M.Sc. Engg. Thesis

Probabilistic Quota Based Adaptive Routing in Opportunistic Networks

by

Md. Anindya Tahsin Prodhan

Submitted to

Department of Computer Science and Engineering
in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka 1000

December 2010

The thesis titled “**Probabilistic Quota Based Adaptive Routing in Opportunistic Networks**,” submitted by Md. Anindya Tahsin Prodhan, Roll No. 040805021P, Session April 2008, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on December 18, 2010.

Board of Examiners

1. ______________________________	
Dr. Md. Humayun Kabir	Chairman
Associate Professor	(Supervisor)
Department of Computer Science and Engineering	
BUET, Dhaka 1000	
2. ______________________________	
Dr. Md. Monirul Islam	Member
Professor & Head	(Ex-officio)
Department of Computer Science and Engineering	
BUET, Dhaka 1000	
3. ______________________________	
Dr. Mahmuda Naznin	Member
Associate Professor	
Department of Computer Science and Engineering	
BUET, Dhaka 1000	
4. ______________________________	
Dr. Khandoker Nadim Parvez	Member
Assistant Professor	(External)
Department of Electrical Engineering & Computer Science	
North South University, Dhaka	

Candidate's Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Md. Anindya Tahsin Prodhan
Candidate

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

All praises due to Allah, the most benevolent and merciful.

I express my heart-felt gratitude to my supervisor, Dr. Md. Humayun Kabir for his constant supervision of this work. He helped me a lot in every aspect of this work and guided me with proper directions whenever I sought one. His patient hearing of my ideas, critical analysis of my observations and detecting flaws (and amending thereby) in my thinking and writing have made this thesis a success.

I would also want to thank the members of my thesis committee for their valuable suggestions. I thank Professor Dr. Md. Monirul Islam, Dr. Mahmuda Naznin, and specially the external member Dr. Khandoker Nadim Parvez.

I also thank one of my friend Rajkumar Das, who was my partner in the early works of my research. He was always there for me when I needed.

In this regard, I remain ever grateful to my beloved parents, who always exists as sources of inspiration behind every success of mine I have ever made.

Abstract

Routing in an Opportunistic Network (OpNet) is challenging due to the absence of a complete end-to-end path from a source to a destination. Consequently, OpNet routing protocols use store and forward routing with multiple replicas of a message in the network to achieve higher delivery ratio. Although flooding the network with many copies of a message improves the chances of message delivery, it causes higher network contention. Quota based routing protocols deal with this problem by placing an upper bound on the number of replicas per message in the network. However, this saving in terms of valuable network resources comes at the cost of delivery performance.

In this thesis, we propose two novel quota based routing protocols, TBR and ProbRoute, which use the network resources efficiently and achieves delivery ratios as high as that of the flooding based protocols. TBR prioritizes both the schedule of messages to be forwarded to the neighbor and the schedule of messages to be dropped from the buffer. These priorities are based on massage time to live (TTL), message hop count, message replication factor and message size. The TTL based message priority enhances the chance of message delivery by preferring to the messages with the earliest deadline. ProbRoute introduces a weighted probability metric, Contact Probability, to guide the messages effectively in the network, and a Priority Queue, to rank the messages based on an adaptive message priority. Our simulation results show that both TBR and ProbRoute outperform all the existing quota based routing protocols in terms of delivery ratio and overhead ratio. Both protocols not only matches with the delivery ratios of flooding based routing protocols but also achieves better delivery ratio than that of those routing protocols while incurring significantly less overhead and less latency. ProbRoute achieves slightly better delivery ratio compared to that of TBR at the cost of slightly more network overhead.

Chapter 1

Introduction

One of the major problem in routing messages in an Opportunistic Network (OpNet) is the absence of a complete end-to-end path from the source to the destination. Due to the inherent adversity of OpNets the conventional routing algorithms does not fit into opportunistic routing. With the emergence of many real life OpNets, routing in OpNet have drawn a lot of attention from the researchers of network community.

1.1 Motivation

OpNet is a type of challenged networks, where network contacts (i.e., communication opportunities) are intermittent, an end-to-end path between the source and the destination may have never existed, disconnection and reconnection is common, and/or link performance is highly variable or extreme. Therefore, traditional Internet and Mobile Ad-hoc Network (MANET) routing techniques can not be directly applied on networks in this category. With numerous emerging opportunistic networking applications, such as wireless sensor networks (WSN), underwater sensor networks (UWSN), transportation networks, pocket switched networks (PSN), people networks, and etc., it remains desirable/necessary to develop effective schemes that can better accommodate the characteristics of OpNets. This is why routing in OpNets stimulated keen interest in the researchers.

Because of the inconsistent connectivity of OpNets, opportunistic routing protocols requires the insertion of multiple copies or replicas of a message in the network to increase the chance of reaching the destination. Many of the earlier researchers used flooding based methods which employs flooding of messages to improve message delivery. But these approaches suffer from high network congestion.

On the other hand, Quota based routing protocols utilize the network resources efficiently by exploiting restricted flooding. Although quota-based protocols are much better stewards of network resources than their flooding-based counterparts, major shortcoming of quota based approaches is their inability to achieve very high delivery ratio. Therefore, designing a quota based routing protocol that can achieve the delivery ratio comparable to the flooding based alternatives would be a very notable contribution. Which lead us to design these new quota based routing protocols.

1.2 Related Works

Delay or disruption tolerant networks (DTNs) [1][14] provide reliable communications in an intermittently connected environment. Routing in such networks [36] is challenging due to the lack of knowledge about the network dynamics and the absence of stable end-to-end path. Factors such as high node mobility, low node density, intermittent power from energy management schemes, environmental interference and obstruction, short radio range and malicious attacks [13] etc. can result in these unstable paths. Examples of DTNs include Interplanetary Internet [6] or Deep Space Networks, Sensor-based Networks using scheduled intermittent connectivity, Military Networks [2], Inhabitant or Wildlife Tracking System [19], Terrestrial Wireless Networks that cannot ordinarily maintain end-to-end connectivity, Satellite Networks with moderate delays and periodic connectivity, and finally Underwater Acoustic Networks with moderate delays and frequent interruptions due to different environmental factors. Routing protocols can be divided into two main groups- *Forwarding based* and *Replication based.* Forwarding based routing protocols keep one copy of each message in the network and attempt to direct the message to the destination. Traditional wireless routing protocols such as AODV [25], DSR [18] etc. are forwarding based routing protocols. OpNets are a special category of DTN where neither the meeting schedule nor the contact period is known in advance. Because of the irregularity in connectivity, very few forwarding based routing protocols [7][15][30] have been proposed for OpNet. Most of the forwarding based protocols use the assumptions of network connectivity and environmental knowledge to take routing decisions. But the performance of these protocols falls drastically when the environment is completely opportunistic.

Due to the intermittent and uncertain connectivity in OpNet, most of the OpNet routing protocols are replication based. Replication based routing protocols follow “store and forward” strategy. These routing protocols take advantage of temporal connectivity to make a sequence of independent and local forwarding decisions based on current connectivity information and future connectivity predictions.

Replication based protocols insert multiple copies of a message into the network to ensure higher message delivery. The earliest replication based routing protocol is epidemic routing [33], which tries to send a copy of a message to each encountered node. This genre of Replication based protocols are called *Flooding based* protocols. Other examples of such flooding based protocols are- PROPHET [23], MaxProp [5], RAPID [4], PREP [26] etc. Flooding based protocols generally try to replicate as many copies of a message as the resources permit. All the flooding based approaches inject many messages in the network in order to achieve higher delivery ratio. This approach is vulnerable to high network contention and could lead to huge overhead and latency.

A Replication based routing protocol is said to be Quota based if the number of copies of each message is kept independent of the network size. Quota based protocols save the network resources by maintaining a controlled number of copies of a message in the network. Spray and Wait [31], ORWAR [27], Spray and Focus [32], and EBR [24] are the examples of some popular Quota based routing protocols. Existing quota based routing protocols suffer from comparatively lower delivery ratios even though they are the better steward of network resources.

1.3 Main Contribution

In this thesis, we propose two new quota based routing protocol for OpNets: TTL Based Routing (TBR) and Probabilistic Quota Based Adaptive Routing (ProbRoute). Both the protocols achieve superior performance than the existing quota based and flooding based routing protocols in the opportunistic environments. TBR introduces a new buffer management strategy to rank the messages in the buffer to schedule the next message to forward or delete. TBR ranks the messages based on message expiry time or TTL, message hop count, message replication count and message size. The use of TTL in ranking the messages allows the message with the earliest deadline to get the preference, while the use of message size allows the shortest message to get the preference. Hop count and replication count are used to ensure network fairness. ProbRoute introduces a probabilistic metric, *Contact Probability*, to disseminate messages in the appropriate direction. Along with that, ProbRoute uses a *Priority Queue* to choose the next message to forward or delete. The priority of a messages in the queue is computed based on the number of the message replicas in the node and the size of the message. Since the number of replicas of a message in a node varies after each successful transmission of that message, the message priority becomes an adaptive parameter, which leads ProbRoute to achieve better delivery ratio than that of the other protocols.

To evaluate the effectiveness of our protocols, we perform simulations under both ‘Map based movement vehicular model’ and ‘Random way point movement model’ [20] and compare the delivery ratios, overhead and latency of our protocols with that of all the other popular OpNet routing protocols. Simulation results show that both the protocols achieve higher delivery ratios than that of the existing quota based counterparts while keeping the resource usage minimum. Our protocols also achieves better delivery ratio compared to that of the best flooding based protocol maintaining a low overhead and low latency.

1.4 Overview of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we introduce Delay Tolerant Network (DTN) and OpNet briefly. In Chapter 3, we present a summary of a few popular routing protocols for OpNets. We describe our first protocol TBR in Chapter 4 and second protocol ProbRoute in Chapter 5. Chapter 6 presents experimental results and their analysis. Finally, Section 7 concludes this thesis with some future research directions.

Chapter 2

DTN and OpNet Basics

Legacy network protocols , such as TCP/IP, perform poorly in many challenged networks where the contacts among the nodes are not persistent rather intermittent or opportunistic, end-to-end delays are relatively high and variable, and the links have non-negligible bit error rate. A new network architecture, called Delay Tolerant Network (DTN), has been defined to meet the challenges of these special type of networks. OpNets are a type of DTN where network communication opportunities appear opportunistic, an end-to-end path between source and destination may have never existed, and disconnection and re-connection is common in the network. In the following sections we introduce the DTN and OpNet briefly.

2.1 Delay Tolerant Network

A DTN is an overlay network on top of some regional networks. The regional networks, which are subject to disruption and disconnection and high-delay, are connected by DTN [35]. The networks with such characteristics are Interplanetary Internet or deep space networks, sensor-based networks using scheduled intermittent connectivity, terrestrial wireless networks that cannot ordinarily maintain end-to-end connectivity, satellite networks with moderate delays and periodic connectivity, and underwater acoustic networks with moderate delays and frequent interruptions due to environmental factors.

Though the Internet interconnects multiple networks with diverse communication characteristics, it has some basic assumptions that are not complied by the DTN regional networks. Those assumptions are as follows:

1. Continuous, bidirectional, end-to-end path are available between source and destination.
2. Relatively short and consistent round-trip delay between the source and destination.
3. Apparently symmetric data rate in both directions between the source and destination.
4. Comparatively low error-rate in data transmission links between the source and the destination.

DTN [1] introduces a new protocol layer called "Bundle Layer to deal with the above non-complying special characteristics of these challenged regional networks. The Bundle Layer [28] is designed to operate above the transport layer of OSI model to send the bundles from the source to the destination. Here, a bundle is a whole message from an application with the bundle protocol header. To overcome the issues associated with intermittent connectivity, long or variable delay, asymmetric data rates, and high error rates bundle layer uses store-and-forward message switching to transfer a bundle from one node to another node.

As opposed to establishing a TCP like connection, a Bundle sender sends application messages encapsulated into bundles in a similar fashion that a UDP sender encapsulates user data into UDP datagram. In order to provide communication reliability, bundle protocol uses a concept called custody transfer. Custody transfer implements node-to-node or hop-by-hop retransmission of lost or corrupt data.

2.1.1 DTN Characteristics

DTNs are specially characterized by their intermittent connectivity, unpredictable and long delay, asymmetric data rate, and high error rate [12]:

Intermittent Connectivity: Network partitioning due to, channel fading, line of sight loss, mobility, power conservation strategy, and security measures can cause intermittent connectivity between the source and the destination.

Long or Variable Delays: TCP assumes 500ms maximum round-trip delay between the source and the destination while round-trip time between Earth and Jupiter's moon Europa, for example, run between 66 and 100 minutes. This round-trip delay in Interplanetary Internet is excessively long to defeat TCP protocol. Intermittent connectivity and variable queuing delays in the nodes contribute to make the round-trip delay very much unpredictable.

Asymmetric Data Rates: Concurrent communication in both directions with an approximately symmetric transmission rate between the source and the destination of the traditional networks does not happen in many challenged networks. For example, the communication in NASA's Deep Space Network (DSN) is only one direction. Even if bi-directional communication is possible the data rate in two directions may vary. For example, uplink and downlink data rate of a satellite communication differs. Asymmetric communication makes the conversational protocols (TCP or SCTP), which are based on end-to-end signaling, perform poor.

High Error Rates: Traditional networks are linked by copper wires or optical fiber cables, Bit error rates of copper (10^{-9}) and optical fiber (10^{-12}) are tolerable. On the other hand, most of the links in DTNs are wireless with a high bit error rate (10^{-2} to 10^{-4}). High bit error rate defeats the algorithms of traditional transport protocols, such as TCP or SCTP.

2.1.2 DTN Contacts

A DTN network can be considered as a multi-graph, where vertices (nodes) are interconnected with more than one edge. Edges in this graph are, in general, time-varying with respect to their delay and capacity and directional because of the possibility of one-way connectivity. When an edge between two nodes has zero capacity, it is considered that they are not in contact. Edges may vary between positive and zero capacity. There might be a period of time interval during which the capacity is strictly positive and the delay and capacity are almost constant. This period of time is called a contact. The product of the capacity and the interval is known as a “contact’s volume”. A DTN node can send, receive, or forward a message when it is in contact. It stores the messages in the persistent storage until it becomes in contact again. If contacts and their volumes are known ahead of time, intelligent routing and forwarding decisions can be made optimally. Several types of contacts are possible in DTNs, such as persistent, on-demand, intermittent-scheduled, intermittent-opportunistic, and intermittent-predicted.

Persistent Contacts: No connection-initiation action is required to instantiate a persistent contact. Persistent contacts are always on. Internet connection through a DSL or Cable Modem gives a persistent contact.

On-Demand Contacts: Connection-initiation actions are required for on-demand contacts. It remains persistent until terminated. A dial-up connection is an example of an on-demand contact.

Intermittent - Scheduled Contacts: A scheduled contact is an agreement to establish a contact at a particular time for a particular duration. An example of a scheduled contact is a link with a low-earth orbiting satellite.

Intermittent - Opportunistic Contacts: Opportunistic contacts are unscheduled; they rather present themselves unexpectedly. For example, an unscheduled aircraft flying overhead and beckoning, advertising its availability for communication, would present an opportunistic contact.

Intermittent - Predicted Contacts: Predicted contacts are neither scheduled nor unexpected, rather predictable. The prediction of likely contact times and durations are based on a history of previously observed contacts or some other information.

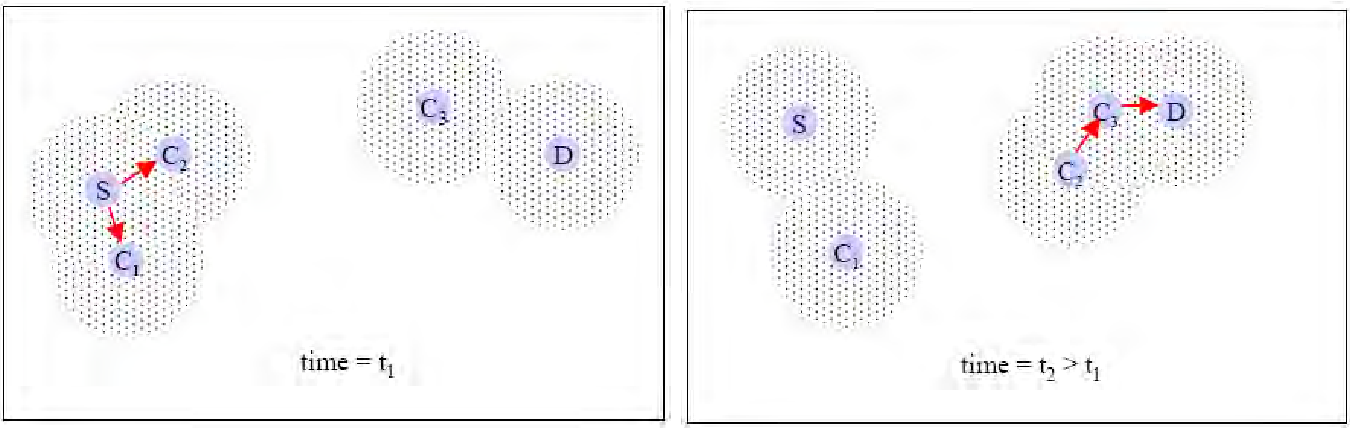
Due to the intermittent properties of DTN most of the routing protocols use store and forward routing strategy.

2.1.3 Store and Forward Message Switching

A DTN node must store a whole message until either another node accepts the custody or the expiration of the messages time-to-live. Persistent storage, such as hard disk, is used to store the messages. All types of DTN nodes need persistent storage to store the messages until outbound links are available. Once a outbound link is available, a DTN node moves the whole message in a single transfer though the fragmentation is possible. This is called store-and-forward message switching. Bundle layer usage of store-and-forward message switching to transfer a bundle from one node to another node helps it to overcome the issues associated with intermittent connectivity, long or variable delay, asymmetric data rates, and high error rates in DTN. Store-and-forward message switching together with custody transfer provides with the communication reliability.

2.2 Opportunistic Network

OpNet is a type of challenged networks, where network contacts (i.e., communication opportunities) are intermittent, an end-to-end path between the source and the destination may have never existed, disconnection and reconnection is common, and/or link performance is highly variable or extreme. As a result routes in OpNets are made by chance. In today's world numerous opportunistic networking applications, such as Wildlife Tracking System [19], Search and Rescue System [16], Underwater Sensor Network [11], Vehicular Ad-hoc Network (VANET) [5][22], Pocket Switched Network [8][9][17], people networks [29][34], etc., are emerging. OpNets apart from being a type of DTN also possess



(a) (b)

Figure 2.1: A source, S, wishes to transmit a message to a destination but no connected path is available in part (a). Carriers, C1-C3 are leveraged to transitively deliver the message to its destination at some later point in time as shown in (b)

a set of defining characteristics-

- Knowledge about the network topology is not needed. The messages are delivered in a hop by hop fashion toward the destination.
- Additional delay in message delivery occurs when no forwarding opportunities exists. Nodes store the messages and wait for future opportunities.
- Routing / Forwarding issues are important. Most of the protocols in OpNets uses flooding to forward the messages towards the destination.

Figure 2.1 depicts an opportunistic routing at a high level, with mobile nodes represented as dark circles and their wireless communication range shown as a dotted circle extending from the node. In Figure 2.1(a), a source, S, wishes to send a message to a destination, D, but no connected path is available from S to D. S transmits its messages to its two neighbors, C1 and C2, within direct communication range. At some later time, as shown in Figure 2.1(b), C2 comes into direct communication range with another host, C3, and transmits the message to it. C3 is in direct range of D and finally sends the message to its destination. Thus opportunistic routing protocols successfully transmits messages even if no path from the source and destination exist.

Chapter 3

State of the Art Routing Protocols in OpNets

Over the years there have been a huge body of works [10][18][25] on routing protocols for multi-hop wireless networks. These protocols can automatically route messages even when nodes are mobile and the link quality varies. However, these protocols always try to find an end-to-end path, and do not support communication between nodes in different network partitions. Thus, traditional Ad-hoc routing protocols do not fit in the opportunistic environment as there is no guarantee of an end-to-end path in OpNets. As a result, the performance of these protocols falls drastically even if the network is slightly disconnected. Due to this inherent adversity in OpNets most of the OpNet routing protocols replicate multiple copies of each message in the network to increase the chance of message delivery. These replication based protocols are mainly stratified into two groups-

- Flooding Based Routing Protocol
- Quota Based Routing Protocol

In the following sections we introduce a few widely known replication based routing protocols.

3.1 Flooding Based Routing Protocol

Most of the earlier replication based protocols tried to flood the messages to as many nodes in the network as the resources permit. Here we present some examples of such protocols-

3.1.1 Epidemic Routing

In epidemic routing protocol [33] the encountered nodes first exchange a summary vector with each other. The summary vector contains the summary of the messages stored in the node's buffer. From the summary vector a node learns about the new messages in the neighbor and sends a data request to the neighbor for those messages. Neighbor replies to the data request by sending the requested messages. Epidemic routing algorithm forwards the messages to the neighbor blindly regardless of their destinations, which causes serious performance penalty.

3.1.2 PROPHET

PROPHET [23] uses a delivery predictability parameter to predict the chance of reaching a destination from a given node. When two node meet each other they exchange their delivery predictability and update their delivery predictability using the mechanism described later in this section. A node using PROPHET protocol forwards a message to its neighbor only if the neighbor has the higher chance of reaching the destination (higher delivery predictability) than the node itself.

Delivery Predicability Calculation

The calculation of the delivery predictabilities have three parts. The first thing to do is to update the metric whenever a node is encountered, so that nodes that are often encountered have a high delivery predictability. This calculation is shown in Equation

3.1.

$$P_{(a,b)} = p_{(a,b)_{old}} + (1 - p_{(a,b)_{old}}) \times P_{init} \tag{3.1}$$

Where $P_{init} \in [0,1]$ is an initialization constant.

If a pair of nodes does not encounter each other in a while, they are less likely to be good forwarders of messages to each other, thus the delivery predictability values must age, being reduced in the process. The aging equation is shown in Equation 3.2.

$$P_{(a,b)} = p_{(a,b)_{old}} \times \gamma^k \tag{3.2}$$

Where, $\gamma \in [0,1]$ is the aging constant and k is the number of time unit elapsed.

The delivery predictability also has a transitive property, that is based on the observation that if node A frequently encounters node B, and node B frequently encounters node C, then node C probably is a good node to forward messages destined for node A too. Equation 3.3 shows how this transitivity affects the delivery predictability.

$$P_{(a,c)} = p_{(a,c)_{old}} + (1 - p_{(a,c)_{old}}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \tag{3.3}$$

Where $\beta \in [0,1]$ is the scaling constant.

3.1.3 MaxProp

MaxProp [5] apparently has the best delivery ratio among the flooding based protocols designed for OpNet. Each node, i, keeps track of a probability of meeting peer j. This probability metric is referred to as delivery likelihood f_j^i. At each contact, the nodes exchange their probabilities after updating the likelihood using incremental averaging method described below. The nodes calculate the cost of reaching a destination through all possible path using the equation, $c(i, i+1, \ldots, d) = \sum_{x=i}^{d-1} [1 - (f_{x+1}^x)]$. The cost of any destination is the lowest path cost among all possible paths to that destination. Thus each node in MaxProp maintains a ranked list of destinations based on their costs and the messages on the nodes are stored by the rank of their destinations. Finally, the nodes flood the messages to their neighbors starting from the top of the list.

Calculation of Delivery Likelihood

Let the set of nodes in the network be s. For all nodes, f_j^i is initially set to $\frac{1}{|s|-1}$. When node i encounters node j, the value of f_j^i is incremented by 1, and then all values of f are re-normalized. Thus the nodes that are seen infrequently obtain lower values over time. This method is called incremental averaging.

3.2 Quota Based Routing Protocols

Flooding based protocols, such as Epidemic [33], PROPHET [23], MaxProp [5], RAPID [4], and PREP [26], trade resource consumption to achieve higher delivery. Their high demand of network resources such as- bandwidth and storage may in many cases effect congestion in the network and increase latency. To mitigate this flooding effect another variety of Replication based protocols named Quota based protocols were proposed. Quota based routing protocols limit the number of replicas of any messages by setting an upper bound. Some examples of such protocols are-

3.2.1 Spray and Wait

Spyropoulos et. al. proposed the first quota based protocol Spray and Wait [31], where the allowable message copies were fixed at the message creation time. Spray and Wait uses two steps to route messages.

Spray Phase: In this phase the nodes spays the message replicas into the network until the replication factor becomes 1.

Wait Phase: In this phase the nodes holding the messages waits for a direct encounter with the destination.

Generally all the quota based routing protocols are based on the same principle as Spray and Wait with some modifications.

3.2.2 ORWAR

One of the best performing quota best protocol is ORWAR [27]. ORWAR assumes each message has a priority: low, medium or high. ORWAR assigns utility of a message based on its priority; 1 for low, 2 for medium and 3 for high. Finally it computes utility per bit by dividing the message utility by message size. In ORWAR, messages are sorted using utility per bit ordering and at each contact, the node sends the message with the best utility per bit that fits into the maximum transmittable message size (s_{max}). Using s_{max}, ORWAR controls the number of retransmissions in the network. s_{max} is computed using the data transfer rate and contact time window. ORWAR also uses the replication factor as a function of utility to allow more copies for high priority messages and less copies for low priority messages. Table 3.1 summarizes the initialization of the replication factors. Here Δ is an algorithm parameter.

Table 3.1: Initial values of the replication factor as a function of utility

Priority Class	**Utility**	L_k = **Replication Factor**
High	3	$L + \Delta$
Medium	2	L
Low	1	$L - \Delta$

Calculation of Maximum Transmittable Message Size

s_{max} is computed based on the current connectivity context from the contact time window (t_{cw}) and the data rate (b) using the Equation 3.4

$$s_{max} = b \times t_{cw} \tag{3.4}$$

Data rate (b) is given by the device radio properties (i.e., for Bluetooth 2.0 the data rates are about 250kBps). Contact time window (t_{cw}) is calculated from the nodes respective speeds ($\overrightarrow{v_1}, \overrightarrow{v_2}$) and transmission range (r_1, r_2) as shown in the Figure 3.1 in which dashed trajectories denote the movement of one node.

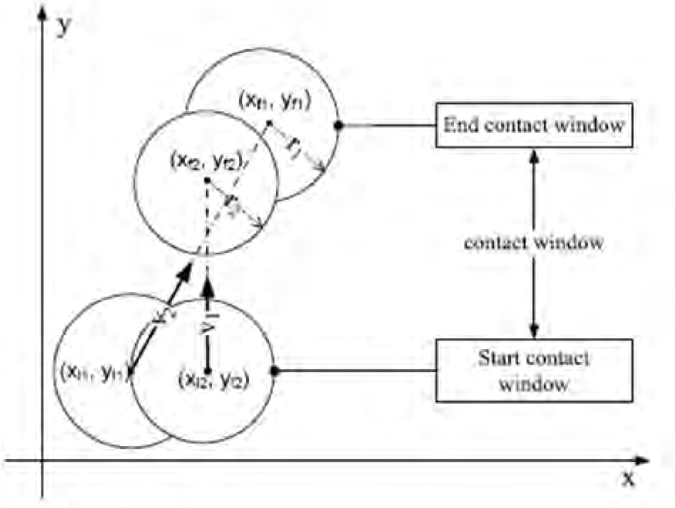


Figure 3.1: Calculation of Maximum Transmittable Message Size

3.2.3 EBR

Nelson et. al. proposed another quota based protocol EBR [24], which sprays the messages based on the node encounter history. They use the assumption that the nodes which experience a large number of encounters are more likely to be successful to pass the message to the final destination. This is why every node running EBR maintains a past rate of encounter average (EV), which is used to predict future encounter rates. When two nodes meet in the spray phase, the relative ratio of their respective rates of encounter determines the appropriate fraction of message replicas the nodes should exchange. For example, if node i contacts node j, i will calculate the number of replicas of any message to be sent to j using the equation 3.5. The wait phase of EBR is similar to that of spray and wait protocol.

$$L_{k_j} = L_{k_i} \times \frac{EV_i}{EV_i + EV_j} \tag{3.5}$$

Where,

L_{k_i} is the total number of message replicas stored in i;

L_{k_j} is the number of replicas to be sent to node j;

EV_i encounter average for node i;

EV_j encounter average for node j.

All the above Quota based routing protocols achieve better network resource utilization, however, suffer from low delivery ratio. In the next two chapters, we propose two improved quota based routing protocols in order to achieve better or comparable delivery rates with the flooding based protocols with extremely low overhead and low resource uses.

Chapter 4

The TBR Protocol

The first protocol that we proposed in our thesis is TTL Based Routing (TBR). In TBR, we used two different lists to rank the messages in the node buffer to assist message forwarding and message deleting decisions. While forwarding, TBR rank the messages based on their TTL, so that the message with the earliest deadline is preferred. On the other hand, in case of buffer overflow, we delete the message with the minimum replication count, so that the message with the highest replicas present in the network is preferred while deleting. To explain our TBR protocol, we first present the Message structure that will be used by the nodes running our TBR protocol, followed by a model of an OpNet. Then we provide a brief narration of our protocol. Finally, we present the buffer management scheme used in TBR. We conclude the chapter with a detailed algorithmic depiction of our protocol.

4.1 Message Structure

In our protocol, each message m_k contains a message header along with the message data. The message header specifies different particulars of the message like- replication factor (L_k) which denotes the intended number of message copies, size of the message (s_k), hop count for the message (H_k), list of nodes that the message have visited (LVN_k), time

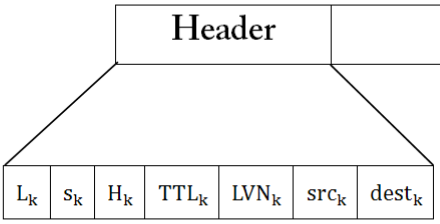


Figure 4.1: Message Format

to live (TTL_k) of the message and the source (src_k) and the destination ($dest_k$) of the message. Figure 4.1 shows the skeleton of a message used in TBR and ProbRoute. In our message structure we introduced a new field LVN_k in the message header. This LVN_k allows our protocol to avoid any kind of routing loop. In our protocol whenever a node gets the opportunity to forward a message (m_k) to its neighbor it checks wether the neighbor is in the LVN_k and the message will only be forwarded if the neighbor is not in LVN_k. As the number of hops are usually very small, adding LVN_k obviously will impose only a very insignificant overhead. Thus using LVN_k our protocol achieves better routing without deteriorating the network performance. The Table 4.1 presents the fields of the message header along with a brief discussion of the fields.

4.2 System Model

Here, we model OpNet as a set of mobile nodes, where neither the contact schedule nor the message arrival rate is known in advance. Two nodes can transfer data packets to each other only when they are within each others direct communication range. Node meetings are assumed to be short-lived. We also assume a node has limited buffer space for storing the messages. A node can deliver a message directly to the destination node or through a set of intermediate nodes in the network. We assume each message being transmitted as a whole in the network, i.e., we do not consider fragmentation of messages.

Table 4.1: Fields in Message Header and their Short Description

Name of the Field	**Short Description**
Replication Factor, L_k	Whenever a new message is created L_k is set to be L (an algorithm parameter) and at each transfer L_k is updated based on the principle used in spray and wait.
Size of the message, s_k	The size of the message represented in bytes.
Hop Count of the message, H_k	At the message creation time H_k is set to be 0 and whenever the message is forwarded to another node the value of H_k is incremented by 1.
Time to Live, TTL_k	Whenever a message is created a TTL value in minutes is assigned with the message. This TTL denotes the number of minutes the message will be active in the network. When this time expires the message will be dropped immediately. This TTL is dissimilar to the TTL field used in the IPv4 packet header where the TTL denotes the maximum number of hops a node can traverse in the network.
List of Visited Nodes, LVN_k	Initially this list is empty. Before forwarding any message to the neighbor the sender adds itself to the list and thus the list grows. This is how LVN_k keeps track of all the visited nodes.
Source of the message, src_k	The identifier of the source of the message
Destination of the message, $dest_k$	The identifier of the destination of the message

Since our goal is to maximize the delivery ratio in the opportunistic environment with minimum network overhead, we use quota based routing with an efficient buffer management strategy. Like many other opportunistic routing protocols, such as - PROPHET [23], MaxProp [5], ORWAR [27] we rank the messages stored in the node buffer. We also use this ranking of the messages to choose the next message either to forward or delete. However, we use two distinct lists of message indices: “forward list” and “delete list” to schedule the next message to forward and to schedule the next message to delete respectively.

Like MaxProp [5] and ORWAR [27] each node of TBR also keeps a list of the messages which have already been delivered. Whenever a message reaches its destination, the destination node inserts the message ID into this list. Nodes exchange and update their list of acknowledged messages when they meet each other. This acknowledgement mechanism results in increased delivery ratio while reducing overhead.

4.3 TBR Protocol

A message can arrive at a node in TBR from any neighbor node or from the application layer of the node itself. Each message is tagged with a replication count (L_k), which denotes the intended number of copies of the message that the node has to spray. The initial value of L_k for any message is set to be L. Here, L is an algorithm parameter which can be tuned to achieve better results. L_k is updated after each successful transmission of a message.

When two nodes meet each other they first exchange their summary vector followed by the list of acknowledged messages. If the nodes posses any message that has been found delivered from the list of acknowledged messages, it deletes that message from its buffer. If a node has some messages destined to the contacted neighbor, it passes those messages to the neighbor first. Then, a node attempts to forward the messages from its buffer. While forwarding, TBR picks successive messages starting from the top of the “forward list” with $L_k > 1$. Like Spray and Wait [31], L_k is divided by 2 before the message is forwarded to the next hop neighbor. When L_k of a message becomes 1, the message is not forwarded to any intermediate nodes. Rather it waits in the current node for its destination. A message with L_k value of 1 is either directly delivered to the destination, whenever such contact becomes available or dropped from the buffer whenever the message TTL expires.

Like ORWAR [27] protocol, TBR computes a maximum transmittable message size (s_{max}) of a contact whenever a new contact is available. Like ORWAR, we forward a message to a neighbor only if its size fits into s_{max} of the contact in order to reduce the number of retransmissions.

To avoid a routing loop, each message in TBR includes a message header with a list of previously visited nodes (LVN). If any neighbor of the node is a member of the LVN of the message, then the node will not forward this message to that neighbor again.

4.4 Buffer Management

Due to limited contact opportunities in OpNets, all the nodes require a buffer to temporarily store the messages. Since the buffer capacity in the nodes is limited, routing protocols also need to address a buffer management mechanism. In TBR, we use two logically separate lists: forward list (FL) and delete list (DL). Forward list stores the indices of the messages ranked by a priority (P_{k_f}) which is used to choose the next message to forward when a contact is available. The priority metric (P_{k_f}), used to rank a message (m_k) in the forward list, is calculated using Equation 4.1.

$$P_{k_f} = \frac{1}{H_k \times TTL_k \times s_k} \tag{4.1}$$

where,

TTL_k- Message Time To Live;

H_k- Message Hop Count and;

s_k- Message Size.

This P_{k_f} rewards messages with lower TTL and hop count and penalizes messages for higher size. The messages with lower TTL are preferred over the other messages as they have the earliest deadline. We observed that if only the messages with lower TTL are given preference, sometimes newer messages get stuck in the buffer and seldom get a chance of being propagated. Using hop count in message priority gives the new messages a head start. Thus preferring the messages with lower hop counts ensure network fairness. Finally, the message size is used as a part of the priority metric to guarantee that the shortest messages get the preference.

Each node usually stores all the messages in its buffer until the message is sent to the destination. A node will delete the copy of a message in the buffer if:

1. The message is timed out (TTL expires),
2. The node is notified of the delivery of the message by an acknowledgement, or
3. The node receives a higher priority message when the buffer is full.

Whenever messages are to be deleted due to a buffer overflow, they are deleted from the bottom of the delete list. Delete list keeps a list of indices of messages ranked by another priority metric (P_{k_d}). The calculation of P_{k_d} is shown in the Equation 5.1. This priority ensures that the largest messages with minimum number of copies left to spray are scheduled to be deleted first in case of buffer overflow or congestion.

$$P_{k_d} = \frac{L_k}{s_k} \tag{4.2}$$

The next few pages presents an algorithmic depiction of TBR protocol.

Algorithm 1 *Procedure* TBR:*contact*(i, j)

{When node i contacts node j this procedure is called}

```
send $ack_i$ to $j$
receive $ack_j$ from $j$
$ack_i \leftarrow ack_i \cup ack_j$
for each message $m_k \in Buffer_i$ do
    if $m_k \in ack_i$ then
        remove $m_k$ from $Buffer_i$
    end if
    if $TTL_k$ expires then
        remove $m_k$ from $Buffer_i$
        remove $m_k$ from $ack_i$
    end if
end for
$s_{max} \leftarrow compute\_s_{max}(i, j)$
while $s_{max} > 0$ do
    for each message $m_k \in Buffer_i$ where $dest(m_k) = j$ do
        $deliver(m_k, j)$          {deliver messages which are destined to $j$}
        if $isDelivered(m_k, j) = true$ then
            $s_{max} \leftarrow s_{max} - s_k$
            remove $m_k$ from $Buffer_i$
            $ack_i \leftarrow ack_i \cup \{m_k\}$
        end if
    end for
end while
$sendMessage()$
```

Algorithm 2 *procedure* TBR:$receiveMessage(m_k)$

```
{When node $i$ receives any message this procedure is called}
  if $dest(m_k) = i$ then
    $ack_i \leftarrow ack_i \cup \{m_k\}$
  else
    $p_{k_f} \leftarrow \frac{1}{H_k \times TTL_k \times s_k}$    {$p_{k_f}$ is the forward priority}
    $p_{k_d} \leftarrow \frac{L_k}{s_k}$    {$p_{k_d}$ is the delete priority}
    if $Buffer_i$ is full then
      if $p_{k_r} > p_{last(DL_i)}$ then
        replace $m_k$ with $last(DL_i)$ in $Buffer_i$
      else
        insert $m_k$ to $Buffer_i$
      end if
    end if
    update $FL_i$ based on $p_{k_f}$
    update $DL_i$ based on $p_{k_d}$
  end if
```

Algorithm 3 *function* TBR:$compute\_s_{max}(i, j)$

```
  $send(\vec{v_i}, r_i, x_i, y_i)$ to $j$
  $reeceive(\vec{v_j}, r_j, x_j, y_j)$ from $j$
  return $s_{max}$
```

Algorithm 4 *procedure* TBR:$sendMessage()$

```
while $s_{max} > 0$ do
  for each message $m_k \in FL_i$ with $L_k > 1$ do
    for all connections $con(i, x)$ do
      if $x \notin LVN(m_k)$ && $m_k \notin Buffer_j$ && $s_k < s_{max}$ then
        $deliver(m_k, x)$          {with $L_{kx} = ceil(L_{ki}/2)$}
        $L_{ki} \leftarrow L_{ki}/2$
        $s_{max} \leftarrow s_{max} - s_k$
      end if
    end for
  end for
end while
```

Chapter 5

The ProbRoute Protocol

Our second protocol is the Probabilistic Quota Based Adaptive Routing Protocol (ProbRoute). One criticism of our TBR protocol could be its blind forwarding of messages. In TBR, a node forwards a message to its neighbor blindly without considering whether the neighbor is a good choice to guide the message to its destination or not. To deal with this shortcoming our ProbRoute protocol introduces a "*Contact Probability*" to ensure guided message forwarding. Moreover, ProbRoute also introduces a Priority Queue where the messages are ranked based on an adaptive priority metric. To explain our ProbRoute protocol, we first present our system model. Then we provide a brief narration of our protocol. Finally, we present the buffer management scheme and the calculations for contact probability used in ProbRoute. We conclude the chapter with a detailed algorithmic depiction of our protocol.

5.1 System Model

Opportunistic Networks can operate in many environments- on vehicles, zebras [19], under water sensors, and pedestrians. Here, we model OpNet as a set of mobile nodes, where we know neither the message arrival rate nor the meeting schedule in advance. Therefore, the routing is completely opportunistic. Two nodes can directly communicate only when they are within each other's transmission range. We also assume a node has limited buffer for the messages. A node can deliver a packet to a destination directly or via some intermediate nodes. We do not consider the fragmentation of the messages in this paper. We assume three stages in OpNet operation-

1. **Neighbor Discovery:** When peers are within each other's transmission range, they must first discover each other before starting any data transfer.

2. **Data Transfer:** Peers can exchange data when they are in contact but the contact time is unknown to the nodes.

3. **Buffer Management:** As the buffer for the messages is limited, a node must have some algorithm to delete messages while the network is under congestion.

Our goal is to maximize the delivery ratio in the opportunistic environment while reducing delay and network overhead. We combine the bests from both flooding and quota based routing protocols. Similar to PROPHET [23] and MaxProp [5], we use a probability metric to forward a message from one node to another when they meet. We use this probability metric to forward a message in the direction that maximizes the probability of delivering the messages to its destination. PROPHET and MaxProp update their probability metric at each contact between two nodes. We, however, update our probability metric at the end of an interval. This interval based update allows us to support MIMO capabilities at the nodes. The nature and the detail calculation of our probability metric are also different from that of PROPHET and MaxProp. We describe our probability metric and its calculation in Section 5.4.

Like all other quota based routing protocols, we limit the number of allowable copies of a message in the network by using a replication factor. However, unlike the existing quota based routing protocols we don't forward a message to a neighbor node blindly. We rather use the above mentioned probability metric to achieve guided forwarding.

Like many other opportunistic routing protocols, such as- MaxProp [5], PROPHET [23], EBR [24], ORWAR [27], we rank the messages in a node. We use the rank of a message to choose the next message either to forward or to delete. However, our ranking mechanism differs from the ranking mechanism used in [5], [23], [24] and [27].

5.2 The ProbRoute Protocol

In our ProbRoute protocol, each node i maintains following data structures-

- A list of delivered messages DL_i,
- A list of Contact Probabilities $CP_{(i,x)}$, for all the known destination x,
- A priority queue PQ_i to store the messages according to their rank.

DL_i keeps the list of the messages which are already acknowledged by the corresponding destination. Whenever a message reaches to its destination node, it inserts the message ID into its DL. Nodes exchange and update their DL when they meet each other.

We name the probability metric mentioned in section 5.1 “Contact Probability” and represent it as $CP_{(i,x)}$, where it denotes the probability of a node i to meet another node x. The details of our “Contact Probability” and its calculation are described later in this Section.

The priority queue stores the messages according to their priority. The management mechanism of the queue and the calculation of the message priority are shown in Section 5.3.

A node can get a message from its application layer or from any other node. Every message has a header that includes a counter L_k which denotes the intended number of replicas in the network for that message. When a new message is created, L_k for the message is initialized to L. Here, L is an algorithm parameter which can be tuned to achieve better results.

When two nodes meet each other, they first exchange their summary vectors followed by the contact probability set and DLs. If the node has any message that has been found delivered from DLs, it deletes that message from its queue. If a node has some messages destined to the contacted neighbor, it first passes those messages to the neighbor. Finally, a node attempts to forward messages from its priority queue to the neighbor. Starting from the top of the priority queue our algorithm picks successive messages only with $L_k > 1$ to forward. The message is forwarded to the neighbor if it has the higher contact probability for the destination of the message than that of the current node. In case of multiple contacts at the same time, a message is forwarded to the node which has the maximum contact probability for the message's destination. Like ORWAR [27] protocol, we compute a maximum transmittable message size (s_{max}) of a contact. We use ORWAR's method to compute (s_{max}). Like ORWAR, we also forward a message to a neighbor only if its size is less than or equal to (s_{max}) of the neighbor's contact in order to reduce the number of retransmissions. To avoid a routing loop each message header includes a list of visited nodes (LVN). If any neighbor of the node is in LVN of a message, then the node will not send this message to that neighbor again.

Like Spray and Wait [31], L_k value in the message header is divided by 2 before the message is forwarded to the next hop neighbor. When L_k of a message becomes 1, it is not forwarded to an intermediate node rather it waits in the current node for its destination. A message with L_k value of 1 is directly delivered to the destination whenever such contact becomes available. If a node does not meet the destination of a message before its time-to-live (TTL) expires, it simply deletes the message from its priority queue.

5.3 Queue Management

Due to limited storage and limited transmission opportunities in OpNet, each node needs to employ a buffer to temporarily store the messages. In ProbRoute, we use a priority queue in each node where each message (m_k) is ranked by a priority $P_k = \frac{L_k}{S_k}$. Here, L_k has its usual meaning as described in the previous section and S_k denotes the size of the message. The L_k of a message changes after each transfer of the message. For this reason message priority adapts and all the messages in the priority queue get the fair opportunity

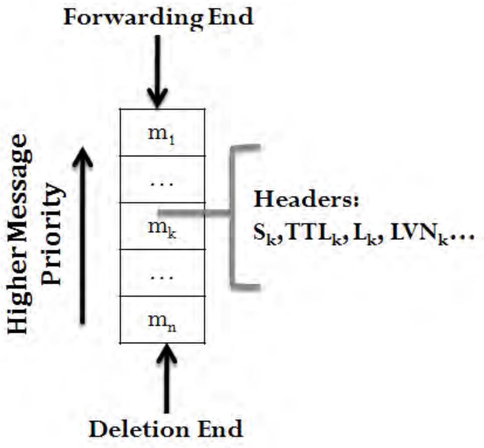


Figure 5.1: Priority Queue

to be forwarded, whereas in other routing protocols, some of the messages get stuck at the lower end of the buffer. Again, S_k is used as a part of the priority to ensure that the shortest job gets the preference. Thus using this adaptive message priority ProbRoute improves the chance of delivery of the message.

Since the buffer space in the nodes are limited, some situations may arise where we need to delete some messages from the queue. A message from the priority queue can only be deleted if

1. A copy of the message is already known to be delivered through the neighbors DL.
2. The TTL of the message expires or
3. The node receives a new message with a higher priority, when the priority queue is full.

When messages are to be deleted due to the last reason, they are deleted from the bottom of the queue. Figure 5.1 shows the structure of the priority queue.

5.4 Calculation of the Contact Probability

In our ProbRoute protocol we used the assumption that the nodes that met the destination more frequently in the past are more probable to meet the destination again in near future. That is why, to calculate the contact probabilities, each node i uses two pieces of local information for each known contact j - the past contact probabilities ($CP_{(i,j)_{old}}$) and the number of contacts (C_j) in the current interval. The count (C_j) of the contacts for any node j in the current interval is incremented whenever the node is encountered. Therefore, the contact probability ($CP_{(i,j)}$) of a particular node j for the current interval can be calculated as Equation 5.1

$$\forall j \in N : CP_{(i,j)_{current}} = \frac{C_j}{\sum_{p \in N} C_p} \tag{5.1}$$

We also give emphasis on both the previous contact probability ($CP_{(i,j)_{old}}$) and the current contact probability of a node. For this reason, we take a weighted average of these two probabilities in order to calculate the new contact probability ($CP_{(i,j)_{new}}$) as equation 5.2

$$\forall j \in N : CP_{(i,j)_{new}} = (1 - \alpha) \times CP_{(i,j)_{old}} + \alpha \times \frac{C_j}{\sum_{p \in N} C_p} \tag{5.2}$$

Here N is the set of nodes in the network and α is the weight parameter.

Algorithm 5 *Procedure* ProbRoute:$contact(i, j)$

{When node i contacts node j this procedure is called}

```
$C_j \leftarrow C_j + 1$
send $ack_i$ to $j$
receive $ack_j$ from $j$
$ack_i \leftarrow ack_i \cup ack_j$
for each message $m_k \in PQ_i$ do
    if $m_k \in ack_i$ then
        remove $m_k$ from $PQ_i$
    end if
    if $TTL_k$ expires then
        remove $m_k$ from $PQ_i$
        remove $m_k$ from $ack_i$
    end if
end for
send $CP_i$ to $j$
receive $CP_j$ from $j$
$s_{max} \leftarrow compute\_s_{max}(i, j)$
while $s_{max} > 0$ do
    for each message $m_k \in PQ_i$ where $dest(m_k) = j$ do
        $deliver(m_k, j)$          {deliver messages which are destined to $j$}
        if $isDelivered(m_k, j) = true$ then
            $s_{max} \leftarrow s_{max} - s_k$
            remove $m_k$ from $PQ_i$
            $ack_i \leftarrow ack_i \cup \{m_k\}$
        end if
    end for
end while
$sendMessage()$
```

Algorithm 6 *procedure* ProbRoute:$endOfInterval()$

```
{This procedure is called at the end of each update interval}
if this is the first interval then
    for all $j \in N$ do
        $CP_{(i,j)} \leftarrow \frac{C_j}{\Sigma_{\forall x \in N} C_X}$    {N is the set of all Nodes}
    end for
else
    for all $j \in N$ do
        $CP_{(i,j)} \leftarrow (1-\alpha) \times CP_{(i,j)old} + \alpha \times \frac{C_j}{\Sigma_{\forall x \in N} C_X}$    {N is the set of all Nodes}
    end for
end if
```

Algorithm 7 *procedure* ProbRoute:$sendMessage()$

```
while $s_{max} > 0$ do
    for each message $m_k \in PQ_i$ with $L_k > 1$ do
        for all connections $con(i,x)$ do
            Find a node x such that $CP_{x,dest(m_k)}$ is maximum
        end for
        if $m_k \notin PQ_j$ && $CP_{i,dest(m_k)} < CP_{x,dest(m_k)}$ && $s_k < s_{max}$ && $x \notin LVN(m_k)$
        then
            $deliver(m_k, x)$    {with $L_{kx} = ceil(L_{ki}/2)$}
            $L_{ki} \leftarrow L_{ki}/2$
            $s_{max} \leftarrow s_{max} - s_k$
        end if
    end for
end while
```

Algorithm 8 *function* ProbRoute:$compute\_s_{max}(i, j)$

```
  $send(\vec{v_i}, r_i, x_i, y_i)$ to $j$
  $reeceive(\vec{v_j}, r_j, x_j, y_j)$ from $j$
  return $s_{max}$
```

Algorithm 9 *procedure* ProbRoute:$receiveMessage(m_k)$

```
{When node $i$ receives any message this procedure is called}
  if $dest(m_k) = i$ then
    $ack_i \leftarrow ack_i \cup \{m_k\}$
  else
    $pr_k \leftarrow \frac{L_k}{s_k}$        {$pr_k$ is the message priority}
    if $PQ_i$ is full then
      if $pr_k > pr_{last(PQ_i)}$ then
        replace $m_k$ with $last(PQ_i)$
      else
        insert $m_k$ to $PQ_i$
      end if
    end if
    sort $PQ_i$ based on priority
  end if
```

Chapter 6

Performance Evaluation

To evaluate our protocols, we first present the metrics based on which we evaluated our protocols, followed by a brief description of our simulation setup and mobility models. After that we present a comprehensive performance comparison of TBR and ProbRoute with four other popular OpNet routing protocols and finally we compare the performances of our two proposed protocols, TBR and ProbRoute.

6.1 Metric

Usually OpNet routing protocols are evaluated based on three metrics- Delivery Ratio (DR), Median Latency (Lat), and Network Overhead ($Over$).

- **Delivery ratio** (Equation 6.1) is defined by the ratio of the total number of messages delivered (m_{del}) to the total number of messages created (m_{cre}).

$$DR = \frac{m_{del}}{m_{cre}} \tag{6.1}$$

- **Median Latency** (Equation 6.2) is the median of the time required for a message to reach its destination.

$$Lat = Median(\forall_{delivered}(t_{del} - t_{cre})) \tag{6.2}$$

where,

t_{cre} = message creation time;

t_{del} = message delivery time;

$delievered$ = list of delivered messages.

- **Overhead** (Equation 6.3) is defined as the ratio of the total number of messages relayed (m_{rel}) which does not to the total number of messages delivered.

$$Over = \frac{m_{rel}}{m_{del}} \tag{6.3}$$

The metrics such as delivery ratio and end-to-end delay (latency) show the effectiveness of the protocol and network overhead measures resource friendliness of the protocol.

In order to get a comprehensive comparison among the protocols, we use a composite metric along with the traditional metrics. The composite metric illustrates the relative relationship between the primary metrics. The composite metric (CM) shown in Equation 6.4 gives credit for higher delivery ratio, while penalizes for both longer latency and higher overhead.

$$CM = DR \times \frac{1}{Over} \times \frac{1}{Lat} \tag{6.4}$$

6.2 Simulation Setup

To evaluate the effectiveness of our protocols, we compare the performance of our protocols with the popular OppNet routing protocols using ONE (Opportunistic Network Environment) [3] [21]. ONE is a powerful tool for generating different movement models, running simulation with various routing protocols, visualizing simulations in real time and generating results and post processing the results. ONE version 1.3 comes with the implementation of the following routing protocols- Epidemic [33], PROPHET [23], Spray and Wait (SNW) [31] and MaxProp [5]. In our simulation, we compared our result with these protocols. We also implemented the ORWAR and EBR protocol using ONE to compare their performance with that of our protocols. All the Quota based protocols limit the message flooding by fixing the number of replicas per message. Simulation result shows that, the protocols achieve good performance in our simulated scenario with an $L = 6$. That is why we used a fixed replication factor ($L = 6$) for all the quota based protocols. ORWAR allows higher replication factor ($L + \Delta$) for high priority messages and lower replication factor ($L - \Delta$) for the low priority messages. For the simulation we used $\Delta = 2$. For the PROPHET [23] protocol, we assumed the initialization constant $P_{(a,b)} = 0.75$, the aging constant $\gamma = 0.98$ and the scaling constant $\beta = 0.25$. In order to find the appropriate value for α of Equation 5.2 as well as for the interval we ran our simulation varying these two parameters. From those simulation runs, we found that $\alpha = 0.5$ and interval period, $T = 60$ sec give better performance.

Since opportunistic networks operate in many different environments, we use two different mobility model to cover a wide variety of opportunistic environments. We use a map-driven movement model to simulate the vehicular mobility model and traditional random way point (RWP) movement to simulate random walks. For both map based model and RWP model we evaluated the impact of the messages size and the number of nodes on the protocols.

In our simulation, we evaluated the impact of the message size and the network size on the different metrics to compare the performance of different protocols. Here, the total number of nodes in a network denotes its size. To evaluate the impact of message size on the metrics, we start our simulation in a city environment of 100 nodes with 1500 messages. The message size (S) is normally distributed with an average size of 2MB. Then, we gradually decrease the message size and proportionately increase the number of messages in order to maintain a constant load in the network.

In order to assess the impact of node density on the network, we vary the number of nodes in the network keeping the number of messages fixed at 10000 and the message size (S) normally distributed with average 500kB. We start with 50 nodes and at each iteration, increase the number of nodes by 25 until it reaches 200.

6.3 Mobility Models

As OpNets can operate in many different environments, we used two different mobility models in our simulation. We use a Map-driven vehicle based movement model to analyze the effectiveness of various protocols in a city environment and a RWP movement model to evaluate the performance of the protocols in a random scenario.

6.3.1 Map-based Vehicular Model

Map-based vehicular model restricts the movements of the network nodes to actual streets in an imported map. In our simulation, we used a map of 4500m x 3400m section of Helsinki, Finland. We used three types of nodes in our simulation- cars, trams and pedestrians. For pedestrians, cars, and trams transmission ranges are assumed to be 10m, 20m, and 20m respectively. Transmission speed for all the nodes is assumed 250KBps (2Mbps). We also assume a buffer of 100MB for trams, 20MB for cars, and 10MB for pedestrians. Cars and pedestrians move with a speed within [2.7, 13.9] m/s and [0.5, 1.5] m/s respectively with random pause. Speed of the trams vary within [7,10] m/s. In each of the iteration we keep the node distribution as- 12% trams, 28% cars and 60% pedestrians.

6.3.2 Random Way Point Movement Model

In RWP movement model, nodes move around in random zig-zag paths. In our simulation with RWP movement model, we use nodes having a transmission range of 30m and buffer space of 50MB. The nodes can move around randomly in a 4500m x 3400m playground with a speed of 0.5m/s to 5m/s and pause at some places for some time between 0 and 120 seconds.

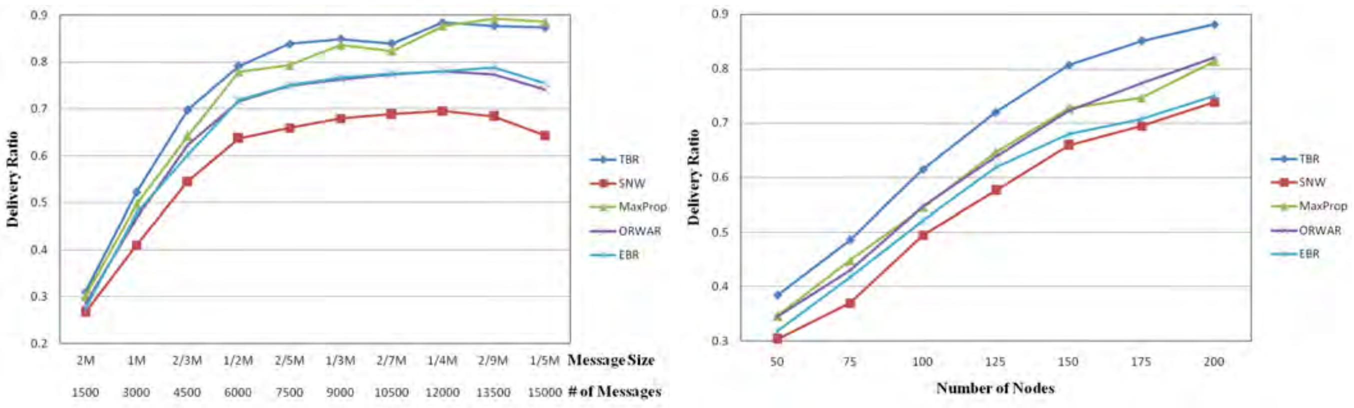


Figure 6.1: TBR - Map Based Vehicular Movement Model- Delivery Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

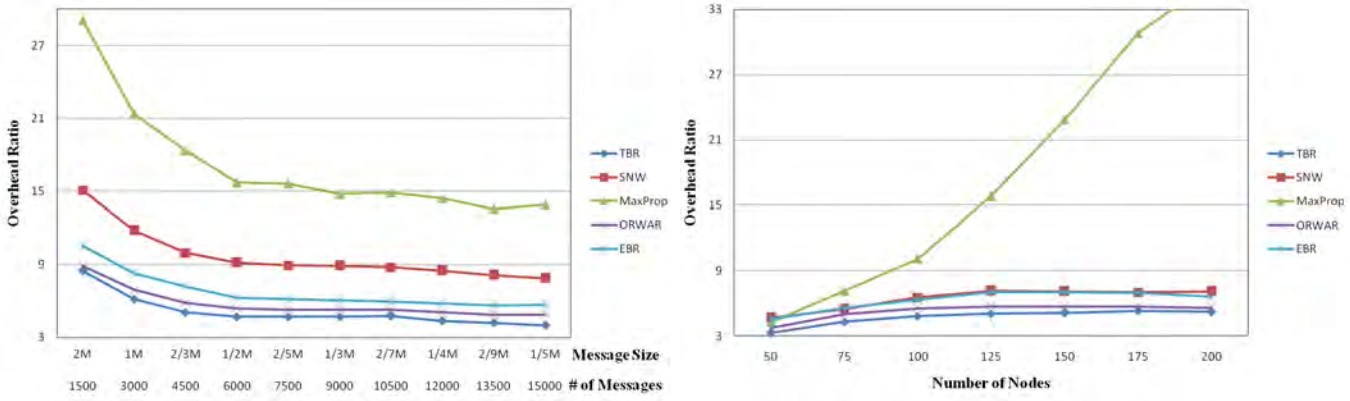


Figure 6.2: TBR - Map Based Vehicular Movement Model- Overhead Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

6.4 Simulation Results

6.4.1 TBR Protocol

At first, we present the results of Map-based vehicular movement model. Figure 6.1(a) and 6.1(b) show the delivery ratios of different opportunistic routing protocols varying the message size and varying the number of nodes respectively. These figures illustrate that, TBR achieves more than 10% higher delivery ratio than that of all the other quota based routing protocols. TBR also achieves higher delivery ratios than to that of MaxProp. In

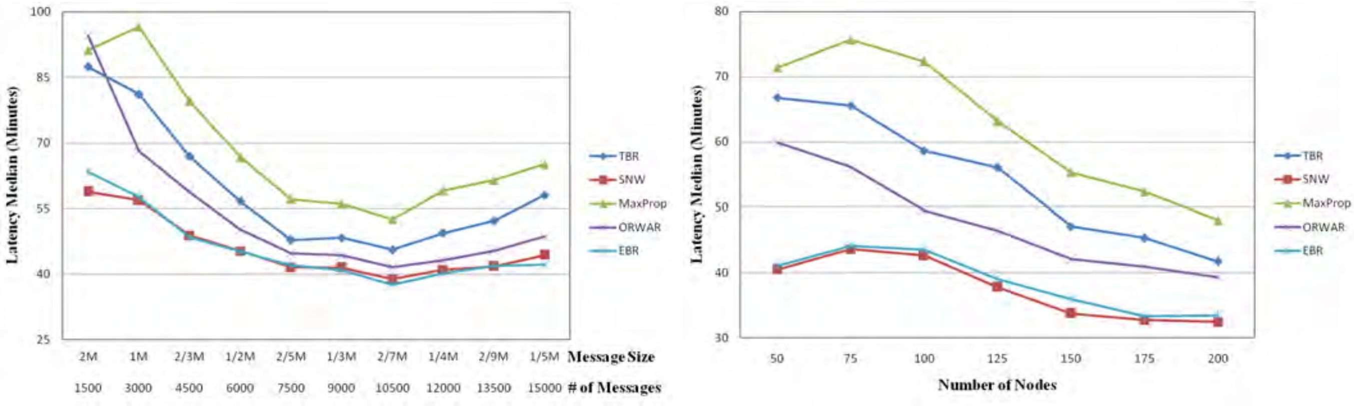


Figure 6.3: TBR - Map Based Vehicular Movement Model- Latency Median: (a) Varying Message Size, (b) Varying Number of Nodes

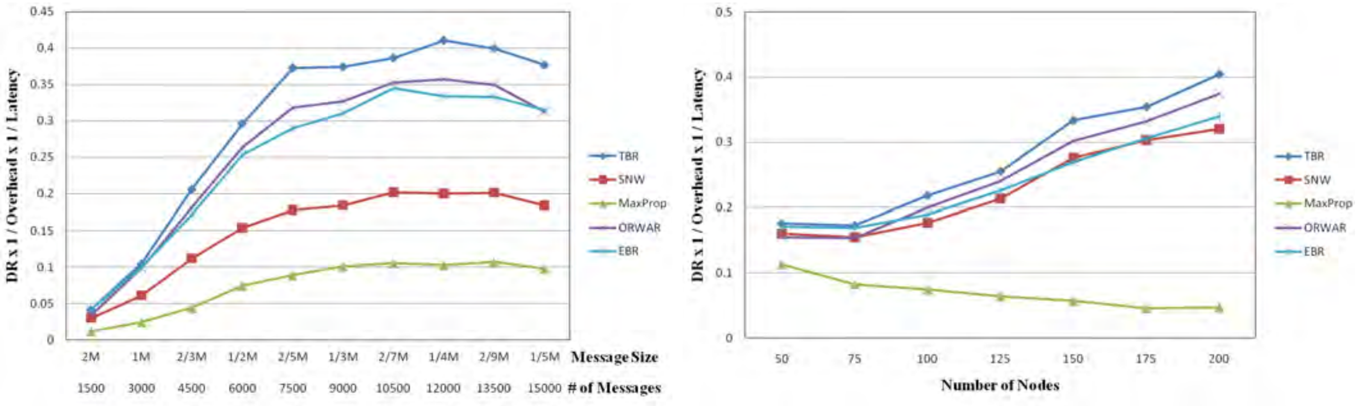


Figure 6.4: TBR - Map Based Vehicular Movement Model- Composite Metric: (a) Varying Message Size, (b) Varying Number of Nodes

Figure 6.1(a) and 6.1(b) TBR achieves such high delivery ratios as it sprays the messages using an efficient buffer management strategy. As we have seen, when a contact opportunity appears protocols like SNW and EBR forward the oldest message from the buffer, while ORWAR selects the message based on only its utility and size. On the contrary our protocol TBR considers the hop count, TTL and size of the messages to choose the best message to forward. So, the use of hop count and TTL to choose the best message to forward allowed TBR to achieve 10% to 15% higher delivery ratio than SNW, EBR and ORWAR.

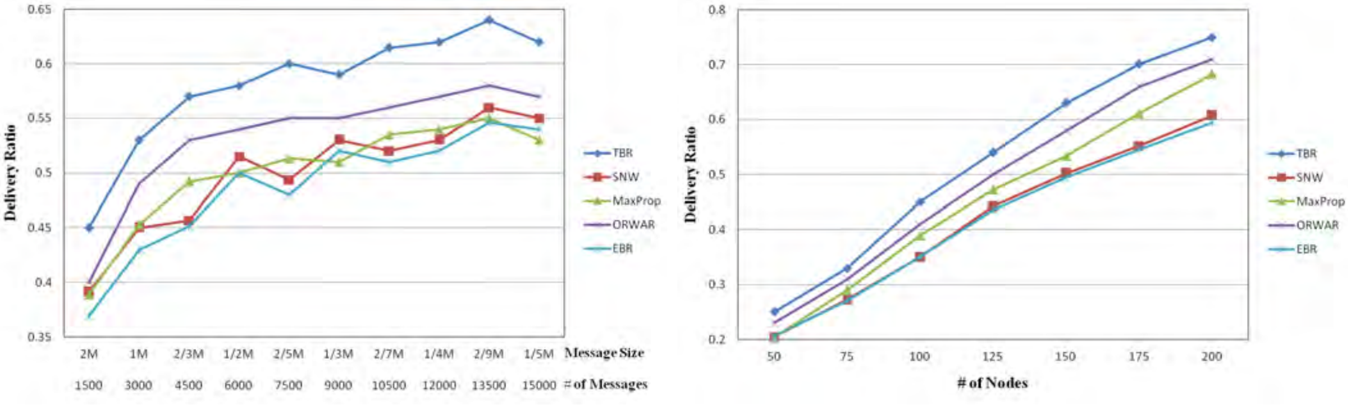


Figure 6.5: TBR - Random Way Point Movement Model- Delivery Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

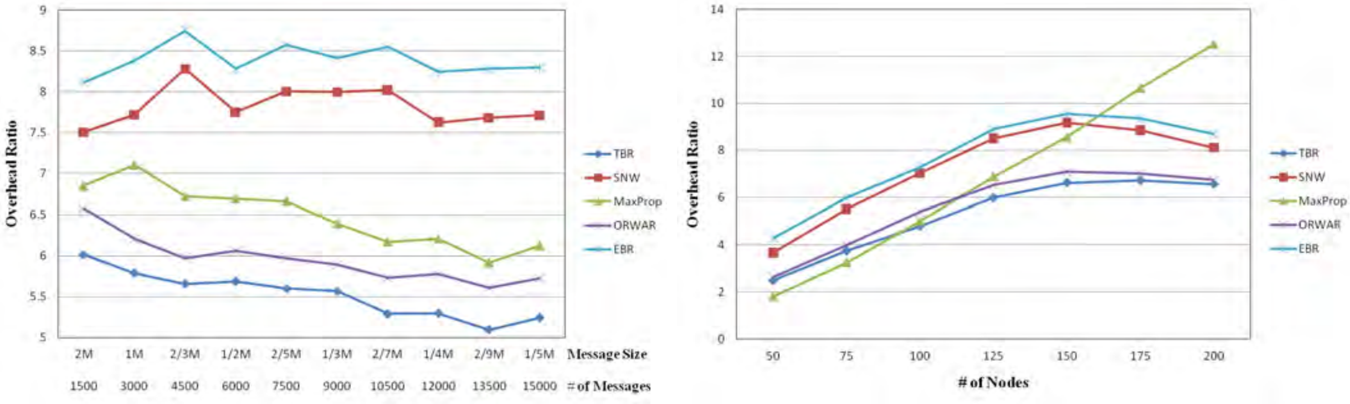


Figure 6.6: TBR - Random Way Point Movement Model- Overhead Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

Figure 6.2 compares the overhead ratio of different protocols. Figure 6.2(a) shows the impact of changing message size, while Figure 6.2(b) demonstrates the effect of changing the number of nodes on the protocols. It is obvious from the figure that the quota based protocols are more resource friendly than their flooding based counterparts as they require lower overheads. The figure also shows that, TBR is the most resource friendly protocol as its overhead is the minimum.

As far as latency is concerned, which is presented in Figure 6.3, our protocol, TBR achieves 15% lower latency on average than that of the best flooding based protcol-

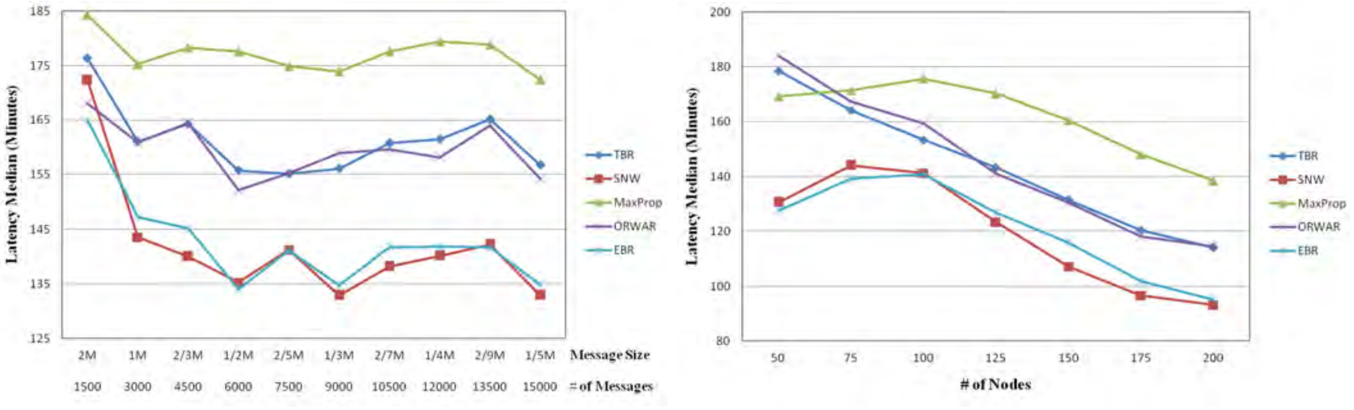


Figure 6.7: TBR - Random Way Point Movement Model- Latency Median: (a) Varying Message Size, (b) Varying Number of Nodes

MaxProp. However, TBR performs poor in terms of latency compared to SNW, EBR and ORWAR. The latency is generally computed over the messages that have been delivered. Many routing protocols like - SNW, EBR, and ORWAR deliver small hop messages but do not deliver most of the high hop messages. As a result the average latency in those protocols remains low . TBR, however, successfully delivers many large hop messages along with the low hop messages. Which contributes to TBR's high average latency.

In Figure 6.4, the performance of the protocols in terms of a composite metric is analyzed. Figure 6.4(a) and 6.4(b) show the composite performance of the protocols by varying the message size and the network size respectively. These figures show that TBR achieves the best performance in terms of this composite metric in both the cases. Although TBR requires a comparatively higher latency than that of SNW, EBR, and ORWAR, its high delivery ratio and low overhead enable TBR to achieve around 10%-15% higher composite performance than that of the second best protocol ORWAR.

Now, we will present the results for Random way point (RWP) movement model. From Figure 6.5(a) and 6.5(b), we see that even in RWP movement model TBR achieves better delivery ratio than that of all other existing protocols. The figures also show that the delivery ratios achieved by the protocols in RWP model is less than that obtained with the Map-based vehicular movement model. This is because of the random movement

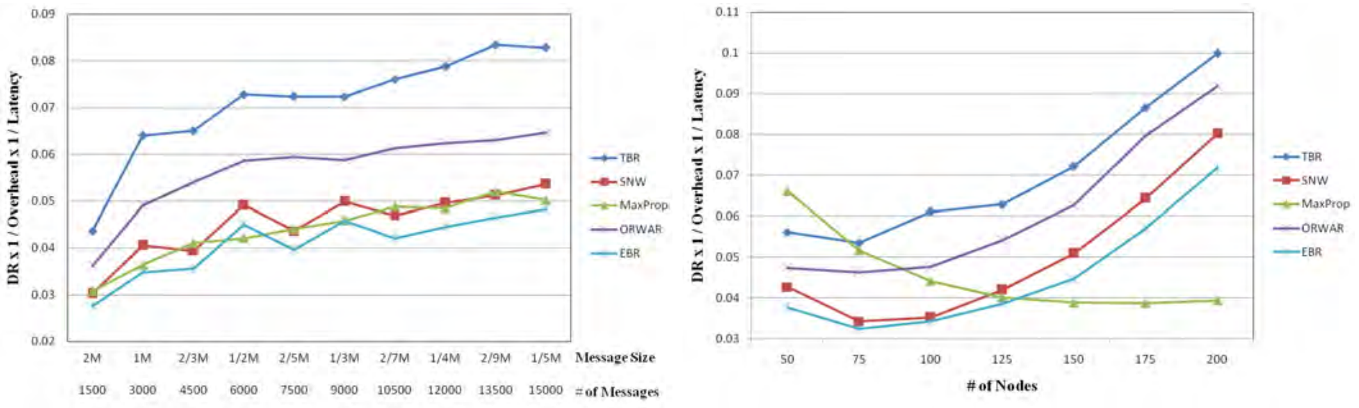


Figure 6.8: TBR - Random Way Point Movement Model- Composite Metric: (a) Varying Message Size, (b) Varying Number of Nodes

of nodes in RWP movement model, which effects a reduction in the number of node contacts. As a result the overall performance of all the protocols suffer. The flooding based protocol, MaxProp, performs poorly in terms of delivery ratio compared to TBR and ORWAR. In our opinion, MaxProp's low delivery ratio is due to its assumption that the past information of node meeting is a good indication of future node meeting. This assumption does not fit well into the RWP movement model. In terms of overhead ratios shown in Figure 6.6(a) and 6.6(b), the performances of the protocols are similar to that of the Map based movement model. Figure 6.7 shows the performance of the protocols in terms of latency. Figure 6.7(a) and 6.7(b) show that, TBR requires similar latency compared to that of ORWAR in RWP movement model. This can be explained by the fact that, because of the random movements neither of the protocol is able to transfer many high-hop messages successfully. As a result both achieve similar latency. Finally, the performance in terms of composite metric also demonstrates the superiority of our TBR protocol over the other protocols as shown in Figure 6.8.

6.4.2 ProbRoute Protocol

At first, we present the results for Map based movement model. Figure 6.9 shows the delivery ratio of different protocols. Figure 6.9(a) shows the impact of changing message

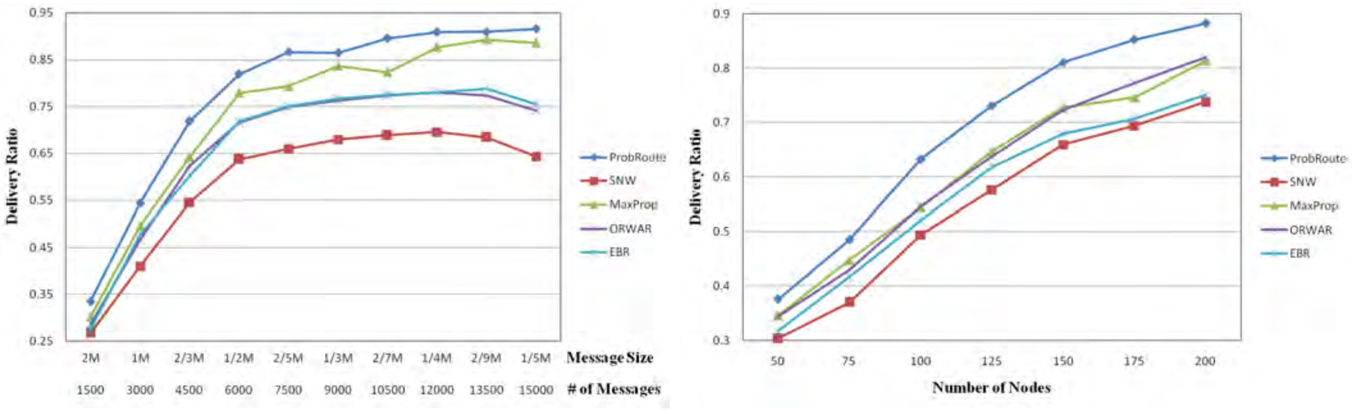


Figure 6.9: ProbRoute - Map Based Vehicular Movement Model- Delivery Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

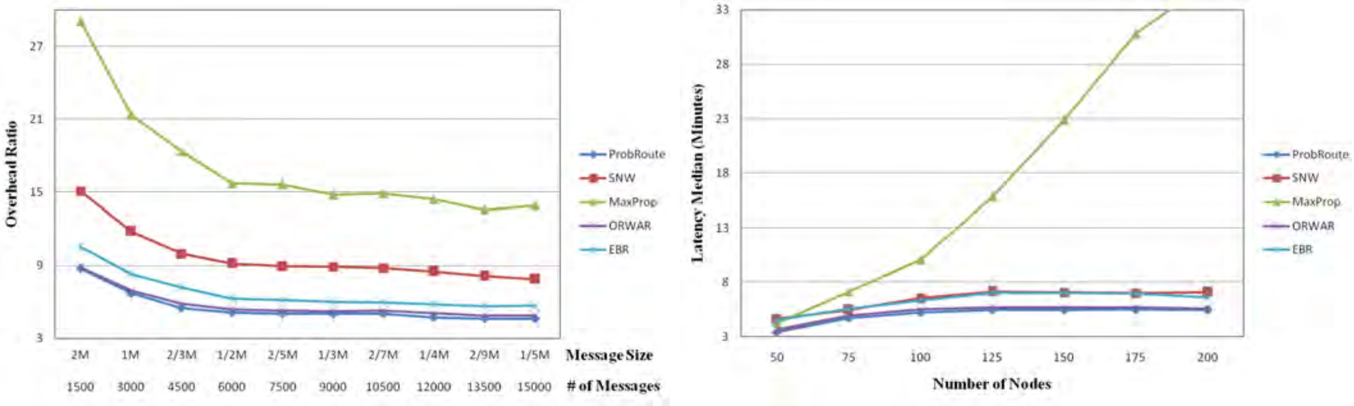


Figure 6.10: ProbRoute - Map Based Vehicular Movement Model- Overhead Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

size, while Figure 6.9(b) demonstrates the effect of changing the number of nodes on the protocols. These figures illustrate that, ProbRoute achieves cosistently higher delivery ratio than that of MaxProp and more than 15% higher delivery ratio than that of the other protocols. Maxprop achieves the second best delivery ratio, but their success is mainly due to the aggressive use of network resources. ProbRoute achieves the highest delivery ratio as it propagates the messages using an adaptive messages priority which gives all the messages a fair opportunity to be transferred. Again the previous quota based protocols (including TBR) generally forwarded the messages to the neighbors blindly without con-

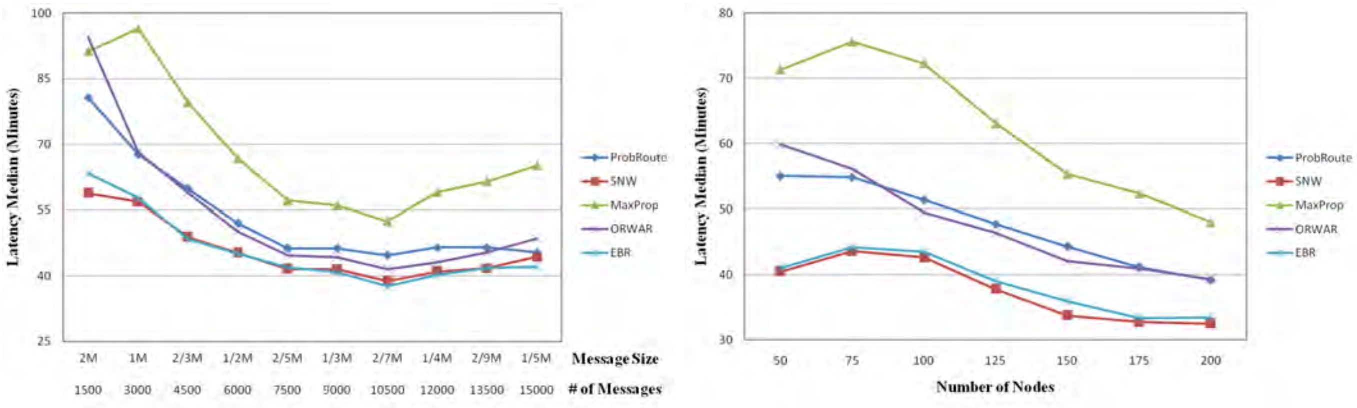


Figure 6.11: ProbRoute - Map Based Vehicular Movement Model- Latency Median: (a) Varying Message Size, (b) Varying Number of Nodes

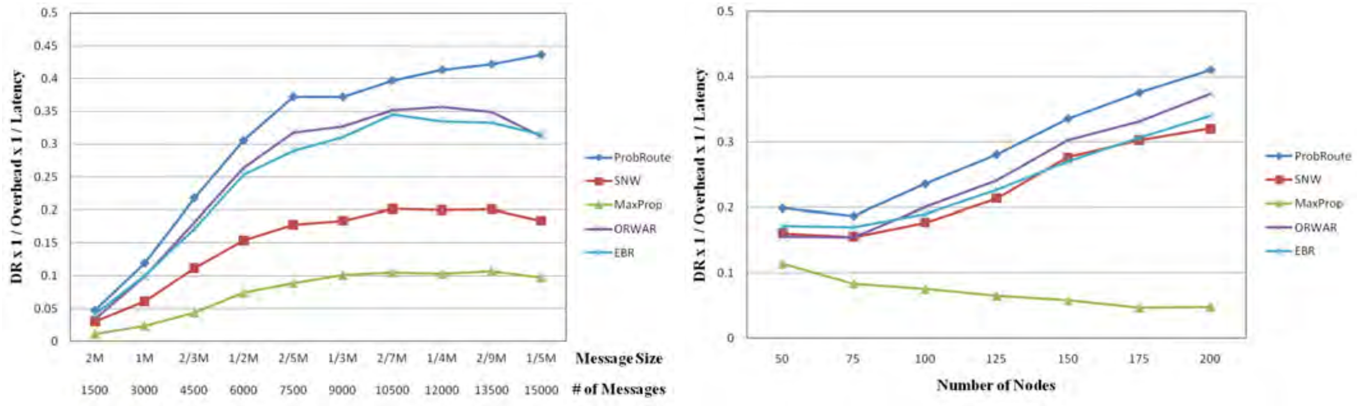


Figure 6.12: ProbRoute - Map Based Vehicular Movement Model- Composite Metric: (a) Varying Message Size, (b) Varying Number of Nodes

sidering whether the selected next hop would be a good choice to guide the message to its destination or not. Our protocol, ProbRoute however, forwards the messages to its neighbor based on a contact probability. That is why our ProbRoute protocol achieves 15% to 20% higher delivery ratio compared to that of the other quota based protocols, that is even 2%-4% higher than our earlier protocol TBR which did not use contact probability for choosing the next hop.

Figure 6.10 compares the overhead ratio of different protocols. It is obvious from the figure that the quota based protocols are more resource friendly as they require lower overheads than that of their flooding based counterparts. The figure also shows that, ProbRoute is the most resource friendly protocol as its overhead is the minimum. The effective queue management, the use of contact probability to forward messages, routing loop avoidance, contact volume computation, and the use of acknowledgements in ProbRoute protocol contribute to achieve this low overhead. Figure 6.10(b) illustrates that, the overhead of the flooding based protocol MaxProp increases proportionately as the number of nodes increases, whereas the overhead of the quota based protocols are more or less invariant to the network size.

As far as latency is concerned, which is studied in Figure 6.11, SNW protocol achieves the smallest latency followed by EBR and ORWAR. Our protocol, ProbRoute stands fourth in terms of latency. Two factors are mainly responsible for this higher latency in ProbRoute. Firstly, the messages in ProbRoute, will have to stay longer in the buffers in order to get a suitable neighbor (a neighbor with higher contact probability for the destination). Secondly, the delay is computed only over the messages that have been delivered. Many routing protocols quickly deliver messages that require a small number of hops, and do not deliver most of the high-hop messages. However, due to the use of contact probability, ProbRoute successfully transfers many high-hop messages which require a comparatively high latency.

In Figure 6.12, the performance of the protocols in terms of a composite metric is analyzed. Figure 6.12(a) and Figure 6.12(b) show that as the message size is reduced or the number of nodes in the system is increased, the composite metric values for all the protocols tends to increase. This is happening due to the increase in the number of node encounters as well as in the number of message deliveries. These figures show that ProbRoute achieves the best performance in terms of this composite metric in all the cases. Although ProbRoute has a comparatively higher latency than that of SNW, EBR, and ORWAR, its high delivery ratio and low overhead enable ProbRoute to achieve the best composite performance, which is nearly 15% to 20% higher than that of the second best protocol ORWAR.

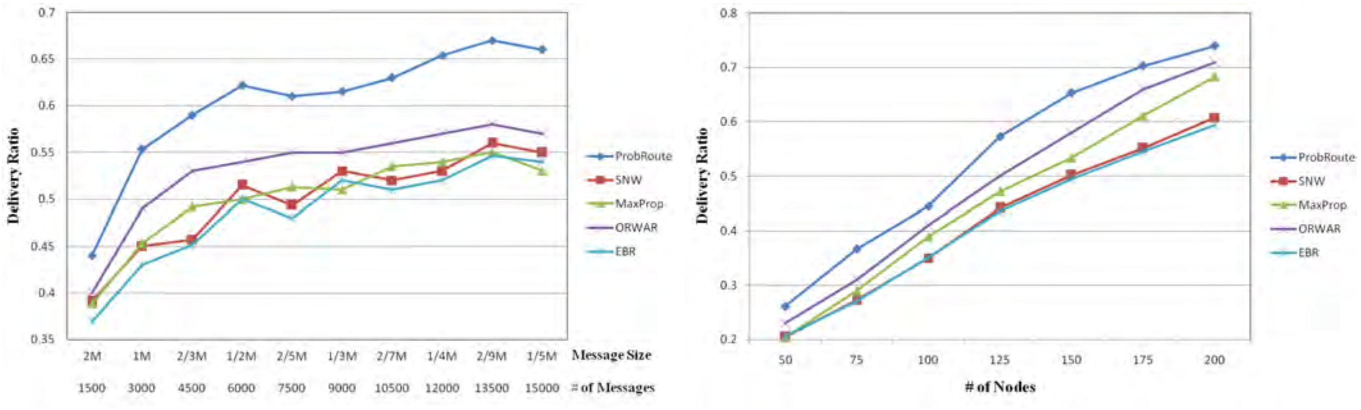


Figure 6.13: ProbRoute - Random Way Point Movement Model- Delivery Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

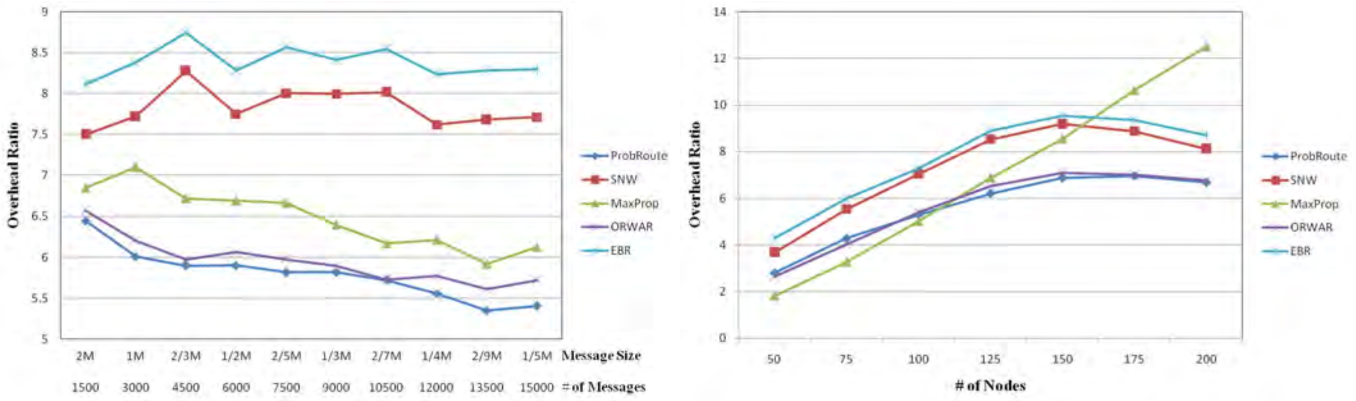


Figure 6.14: ProbRoute - Random Way Point Movement Model- Overhead Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

Now, we will present the results for Random way point(RWP) movement model. From Figure 6.13(a) and 6.13(b), we see that even in RWP movement model ProbRoute achieves better delivery ratio than that of all other existing protocols. The figures also show that the gap between ProbRoute and ORWAR is closer than that obtained with the Map based movement model. This is because our guided message forwarding has little impact on RWP movement model. The flooding based protocol, MaxProp, achieves less delivery ratio than that of ProbRotue and ORWAR. In our opinion, MaxProp's low delivery ratio is due to its assumption that the past information of node meeting is a good indication

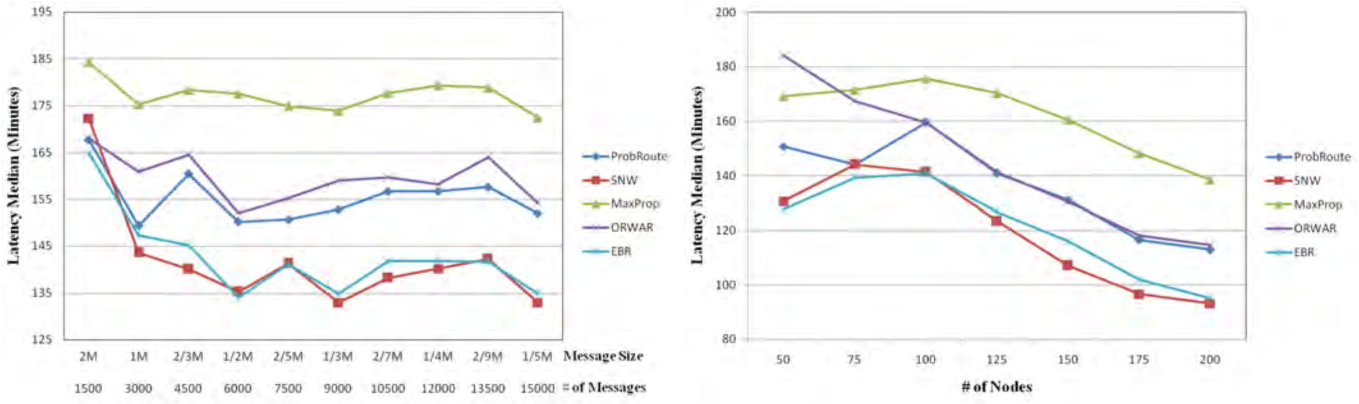


Figure 6.15: ProbRoute - Random Way Point Movement Model- Latency Median: (a) Varying Message Size, (b) Varying Number of Nodes

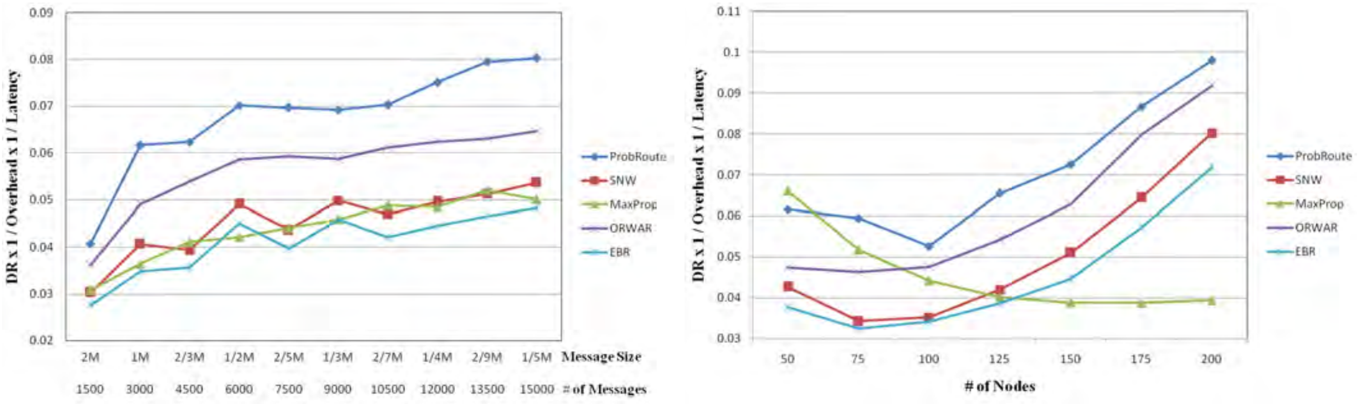


Figure 6.16: ProbRoute - Random Way Point Movement Model- Composite Metric: (a) Varying Message Size, (b) Varying Number of Nodes

of future node meeting. This assumption does not fit well into RWP movement model, where the nodes move randomly in the playground.

In terms of overhead ratios shown in Figure 6.14(a) and 6.14(b), the performances of the protocols are similar to that of the Map based movement model. Figure 6.15 shows the performance of the protocols in terms of delay. Figure 6.15(a) and 6.15(b) show that, ProbRoute requires less latency than that of ORWAR in RWP movement model. This can be explained by the fact that, as the node movements are random, none of the protocols is able to transfer many high-hop messages, however, ProbRoute can transfer the small-hop

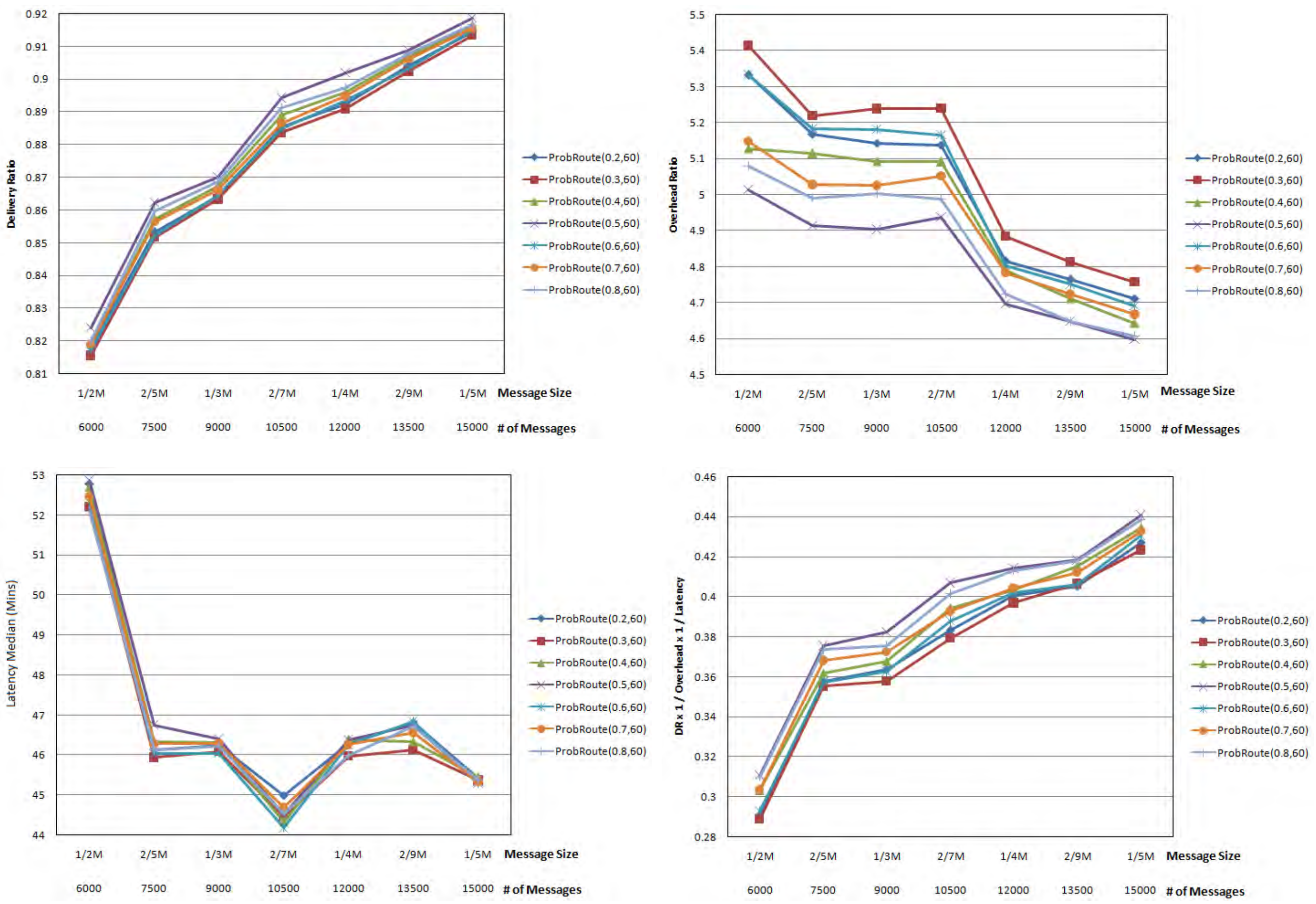


Figure 6.17: Performance of ProbRoute Protocol Varying Weight Parameter: (a) Delivery Ratio, (b) Overhead Ratio, (c) Latency Median, and (d) Composite Metric

messages faster than ORWAR. Finally, the performance in terms of composite metric also demonstrates the superiority of our ProbRoute protocol over the other protocols as shown in Figure 6.16. The results of all the protocols for both Map based movement model and RWP movement model are presented in Appendix-II.

6.4.3 ProbRoute Parameter Results

In ProbRoute we used two internal parameters, the weight parameter (α) and the interval period (T) used in the CP calculation. To determine how ProbRoute reacts to the changes in these internal parameters, we evaluate ProbRoute against itself using different values

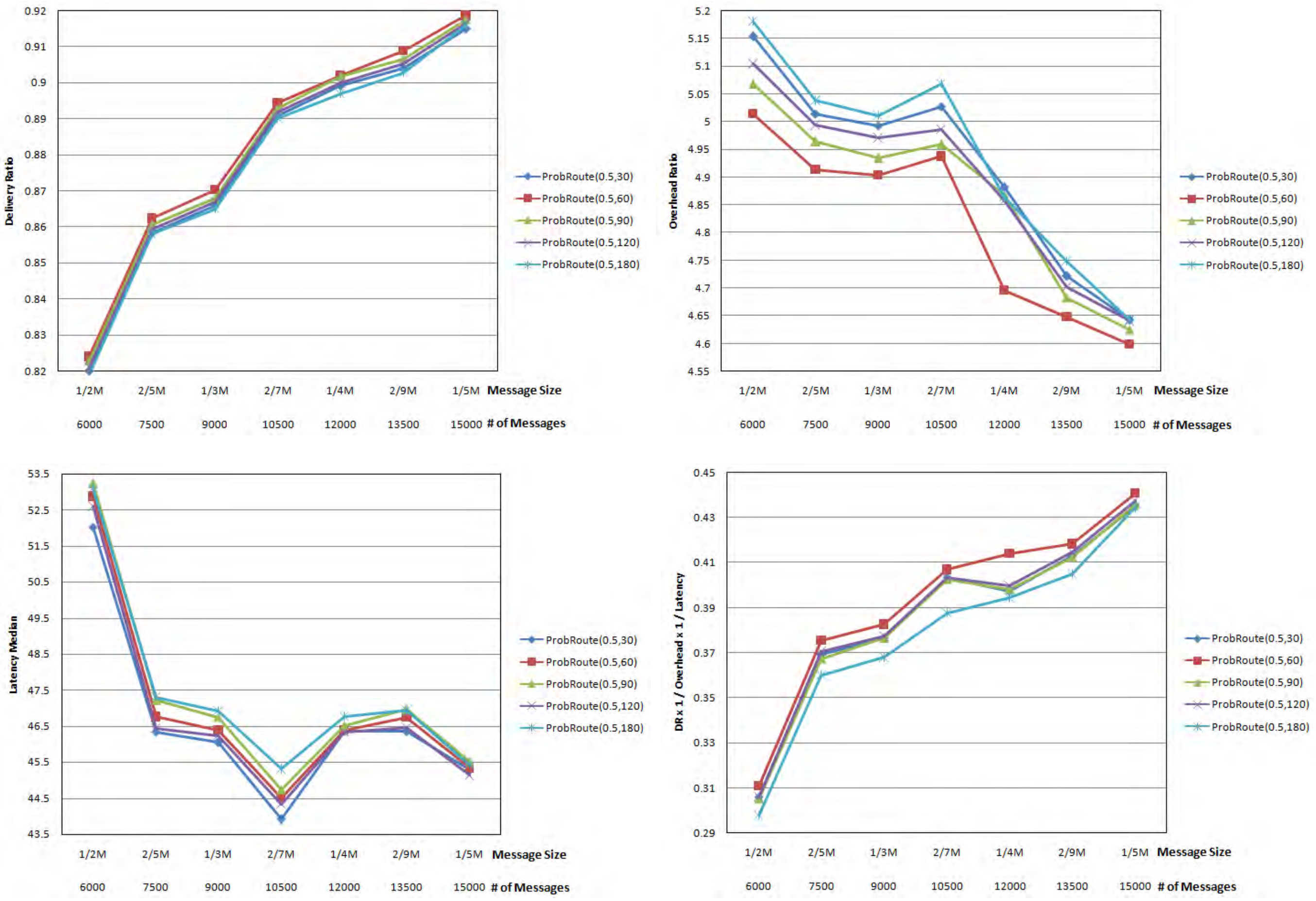


Figure 6.18: Performance of ProbRoute Protocol Varying Interval Period: (a) Delivery Ratio, (b) Overhead Ratio, (c) Latency Median, and (d) Composite Metric

of α and T. To evaluate the impact of α on ProbRoute protocol, we vary α from 0.2 to 0.8 while keeping the T fixed. On the other hand, to evaluate the impact of T we vary T between 30, 60, 90, 120, and 180 while keeping the α constant.

Figure 6.17(a) shows the delivery ratios of ProbRoute protocol varying α. Although in terms of delivery ratio α does not make a substantial difference, $\alpha = 0.5$ gives the best delivery ratio. In Figure 6.17(b) we present the overhead ratios of ProbRoute protocol varying α. The figure illustrates that even in terms of overhead $\alpha = 0.5$ yields the best result. Finally in Figure 6.17(c) we analyze the effect of α in terms of latency median. Again changing the value of α has little impact on latency median. But $\alpha = 0.3$ attains the minimum latency. So, if we do not consider latency, $\alpha = 0.5$ allows the best performance for ProbRoute. However, when latency is considered, even then making a little trade off, we see $\alpha = 0.5$ achieves the best overall composite performance. We present the composite performance in Figure 6.17(d).

Figure 6.18 demonstrates the effect of changing interval period on the performance of ProbRoute protocol. In Figure 6.18(a) the impact on delivery ratio is exhibited. In terms of delivery ratio $T = 60sec$ achieves the best result. From Figure 6.18(b), we find that $T = 60sec$ attains the minimum overhead ratio. Although in terms of latency median shown in Figure 6.18(c) $T = 60sec$ does not yield the best result, its high delivery ratio and low overhead allows the best overall performance for $T = 60sec$. For the above stated reasons we have chosen $\alpha = 0.5$ and $T = 60sec$ as default for ProbRoute throughout the entire simulation.

6.5 Comparison between TBR and ProbRoute

From the performance evaluation it is obvious that both TBR and ProbRoute outperform all the other established OpNet routing protocols. Although TBRs effective buffer management mechanism ensures its high performance, the messages in TBR are forwarded blindly. ProbRoute deals with this problem by introducing a contact probability and thus guarantee a even better delivery ratio but at the cost of slightly higher overhead. Here we present a comprehensive comparison between the protocols TBR and ProbRoute in terms of Delivery Ratio, Overhead, Latency, and composite performance.

6.5.1 Delivery Ratio

Figure 6.19 and 6.20 shows the performance of TBR and ProbRoute in terms of delivery ratios in Map based vehicular model and RWP movement model respectively. The results illustrates that both protocols achieve similar delivery ratios with ProbRoute performing slightly better. We believe the “Contact Probability” introduced in ProbRoute yields to its higher delivery ratio.

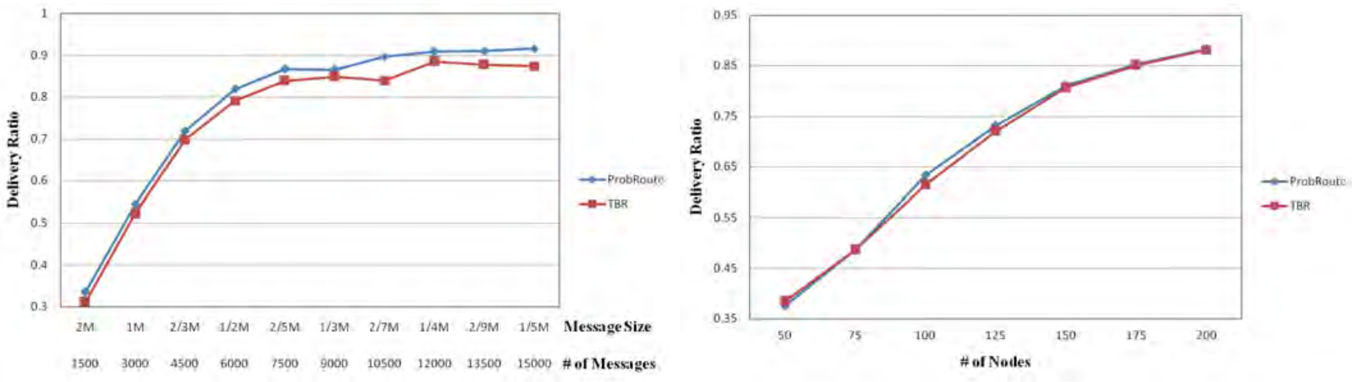


Figure 6.19: TBR vs ProbRoute (Map Vehicular Model)- Delivery Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

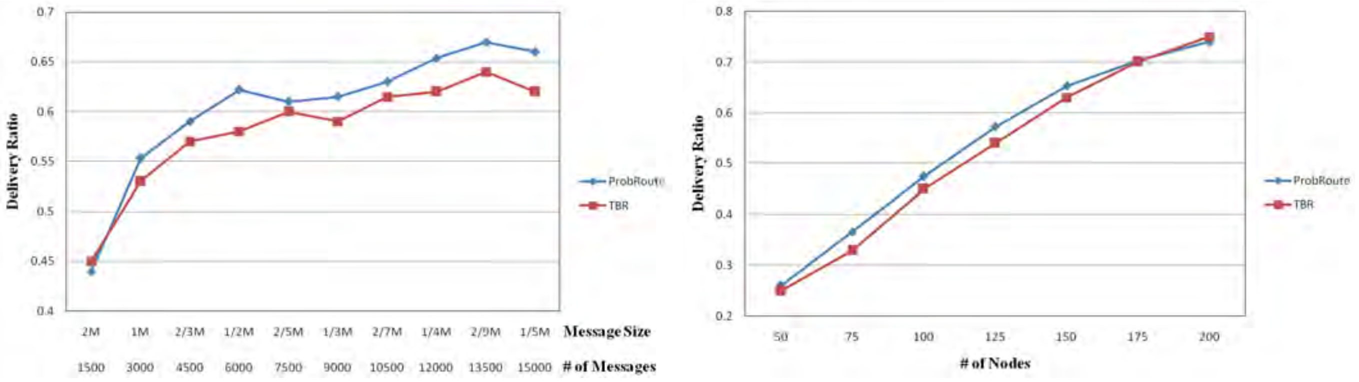


Figure 6.20: TBR vs ProbRoute (Random Way Point Movement Model)- Delivery Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

6.5.2 Overhead Ratio

Figure 6.21 and 6.22 exhibits the overhead ratios of TBR and ProbRoute protocols in Map based vehicular model and RWP movement model respectively. From the figures we can easily imply that TBR achieves lower overhead compared to that of ProbRoute. The main reason behind TBR's lower overhead is its schedule of messages to forward. While forwarding messages to the neighbor, TBR prefers the messages which have the closest deadline. On the other hand, ProbRoute always prefers the newer messages (messages with the lower L_k). This is why, ProbRoute forwards many messages which ultimately

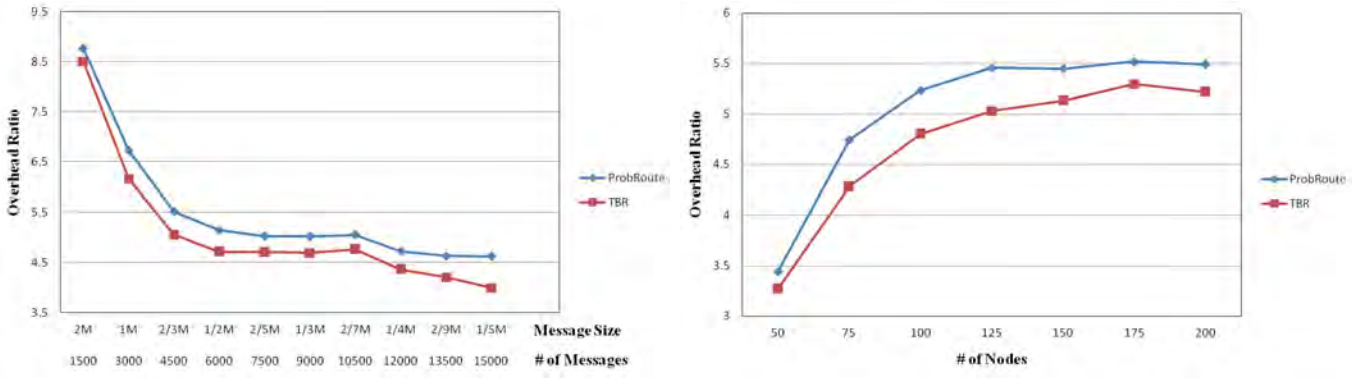


Figure 6.21: TBR vs ProbRoute (Map Based Vehicular Model)- Overhead Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

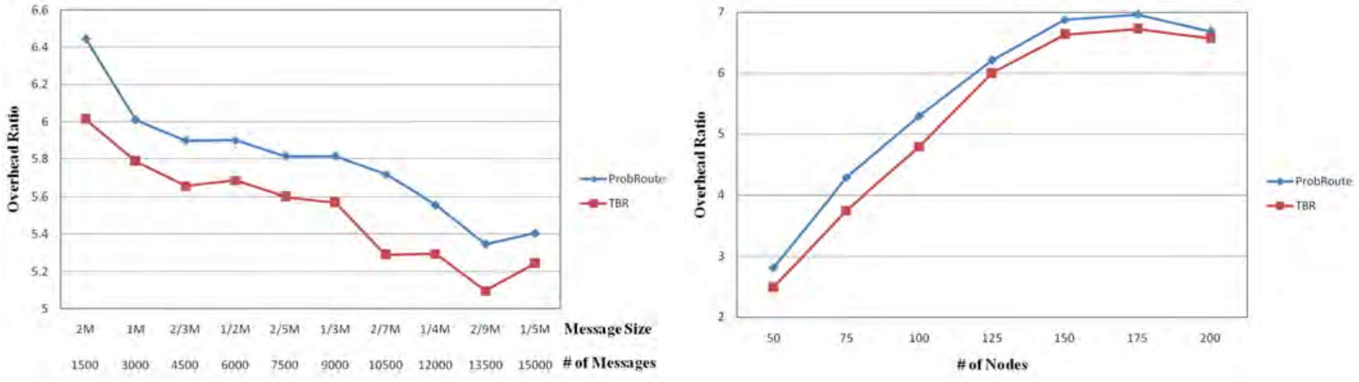


Figure 6.22: TBR vs ProbRoute (Random Way Point Movement Model)- Overhead Ratio: (a) Varying Message Size, (b) Varying Number of Nodes

fail to reach the destination as the TTL of the messages expire. TBR reduces such unsuccessful message forwarding by allowing messages with the lower TTL a chance to reach their destination.

6.5.3 Latency Median

We present a comparison of the Latencies of TBR and ProbRoute in Figure 6.23 and 6.24. In terms of latency, ProbRoute performs better than TBR as the messages experience

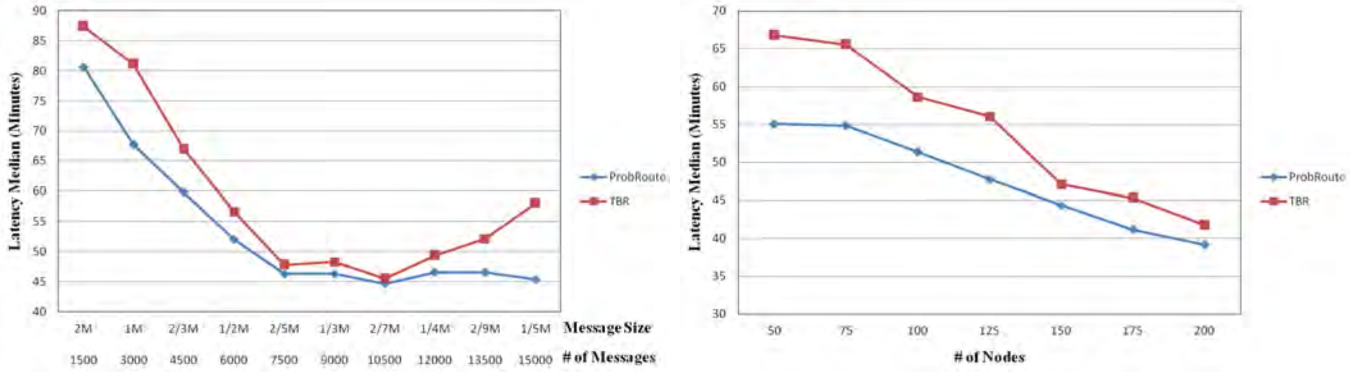


Figure 6.23: TBR vs ProbRoute (Map Based Vehicular Model)- Latency Median: (a) Varying Message Size, (b) Varying Number of Nodes

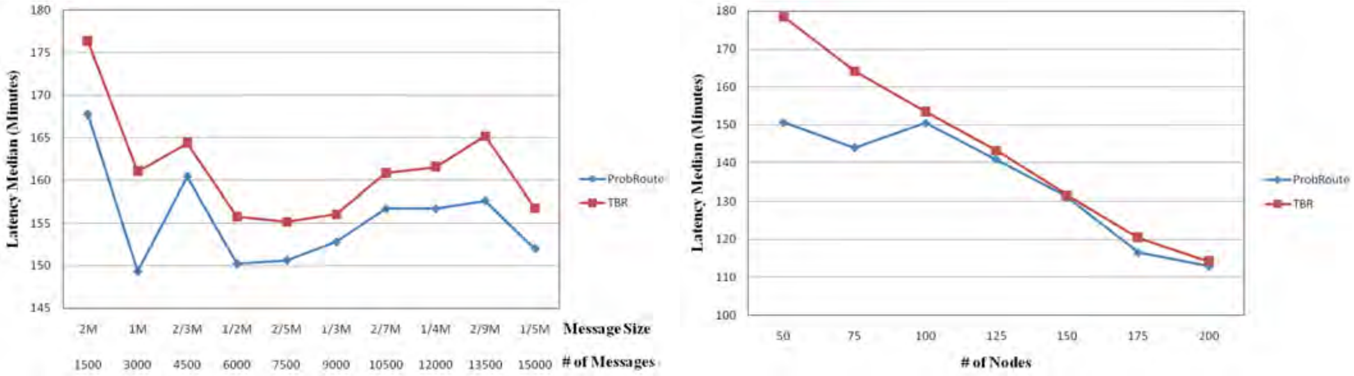


Figure 6.24: TBR vs ProbRoute (Random Way Point Movement Model)- Latency Median: (a) Varying Message Size, (b) Varying Number of Nodes

lower average delay in ProbRoute. When forwarding messages, TBR prefers a message with the closest deadline. Thus many new messages has to wait in the buffer before it gets a chance to be forwarded. As a result the average latency of the messages tend to increase. On the other hand, in ProbRoute we prefer newer messages while forwarding a message. So the newer messages does not always experience a delay before being forwarded and the overall average latency remains small.

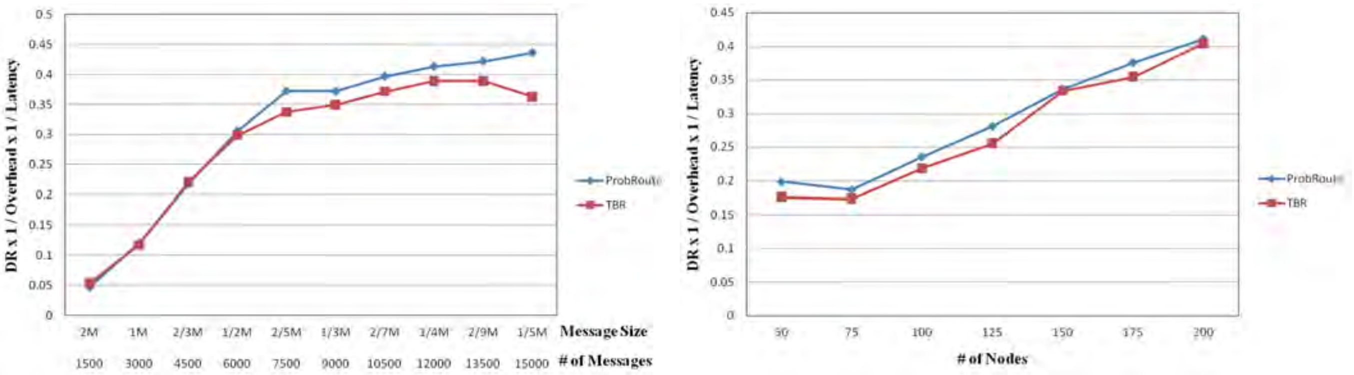


Figure 6.25: TBR vs ProbRoute (Map Based Vehicular Model)- Composite Metric: (a) Varying Message Size, (b) Varying Number of Nodes

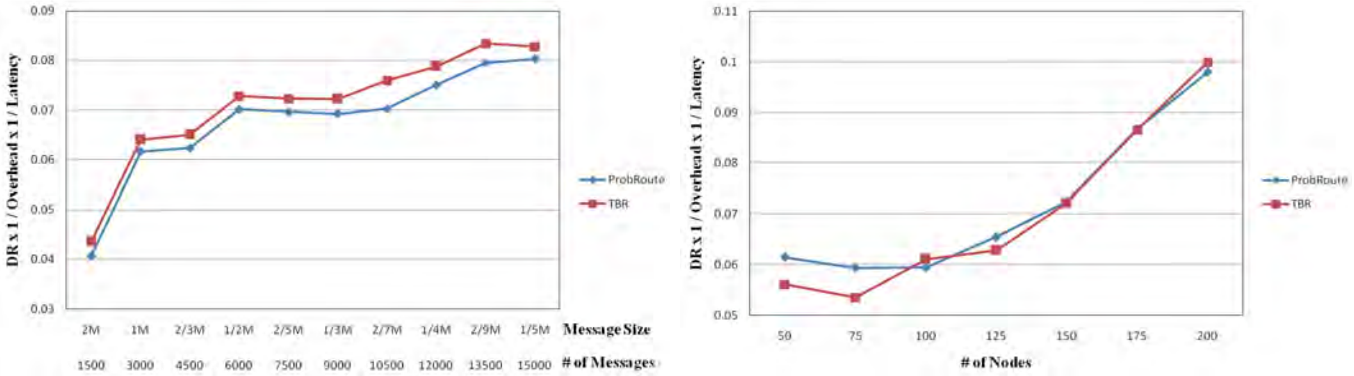


Figure 6.26: TBR vs ProbRoute (Random Way Point Movement Model)- Composite Metric: (a) Varying Message Size, (b) Varying Number of Nodes

6.5.4 Composite Metric

In terms of composite metric the comparison in performances of the protocols TBR and ProbRoute is not so straight forward. Figure 6.25 shows that, when we compare the composite metric in Map based movement model ProbRoute performs slightly better than TBR. The directed message forwarding strategy applied in ProbRoute effects this slight improvement. On the other hand, in case of the performances in RWP movement model (Figure 6.26), TBR achieves a slight edge over ProbRoute. This occurs because the directed forwarding in ProbRoute has little impact on RWP movement model.

Table 6.1: Variance Result for TBR

Variance For	Delivery Ratio	Overhead	Latency
Varying Message Size	2.12342×10^{-06}	0.005508	0.01377
Varying Network Size	1.89×10^{-06}	0.003749	0.015972

Table 6.2: Variance Result for ProbRoute

Variance For	Delivery Ratio	Overhead	Latency
Varying Message Size	5.97583×10^{-06}	0.0014	0.0564143
Varying Network Size	2.74345×10^{-06}	0.002137	0.00729471

6.6 Variance of the Results

In this section we present the variances of our two protocols. To compute the variance we have taken 4 (four) different runs for each of the iteration with the same simulation setup. Table 6.1 shows the variance of TBR protocol runs and Table 6.2 shows the variance of ProbRoute protocol runs. From the variance it is quite obvious that the performances achieved by ProbRoute and TBR are very stable. The details results of each run is shown in the Appendix-I at the end of this chapter.

Appendix-I

Variance Calculation

Table 6.3: Variance for Delivery Ratios of TBR: Varying Network Size

No. of Nodes	Run 1	Run 2	Run 3	Run 4	Average	Variance
50	0.3843	0.3826	0.3849	0.3845	0.3841	1.02917E-06
75	0.486	0.4831	0.4867	0.4869	0.4857	3.09583E-06
100	0.6145	0.6147	0.6125	0.613	0.6137	1.18917E-06
125	0.7198	0.7179	0.7176	0.7192	0.7186	1.09583E-06
150	0.8063	0.8086	0.8056	0.8043	0.8062	3.24667E-06
175	0.8508	0.8526	0.8492	0.8521	0.8512	2.30917E-06
200	0.8812	0.8801	0.8809	0.8828	0.8813	1.28333E-06

Table 6.4: Variance for Overhead Ratios of TBR: Varying Network Size

No. of Nodes	Run 1	Run 2	Run 3	Run 4	Average	Variance
50	3.2698	3.36	3.2171	3.2828	3.2824	0.003481589
75	4.2796	4.21	4.2218	4.1868	4.2246	0.001558277
100	4.7993	4.8355	4.7939	4.6735	4.7756	0.004969663
125	5.0279	4.9925	5.0869	4.9856	5.0232	0.002145343
150	5.1313	5.1691	5.0693	5.2439	5.1534	0.005332653
175	5.2961	5.2414	5.3054	5.3456	5.2971	0.001841543
200	5.2186	5.2706	5.3023	5.1125	5.226	0.006915953

Table 6.5: Variance for Latency Median of TBR: Varying Network Size

No. of Nodes	Run 1	Run 2	Run 3	Run 4	Average	Variance
50	66.765	66.4422	66.4156	66.4277	66.5126	0.028426309
75	65.56	65.4764	65.4963	65.7057	65.5596	0.010758167
100	58.6	58.7183	58.479	58.4	58.5493	0.019454689
125	56.045	55.9934	56.1894	56.0487	56.0691	0.007066516
150	47.0833	47.2479	47.1686	46.9575	47.1143	0.015448229
175	45.3083	45.1531	45.3682	45.4038	45.3084	0.012265097
200	41.74	41.7845	41.4851	41.7326	41.6856	0.01838327

Table 6.6: Variance for Delivery Ratios of TBR: Varying Message Size

Messages	Message Size	Run 1	Run 2	Run 3	Run 4	Average	Variance
1500	2M	0.3107	0.3124	0.3086	0.3105	0.31055	2.41667E-06
3000	1M	0.5217	0.5212	0.525	0.5241	0.523	0.00000338
4500	2/3M	0.6978	0.6979	0.6987	0.7006	0.69875	1.68333E-06
6000	1/2M	0.791	0.7915	0.7905	0.7938	0.7917	2.12667E-06
7500	2/5M	0.8385	0.8373	0.8389	0.8409	0.8389	2.24E-06
9000	1/3M	0.8487	0.8476	0.8501	0.8514	0.84945	2.73667E-06
10500	2/7M	0.8391	0.8411	0.8388	0.8399	0.839725	1.05583E-06
12000	1/4M	0.8845	0.8826	0.8857	0.8837	0.884125	1.70917E-06
13500	2/9M	0.8771	0.8754	0.8792	0.8782	0.877475	2.64917E-06
15000	1/5M	0.8735	0.8719	0.8745	0.8727	0.87315	1.23667E-06

Table 6.7: Variance for Overhead Ratios of TBR: Varying Message Size

Messages	Message Size	Run 1	Run 2	Run 3	Run 4	Average	Variance
1500	2M	8.5043	8.5004	8.4326	8.6298	8.516775	0.006761282
3000	1M	6.1603	6.0537	6.1075	6.0124	6.083475	0.004139162
4500	2/3M	5.0592	5.048	5.1093	5.029	5.061375	0.001176189
6000	1/2M	4.7132	4.7744	4.714	4.6471	4.712175	0.002703696
7500	2/5M	4.7044	4.9982	5.0632	4.9483	4.928525	0.024538342
9000	1/3M	4.6939	4.5911	4.7674	4.7108	4.6908	0.005405753
10500	2/7M	4.7659	4.643	4.7164	4.7704	4.723925	0.003509103
12000	1/4M	4.3618	4.2951	4.4075	4.3467	4.352775	0.002146529
13500	2/9M	4.2077	4.1829	4.1687	4.2875	4.2117	0.00281336
15000	1/5M	3.9944	3.8944	3.9338	3.9688	3.94785	0.001886703

Table 6.8: Variance for Latency Median of TBR: Varying Message Size

Messages	Message Size	Run 1	Run 2	Run 3	Run 4	Average	Variance
1500	2M	87.4017	87.5569	87.3085	87.4597	87.4317	0.01084576
3000	1M	81.1383	81.0819	81.3012	81.2817	81.200775	0.011556142
4500	2/3M	66.9533	66.9923	66.7562	66.8594	66.8903	0.01110354
6000	1/2M	56.6633	56.821	56.9781	56.8169	56.819825	0.016520329
7500	2/5M	47.805	47.827	47.6972	47.9603	47.822375	0.011671789
9000	1/3M	48.2967	48.4858	48.2532	48.3956	48.357825	0.010829069
10500	2/7M	45.535	45.398	45.4693	45.5445	45.4867	0.004614727
12000	1/4M	49.3817	49.1231	49.4053	49.1014	49.252875	0.026538496
13500	2/9M	52.1517	52.2669	52.1789	52.3748	52.243075	0.010128976
15000	1/5M	58.065	58.2798	57.9196	58.1776	58.1105	0.023892653

Table 6.9: Variance for Delivery Ratios of ProbRoute: Varying Network Size

No. of Nodes	Run 1	Run 2	Run 3	Run 4	Average	Variance
50	0.3756	0.3747	0.3721	0.3743	0.3742	2.20917E-06
75	0.4855	0.4864	0.485	0.4832	0.485	1.81583E-06
100	0.633	0.6302	0.6348	0.6338	0.633	3.90333E-06
125	0.7311	0.7307	0.7334	0.7299	0.7313	2.25583E-06
150	0.8107	0.8081	0.8129	0.8113	0.8108	3.98333E-06
175	0.8529	0.8555	0.8526	0.8532	0.8536	1.75E-06
200	0.8831	0.8838	0.8803	0.8804	0.8819	3.28667E-06

Table 6.10: Variance for Overhead Ratios of ProbRoute: Varying Network Size

No. of Nodes	Run 1	Run 2	Run 3	Run 4	Average	Variance
50	3.4401	3.5215	3.5446	3.4842	3.4976	0.002088673
75	4.7419	4.6822	4.7144	4.8211	4.7399	0.00352566
100	5.2324	5.2707	5.1516	5.2287	5.2209	0.00249187
125	5.4579	5.4896	5.4382	5.5096	5.4738	0.001017149
150	5.4468	5.5044	5.4249	5.459	5.4588	0.001124203
175	5.5178	5.4116	5.5246	5.4961	5.4875	0.002709756
200	5.4905	5.4579	5.5484	5.5481	5.5112	0.002004942

Table 6.11: Variance for Latency Median of ProbRoute: Varying Network Size

No. of Nodes	Run 1	Run 2	Run 3	Run 4	Average	Variance
50	55.03	55.1573	55.0428	55.0977	55.082	0.003385737
75	54.8417	54.9493	55.0711	54.9194	54.9454	0.009081796
100	51.3683	51.2527	51.4528	51.3881	51.3655	0.006954576
125	47.725	47.8572	47.6754	47.8665	47.781	0.009134682
150	44.3333	44.1878	44.41	44.3504	44.3204	0.008892442
175	41.1467	41.094	41.2163	41.2412	41.1746	0.00448307
200	39.1567	39.2224	39.3076	39.0831	39.1925	0.00913067

Table 6.12: Variance for Delivery Ratios of ProbRoute: Varying Message Size

Messages	Message Size	Run 1	Run 2	Run 3	Run 4	Average	Variance
1500	2M	0.3347	0.3352	0.3317	0.334	0.3339	2.39333E-06
3000	1M	0.5437	0.5457	0.5419	0.543	0.543575	2.55583E-06
4500	2/3M	0.7191	0.7124	0.7181	0.7116	0.7153	1.47933E-05
6000	1/2M	0.819	0.8186	0.8134	0.8159	0.816725	6.80917E-06
7500	2/5M	0.8665	0.8676	0.8623	0.8695	0.866475	9.2825E-06
9000	1/3M	0.8651	0.8611	0.8631	0.8671	0.8641	6.66667E-06
10500	2/7M	0.8959	0.8949	0.8952	0.8983	0.896075	2.37583E-06
12000	1/4M	0.9088	0.9036	0.905	0.9048	0.90555	5.07667E-06
13500	2/9M	0.9098	0.9088	0.9066	0.9067	0.907975	2.50917E-06
15000	1/5M	0.9155	0.9166	0.9104	0.9142	0.914175	7.29583E-06

Table 6.13: Variance for Overhead Ratios of ProbRoute: Varying Message Size

Messages	Message Size	Run 1	Run 2	Run 3	Run 4	Average	Variance
1500	2M	8.7689	8.7105	8.8053	8.7892	8.768475	0.001715629
3000	1M	6.7241	6.6891	6.7422	6.7348	6.72255	0.000552497
4500	2/3M	5.5163	5.5846	5.5667	5.5984	5.5665	0.001288433
6000	1/2M	5.1484	5.1634	5.2408	5.1806	5.1833	0.00164252
7500	2/5M	5.0276	4.9982	5.0632	4.9483	5.009325	0.002361436
9000	1/3M	5.0249	5.0872	5.0473	4.9832	5.03565	0.001886563
10500	2/7M	5.0512	5.0827	5.0633	4.9931	5.047575	0.001487236
12000	1/4M	4.7226	4.7874	4.7653	4.7712	4.761625	0.000764163
13500	2/9M	4.6326	4.6219	4.6731	4.6745	4.650525	0.000741709
15000	1/5M	4.6271	4.5918	4.6872	4.641	4.636775	0.001558962

Table 6.14: Variance for Latency Median of ProbRoute: Varying Message Size

Messages	Message Size	Run 1	Run 2	Run 3	Run 4	Average	Variance
1500	2M	80.6817	80.9503	80.4532	80.2398	80.58125	0.093091003
3000	1M	67.7383	67.545	67.1283	67.67	67.5204	0.074736047
4500	2/3M	59.84	59.98	59.562	59.8104	59.7981	0.030245907
6000	1/2M	52.06	52.7512	52.4833	52.6704	52.491225	0.095236162
7500	2/5M	46.285	46.5631	46.7723	46.198	46.4546	0.06910442
9000	1/3M	46.2933	46.5893	46.87	46.3984	46.53775	0.064073897
10500	2/7M	44.7	44.364	44.5945	44.487	44.536375	0.020767563
12000	1/4M	46.5583	46.3498	46.655	46.3984	46.490375	0.019978642
13500	2/9M	46.5583	46.893	46.694	46.954	46.774825	0.033161589
15000	1/5M	45.3717	45.564	45.0985	45.675	45.4273	0.06374766

Appendix-II

Detailed Performance Results of the Protocols

Table 6.15: Map Based Movement Model: Delivery Ratios Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	0.3742	0.3843	0.3035	0.346	0.3443	0.3178
75	0.485	0.486	0.3699	0.4482	0.4304	0.4169
100	0.633	0.6145	0.4945	0.5447	0.5464	0.5208
125	0.7313	0.7198	0.5761	0.6465	0.6376	0.6185
150	0.8108	0.8063	0.6593	0.7273	0.7228	0.6793
175	0.8536	0.8508	0.6942	0.7463	0.7731	0.7074
200	0.8819	0.8812	0.738	0.8135	0.82	0.7505

Table 6.16: Map Based Movement Model: Overhead Ratios Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	3.4976	3.2824	4.6559	4.2605	3.6953	4.5283
75	4.7399	4.2246	5.4799	7.1249	4.9696	5.5695
100	5.2209	4.7756	6.5498	10.0751	5.5196	6.3174
125	5.4738	5.0232	7.1437	15.8241	5.7088	7.0099
150	5.4588	5.1534	7.0787	22.864	5.6868	6.9985
175	5.4875	5.2971	7.0063	30.7952	5.6955	6.9392
200	5.5112	5.226	7.0928	35.5378	5.576	6.5989

Table 6.17: Map Based Movement Model: Latency Median Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	55.082	66.5126	40.475	71.3667	59.8717	41.0333
75	54.9454	65.5596	43.585	75.6517	56.125	44.1467
100	51.3655	58.5493	42.6367	72.33	49.4433	43.485
125	47.781	56.0691	37.7517	63.1117	46.375	39.0233
150	44.3204	47.1143	33.7317	55.3133	42.035	35.9367
175	41.1746	45.3084	32.715	52.3433	40.92	33.3183
200	39.1925	41.6856	32.4517	48.04	39.31	33.4583

Table 6.18: Map Based Movement Model: Composite Metric Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	0.194233	0.176025	0.161053	0.113794	0.15562	0.171034
75	0.186226	0.175475	0.154873	0.083152	0.15431	0.169558
100	0.236041	0.219772	0.177074	0.074746	0.200215	0.18958
125	0.279609	0.255569	0.213618	0.064735	0.240835	0.226102
150	0.33513	0.332086	0.276116	0.057508	0.30237	0.270096
175	0.37779	0.354495	0.302865	0.046299	0.331717	0.305966
200	0.408291	0.4045	0.320628	0.04765	0.3741	0.339919

Table 6.19: Map Based Movement Model: Delivery Ratios Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	0.3339	0.31055	0.2673	0.302	0.286	0.2767
3000	1M	0.543575	0.523	0.41	0.4963	0.4663	0.477
4500	2/3M	0.7153	0.69875	0.5453	0.6424	0.624	0.6016
6000	1/2M	0.816725	0.7917	0.637	0.779	0.7158	0.7183
7500	2/5M	0.866475	0.8389	0.6595	0.7932	0.7487	0.75
9000	1/3M	0.8641	0.84945	0.6792	0.8366	0.7627	0.7661
10500	2/7M	0.896075	0.839725	0.6884	0.8231	0.7732	0.7751
12000	1/4M	0.90555	0.884125	0.6953	0.8768	0.7802	0.7804
13500	2/9M	0.907975	0.877475	0.6837	0.8932	0.7738	0.7884
15000	1/5M	0.914175	0.87315	0.6429	0.8863	0.7412	0.7547

Table 6.20: Map Based Movement Model: Overhead Ratios Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	8.768475	8.516775	15.0948	29.0839	8.8741	10.5327
3000	1M	6.72255	6.083475	11.8163	21.3593	6.9264	8.2935
4500	2/3M	5.5665	5.061375	9.9772	18.3438	5.8437	7.2095
6000	1/2M	5.1833	4.712175	9.1672	15.7142	5.3995	6.2782
7500	2/5M	5.009325	4.928525	8.9307	15.6295	5.2606	6.1518
9000	1/3M	5.03565	4.6908	8.9014	14.7692	5.2595	6.0345
10500	2/7M	5.047575	4.723925	8.7803	14.8917	5.2748	5.9646
12000	1/4M	4.761625	4.352775	8.4971	14.4285	5.0691	5.8076
13500	2/9M	4.650525	4.2117	8.1286	13.5717	4.8807	5.6479
15000	1/5M	4.636775	3.94785	7.8847	13.9382	4.8886	5.691

Table 6.21: Map Based Movement Model: Latency Median Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	80.58125	87.4317	58.8317	91.2333	94.515	63.3417
3000	1M	67.5204	81.200775	56.8967	96.505	68.1367	57.7233
4500	2/3M	59.7981	66.8903	48.895	79.62	58.88	48.47
6000	1/2M	52.491225	56.819825	45.275	66.7867	50.1917	45.1333
7500	2/5M	46.4546	47.822375	41.5633	57.1167	44.735	42.04
9000	1/3M	46.53775	48.357825	41.4783	56.0717	44.3083	40.8667
10500	2/7M	44.536375	45.4867	38.8567	52.4967	41.565	37.6367
12000	1/4M	46.490375	49.252875	40.9783	59.0883	43.1117	40.195
13500	2/9M	46.774825	52.243075	41.785	61.5033	45.3783	41.8983
15000	1/5M	45.4273	58.1105	44.38	65.105	48.5717	42.1017

Table 6.22: Map Based Movement Model: Composite Metric Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	0.047256	0.041705	0.0301	0.011382	0.034099	0.041474
3000	1M	0.119754	0.105874	0.060984	0.024077	0.098805	0.099639
4500	2/3M	0.214891	0.206391	0.11178	0.043984	0.181355	0.172159
6000	1/2M	0.300181	0.295692	0.153477	0.074226	0.264123	0.253497
7500	2/5M	0.372347	0.355928	0.177672	0.088853	0.318145	0.289999
9000	1/3M	0.368725	0.374476	0.183958	0.101022	0.327284	0.310652
10500	2/7M	0.398609	0.390796	0.201774	0.105287	0.352662	0.345275
12000	1/4M	0.409067	0.412397	0.199686	0.102844	0.35701	0.334309
13500	2/9M	0.417407	0.398794	0.201293	0.107008	0.34938	0.333168
15000	1/5M	0.434007	0.380604	0.183726	0.09767	0.312153	0.314982

Table 6.23: RWP Movement Model: Delivery Ratios Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	0.2609	0.25	0.2052	0.2037	0.23	0.2062
75	0.3667	0.33	0.2728	0.2901	0.31	0.2709
100	0.4746	0.45	0.3502	0.3894	0.41	0.3512
125	0.5727	0.54	0.4417	0.4721	0.5	0.4357
150	0.653	0.63	0.5018	0.5337	0.58	0.4947
175	0.7031	0.7012	0.5517	0.6111	0.66	0.5454
200	0.74	0.75	0.6074	0.683	0.71	0.5945

Table 6.24: RWP Movement Model: Overhead Ratios Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	2.8137	2.4955	3.6803	1.8208	2.6364	4.2881
75	4.2874	3.7551	5.5268	3.2692	3.9994	5.9919
100	5.2991	4.796	7.0405	5.0157	5.3971	7.2862
125	6.209	5.9978	8.525	6.8964	6.5324	8.9031
150	6.8747	6.6313	9.1903	8.5589	7.0852	9.5426
175	6.9612	6.7263	8.8708	10.648	7.0072	9.3645
200	6.6855	6.5737	8.1264	12.5195	6.7502	8.7013

Table 6.25: RWP Movement Model: Latency Median Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	150.6517	178.5067	130.545	169.1817	184.1117	127.5983
75	144.0083	164.165	144.045	171.49	167.3633	139.235
100	150.515	153.4217	141.2583	175.6383	159.4333	140.81
125	140.8	143.245	123.3533	170.2517	141.1767	126.7817
150	130.9967	131.5367	107.105	160.4267	130.45	115.9933
175	116.545	120.4033	96.6033	148.1283	118.1417	101.915
200	112.965	114.1583	93.1933	138.4417	114.5717	95.155

Table 6.26: RWP Movement Model: Composite Metric Varying Network Size

No. of Nodes	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
50	0.061549	0.056121	0.04271	0.066126	0.047384	0.037686
75	0.059392	0.053532	0.034267	0.051745	0.046313	0.032471
100	0.059504	0.061157	0.035213	0.044202	0.047648	0.034231
125	0.065509	0.062852	0.042003	0.040209	0.054217	0.0386
150	0.07251	0.072226	0.050979	0.038869	0.062753	0.044693
175	0.086664	0.086582	0.06438	0.038744	0.079725	0.057147
200	0.097984	0.099941	0.080203	0.039406	0.091805	0.071802

Table 6.27: RWP Movement Model: Delivery Ratios Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	0.44	0.45	0.3913	0.3887	0.4	0.3693
3000	1M	0.5533	0.53	0.45	0.453	0.49	0.43
4500	2/3M	0.59	0.57	0.4567	0.4922	0.53	0.4516
6000	1/2M	0.6218	0.58	0.515	0.5	0.54	0.5
7500	2/5M	0.61	0.6	0.4935	0.5131	0.55	0.4799
9000	1/3M	0.615	0.59	0.53	0.51	0.55	0.52
10500	2/7M	0.63	0.6147	0.5201	0.535	0.56	0.5098
12000	1/4M	0.6538	0.62	0.53	0.54	0.57	0.52
13500	2/9M	0.67	0.64	0.5599	0.5502	0.58	0.5462
15000	1/5M	0.66	0.62	0.55	0.53	0.57	0.54

Table 6.28: RWP Movement Model: Overhead Ratios Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	6.4452	6.0135	7.5026	6.8491	6.5742	8.1155
3000	1M	6.0102	5.7875	7.7193	7.1038	6.2048	8.382
4500	2/3M	5.8986	5.6558	8.2793	6.7201	5.9719	8.7451
6000	1/2M	5.901	5.6862	7.7508	6.6919	6.0605	8.2801
7500	2/5M	5.8154	5.5991	8.0046	6.6635	5.973	8.571
9000	1/3M	5.8156	5.5693	7.9968	6.3947	5.8926	8.4113
10500	2/7M	5.7189	5.2904	8.0165	6.17	5.73	8.5481
12000	1/4M	5.5574	5.2949	7.6226	6.2115	5.7758	8.2395
13500	2/9M	5.3476	5.096	7.6809	5.9198	5.6087	8.2802
15000	1/5M	5.4068	5.2437	7.7127	6.1249	5.7205	8.2953

Table 6.29: RWP Movement Model: Latency Median Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	167.7233	176.3217	172.3	184.3183	168.0483	164.9017
3000	1M	149.3533	161.0017	143.5683	175.23	160.9317	147.29
4500	2/3M	160.415	164.3167	140.0933	178.3217	164.5083	145.1833
6000	1/2M	150.21	155.73	135.285	177.535	152.0733	134.0917
7500	2/5M	150.64	155.1667	141.3017	174.8883	155.21	141.1317
9000	1/3M	152.8067	156.0467	132.9467	173.9167	158.965	134.7917
10500	2/7M	156.695	160.8133	138.2017	177.635	159.6283	141.7783
12000	1/4M	156.695	161.5083	140.1617	179.3783	158.1517	141.8617
13500	2/9M	157.6333	165.1817	142.2267	178.8517	163.9967	141.7267
15000	1/5M	152.0283	156.785	132.9283	172.405	154.1983	134.855

Table 6.30: RWP Movement Model: Composite Metric Varying Message Size

Messages	Message Size	ProbRoute	TBR	SNW	MaxProp	ORWAR	EBR
1500	2M	0.040703	0.04244	0.03027	0.03079	0.036206	0.027596
3000	1M	0.061639	0.056879	0.040605	0.036391	0.049071	0.03483
4500	2/3M	0.062353	0.061334	0.039375	0.041073	0.053948	0.035569
6000	1/2M	0.07015	0.065499	0.049115	0.042086	0.058591	0.045033
7500	2/5M	0.069632	0.069061	0.043631	0.044029	0.059327	0.039673
9000	1/3M	0.069205	0.067889	0.049852	0.045857	0.058716	0.045865
10500	2/7M	0.070303	0.072252	0.046945	0.048814	0.061224	0.042065
12000	1/4M	0.075079	0.0725	0.049607	0.048465	0.062401	0.044487
13500	2/9M	0.079482	0.076031	0.051253	0.051966	0.063057	0.046544
15000	1/5M	0.080293	0.075414	0.053646	0.050191	0.064619	0.048272

Chapter 7

Conclusion and Future Direction

Most of the earlier ad hoc routing protocols have assumed an end-to-end connectivity between the source and the destination nodes. This is not true in many applications of wireless ad hoc networks such as OppNets. To deal with unpredictable connectivity, many routing protocols utilize message flooding to improve the chance of message delivery. However, this approach comes at the expense of higher network resource consumption mainly- bandwidth, battery power and storage. Quota based routing protocols ensure efficient use of network resources but sacrifices delivery ratio. In this thesis, we proposed two novel quota based routing protocols- TBR and ProbRoute. Both protocols outperform many popular flooding and quota based approaches.

TBR introduces an efficient buffer management strategy in OpNets that offers higher priority to the messages with the closest deadline. TBR also prioritizes messages to be dropped from the buffer. The two fold buffer management mechanism of TBR prefers the messages with the closest deadline to be forwarded first and the messages with the minimum number of copies left to spray to be deleted first. The introduction of TTL in the buffer management scheme reduces unsuccessful message forwarding.

ProbRoute adds a new dimension in quota based approach by introducing a probabilistic metric, “Contact Probability”. The contact probability enables ProbRoute to achieve guided forwarding. ProbRoute is the first quota based protocol to achieve such feature. ProbRoute also introduces an adaptive *message priority* to forward the messages buffered in a node. This message priority ensures a fair message dissemination scheme which contributes to ProbRoutes very high delivery ratio.

In our simulation, we demonstrate the superiority of TBR and ProbRoute by comparing their performance with that of many popular OpNet routing protocols such as-MaxProp, Spray And Wait, ORWAR and EBR. TBR achieves more than 10%-15% higher delivery ratio that that of any other quota based protocol with 10%-15% less overhead. On the other hand, ProbRoute achieves 15% - 20% higher delivery ratio than that of the existing quota based alternative while maintaining 5%-7% less overhead. Both TBR and ProbRoute achieve better delivery ratio than that of the best performing flooding based protocol MaxProp while incurring 15%-20% less delay and 70%-80% less overhead. We have also presented a comprehensive comparison between the two proposed protocols illustrating their relative advantages and drawbacks.

Our work can be extended in various directions. An obvious extension of the work could be the evaluation of our approach on an real time network with physical nodes. Specially the effect of our protocol in real life DTN test beds such as *UMassDieselNet* can easily be the next step for us. Moreover, studying the protocol's performance in other DTN scenarios, e.g. in a disaster scenario, is also an interesting direction to pursue. Another extension of our work can be done by introducing the Type of Service in our protocols. In our work we have not considered the type of messages and considered all the messages have same quality of service requirement. But this will impede the voice, video or other real time traffic service over OpNet. So, introducing a message priority based on type of service and tuning our priorities to adapt this types may increase the usability of our protocols and improve overall performance of the network. . Finally, adapting our protocol to achieve a congestion less network can be another direction of our future work.

Bibliography

[1] Delay tolerant networking research group. http://www.dtnrg.org. Available at HTTP: November 2010.

[2] Disruption tolerant networking. http://www.darpa.mil/ato/solicit/DTN/. Available at HTTP: November 2010.

[3] One simulator. http://www.netlab.tkk.fi/tutkimus/dtn/theone/. Available at HTTP: November 2010.

[4] A. Balasubramanian, B. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.*, 37(4):373–384, 2007.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *IEEE INFOCOM*, pages 1– 11, 2006.

[6] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, 2003.

[7] B. Burns, O. Brock, and B. N. Neil Levine. Mv routing and capacity building in disruption tolerant networks. In *INFOCOM*, pages 398–408, 2005.

[8] 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. Technical Report UCAM-CL-TR-617, Computer Laboratory, University of Cambridge, February 2005.

[9] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *IEEE Infocom*, pages 1–13, 2006.

[10] D. J. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *In ACM MobiCom*, pages 134–146, 2003.

[11] J. H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. In *IEEE Network, Special Issue on Wireless Sensor Networking*, pages 12–18, May 2006.

[12] K. Fall. A delay-tolerant network architecture for challenged internets. In *ACM SIGCOMM*, pages 27–34, 2003.

[13] S. Farrell and V. Cahill. *Delay and Disruption Tolerant Networking.* Artech House, 2006.

[14] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald. When tcp breaks: Delay- and disruption- tolerant networking. *IEEE Internet Computing*, 10(4):72–78, 2006.

[15] D. Henriksson, T. F. Abdelzaher, and R. K. Ganti. A caching-based approach to routing in delay-tolerant networks. In *ICCCN*, pages 69–74, 2007.

[16] J. H. Huang, S. Amjad, and S. M. Cenwits. A sensor based loosely coupled search and rescue system using witnesses. In *ACM SenSys*, pages 180–191, 2005.

[17] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *ACM SIGCOMM Workshop on DTN*, pages 244–251, 2005.

[18] D. B. Johnson and D. B. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, pages 153–181, 1996.

[19] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS-X: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 96–107, 2002.

[20] A. Keränen. Opportunistic network environment simulator. Technical report, Department of Communications and Networking, Helsinki University of Technology, Helsinki, Finland, May 2008.

[21] A. Keränen, J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. In *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pages 1–10, 2009.

[22] J. LeBrun, C. N. Chuah, and D.Ghosal. Knowledge based opportunistic forwarding in vehicular wireless ad hoc networks. In *IEEE VTC Spring*, pages 2289–2293, 2005.

[23] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[24] S. C. Nelson, M. Bakht, and R. Kravets. Encounter-based routing in dtns. In *INFOCOM*, pages 846–854, 2009.

[25] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pages 90–100, 1999.

[26] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan. Prioritized epidemic routing for opportunistic networks. In *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 62–66, 2007.

[27] G. Sandulescu and S. Nadjm-Tehrani. Opportunistic dtn routing with window-aware adaptive replication. In *AINTEC '08: Proceedings of the 4th Asian Conference on Internet Engineering*, pages 103–112, 2008.

[28] K. Scott and S. Burleigh. *Bundle Protocol Specification*, November 2007.

[29] D. Snowdon, N. Glance, and J. L. P. Meunier. Using people as a communication medium. *Elsevier Computer Networks*, 35(4):429–442, Feburuary 2001.

[30] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *IEEE SECON*, pages 235–244, 2004.

[31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, 2005.

[32] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 79–85, 2007.

[33] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-200006, Duke University, 2000.

[34] R. Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy. Turning the postal system into a generic digital communication mechanism. In *ACM SIGCOMM*, pages 159–166, 2004.

[35] F. Warthman. *Delay-Tolerant Networks (DTNs): A Tutorial.* http://www.dtnrg.org/wiki/Docs/. Available at HTTP: November 2010.

[36] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys and Tutorials*, 8(1-4):24–37, 2006.