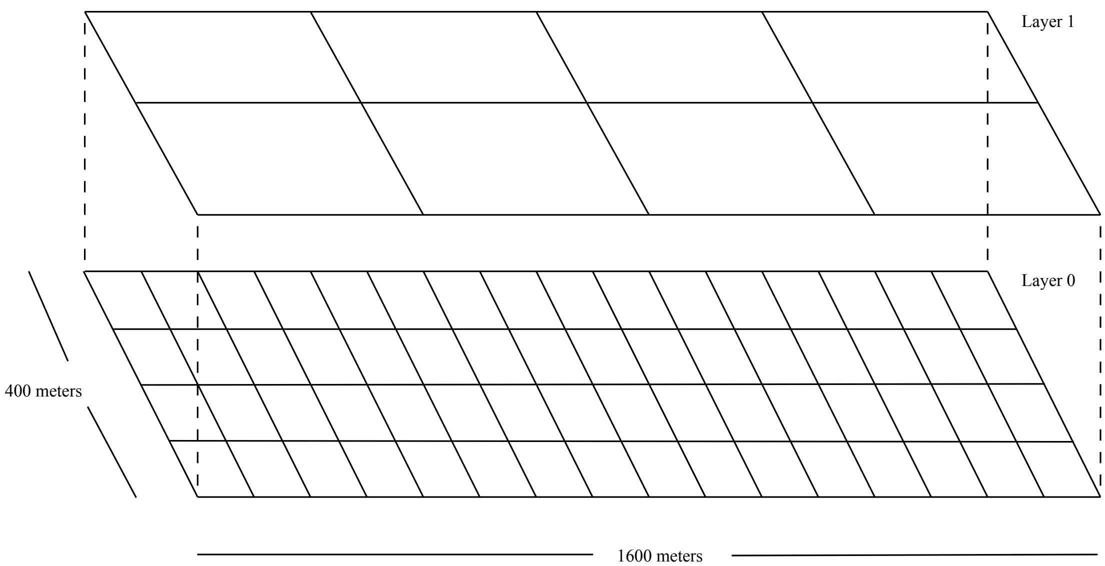

Figure 4.1: Network area and multi-layered grid in simulations.

In all cases, first 1000 seconds is used as a warm up period in order to collect probability data. After this period, in each second a randomly chosen node generates a message to randomly chosen destination for 2000 seconds of simulation time for random way-point mobility, and 5 nodes are chosen randomly to generate one message to random destination in CBMM and CBMM_TA. Finally nodes are allowed to communicate with each other for another 1000 seconds in order to let the messages reach to their destination.

Simulations with light message load is also performed in CBMM. In this case, one node is chosen as message generator in each 5 seconds randomly for a total of 2000 seconds. In other words, after 1000 seconds of warm up period, total of 400 messages are generated all of which has random sources and destinations. Simulation is continued further 1000 seconds in order to let the messages reach to their destinations.

2-layered grid is established on 400 m x 1600 m area for all GridRoute simulations as it is indicated in Figure 4.1. Layer 0 grid includes 64 cells with size of 100 m x 100 m. Upper layered grid consists of 8 grid cells with sizes of 200 m x 400 m each. Each node in GridRoute simulations, maintains their visiting probabilities of these 72 grid-cells and it is assumed that each node has a GPS chip to calculate their exact position.

GridRoute, Epidemic Routing and PROPHET is compared in these three mobility patterns with different communication range, buffer capacity and hop limit parameters. If it is not stated otherwise, hop limit of 3, transmission range of 100 meters and 50 nodes are used in the simulations. It is important to note that hop limit is defined for per message. For example if a message has hop limit of 3, before that message is transmitted to a node, hop limit is decreased by one so, both the forwarding and the forwarded nodes have messages with hop limit of two if the forwarding node does not delete the message or mark it as inactive. Hence, a generated message with hop limit of 3 can be forwarded to 8 different nodes. Also in Epidemic Routing and PROPHET some nodes may receive the same message more than one time. For example, a node may receive one copy of message m and after some time it may be deleted from its buffers. However, if that particular node contact with another holder of m after it deletes the previous copy, it would receive the message again. This is one particular reason of high redundant message traffic of replication based protocols which will be discussed in detail in the next section. In the simulations no method is implemented to prevent multiple reception of the same message as this may be the case in real networks when the Epidemic Routing or PROPHET is used as the routing protocol.

The effect of network parameters like node density, message generation rate, node mobility and transmission range are also analyzed and the results are presented in the next section. For each simulation case, simulations are run 10 times and their average is presented in graphs. Then the evaluation on results are presented in Section V.

4.2 Number of Received Messages in RWMM

Figure 4.2 and Figure 4.3 show simulation results of random way-point mobility model (RWMM). To begin with delivery ratios, Figure 4.2 and Figure 4.3 indicate that Epidemic Routing and GridRoute deliver approximately same number of messages. Although there is no acknowledgment mechanism, both protocols are able to deliver nearly all of their messages to destination with a buffer of size 200. Epidemic Routing seems to be more successful as it delivers 2 - 10% more messages than GridRoute but, when delivery ratio is compared with the total number of message transmission presented at the next page, it will be seen that GridRoute is a much more efficient algorithm in terms of successful transmission / total number of transmission count ratio. Moreover, whenever an adequate buffer space is provided, GridRoute even outperforms Epidemic Routing marginally as it can be seen in Figure 4.2 with buffer size of 100 messages.

Buffer capacity has an important effect on the delivery ratio for both algorithms. As it can be seen in Figure 4.2 and Figure 4.3, it is more important than communication range or hop limit in terms of successful transmission count. The reason is that small buffers delete most of the messages that are not transmitted yet as they receive lots of new ones. Hence effect of buffer size is directly related with the message generation rate in the network. Nodes with small buffers in a small message generation rate environment may have bigger delivery ratio than nodes with huge buffers that generates messages very frequently. The effect of message generation rate will be analyzed in detail later in this section.

Also it is interesting to point out that, transmission range and hop limit do not have significant effect on delivery ratio in random mobility model. Although increasing these two parameters allow message transmission to greater number of nodes, these transmissions waste lots of buffer space hence the increase in delivery ratio is limited. This limited effect is also related with the node mobility in the network. Transmission range of 50 m together with 10 m/s node mobility on average allows nodes to traverse nearly whole network area in a limited time. Hereat, increasing the range to 100m does not provide significant contact opportunity between message and receiver but decrease the average message delay to some extend as described later in this section. Similarly, effect of node mobility will also be presented and discussed in detail.

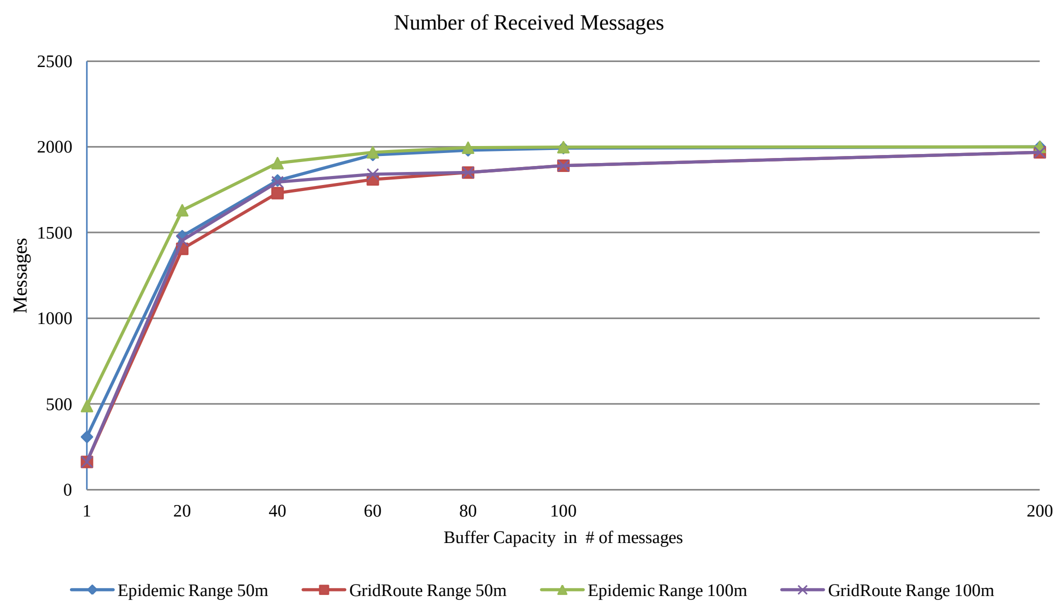

Figure 4.2: Number of received messages, RWMM, hop limit: 3.

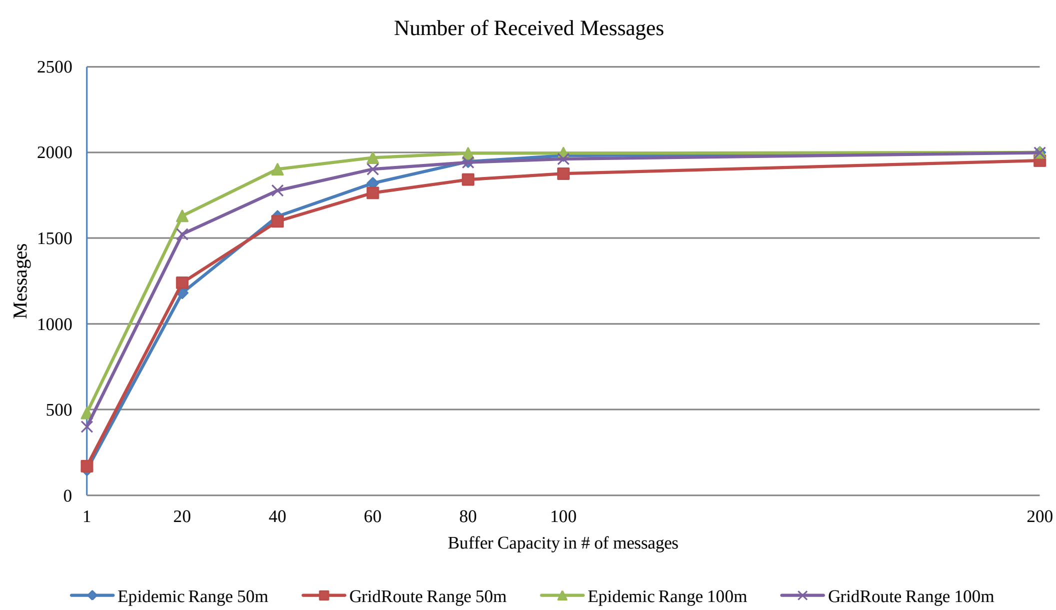

Figure 4.3: Number of received messages, RWMM, hop limit: 11.

4.3 Number of Forwarded Messages in RWMM

Figure 4.4 and Figure 4.5 show the redundant message transmission for both protocols. Similar with delivery ratio, buffer capacity has important effect on number redundant messages. However buffer capacity is not the only significant factor that effects the number of redundant messages. As Epidemic Routing delivers message to each uninfected node in communication range, increasing buffer size even by 1 enables nodes to remember one more message which results in an avalanche effect in terms total number of transmissions. As nodes can hold more messages in their buffers, they can relay more messages to their neighbors which result in excessive redundant message transmission.

On the other hand, GridRoute first checks the gird-cell probability values and forwards message only if the receiver has higher existence probability. Epidemic Routing transmits 3 times more messages than GridRoute which means that GridRoute can decrease the amount of energy that is spend on message transmissions approximately to 1/3 of the Epidemic Routing. The reason is that, GridRoute transmits messages only to better nodes which is also valid for PROPHET. This limits the number of transmissions and saves buffer space. On the other hand Epidemic Routing transmits the message to any encountered node that is not infected.

Different from delivery ratio, transmission range has an important effect on redundant messages. Nodes with bigger communication range spreads the message to greater number of nodes in Epidemic Routing and can find more nodes that has higher existence probability in GridRoute. As a result in both cases number of redundant messages increases up to 33% for both of the protocols.

Comparison of Figure 4.4 with Figure 4.5 points out that, hop limit does not have any significant effect on the redundant message traffic for both protocols. For Epidemic Routing increasing the hop limit nearly 4 times only increases the redundant message traffic about 20% because most of the messages are overwritten by newly generated ones before they can be forwarded 11 times. Also GridRoute delivers the messages to the destination in less than 11 hops in most of the time so the total number of redundant messages is not effected very much by the hop limit.

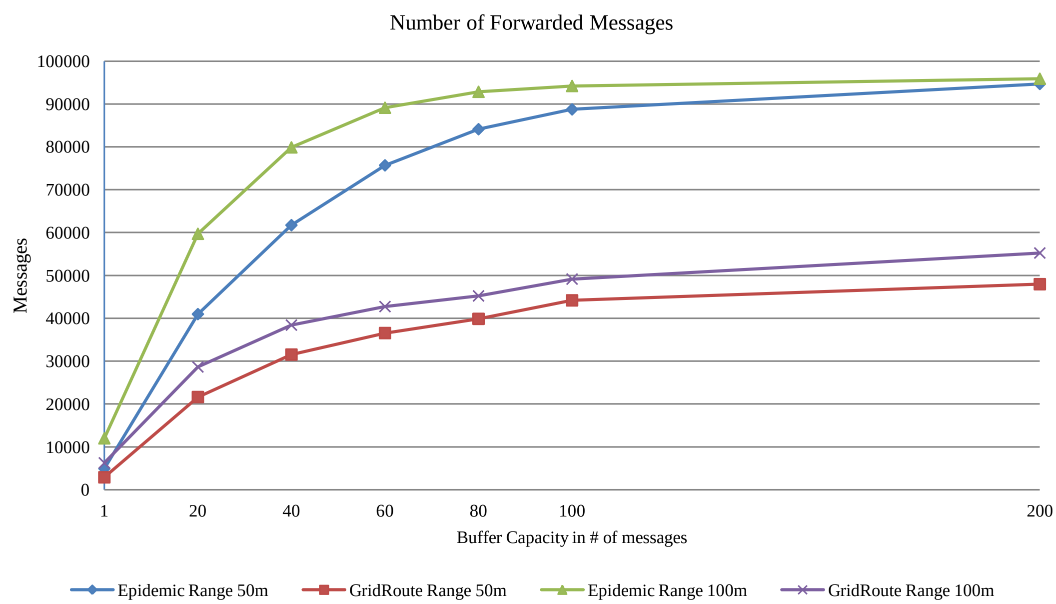

Figure 4.4: Number of forwarded messages, RWMM, hop limit: 3.

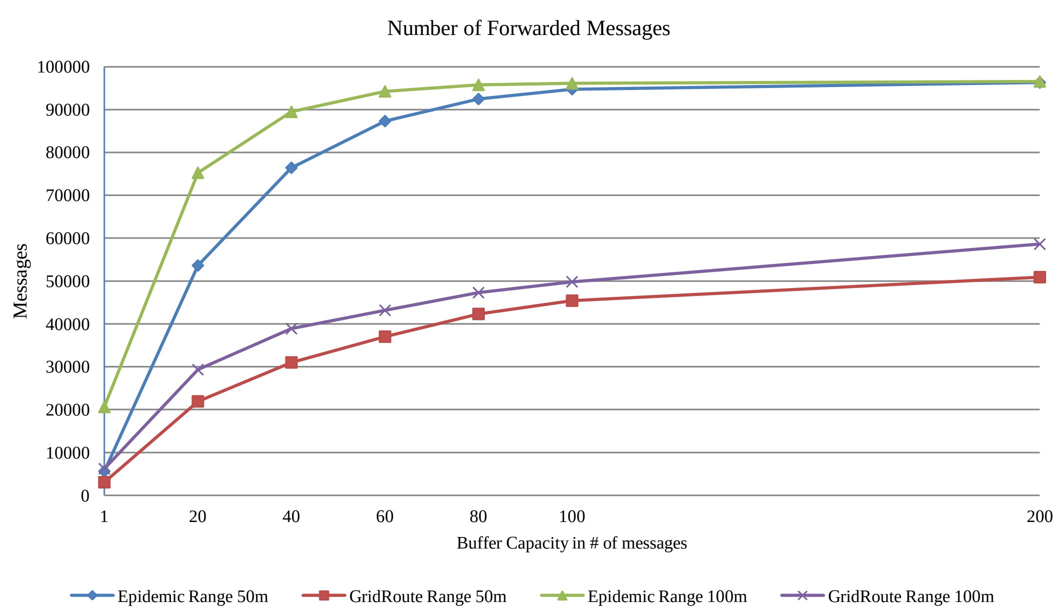

Figure 4.5: Number of forwarded messages, RWMM, hop limit: 11.

4.4 Average Delay in RWMM

Average delays in random mobility can be seen in Figure 4.6 and Figure 4.7. Epidemic Routing performs slightly better than GridRoute as it spreads lots of redundant messages through the network. However, although it spreads 3 times more messages to network, it obtains limited decrease in average delay. Figure 4.6 and 4.7 point out that the average delay increases with the increase in buffer space. However it is important to note that more buffer space also increases the delivery ratio. Bigger buffers allows messages to reside in queues longer until the delivery.

In random way point mobility model, number of replicas in the network seems to be directly proportional to the success change of message delivery. However spreading the messages 3 times more does not provide 3 times of reduce in average delay with hop limit of 3. This means that Epidemic Routing spreads the messages to lots of redundant nodes. GridRoute's selective forwarding mechanism limits this number of redundant messages. However it may also miss some important nodes that can deliver the message to the destination. As a result it has up to 40% more average delay with hop limit of 3.

Hope limit of 11 produces different results. In this case, Epidemic Routing gains the above mentioned 3 times of reduce in average delay. One important reason for this is the node density and simulation area. Hop limit of 11 and communication range 100 meters with 50 nodes with buffer capacity more than 20 is enough to spread the message to nearly to all network area. Thus this can be think as a unlimited resource case. In this case Epidemic Routing decreases the average delay with the cost of spreading the each message to each node. However this unlimited resources probably will not be feasible in a real environment. On the other hand GridRoute uses limited amount resources and provides feasible average delays on message transmission.

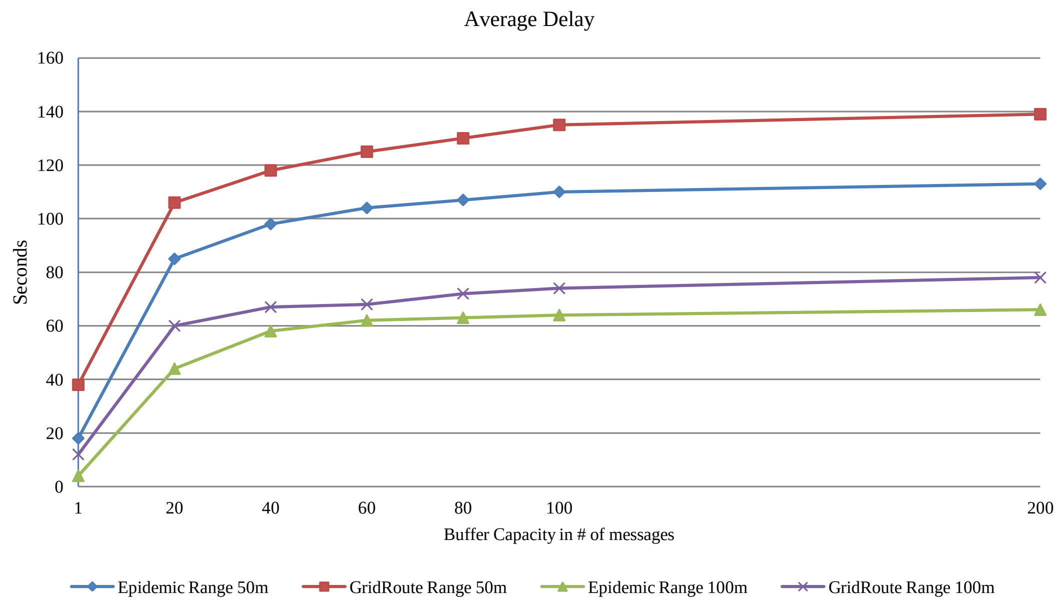

Figure 4.6: Average delay, RWMM, hop limit: 3.

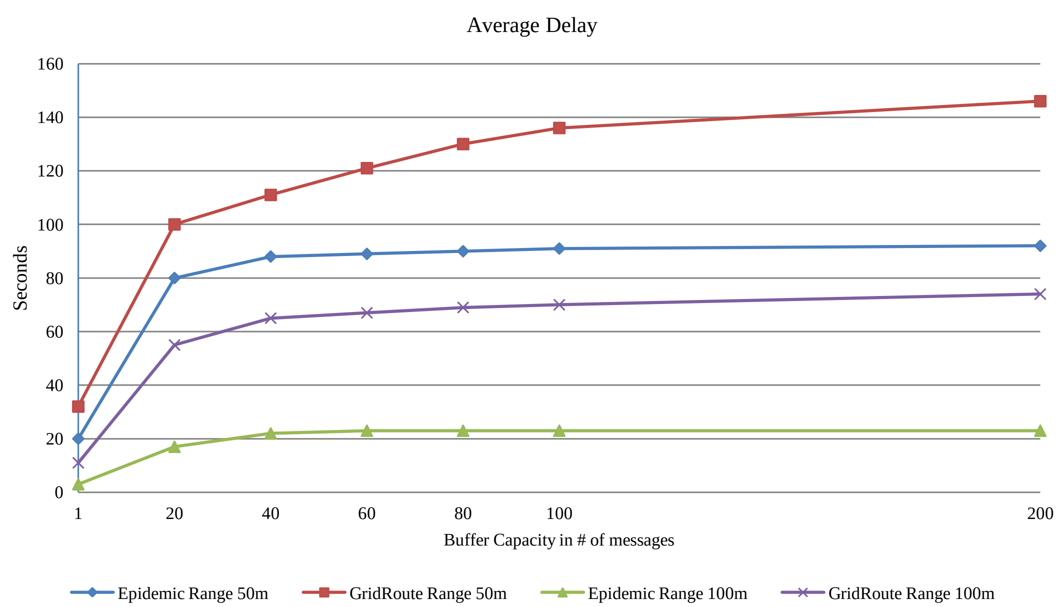

Figure 4.7: Average delay, RWMM, hop limit: 11.

4.5 Number of Received Messages in CBMM

Figures 4.8 and 4.9 show the simulation results of community based model. As it can be seen in these figures, GridRoute outperforms Epidemic Routing and PROPHET in terms of delivery ratio regardless of the buffer capacity, transmission range or hop limit. These 2 graphs indicate that, buffer capacity increases the delivery ratio for all three protocols and also transmission range of GridRoute can increase delivery ratio significantly when the buffer size is small.

The advantage of GridRoute in limited buffer space case mainly depends on its single copy message transmission. Epidemic Routing and PROPHET discard lots of messages from their buffers before these can be delivered to their destinations as new messages are generated and as they spread the messages to multiple nodes throughout time. If there are n nodes in the network, GridRoute can keep n times more messages than Epidemic Routing and PROPHET at the best case. Hence in GridRoute it is much less likely that a particular message is discarded which increase the chance of message delivery to destination.

Additionally, GridRoute also outperforms the other two protocols in case of large buffered nodes. The main reason behind this improvement is the locality of the mobility as stated in [2] and [4]. Multiple layered grid in GridRoute captures this feature on human mobility. Applying a forwarding strategy based on areal approach allows increasing the message delivery probability in more relaxed conditions when GridRoute is compared with PROPHET. In GridRoute, message can be forwarded to a node that has no chance of delivering the message to the destination. However that node carries the message to a closer area to the destination's favorite grid-cell and due to locality principal message delivery probability is increased. However, PROPHET omits this kind of opportunities as it focuses on several hop contact probabilities between nodes.

The performance of Epidemic Routing is decreased in community based mobility model when it compared with random way-point mobility model. In this model, as nodes do not move randomly but rather tend to stay in communities, message spread rate is lower. The power of Epidemic Routing is the its ability of spreading the message to whole network. Due to lower spread rate of the messages, it is harder for Epidemic Routing to spread the message to whole nodes and its aggressive buffer usage limits its delivery ratio. GridRoute outperforms Epidemic Routing up to 40%.

Increasing the communication range effects all of the above mentioned 3 protocols. The delivery ratio of the Epidemic Routing is increased approximately 35%. Although, increasing the range in Epidemic Routing cause wasting more buffer space, it also allows delivering the message to destination faster. Hence more messages can be delivered to destination before they are removed from the buffers. Also bigger range spreads the message to greater number of nodes so the delivery ratio is increased with the increase in communication range for Epidemic Routing despite the increase in waste of buffer space.

The advantage of bigger of transmission range is similar for both PROPHET and GridRoute. Increasing the range allows these two protocols to communicate with greater number of nodes. Hence the probability of finding a promising node for delivery of the message to the destination increases. One main difference is the capture of indirect promising nodes. In other words, nodes that do not likely to deliver the message to the destination but have a reasonable chance to forward the message to a node that can deliver the message to the destination in several hops. Capturing this information directly is computationally very costly as all nodes need to calculate their indirect communication probability with each other node in the network. However, GridRoute captures this feature implicitly as it operates on areal existence probabilities. This property of GridRoute brings a reasonable delivery advantage to nodes especially when a node does not have any neighbor that has a promising probability of direct delivery of message to the destination.

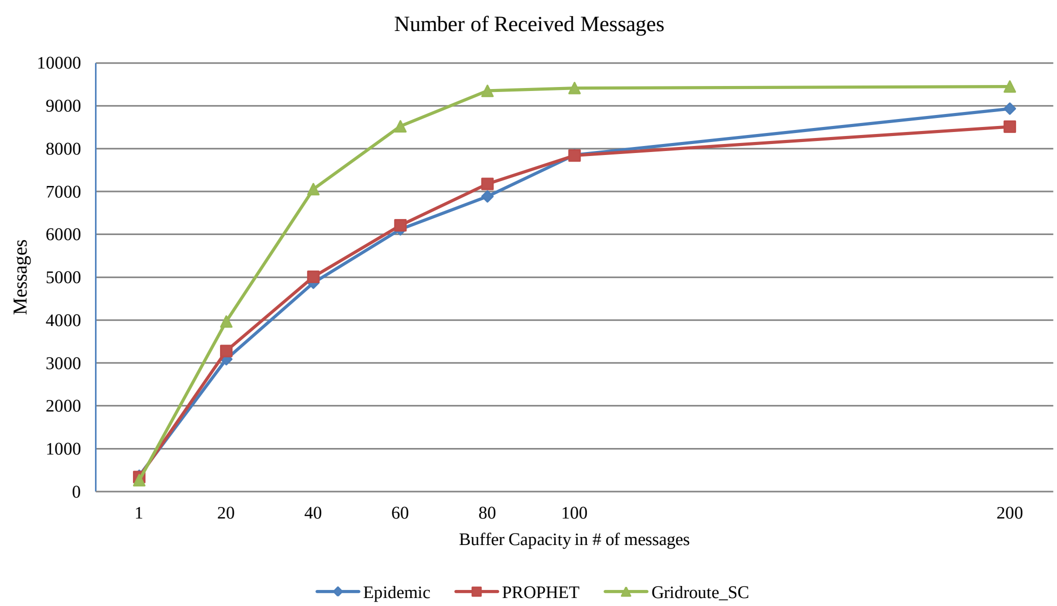

Figure 4.8: Number of received messages, CBMM, range: 50 m.

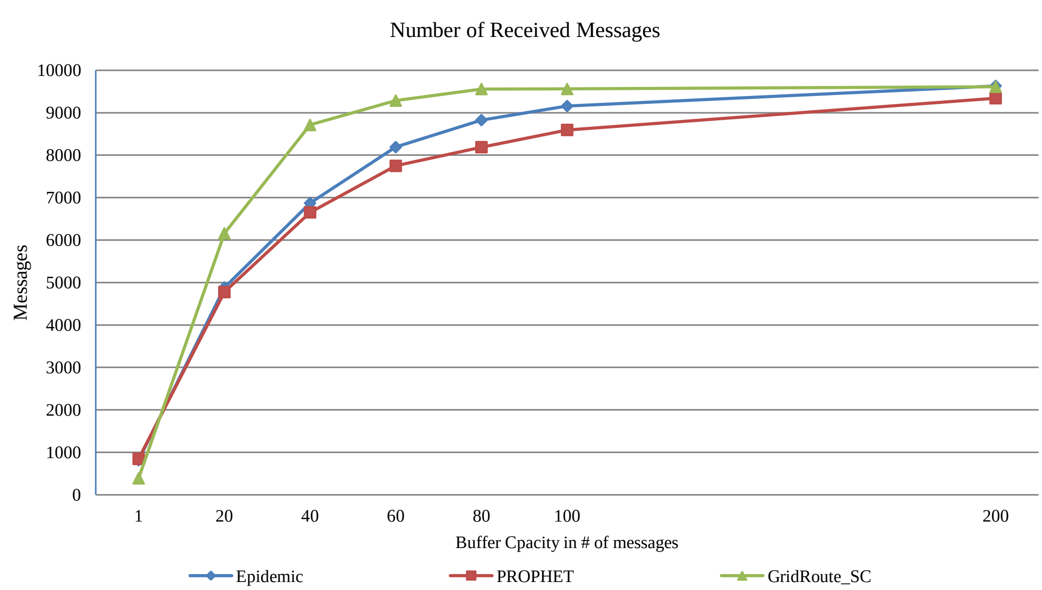

Figure 4.9: Number of received messages, CBMM, range: 100 m.

4.6 Number of Forwarded Messages in CBMM

Looking at the Figures 4.10 and 4.11 it can be clearly seen that GridRoute does not produce almost any redundant messages. It is a single copy routing protocol so it does not spread the messages through network. Also these figures indicates that GridRoute is able to deliver messages in 2 or 3 hops approximately. Its message overhead is nearly constant against the increase in buffer size(after buffer size of 20) or communication range. It produces approximately 4 times less message traffic than PROPHET and 6 times less messages than Epidemic Routing. This also indicates that, GridRoute can clearly improve the lifetime of the network and can save lots of energy.

To begin with analyzing GridRoute, average hop count validates the motivation behind the protocol. Note that two layered grid is used in the simulations. This means that, first message transmission from the message originator more probably checks for the higher level grid-cell probability with its neighbors to choose the best candidate. This message transmission apparently allows to the forwarding node to deliver the message to the destination even if the favorite grid cell of the forwarder is not same with the destination as average hop count is 2 roughly. Thus locality of mobility can be captured by GridRoute efficiently.

Moreover, when number of redundant message is compared with the number of received messages in Figure 4.9 and 4.10, it can be seen that the efficiency of GridRoute much higher. $Received\_messages/Redundant\_messages$ is good metric to compare the efficiency. It can be seen in Table 4.1 that GridRoute is much more efficient than both of other protocols. It can reach 0.684 efficiency while maximum efficiency is 0.147 for Epidemic Routing and 0.290 for PROPHET. Moreover if the buffer size of 1(which is an extreme condition) is not considered, lowest efficiency of GridRoute is 0.502 while it is 0.096 for Epidemic Routing and 0.155 for PROPHET. This means that GridRoute is able to deliver the messages by generating significantly fewer message traffic. More importantly, GridRoute can deliver more messages than Epidemic Routing and PROPHET by generating 6 and 4 times less redundant messages respectively.

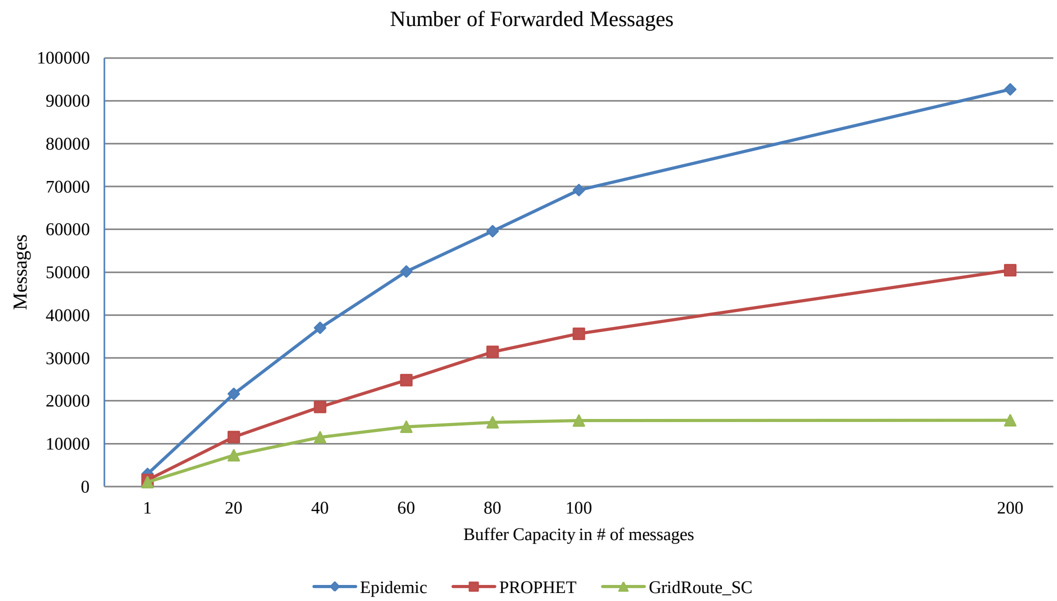

Figure 4.10: Number of forwarded messages, CBMM, range: 50 m.

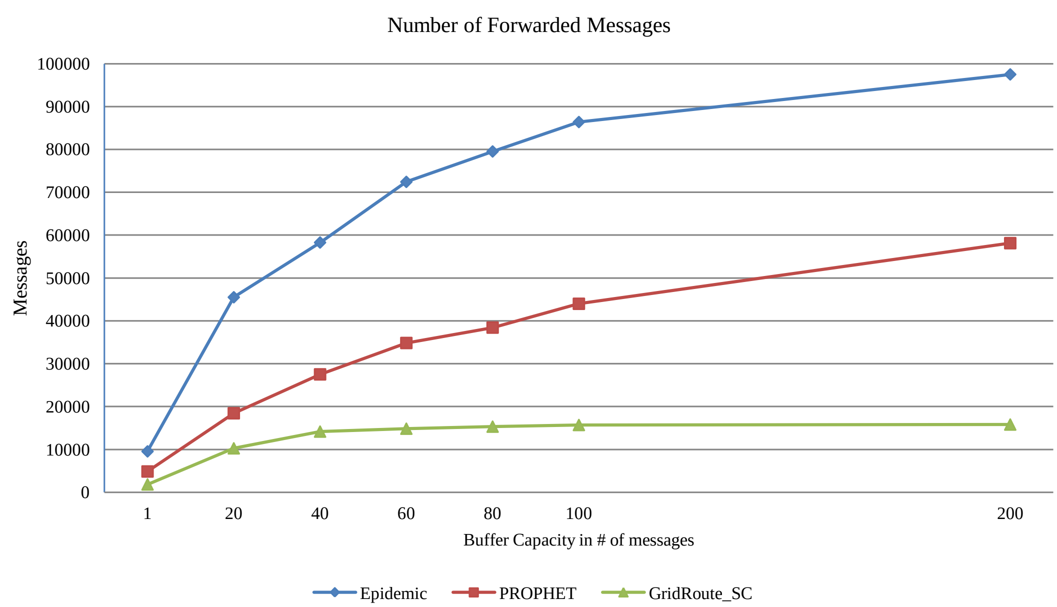

Figure 4.11: Number of forwarded messages, CBMM, range: 100 m.

Comparison between Figures 4.4 and 4.5 with 4.10 and 4.11 point out that, when the simulation model is changed to more realistic case, the redundant message traffic of the spreading based protocols drop. The main reasons for this diminution is that the spreading of the message to whole network is not very easy when it is compared with the random mobility. Although this seems to be a desired property, it also decreases the successful transmission ratio. Because the message deliveries that are based on random contacts are not very likely to occur in community based mobility. Decrease in this kind of message deliveries is the main reason in the performance and efficiency drop in community based mobility for Epidemic Routing and PROPHET.

Note that there are some randomness even in the community based model so in a more realistic environment, the performance of the Epidemic Routing and PROPHET will be lower. On the other hand GridRoute will take advantage of the more accurate data so the performance of it will probably increase. GridRoute uses single copy approach and randomness in the network may cause forwarding the message to a node that has higher existence probability of destination grid-cell due to random movement. After all GridRoute achieves higher efficiency even there is an randomness in the movement that can not be underestimated.

4.7 Average Delay in CBMM

Figures 4.12 and 4.13 present the average delay results from community based mobility simulations. PROPHET and Epidemic Routing has approximately same delay and GridRoute is very close to these two in most of the cases. At the worst case, GridRoute delivers message using 50% more time than Epidemic Routing. This is due to fact that, GridRoute does not spread the message to the network and waits for contact of single copy with the receiver in order to decrease message overhead. Increase in latency (less than 50%) is reasonable as it decreases the redundant message overhead approximately 6 times.

In Delay Tolerant Networks, delay of a Epidemic Routing with unlimited

Range	Buffer Capacity	Epidemic	PROPHET	GridRoute
50 m.	1	0.066	0.190	0.225
	20	0.147	0.290	0.502
	40	0.128	0.262	0.666
	60	0.122	0.240	0.669
	80	0.115	0.229	0.650
	100	0.112	0.213	0.613
	200	0.097	0.170	0.620
100 m.	1	0.090	0.182	0.333
	20	0.104	0.252	0.610
	40	0.116	0.234	0.684
	60	0.114	0.225	0.650
	80	0.111	0.205	0.633
	100	0.103	0.194	0.607
	200	0.096	0.155	0.602

Table 4.1: Efficiency of compared protocols in CBMM in terms of delivery ratio and number of redundant messages.

resources is considered as the lower bound. Because the message is delivered to all encountered nodes with no restriction. In CBMM simulation Epidemic Routing is nearly in this condition when the buffer size is more than 200. However, in the simulations packet drops in buffers increase the message delay of the Epidemic Routing. Nevertheless, Epidemic Routing generally spreads the message to nearly whole network so it has very low average message delay.

Interestingly, PROPHET performs better than Epidemic Routing in terms of delay. It selectively spreads the message so, the number of message drops in its buffers are significantly lower than Epidemic Routing. It makes intelligent forwarding decisions by taking contact probabilities into the consideration and it selectively spreads the message which increase the delivery probability. All of these advantages of PROPHET enables the delivery of the messages faster than the Epidemic Routing as the resources are limited in the simulations.

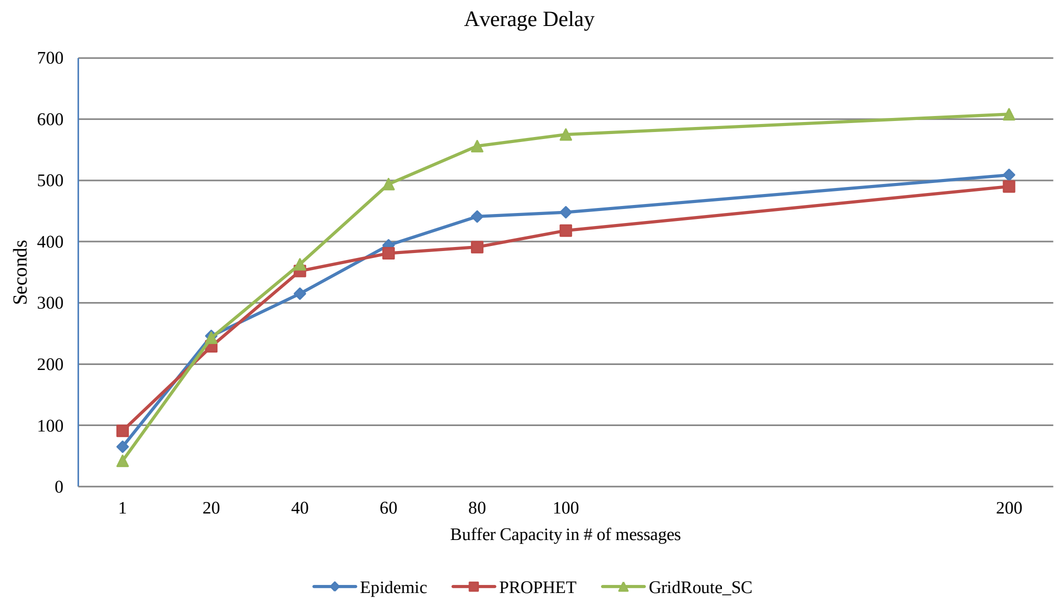

Figure 4.12: Average delay, CBMM, range: 50m.

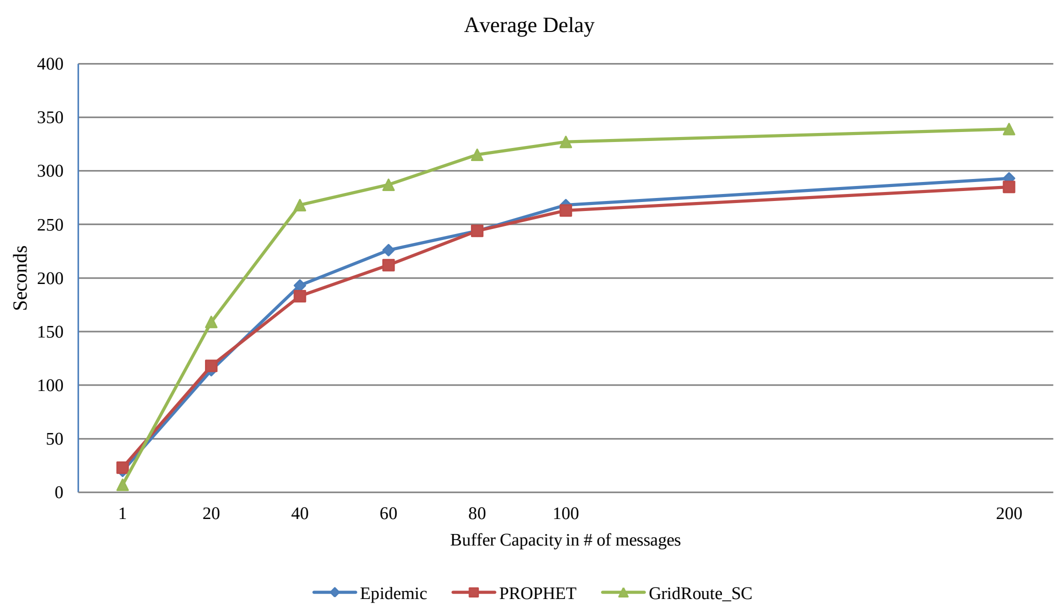

Figure 4.13: Average delay, CBMM, range: 100 m.

Range	Buffer Capacity	Epidemic	PROPHET	GridRoute
50 m.	1	2.85	2.11	3.60
	20	12.40	13.91	16.60
	40	15.80	14.28	19.70
	60	15.25	15.89	17.75
	80	15.68	17.97	16.25
	100	17.55	19.26	15.86
	200	17.64	17.34	15.24
100 m.	1	36.02	36.23	32.04
	20	40.83	40.12	39.35
	40	36.31	37.77	32.96
	60	36.81	37.61	32.50
	80	37.08	33.33	30.64
	100	33.70	33.46	29.09
	200	33.10	32.85	28.23

Table 4.2: Efficiency of compared protocols in CBMM in terms of delivery ratio and average delay.

Table 4.2 presented an important efficiency metric. One other equally important efficiency criteria is $delivery\_ratio/average\_delay$. Although GridRoute is a single copy routing protocol and the redundant messages are not included in the efficiency metric, its efficiency is very close to PROPHET and Epidemic Routing. GridRoute even outperforms these two protocols when the resources are limited like the case with range of 50 m. and buffer capacity of 40 messages. In general, relatively high latency of GridRoute decrease its efficiency even though it delivers more messages than other two protocols. To sum up, if the resources are not limited and the number of redundant message is not very important, using PROPHET or Epidemic Routing can be a better choice but this is not the case mostly in real life.

In order to achieve a more realistic efficiency metric, all of the metrics should be included in the calculation. The new efficiency is calculated by $E = D/(A*R)$ where E is efficiency, D number of successful messages, R is number of message transmissions (including redundant messages) and A is average delay. An ultimate routing protocol in an utopic environment will have an efficiency metric of

Range	Buffer Capacity	Epidemic	PROPHET	GridRoute
50 m.	1	0.00095	0.00211	0.00450
	20	0,00059	0.00120	0.00200
	40	0.00041	0.00075	0.00180
	60	0.00030	0.00063	0.00130
	80	0.00026	0,00057	0.00110
	100	0.00025	0.00052	0.00104
	200	0.00019	0.00034	0.00101
100 m.	1	0.00360	0.00730	0.03560
	20	0.00086	0.00210	0.00393
	40	0.00061	0.00130	0.00253
	60	0.00051	0.00107	0.00232
	80	0.00046	0.00085	0.00204
	100	0.00038	0.00074	0.00181
	200	0.00033	0.00055	0.00176

Table 4.3: Efficiency of compared protocols in CBMM in terms of delay, delivery ratio and redundant message traffic.

1 as $D = 10000, A = 1$ and $R = 10000$ which results in $10000/(10000 * 1) = 1$. It is pointed out in Table 4.3 that, GridRoute is much more efficient than the Epidemic Routing and PROPHET and closer to the ultimate efficiency. It achieves higher delivery ratios than these two protocols by using much less resources. Although it has higher average delay, the gain in other two parameters increases the efficiency of GridRoute and covers for the higher latency in the message delivery which is also not very high from other two protocols.

Communication range has a positive effect on efficiency while the buffer capacity effects it contrarily. Increasing the communication range allows greater number of successful transmission at the cost of relatively less redundant messages. As a result the more efficient communication can be achieved by increasing the range. Buffer capacity decreases the efficiency for different reasons. Increasing it causes lots of redundant message generation in Epidemic Routing and PROPHET. More buffer space in GridRoute increases the message delivery at the expense of greater message delays. As the performance of GridRoute is very efficient in small buffered scenarios, more buffer space provides a little increase in delivery ratio but increases the average delay a lot. As a result the efficiency of all protocols are affected negatively from increase in buffer capacity.

Note that the average message delay is calculated only for successful message transmissions. Although this calculation provides good results on expected delivery latency, it also favors protocols that are able to deliver less messages in a short amount of time. For example a protocol that sends 1% of messages in 1 second would have an average delay of 1 which may result in inacurate

4.8 Average Delay With Penalty in CBMM

interpretation when the concern is only delivery latency. Also this problem effects the comparison of average delay among different buffer sizes. If a penalty is added to average delay for unsuccessful message transmissions, the efficiency of GridRoute will increase relatively as its delivery ratio is higher than other two protocols

It seems that protocols in small buffered simulations have smaller average delay. However, they only deliver less amount of messages in a relatively shorter time and can not deliver lots of messages to the destination. In order to overcome this problem, a penalty of 1000 seconds of delay is added for unsuccessful transmissions.

Figures 4.14 and 4.15 indicate that buffer capacity has a positive effect on average delay when the unsuccessful transmissions are taken into consideration. GridRoute performs relatively better when the resources are limited. Although it is a single copy protocol it can deliver messages in a shorter time when the buffer capacity is less than 80. In simulations with bigger buffer sizes, Epidemic Routing and PROPHET spreads the message to large portion of network so they can achieve better delay values. However even in these simulations GridRoute only has 9.3% more average delay at most. To sum up, GridRoute is able deliver more messages by generating far less traffic in the network and it only has marginally higher average delay.

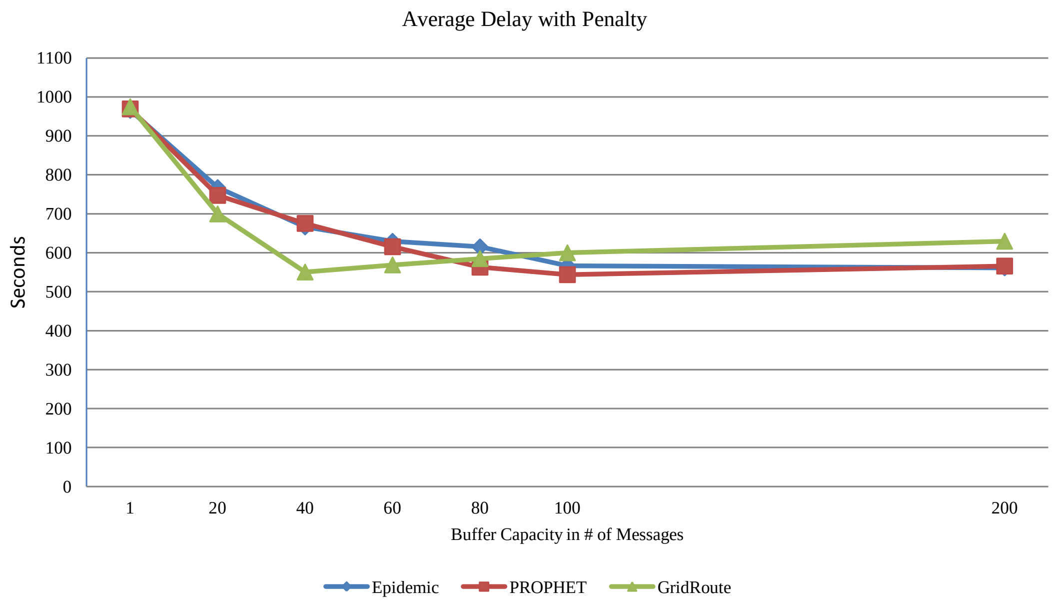

Figure 4.14: Average delay with undelivery penalty, CBMM, range: 50 m.

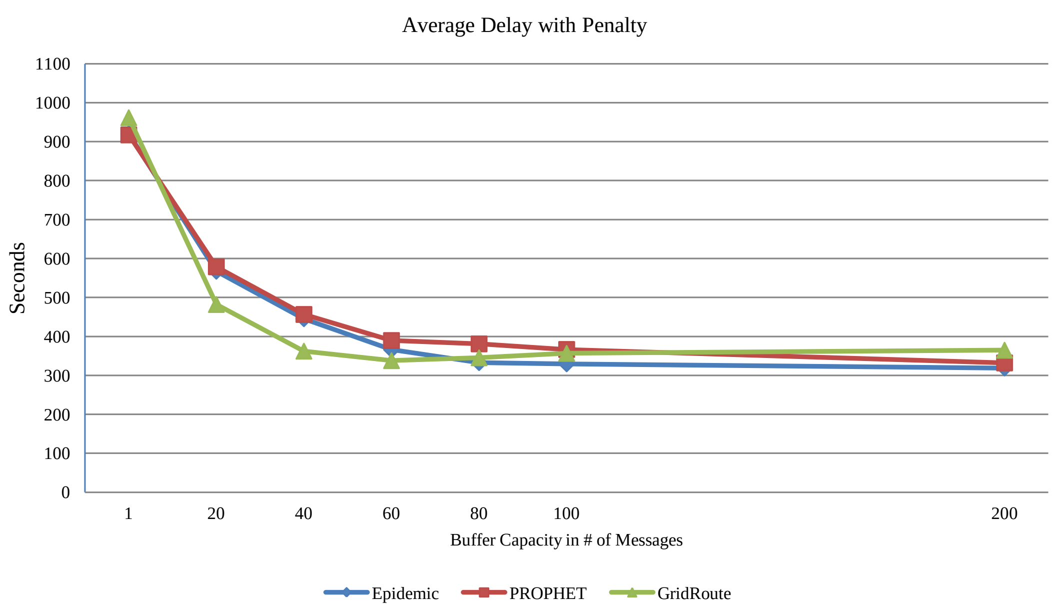

Figure 4.15: Average delay with undelivery penalty, CBMM, range: 100 m.

4.9 Simulation Results of GridRoute_NM

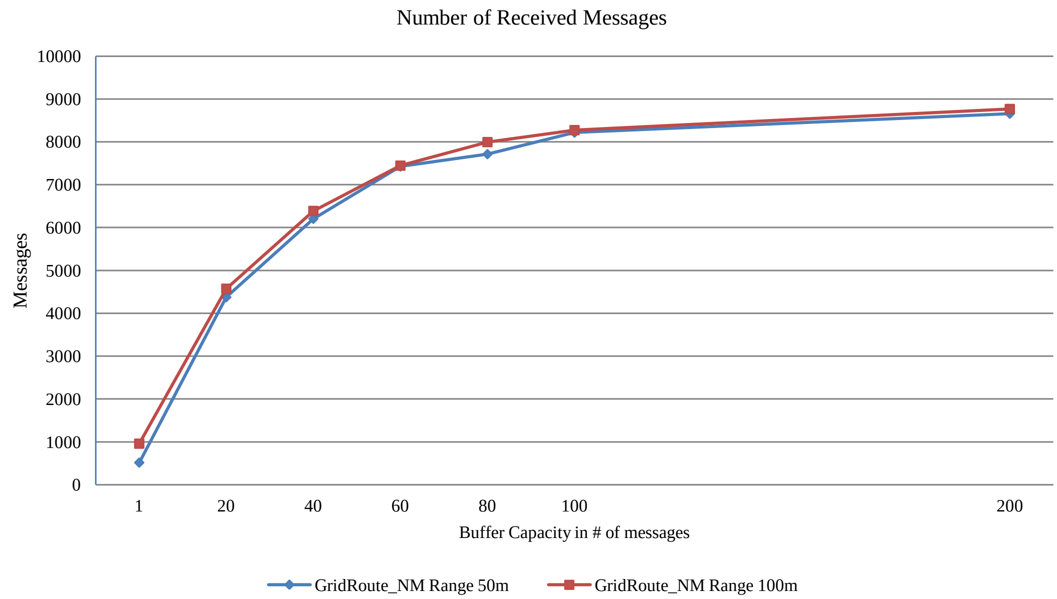

Figure 4.16: Number of received messages, GridRoute_NM, gateway simulation, range: 50 m, hop limit: 11.

Simulation results of GridRoute_NoMemory in gateway scenario can be seen in Figure 4.16. GridRoute_NM nodes do not store any information about other network agents. Actually the only destination in gateway simulation is the grid-cell of the gateway. Note that all cluster based or social contact information based protocols must store data in order to operate properly. However GridRoute can deliver up to 90% of the messages successfully with no storage.

Increase in buffer capacity allows storage of more messages before they are dropped from the buffer. As a result this feature increases the average lifetime of a message on the network which increases the probability of successful message delivery. On contrary, transmission range does not have any significant effect on delivery ratio. Because a random central cell is selected for gateway and range of 50 meters seems to be sufficient to select best candidate to forward the message.

As it can be seen in Figure 4.17 GridRoute_NM produces small amount of

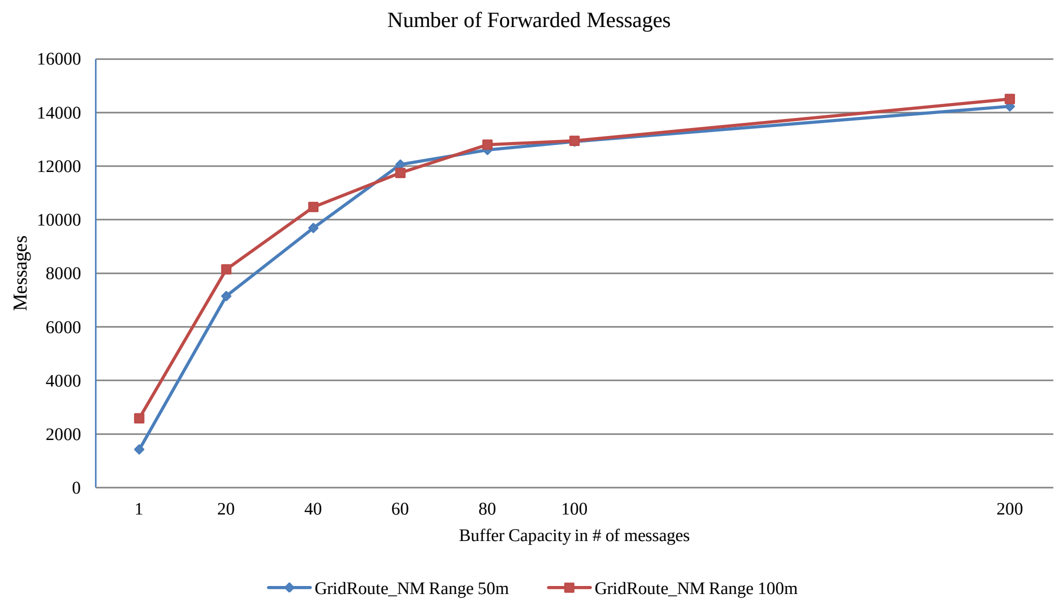

Figure 4.17: number of forwarded messages, GridRoute_NM, gateway simulation, range: 50 m, hop limit: 11.

redundant message traffic. This property of GridRoute_NM depends on two facts. Firstly, as gateway is placed a central grid, sender-gateway distance is relatively smaller when it is compared with the simulation of ordinary GridRoute. Due to locality of mobility, senders can more easily find a node that has high probability to deliver the message. Message is generally delivered in two hops. Secondly, GridRoute_NM waits to find a promising node before it performs forwarding operation. This feature of GridRoute increases the average delay but it is one the main reasons of having such a low redundant message traffic.

Buffer capacity increases the redundant message traffic due to same reasons that is discussed for ordinary GridRoute. Also transmission range do not have any significant effect owing to same reasons discussed in delivery ratio of GridRoute_NM.

GridRoute_NM has reasonable average delay as it is presented in Figure 4.18. Single copy feature of GridRoute still affects the delay in a negative way. Although the distance between sender and receiver is approximately halved when it

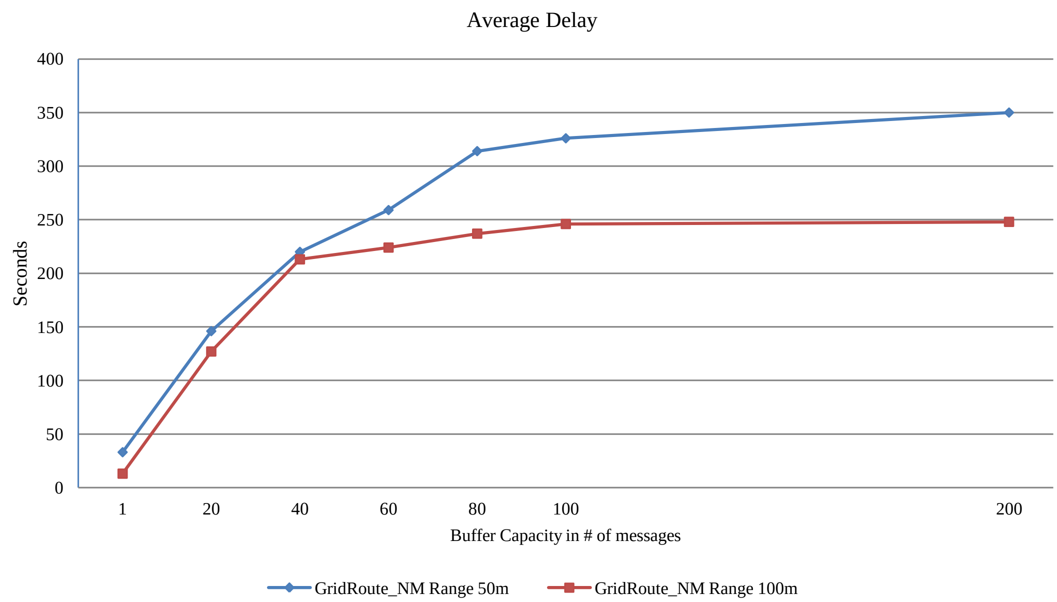

Figure 4.18: Average delay, GridRoute_NM, gateway simulation, range: 50 m, hop limit: 11.

is compared with the simulation of GridRoute, the decrease in latency is marginal. Because the bottleneck of average delay is time of waiting an encounter with a promising node that has high probability to delver the message to destination. After the message is delivered to that node, in most of the cases message is delivered to destination in a short amount of time. Decreasing the average delay is only possible by spreading the message which would increase the number of redundant messages. Thus GridRoute_NM achieves reasonable and acceptable message delays even if it can not take advantage of the smaller distance between sender and receiver.

GridRoute_NM in CBMM with random destinations is the hardest simulation case among above mentioned conditions. In this case both Epidemic Routing and GridRoute should perform efficiently in order to achieve acceptable communication in the network. As it can be seen in Figure 4.19 GridRoute_NM delivers as less as 60& messages than GridRoute when the resources are limited. However, the increase in the performance of Epidemic Routing and GridRoute effects

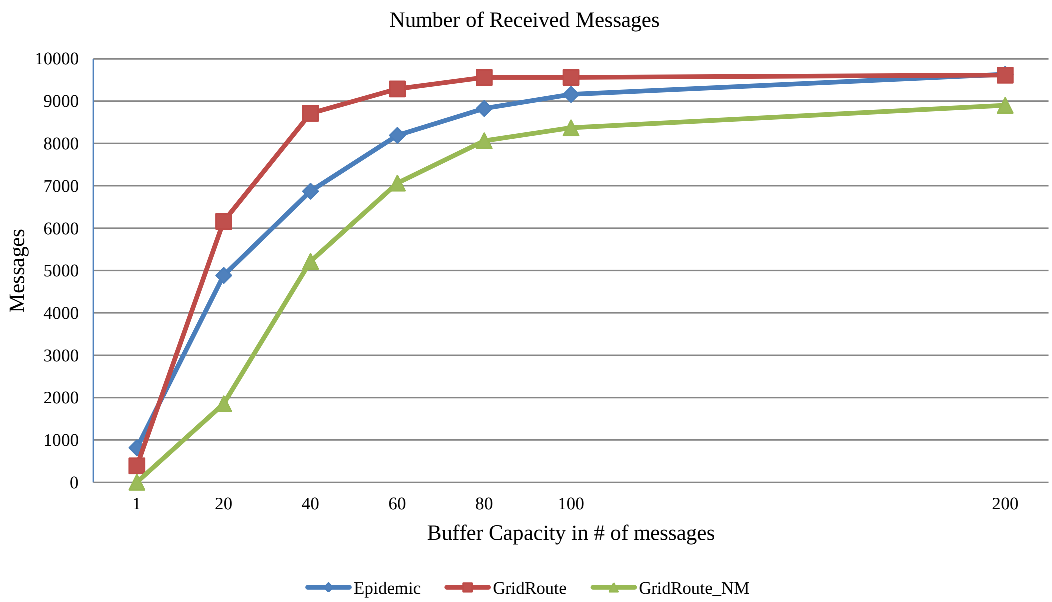

Figure 4.19: Number of received messages, GridRoute_NM, CBMM simulation, range: 100 m, hop limit: 11.

GridRoute_NM in a positive way. GridRoute_NM can deliver 90% ofthe messages with buffer size 200 messages which is about 6% less than ordinary GridRoute.

If it is assumed that the favorite grid-cells of destinations is known by the senders; similar to knowing the e-mail address of destination in regular mail transfer protocols, then the performance of GridRoute_NM will be exactly same with GridRoute. Because GridRoute is able to operate without requirement of storing any information about other nodes. This simulations shows even if the favorite grid-cell information is not known, it can be acquired using Epidemic Routing. The advantage of using GridRoute_NM rather than Epidemic Routing directly will be explained in the analysis of number of forwarded messages.

GridRoute_NM performs one Epidemic Routing and 2 GridRoute message transfer in order to deliver the message to the destination. All of the redundant messages during these 3 steps are considered as message traffic of GridRoute_NM for one message delivery. As it can be seen in Figure 4.20, GridRoute_NM performs approximately 130000 message transmission to deliver 10000 messages at

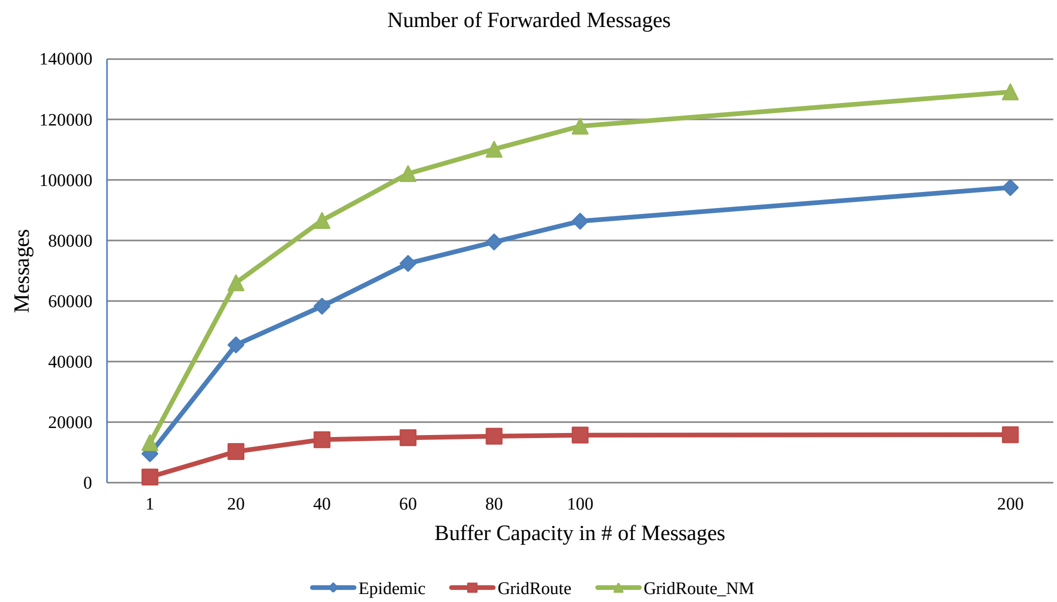

Figure 4.20: Number of forwarded messages, GridRoute_NM, CBMM simulation, range: 100 m, hop limit: 11.

the worst case. Again, the main part of this redundant message traffic is generated to receive the favorite grid-cell information. When it assumed that users knows the favorite grid-cells of destinations, redundant message traffic of GridRoute_NM will be same as ordinary GridRoute.

Different from Epidemic Routing, GridRoute_NM does not spread the message to whole network. Most of this redundant messages are the grid-cell information requests which are generally much smaller than than the actual messages. Assuming that the grid-cell information request is 10 bytes and a message is 10 KB, the total size of messages spread to network by Epidemic Routing and GridRoute_NM are as follows. Epidemic Routing spreads 99000 messages to the network which means the total size is 10000 bytes*(99000) = 990000000 bytes = 990 MB. On the other hand GridRoute_NM uses one Epidemic Routing (99000 messages approximately) and two GridRoute (2 * 19000 messages approximately) which results in total of $(2 * 19000 * 10000) + (99000 * 10) \simeq 381000000$ bytes = 380 MB.

GridRoute_NM decreases the total size of messages that are spread to network

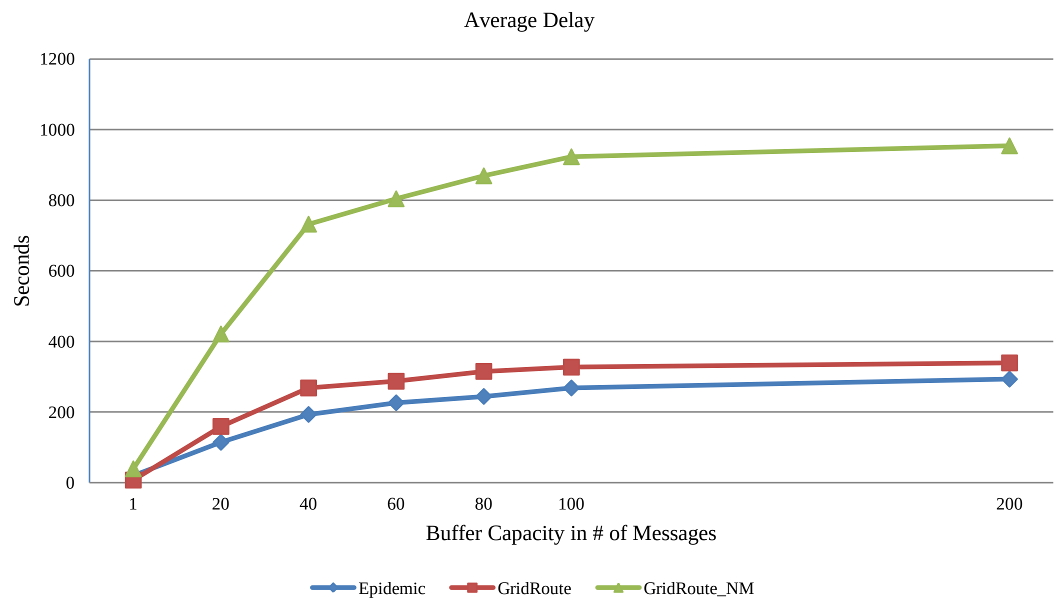

Figure 4.21: Average delay, GridRoute_NM, CBMM simulation, range: 100 m, hop limit: 11.

about 61.5%. This feature may be crucial to increase the lifetime of the network as it decreases the energy consumption. On the other hand, ordinary GridRoute spreads $19000 * 10000 = 190000000$ bytes = 190 MB of messages to the network. This will be the case in relaxed Gridroute_NM when the senders know the grid-cells of destinations. In this case, GridRoute performs 80.8% more efficient than Epidemic Routing without need of any data storage.

Average delay of GridRoute_NM is much higher than Epidemic Routing or GridRoute. Because 3 message transmissions are considered as one successful message delivery. The hot spot of delay in GridRoute_NM is getting the grid-cell information. It accounts for more than 65% of the delay in message delivery as it requires one Epidemic and one GridRoute message delivery. Once the grid-cell information is retrieved, the average delay of message delivery is same with the GridRoute. Note that the messages that can not be delivered are not included in the average delay calculation. Hence buffer capacity do not have a negative effect on average delay as it is presented in Figure 4.21. Similar with Figure 4.14 and

4.15, if a penalty is added for unsuccessfull message transmissions, increasing the buffer capacity decreases the average delay.

This figure shows the case when the favorite grid-cell information is not known. Similar with simulations of Figure 4.19 and 4.20, if it is assumed that the favorite grid-cell information is known, GridRoute can be used directly as routing protocol that does not require any data storage of other nodes. In this case, as Figure 4.21 presents, GridRoute is able deliver messages in approximately same time when it compared with Epidemic Routing. At the worst case, GridRoute delivers messages using approximately 15% more time than Epidemic Routing. When the number of redundant messages are taken into consideration, GridRoute is much more efficient than Epidemic Routing even if the average delay of it is slightly higher.

E-mail or any other proper addressing scheme can be adopted to use GridRoute with no memory requirement. In this case e-mail addresses may not be memorized by the users, thus no extra memory requirement is needed. E-mail or other addressing schemes can be obtained by memorizing or sending a request to central server can be an option. The difference from gateway simulation is that, in this case the request does not needed to be send epidemically. GridRoute can be used in each step so the redundant message traffic can be lowered significantly when it is compared with the GridRoute_NM gateway scenario.

4.10 Simulation Results of GridRoute_IR

Figure 4.22 indicates that GridRoute_IR is less efficient than ordinary GridRoute in terms of message delivery ratio. As explained in the protocol description in Sect III, no messages are deleted from buffers in GridRoute_IR until they are overwritten by FIFO buffer procedure. However, not deleting forwarded messages may cause deleting active messages that no node in the network may have an inactive copy of it. These messages are lost and can not be delivered to destination. Even if an inactive copy of these deleted messages exists, an active message is always

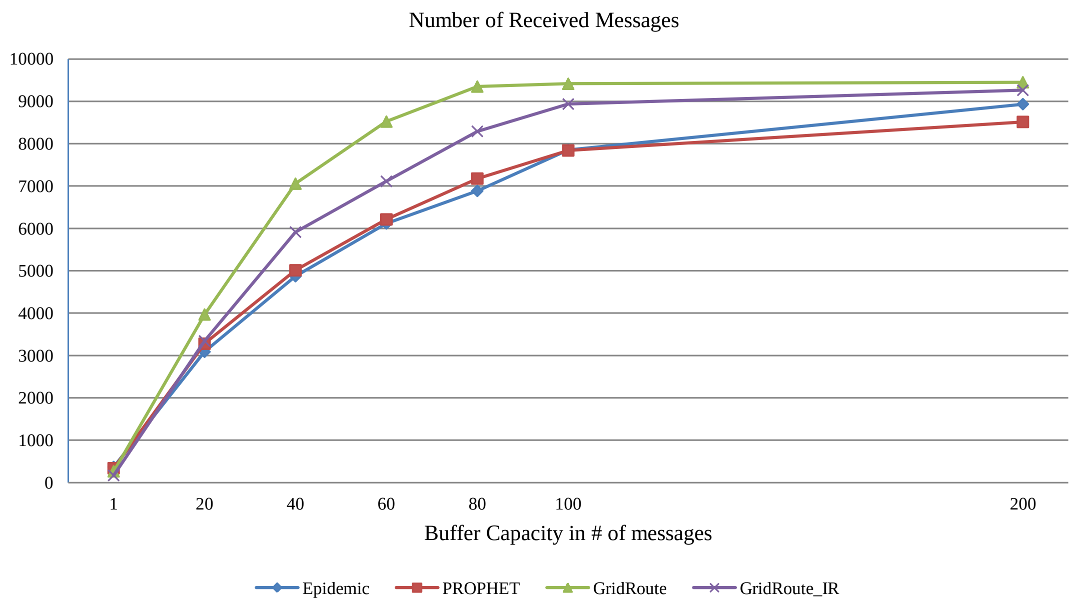

Figure 4.22: Number of received messages, CBMM, buffer size: 50 messages, node count: 50, transmission range: 100 m, hop limit: 11.

in the node that has the highest probability to deliver the message to destination among nodes that have replicas of that particular message.

As a result, allowing inactive existence of forwarded messages causes relatively ineffective utilization of the buffer spaces of nodes. Thus, compared to GridRoute, a significant portion of the messages (up to 15%) can not be delevired to their destination especially when the nodes have small buffers. In case of simulations of the small buffered nodes (buffer size of 80 or smaller) the inefficiency in the buffer usage is much higher but when there is enough buffer space like buffer size of 200 messages, inactive messages generally do not cause deletion of active ones so, the message delivery ratios of GridRoute and GridRoute_IR are nearly identical.

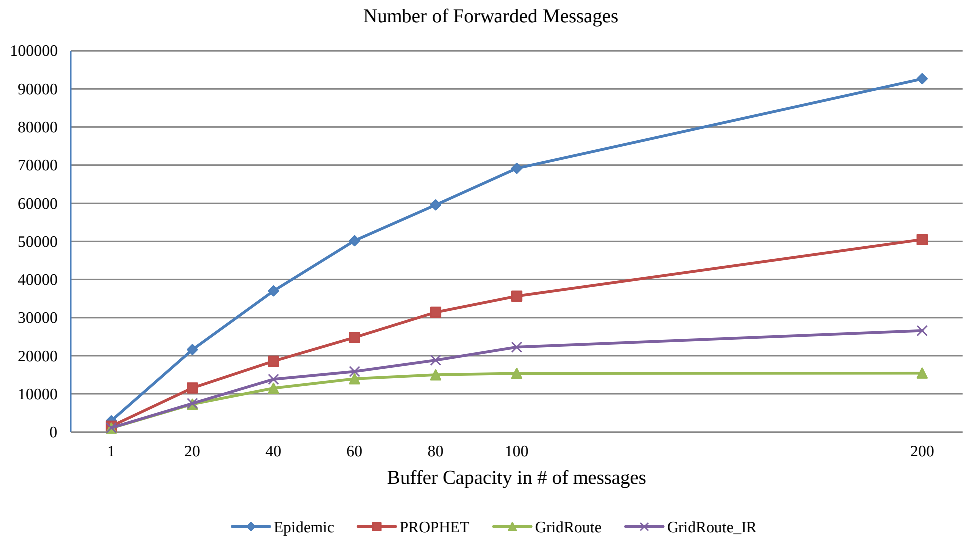

Figure 4.23: Number of received messages, CBMM, node count: 50, transmission range: 50 m, hop limit: 11.

GridRoute_IR spreads more messages to network when it is compared to GridRoute. Although it is still more efficient than Epidemic Routing and PROPHET, the inactive messages in the buffers of nodes bring extra redundant message transmission overhead. Even if the inactive messages can not be forwarded to any node other than the destination, it still possible that a multiple copies of same message can be delivered to destination. In general the inactive message holders still have a significant probability to deliver the message to destination. As a result, Figure 4.23 indicates these multiple transmissions bring an important message overhead to network up to 35%. However, inactive message replication clearly decreases the message overhead as it can be seen when GridRoute_IR is compared with GridRoute or PROPHET.

Buffer capacity has an important effect on the number of this extra message transmissions. As smaller buffered nodes can not store a lot of inactive messages, their redundant message traffic is very close to GridRoute, however, as the number of inactive messages that can be stored in buffers increases, the redundant message traffic of GridRoute_IR becomes significantly higher than the GridRoute.

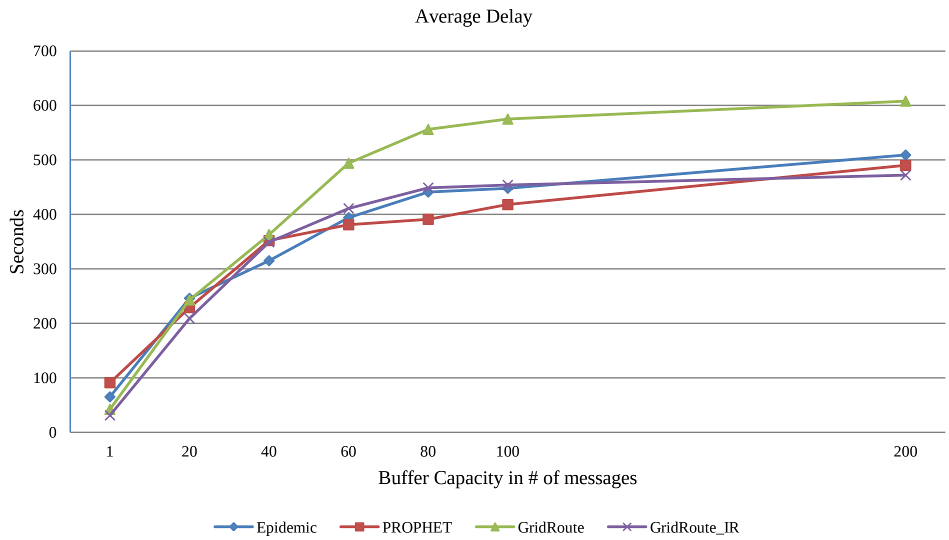

Figure 4.24: Number of received messages, CBMM, node count: 50, transmission range: 50 m, hop limit: 11.

GridRoute_IR is developed to decrease the average delay of GridRoute. Although it has relatively lower delivery ratio and higher redundant message traffic, as it can be seen in Figure 4.24, the average message delay of it is very similar to Epidemic Routing and even better when the buffer size is 200 messages. Similar with PROPHET, GridRoute_IR carries out a selective replication procedure but in different, the message replicas produce much less overhead in message traffic. Multiple copies of messages allow earlier message delivery like in other two protocols and inactive message replication allows storing more message replicas in buffers as it creates less traffic.

Generally, GridRoute_IR performs very similar to PROPHET in terms of average delay, however, when the buffer size increases, as it can store more messages in buffer due to lessened message overhead, GridRoute_IR is able to deliver the messages to nodes earlier. Excessive message traffic in Epidemic Routing in big buffer sized simulations prevents earlier message delivery as lots of message replicas are deleted from buffers before they can be forwarded to the destinations. With limited declension in the redundant traffic (35%) and delivery ratio (15%), Gridroute_IR can decrease the average delay of GridRoute up to 37%.

4.11 Simulation Results of GridRoute_TA

A different mobility is used to compare the GridRoute_TA. For details of the mobility model please refer to Section IV. In this case time information is added to GridRoute in order to increase the delivery probability of message to destination. An array of existence probabilities for each cell is compared using cosine angle separation metric. GridRoute_TA can effectively differentiate nodes that spend lots of time in the destination grid-cell of the message at the same time with destination while other three protocols are unable. Epidemic Routing does not include any network information in forwarding decision and PROPHET uses contact probabilities.

In PROPHET a node x that contacts regularly with node y in the morning can be selected as a promising node to forward the message even if the message is generated at night. Same reasoning is also valid for ordinary GridRoute. Also even if the nodes has high existing probabilities in the same grid-cell, they may visit those grid-cells in different times. GridRoute_TA can efficiently handle these situations and as a result number of received messages of GridRoute_TA is more than the compared protocols. Approximately, gains of 25%, 50% and 33% can be achieved when it is compared to GridRoute, Epidemic Routing and PROPHET respectively.

The buffer capacity does not effect the ratio between GridRoute and GridRoute_TA, however it has a significant effect on ratio between replication based protocols. Epidemic Routing and PROPHET are able to deliver more messages with the increase in buffer capacity so their the gap of delivery ratio decreases as the buffer size becomes bigger as bigger buffers can handle excessive message overhead. However, until buffer size is close to unlimited, GridRoute_TA is able to deliver more messages than these two. Also the ratios between delivery rates of protocols are very similar when the node density or transmission range is altered. As a result, GridRoute_TA is able to deliver much more messages than compared protocols.

There is not much difference between GridRoute and GridRoute_TA in terms

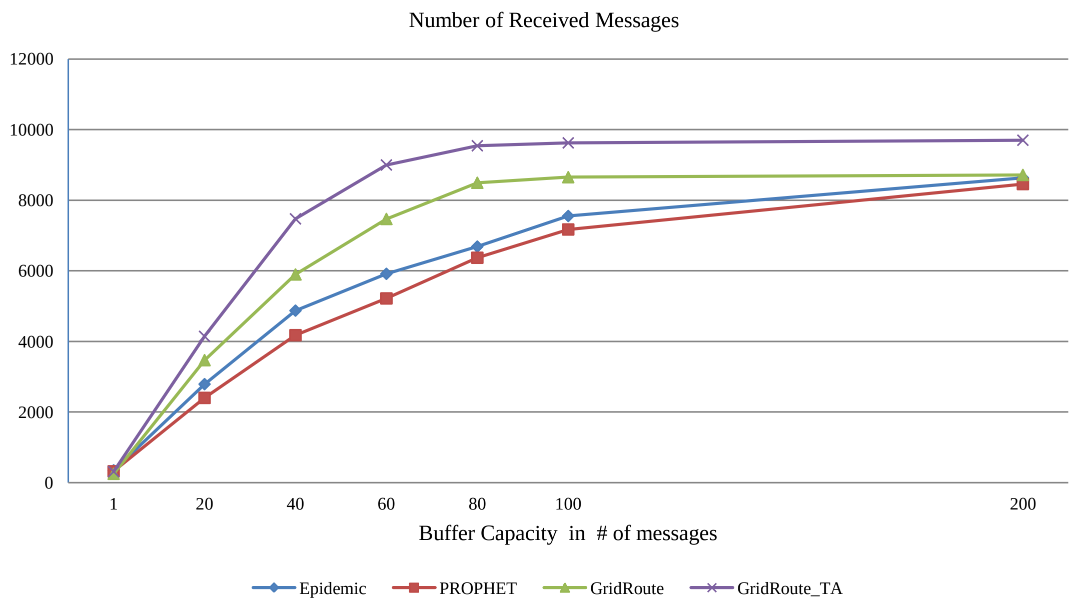

Figure 4.25: Number of received messages, CBMM, node count: 50, transmission range: 50 m, hop limit: 11.

of total number of forwarded messages as it can be seen in Figure 4.26. Both protocols are single copy and operate based on location informations. Only difference occurs in the node selection for forwarding messages. As GridRoute_TA chooses nodes more conservatively, there is a little decrease in the number of forwarded messages less than 5%. However GridRoute also chooses nodes carefully and generally does not forward messages to nodes that do not have good probability to deliver the message. The main difference between these two protocols is their average message delay which will be discussed below.

The comparison of GridRoute_TA to PROPHET and Epidemic Routing is nearly identical to comparison of ordinary GridRoute with these two protocols even if the simulation model is changed. For detailed analysis please refer to explanations of Figure 4.10 and 4.11 in this section.

Figure 4.27 indicates that, GridRoute_TA has much lower average message delay. It is the only protocol that takes time into account. GridRoute only considers the location information and does not interpret the time of the message

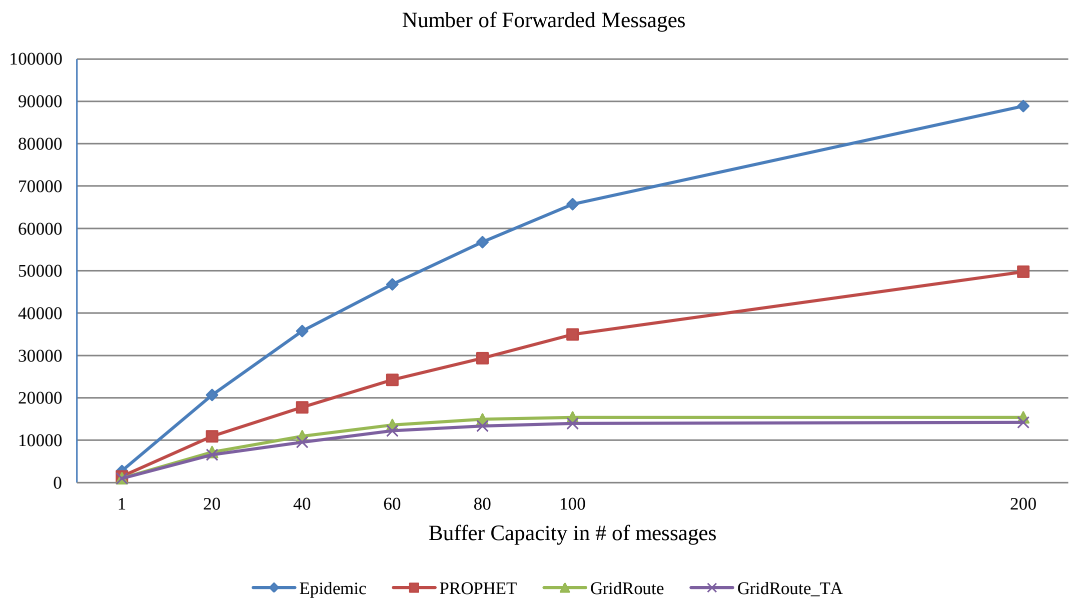

Figure 4.26: Number of received messages, CBMM, node count: 50, transmission range: 50 m, hop limit: 11.

delivery or whether forwarded node and destination can meet in the grid-cell even if they spend lots of time in it. Similarly, PROPHET does not considers whether two node can meet in later or in near future. On the other hand, in GridRoute_TA, a node forwards a message only if a neighbor has higher probability to deliver the message in the near future. Apart from small decrease in number of redundant messages, this selection criteria of GridRoute_TA creates an avalanche effect and the messages are delivered to their destination in shorter times.

Avalanche effect can be expressed as this. Whenever a message is forwarded to a node that can deliver the message to the destination by comparing higher level grid-layers, due to locality of mobility finding a more promising node or meeting with destination significantly increases. In simulations, most of the time in message delivery of GridRoute_TA is spent in the period until the first message transfer occurs. Once the message is forwarded to a promising node, that node generally travels towards the destination grid-cell of the message and message is delivered or forwarded to a more promising node in short amount of time which is not the case for other three protocols. Due to this, average delay of GridRoute_TA is significantly lower than ordinary GridRoute, PROPHET and Epidemic Routing.

Increasing the buffer capacity effects all three protocols positively even if the average delays are seem to be increasing. Again the undelivered messages are not included in the average delay calculation. More buffer space allows increasing message delivery ratio and allows storing messages in the buffers for longer times. As a result, the average delay increases for all four protocols but if a penalty is added for undelivered messages it would be seen that the average delay actually drops in all protocols and as GridRoute_TA able to deliver much more messages, the difference between GridRoute_TA and other three protocols would be much bigger. Buffer capacity nearly has a similar effect on all of the protocols so the ratio between average delay of these protocols does not change much when the buffer capacity is increased.

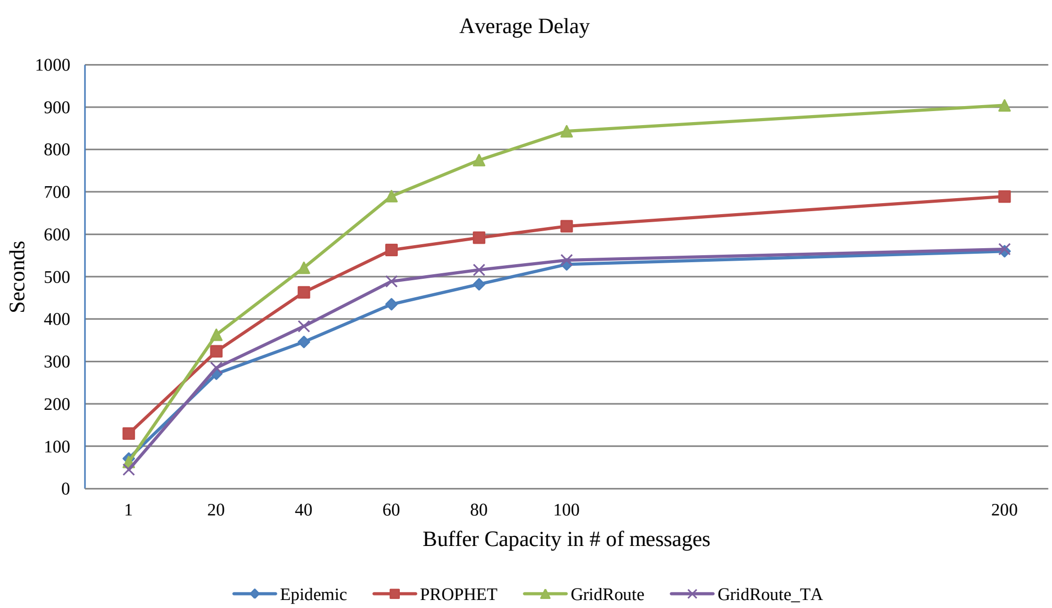

Figure 4.27: Number of received messages, CBMM, node count: 50, transmission range: 50 m, hop limit: 11.

4.12 Simulation Results Under Light Load

All of the protocols other than these simulations are compared in an environment with high traffic load. It is important to test the protocols with high message generation rate in order to point out their differences, however their performance under light load is also important.

Figure 4.28 presents delivery ratios of compared protocols. A total of 400 messages are generated and it is evident that all of the protocols works quite efficiently. Epidemic Routing is able to deliver all of the messages in almost all cases. Low message generation rate limits the excessive redundant message traffic of it so the messages can be stored in buffers until their delivery to destination.

Although PROPHET seems to have poor performance, its delivery ratio is only 5% less than Epidemic Routing at the worst case approximately. The performance of GridRoute and GridRoute_IR are almost identical to Epidemic Routing and the inactive message replication of GridRoute_IR provides a small increase in delivery ratio.

The comparison of light load with high load scenarios points out that, in DTNMs the main reason of unsuccessful delivery is the ineffective usage of the buffers. As GridRoute and its variations use buffers space very efficiently, they shows better performance in high-load simulation scenarios relative to compared protocols.

Epidemic Routing spreads messages to all nodes in the network when the message generation rate is low. Figure 4.29 shows that even if the nodes has buffer space of 10, as Epidemic Routing is able deliver messages less than 50 seconds as 1 message is generated in each five second. As a result all of the nodes in the network get a copy of the message and 20000 message traffic is generated by Epidemic Routing in order to deliver 400 messages. Although it is able to deliver lots of messages in short amount time, light traffic load increases the redundant message traffic ratio of Epidemic Routing.

PROPHET limits the message replication with its selective replication bu

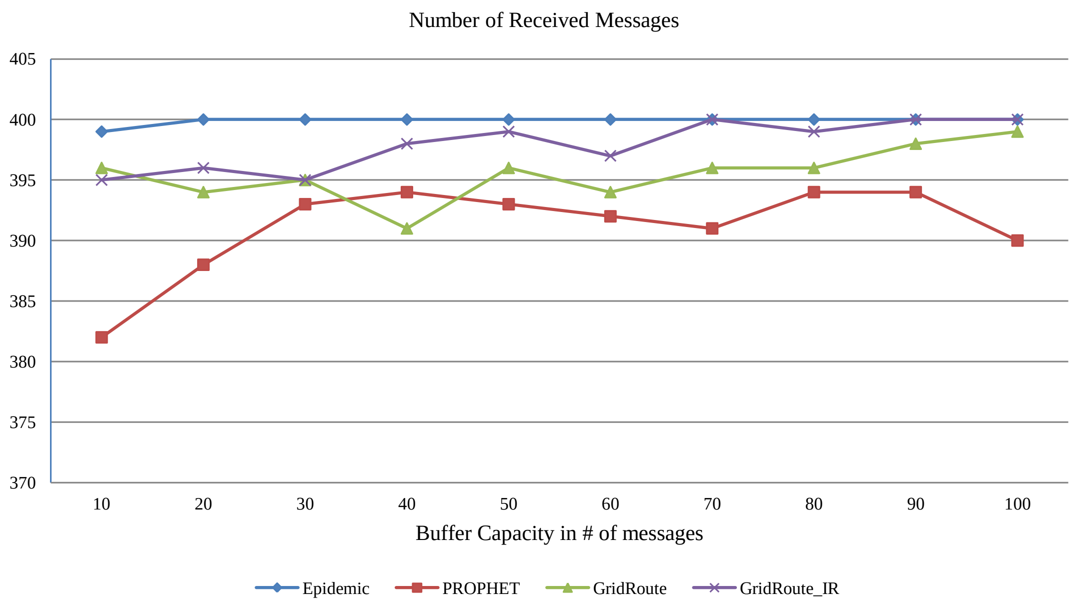

Figure 4.28: Number of received messages, CBMM, node count: 50, transmission range: 100 m, hop limit: 11.

it still generates lots of messages due to similar reasons to Epidemic Routing. On the other hand, single copy GridRoute is able to deliver messages approximately 3-4 hops and its message traffic is approximately constant at 1200 messages. Although it is not evident on Figure 4.29 GridRoute_IR generates slightly more messages than GridRoute as multiple messages are delivered to destination. However, even in this case a total of less than 1500 messages are generated by GridRoute_IR.

Average message delay of compared protocols are presented in Figure 4.30. Epidemic Routing has a very low delay and is able to deliver messages in 25 seconds approximately. Light message load allows it to spread the messages to network easily so the messages can be delivered in small amount of time. Similarly, redundant message traffic of PROPHET is quite high. However as mentioned earlier, excessive redundant message traffic does not cause message drops from buffers. Hence, average delay of PROPHET is very similar to Epidemic Routing and it can deliver messages to destination in approximately 40 seconds.

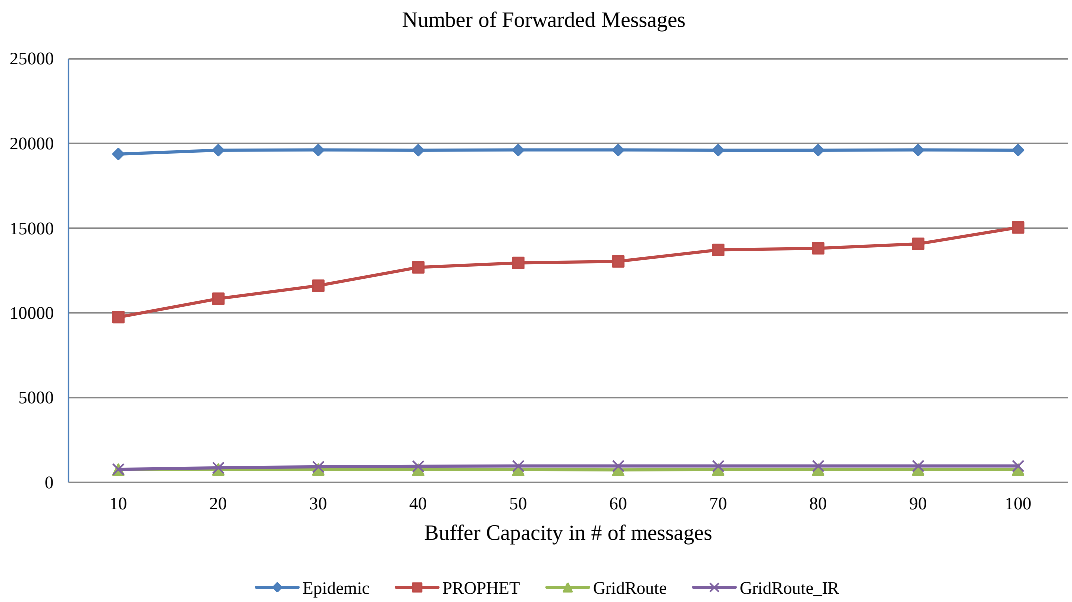

Figure 4.29: Number of received messages, CBMM, node count: 50, transmission range: 100 m, hop limit: 11.

On the other hand, GridRoute has a high message delay as it waits for the contact of one message copy with the destination node. Even it take advantage of locality of mobility, as PROPHET and Epidemic Routing spreads message copies to almost all of the nodes, average delay of GridRoute is much higher. GridRoute_IR provides a significant improvement. Simulation results show that some inactive message replicas are delivered to the destinations before the active copy. This is due to random movement possibility of the mobility model. As buffer spaces become bigger, GridRoute_IR is able store more inactive messages in the buffers so the average delay of it decreases significantly up to some point (up to buffer size of 70 messages). GridRoute_IR delivers messages using 3 times more time than Epidemic Routing but it generates approximately 15 times less redundant messages.

Other than redundant message traffic, compared protocols work more efficiently than GridRoute under light load. However, in real world, hot spots like restaurants or shopping malls has potential to cause excessive message exchange between nodes. As a result, even if the network has low message generation rate in general, in some parts of the network routing protocols may need to handle lots of new coming messages and this places may be the bottleneck of the network if the protocols can not work efficiently under high message traffic. Thus it is important have a routing protocol that can handle excessive message traffic.

Apart from traffic load or comparison of protocols under given network conditions, parameters of the network have important effect on the simulation results. The effect of hop limit, node density, message generation rate and transmission range are analyzed below:

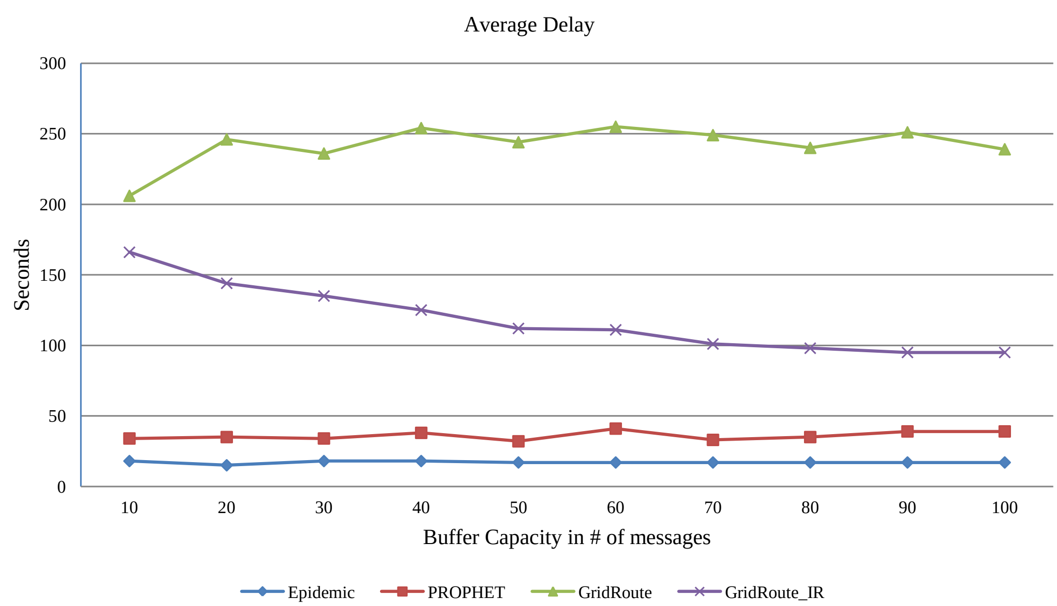

Figure 4.30: Number of received messages, CBMM, node count: 50, transmission range: 100 m, hop limit: 11.

4.13 Effect of Hop Limit

Figure 4.31 points out that, hop limit has a positive effect on each protocol in terms of message delivery up to some point. In general GridRoute succeeds to deliver the message in 3-4 hops. As a result increase in the hop limit after that does not have any significant effect. However it is important allow certain number of message transmissions. Comparison of hop limit 1 with hop limit 5 stress out the importance of this as the delivery ratio is more than doubled at the latter case.

Logically, Epidemic Routing and PROPHET should perform better with the increase in the hop limit as they spread the messages throughout the network. However increase in the redundant messages overwrites undelivered messages in the buffers so after hop limit of 3, delivery ratio effected negatively from the increase in hop limit for Epidemic Routing. For PROPHET, after hop limit of 4 it can not take advantage of more forwarding opportunity as it selectively spreads the messages and it spreads the message to all of the promising candidates in hop limit of 4.

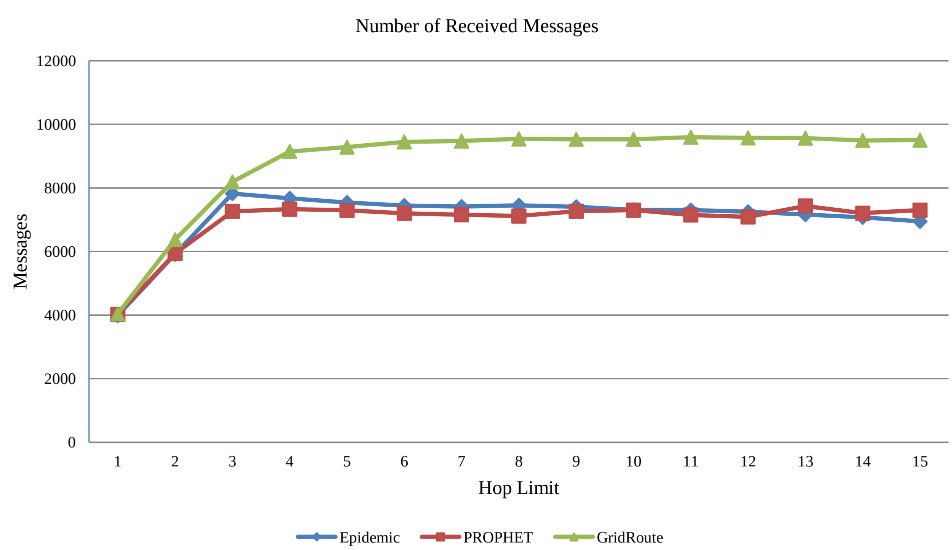

Figure 4.31: Number of received messages, CBMM, node count: 50, transmission range: 50 m.

Number of forwarded messages has different effect on compared protocols. Figure 4.32 indicates that the number of redundant messages is effected by the hop limit in proportional with the message spread ratio. The full spreading based Epidemic Routing approximately respond the increase in hop limit linearly in terms of number of forwarded messages. More hop limit allows Epidemic Routing to spread the message to greater number of nodes which also creates a avalanche effect on total number of forwarded messages.

PROPHET is effected by the hop limit limitedly. Although it is able to spread messages more with the increase in hop limit, its selective spreading mechanism limits the number of replicas in the network. On the other hand, hop limit does not have any significant effect redundant message traffic of GridRoute. The main reason for this is the single copy nature of GridRoute. Increasing the maximum allowable transmission count does not spread the message to greater number of nodes but may cause some few more forwarding in the network so almost no effect of hop limit can be seen on redundant message traffic of GridRoute.

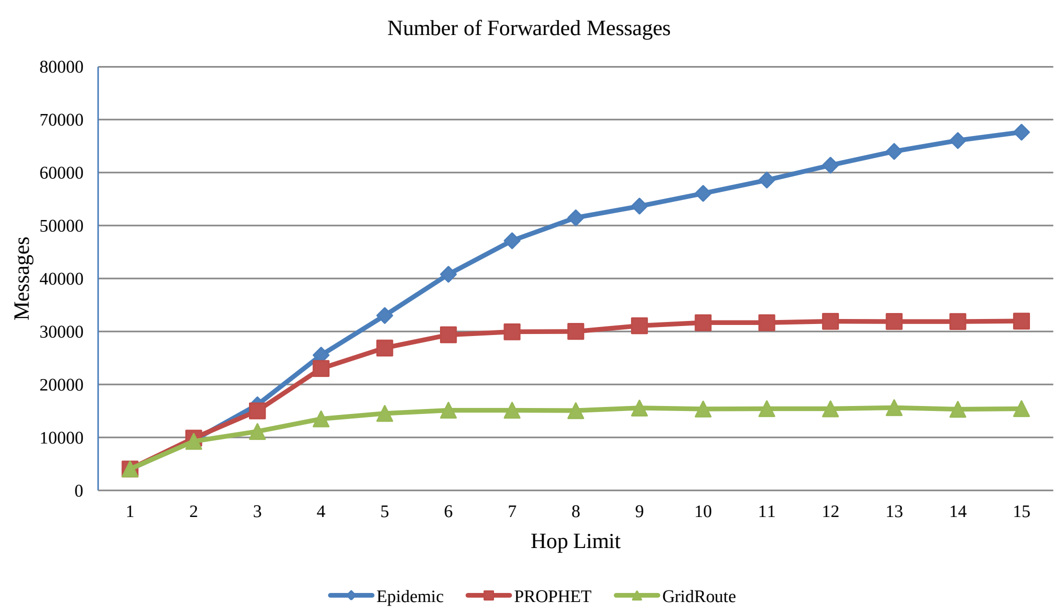

Figure 4.32: Number of forwarded messages, CBMM, buffer size: 50 messages, node count: 50, transmission range: 100 m.

Epidemic Routing and PROPHET have nearly identical average delays in terms of effect of hop limit. Epidemic Routing spreads the message to a lot of nodes including promising nodes (the ones that has high probability to deliver the message to destination) and in PROPHET more promising nodes gets the replica of message with the increase in hop limit. As a result the average delay is decreases up to certain point where the spread of message is enough to deliver the message in lower bound (in fastest time possible for given network).

GridRoute has a similar pattern but it has higher average delay values. The reason is the same with other average delay graphs in this thesis, namely the single copy nature. Increasing the hop limit allows GridRoute to deliver messages to more promising nodes. As hop limit does not have any significant effect on redundant message traffic, a high hop limit is preferable for GridRoute as it decreases the average delay significantly.

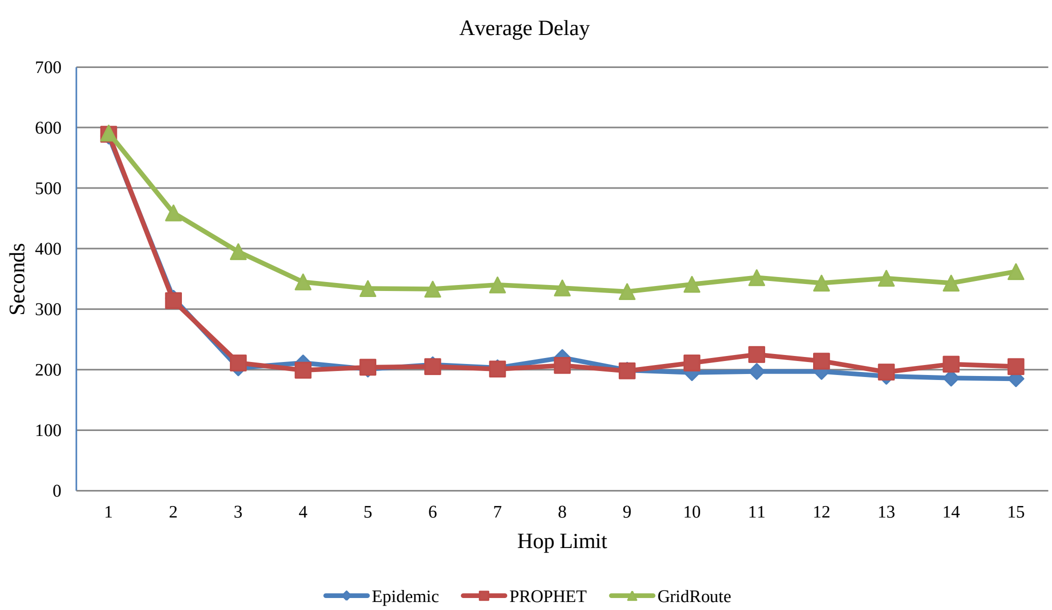

Figure 4.33: Average delay, CBMM, buffer size: 50 messages, node count: 50, transmission range: 100 m.

4.14 Effect of Transmission Range

Figure 4.34 indicates that, transmission range has a positive effect on each three protocol in terms of delivery ratio. Spreading based protocols are not very effective in limited communication range. As they can not spread the message easily through network but replicate it locally, before the message is delivered to the destination, it is overwritten by newly generated ones. Thus in simulations with transmission range of less than 50 meters, PROPHET and Epidemic Routing are not very efficient in terms of delivery ratio. However, increasing the range allows them to deliver more messages successfully as one message more easily can be spread to the whole network.

Conversely, GridRoute can operate effectively in limited communication range case. It nearly delivers 100% more messages than other two protocols with transmission range of 40 meters. GridRoute always tries to bring the message to favorite grid-cell of destination hence delivery ratio is less effected from communication range when it is compared with spreading or contact probability based protocols. Spreading based protocols can not spread the message effectively in low transmission range and there are not much contacts between nodes. As a result, the contact probability calculation is effected for protocols like PROPHET when the communication range is decreased. However, communication range does not have any effect on location or areal probabilities. Only important case in GridRoute is the number of promising nodes in the communication range. Smaller range decreases the number of nodes that has higher exsiting probability of the destination grid-cell of message. As a result it is harder to find a promising node for GridRoute in small communication range case. However, as areal probabilities are not effected, the effect of transmission range is limited when it is compared with Epidemic Routing and PROPHET, and GridRoute can deliver much more messages than other two in limited communcation range case.

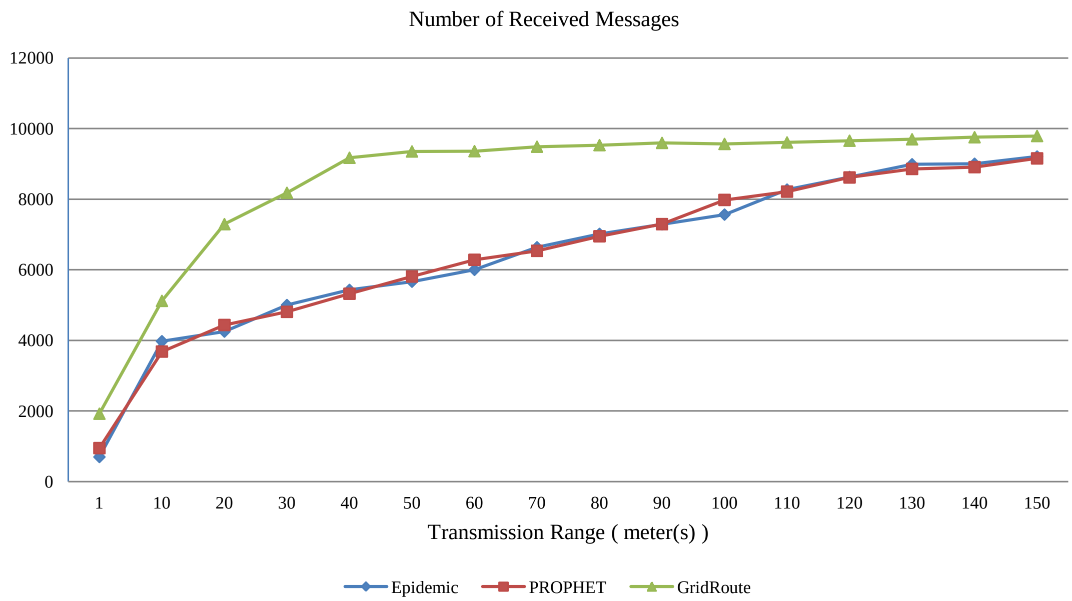

Figure 4.34: Number of received messages, CBMM, buffer size: 50 messages, node count: 50, hop limit: 11.

The difference in number of forwarded messages against change in transmission range is similar to the effect of hop limit. Bigger transmission range allows nodes to have more neighbors in their communication range. By this way in a single forwarding session, Epidemic Routing can infect much more nodes with bigger communication range. As a result, number of redundant messages generated by Epidemic Routing increases linearly with the increase in the size of the transmission range. Similar reasoning is valid for PROPHET. Only difference of it from Epidemic Routing is its selective replication process. By this way it is able to limit the number message replicas in the network relatively.

GridRoute is not effected significantly from transmission range in terms of number of redundant messages. Other than it is a single copy protocol, Figure 4.35 indicates that, range of 30 meters is enough for GridRoute to find a promising node to forward the message. Bigger range beyond 30 meters does not bring any extra message overhead so GridRoute has a nearly stable redundant message performance.

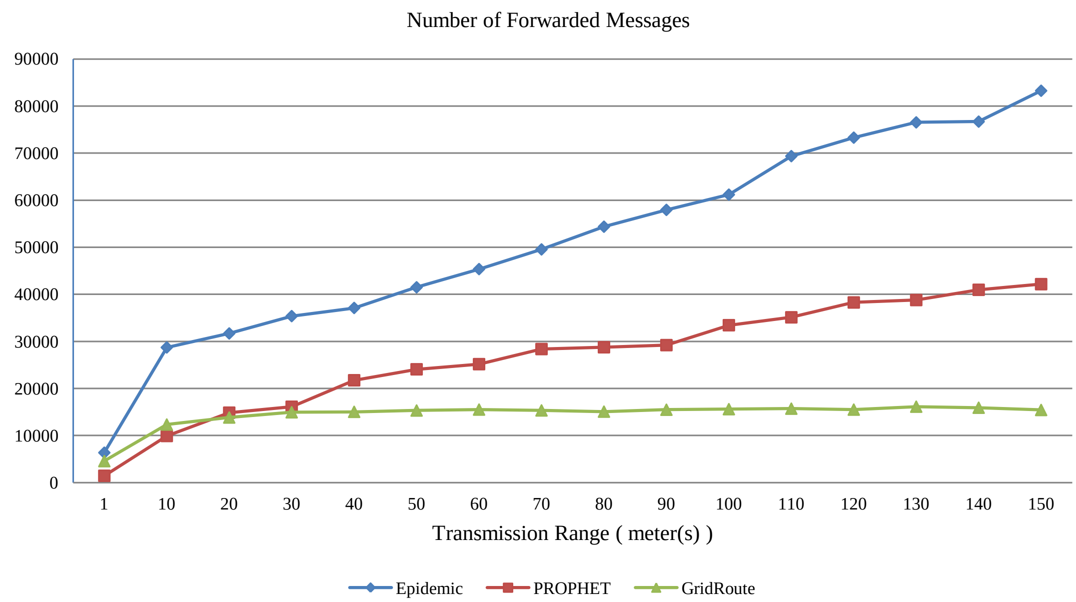

Figure 4.35: Number of forwarded messages, CBMM, buffer size: 50 messages, node count: 50, hop limit: 11.

Transmission range has positive effect on average delay for each three protocols as it is indicated in Figure 4.36. Messages can more easily be spread to network. Note that the width of the simulated network is 1600 meters. With transmission range of 10 meters, at the best case (in shortest time) it requires 160 message transfers or 160 seconds to send the message to the other end of the network. However, with a range of 150 meters, message can be sent there in 10 seconds. PROPHET and Epidemic Routing takes advantage of this and their average delay decreases with the increase in transmission range.

Transmission range decreases the average delay of GridRoute for a different reason. Although transmission range of 30 meters is enough for GridRoute to find a path between sender and receiver in most of the cases, increasing the range allows finding this path faster as one node has much more neighbors in a bigger communication rage. Moreover the probability of encounter of message holder with the destination increases as the message holder covers bigger network area when its communication range is increased.

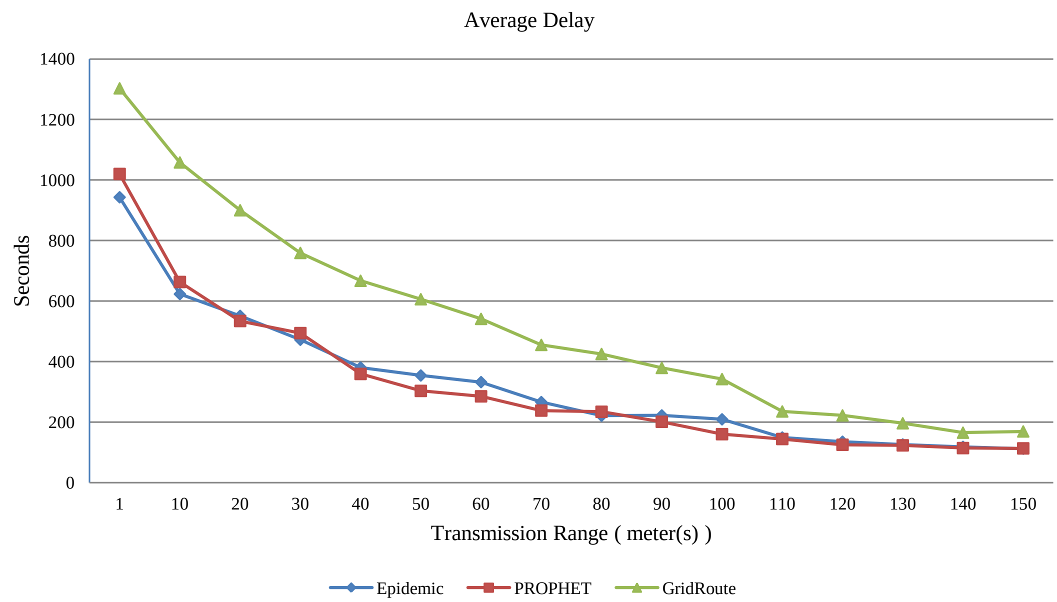

Figure 4.36: Average delay, CBMM, buffer size: 50 messages, node count: 50, hop limit: 11.

4.15 Effect of Node Density

Node density has a positive effect on deliver ratio. Less than 20 nodes on network area makes the node density so scarce that, end to end path between sender and receiver is generally not possible through time and the messages are overwritten on on the buffers before they can be delivered. Increasing the density allows finding a more promising node more easily.

PROPHET and Epidemic Routing spread the message easily in dense network. Moreover, the hop limit limits the message overhead caused by more message spread. As a result, increasing the node count allows delivery of more messages to destination.

GridRoute is better than these two in all cases. The difference is more evident in scarce network case as GridRoute is able to cover the locality of human mobility. This principle allows GridRoute to work relatively very effectively in constrained

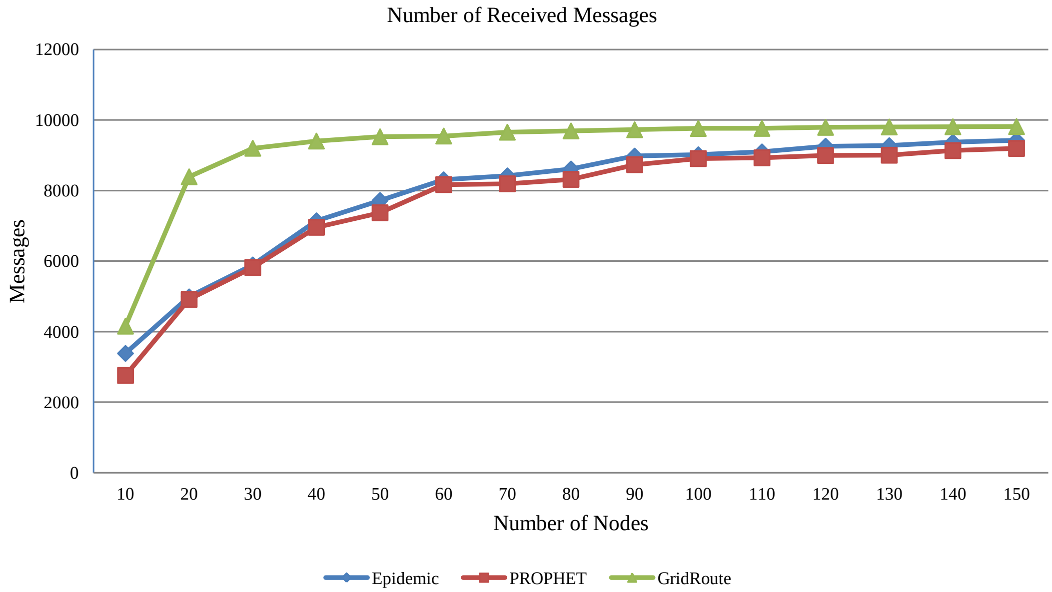

Figure 4.37: Number of received messages, CBMM, buffer size: 50 messages, transmission range: 100 m, Hop Limit: 11.

environments such as low node density or memory capability.

Increasing the node count allows delivery of more messages again due to increase in network connectivity. However denser networks causes a disadvantage in GridRoute as it described in average delay analysis of the effect of node density.

Number of forwarded messages vs. number of nodes is a very important performance metric which provides an insight about the scalability of the protocols. As it can be seen in Figure 4.38, even the number of generated messages is fixed, redundant message traffic of compared protocols varies significantly.

Epidemic Routing tries to spread the message to each node so increasing the node count brings extra overhead for each message. As a result, more node density increases the redundant message traffic linearly. Nearly 200000 messages are generated to deliver 10000 messages in the simulation with 150 nodes. In a real scenario, increasing the node count most probably increases the number of generated messages so the message redundancy of Epidemic Routing would be more severe.

PROPHET has a linear dependency to node count in terms of redundant message traffic. However the slope of PROPHET is much smaller when it compared with Epidemic Routing. Like in most cases, the decline is due to selective replication of PROPHET. When the nodes in network is increased, PROPHET just spreads the messages to some number of nodes where the Epidemic Routing tries to spread to all of them.

Figure 4.38 indicates that, GridRoute is not effected from the node density in terms of redundant message traffic. GridRoute does not spread a message but more nodes may increase the hop count of a message between sender and receiver. As it is depicted in Figure 4.38, after node count of 40, even the hop count of individual messages are nearly fixed so GridRoute has a fixed redundant message traffic against node density.

These comparisons point out that, GridRoute is very scalable. Unlike with Epidemic Routing or PROPHET, GridRoute can effectively be used in big networks with thousands of nodes with no extra overhead.

Effect of node density to average message delays is depicted in Figure 4.39. It seems that increasing the number of nodes also increases the time needed to deliver the messages. However, this is not the case. Because the average delay calculation does not include undelivered messages. The delivery ratios of each 3 protocols is relatively smaller when the node density is low. If a penalty is added for undelivered messages, it will be seen that the delivery ratios is decreases when there are more nodes in the network.

Epidemic Routing and PROPHET gain advantage of easier message spread while suffering from the increase in waste of buffer space. As a result, they have limited gain from increase in node density in terms of average delay in message delivery.

Additionally, as it can be seen in Figure 4.39, the average delay of GridRoute increases after node count of 30, even if the delivery ratio of it nearly stable. The

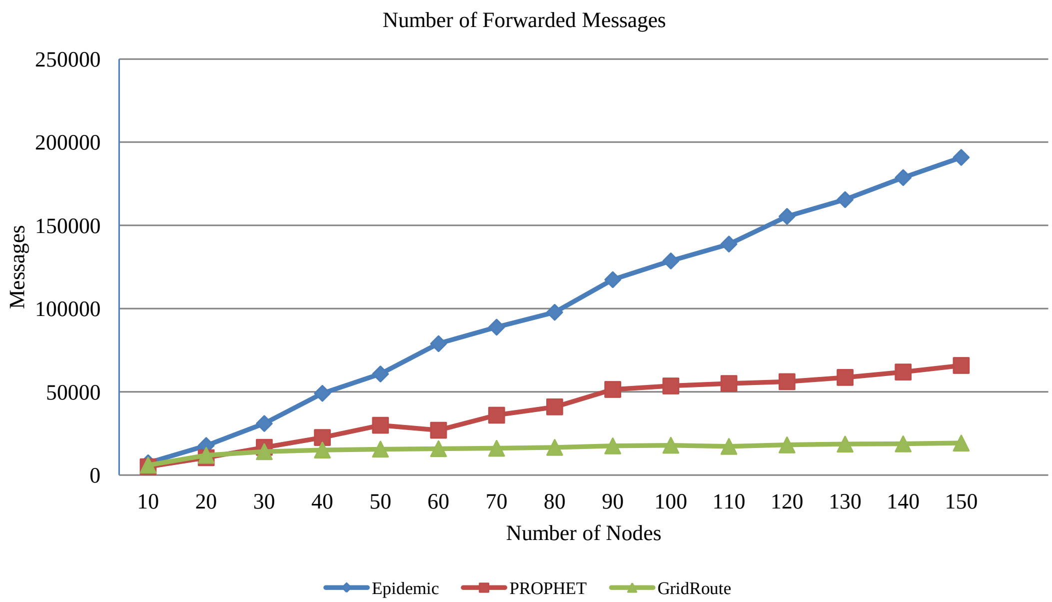

Figure 4.38: Number of forwarded messages, CBMM, buffer size: 50 messages, transmission range: 100 m, hop limit: 11.

reason for this is the increase in count of nodes that has marginally higher existence probability in the destination of the message. In the simulations with few nodes, when a message is forwarded with GridRoute, the forwarded node generally have significantly higher existence probability with the destination. However, increasing the node count also increases the standard deviation of existence probabilities of nodes for a given grid-cell. As a result, a node more probably meets with a node that has marginally bigger existence probability in the destination of message and forwards the message to that node. As a result it lost the opportunity to deliver the message to more promising node that it may encounter later. As a result, the average delay of GridRoute increases slightly (approximately 16%) with the increase in node density.

Augmenting GridRoute with a threshold in message delivery in existence probability comparison may be good solution for this situation. In this case a node

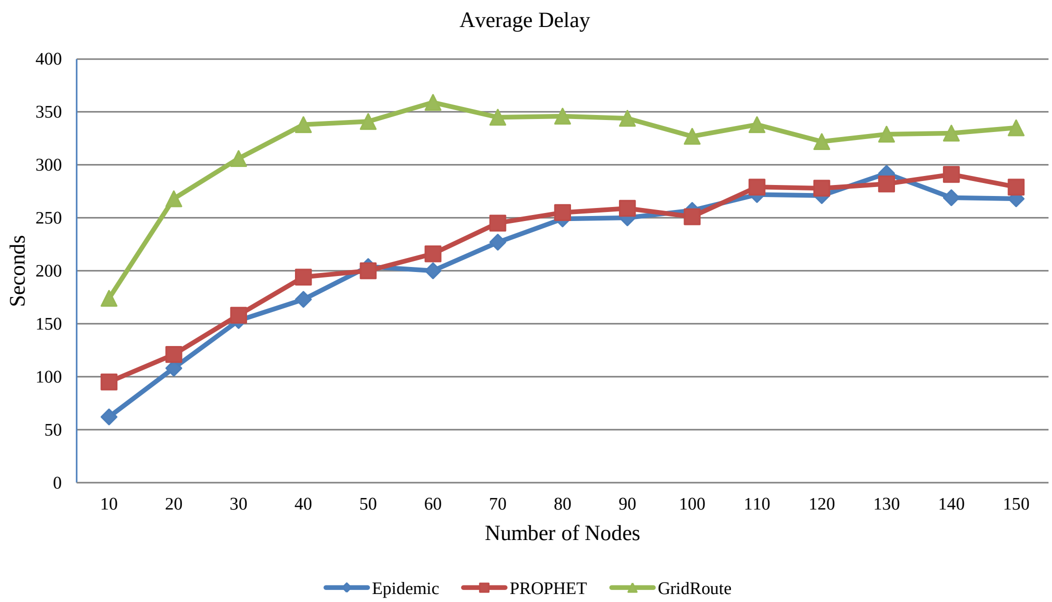

Figure 4.39: Average delay, CBMM, buffer size: 50 messages, transmission range: 100 m, hop limit: 11.

would forward the message only if a node in its communication range has threshold amount of more probability to visit the destination grid-cell of message. However, determining the value of threshold is very related the with dynamics of the network like node density and mobility and there is a marginal increase in average delay due to this effect. Hence, not using the threshold may be a better option in DTNs where the dynamics of the network are very unsteady in general.

Mobility is the main tool in DTMNs to deliver the messages. As a result, mobility is one of the most influential parameters that effects the performance of routing protocols. Figure 4.40 indicates that all three protocols take advantage of increase in node speed. A node using Epidemic Routing with speed of 20 m/s is able traverse the 400 m x 1600 m network area easily and can infect lots other nodes during this time. Although this process increases the buffer usage, most of the messages are delivered to destination before they get overwritten.

Similar reasoning is also valid PROPHET. Limited number of replicas can travel in the network faster and the average number of node contacts increases when the nodes become more mobile. Increasing contact frequency limits the effect of buffer overflows as the messages are delivered before overwritten. As a result, the delivery ratio of PROPHET increases with the node speed.

GridRoute also takes advantage of more mobile nodes. Even if there is no message spread in GridRoute, the time a node stores the message in buffer decreases, as it encounters a more promising node in shorter amount of time. Also, similar to other two, increase in node contacts pave the way for finding a promising node for forwarding. Note that GridRoute reaches its maximum delivery ratio when the node speed is about 5 m/s. 5% of data drop rate in simulations precludes GridRoute to achieve 100% delivery ratio as it is a single copy routing protocol.

It is important to point out that, node speed may effect the performance of these protocols negatively in real life. In the simulations it is assumed that if two nodes are in the communication range of each other, they can send messages instantly without any transmission or calculation delay. In real life, very mobile nodes can go out of the transmission range before they retrieve the messages and increasing the mobility of nodes raises the change of this situation. However, in the simulations this case is ignored.

4.16 Effect of Node Speed

As it is claimed in delivery ratio analysis of effect of node mobility, increase in node speed allows easier message spread in PROPHET and Epidemic Routing. As a result, the redundant message traffic of these two protocols increases when there are more mobile nodes in the network.

Naturally, unlimited message replication of Epidemic Routing is effected severely from the mobility whereas the this effect is more limited in PROPHET. As it can be seen in Figure 4.41, number of forwarded messages of PROPHET relatively stable after node speed of 14 m/s, however the redundant message traffic of Epidemic Routing keeps increasing.

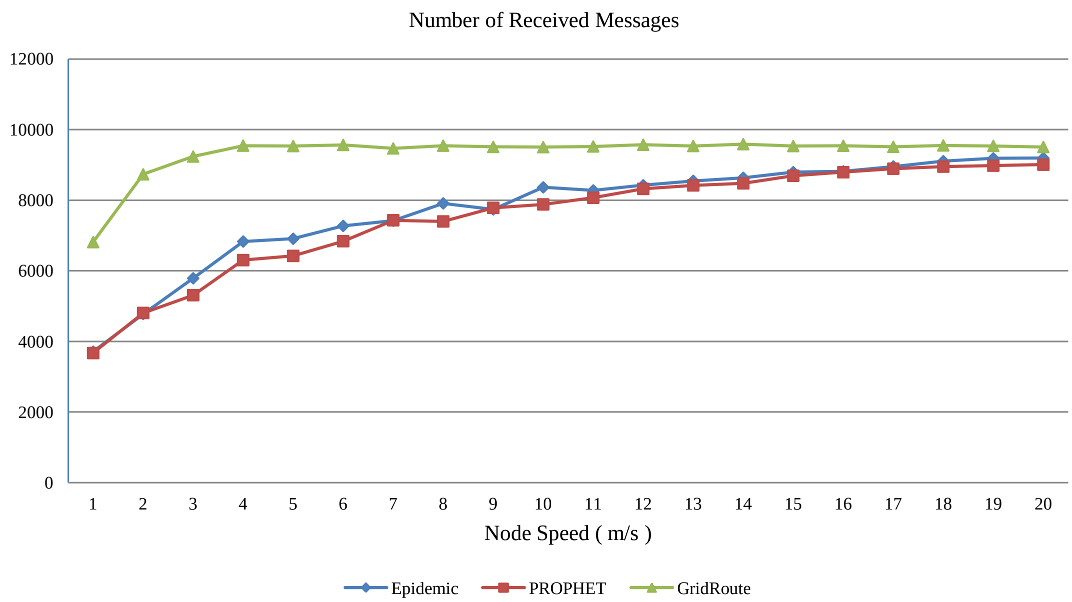

Figure 4.40: Number of received messages, CBMM, node count: 50, buffer size: 50 messages, transmission range: 100 m, hop limit: 11.

GridRoute is not affected from the mobility in terms of total number of forwarded messages. As it is stated before, increasing the mobility only allows faster forwarding in relay nodes but does not bring any significant extra message overhead mainly due to single copy nature. This feature is another advantage of GridRoute that allows efficient data communication with no extra message overhead that is caused by change in node mobility. Also this indicates that GridRoute can operate more stable in networks where nodes has different mobility degrees.

Comparison of Figure 4.36 and Figure 4.42 shows that node speed is as important as transmission range in terms of average delay. Actually increasing one of this two parameters provides the same functionality.

Increasing the transmission range allows transmitting messages to greater number of nodes at a given time but if the node mobility is low encountering a new node may take long time. On the other hand if a node has high mobility even if it has small communication range, it will be able communicate new nodes

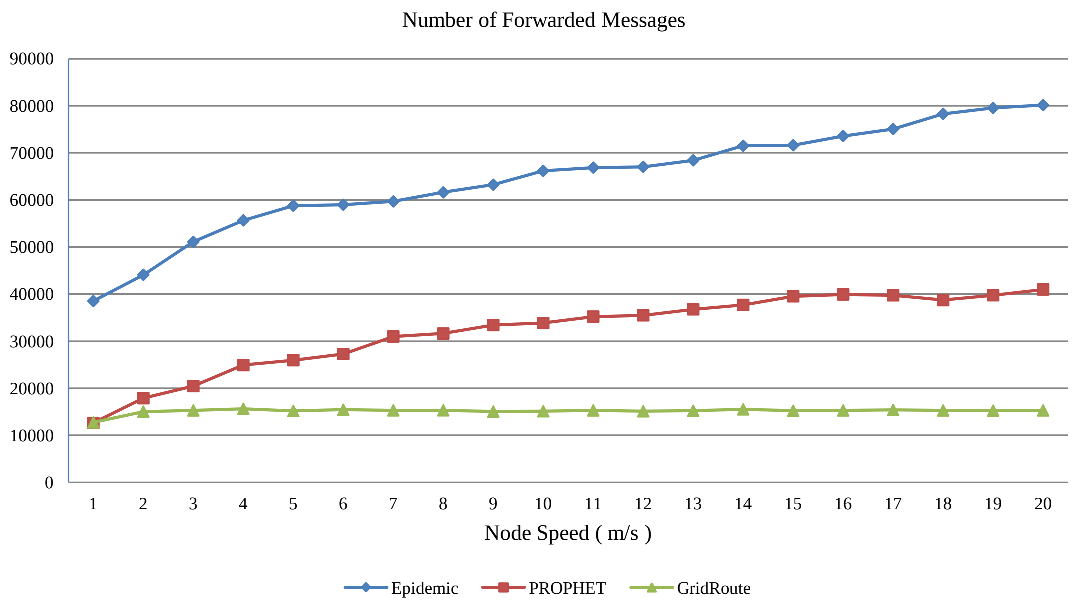

Figure 4.41: Number of forwarded messages, CBMM, node count: 50, buffer size: 50 messages, transmission range: 100 m, hop limit: 11.

easily as it can traverse the network area faster.

Figure 4.42 shows that, increase in mobility decreases the average message delay considerably. In Epidemic Routing and PROPHET other than the nodes are infected faster, infected nodes can contact with the destination more quickly. As a result, the average message delay inversely proportional to the message delay.

In GridRoute when a node receives a message to forward, it waits to encounter of a more promising node or the receiver. Increasing the mobility decreases this waiting time and when the nodes are very mobile, GridRoute can deliver messages to their destination with the same latency as GridRoute and PROPHET have.

To sum up, this section presents the performance results of GridRoute by comparing it with GridRoute and Epidemic Routing. Simulation results indicate that, GridRoute is able achieve higher delivery ratios even it is a single copy protocol. It can effectively capture the locality of human mobility and provide intelligent forwarding steps to deliver the messages to their destination. It can

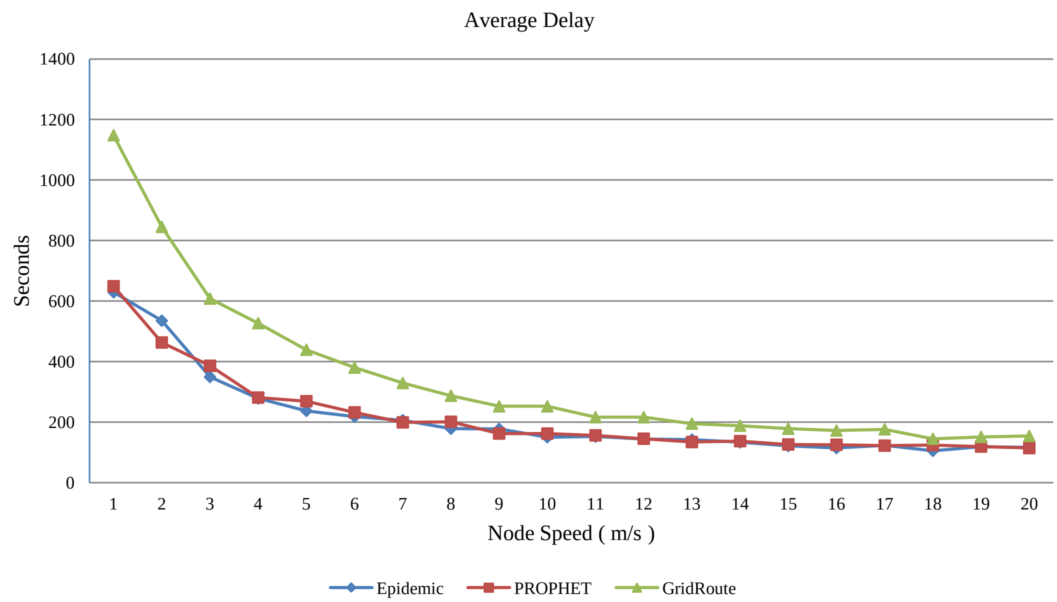

Figure 4.42: Average delay, CBMM, node count: 50, buffer size: 50 messages, transmission range: 100 m, hop limit: 11.

significantly lower the message overhead and increase the lifetime of the network. Also it has reasonable end to end delay.

Simulation results show that, GridRoute take advantage of increase in beneficial network parameters like mobility or transmission range while it is resistant to changes that effects other protocols negatively (like increase in node density in terms of redundant message traffic).

Although CBMM can be considered as a realistic mobility scenario, in real life efficiency of GridRoute may be more evident as the decrease in randomness will provide more information to it to use in forwarding decisions.

Chapter 5

CONCLUSION AND FUTURE WORK

This work addressed the routing problem in Delay Tolerant Mobile Networks. A probabilistic routing protocol based on predicted node contacts that operates on multi-layered grid called GridRoute is presented. GridRoute stores location probabilities of nodes and forwards messages to receiver's favorite location in general. Furthermore, GridRoute can make intelligent routing decisions with no storage of any information on network agents and it is able to deliver messages without running of multiple copies. Furthermore, GridRoute can decrease hardware (GPS chip) dependency which is the remedy of location based techniques by adapting existing localization techniques. Lastly, GridRoute can provide efficient message secrecy together with authenticity and security using PKI and Identity Based Encryption.

Simulation results indicated that GridRoute outperformed existing routing protocols in terms of memory requirement. It achieved high delivery ratio, reasonable end-to-end delay and significantly lower message overhead. This work also analyzed the performance effect of network parameters like node density, message generation rate, node mobility and transmission range.

5.1 Possible Improvements

Broadcast in DTMN using GridRoute is another possible problem that should be addressed. One possible solution can be a divide and conquer type of broadcast technique using GridRoute. In this case, a node that aims to broadcast a message can divide the network into two parts based on its favorite grid-cell. For example, if its favorite grid-cell is in the left half of network area, it may try to find a node that spends lots of time in the other half of the network and delivers the message to that node. Once the message is delivered, message holder and receiver further divide their own halfs of the network and repeat the procedure. Once the messages are replicated to a node for each grid-cell, message holders can spread the message to nodes that have the same grid-cell as their favorite position with the message holder. Further precautions must be addressed in this type of broadcast technique. For example, a particular grid-cell may not have any node that it is the favorite grid-cell of them or message holder may not contact with that kind of cell for a long time. In this case a message holder that operates in that area may hold a timer and if it can not find a candidate node for that grid-cell in a given time, it can take the responsibility to spread the message to possible nodes that spends lots of time in that cell.

Furthermore, like broadcasting, multicast message forwarding or relay node selection based on willingness can be adopted for GridRoute. For example, a node may be willing to deliver a message to a friend but it may not want to burden itself with the message of a foreigner. These are some possible extensions but we believe GridRoute can also be extended or improved by capturing the location information and human mobility more precisely. Also, test results on real DTMN networks can provide a better and more realistic benchmark for compared protocols.

5.2 Summary

In this thesis a brief introduction to DTMNs, some important related works are mentioned in Section II. In Section III, GridRoute and 5 variations of it which have their own advantages according the needs of deployed network are presented. Information about the simulations, simulation parameters and mobility scenarios are presented in Section IV and results of the simulations are presented in Section V with detailed interpretations of the results.

Bibliography

[1] I. Akyildiz, O. Akan, C. Chen, J. Fang, and W. Su. InterPlaNetary Internet: state-of-the-art and research challenges. *Computer Networks*, 43(2):75–112, Oct. 2003.

[2] M. Balazinska and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *Proceedings of the 1st international conference on Mobile systems, applications and services - MobiSys '03*, pages 303–316, New York, New York, USA, 2003. ACM Press.

[3] G. W. Boehlert, D. P. Costa, D. E. Crocker, P. Green, T. O'Brien, S. Levitus, and B. J. Le Boeuf. Autonomous Pinniped Environmental Samplers: Using Instrumented Animals as Oceanographic Data Collectors. *Journal of Atmospheric and Oceanic Technology*, 18(11):1882–1893, Nov. 2001.

[4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, Oct. 2002.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. *University of Cambridge, Computer Lab, Tech. Rep. UCAM-CL-TR-617, Feb*, (617), 2005.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, volume 7. MIT Press, 2001.

[7] H. Dang and H. Wu. Clustering and cluster-based routing protocol for delay-tolerant mobile networks. *IEEE Transactions on Wireless Communications*, 9(6):1874–1881, June 2010.

[8] A. Doria, M. Uden, and D. Pandey. Providing connectivity to the saami nomadic community. In *Proceedings of the 2nd International Conderance on Open Collaborative Design for Sustainable Innovation (dyd 02)*, page 3. Citeseer, 2002.

[9] K. Fall. A delay-tolerant network architecture for challenged internets. *Proceedings of the 2003 conference on Applications,*, page 27, 2003.

[10] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 187–201. ACM, 2004.

[11] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking - MobiCom '04*, page 45, New York, New York, USA, 2004. ACM Press.

[12] S. Jain and K. Fall. Routing in a delay tolerant network. *ACM SIGCOMM Computer Communication*, 34(4):145, Oct. 2004.

[13] A. Joseph, J. Tauber, and M. Kaashoek. Mobile computing with the Rover toolkit. *IEEE Transactions on Computers*, 46(3):337–352, Mar. 1997.

[14] A. Kate, G. M. Zaverucha, and U. Hengartner. Anonymity and security in delay tolerant networks. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, pages 504–513. IEEE, 2007.

[15] M. Kim, D. Kotz, and S. Kim. Extracting a Mobility Model from Real User Traces. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–13. IEEE, 2006.

[16] S. Lee, S. Wong, K. Lee, and S. Lu. Content management in a mobile ad hoc network: beyond opportunistic strategy. In *Proceedings of INFOCOM, 2011*, pages 266–270. IEEE, 2011.

[17] J. Leguay, T. Friedman, and V. Conan. DTN routing in a mobility pattern space. *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking - WDTN '05*, pages 276–283, 2005.

[18] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, July 2003.

[19] C. Liu. Scalable routing in delay tolerant networks. *international symposium on Mobile ad hoc networking*, 2007.

[20] M. McNett and G. M. Voelker. Access and mobility of wireless PDA users. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(2):40, Apr. 2005.

[21] C. Mergenci. Routing in delay-tolerant networks with periodic connections. Master's thesis, Institute of Engineering and Science of Bilkent University, August 2010.

[22] S. Merugu, M. Ammar, and E. Zegura. Routing in Space and Time in Networks with Predictable Mobility. *Time*, (iii):1–13, 2004.

[23] V. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for Internet hosts. *ACM SIGCOMM Computer Communication Review*, 31(4):173–185, 2001.

[24] C. Perkins. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer*, pages 234–244, 1994.

[25] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*, pages 90–100. IEEE, June 1999.

[26] C. Saad, A. Benslimane, and J. Konig. A distributed method to localization for mobile sensor networks. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 3058–3063. IEEE, 2007.

[27] a. Seth and S. Keshav. Practical security for disconnected nodes. In *1st IEEE ICNP Workshop on Secure Network Protocols, 2005. (NPSec).*, pages 31–36. IEEE, 2005.

[28] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in cryptology*, pages 47–53. Springer, 1985.

[29] T. Small and Z. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 233–244. ACM, 2003.

[30] T. Spyropoulos, K. Psounis, and C. Raghavendra. Performance analysis of mobility-assisted routing. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 49–60, New York, New York, USA, 2006. ACM.

[31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks 2004 IEEE SECON 2004*, volume 00, pages 235–244. IEEE, 2004.

[32] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and Wait : An efficient routing scheme for intermettently connected mobile networks. *Direct*, pages 252–259, 2005.

[33] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Proc AAAI Workshop on AI for Web Search AAAI 2000 Austin*, pages 58–64. AAAI, 2000.

[34] Y. Tian and J. Li. Location-aware routing for delay tolerant networks. In *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on*, pages 1 –5, Aug. 2010.

[35] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Duke University, 2000.

[36] W. Vogels, R. V. Renesse, and K. Birman. The power of epidemics: robust communication for large-scale distributed systems. *Computer Communication*, 2003.

[37] F. Yasmeen, S. Urushidani, and S. Yamada. A probabilistic position-based routing scheme for delay-tolerant networks. In *Computers and Information Technology, 2009. ICCIT '09. 12th International Conference on*, pages 88 –93, Dec. 2009.

[38] X. Zhang and K. G. Shin. E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks. In *Power*, MobiCom '11, pages 205–216. ACM Press, 2011.