2014

Data management in delay tolerant networks

Saeid Iranmanesh
University of Wollongong

Data Management in Delay Tolerant Networks

A thesis submitted in fulfilment of the
requirements for the award of the degree

Doctor of Philosophy

From

University of Wollongong

By

Saeid Iranmanesh

School of Electrical, Computer and
Telecommunications Engineering

August 2014

To my parents

Abstract

Delay Tolerant Networks (DTNs) are characterized by the lack of contemporaneous paths between any source and destination node. In these networks, nodes act as relays, whereby they cooperatively help forward data bundles from a source to a destination node. As a basic forwarding strategy, nodes may flood bundles to every encountered node. However, flooding results in congestion and unnecessarily consume precious resources such as buffer space and bandwidth. To this end, many routing protocols select a next hop node based on metrics such as delivery probability and encounter rates. Another strategy is one adopted by quota based protocols in order to reduce resource usage. Namely, for each bundle, only a limited number of copies or replicas are disseminated throughout the network. However, they suffer from low delivery ratios as their dissemination rate is low. Hence, bundles need to be efficiently managed in order to achieve high delivery ratios, low delays and low overheads. Another key challenge is considering both routing and buffer management simultaneously when network resources such as bandwidth and buffer are limited and the number of replicas for each bundle is finite. Under such conditions, sender nodes need to select a next hop node that results in a high delivery ratio. In addition, as nodes may need to send a large number of bundles in each contact, their communication bandwidth may not be sufficient to transmit all buffered bundles. In addition, due to limited buffer size, when replicas are dropped by nodes when their buffer overflows, the delivery probability of the corresponding bundles reduces. This is because no provisions are provided to replace a dropped replica in order to maintain a high delivery ratio.

This thesis proposes a quota-based protocol that is based on weighting nodes that have encountered the final destination higher than any other nodes. This fact is based on the idea that regardless of how small an encounter rate with the destination is, given a highly correlated movement model, e.g., human, we will end up with a high delivery ratio. This idea is then studied analytically using a time homogeneous semi-Markov process (THSMP). Analysis shows that a targeted forwarding strategy based

on contact history with a destination improves bundle delivery when there are finite replicas. A destination-based routing protocol (DBRP) is then proposed to specifically target nodes that have a history with a bundle's destination. Simulation studies over three scenarios show that in terms of a composite metric comprising of delivery, delay and overhead, DBRP achieves up to 57% improvement over three well-known routing protocols, namely PROPHET, EBR and Spray and Wait. Moreover, DBRP results in nodes experiencing at least 28% lower buffer consumption.

The second proposed method investigated in this thesis is an efficient scheduling and drop policy called QM-EBRP for use under *quota* based protocols. In particular, QM-EBRP makes use of the encounter rate of vehicles and context information such as time to live, number of available replicas and maximum number of forwarded bundle replicas to derive a bundle's priority. Simulation results, over a service quality metric comprising of delivery, delay and overhead, show that the proposed policy achieves up to 80% improvement when vehicles have infinite buffer space and up to 35% when vehicles have finite buffer space over six popular queuing policies: Drop Oldest (DO), Last Input First Output (LIFO), First Input First Output (FIFO), Most FOrwarded first (MOFO), LEast PRobable first (LEPR), and drop bundles with greatest hop count (HOP-COUNT).

Lastly, this thesis considers a Mobility-Based Routing Protocol (MBRP) that constructs a space-time graph at every node by recording the mobility pattern of nodes upon each contact. In particular, nodes do not have full knowledge of the network topology. Also, the space-time graph is dynamic, meaning the trajectory of nodes may only be valid for a given period of time. As the space-time graph may be incomplete, MBRP presents a heuristic that evaluates encountered nodes based on their recorded mobility patterns in order to disseminate a finite number of replicas. MBRP has been evaluated over a realistic environment comprising of vehicles with both periodic and dynamic mobility patterns. The simulation results, over a service quality metric comprising of delivery, delay and overhead, show that MBRP achieves up to 105% improvement as compared to four well-known routing protocols namely, EBR, EPIDEMIC, MAXPROP, and PROPHET. Finally, MBRP is capable of achieving 50% of the performance attained by the optimal algorithm, whereby all nodes are preloaded with the space-time graph.

Statement of Originality

I, Saeid Iranmanesh, declare that this thesis, submitted in fulfilment of the requirements for the award of the degree Doctor of Philosophy in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work, except where due reference is made in the text. No work in this thesis has been submitted for a degree at another university or institution.

Signed

Saeid Iranmanesh

August 2014

Acknowledgements

This thesis is the culmination of my academic journey in Computer/Electrical Engineering. Honestly, this would not have been possible without the support of many people.

First of all, it is my great fortune to have pursued my PhD studies under the supervision of such great academic staff Dr. Raad Raad and A/Prof Kwan-Wu Chin who guided me with their profound knowledge, insight and wisdom. I would like to thank Dr Raad Raad for his continuous support, encouragement and invaluable concern throughout this research, especially, his pleasant and friendly personality and precise guidance made this graduate study more enjoyable. I would like also express my appreciation to A/Prof Kwan-Wu Chin for his guidance, patience and support throughout this research and the writing of this thesis.

I also would like to thank Mr. Sasha Nikolic for offering me teaching positions. I also would like to thank my colleague, Dr Hamid Salarian, for his friendship and sharing life with me. Finally, my deep gratitude goes to my parents. I am very grateful to them for standing by me in everything I have done and giving me whatever they can. They have always provided me continuous support, encouragement and their love. I dedicate this thesis to them.

Table of Content

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Delay Tolerant Networks

Delay Tolerant Networks (DTNs) [1] can be viewed as a unconnected graph where there is no direct path between a source and destination node. In other words, DTNs are characterized by frequent disconnections, large delays and may have no contemporaneous paths. The intermittent connectivity experienced by nodes is due to mobility, power management, node density, and limited radio range. Apart from that, nodes may also have resource constraints, such as finite buffer space and low transmission rates or limited bandwidth [2]. Figure 1.1 shows an example DTN comprising of vehicles and pedestrians. All vehicles, e.g., buses and cars, and pedestrians are equipped with a radio transceiver that allows them to communicate with each other. All mobile nodes will help each other forward messages. Consider User-A who wants to send a message to one or more students attending the School of Electrical, Computer and Telecommunication Engineering (SECTE). Also shown is a possible path via a number of people and cars. Inevitably, the topology or path taken changes over time and prediction of contacts is challenging. This is due to the following three factors. Firstly, the time between contacts may be large. In particular, the study reported in [3] on the attendance of students at the University of Cambridge shows that students are not always connected. For example, students may meet each other during classes, and do not meet between classes. Secondly, the duration of contacts is likely to be random. Close friends may remain in contact between classes, but otherwise, contacts are mainly opportunistic or by coincidence. Thirdly, users may move under a mobility model that coincides with contact times, e.g., lectures, and take popular paths to lecture rooms. In general, nodes have different types of contacts based on their mobility model. In particular, the contacts can be one of the following:

1. *Permanent*: two nodes may have a persistent network connection, e.g., a node connecting through a Digital Subscriber Loop (DSL) connection.
2. *On-demand*: nodes establish connections on a demand basis; e.g., a dial-up connection.
3. *Scheduled*: these contacts are determined in advance and are governed by predetermined mobility patterns; e.g., orbiting satellites.
4. *Opportunistic*: contacts are random, and hence, not predictable.
5. *Predictable*: a hybrid between scheduled and opportunistic contacts where future contacts are predictable or at least semi-predictable based on a node's movement pattern [4] or its history.

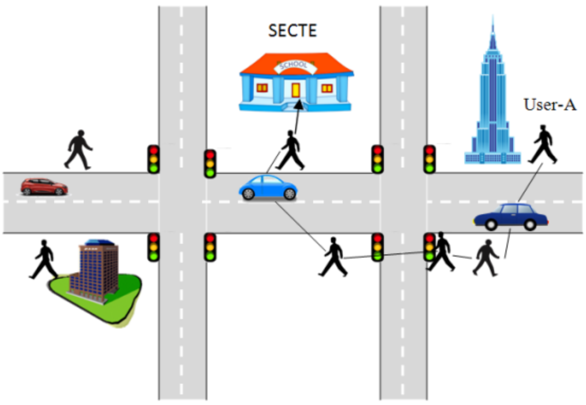


Figure 1-1 An example DTN formed by vehicles and people.

DTNs have many potential applications. For example, the Inter-Planetary Networks (IPNs) [2] is a DTN comprising of robotic spacecrafts and planet orbiting vehicles. Notably, in November 2008, NASA’s Jet Propulsion Laboratory used a DTN to transmit images through the EPOXI spacecraft located about 20 million miles from Earth. Another DTN application is providing data communications to/from rural areas [5, 6]. The Wizzy Digital Courier service provides off-line Internet access to schools in remote villages of South Africa [7]. Internet access is enabled by a person on a motorbike, with a USB storage device of 128MB space and may also be equipped with an IEEE 802.11b access point that allows the courier to collect data from a village before he/she travels to a city with Internet connectivity. A DTN may consist of students on a college campus [8], or buses [9], or a wireless sensor network with mobile nodes used to collect sensed data [10, 11]. In [9], 30 buses move along predefined paths in a 388 km^2 area. Each bus generates between two and 18 bundles per hour and is capable of storage between 50 and 148000 bundles that are 10 KB in size with an average transfer rate of 120 KB/s. The data mules in [10] move randomly and collect data from sensors and forward them to access points. Data mules are independent from each other and do not exchange any data among themselves. The key characteristics of a data mule are large storage capacity, renewable power, and the ability to communicate with sensors and networked access points.

The characteristics of DTNs pose significant challenges and problems to conventional ad-hoc routing protocols. Well-known routing protocols such as Ad hoc On-Demand Distance Vector (AODV) [12] and Dynamic Source Routing (DSR) [13], Destination-Sequenced Distance-Vector (DSDV), Location Aided Routing (LAR) [14], Exponential Age SEarch (EASE) [15, 16], On-Demand Multicast Routing Protocol (ODMRP) [17] fail to operate properly in DTNs [18]. As an example, consider using Destination-Sequenced Distance-Vector (DSDV) in a DTN comprising of nodes that correspond to pedestrians, trams or cars. A key assumption of DSDV is that nodes are able to pro-actively learn the topology by flooding link state bundles throughout the network. Unfortunately, the significant delays between node contacts make it impossible for nodes to learn the topology of a DTN. A similar problem arises with reactive routing protocols such as AODV [12] because their route establishment process will likely fail to find a complete route. Apart from that, these protocols assume transmission times that are in the order of seconds as opposed to days or months. This also means any retransmissions will cause unnecessary traffic as packets may not reach their respective destination when nodes experience timeouts. Note, in this thesis, the terms bundle, message and packet are used interchangeably.

To this end, routing protocols developed for DTNs use a store-carry-forward model. That is, when a node receives a message but if there is no path to the destination or even a connection to any other nodes, the message is buffered awaiting future contact opportunities. More details concerning these protocols/policies are elaborated in Chapter 2. In general, DTN routing strategies need to overcome the following main challenges. Firstly, nodes may lack future contacts information. As a result, their forwarding strategy may be sub-optimal. In this case, DTN routing protocols have to rely on local information such as history of encounters in order to predict future contacts. However, nodes may move under various mobility patterns [19]. Consequently, the movement of nodes is unpredictable/semi predictable/predictable and deterministic. For example, nodes may be buses or trams that have a predetermined path and scheduled contacts. In this case, it is possible to predict future contact opportunities but due to delays between contacts, contacts information may not be available at sender nodes. This may result in protocols with low delivery ratios. In another example, nodes may be animals. In this case, the network topology is unpredictable. In addition, nodes are not able to learn the network topology due to large delays and highly dynamic node movements. Secondly, as mentioned, nodes may have limited network resources such as battery, buffer and bandwidth. For example, mobile phones have limited memory, radio range and battery. In this case, a resource friendly routing protocol is required. For example, soldiers on a battle field may not have access to a power supply for hours to charge their cell phone. Accordingly, soldiers have to manage their phone's battery usage efficiently. In addition, people in high density areas may experience congestion, which require them to drop messages. Another critical consideration is that bandwidth may be limited or the duration of contacts may not be sufficiently long for people to exchange all their bundles. Note that the duration of contacts is affected by the speed of nodes. For example, in a study on vehicular networks [20], the authors show that the duration of contacts between cars using IEEE 802.11g crossing at 20 Km/h is about 40 seconds, at 40 Km/h it is about 15 seconds and at 60 Km/h it is about 11 seconds.

1.2 Research Problems

Given the above challenging issues, this thesis will investigate the following research questions:

- How to efficiently use history of encounters to effectively forward bundles?
- What is an effective buffer management policy for use with quota routing protocols that yield high delivery ratios and low delays?

- How to exploit mobility patterns of nodes to yield better network performance when some nodes have a predictable trajectory for a given time period?

As it will become clear in Chapter 2, many routing protocols have been proposed for DTNs. They can be categorized based on the number of bundles replication. Specifically, (i) flooding, or (ii) quota. Flooding-based protocols send a replica of each bundle to any encountered nodes, whereas quota-based protocols restrict the number of replicas. In fact, unlike flooding based routing protocols, the number of replicas in quota-based routing protocols is not dependent on the number of encounters [21]. Flooding based protocols do not require any knowledge of the network topology [21-23]. Despite their robust delivery ratio and low delay, flooding-based protocols have higher energy usage, bandwidth and buffer space consumption [11, 23, 24]. However, the buffer size of devices may be limited, which may lead to bundle loss and low delivery ratios, especially during high traffic loads [21, 22, 25]. In contrast, quota based protocols employ a limited number of replicas, which improve network resource usage [26]. This means, under quota protocols, if senders forward all replicas of a bundle to encountered vehicles, they are no longer allowed to replicate said bundle. In fact, quota based protocols have been proven to achieve a reasonable trade-off between routing performance and resource consumption [27]. However, these routing protocols suffer from comparatively lower delivery ratios even though they are resource friendly [28]. Moreover, a fixed number of replicas for bundle replication lacks the flexibility to react to any changes in resource capacity [29].

This thesis investigates the following research problems. First, it addresses a key limitation of current quota protocols. Specifically, the lack of targeted (efficient) forwarding strategy for semi-predictable DTNs. For example, In the Encounter-Based Routing (EBR) [30] protocol , encountered nodes can receive more replicas if their rate of contacts with other nodes is high. Therefore, replicas are disseminated to area(s) of the network where the rate of encounters is higher than other regions. This means bundle delivery will fail if the destination is in an area where the rate of encounters is lower than other regions. Recall that in semi predictable DTNs, due to their dynamic topology, nodes may not be able to learn the network topology. Also, due to large delays, a key characteristic of DTNs, providing real-time information about the network topology is impractical.

The second research problem addresses the lack of policies for quota-based protocols to efficiently manage bundles. As elaborated in Chapter 2, to date, all buffer management schemes are targeted at flooding based protocols. This is logical as congestion occurs more frequently as compared to quota based protocols. However, under flooding protocols, if a bundle is dropped, there is still a high probability for it to be delivered to its destination. On the other hand, in quota based protocols, as each bundle has finite copies, once a replica is dropped, the delivery probability of the corresponding bundle reduces. In other words, no provisions are provided to replace a dropped replica in order to maintain a high delivery ratio [29]. In the worst case scenario, all replicas may be removed from the network.

The third research problem is the lack of an efficient forwarding strategy for semi-deterministic DTNs. Thus far, past work assumes nodes are pre-loaded with a space-time graph that describes the mobility patterns of nodes. This means routing protocols can take advantage of this graph to improve network performance. For example, given the movement patterns of nodes, it is possible to determine the remaining time until a pair of nodes meets each other again. Similarly, it is possible to calculate the duration of contacts. Consequently, bundles will be forwarded on a predetermined route. In addition, the amount of data that can be transferred during the contact period can be estimated in advance. To date, current space-time graph routing protocols assume that every node is aware of the mobility pattern of all nodes. In other words, nodes are assumed to have the complete space-time graph. However, in practice, this may not be the case. Hence, if a bundle is generated when the space-time graph is not complete, a source node may not find a route towards a destination. Alternatively, the source node may find a route towards destination nodes, but the route may not be optimal.

1.3 Contributions

Henceforth, in light of the aforementioned problems and limitations, this thesis makes the following contributions:

- A comprehensive and in-depth review of the state-of-the-art in DTNs, covering routing protocols, and buffer management protocols. Key strengths and constraints of current protocols are examined and presented. Also, a taxonomy of current protocols is provided based on their features.

- It proposes a novel destination based routing protocol, called DBRP, that determines the optimal number of replicas to forward based on whether a node has met the bundle's destination. In other words, DBRP will forward more replicas to nodes that have met the destination even though the rate of contact may be low in comparison to other nodes. This in effect allows DBRP to disseminate a large number of replicas to the region containing the destination node, which, in turn, increases the probability of delivery. This thesis also studies this idea using a Time Homogeneous Semi-Markov Process (THSMP) and show that a targeted forwarding strategy based on contact history with a destination improves bundle delivery when there are finite number of replicas. Simulation studies over three scenarios show that in terms of a composite metric comprising delivery, delay and overhead, DBRP achieves up to 57% improvement over three well-known routing protocols, namely PROPHET, EBR and Spray and Wait. Moreover, DBRP results in nodes experiencing at least 28% lower buffer consumption.

- It studies a novel queue management policy called QM-EBRP for managing the buffer of nodes when there are finite number of bundles replicas. This is because under quota based protocols, if congestion occurs, dropping a bundle may reduce the probability of delivery. In this respect, QM-EBRP is the first buffer management policy designed for quota based routing protocols. In particular, this thesis makes use of the encounter rate of nodes and context information such as time to live, number of available replicas and maximum number of forwarded bundle replicas to derive a bundle's priority. Simulation results, over a service quality metric comprising of delivery, delay and overhead, show that the proposed policy achieves up to 80% improvement when nodes have infinite buffer and up to 35% when nodes have finite buffer over six popular queuing policies: Drop Oldest (DO), Last Input First Output (LIFO), First Input First Output (FIFO), Most FOrwarded first (MOFO), LEast PRobable first (LEPR), and drop bundles with greatest hop count (HOP-COUNT).

- This thesis also proposes a Mobility-Based Routing Protocol (MBRP) that constructs a space-time graph at every node by recording the mobility pattern of nodes upon contacts. Hence, nodes do not have full knowledge of the network topology. In addition, the space-time graph is dynamic, meaning the trajectory of nodes may only be valid for a given period of time. As the space-time graph may be incomplete, MBRP presents a heuristic that evaluates encountered nodes based on their recorded mobility patterns in order to disseminate a finite number of replicas. The simulation results, over a service quality metric comprising of delivery, delay and overhead, show that MBRP achieves up to 105% improvement as compared to four well-known routing protocols namely, EBR, EPIDEMIC, MAXPROP, and PROPHET. Finally, MBRP is capable of achieving 50% of the performance attained by the optimal algorithm, whereby all nodes are preloaded with the space-time graph.

1.4 Publications

The following papers contain key findings from this thesis.

- Saeid Iranmanesh, Raad Raad and Kwan-Wu Chin, "A Novel Destination-Based Routing Protocol (DBRP) in DTNs", *IEEE International Symposium on Communications and Information Technologies (ISCIT*), Gold Coast, QLD, Australia, 2012.

- Saeid Iranmanesh, Raad Raad and Kwan-Wu Chin, "An Efficient Destination-Based Routing Protocol (DBRP) in DTNs", *Elsevier Journal of Network and Computer Applications,* Under review.

- Saeid Iranmanesh, Raad Raad and Kwan-Wu Chin, "A Novel Queue Management Policy for Intermittently Connected Vehicular to Vehicular Networks ", *Elsevier Pervasive and Mobile Computing*, Under review.

- Saeid Iranmanesh, and Kwan-Wu Chin, "A Mobility Based Routing Protocol in Deterministic DTNs", *Springer International Journal of Wireless Information Networks*, Under review.

1.5 Thesis Outline

This thesis has the following structure:

- *Chapter 2* presents a comprehensive literature review of relevant routing protocols designed for DTNs. Specifically, routing protocols are categorized based on available knowledge of the network topology. In addition, this chapter investigates current buffer management policies. To this end, an extensive qualitative comparison is provided that highlights the gaps in the literature of both routing and buffer management policies.

- *Chapter 3* proposes a quota based routing protocol that considers contact history of nodes when selecting the next hop node. In addition, it presents an analysis of contact prediction based on a semi-Markov model which shows that if nodes know that a contact will happen between a node and a destination in a given period of time, the probability of delivery through that node is maximum.

- *Chapter 4* proposes a queue management policy that works under encounter based quota protocols. Specifically, it prioritizes buffered bundles during congestion in order to drop/forward bundles and/or contact duration is short.

- *Chapter 5* proposes a forwarding strategy that exploits predictable mobility patterns of nodes, and consider space-time graph with expiration time. A heuristic is proposed to forward bundles when the space-time graph is not complete.

- *Chapter 6* summarizes the research challenges addressed in this thesis, and outlines findings and open problems.

Chapter 2

Literature Review

This chapter consists of two main parts. The first part provides a comprehensive review of current DTN routing protocols. It also highlights the key problems solved by these routing protocols and their characteristics. The second part reviews queue management policies, and outlines their limitations.

The following sections are organized according to the said parts. Section 2.1 provides an overview of current routing protocols, and classifies them into three groups. Section 2.1.1 investigates dynamic routing protocols. Section 2.1.2 reviews history based routing protocols and Section 2.1.3 considers space-time routing protocols. As for the second part, Section 2.2 provides an overview of current queue management policies. It presents two categories of policies. The first, as outlined in Section 2.2.1, are global knowledge schemes, followed by those that use local knowledge; see Section 2.2.2. Finally, Section 2. 3 provides an extensive qualitative comparison that highlights the gaps in current routing and buffer management policies.

2.1 Overview of Routing Protocols

Current routing protocols can be categorized into three groups: (a) *Space-time graph routing protocols*, where the network is deterministic and every node has a complete space-time graph. These protocols are suited for applications such as interplanetary communications, where contacts are scheduled and the trajectory of nodes is known in advance [31-34], (b) *History-based routing protocols*, where the network is at least semi-predictable. These protocols are designed for applications such as communications involving buses, where their contacts may not be completely predictable due to environmental conditions. Note that nodes have a predefined mobility pattern. However, contacts may be affected by unexpected conditions [35-41], and (c) *Dynamic routing protocols*, where nodes have random movement patterns. These protocols are suitable for applications such as wildlife communications where tagged animals have random movement.

Routing protocols can also be classified into two groups based on the number of bundle replications: (i) Flooding, and (ii) Quota. Each type of routing protocols has its advantages and disadvantages. For example, non-replication based protocols consume much less network resources such as buffer and bandwidth. This is because only a single-copy of a bundle is forwarded at any given time [42]. In addition, when a bundle is delivered to its destination, no node has a copy of the bundle. This requires the destination to generate an acknowledgement message. However, these protocols cannot guarantee a high delivery ratio if the network topology is highly dynamic. As a result, these protocols are suitable for deterministic/completely predictable networks [25]. In contrast, replication-based protocols achieve higher delivery ratios if the network is not completely predictable [9]. Hence, history and dynamic routing protocols use multiple copies to improve the delivery ratio and delay. On the downside, these protocols consume more resources as compared to non-replication based protocols. Furthermore, flooding protocols inherently do not have any a bundle replication limit. This results in higher resource consumption as compared to quota protocols. Table 2-1 shows the taxonomy of all relevant routing protocols. Notice that in [46], the protocol may experience a variable dissemination rate “low-Medium-High”. In details, if source node does not receive delivery acknowledgement, the source node forwards additional n copies. So, in the best case, the number of disseminated replicas is n whereas in the worse case, it increases to $T \times n$ where T is the number of periods. Also, note that local information refers to the information that locally exists at each node and/or can be used through one hop (1-neighbor information). In addition, the information used is distributed rather than being centralized to a particular node.

Table 2-1 A comparison of routing protocols

Protocols	Flooding / Quota	Protocol type	Information type	Decision criterion	Estimation of link forwarding probability?	Computational complexity	Resource friendly?	Dissemination rate?
Epidemic [43]	Flooding	Dynamic	None	Random	No	O(n)	No	High
(PQERPV) [44]	Flooding	Dynamic	None	Probabilistic	No	O(n)	No	Medium-High
Spyropoulos *et al*. [40]	Quota	Dynamic	None	None	No	O(1)	Yes	Low
Grossglauser *et al*. [45]	Quota	Dynamic	None	Random	No	O(1)	Yes	Low
Spyropoulos *et al*. [25]	Quota	Dynamic	None	Random	No	O(r)	Yes	Low
Bulut *et al*. [46]	Quota	Dynamic	Global	Random	No	O(r)	No	Low-Medium-High
Sandulescu *et al*. [47]	Quota	Dynamic	Local	Contact duration	No	O(r)	Yes	Low
Zebranet project [11]	Flooding	History	Local	Node	Yes	O(n)	No	Medium
PROPHET [5]	Flooding	History	Global	Link	Yes	O(n)	No	Medium
CAR [48]	Quota	History	Local	Node	Yes	O(n)	Yes	Low
NECTAR [50]	Flooding	History	Global	Link	Yes	O(n)	No	Medium

Protocols	**Flooding / Quota**	**Protocol type**	**Information type**	**Decision criterion**	**Estimation of link forwarding probability?**	**Computational complexity**	**Resource friendly?**	**Dissemination rate?**
Davis *et al*. [49]	Flooding	History	Global	Link	Yes	O(n)	No	medium
Kalantari *et al*. [51]	Quota	History	Local	Node	Yes	O(1)	Yes	Low
UDP [52]	Quota	History	Local	Node	Yes	O(r)	No	Medium
EBR [30]	Quota	History	Local	Node	Yes	O(r)	Yes	Low
Spray & Focus [53]	Quota	History	Local	Node	Yes	O(r)	Yes	Low
FRESH [16]	Flooding	History	Local	Node	Yes	O(n)	No	Medium
SEPR [54]	Flooding	History	Local	Node	Yes	O(n)	No	Medium
MEED [23]	Flooding	History	Global	Link	Yes	O(n+m)	No	Medium
MV [55]	Flooding	History	Global	Link	Yes	O(n+m)	No	Medium
MaxProp [9]	Flooding	History	Global	Link	Yes	O(n+m)	No	Medium
GeOpps [56]	Quota	History	Global	Node	Yes	O(r+m)	Yes	Low
GeoSpray [57]	Quota	History	Global	Node	Yes	O(r+m)	Yes	Low
Leguay *et al.* [36]	Flooding	History	Global	Node	Yes	O(n+m)	No	Medium
Huang *et al*. [58, 59]	Quota	Space-time	Global	Link	Yes	O(m^2+m)	Yes	Low

Protocols	Flooding / Quota	Protocol type	Information type	Decision criterion	Estimation of link forwarding probability?	Computational complexity	Resource friendly?	Dissemination rate?
Xuan *et al*. [60] and Ferreira [61]	Quota	Space-time	Global	Link	Yes	O(m^2+m)	Yes	Low
Handorean *et al*. [62]	Quota	Space-time	Global	Link	Yes	O(m^2+m)	Yes	Low
Jain *et al*.[31]	Quota	Space-time	Global	Link	Yes	O(m^2+m)	Yes	Low

Abbrevations:

m = Number of nodes

n = Number of nodes which do not have a given bundle

r = Number of bundle's replicas

* The computational complexity of a routing algorithm is the number of runs that the algorithm will require in the worst case for a bundle.

** For the algorithms that require global information the complexity of data collection is also applied.

2.1.1 Dynamic routing protocols

In this category of routing protocols, sender nodes forward bundles to their neighbours without using any knowledge of links or paths. For example, Vahdat *et al*. [43] propose a pure epidemic routing protocol where every sender node floods its buffered bundles to every encountered node. If a sender has a high encounter rate, the number of disseminated bundles is large. Hence, bundles are quickly disseminated throughout the network. Their simulation results show that epidemic routing can deliver all bundles when nodes have an infinite buffer size and bundles have a large expiration time. Although this protocol achieves a high delivery ratio and low delay when nodes have unlimited buffer space, it suffers from high overhead due to the high dissemination rate of bundles. In addition, when nodes have limited memory, due to the high rate of arriving bundles, receiver nodes have to drop a large number of bundles. This results in two main problems. First, nodes may receive bundles that had existed in their buffer. Second, bundles may not be carried for a sufficient duration to be forwarded in future contact opportunities.

To improve the performance of pure epidemic [43], Matsuda *et al*. propose the (p-q) epidemic with vaccination routing protocol (PQERPV) [44]. Their proposed algorithm forwards bundles according to a probability value. For example in [43] the probability of forwarding is one, meaning that upon each contact, all bundles are forwarded. In contrast, if the probability of forwarding is zero, no bundle is forwarded. PQERPV assigns two probabilities for forwarding: q indicates the probability of receiving a bundle from a source and p represents the probability of receiving a bundle from other nodes. Hence, given q and p, bundles are received from a source and relays with the probability of q and p respectively. Notice that in PQERPV, bundles are blindly forwarded in a probabilistic manner. Hence, if p and q are high, PQERPV works similarly to pure epidemic [43]. In contrast, if p and q are low, bundles experience a low dissemination rate.

Spyropoulos *et al*. [40] propose a single copy scheme that involves the source directly delivering a bundle to the destination. In this case, if a destination is located in an area far away from a source node, bundles will never be delivered. Similarly, Grossglauser *et al*. [45] propose a two-hop forwarding approach. They assume that nodes with infinite buffer move independently in a DTN, and every node will be in contact with other nodes for a short period of time. Given said assumption, sender nodes exchange bundles with randomly encountered nodes. These nodes do not exchange the bundles with any other nodes but the destination. Hence, a bundle will be delivered over two hops. They also prove that a bundle is guaranteed to be delivered. Although their approach has less overhead as compared to [43], bundles may fail to be delivered if a destination node is not reachable via two hops. In addition, as the bundle dissemination rate is low, bundles experience large delays.

In order to overcome the problems in [40, 45], Spyropoulos *et al*. [25] propose 'Spray and Wait'. Source nodes make *n* copies of each generated bundle. Upon each contact source nodes send a copy of each buffered bundle to any encountered node. As bundles can be replicated *n* times, each bundle at a source is forwarded to the first *n* encountered nodes. From then onwards, these nodes are responsible for carrying the copies until they encounter the destination. Thus, this algorithm is a multi-copy, two-hop scheme. Although 'Spray and Wait' is a resource friendly protocol, it still suffers from the following problem. In particular, it sends replicas to nodes that move in areas that are close to the source node. As a result, bundles may not be delivered if the destination is in a different area. To resolve this issue, the authors also proposed binary 'Spray and Wait'. Upon each contact, a node forwards half of a bundle's replicas. Hence, contrary to 'Spray and Wait' and [40, 45], if a destination is reachable via two hops, a bundle can be delivered.

In a similar work, Bulut *et al*. [46] propose an algorithm that broadcasts replicas in different periods. The main approach is that source nodes generate a finite number of replicas in each period. Hence, they assume a number of periods based on a bundle's lifetime. Initially, a source node forwards *n* copies to the first *n* encountered nodes, and waits to receive an acknowledgment. If delivery fails, the source node forwards additional copies to encountered nodes that do not have a copy of the bundle. As a result, with each passing period, more copies are injected into the network to increase the probability of delivery. However, due to the large delays in DTNs, if a bundle is delivered, its acknowledgement may not reach the bundle's source node promptly, causing a large number of replicas to be forwarded to nodes.

In a different work Sandulescu *et al*. [47] propose ORWAR, a protocol that limits the number of replicas to n. In addition, ORWAR assumes that each node has a priority. Their proposed algorithm utilizes local connectivity knowledge such as node speed, direction of movement and radio properties i.e., data rate, and GPS, to decide the contact period time C. This time and given data rate are then used by ORWAR to compute the data size to be transmitted in each contact. Accordingly, bundles are sorted based on their priority and size. Relay nodes forward half of the available bundle replicas with the highest priority if the bundle has size S_{max}, where,

$$S_{max} = b \times C \tag{2.1}$$

where b is the data rate and is given by the device radio properties. For example, consider Bluetooth 2.0 with a data rate of 250kBps. Assuming a contact with a duration of 10 seconds, 2500 kB of data can be transferred. In this case, a sender node forwards half of the replicas that have the highest priority if the bundle's size is less than S_{max} i.e., 2500 kB.

The dynamic routing protocols discussed thus far suit unpredictable DTNs where nodes' movement is random, and unpredictable. Consequently, these protocols do not consider any contact information between nodes. The flooding schemes such as [43, 44] suffer from high overhead especially when nodes have a limited buffer size. In contrast, quota protocols [25, 40, 45, 47] are resource friendly but they suffer from low delivery ratios. This is due to nodes blindly forwarding a finite number of replicas. In this case, replicas may be forwarded to areas far away from the destination.

2.1.2 History based routing protocols

This section considers routing protocols in semi-predictable networks. In these networks, a route between a source and a destination node may not be fully predictable. Hence, source nodes may wait until a route becomes available. However, if source nodes do not send bundles in the hope of better forwarding opportunities, bundles may expire. Hence, upon each contact, routing protocols in this category will rely on (i) next hop information, such as the history of a node's encounter rate, or (ii) end-to-end metrics, such as the expected shortest path or average end-to-end delay [18].

Initially, history based schemes target flooding protocols. These protocols decrease overheads by forwarding bundles to nodes that have a high rate of contact. The Zebranet project [11] is one of the earliest attempts to use the history of encounters. Zebras are fitted with tracking collars, and periodically, a researcher (base station) moves into a zebra habitat to collect data. Each zebra has a hierarchy level based on its frequency of contact with a base station and exchanges data only with another node that has a higher hierarchy level. The problem with this method is that nodes with a higher hierarchy level are responsible for delivering data to those at lower hierarchy levels. In other words, nodes experience non-uniform resource consumption. In another scheme, Lindgren *et al.* [5] propose PROPHET, which uses a metric that indicates how likely a node will deliver a bundle to a given destination successfully. For a given pair of sender and destinations nodes, the delivery predictability is calculated based on three parts. In the first part it updates the delivery predictability whenever the destination is encountered. Specifically, this update is calculated as follows,

$$P_{(a,b)} = P_{(a,b)_{old}} + \left(1 - P_{(a,b)_{old}}\right) \times P_{init} \tag{2.2}$$

where $P_{init} \in [0,1]$ is an initialization constant. In other words, if destination *b* is frequently encountered by node *a*, there is a high delivery predictability from node *a* to destination *b*. In contrast, If nodes *a* and *b* do not meet each other for a while, they are less likely to meet each other in the future. Thus, the delivery predictability is updated by an aging equation as follows,

$$P_{(a,b)} = P_{(a,b)_{old}} \times \gamma^{k} \tag{2.3}$$

where $\gamma \in [0,1)$ is a constant to age the delivery predictability, and k is the number of elapsed time units since their last aging. PROPHET also supports transitive property for delivery predictability. This is based on the observation that if sender node a and destination node c frequently meet node b, node b is a good bundle carrier. The following equation considers the effect of this transitivity on delivery predictability.

$$P_{(a,c)} = P_{(a,c)_{old}} + \left(1 - P_{(a,c)_{old}}\right) \times P_{(a,b)} \times P_{(a,c)} \times \beta \qquad (2.4)$$

where $\beta \in [0,1]$ is a scaling constant that determines the impact of the transitivity on the delivery predictability. According to the obtained delivery predictability, if the delivery predictability of an encountered node is greater than the sender's delivery predictability, a bundle is forwarded. However, if a source meets many nodes that have a high delivery predictability, bundles are flooded throughout a network. This results in high overheads. On the other hand, if a source meets many nodes that have a low delivery predictability, bundles may never leave the source. Similar to PROPHET, the Context-Aware Routing (CAR) protocol [48] considers context information such as the likelihood of meeting other nodes and the remaining energy level of nodes to deliver a bundle. This context information is then fed into a Kalman Filter [63] in order to predict future energy values.

In [49], the authors consider the likelihood of delivery. When two nodes meet each other, the bundles at the sender node are sorted based on the likelihood of delivery. Amongst the bundles that are missing at a receiver, a sender node selects the top n bundles that have the highest delivery probability. The probability of delivery is calculated based on the likelihood of contacts. Specifically, when node a meets node d, the likelihood of their meeting is updated as follows,

$$P_{(a,d)} = \lambda \times P_{(a,d)_{old}} + 1 \qquad (2.5)$$

where initially $P_{(a,d)} = 0$ and $\lambda = 0.95$ is the decay rate of the meeting likelihood. Node a also needs to update its other contacts probabilities with other nodes.

$$P_{(a,b)} = \lambda \times P_{(a,b)_{old}} + \beta \times P_{(b,d)} \tag{2.6}$$

where $b \neq d$ and $\beta = 0.15$. Equations (2.5) and (2.6) can be interpreted as follows. First, if node a encounters d, node a is likely to encounter node *d* again in the future and is a good candidate for passing bundles to node *d*. Second, if node a encounters node *d* and node *d* has a high encounter value for destination *c*, then node *d* is a good bundle carrier for destination node *c*. Lastly, the contact probability degrades over time such that the links that occur infrequently have a low delivery probability.

The NECTAR protocol [50] uses a metric called Neighbourhood Index when selecting the next hop. This index is based on the history of a node's contacts where those that it encounters frequently have a high index value. As an example, when nodes *i* and *j* meet each other for the first time, the Neighbourhood Index assigned to each other is one. From then onwards, whenever they meet each other again, the Neighborhood Index and the Contact counter increase linearly. Based on the calculated Neighbourhood Index, upon contact, nodes exchange Neighbourhood Index, and use an encounter node's index with a bundle's destination to determine whether it is a good next-hop node for the bundle.

In [51], the Kalantari *et al*. propose a single-copy forwarding protocol that is inspired by thermodynamics where heat is exchanged between objects. They use a metric called "temperature" whereby a destination node termed the 'sink' has a high constant value. Hence, when nodes meet the sink, they will be "heated", meaning their temperature metric increases. This implies that the nodes with a higher temperature have recently encountered the sink, meaning they are good candidates to be given bundles for the sink. Upon each contact, say between node *a* and *b*, the temperature of node a is updated as follows,

$$T_a = \gamma_{ab} \times (T_b - T_a) \tag{2.7}$$

where γ_{ab} is a heat exchange coefficient that is symmetric between connected nodes. In other words, when a node with a high temperate encounters a node with a low temperature the one with the higher temperature will decrease in value. Hence, this parameter is sensitive to the mobility of nodes and the frequency of encounters. Sender nodes forward a single-copy of the buffered bundles toward nodes with a higher temperature, meaning these nodes have recently encountered the sink.

Contrary to [51] where there is only a single copy of a bundle in the network, Li *et al*. propose a multi copy scheme, called utility based distributed routing protocol (UDP) [52], that selects hops based on a utility function. The proposed utility function is derived from the number of connections a node has with their home communities. Specifically, a node that visits these communities frequently makes it a good bundle carrier for any destinations that belong to these communities. In UDP, the number of replicas for each generated bundle is limited to *k*. Hence, when a bundle is generated at a source, the *k* replicas are forwarded to the first *k-1* encountered nodes. After that, each relay sends its only copy of a bundle based on the following utility function,

$$P_{ij} = \frac{T_{ij}}{T_{(i)}} \tag{2.8}$$

where P_{ij} is the utility that node *i* meets node j, and T_{ij} is the number of times that node *i* encounters node j within a time interval $T_{(i)}$. Here $T_{(i)}$ is the period of time between two consecutive contacts that node *i* has with a given home community. However, nodes need to update their utility if they already have a utility value. This is carried out as follows[1],

$$P_{ij(updated)} = \alpha \times P_{ij(old)} + (1-\alpha) \times P_{ij(new)} \tag{2.9}$$

where $\alpha \in (0,1)$ is a weighting constant. In words, a node with a high utility value is more likely to deliver bundles destined to their home community. Hence, when a relay node encounters a node with a higher utility value, the bundle is forwarded to the encountered node. However, if a node from a destination's home community is not encountered, bundles will never leave the source.

[1] The authors have not specified the value of alpha. If the value of alpha is the same as in [30], the impact of $P_{ij(new)}$ on the updated value is less than $P_{ij(old)}$, which is unreasonable

Similarly, Nelson *et al*. [30] propose an encounter-based routing (EBR) protocol that generates a finite number of replicas for each bundle and also considers the history of nodes' encounters in order to maximize bundle delivery. Every vehicle running EBR is responsible for maintaining its past average rate of encounters with other vehicles, which is then used to predict future encounter rates. In terms of its encounter rate, a vehicle maintains two pieces of local information: an encounter value (EV), and a current window counter (CWC). The variable EV represents a vehicle's past rate of encounters as an exponentially weighted moving average, while CWC is the number of encounters in the current time interval. EV is updated periodically to account for the most recent CWC. Specifically, EV is computed as follows:

$$EV = \alpha \times CWC + (1-\alpha) \times EV_{(old)} \tag{2.10}$$

where $\alpha \in (0,1)$ is a weighting coefficient; i.e., $\alpha = 0.85$. In EBR, every 30 seconds, the encounter rate of nodes is updated and the CWC is reset to zero.

The primary purpose of tracking the rate of encounters is to decide how many replicas of a bundle a vehicle will transfer during a contact opportunity. Hence, when vehicles a and b meet each other, vehicle a sends a proportional number of the i^{th} bundle M_i based on the encounter rate of both sender and receiver. Specifically,

$$k = m_i \times \frac{EV_b}{EV_b + EV_a} \tag{2.11}$$

where m_i is the available number of replicas for the i^{th} bundle at node a. The terms EV_a and EV_b respectively represent the encounter rate for nodes a and b. As a result, k replicas of bundle M_i is forwarded to node b. In words, the nodes that experience a large number of encounters are most likely to successfully pass the bundle along to the final destination than nodes that do not encounter other nodes frequently. However, if a destination is located in a low density area where the rate of encounters is low, it may never receive transmitted bundles.

In [53], Spyropoulos *et al*. propose a quota protocol, called 'Spray and Focus'. This algorithm performs similarly to 'Spray and Phase' in the first phase where replicas are forwarded to the first n encountered nodes. A key difference, however, is that 'Spray and Focus' uses a utility function based on a timer that records the elapsed time since a pair of nodes met each other. The authors assume that a small timer value implies two nodes are close to each other in terms of distance. This means, when the time between contacts of two nodes is short, the mobility pattern of these two nodes is approximately similar. In order to calculate the utility function, every pair of nodes i and j records the time elapsed since their last contact, called T_{ij}. They also update the utility value in a manner similar to PROPHET [5]. Accordingly, node A forwards a bundle copy to node B for destination D if $T_{AD} > T_{BD}$.

Other aspects of contact history are used in FRESH [16] and SEPR [54]. In FRESH, encounter time is considered and a node that was encountered five minutes ago is deemed to be closer than a node that was encountered five hours ago. A key limitation, however, is that FRESH does not consider nodes moving with different speeds. In particular, high speed nodes are likely to have more encounters as compared to low speed nodes. As a result, traffic will be directed to parts of the network where relayed nodes have a higher speed even if the distance between the relayed nodes and destination is long. Moreover, FRESH may cause congestion as traffic is only directed to nodes with high mobility. Tan *et al*. propose Shortest Expected Path Routing (SEPR) [54] to address the issue of hop selection by considering contact duration of nodes with the required destination. They believe contact duration between nodes determines how likely nodes are in contact with each other. SEPR calculates the occurrence probability of link i as follows,

$$P_i = \frac{T_c}{T_w} \tag{2.12}$$

where T_c is the duration of contact for link i. Here, T_w is the length of the sampling time. Using Equation (2.12), the authors then calculate the expected path length towards a destination as follows,

$$E_{Path} = \sum_{i \in P} \frac{1}{P_i} \tag{2.13}$$

where i represents the links in path P. From Equation (2.13), if the expected length of a path is small, the authors assume a higher probability of delivery. In order to calculate the expected path length, each node maintains the contact probability of its encounters in a table and exchanges the table with any encountered node. This way, any update in the probability of contacts is propagated through out the network. Using this information, nodes update their local table and perform a modified Dijkstra algorithm to calculate the shortest path length to all other nodes. Each buffered bundle for a given destination is then assigned a path length from the current node. To forward bundles upon a contact, for every buffered bundle, if the path length from the encountered node is less than the value recorded at the sender node, the bundle is forwarded. A drawback is that two nodes may have many short contacts duration instead of one long contact duration. In this case, nodes that have a large number of short contacts may be more reachable in the future than nodes that have a small number of long contacts.

Similar to SEPR, Jones *et al*. [23] improved the method in [31] by proposing Minimum Estimated Expected Delay (MEED). It computes the expected delay (ED) based on the recorded connection and disconnection time of nodes' contacts in a given time interval. Specifically,

$$ED = \frac{\sum_{i=1}^{n} d_i^2}{2t} \tag{2.14}$$

where n is the total number of disconnected periods, d_i is the duration of the i-th disconnection, and t is the total time slots during these disconnections. Based on the distribution of expected link delays, a modified Dijkstra algorithm is used to find the optimal route. However, if the time interval is large, the metric *ED* slowly changes when frequent contacts happen. On the other hand, although a small time interval can help the metric adapt quickly to frequent connections, the metric is sensitive to random fluctuations. The difference in MEED and [31] is that under MEED a decision is made with the most recent information, while in [31] a decision is made offline as the information will not change over time [18].

Burns *et al.* in [55] presented an extension of the work in [49]. They propose the meets and visits (MV) protocol where every node visits certain regions and learns the frequency of encounters between nodes. From the history of encounters, the likelihood of delivering a bundle via a specific path is calculated. Then, bundles are prioritized based on the obtained delivery probability. Specifically, in a network comprising of *N* nodes, the delivery probability of a bundle from the current node *k* to a region *i* with *n* hops is calculated as follows,

$$P_n^k(i) = 1 - \prod_{j=1}^{N} [1 - m_{(j,k)} \times P_{n-1}^j(i)] \tag{2.15}$$

where

$$P_0^k(i) = \frac{t_i^{(k)}}{t} \tag{2.16}$$

where $t_i^{(k)}$ is the number of time units that node *k* has visited region *i* within the past *t* time units. Finally, the probability of meeting based on the contacts in the last *t* time units is calculated as follows,

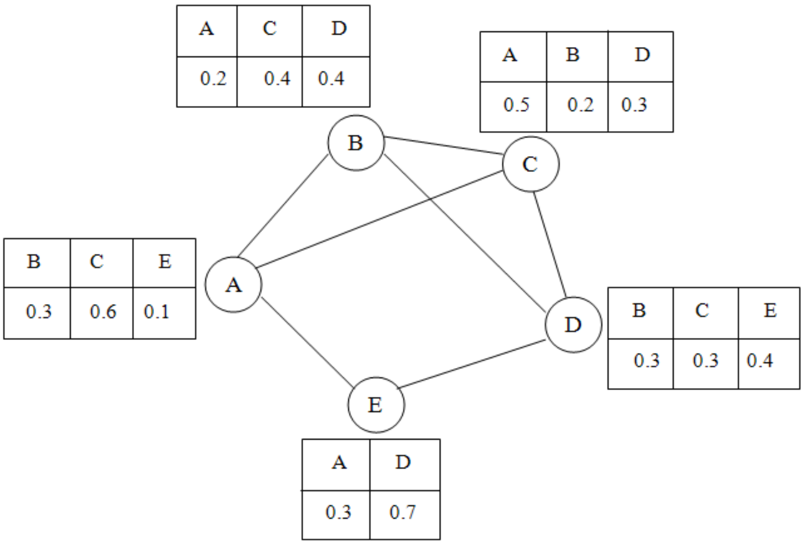
$$m_{(j,k)} = \frac{t_{(j,k)}}{t} \tag{2.17}$$

where $t_{(j,k)}$ is the number of contacts between nodes *j* and *k*. The forwarding process of MV algorithm works in the same manner as [49] where bundles are sorted based on the delivery probability. Then, the top *n* bundles that have with the highest delivery probability and do not exist at receiver node are forwarded.

Burge *et al.* present MaxProp [9], a protocol that assigns a weight to each link and derives a cost for each possible route. In fact, each node keeps track of the probability of meeting other nodes. For example f_j^i represents the probability that node *i* meets node *j*. For all nodes, the meeting probability is initially set to $\frac{1}{|s|-1}$, where s is the number of nodes in the network. When node i encounters node j, the value of f_j^i is incremented by one. Then, all the probabilities at node i are re-normalized. This way, nodes that are encountered infrequently obtain lower values over time. Upon contact, nodes exchange these values. They then calculate a cost for each possible path towards destination nodes. The cost for a path via nodes (*i, i+1, . . . , d*) is calculated as follows.

$$Cost(i, i+1, \ldots, d) = \sum_{x=i}^{d-1} [1 - f_{x+1}^{x}] \tag{2.18}$$

In other words, the cost of a link is the probability that the link does not occur. Hence, the cost of a path is the summation of the links' cost. MaxProp uses Equation (2.18) to find the lowest path cost amongst all possible paths. Figure 2-1 shows a network comprising of five nodes namely A, B, C, D and E where their contacts are represented by edges. The table next to each node shows the probability of contacts with other nodes. For example, the probability that node A meets node B is 0.3. Now assume that node A generates a bundle for destination D. In this case, based on Equation (2.18), MaxProp calculates the cost of each possible path from A to D. Then, the path with the minimum cost is selected. In this example, the path via node C, i.e., ACD, has a minimum cost of 1.1. A key limitation of MaxProp is that when a contact happens, the probability of other contacts changes. This implies that the probability of contacts is dependent on each other. However, contacts may happen independently.



Path Costs:

ABCD = $(1 - 0.3) + (1 - 0.4) + (1 - 0.3) = 2.0$

ABD = $(1 - 0.3) + (1 - 0.4) = 1.3$

ACD = $(1 - 0.6) + (1 - 0.3) = 1.1$

ACBD = $(1 - 0.6) + (1 - 0.2) + (1 - 0.4) = 1.8$

AED = $(1 - 0.1) + (1 - 0.7) = 1.2$

Figure 2-1 An example of MaxProp where the cost, using Eq. 2.18, from node A to D is calculated to have the minimum value of 1.1.

A number of routing protocols rely on the location information of nodes and other mobility parameters provided by GPS. The majority of these protocols cannot be applied to DTNs as they do not support the store-carry-forward paradigm. For example, in [64] and [65], the presented geographic routing strategies for vehicular ad hoc networks are not able to deal with intermittent network partitions that can last for a long period of time. In contrast, Leontiadis *et al*. propose a Geographical Opportunistic Routing (GeOpps) protocol [56] for DTNs and use nodes' geographical information to route bundles. Hence, they assume nodes are aware of their geographical position. Accordingly, nodes are able to calculate the route, distance, and time between two points. In addition, they assume that nodes know the location of destination nodes. Hence, every node is aware of the speed, and current route of destination nodes. GeOpps maintains a single-copy of each bundle in the network, and forwards bundles as follows. Every node i determines the nearest point, called NP_i, on its predetermined route to a destination (D). Then, GeOpps computes ETA_i which is the time that node i arrives at NP_i. In addition, GeOpps computes ETA_{NP_i} which is the time that the node i meets destination *D*. Based on ETA_i and ETA_{NP_i}, a utility function, called the minimum estimated time of delivery ($METD_i$) is derived for node i.

$$METD_i = ETA_i + ETA_{NP_i} \tag{2.19}$$

In other words, Equation (2.19) determines the closeness betwee node i and a bundle's destination. When nodes pass bundles to a node that is closer to a destination, the bundle will have a higher chance of being delivered. Based on this observation, a sender node only forwards a bundle if the minimum time of delivery via an encountered node is lower than the minimum time of delivery via the sending node. For example in Figure 2-2, vehicle *X* carries a bundle for destination *D*. Vehicle *X* meets vehicle *Y* at location P_1. If $METD_Y$ is lower than $METD_X$, the bundle is forwarded to vehicle *Y*. This implies that the time to go from P_1 to NP_Y and then from NP_Y to *D* is lower than time to go from P_1 to NP_X and then from NP_X to *D*. As a result, node *X* forwards the bundle to node *Y*. From then onwards, if node *Y* meets another vehicle that has a lower time of delivery i.e., is faster or close to *D*, bide *Y* passes the bundle to the vehicle.

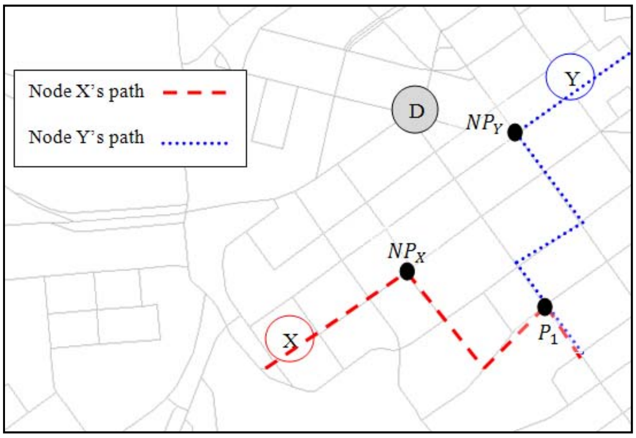


Figure 2-2 An example of GeOpps.

Similarly, Soares *et al*. propose the GeoSpray routing protocol [57] which is inspired from [56]. The only difference is the number of replicas in the network. Contrary to GeOpps [56] that maintains only one copy of a bundle, GeoSpray generates up to *n* replicas for each bundle. Upon contact, if the *METD* of an encountered node is lower than the *METD* of the sender node, half, i.e., *n*/2, of the replicas are sent to the encountered node. In addition, when nodes have a single copy of a bundle, similar to GeOpps, they are allowed to forward the single copy to another node that can deliver the data closer to the destination.

In a different work, the authors of [36] use the mobility pattern of nodes to derive four functions as the measure of similarity between nodes. Specifically, when nodes meet each other they exchange their learned mobility patterns. Based on the learned mobility patterns, the similarity of a node and destination can be calculated via the following functions: (i) Euclidean distance, (ii) Canberra distance, (iii) Cosine angle separation, and (iiii) Matching distance. Briefly, if p = (p_1, p_2,) and q = (q_1, q_2) are two points, then the Euclidean distance between p and q is calculated as

$$\sqrt{\sum_{i=1}^{2}(p_i - q_i)^2} \tag{2.20}$$

Canberra distance is the sum of a series of fractional differences of two points. Specifically,

$$\sum_{i=1}^{2}\frac{|p_i - q_i|}{|p_i| + |q_i|} \tag{2.21}$$

Cosine similarity measures the cosine angle between two points; i.e.,

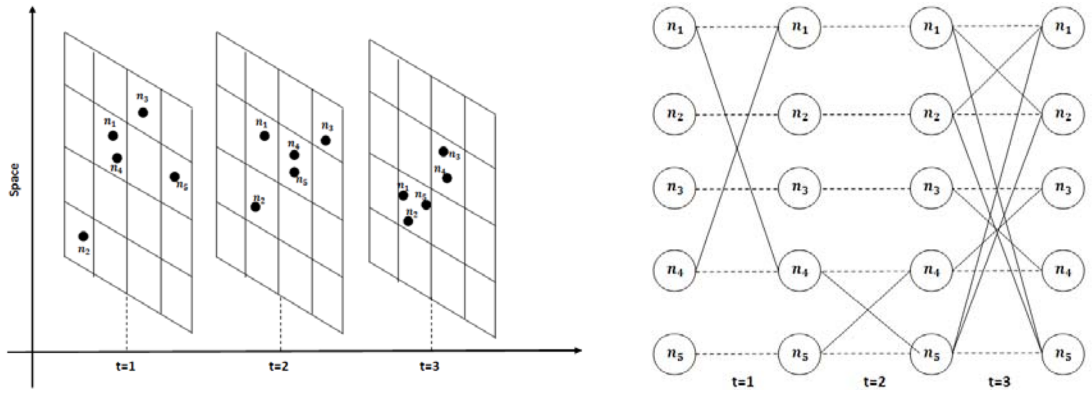
$$\frac{\sum_{i=1}^{2}(p_i \times q_i)}{\sqrt{\sum_{i=1}^{2} {p_i}^2 \cdot \sum_{i=1}^{2} {q_i}^2}} \tag{2.22}$$

Matching distance considers two points on a given axis are similar if their difference is less than or equal to a value. According to these measurements, a sender node can decide to send bundles to nodes that are closer to the destination or they are going towards the destination.

In summary, compared to dynamic routing protocols, history based protocols offer the best network performance in terms of delivery ratio and delay. However, the majority of history based routing protocols are flooding based where despite their robustness, they suffer from high overhead and do not use resources efficiently. For example, in PROPHET [5], a controlled flooding protocol, if a source node encounters many nodes with a low contact rate for a given destination, bundles may never leave the source [30]. Conversely, if a source meets many nodes with a high contact rate, bundles are flooded throughout a network [18, 66]. A solution is to employ quota protocols to limit the number of replicas for each bundle. Hence, these protocols need to efficiently forward a limited number of replicas such that the delivery ratio increases. For example, in EBR [30], an encountered node can receive many replicas if its rate of contact with other nodes is higher than the sender. Therefore, replicas are disseminated to area(s) of the network where the rate of encounters is higher than other regions. However, bundle delivery will fail if the destination is in an area where the rate of encounters is lower than other regions.

2.1.3 Space-time graph routing protocols

This section reviews routing protocols designed for DTNs where their topology can be represented by different graphs over time, a so called space-time graph. As shown in Figure 2-3 (a), the location of nodes and network topology change over time. Also, notice that nodes come within communication range if they are in the same cell. Figure 2-3 (b) shows the corresponding space time graph for the DTN in Figure 2-3 (a).



(a) (b)

Figure 2-3 A time-evolving DTN, a) time-evolving topologies of a DTN (a sequence of snapshots), b) corresponding space-time graph

Xuan *et al.* [60] and Ferreira [61] use a space-time graph to model a dynamic topology where contacts are scheduled in advance. Each edge of the graph is assigned a time interval to represent the link's active time. We can see this in Figure 2-4 where the intervals are represented by edges. For example, the link from node *S* to *A* is available at time one, and the link from node *C* to node *D* is available from time one to three. Accordingly, their proposed forwarding strategy aims to find (i) the earliest time to reach one or all destinations, and (ii) has minimum hops. As an example, in Figure 2-4, the minimum hop path for a given bundle from node *S* to node *D* is four hops within one time interval whereas if node *S* carries the bundle up to time four, node *S* can directly deliver the bundle through one hop at time four. In their proposed algorithm, the fastest path amongst all possible paths with the minimum hop count is selected. This does not necessarily yield a path with the lowest delay. In other words, a path with a higher hop count may exist which allows bundles to arrive earlier.

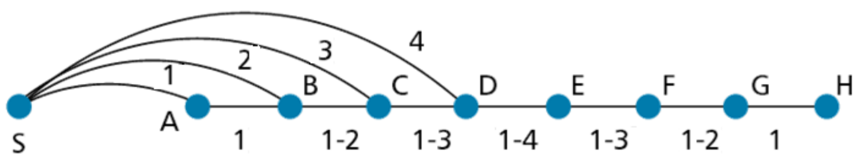


Figure 2-4 The minimum hop path from S to D takes four hops at time interval one, whereas the shortest path to D takes only one hop, but at time interval four.

Similarly, Handorean *et al.* [62] propose different path selection algorithms with consideration for full or partial topological information. They first consider the case where all nodes have full knowledge of the network topology with respect to space and time. In this case, all possible paths from source to destination nodes are extracted. Then, each path is evaluated based on delay or number of hops. In the second case, nodes are assumed to learn their own mobility pattern over time, meaning that nodes do not have a full knowledge of the future network topology. Hence, in order to learn the network topology, nodes exchange their recorded mobility pattern when they meet each other. Also, if a node wants to send a bundle, it computes a route using its incomplete space-time graph. Naturally, the discovered path may not be optimal. Moreover, a source node may fail to discover any path to a destination. In this case, sender nodes forward bundles to any encountered node. Accordingly, these bundles record the sequence of nodes that they traverse. This facilitates two kinds of information. First, receiver nodes are able to learn which nodes have a copy of the bundle. Second, the history of relays can be used as a prediction of future delivery where another bundle may be delivered through the same set of relays.

In [31], Jain *et al*. consider a space-time graph where the edges are weighted based on the arrival time of a bundle at a given node. In order to find the optimal route that has the minimum delay in delivering bundles, Jain *et al.* use a modified Dijkstra algorithm. Similarly, in [67], Dijkstra [68] or Floyd-Warshall [69], are used in the proposed space-time graph routing protocol that has two phases (i) initialization, and (ii) the shortest path computation. The initialization phase computes the delay between source nodes and uses this as the link cost. In the shortest path computation phase, the Floyd-Warshall algorithm is used to find the shortest path between a source and a destination. In other words, similar to [60], the fastest path amongst all possible paths with the minimum hop count is selected. In another work, Hay *et al*. [70] propose a space-time graph to minimize delay and the number of hops. For a given delay t, they prune all edges that occur outside time t. Lastly, the Dijkstra algorithm is applied on the pruned space-time graph to find the shortest path.

Recently, Huang *et al*. [59] proposed a number of heuristics to construct an efficient space-time graph in deterministic DTNs where the network topology is known in advance or can be predicted. They build a weighted space-time graph that includes both spatial and temporal links to model a DTN topology. A spatial link is a directed edge between two nodes if they meet each other at a given time. Temporal links on the other hand capture the connection between the same nodes across consecutive time slots. Their approach aims to extract a graph from the original space-time graph such that (1) there is at least one route between any pair of nodes, (2) a route between any two nodes is cost efficient, and (3) the dense structure of the space-time graph is minimized. They propose the following heuristic algorithms: (i) *Union of Shortest Path algorithm (USP),* which finds the shortest path between any pair of nodes and constructs a sub-graph of the original space-time graph to route accordingly, (ii) *Greedy algorithm to Delete Links (GDL)*, which removes links from the original space-time graph in a descending order of link cost until only the route with the minimum cost exists between any pair of nodes, and (iii) *Greedy Algorithm to Add Links (GAL)*, which builds a full connected graph. Then, the algorithm finds the minimum cost path between any pair of nodes and adds the links to the built graph. Similarly in [58], Huang *et al*. propose a heuristic algorithm called *Greedy Algorithm based on Least Density Bunch* that considers all possible structures of connected pairs of nodes and the one with the smallest density is selected. Then, all edges in the selected bunch are added to a sub-graph. This procedure is repeated until at least one route is detected between any pair of nodes.

In [71], Liu *et al*. use the expected minimum delay as a new delivery probability metric in DTNs, where the mobility pattern of nodes is repetitive. In this case, they model the network as a probabilistic space-time graph using information from previous contacts. Then, in order to calculate the expected minimum delay of a bundle, they map the resulting graph to a probabilistic state-space graph, meaning that the time dimension is removed. Lastly, a Markovian decision process is applied to derive the expected minimum delay of messages.

For the space-time graph protocols described in [31, 58-60, 67, 70], every node has a fixed mobility pattern for an unspecified time period, meaning the space-time graph is not dynamic. Hence, the authors assume that the space-time graph is available in full at each node. Also, in both [62] and [71], all nodes are preloaded with a space-time graph and have a predictable mobility pattern, one that is repeated periodically or fixed for a given time period. As will be pointed out in Chapter 5, nodes may have a different mobility pattern within varying periods of time. In this case, as the future mobility pattern of nodes is unknown, the complete space-time graph cannot be pre-loaded at nodes. Hence, routing protocols have to consider the expiration time of each learned mobility pattern. This gives rise to a space-time graph with expiration time. Consequently, pre-loading a space-time graph at every node becomes impractical. Although in [62] nodes start with zero information and gradually learn the network topology, the employed routing algorithm will flood bundles throughout the network if a route is not present in the current space-time graph. This thus increases signalling overheads. Also, when a space-time graph is not complete, a detected route may not be optimal.

2.2 Queue management

Current buffer management schemes are categorized into two groups: (a) *Local Knowledge Schemes* [55, 72-78], and (b) *Global Knowledge Schemes* [79-89]. The following sections will review drop/forward policies in each category. Table 2-2 shows a taxonomy of all reviewed buffer management policies.

Table 2-2 A classification of related buffer management policies

Queue Management Policies	Drop Policy	Forward Policy	Required Knowledge	Non-Valid Information?	Consider Delivered Bundles?	Predict Bundle Delivery?	Global Information Collection?	Utility	Consider Finite Replicas?	Consider Meeting Rate?
MV [55]	No	Yes	Local	No	No	Yes	No	DV	No	Yes
T-drop [72]	Yes	No	Local	No	No	No	No	N/A	No	No
Zhang *et al*. [73]	Yes	No	Local	No	No	No	No	N/A	No	No
Lindgren *et al*. [74]	Yes	No	Local	No	Yes	Yes	No	DV	No	Yes
Pan *et al*. [75]	Yes	Yes	Local	No	Yes	Yes	No	DV	No	Yes
LPS and LRF [76]	Yes	No	Local	No	No	No	No	DV	No	Yes
Fathima *et al*. [77]	Yes	Yes	Local	No	No	No	No	N/A	No	Yes
Rohner *et al*. [78]	No	Yes	Local	No	No	Yes	No	DV	No	Yes
Pan *et al*.[83]	Yes	No	Global	Yes	Yes	Yes	Yes	DV,DL	No	Yes
Yin *et al*. [82]	Yes	Yes	Global	Yes	Yes	Yes	Yes	DV+DL+OV	No	Yes
PREP [89]	Yes	Yes	Global	Yes	No	Yes	Yes	DV, DL	No	Yes
Yong *et al*. [84]	Yes	No	Global	Yes	Yes	Yes	Yes	DV	No	Yes

Queue Management Policies	Drop Policy	Forward Policy	Required Knowledge	Non-Valid Information?	Consider Delivered Bundles?	Predict Bundle Delivery?	Global Information Collection?	Utility	Consider Finite Replicas?	Consider Meeting Rate?
Dohyung *et al*. [85]	Yes	No	Global	Yes	Yes	No	Yes	DV	No	No
Krifa *et al*. [80]	Yes	Yes	Global	Yes	Yes	Yes	Yes	DV, DL	No	Yes
RAPID [86]	Yes	Yes	Global	Yes	Yes	Yes	Yes	DV	No	Yes
Krifa *et al*. [81]	Yes	Yes	Global	Yes	Yes	Yes	No	DV, DL	No	Yes
Elwhishi *et al*. [79]	Yes	Yes	Global	Yes	Yes	Yes	No	DV, DL	No	Yes
Liu *et al*. [87]	Yes	Yes	Global	Yes	Yes	Yes	Yes	DV	No	No
Shin *et al*. [88]	Yes	Yes	Global	Yes	Yes	Yes	Yes	DV	No	No

2.2.1 Local knowledge schemes

To date, past works have considered classical buffer management policies such as Drop Oldest (DO), Drop Random (DR), Last Input First Output (LIFO) and First Input First Output (FIFO) for use in DTNs. In DO, a node drops the bundle with the shortest TTL. The assumption is that a bundle with a short TTL implies it has been in the network for a long time, and thus is likely to have been delivered. DR drops a bundle randomly. LIFO considers the arrival time of a bundle and drops the most recent bundle. In contrast, FIFO drops the bundle at the head of the queue, i.e., waited the longest. As long as the contact duration is sufficient to transmit all bundles, FIFO is a suitable policy. On the other hand, if the contact duration is limited, then FIFO fails because it does not provide any mechanism for preferential delivery or storing high priority messages. In [90], Dias *et al.* evaluated the impact of the said policies on the performance of two routing protocols: epidemic [43] and Spray and Wait [25]. However, a bundle may have a small TTL but has a high delivery probability. In this case, DO drops the bundle despite its high delivery probability.

In [73], Zhang *et al*. present the impact of finite buffer and short contact duration when using an epidemic routing protocol [43], and evaluated drop policies such as drop-head (drop oldest), drop-tail and drop-head high priority. For the drop-head policy, when a node receives a new bundle and its buffer is full, the node drops the oldest bundle. Using drop-tail, when the buffer of a node is full, the node will not accept any bundle. As for the last policy, (i) if a source bundle, one that is transmitted by a source vehicle, is sent to a node with a full buffer, the receiving node will first drop the oldest relayed bundle. Here, a 'relayed bundle' is one forwarded by a non source node. If there are bundles to be relayed, the node drops the oldest source bundle, (ii) if a relayed bundle is sent to a node with a full buffer, the receiving node drops the oldest relayed bundle and if there is no relayed bundle, the new relayed bundle is not accepted.

Recent work uses local knowledge in their forward/drop policies. For example, Naves *et al*. [76] propose two drop policies: Less Probable Spray (LPS) and Least

Recent Forward (LRF). In the former, a node uses the bundle delivery probability and estimates the number of replicas already disseminated to decide which bundle to drop. Hence, a node drops a bundle with the lowest delivery probability only if it has disseminated the minimum number of replicas. This minimum is set according to network characteristics such as connectivity degree and inter-contact time. On the other hand, LRS as its name implies, forwards the bundle that has not been forwarded over a certain period of time. In a similar work, Lindgren *et al*. [74] evaluated the following buffer management policies under the PROPHET [5] routing protocol: most forwarded first, most favourable first, DO, and least probable first. In the most forwarded first policy, bundles that have been forwarded the most are dropped. In the most favourable first policy, the bundle with the highest delivery probability is dropped. The least probable first drops the bundle with the lowest delivery probability. The problem with the most forwarded first policy is that it does not consider a bundle's life time, meaning a bundle with insufficient lifetime for delivery will not be dropped if the bundle has not been forwarded the most.

In another work, Burns *et al*. [55] propose Meets and Visits (MV), a scheme that learns the frequency of meetings between nodes and how often they visit a certain region. This information is used to rank each bundle according to the likelihood of delivering a bundle through a specific path. However, many bundles with the same destination may exist in a node's buffer. Hence, in this case, all of them have the same priority to be forwarded whereas their different TTL values can affect bundle delivery. In another work, Pan *et al.* [75] propose a comprehensive buffer management policy based on state information such as node ID, list of buffered bundles and the five nodes that have the highest encounter rate. During routing, for a given bundle, a sender determines whether encountered nodes have recently met the bundle's destination. If so, the sender forwards the bundle to these nodes. It then arranges bundles in ascending order based on hop-count and number of forwards. Bundles with a hop-count greater than a threshold as well as having a size that is larger or equal to the size of a newly received bundle are selected for dropping and are arranged in ascending order based on the number of forwards. Accordingly, a node drops the bundle that has been forwarded the most. In another drop policy, Ayub *et al*. [72] propose T-drop, a policy that considers the size of bundles during congestion. Specifically, by defining a threshold range, a bundle is dropped if its size is within said threshold.

In [77], Fathima *et al*. classify bundles based on three priority queues: high, medium and low. When a node's buffer is full, those with a low priority are dropped first followed by those with medium priority. Apart from that, they also consider the TTL value of bundles. Another condition is that nodes do not drop their own bundles. In a similar work, Rohner *et al*. [78] propose an ordering policy that uses a relevance score to determine whether there is a match between a node's interests and a bundle's metadata.

In the schemes discussed thus far, references [73, 90] have considered classical drop/forward policies to deal with limited bandwidth (short contact duration) and finite buffer (congestion). However, these policies have not considered the parameters that are relevant to bundle delivery such as number of replicas. Although references [74, 76] have considered using the number of replicas disseminated by a given node, it does not represent the total number of disseminated replicas globally. In [55] and [75], the authors take advantage of encounter rates to estimate the probability of delivery. However, similar to references [74, 76], they do not know how many replicas have been disseminated throughout a DTN. None of the local knowledge schemes proposed thus far consider the number of disseminated replicas and/or number of replicas that will be disseminated in the future. This information can be used to evaluate bundle delivery probability. However, under flooding based protocols, it is impractical to obtain this information in order to improve forwarding decisions.

2.2.2 Global knowledge schemes

This section will review global knowledge schemes and outline how they use the number of disseminated replicas and the number of nodes that have seen a given bundle. RAPID [86] is the first protocol that considers both buffer and bandwidth constraints. RAPID assigns a utility to each bundle. A bundle's utility measures its expected contribution in maximizing a metric such as delay. RAPID replicates bundles that lead to the highest increase in utility. A key limitation of RAPID is that in order to derive the utility of bundles, information about replicas has to be flooded throughout the network. This causes high overheads and due to delays, the propagated information may be obsolete when it reaches nodes. Also, their results show that whenever traffic increases, their meta-data channel consumes more bandwidth. This is undesirable because meta-data amplifies the effects of congestion by occupying precious buffer space. In another work [84], Yong *et al*. present a drop policy that uses the control channel in [86] to help vehicles obtain global network information such as transmission opportunities of bundles, node meeting times and duration. However, the forwarding issue is not addressed. In [85], Dohyung *et al*. propose a drop policy to minimize the impact of buffer overflow. When the buffer overflows, a node discards the bundle with the largest expected number of copies. That is, the authors assume that by retaining bundles with a small number of replicas, the delivery ratio will increase.

Krifa *et al*. [80] introduce a distributed algorithm to approximate the number of replicas, and number of nodes (excluding sources) that have seen a bundle *i* since its creation. This estimation is based on the number of buffered bundles that were created before bundle *i*. As a result, this algorithm is dependent on the dissemination rate of previous bundles. This means any change in topology will result in inaccurate/obsolete information, especially for newly generated bundles [79]. In a similar work to [80], Yin *et al.* [82] propose an Optimal Buffer Management (OBM) policy to optimize the sequence of bundles for forwarding/discarding. They use a multi-objective utility function that considers metrics such as delivery, delay and overhead concurrently. In another work, Pan *et al*. [83] combine two routing protocols: PROPHET [5], and binary Spray & Wait [25]. They calculate the contact probability as per PROPHET; namely Eq. (2.2), (2.3) and (2.4). Then, upon contact, if the probability of meeting the destination via an encountered node is higher than the sender node, half of the replicas are forwarded to the encountered node. In order to manage bundles when a node's buffer is full, they use the bundle utility in [80] to drop bundles with the lowest utility value. Moreover, if the last copy of a bundle is left at a sender and its utility is greater than a threshold, the last copy is forwarded.

Otherwise the copy will remain at the sender. However, similar to [80], this method suffers from obsolete/inaccurate information.

In a recent work [81], Krifa *et al*. propose a drop and forward policy that permits vehicles to gather global knowledge at different times. Hence, during contacts, vehicles flood information such as “a list of encountered vehicles” and “the state of each bundle carried by them” as a function of time. However, due to large delays, this information may take a long time to propagate to all nodes. The authors estimate the dissemination rate of a bundle based on the average dissemination rate of older bundles. However, the computed rate may have a large variance, causing errors when computing the resulting utility function. Elwhishi *et al*. [79] use the Markov chain model of [39] to predict the delay and delivery ratio under epidemic forwarding. However, as computing the stationary probabilities of the Markov chain incurs high computational complexity, they propose a forward/drop policy called Global History-based Prediction (GHP) that uses Ordinary Differential Equations (ODEs). The ODEs, which calculate the utility of each bundle, incorporate two global parameters: the number of bundle copies and the number of vehicles that have seen a bundle.

In [87], Liu *et al*. use a utility that estimates the total number of replicas and the dissemination speed of a bundle. Nodes update this information when they meet each other. During congestion and forwarding, nodes drop the bundle that has the maximum utility value, and forward those with the minimum utility value. Also, during forwarding, if the maximum utility of bundles in a sender’s queue is smaller than the minimum utility value of bundles in a receiver's node, the sender forwards all its bundles to the receiver. In addition, if the minimum utility value of bundles in a sender’s queue is greater than the maximum utility value of bundles in a receiver's node, the sender will only forward bundles if the receiver has free space. In a similar work to [87], Shin *et al*. [88] propose a forward/drop policy that uses i) for a given bundle, an estimate of the total number of replicas, in a DTN, and ii) for a given node, the number of replicas of a bundle it has replicated. Based on said parameters and the elapsed time since a bundle was generated, a per bundle delivery utility is calculated. Also, a per bundle delay utility is derived from parameters (i) and (ii) and the bundle's remaining life time.

Ramanathan *et al.* [89] propose PRioritized EPidemic scheme (PREP), a drop and forward policy for epidemic routing protocols. PREP prioritizes bundles based on source-destination cost and bundle expiry time. Here, cost is the average outage time of links on a path, and this information is flooded throughout a DTN and is used by the Dijkstra algorithm to compute the minimum source-destination cost. In their drop policy, a node with a full buffer first selects bundles that have a hop-count value greater than a threshold. Accordingly, selected bundles are sorted based on their cost to their intended destination and the bundle with the maximum cost is dropped first. In terms of transmission priority, if a bundle incurs a lower cost of delivery through an encountered node, the bundle with the longest remaining lifetime will be forwarded first. The main limitation of PREP is that it requires the link cost to be flooded. However, due to large delays and topological changes, the computed path cost may become dated quickly.

In summary, the aforementioned local and global policies, namely [55, 72-89], are designed for flooding protocols e.g., [5, 43]. This means they are allowed to replicate a bundle without any limit. However, under quota based protocols, if a replica is dropped, the bundle will have one less copy. This may reduce the probability of delivery. Although many schemes, e.g., [74, 75, 79-88], have considered the number of disseminated replicas to estimate the delivery probability, they do not take into consideration the remaining number of replicas that nodes are permitted to replicate. Moreover, if we use a flooding protocol, buffer management is exacerbated by the difficulty in obtaining global knowledge of bundles and other nodes. For example, prior works [74, 75, 79-88] consider a bundle with a larger number of disseminated replicas to have a higher chance to be delivered. However, due to large delays, collected information may become obsolete. References [79, 81] address this problem by approximating the required information via a Gaussian distribution. However, the resulting estimates are not accurate under different forwarding strategies.

2.3 Summary

This chapter has reviewed both flooding and quota protocols. Although there have been extensive studies, little progress has been made to find a trade-off between delivery delay, delivery ratio and overhead simultaneously. As an example, EBR [30] reduces overheads by limiting the number of replicas. Although EBR works better than current well-known routing protocols, it does not work efficiently if a destination node is not located in a high density area. As another example, PROPHET [5] targets the nodes that have encountered a destination. However, under PROPHET, nodes can generate an unlimited number of replicas. This causes network overheads to increase. In addition, when nodes have a limited buffer size, the number of dropped bundles increases. In turn, this affects delivery ratio and delay. In the next chapter, a novel investigation will be carried out to determine the efficacy of forwarding bundles only to nodes that have had contacts with the destination of a bundle regardless of its encounter rate with other nodes. As we will see, the resulting protocol has a higher delivery ratio than competing approaches.

From studies that consider predictable or scheduled mobility patterns, we see that nodes are able to route bundles efficiently toward their destination. As mentioned, current space-time graph protocols assume every node has a complete knowledge of the network topology. However, in some scenarios the mobility pattern of nodes may not be predictable in advance or is only valid for a short period of time; e.g., a taxi ferrying passengers to a given destination. This causes the space-time graph to be staled as it contains node trajectories that are no longer valid. Hence, if a route is not discovered for any generated bundles, they will be held at sources. In this case, current routing protocols may be used until every node constructs its complete space-time graph. However, these routing protocols do not take advantage of any available trajectory information that nodes have learned thus far. In particular, dynamic protocols such as flooding or quota based protocols do not use trajectory information. History based routing protocols assume nodes have some relationship with each other. In the case where nodes have independent or dependent mobility pattern for a short period of time, previous encounters may not be indicative of future contacts. Apart from that, one can use protocols such as [36, 56, 57] that take advantage of recorded mobility patterns to evaluate nodes based on how close they are to a given destination. However, they assume that every node is aware of the destination's mobility pattern. According to the aforementioned gaps, Chapter 5 will address the problem of routing in semi-predictable DTNs where contacts are not completely predictable. Moreover, it will propose heuristics that make use of available, but incomplete, space-time graph at each node.

Lastly, this chapter has provided a comprehensive literature review of current buffer management policies where nodes use local and/or global knowledge. Although current local knowledge schemes are of low complexity in terms of computation, they are inefficient and do not make full use of the following fact. They disregard the number of disseminated replicas. This is a key parameter that has non negligible impact on delivery ratio and delay. However, flooding this global information throughout the network imposes a high overhead. In addition, due to large delays, collected information may become obsolete. In Chapter 4, this thesis will address the problem of buffer management in quota based protocols by taking advantage of both local and global information. The main objective is to manage bundles in terms of routing and queues such that delivery delay and overhead are minimized and delivery ratio is maximized.

Chapter 3

A Novel Destination Based Routing Protocol (DBRP)

3.1 Introduction

The previous chapter has reviewed both flooding and quota protocols. This chapter addresses the routing problem when network resources are limited. In this case, the aim of any forwarding/routing protocol for DTNs is to achieve a high delivery ratio of packets/bundles using the lowest possible bandwidth cost, buffer space and energy. As indicated in Chapter 2, one key approach is to flood bundles to increase the probability of delivery. However, such protocols can cause high overheads and large delays due to a high rate of dropped bundles when network resources are limited. To address this problem, quota protocols limit the number of replicas for each generated bundle. However, quota protocols cannot efficiently deliver a message as their bundle dissemination rate is low.

In order to solve this issue, this chapter investigates the hypothesis that a targeted forwarding strategy based on contact history with a destination improves bundle delivery when there are a finite number of replicas. This hypothesis is first verified using a time homogeneous semi-Markov process (THSMP). Then, in Section 3.3, a destination-based routing protocol (DBRP) is proposed to take advantage of this hypothesis. Specifically, DBRP is a quota protocol that weights nodes that have had any encounters with the final destination higher than any other node encounters. In fact, the proposed method takes advantage of the following observation**.** Consider person *A*, who goes to work and meets person *C* every day (and this meeting may only be brief). This means person *A* is an ideal bundle carrier for person *C* because delivery is guaranteed (it may take long time but it is guaranteed nonetheless). It should also be noted that person *A* may meet many other people for much longer periods, and hence these carriers may seem to be better options to pass the data to as they seem more active. The hypothesis here is that it is much better to weigh person *A*'s connection to person *C* higher than other contacts even though a person may have high encounter rates with people other than *C*. This hypothesis is inspired by recent studies [91, 92] on the characteristics of human mobility from real world traces. They demonstrate that people usually roam in relatively small regions. Hence, this fact is based on the idea that regardless of how small an encounter rate with the destination is, given a highly correlated movement model, e.g., human behaviour, we will end up with a high delivery ratio. Simulation studies presented in Section 3.5 over three scenarios show that in terms of a composite metric comprising delivery, delay and overhead, DBRP achieves up to 57% improvement over three well-known routing protocols, namely PROPHET, EBR and Spray and Wait. Moreover, DBRP results in nodes experiencing at least 28% lower buffer consumption.

The remainder of this chapter is organized as follows. Section 3.2 provides a theoretical formulation to analyse nodes' encounters and the delivery probability of bundles. Section 3.3 presents DBRP, a routing protocol that exploits said observation. Section 3.4 describes the simulation set-up. This is then followed by experimental results in Section 3.5. Finally, Section 3.6 concludes this chapter.

3.2 Motivation

This section now proves the assertion that nodes with any contact history (regardless how small) with a destination make good forwarders. First the following terms should be defined precisely. *Contact probability* is the chance that two nodes will come into each other's radio range during a time unit. A time unit is a fixed discrete period of time. *Delivery probability* is the likelihood that a bundle will be delivered to its intended destination. The following assumptions are made only in the theoretical framework:

1. The trajectory of nodes is known in advance.
2. The network topology is at least semi-predictable.
3. Each node has sufficient buffer to receive all bundles at each contact.
4. Bundles have unlimited lifetime.
5. Two nodes can communicate if they are in the same geo location.
6. Nodes have equal speed.
7. The duration of contacts is long enough for transferring all queued bundles.
8. Time is discrete.

3.2.1 Preliminaries

To verify the hypothesis stated in Section 3.1, this thesis uses a Time Homogeneous Semi-Markov Process (THSMP), a discrete time, stochastic process with the Markov property for which the transition probabilities are time-homogeneous [93, 94]. A THSMP is defined by (i) its system states, (ii) residence time at each state, (iii) transition probabilities between states, and (iiii) kernel, which describes the probability of being in a state at a specific time. This section will define these aspects more precisely in order to characterize the movement of nodes on the grid, as shown in Figure 3-1.

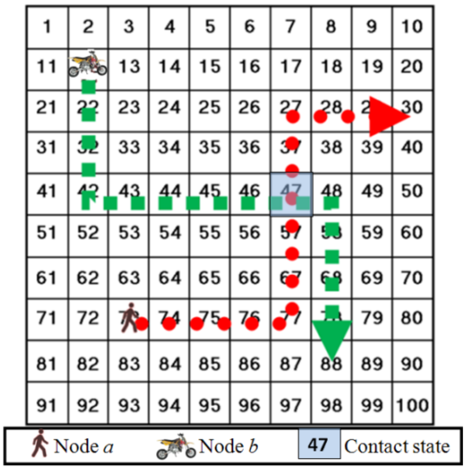


Figure 3-1 Sample paths on a grid

Let S be the set of all states with cardinality $m=|S|$ and S_n be the state of the n-*th* transition from the current state and T_n be the time of the n-*th* transition. Here, states correspond to squares of a grid. A THSMP is defined by the tuple $\{(S_n,\ T_n) \mid n \geq 0\}$. Assuming n=0 to be the initial transition, denote the probability that an arbitrary node a will be in state j after t time units after having started from state i as $P(X_t = j|X_0 = i)$, where $X_t \in S$ is the node state at time t. Let the path (or states) followed by node a be X^a, and a sub-path/states from 1 to t time units be $X^a_{1,t}$= $\{X_1,\ldots,X_t\}$. As an example, in Fig. 1, there are 100 states (m=100). If node a is assumed to have a transition in each time unit, then the dotted line represents the path for node a up to t time units, where $X^a_{1,t} = \{73, 74, 75, 76, 77, 67, 57, 47, 37, 27, 28, 29, 30\}$, with t=13.

As a system enters a state i, it stays there for a time called *residence or sojourn time*; i.e., the time between transition S_n and S_{n+1}. Let $T_{n+1} - T_n$ be the residence time, which is obtained through a cumulative distribution function (CDF). The probability that the residence time will be less than or equal to t time units is defined as

$$H_{i,j}(t) = P(T_{n+1} - T_n \leq t|S_{n+1} = j, S_n = i) \qquad (3.1)$$

Let $P_{ij} = P(S_{n+1} = j|S_n = i)$ be the transition probability that a node moves from state i to j. Here, P_{ij} is a matrix with row i indicating the current state of a node and column j indicating its next state. The probability of a transition depends on the mobility model. For example, under a map-based mobility [95], where paths are predefined, nodes move to predetermined states. Hence, the probability of moving from a node's current state to a predetermined neighbor state is one, and to the rest of its neighbours is zero. On the other hand, under a random mobility model where nodes can move to any neighbouring states, the probability of each transition is dependent on the number of neighbours. For example in Fig. 3-1, if a node is in state 45, then its four neighbours are {35, 44, 46, 55}. That is, if a node has four neighbours, its transition probability to each neighbor is 0.25. At any time, the sum of the probabilities of moving into neighbouring states is equal to 1, hence if a node goes over the same state more than once during its movement, the transition probability is updated based on the new movement. For example, assume node *a* has moved from state 27 to 28 at time 15 with $P_{27,28} = 1$. Now, at time 30, node *a* reaches state 27 but its next movement is state 26. Hence, the previous transition probability is set to zero ($P_{27,28} = 0$) and its new transition is set to one ($P_{27,26} = 1$). As a result, the following condition is applied: $\sum_{j \in R_i} P_{i,j} = 1$, where R_i indicates the neighbor states of state *i*.

The next step is to derive the probability that a node moves from state *i* to *j* in *t* time units. This thesis uses the kernel of the THSMP, which describes the probability of being in state *j* in the next transition within time *t*. The kernel is defined as

$$Q_{i,j}(t) = P(S_{n+1} = j, T_{n+1} - T_n \leq t | S_n = i) = P_{ij} \times H_{ij}(t) \tag{3.2}$$

In other words, by the Markovian property, only a node's current state and its residence time is considered when determining its next transition to state *j* within time *t*. Hence, the probability that a node will be in state *j* next is determined by the probability that it will transition from state *i* to *j*, and the probability that the residence time is within *t*. At steady state,

$$lim_{t \to \infty} Q_{i,j}(t) = P(S_{n+1} = j | S_n = i) = P_{ij} \tag{3.3}$$

The residence time is modelled irrespective of the next state by defining $D_i(t)$ as $P(T_{n+1} - T_n \leq t | \, S_n = i)$. This is the probability that a node will leave state *i* in time *t* independent of its next state. It is computed as

$$D_i(t) = \sum_{j=1}^{m} Q_{i,j}(t) \tag{3.4}$$

Now, assuming t=0 to be the current time. Let us define $\Psi_{i,j}(t) = P(X_t = j|X_0 = i)$ as the probability that a node will have a transition from state *i* to state *j* at time unit *t*. Let $\Psi_{i,j}(0) = \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker delta and its value is one if $i = j$, otherwise it is zero. If a node stays in state *i* between time 0 and *t* without transitioning, then

$$\begin{aligned} &P(X_t = i|X_0 = i \; and \; T_1 - T_0 > t) \\ &= P(T_1 - T_0 > t|\, S_0 = i) \\ &= (1 - D_i(t)) \end{aligned} \tag{3.5}$$

where $T_1 - T_0$ represents the residence time before the first transition. On the other hand, if a node experiences at least one transition at time *k* between time 0 and *t*

$$\begin{aligned} &P(X_t = j|X_0 = i \; and \; at \; least \; one \; transition) \\ &= \sum_{l=1}^{m} \sum_{k=1}^{t-1} \frac{\partial Q_{i,l}(k)}{\partial k} \times \Psi_{l,j}(t-k) \end{aligned} \tag{3.6}$$

where $\frac{\partial Q_{i,l}(k)}{\partial k} = Q_{i,l}(k) - Q_{i,l}(k-1)$. Hence, the probability of moving from state *i* to *j* in *t* time units is

$$\begin{aligned} &\Psi_{i,j}(t) = \big(1 - D_i(t)\big) \times \\ &\delta_{i,j} + \sum_{l=1}^{m} \sum_{k=1}^{t} (Q_{i,l}(k) - Q_{i,l}(k-1)) \times \Psi_{l,j}(t-k) \end{aligned} \tag{3.7}$$

Thus far, the probability of being in each state is determined based on the transition probability and residence time. Next, the probability of contact between two nodes, say between node *a* and *d*, at time *t* is determined according to their common states. That is, a node meets another node by crossing the same state at the same time. The probability of contact for nodes *a* and *d* is obtained as follows

$$P_{a,d}(t) = \sum_{\mathcal{L} \in X^a \cap X^d} \Psi_{X_0^a,\ \mathcal{L}}(t) \ \times\ \sum_{\mathcal{L} \in X^a \cap X^d} \Psi_{X_0^d,\ \mathcal{L}}(t) \tag{3.8}$$

where X_0^a and X_0^d are the respective nodes' current state. Without loss of generality, this chapter will denote the current time as t=0. The element $\mathcal{L}$ is a member of the common states set that belongs to two nodes' states. In other words, Equation (3.8) considers the common states of two nodes and calculates the probability of both nodes being in these states at time *t*. Table 3-1 contains the notations used in the discussion of the theoretical framework.

Table 3-1 Notations

Notation	Description
S	Set of all squares on the grid
$X_{<a,(0,t)}$	A subset of S indicating the path followed by node *a* from its current state $X_{<a,0>}$ (time 0) up to time *t*, denoted as $X_{<a,t>}$
X^a	The path taken by node *a*
T_n	Time of the n^{th} transition
$T_n - T_{n-1}$	Residence time for the n^{th} transition
P_{ij}	Probability of transition from *i* to *j*
S_n	The n^{th} transition
$X_{<a,0>}$	Current states of node *a*
$X^a \cap X^d$	Common states of nodes *a* and *d*
$\mathcal{L}$	The common states of two nodes

3.2.2 Delivery Probability

Given the above framework, we are now ready to calculate the delivery probability according to the approach used in [96]. The aim is to study how bundles propagate from one node to another given their contact profile. Unlike [96], where they consider unlimited replicas, in this work, the number of replicas is limited and is affected by the following two factors: (i) available replicas, and (ii) contact schedule.

The contact probability given by Equation (3.8) helps us to find the (first) contact at time *t* between nodes *a* and *b* with the given probability $P_{a,b}(t)$. Since the probability of the first contact $d_{a,b}(t)$ at time *t* is the probability of meeting at time unit *t* and the probability not to meet at time units *0, 1, . . . , t−1*. Therefore, the probability of the first contacts at time *t* is calculated as follows:

$$d_{a,b}(t) = P_{a,b}(t) \prod_{i=0}^{t-1} \left(1 - P_{a,b}(i)\right) \quad (3.9)$$

Let $D_{a,b}(T,t)$ be the probability distribution that nodes *a* and *b* require a delay of *t* time steps to meet for the first time after time step *T*. This distribution allows us to compute when a bundle can be delivered to its destination. Mathematically, $D_{a,b}(T,t)$ is calculated as follows

$$D_{a,b}(T,t) = P_{a,b}(T + t) \prod_{i=T}^{T+t-1} \left(1 - P_{a,b}(i)\right) \quad (3.10)$$

Let $D_{s,b,d}$ be the delivery distribution as a bundle from a source node *s* reaches destination *d* via node *b*. More precisely, if *s* decides to send a bundle at time T, it will reach node *d* after a delay that $D_{s,b,d}(T,.)$ indicates delay distribution. $D_{s,b,d}$ can be presented with respect to $D_{s,b}$ and $D_{b,d}$ as

$$D_{s,b,d} \equiv D_{s,b} \otimes D_{b,d} \quad (3.11)$$

The forwarding operator ⊗ is defined to incorporate the probability distribution of intermediate nodes. Therefore, a bundle could be forwarded through several intermediate nodes before reaching its destination. Specifically, the forwarding operator is applied on two distribution pairs as follows:

$$(D_1 \otimes D_2)(T,t) = \int_0^t D_1(T,x)\, D_2(T + x, t - x)\, dx \quad (3.12)$$

Therefore, the total delivery delay is equal to *t* if the delay to reach node *b* is equal to *x* $(0 \leq x \leq t)$, then the delay from node *b* to node *d* is *t-x*. For example, if node *s* encounters node *b* at time unit 15 (*x*=15) and node *b* meets node *d* at time unit 19, then node *s* can deliver a bundle through node *b* to destination *d* within 19 time units (*t*= 19). Hence, the first hop takes 15 time units and the second hop takes 4 time units (t-x=4).

3.2.3 Simulation and Analysis

This section outlines two objectives: to verify the proposed model by comparing it to a simulated network and to test the hypothesis that forwarding replicas to nodes that have had contact with the destination even briefly results in the highest delivery ratio. The probability of contact between nodes in a simulated network is used to verify the proposed mode. Also studied is the impact of destination contact probability on the total delivery probability.

The following network is simulated. Suppose that vehicles move along predetermined paths with a constant speed of 7m/s in the area of 4.8×4.8 km^2 that is overlaid on a grid size of 6×6. This grid size makes 36 geographical states (*m=36)*. The simulation lasts for an hour resulting in 30 discrete time units. In terms of mobility pattern, each node travels on a shortest path trajectory towards the point of interest and hence, during this period, motion is not random. This mode is referred to as the 'shortest map-based' model. During the simulation for each node i, its path X_i is extracted. Accordingly, the transition probability matrix is built such that if a node moves from its current state to another state, the corresponding element is set to one, otherwise, zero. Given that the path of the node is directed and non-random, the probability of transition is one along the path from one state to the next.

From the simulation, each node's position is sampled at every discrete time unit; i.e., 120 seconds. This yields in which time unit a transition was made by each node and also the residence time in each state. Figure 3-2 shows the probability of contact between random pairs of nodes. For example, Figure 3-2 (a) shows the contact probability between nodes a and b for each time unit. One line shows the contact probability as calculated from the model and the other shows the measured probability from the simulation. The figure shows that there is a high degree of correlation between the actual and predicted results for all cases. The reason for the shift between the two lines is because of the non-precise residence time used in

Equation (3.1). This is sampled from the simulation. Hence, using the exact residence time of being in a state, the predicted contact will overlap directly with simulated contact without any shift. This shows that the proposed model is an accurate representation of the system under consideration.

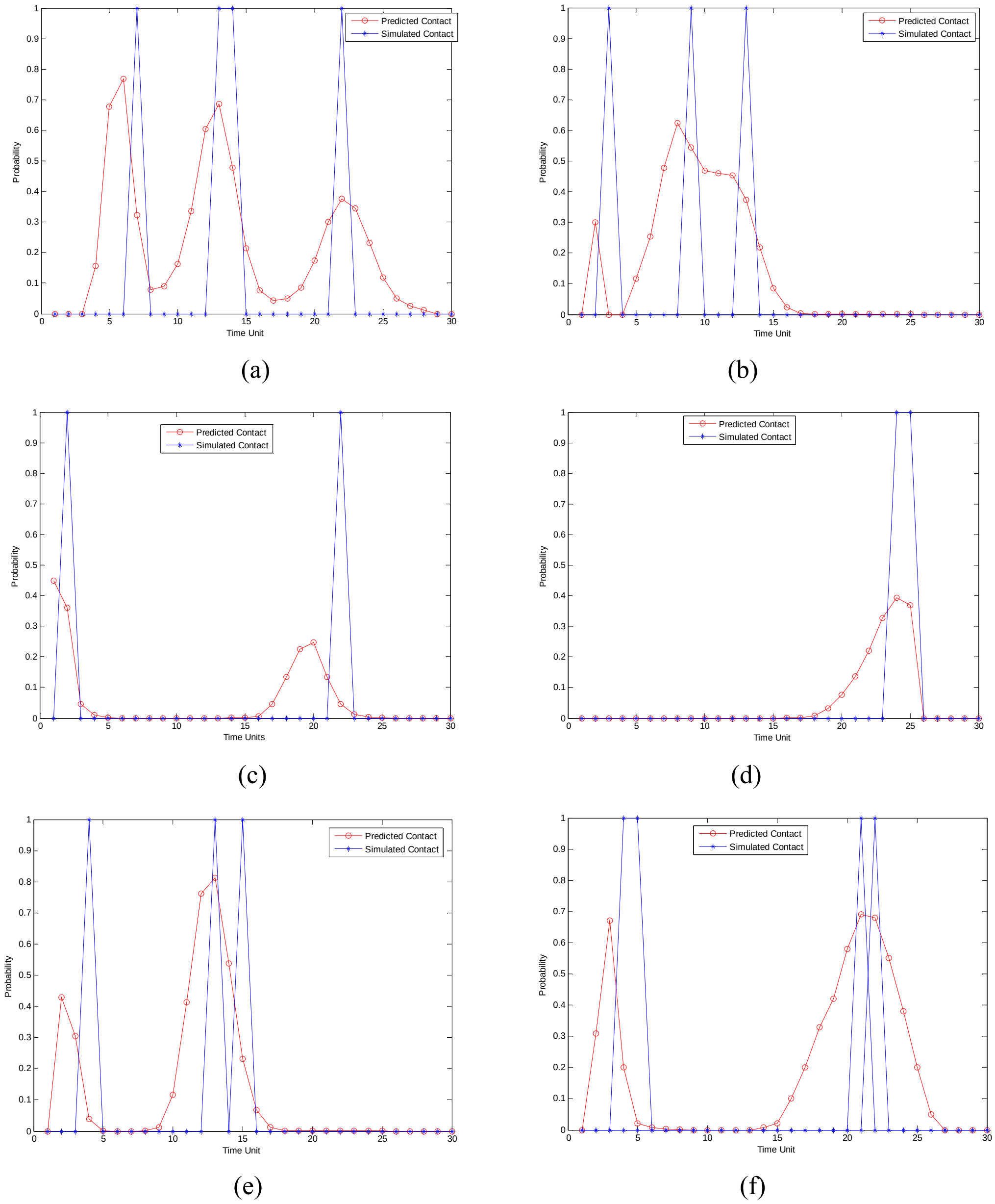


Figure 3-2 Probability of contacts using Markov model for different pair of nodes within an hour.

In order to study the shift in probabilities further, the following example gradually shows more and more accurate residence time distributions and compares them to the simulated outcomes. Figure 3-3 shows the evolution of contact prediction using different residence time distributions. In the worst case, Figure 3-3 (a) shows that the correlation between the model and a simulated contact is zero as the residence time is not accurate. As shown Figure 3-3 (b)(c)(d)(e), as the residence time becomes more accurate, the correlation of contact determination increases to one.

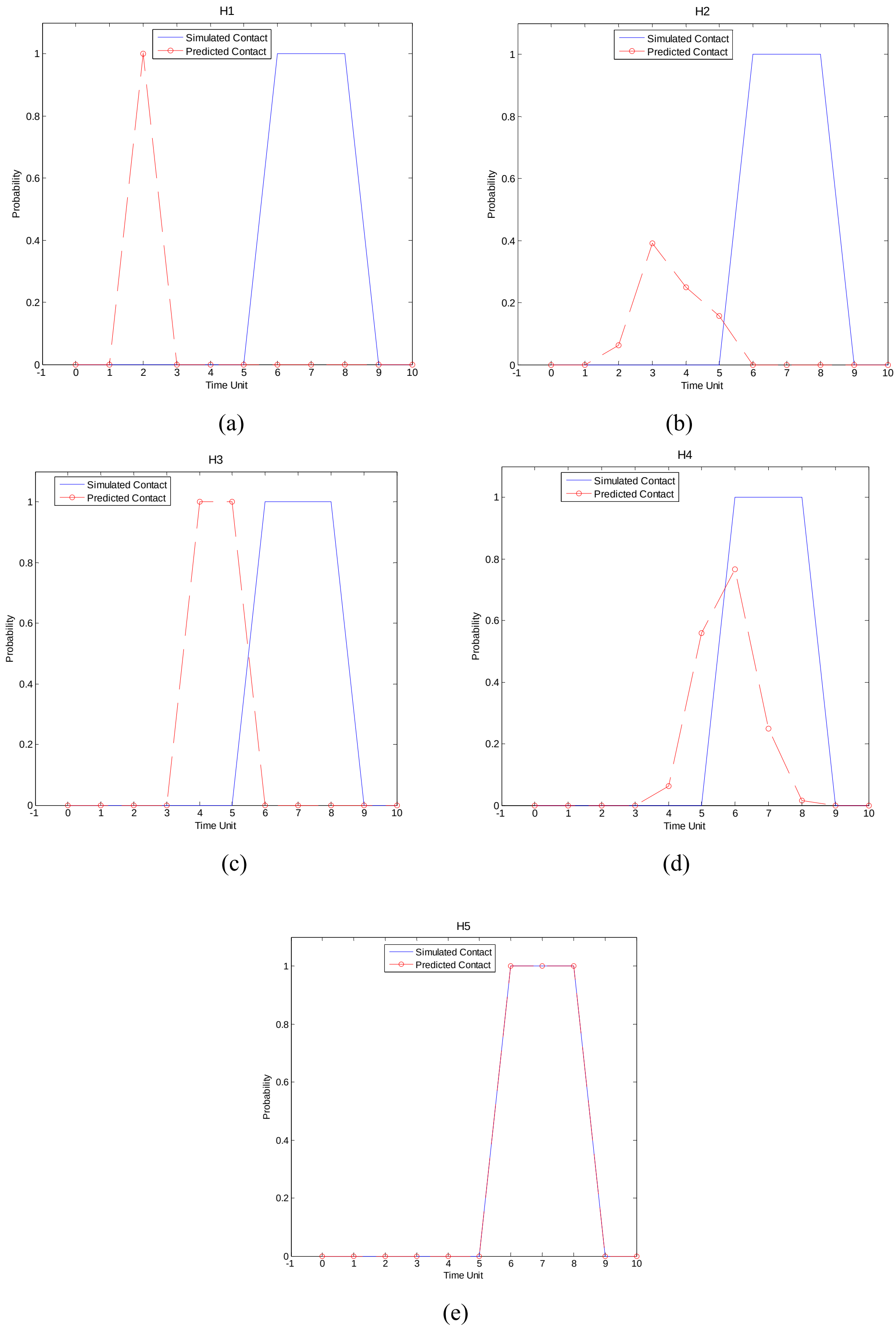


Figure 3-3 The impact of different residence time on contact prediction.

It should be pointed out here that determining contact probabilities under random mobility is impossible because by definition, all movement is random and hence any probability of meeting another node will also be random. In contrast, under map-based mobility, predetermined paths result in a high degree of correlation between the actual and predicted results. In other words, if accurate knowledge of nodes' residence time is available, this model knows exactly when and for how long contacts occur and the maximum degree of correlation is achieved.

When implementing a practical system, it is difficult to maintain a probability distribution of node encounters for each node. In most cases, only the encounter rate with certain nodes can be maintained for a period of time. In order to maximise the delivery probability, a node with replicas to pass on needs to identify those nodes (if any) that it encounters frequently. This is so that the node can select them to receive the replicas and to decide how many of the replicas to send. Indeed, these encounter rates between nodes now need to be translated into the number of replicas to pass on in order to maximise the delivery rate.

From these results, a key observation is that if the movement of nodes is correlated, then even if there is only one encounter with the final destination then this will result in the delivery of the bundle. On the other hand, by relying only on the measured rates of encounters, then this single encounter will be buried under many higher encounter rates with nodes that may or may not meet the final destination. Hence, this chapter hypothesizes that any encounter rate with the final destination needs to be weighted regardless of how small the rate may be in order to pass as many replicas to it as possible.

This means that if a contact happens between a node and a destination in a given period of time, the probability of delivery through that node is maximum. Let D_{s-d}^{i} be the probability of delivering a replica of a given bundle through route *i*. Hence, if nodes flood replicas through all possible routes, the delivery probability (DP) is calculated as

$$DP = max_i(D_{s-d}^{i}) \tag{3.13}$$

Now, suppose that over route *i*, sender node *a* has a message and meets node *b*. Node *a* detects that node *b* has met the destination, but the rate may be very small, as node *b* only meets the destination very rarely. So, the probability that node *b* meet the destination is one if time approaches infinity. Accordingly, if node *a* forwards the message to node *b*, the maximum delivery probability will be achieved if time approaches infinity. Motivated by the above results, the following section proposes a routing algorithm for use under quota-based protocols.

3.3 Destination Based Routing Protocol (DBRP)

DBRP is a quota-based routing protocol that limits the number of replicas for each generated bundle in order to achieve low overhead ratio. A sender forwards only a portion of replicas to the receiver. This strategy is based on the rate of encounters that the sender and receiver have had with the destination and other nodes. According to the previously proposed model, the nodes' movement based on the predefined path is defined and a random distribution for sojourn time is applied on the model. DBRP uses the history of encounters rather than any knowledge about the predefined path. Based on the history of encounters, nodes can predict how likely it is that they will encounter each other. Since DBRP gives a higher weight to nodes that have encountered the destination. In the case of high node density areas where nodes have high encounter rates, DBRP ensures all nodes with contact to the destination receive a significantly higher weight.

3.3.1 Algorithm

In DBRP, every node *a* establishes a metric called the encounter history, $en\_His_{(a,b)}$, for each destination b. This metric is obtained through the combination of two counters: $en_{(a)}$, for counting the number of times that *a* encounters other nodes and $en_{(a,b)}$, which counts the number of times *a* has met *b*. This encounter history is much more informative than an absolute number of encounters. If routing protocols simply rely on the number of encounters, the forwarding strategy can be ineffective because a node with a high encounter frequency may never meet the target destination. Therefore, encounter history as used in DBRP indicates a rough prediction of the future rate of encountering a destination node.

The encounter history, *en_His*, for a node to any other node in a given time interval is calculated as follows:

$$en\_His_{new(a,b)} = \beta \times en_{(a)}^{\gamma \times en_{(a,b)}} + (1 - \beta) \times en\_His_{curr(a,b)} \tag{3.14}$$

where $0<\beta<1$ is the weight of the most recent encounter information. The variable $en_{(a)}$ is the total number of encounters that node *a* has had over a specific time interval with all nodes. The variable $en_{(a,b)}$ represents only the encounters between nodes *a* and *b*. Hence, if this variable is zero, then this node has never encountered destination *b* in a given time interval. The term 'time interval' is used to consider the network parameters in time slices. For example, in a time interval a node may have 20 encounters with different nodes and in the next interval, 10 encounters. Therefore, we can evaluate the rate of encounters in each interval. In this algorithm, the time interval is set to 1000 seconds. The proposed algorithm uses a larger interval than that used in EBR [30] because in small time intervals the destination may be encountered only a few times or no times at all. In Equation (3.14), DBRP exponentially weights the encounter rate. The variable $\gamma > 0$ is a weight function. Meanwhile, $en\_His_{curr(a,b)}$ is the value of $en\_His_{(a,b)}$ before an update and $en\_His_{new(a,b)}$ is the new value after the update.

As an example, consider node *A* which has four encounters out of 10 with node *B*, two with node *C*, one with node *D* and three with node *E*. The encounter history for node *A* is computed as follows (assuming β =0.85 and γ=1.4):

$$en\_His_{new(A,B)} = 0.85 \times 10^{1.4\times4} + (1 - 0.85) \times 0 = 338390 \tag{3.15}$$

$$en\_His_{new(A,C)} = 0.85 \times 10^{1.4\times2} + (1 - 0.85) \times 0 = 536.3 \tag{3.16}$$

$$en\_His_{new(A,D)} = 0.85 \times 10^{1.4\times1} + (1 - 0.85) \times 0 = 21.35 \tag{3.17}$$

$$en\_His_{new(A,E)} = 0.85 \times 10^{1.4\times 3} + (1 - 0.85) \times 0 = 13472 \tag{3.18}$$

This example shows the encounter history of node A with the four destinations. Therefore, a node that frequently encounters A gets a higher weight. Here, node A has encountered node B four times and node C two times, whereas their encounter history shows that node A has visited node B $\frac{338390}{536.3} = 630$ times more than node C.

The number of replicas is dependent on the encounter history of the sender and receiver. Specifically, the number of replicas is proportional to the ratio of the encounter history of the nodes. For two nodes a and b, for i^{th} bundle M_i, that is headed to destination d, node a sends:

$$m_i \times \frac{en\_His_{(b,d)}}{en\_His_{(b,d)} + \eta \times en\_His_{(a,d)}} \tag{3.19}$$

replicas of M_i, where m_i is the available number of replicas for the i^{th} bundle at node a, and η is a scaling factor. When sender a has encountered destination d frequently, it means the bundle can be delivered through the sender. Therefore, it is better for node a to give more opportunities to receiver b to receive more replicas. This means at each contact, when node a has a high encounter rate with d, there is no need to keep the large number of replicas for itself. This is due to node a having a better chance to directly deliver the bundle even with only one copy. As a result, η is used to decrease the effect of the original sender's $en\_His_{(a,d)}$ in forwarding replicas. Here, the values of beta, gamma and eta are determined heuristically. The values were chosen to provide the greatest discrepancy in weight values between the final destination and other nodes.

For example, assume node a has eight replicas of bundle m_1 with destination d and nine replicas of bundle m_2 with destination z. Furthermore, assume node a, with $en\_His_{(a,d)}$ = 2000 and $en\_His_{(a,z)}$ = 5500 comes in contact with node B, with $en\_His_{(b,d)}$ = 5000 and $en\_His_{(b,z)}$ = 2500. Node a sends $\frac{5000}{5000 + 0.6\times 2000} = \frac{50}{62}$ of the replicas of bundle m_1 and $\frac{2500}{2500+0.6\times 5500}=\frac{25}{58}$ of the replicas of bundle m_2. Therefore, Node *a* forwards six replicas of bundle m_1 and three replicas of bundle m_2.

3.4 Research Methodology

The Opportunistic Network Environment (ONE) [95] is a Java-based simulator that is able to generate node movement using different mobility models. ONE can import mobility data from real-world traces or other mobility generators. The performance of DBRP is evaluated using ONE simulator over the map-based model [95]. In this model, nodes have predefined movement in an area of approximately 5×3 km^2 of downtown Helsinki, Finland. In addition, a majority of these nodes are pedestrian. Specifically, ONE's default settings are used, whereby 64% of nodes model pedestrians that follow the shortest path from their current location to a random chosen point with speed between 0.5 and 1.5 m/s. Another 32% of nodes are vehicles that have the same movement but with speed ranging from 2.7 and 13.9 m/s. The remaining nodes are configured to follow pre-defined routes (like tram lines) with speed between 7 and 10 m/s. All nodes have a transmission range of 20m except trams that have a 200m range.

The number of nodes is varied from 50 to 200 in increments of 50 but the number of source and destination pairs is fixed to 50. Also the offered load is varied by adjusting the time between generated bundles from 10 seconds (high load), to 30 seconds (medium load), to 60 seconds (light load). In another experiment, the behaviour of the protocols is considered when nodes have infinite buffer space. In all simulations, bundles have unlimited lifetime and their size is 25 KB. Each node has one MB buffer space, and all nodes have a transmission speed of 250 kBps. Each simulation lasts for 12 simulated hours and each data point is an average of 10 runs, with 95% confidence intervals. Note, in each run, random seeds are used.

To illustrate the performance of each protocol, this thesis evaluates DBRP against three other well known protocols with respect to node density and load: (1) PROPHET [5], (2) Spray and Wait [25], and (3) EBR [30].

The metrics collected are as follows:

- Delivery ratio is defined as the ratio of the Number of Delivered Bundle (NDB) to the Number of Generated Bundles (NGB),

$$Delivery\ ratio = \frac{NDB}{NGB} \tag{3.20}$$

- Equation (3.21) defines the average delay of all delivered bundles, where *t* is the delay experienced by bundle *i*:

$$Average\ latency = \frac{\sum_{i=1}^{NDB} t_i}{NDB} \tag{3.21}$$

- Equation (3.22) defines the ratio of NDB and Number of Relayed Nodes (NRN).

$$Overhead = \frac{NRN\text{-}NDB}{NDB} \tag{3.22}$$

In DTNs viewing delay and overhead in isolation from each other may lead to erroneous conclusions, since many protocols quickly deliver bundles that take a small number of hops, and do not deliver most bundles that require a high number of hops. To overcome this issue, the experiments use composite metrics to incorporate delivery ratio and other metrics:

- Equation (3.23) defines DL based on Delivery Ratio (DR) and Latency Average (LA).

$$DL = DR \times \frac{1}{LA} \tag{3.23}$$

- Equation (3.24) defines DO based on DR and Overhead Ratio (OR).

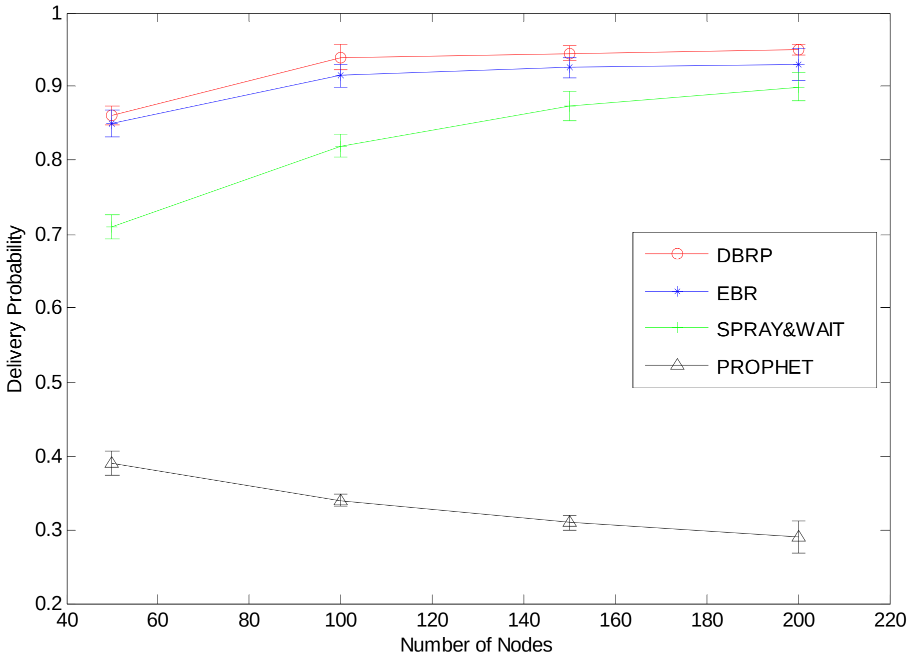
$$DO = DR \times \frac{1}{OR} \tag{3.24}$$

- Equation (3.25) defines DLO based on DR, LA and OR.

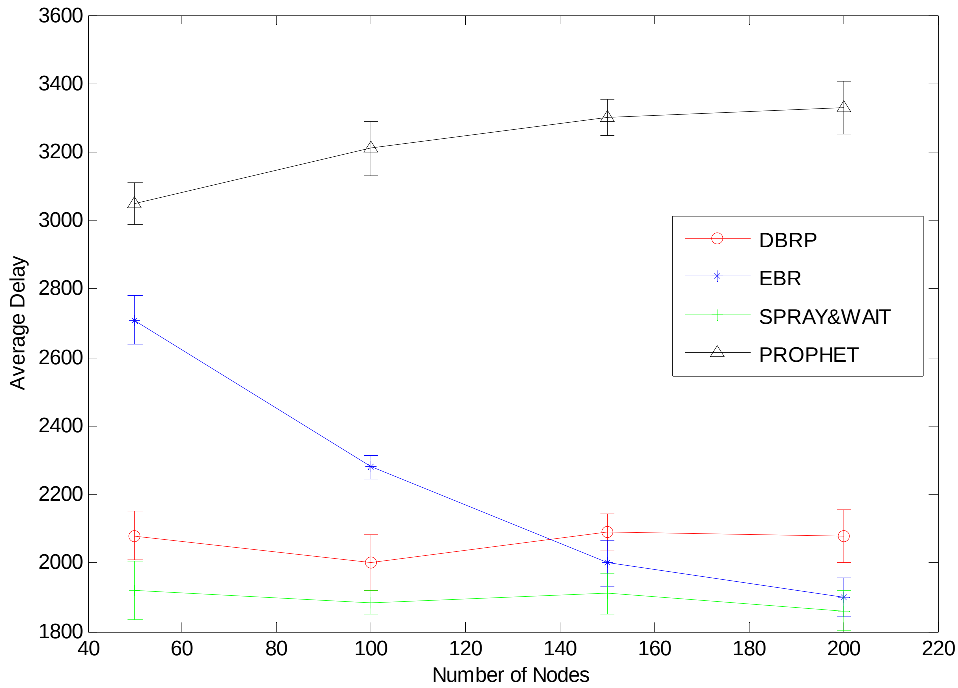
$$DLO=DR \times \frac{1}{LA} \times \frac{1}{OR} \qquad (3.25)$$

3.5 Results

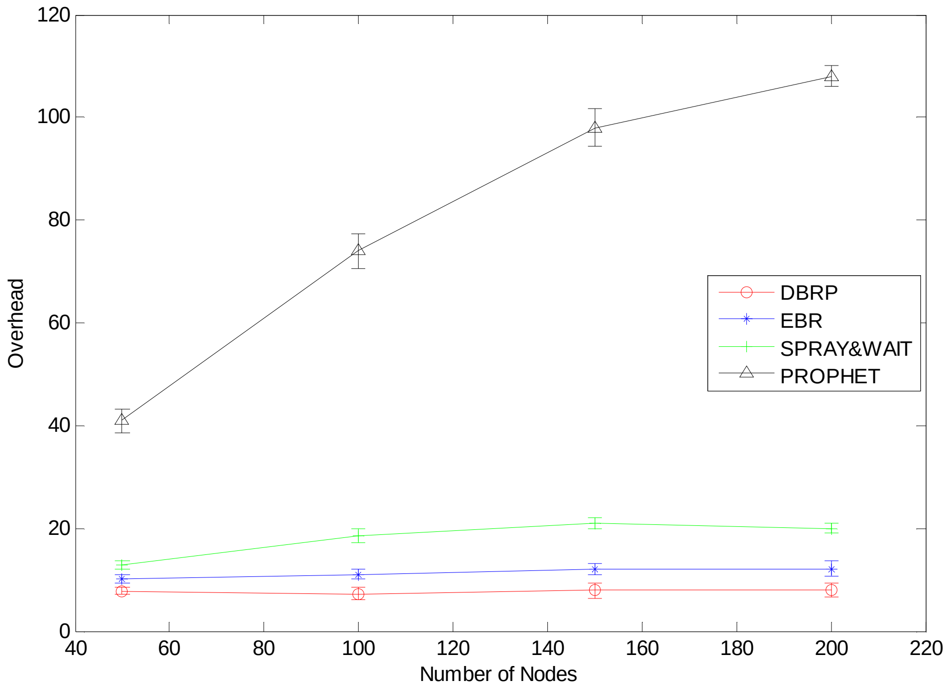
Figure 3-4 shows the impact of node density. As shown in Figure 3-4 (a)(c), DBRP is comparable to EBR in terms of delivery, while DBRP uses 28% fewer relayed nodes than EBR. This is because DBRP mostly targets the nodes that will meet the destination, meaning relay nodes may directly deliver bundles to the destination without disseminating more replicas. In terms of *DLO*, Spray and Wait works better than PROPHET but DBRP is 45% better than Spray and Wait. This is due to two factors. First, this mobility model fits perfectly into the hypothesis that past information on rate of encounters is an estimator for future rate of encounters. Therefore, nodes have a higher probability to visit each other in the future if they have met in the past. PROPHET also uses the history of observations in this mobility but its overhead and rate of dropped bundles do not allow it to surpass the performance of Spray and Wait, EBR and DBRP. Second, network utilization is correlated with delivery ratio, delay and overhead due to constrained buffer space and the number of nodes. As Spray and Wait floods the n replicas, we can see in Figure 3-4 (c) that in high density scenarios, the dissemination rate increases. Consequently, as all replicas have the opportunity of being forwarded, overhead increases. Spray and Wait has approximately 120% higher overhead than DBRP. The overhead of DBRP with an average of eight is, by far, the most resource-friendly, as shown in Figure 3-4 (c)(e).


DBRP
EBR
SPRAY&WAIT
PROPHET
Delivery Probability
Number of Nodes
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
40
60
80
100
120
140
160
180
200
220

(a)

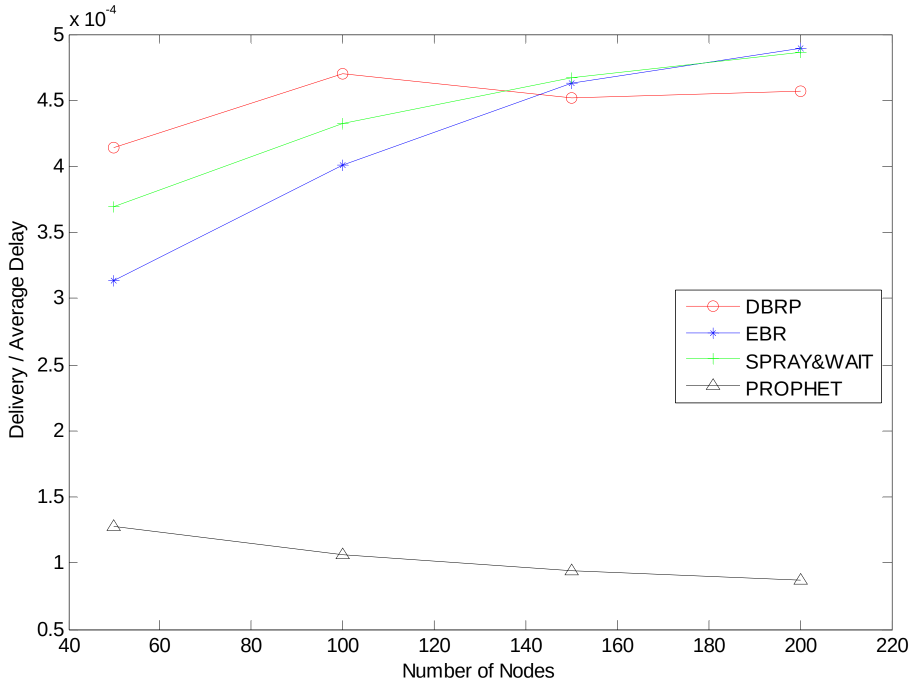

DBRP
EBR
SPRAY&WAIT
PROPHET
Average Delay
Number of Nodes
3600
3400
3200
3000
2800
2600
2400
2200
2000
1800
40
60
80
100
120
140
160
180
200
220

(b)



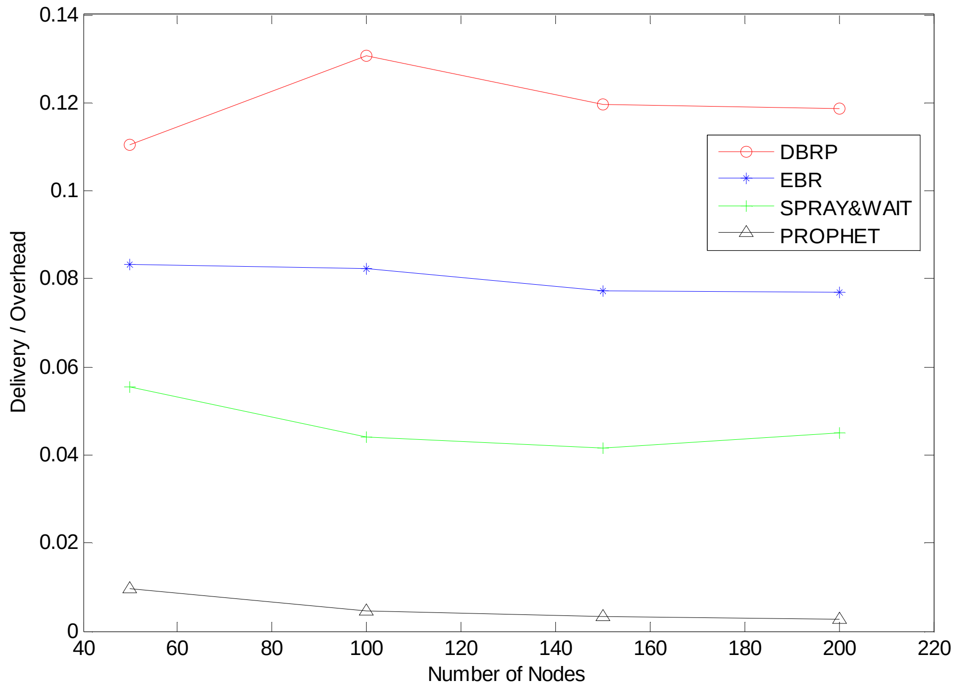
Overhead
Number of Nodes
DBRP
EBR
SPRAY&WAIT
PROPHET
0
20
40
60
80
100
120
40
60
80
100
120
140
160
180
200
220

(c)

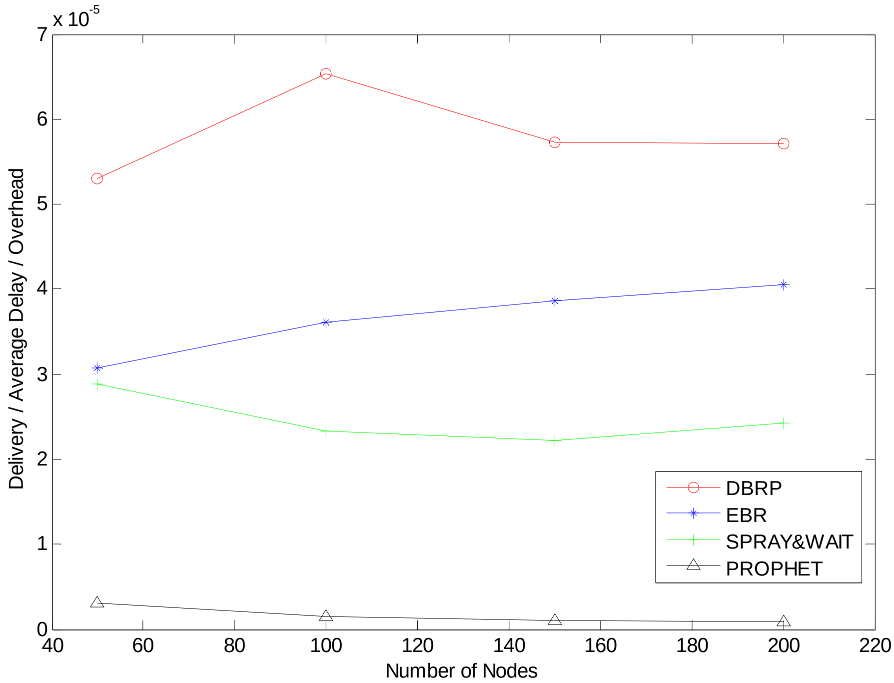


x 10-4
Delivery / Average Delay
Number of Nodes
DBRP
EBR
SPRAY&WAIT
PROPHET
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
40
60
80
100
120
140
160
180
200
220

(d)



(e)

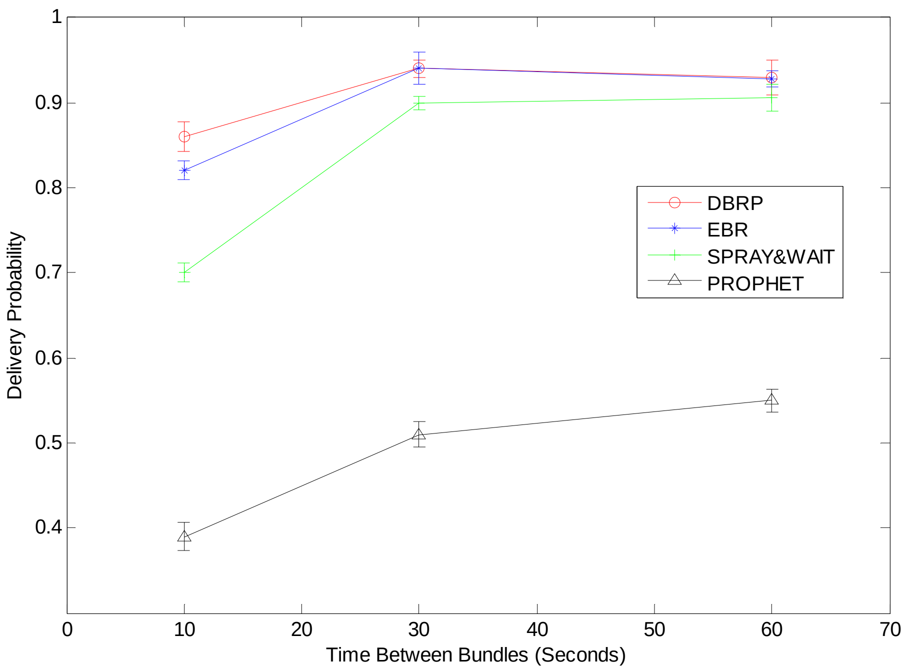


(f)

Figure 3-4 Network performance in different node densities, a) Delivery Probability, b) Latency Average, c) Overhead Ratio, d) Delivery * (1/ Latency Average), e) Delivery * (1/ Overhead), f)Delivery * (1/ Latency Average)* (1/ Overhead)

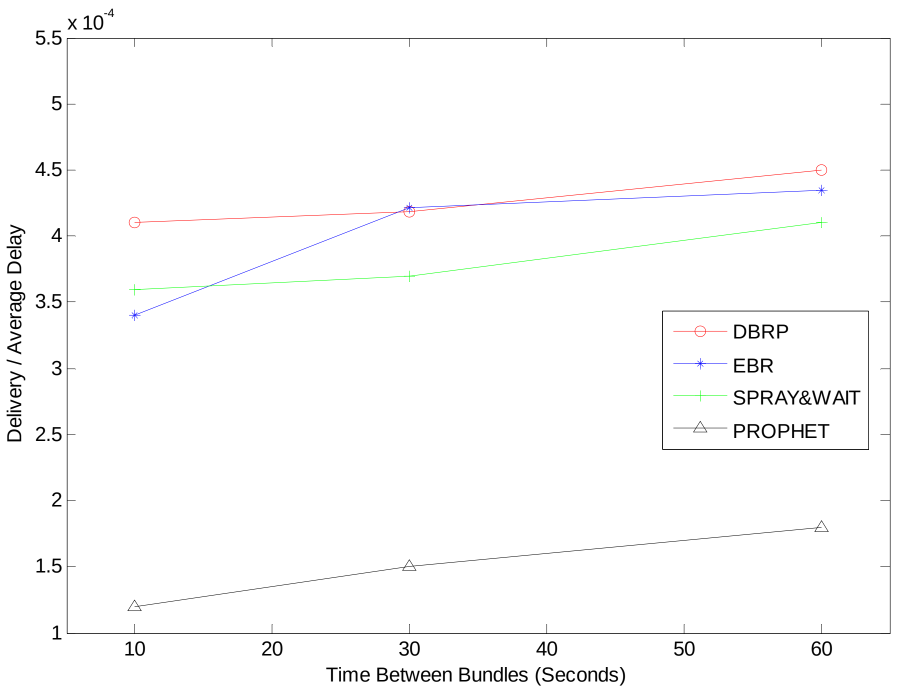
Figures 3-4 (d), (e) and (f) show the composite metrics *DL*, *DO* and *DLO*. In Figure 3-4 (b)(d), DBRP is shown to have 10% larger delays as compared to Spray and Wait when node density is low. DBRP is 20% better than Spray and Wait in terms of *DL*. This means, despite larger delays in DBRP, more bundles are delivered. The resulting delay is due to (i) the low dissemination rate of replicas, and (ii) the high ratio of dropped bundles as nodes have limited buffer size. We also see that in high density cases, EBR delivers bundles up to 25% quicker than DBRP. The reason is because EBR has a higher dissemination rate when number there are more nodes (see Figure 3-4 (c)(e)). Figure 3-4 (f) shows that in terms of the *DLO* metric, DBRP is up to 57% better than EBR.

In the second group of simulations, the offered load alternates between 1, 2 and 6 bundles per minute. There are 50 source and destination nodes. DBRP has the best performance in all categories. All the protocols suffer from low performance as the offered load increases. The average latency, however, shows PROPHET performed much worse than the other protocols. This is due to its reliance on a much larger buffer and hence an increase in load results in a higher rate of dropped bundles as compared to other protocols. In terms of delivery, by decreasing the load, the gap between PROPHET and the other protocols decreases. This is because the light load and the rate of dropped bundles decrease for PROPHET (see Figure 3-5 (d)). The composite metric in Figure 3-5 (e) shows that DBRP is at least 40% better than the other protocols.



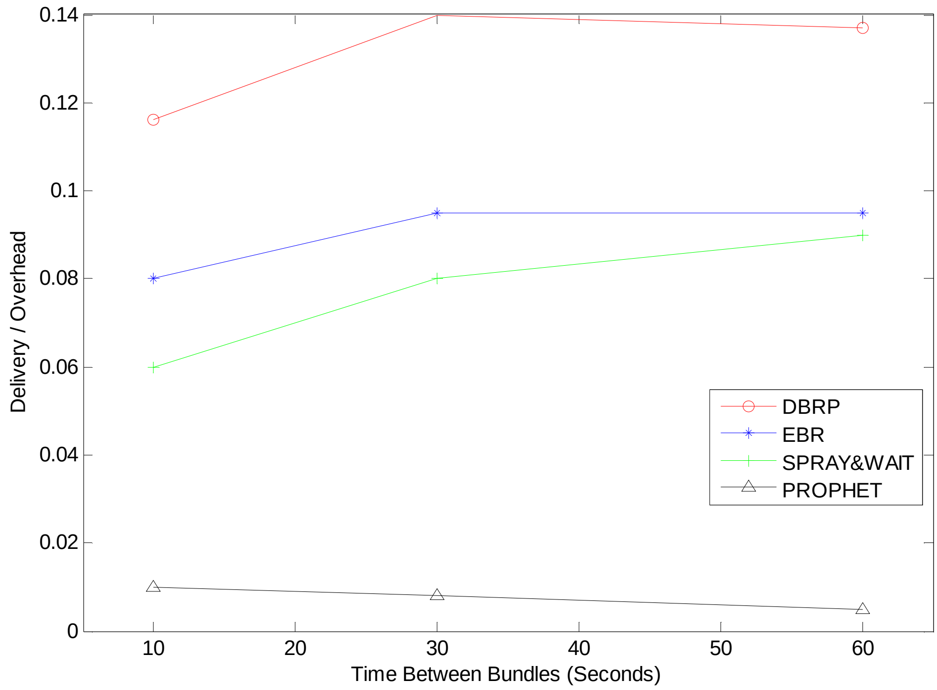
Delivery Probability
Time Between Bundles (Seconds)
DBRP
EBR
SPRAY&WAIT
PROPHET
1
0.9
0.8
0.7
0.6
0.5
0.4
0
10
20
30
40
50
60
70

(a)

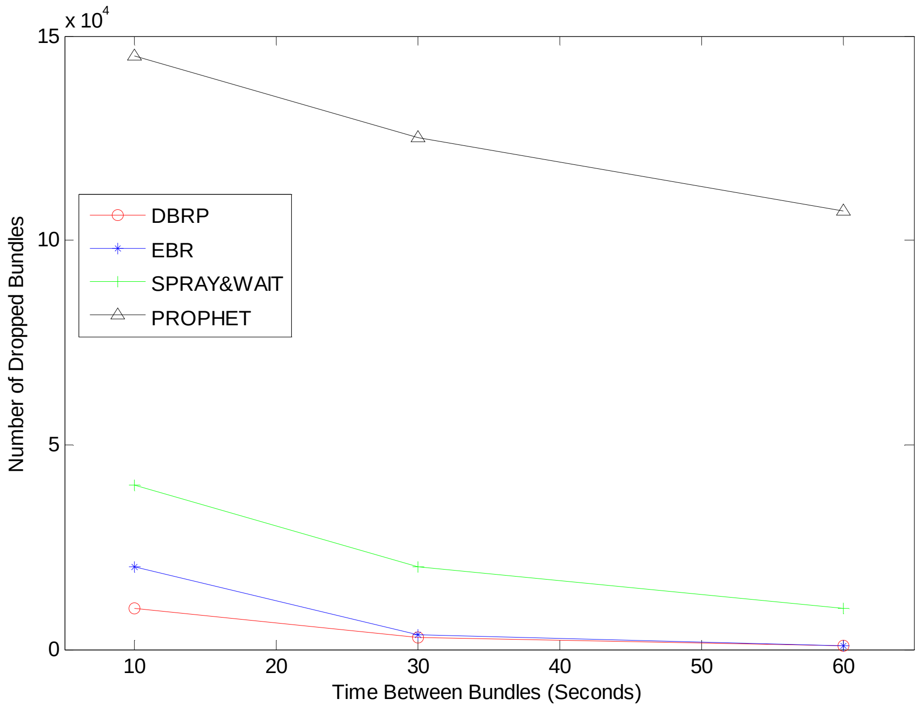


x 10-4
Delivery / Average Delay
Time Between Bundles (Seconds)
DBRP
EBR
SPRAY&WAIT
PROPHET
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
10
20
30
40
50
60

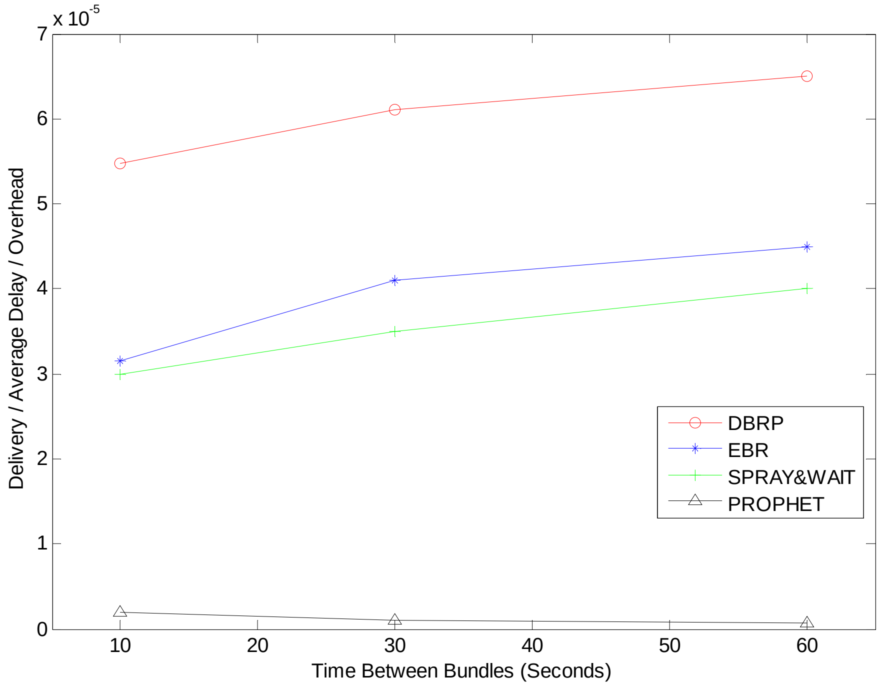
(b)


0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Delivery / Overhead
10
20
30
40
50
60
Time Between Bundles (Seconds)
DBRP
EBR
SPRAY&WAIT
PROPHET

(c)


x 10^4
15
10
5
0
Number of Dropped Bundles
10
20
30
40
50
60
Time Between Bundles (Seconds)
DBRP
EBR
SPRAY&WAIT
PROPHET

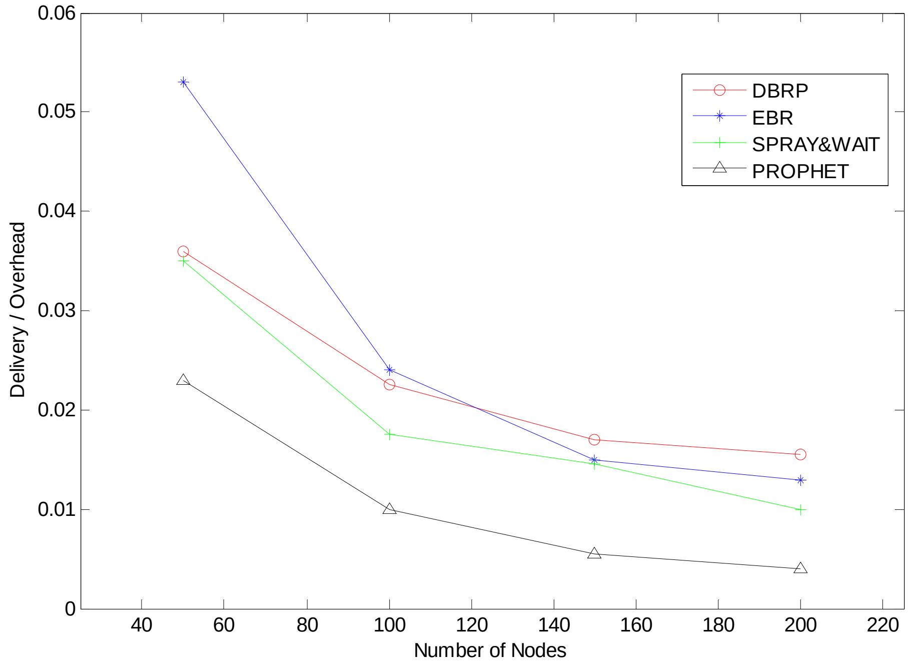
(d)



(e)

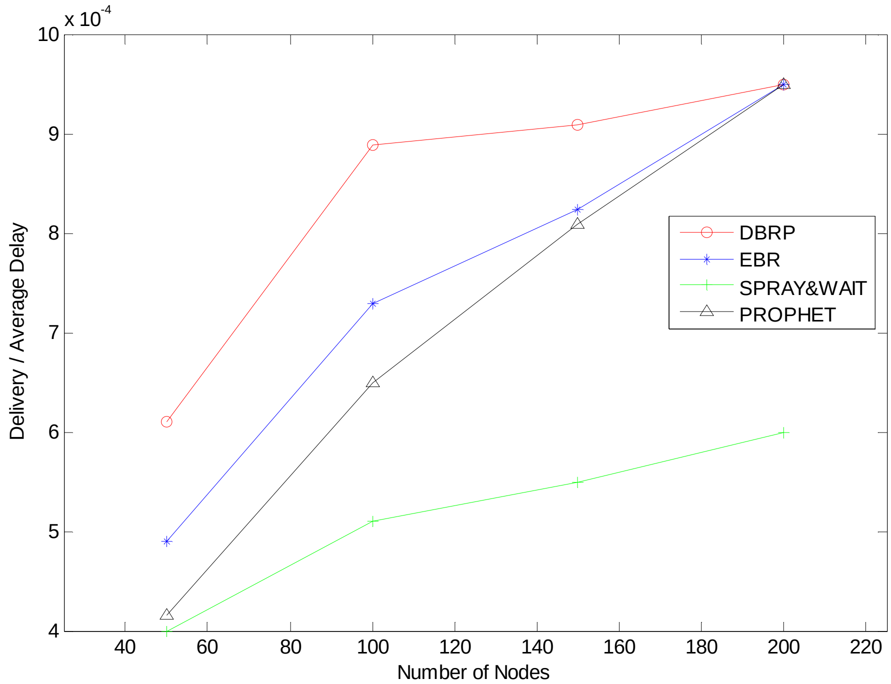
Figure 3-5 Network performance in different loads, a) Delivery Probability, b) Delivery * (1/ Latency Average), c) Delivery * (1/ Overhead), d) Number of Dropped Bundles, e) Delivery * (1/ Latency Average)* (1/ Overhead)

In the next experiment, each node has an infinite buffer. In addition, there is no constraint on the number of replicas. The observation here is that in low node density scenarios, history of encounters become in-effective. This is due to, in low density, a small number of contacts happen to generate a history of encounters. In this case, contact with destination is at a low rate that results the impact of encountering destination on the rate of encounters becomes less compared to in high node density scenarios.



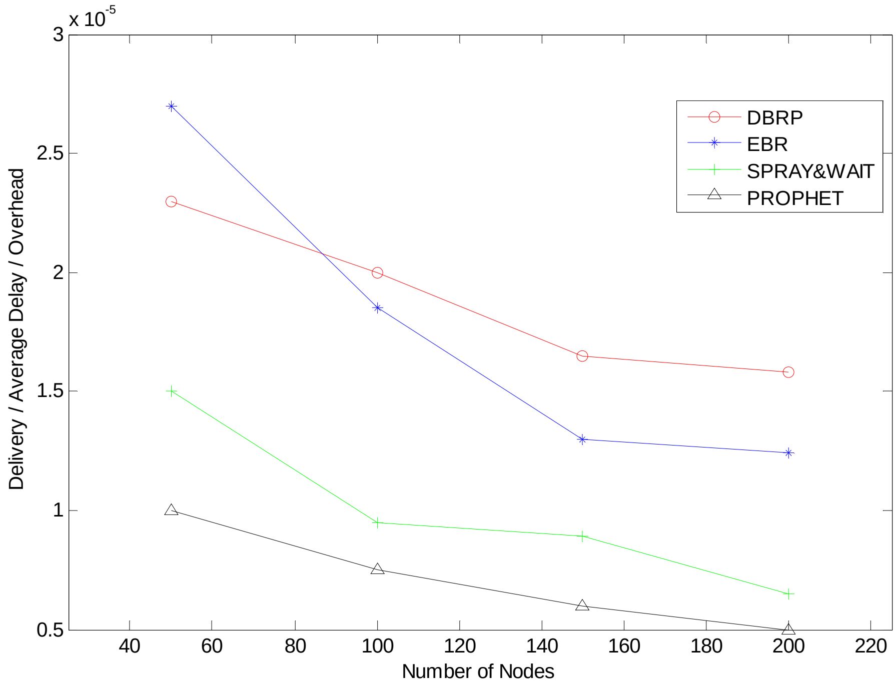
0.06
0.05
0.04
0.03
0.02
0.01
0
40
60
80
100
120
140
160
180
200
220
Number of Nodes
Delivery / Overhead
DBRP
EBR
SPRAY&WAIT
PROPHET

(a)



x 10^-4
10
9
8
7
6
5
4
40
60
80
100
120
140
160
180
200
220
Number of Nodes
Delivery / Average Delay
DBRP
EBR
SPRAY&WAIT
PROPHET

(b)



(c)

Figure 3-6 Network performance with unlimited buffer space, a) Delivery * (1/ Overhead), b) Delivery * (1/ Latency Average) c) Delivery * (1/ Latency Average)* (1/ Overhead)

As can be seen in Figure 3-6 (a), DBRP has higher performance in high density experiments (100, 200). This means that if the node density is high, many nodes around the destination can help to deliver the bundle. Spray and Wait also performs worse than the other quota protocols. This is due to the fact that replicas are only limited to nodes around the source, whereas DBRP and EBR forward replicas to high density areas in the direction of a destination. In terms of *DL*, Figure 3-6 (b) shows that DBRP delivers bundles up to 20% more quickly than EBR. This is because DBRP targets primarily the nodes that may meet a destination node in the future as relay nodes. In addition, as nodes have an unlimited buffer, forwarded bundles will not be dropped. It should be noted that DBRP may also forward replicas toward high density areas where the bundle dissemination rate is high. Figure 3-6 (b) shows that all the history-based protocols tested have better results than Spray and Wait. This is because of the network characteristic that nodes do not have random mobility patterns. In terms of *DLO*, Figure 3-6 (c) shows that DBRP is 36% better than the other protocols in high node density scenarios i.e., 150 to 200 nodes, where the node encounter rate is high. This is because in high density scenarios, nodes have high encounter rates, meaning forwarding opportunities are greater than in low density scenarios.

3.6 Conclusion

The ability to efficiently and effectively route data through intermittently connected networks is of critical importance to DTNs. Many current routing protocols utilize flooding-based techniques to obtain relatively high bundle delivery ratios. This, however, comes at the expense of high network resources such as bandwidth and storage. This chapter has proposed a destination based routing protocol that relies on forwarding replicas to the nodes that have some probability of encountering the destination node rather than forwarding to many nodes that have no encounters with the destination but may have high encounter rates with many other nodes. In other words, the probability of direct delivery is more reliable and has less resource intensive than delivery through many hops. To verify this hypothesis, this chapter used a Markov model to predict contacts between nodes. This prediction implies that if a sender node knows that a contact will happen between an intermediate node and a destination in a given period of time, the probability of delivery through the intermediate node is maximum.

As shown in Section 3.5, DBRP provides a comparable or better trade-off between bundle delivery, overhead and latency than flooding-based and quota-based protocols. However, DBRP may encounter congestion if nodes have a small buffer size and do not have sufficient opportunities to forward buffered bundles. In addition, due to short contacts, nodes may not be able to transmit all their queued bundles. This means in periods of congestion, under quota based protocols, if replicas are dropped due to limited buffer size, nodes cannot regenerate replicas. To address this gap, in the next chapter, a drop/forwarding policy is proposed for quota protocols.

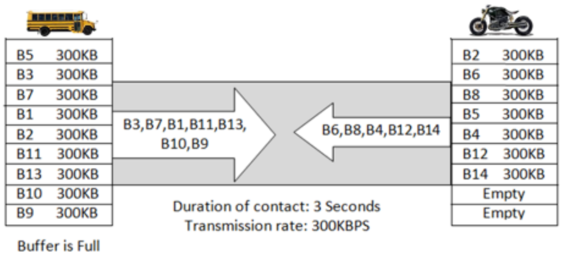
Chapter 4

A Novel Queue Management Policy for Intermittently Connected Vehicular to Vehicular Networks

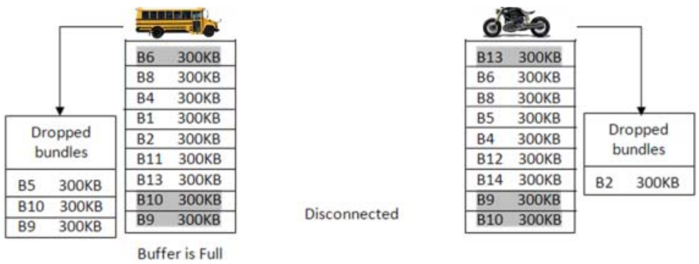
4.1 Introduction

The previous chapter has proposed a resource friendly protocol that considers whether a node has encountered the required destination node. Although quota protocols are resource friendly, under both quota and flooding protocols, nodes may have to buffer bundles for a long period of time. This gives rise to congestion if a node/vehicle has insufficient opportunities to forward buffered bundles; for example, due to short contact periods or not meeting a suitable next-hop vehicle frequently. Let us consider two vehicles moving at a speed of 20m/sec and have a radio range of 40 meter. Then the link between the two vehicles will last for 40/20 = 2 sec. A study on vehicular networks [20] shows that the duration of contacts between cars using IEEE 802.11g crossing at 20 Km/h is about 40 seconds, at 40 Km/h it is about 15 seconds and at 60 Km/h it is about 11 seconds. Consequently, vehicles need to determine: (i) the delivery order of bundles at each forwarding opportunity, and (ii) the bundles that should be dropped when their buffer is full.

As an example, Figure 4-1(a) shows that a bus and a motorbike have a three seconds contact period. The communication channel has a capacity of one bundle per second. Notice that the bus's buffer is full. Hence, the bus must determine which bundle(s) to drop; see Figure 4-1(b). However, dropping bundles arbitrarily may cause delivery failure. In addition, the bus and motorbike may have a short contact duration, meaning they are unable to exchange all bundles. Hence, the bus and motorbike must decide which bundles to forward first. In this case, the bus and motorbike need to prioritize their respective bundles with the goal of maximizing delivery ratio. In summary, it is important to have an efficient (i) bundle drop policy, and (ii) scheduling policy to decide the best bundle(s) to exchange.



(a)



(b)

Figure 4-1 An example of bundle transmission, a) connection is up, and b) connection is down

As mentioned in Chapter 2, to date, all buffer management schemes are targeted at flooding protocols. This is logical as congestion occurs more frequently than quota based protocols. However, under flooding protocols, if a bundle is dropped, there is still a high probability for the bundle to reach its destination. On the other hand, in quota protocols, as each bundle has finite copies, once a replica is dropped, the delivery probability of the corresponding bundle reduces. In other words, no provisions are provided to replace a dropped replica in order to maintain a high delivery ratio [29]. In the worst case scenario, source vehicles may remove all replicas of a bundle.

Given the said observations, this chapter proposes an efficient scheduling and drop policy for use under *quota* based protocols. This policy, called Queue Management in Encountered Based Routing Protocol (QM-EBRP), takes advantage of the following bundle and vehicle information: number of available replicas, maximum number of forwarded replicas, time to live and rate of encounters. This information is encapsulated in a multi-objective utility function that is then used for dropping or forwarding bundles. The proposed multi-objective utility function incorporates two metrics: (i) delivery ratio, and (ii) delay. The metric *delay* specifies how long it takes for a bundle to travel from a source to its destination, whilst *delivery ratio* is the total number of bundles that arrive at their intended destination successfully. To this end, the objective function considers how fast a bundle will be delivered and/or how fast the average delivery ratio increases. This information is encapsulated as the rate of change of the utility function with respect to two parameters: *number of available replicas* and *time to live*. Hence, forwarding bundles with the highest rate of change will improve delivery ratio and delay.

The rest of this paper is organized as follows. Section 4.2 describes the system and Section 4.3 formulates the problem and proposes the queue management policy QM-EBRP. Section 4.4 describes the research methodology and the results are discussed in section 4.5. Finally, section 4.6 concludes this chapter.

4.2 System Description

Let's consider a DTN where source vehicles generate bundles periodically. Each bundle specifies the number of copies which a relay is allowed to create. Each bundle must be delivered to its destination within a given time to live (TTL). Moreover, each vehicle records its rate of encounters with other vehicles. This will be used to determine the forwarding priority of a bundle at each contact, and which bundles to drop when buffer overflows. This section first describes system settings. Specifically, this section first expounds the routing protocol (forwarding strategy), mobility model and assumptions before formulating the problem precisely.

4.2.1 Routing

As mentioned, this chapter considers encounter based quota protocols [30, 97], specifically EBR [30]. In details, EBR generates a finite number of replicas for each bundle. Every vehicle running EBR is responsible for maintaining its past average rate of encounter with other vehicles, which is then used to predict future encounter rates. As mentioned in Chapter 2, in order to track a vehicle's rate of encounter, the vehicle maintains two pieces of local information: an encounter value (EV), and a current window counter (CWC). The variable EV represents a vehicle's past rate of encounters as an exponentially weighted moving average, while CWC is the number of encounters in the current time interval. EV is updated periodically to account for the most recent CWC. Specifically, EV is computed as follows:

$$EV = \alpha \times CWC + (1 - \alpha) \times EV_{(old)} \tag{4.1}$$

where $\alpha \in (0,1)$ is a weighting coefficient; i.e., $\alpha = 0.85$. In EBR, every 30 seconds, nodes' encounter rate is updated and the CWC is reset to zero.

The primary purpose of tracking the rate of encounters is to decide how many replicas of a bundle a vehicle will transfer during a contact opportunity. Hence, when vehicles a and b meet each other, vehicle a sends a proportional number of the i^{th} bundle M_i based on the encounter rate of both sender and receiver. Specifically,

$$k = m_i \times \frac{EV_b}{EV_b + EV_a} \tag{4.2}$$

where m_i is the available number of replicas for the i^{th} bundle at node a. The terms EV_a and EV_b respectively represent the encounter rate for nodes a and b. As a result, k replicas of bundle M_i is forwarded to node b. In words, the nodes that experience a large number of encounters are most likely to successfully pass the bundle along to the final destination than nodes that do not encounter other nodes frequently.

This method adopts EBR because of the following reasons. Firstly, it uses encounter rates when forwarding bundles. In DTNs, vehicles will naturally have varying rates of encounters [30]. This parameter is used to derive the service rate of a vehicle. Secondly, EBR limits the number of replicas for each generated bundle. Therefore, for each bundle, a fixed number of replicas exist in the network that gives knowledge to each vehicle to know the maximum number of replicas of each bundle that can be disseminated in the network.

4.2.2 Mobility Model

Vehicles change their location, velocity and acceleration over time. These parameters are governed by the mobility model. In general, mobility models [98-100] can be categorized into (i) map, and (ii) random. Map based models dictate vehicles' movement according to predefined paths and routes derived from real map data. In random mobility models, vehicles do not follow any predetermined paths. However, random mobility models are not realistic as humans do not move randomly. Hence, this chapter considers mobility models, e.g., [98-100], where meeting times between vehicles are exponentially distributed. Here, 'meeting' refers to the time when two vehicles come within radio range of each other. We now show that exponentially distributed meeting rate results in an exponential delivery ratio.

Lemma 1. Let γ be the average meeting rate of L vehicles is modelled as an exponential distribution. Then the Delivery Probability (DP) is also exponentially distributed.

Proof. Assume a bundle has N replicas to be disseminated. Also assume that all vehicles, including the destination, have the same chance to see the bundle. Therefore, the probability that the bundle has been delivered is,

$$DP = \frac{N}{L-1} \tag{4.3}$$

As mentioned, replicas are forwarded upon contact or at meetings. Also, the dissemination rate of a bundle is dependent on the number of replicas and meeting rate. Hence, if the meeting rate is governed by an exponential distribution, the dissemination rate will also follow the same distribution. That is,

$$DP = \frac{\gamma}{L-1} \tag{4.4}$$

In this chapter the following assumptions are made:

1. Each bundle has a finite number of replicas.
2. In order to replicate a bundle, a vehicle will keep one replica for itself and the other replicas are forwarded to other vehicles.
3. Each vehicle has a finite buffer.
4. Short contact duration, meaning vehicles do not have sufficient bandwidth to empty their buffer.
5. Vehicles have different speeds.
6. Vehicles move independently of each other.
7. Vehicles have different meeting rates at different time *t*.

4.3 Problem Formulation

Let us consider a contact between vehicles *i* and *j*, with both vehicles having limited resources; i.e., low data rate and buffer space. In this setting, there are two sub-problems:

- *Priority forwarding*. If vehicle *i* has bundles to forward to vehicle *j*, but is faced with a short contact duration or low data rate, both of which prevents it from forwarding all bundles to vehicle *j*, the question then is to determine which bundles to forward such that the delivery ratio is maximized and the delay is minimized.

- *Buffer management*. Consider when one or more bundles arrive at vehicle *j* with a full buffer. The question then is to determine which bundles to discard whilst maximizing delivery ratio and minimizing delay.

4.3.1 Proposed Queue Management Policy

The objective of queue management is to control congestion in order to improve delivery and delay. However, queue management becomes challenging when there are only a finite number of replicas, as is the case with quota protocols. To this end, this section proposes a Queue Management policy for Encounter-Based Routing Protocols (QM-EBRP), designed specifically for quota based protocols with the aim of (i) maximizing the expected delivery ratio of all bundles, and (ii) the expected average delay of all delivered bundles.

4.3.2 Overview

Algorithm 1 presents the steps performed by QM-EBRP. Figure 4.2 provides an overview of QM-EBRP's functional modules and their relationships. The proposed algorithm starts whenever a connection is up (line 2). Upon contact, a node can either be in the transmit or receive mode, depending on the summary vector exchange during contact. In the receiving mode, for every bundle i in a receiver’s buffer, the multi objective utility $UF_i()$ is called to determine the bundle's utility. After that, bundles are sorted in ascending order. Finally, based on the sorted bundle list, *dropQueue* bundles are dropped from the head of the queue (lines 4 - 9). In the sending mode, the EBR [30] routing protocol selects bundles to forward. Hence, there is a list of bundles for forwarding, called *forwardSelection*. In the next step, a multi-objective utility is calculated for every bundle in the *forwardSelection* list. Bundles are then sorted in descending order. Finally, bundles are dropped from the head of the sorted list *forwardQueue* (line 18).

Algorithm 1 QM-EBRP drop/forward policy

1- **Start**
2- **while** connection **is** up
3- **if** mode = receiving
4- *ReceiverQueue* ← bundles in receiver's buffer
5- **for** every bundle *i* in *ReceiverQueue*
6- UF_i ← multi_objective_utility_function (i)
6- **end for**
8- *dropQueue* ← sort(UF, ReceiverQueue, 'increase')
9- DROP(*dropQueue*)
10- **end if**
11- **if** mode = sending
12- *SenderQueue* ← bundles in sender's buffer
13- *forwardSelection* ← EBR(*SenderQueue*)
14- **for** every bundle *i* in *forwardSelection*
15- UF_i ← multi_objective_utility_function (i)
16- **end for**
17- *forwardQueue* ← sort(UF, *forwardSelection,* 'decrease')
18- FORWARD(*forwardQueue*)
19- **end if**
20- **End**

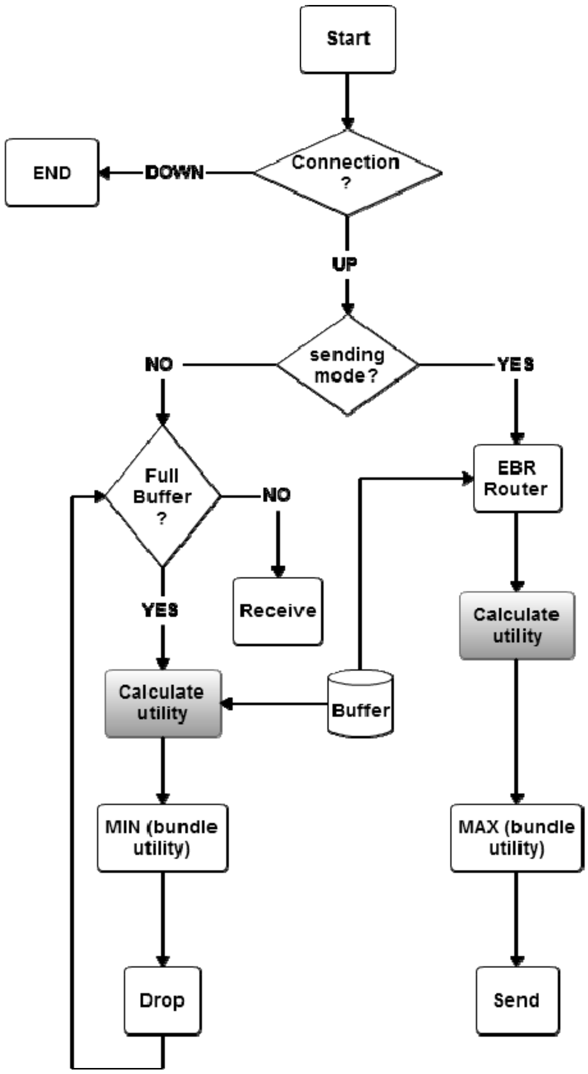


Figure 4-2 QM-EBRP flowchart for forward or drop policy

A key module used by the forwarding process is the multi-objective utility function, which uses the delay and delivery function. Figure 4-3 depicts the components of the proposed multi-objective function. Briefly, as explained in Section 4.3.3, the delivery function () considers the probability of delivery for every bundle *i*. To calculate the delivery probability, the system needs to calculate how likely bundle *i* has been delivered or will be delivered in the future. This is carried out, for a given bundle *i*, using the number of disseminated replicas and the number of replicas that will be disseminated in the future. The delay function considers the expected delay of bundle *i* if the bundle is not yet delivered (details in Section 4.3.4). The expected delay of bundle *i* is the time until the first copy of bundle *i* is delivered to its destination. Given both functions, their rate of change with respect to two parameters; namely, number of current replicas ( ) and bundle's lifetime ( ) are used to derive a bundle i's maximum delivery ratio and minimum delay; see Section 4.3.3.1 and 4.3.4.1. Both functions are then used in a multi-objective function, which is then responsible for prioritizing bundles during congestion and forwarding. Table 4-1 lists a summary of all notations used in the following sections.

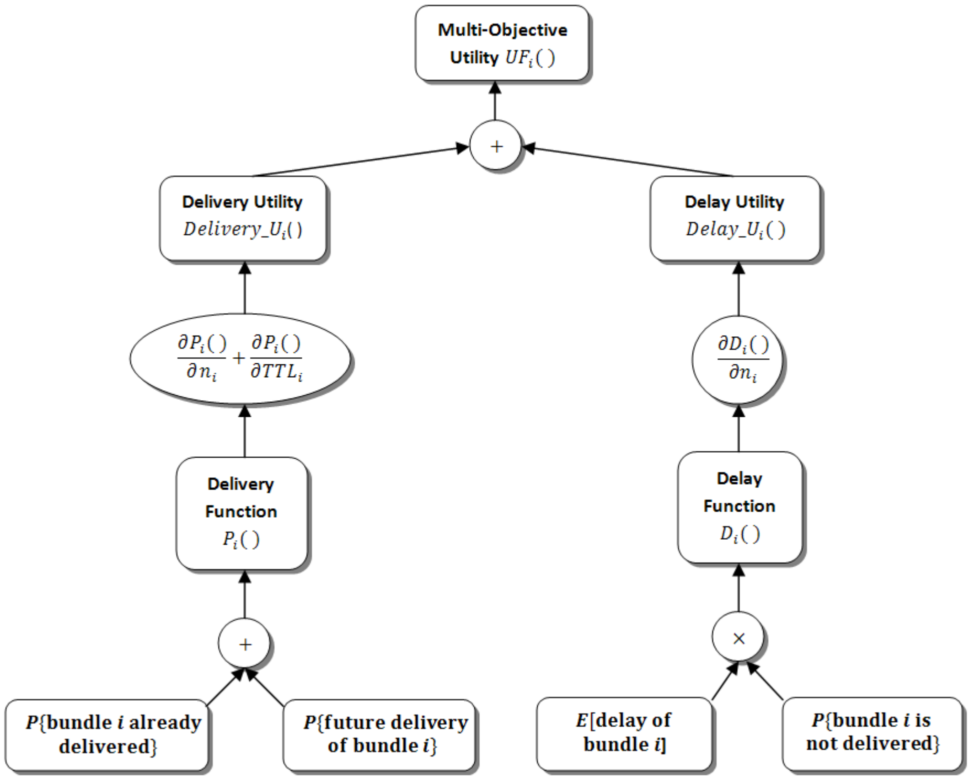


Figure 4-3 Multi-objective function components

4.3.3 Delivery Function

Let L denote the number of vehicles. Denote the number of bundles at time t by $K(t)$. Each bundle has N replicas. Assume each vehicle has a different meeting probability $P_{con}(t)$, and each bundle i has a lifetime at time t of $TTL_i(t)$. In fact, $P_{con}(t)$ determines the service rate of a vehicle. Hence, the probability that a copy of bundle i will not be delivered by a vehicle is dependent on the probability that a vehicle's next meeting time with the destination is greater than $TTL_i(t)$. This probability is equal to $\exp\left(-P_{con}(t) \times TTL_i(t)\right)$.

For each bundle $i \in [1, K(t)]$, let $n_i(t)$ be the number of replicas that a vehicle has in its buffer at time *t*. Also, denote $m_i(t)$ the number of replicas of bundle i that has been forwarded to other vehicles at time *t; i.e.,* let $n_i(t) + m_i(t) = N$. For example, a source node generates bundle i with 10 replicas ($N = 10$), after two contacts with other nodes, only three replicas are left at source nodes ($n_i(t) = 3$). Hence, the maximum number of replicas that has been disseminated throughout the network is seven ($m_i(t) = 7$). Also, define 'A' and 'B' to be the event "bundle i has been delivered' and "bundle i will not be delivered in the future" respectively. Then if bundle *i* has $n_i(t)$ replicas at time *t,* the required conditional probability is calculated as,

$$P_i\{B|A\} = \prod_{k=1}^{n_i(t)} exp(-P_{con}(t) \times TTL_i(\mathrm{t}))$$
$$= exp\big(-P_{con}(t) \times n_i(t) \times TTL_i(\mathrm{t})\big) \tag{4.3}$$

Equation (4.3) has not taken into account whether a copy of bundle i has been delivered up to time *t*. Hence, if all vehicles including bundle *i*'s destination are assumed to have the same chance to receive bundle i, the probability that one of the $m_i(t)$ replicas of bundle *i* has been delivered is,

$$P_i\{\bar{A}\} = \frac{m_i(t)}{L-1} \tag{4.4}$$

where $\bar{A}$ corresponds to the event "bundle *i* is delivered". Combining Equation (4.3) and Equation (4.4), the probability that a bundle i with *N* replicas will be delivered before its TTL expires is,

$$P_i = P_i\{A\} \times (P_i\{\bar{B}|\bar{A}\}) + P_i\{\bar{A}\}$$
$$= \left(1 - \frac{m_i(t)}{L-1}\right) \times (1 - \exp\big(-P_{con}(t) \times TTL_i(t) \times n_i(t)\big)) + \frac{m_i(t)}{L-1} \tag{4.5}$$

In other words, Equation (4.5) calculates the delivery probability of each bundle. Hence, the global delivery ratio (DR) of all existing bundles at time *t* is calculated as follows,

$$DR = \sum_{i=1}^{K(t)} \left[\left(1 - \frac{m_i(t)}{L-1}\right) \times \left(1 - \exp\left(-P_{con}(t) \times TTL_i(t) \times n_i(t)\right)\right) + \frac{m_i(t)}{L-1} \right] \tag{4.6}$$

4.3.3.1 Delivery Utility

To maximize the delivery ratio, we will need the rate of change with respect to $n_i(t)$ and $TTL_i(t)$. Specifically, the gradient of the delivery ratio is,

$$\nabla P_i = \frac{\partial P_i}{\partial n_i(t)}\, \mathrm{d}n_i(t) + \frac{\partial P_i}{\partial TTL_i(t)}\, \mathrm{d}TTL_i(t) \tag{4.7}$$

where $\frac{\partial P_i}{\partial n_i(t)}$ and $\frac{\partial P_i}{\partial TTL_i(t)}$ are the rate of change of the delivery ratio with respect to $n_i(t)$ and $TTL_i(t)$ and are defined as follows,

$$\frac{\partial P_i}{\partial n_i(t)} = \left(1 - \frac{m_i(t)}{L-1}\right) \times P_{con}(t) \times TTL_i(t) \times \exp\left(-P_{con}(t) \times TTL_i(t) \times n_i(t)\right) \tag{4.8}$$

$$\frac{\partial P_i}{\partial TTL_i(t)} = \left(1 - \frac{m_i(t)}{L-1}\right) \times P_{con}(t) \times n_i(t) \times \exp\left(-P_{con}(t) \times TTL_i(t) \times n_i(t)\right) \tag{4.9}$$

The maximal directional directive is then,

$$Delivery\_U_i = \sqrt{\frac{\partial P_i}{\partial n_i(t)}^2 + \frac{\partial P_i}{\partial TTL_i(t)}^2} \tag{4.10}$$

As mentioned in Section 4.3.5, QM-EBRP uses Equation (4.10) as the delivery utility for a copy of bundle *i* with respect to the total delivery rate.

4.3.4 Delay Function

This section considers delay. Let X_i be a random variable corresponding to the delay of bundle *i*. Also, let T_i be the elapsed time for bundle *i*. In other words, it measures the time since bundle *i* was generated by its source vehicle. Then, the expected delay for bundle *i* for which none of its copies are delivered is given by

$$D_i = \left(1 - \frac{m_i(t)}{L-1}\right) \times \ E[X_i > T_i] \tag{4.11}$$

The mean or expected value of an exponential distribution with rate parameter λ is $\frac{1}{\lambda}$ [101]. The analysis proved that the time until the first copy of bundle *i* reaches the destination follows an exponential distribution with rate parameter $P_{con}(t) \times n_i(t)$. Hence, the mean or expected value of this distribution is $\frac{1}{P_{con}(t) \times n_i(t)}$ [81]. It follows that,

$$E[X_i > T_i] = T_i + \frac{1}{P_{con}(t) \times n_i(t)} \tag{4.12}$$

Substituting Equation (4.5) into Equation (4.6), we have,

$$D_i = \left(1 - \frac{m_i(t)}{L-1}\right) \times \left(T_i + \frac{1}{P_{con}(t) \times n_i(t)}\right) \tag{4.13}$$

Hence, D_i is the expected delay for each bundle *i*. The following equation is used to calculate the average delay (AD) of all bundles at time *t*,

$$AD = \sum_{i=1}^{K(t)} \left[\left(1 - \frac{m_i(t)}{L-1}\right) \times \left(T_i + \frac{1}{P_{con}(t) \times n_i(t)}\right)\right] / K(t) \tag{4.14}$$

4.3.4.1 Delay Utility

This section outlines a method to minimize the average delay. Equation (4.15) represents the delay utility for bundle i. The rate of change for delay is derived, see Equation (4.13), in the direction of the negative gradient with respect to $n_i(t)$. The derived equation represents how fast a bundle will be delivered. This means a bundle with a large delivery utility will experience minimum delay. Hence, a node needs to apply the following delay utility for each bundle i,

$$Delay\_U_i = \frac{\partial D_i}{\partial n_i(t)} = \left(-1 + \frac{m_i(t)}{L-1}\right) \times \left(\frac{1}{P_{con}(t) \times n_i(t)^2}\right) \tag{4.15}$$

4.3.5 Multi Objective Utility Function

Now, a multi objective function is used to incorporate delivery (see Eq. 14) and delay utility (see Eq. 15). Briefly, a multi objective utility function is represented as the following multi-objective optimization problem,

$$\min(f_1(x), f_2(x), \dots, f_k(x)) \qquad or \qquad \max(f_1(x), f_2(x), \dots, f_k(x)) \tag{4.16}$$

where the integer $k \geq 2$ is the number of objectives and x is a vector of decision variables in the set X. A key issue when incorporating the said utilities is that their values are in a different domain. For example, the domain of the delivery utility belongs to $\mathbb{R}^+$ and for the delay utility it is $\mathbb{R}^-$. To this end, the delay and delivery utility are normalized as follows,

$$\varphi(Delivery\_U_i) = \frac{Delivery\_U_i - \mu_{dvu}}{\sigma_{dvu}} \tag{4.17}$$

where μ_{dvu} is the mean of delivery utility of all bundles in a vehicle's queue. Also, σ_{dvu} is the standard deviation of delivery utility of considered bundles. The same procedure applies to $Delay\_U_i$. Specifically,

$$\varphi(Delay\_U_i) = \frac{Delay\_U_i - \mu_{dlu}}{\sigma_{dlu}} \tag{4.18}$$

where $\mu_{dlu}$is the mean of delay utility of all bundles in a vehicle's queue. Also, σ_{dlu} is the standard deviation of delay utility of the considered bundles. Hence, the multi-objective utility function UF_i used by QM-EBRP is follows,

$$UF_i = \varphi(Delivery\_U_i) + \varphi(Delay\_U_i) \tag{4.19}$$

In words, Equation (4.19) represents how fast bundle i reaches the maximum delivery rate and minimum delay. Hence, if bundle i has a greater utility value than bundle j, bundle i will have a higher delivery probability and lower delay. Hence, in this QM-EBRP, Equation (4.19) is used in order to obtain the utility for each bundle.

Table 4-1 Summary of notations

Variable	Description
L	Number of vehicles
$n_i(t)$	Number of available replicas of bundle *i* at a vehicle at time *t*
$TTL_i(t)$	Remaining time to live for bundle *i*
$m_i(t)$	Number of forwarded replicas of bundle *i* up to time *t*
$N_i = n_i(t) + m_i(t)$	Total number of replicas for bundle *i*
$P_{con}(t)$	Vehicle's encounter rate
$K(t)$	Number of bundles in the system at time *t*
T_i	Elapsed time for bundle *i*

4.4 Evaluation

The experiments are conducted in the Java based simulator, Opportunistic Network Environment (ONE) [95]. It is able to generate vehicle movements using different mobility models. Example mobility models [98-100] include shortest map based model, working day movement model, and random walk model.

This section evaluates QM-EBRP against six local knowledge policies and one optimal global knowledge policy. Briefly, the investigated policies include: Drop

Oldest (DO), Last Input First Output (LIFO), First Input First Output (FIFO), Most FOrwarded first (MOFO), LEast PRobable first (LEPR), and drop greatest HOP-COUNT. DO drops the oldest bundle if a node's buffer is full and forwards the bundle that has maximum lifetime. LIFO drops the last arriving bundle and forwards the bundle at the head of queue. FIFO drops the bundle at the head of the queue and forwards the last bundle that has arrived. In MOFO, every node maintains a variable *FP,* which is initialized to zero, for each bundle. Each time a bundle is forwarded, *FP* is updated according to Equation (4.20), where P is the delivery probability that is used in PROPHET [5].

$$FP = FP_{old} + P \tag{4.20}$$

The bundle that has been forwarded the most i.e., highest *FP*, is dropped first and the bundle that has been forwarded the least i.e., lowest *FP,* is forwarded first. LEPR drops the bundle with the lowest delivery probability. In other words, LEPR drops the bundle that has the lowest *P*. Lastly, HOP-COUNT drops the bundle that has the greatest number of hops and forwards the bundle that has the smallest number of hops. QM-EBRP is also evaluated against Optimal Global Knowledge (OGK), a scheme that is similar to [81] and [87]. In this policy, nodes are assumed to be synchronized with a shared global memory to update bundle information such as the number of disseminated replicas. Accordingly, every node is instantly aware of the accurate number of disseminated replicas of each bundle in the network. This policy thus allows us to compare QM-EBRP against a theoretical scheme.

The experiments in this section are categorized into three groups based on mobility models. In the first group of experiments, a shortest map based model is considered in a 5×3 km^2 area of downtown Helsinki, Finland. There are 60 vehicles, each with a radio range of 20 meters. First, all vehicles are assumed to have infinite buffer space and the speed of vehicles is varied from 0.5 to 60 m/s, at an increment of 10. This causes vehicles to have different contact durations. After that, all vehicles are assumed to have finite buffer space and move at a constant speed of 30m/s. In this case, vehicles' buffer space is varied from five to 40 bundles, where the buffer size is doubled that of the previous experiment; i.e., 5, 10, 20 and 40 bundles. Lastly, this thesis studies the scenario where vehicles have space for five bundles and the number of source/destination is varied from 10 to 60. In this experiment, bundles have 60 minutes lifetime and the simulations last for 12 simulated hours.

In the second experiment group, the working day movement of 60 people and 50 taxi cabs is simulated in a 10×8 km^2 area of Manhattan, New York, United States of America [95]. People use their car with probability 0.5 to go shopping or work. Otherwise they have to walk or catch a taxi cab with a probability of 0.5. Cars and taxi cabs move at a minimum speed of 20 m/s and a maximum speed of 30 m/s, and pedestrians move at 2 m/s. Note, nodes are either at home, working or carrying out other activities such as shopping and meetings. These activities are deem to be the most common and capture a typical working day for most people [102]. This experiment evaluates the network performance when the buffer space is varied from 10 to 70 bundles in increments of 10 bundles. All nodes are equipped with a radio range of 30 meters. In this experiment, bundles have eight hours lifetime and the simulations last for three simulated days.

In the third group of experiments, 60 nodes with a radio range of 30 meters move randomly in a 2×2 km^2 area. This experiment evaluates the network performance when the buffer space is varied from 10 to 200 bundles in increments of 20 bundles. Bundles have five hours lifetime and the simulations last for 24 simulated hours. Note, in all experiments, the bundle size is 100 KB, and sources generate a bundle every 10 seconds. All vehicles, upon contact, have a transmission speed of 100 KBps. Also, each data point is an average of 10 runs, with minimum and maximum confidence intervals.

As mention in Chapter 3, this thesis considers three conventional performance metrics as well as introducing three other metrics used by the authors of EBR [30] to show the relative relationship between conventional metrics. Conventional metrics used include 1) *delivery probability*, defined as the ratio between the number of delivered bundles to the number of generated bundles, 2) *overhead*, defined as the ratio of the number of delivered bundles and number of carrier nodes, 3) *average delay*, defined as the time from when a bundle is generated to its reception time.

While these three conventional metrics provide a comprehensive comparison, many protocols optimize one metric at the expense of another. Consider a protocol that delivers bundles quickly by preferentially using routes with a small number of hops. Otherwise, it does not forward bundles. Consequently, the protocol has a low overhead but delivery ratio is low. To overcome this issue, the composite metrics used in chapter 2 are used to penalize protocols that unfairly optimize a metric. To remind the reader, Equation (4.21) defines *DA* based on Delivery Ratio (*DR*) and Average Delay (*AD*).

$$DA = DR \times \frac{1}{AD} \tag{4.21}$$

In other words, *DA* scales the performance accordingly if a protocol optimizes for delivery ratio but has poor delay. Equation (4.22) defines DOR based on DR and Overhead Ratio (OR), i.e.,

$$DOR = DR \times \frac{1}{OR} \tag{4.22}$$

Hence, *DOR* captures the trade-off between DR and resulting overheads. Lastly, Equation (4.23) defines *DAO* based on *DR*, *AD* and *OR*.

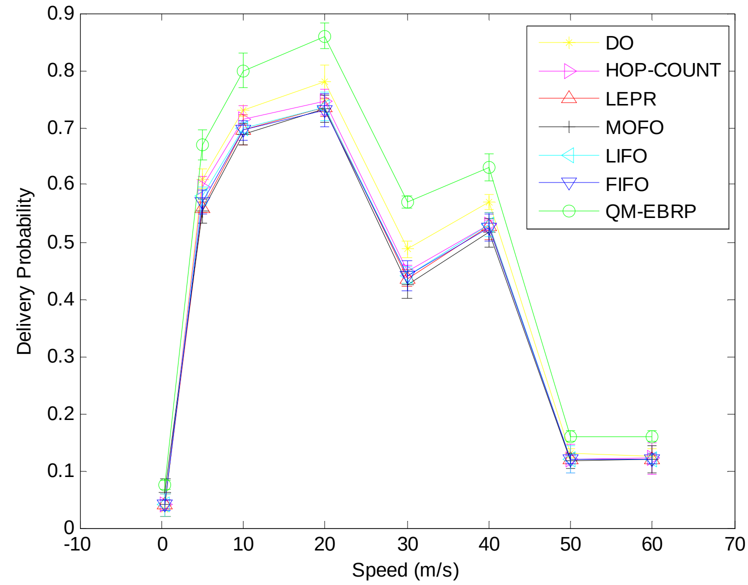
$$DAO = DR \times \frac{1}{AD} \times \frac{1}{OR} \tag{4.23}$$

In other words, DAO quantifies the performance of a protocol that myopically optimizes delivery ratio at the expense of average delays and overheads.

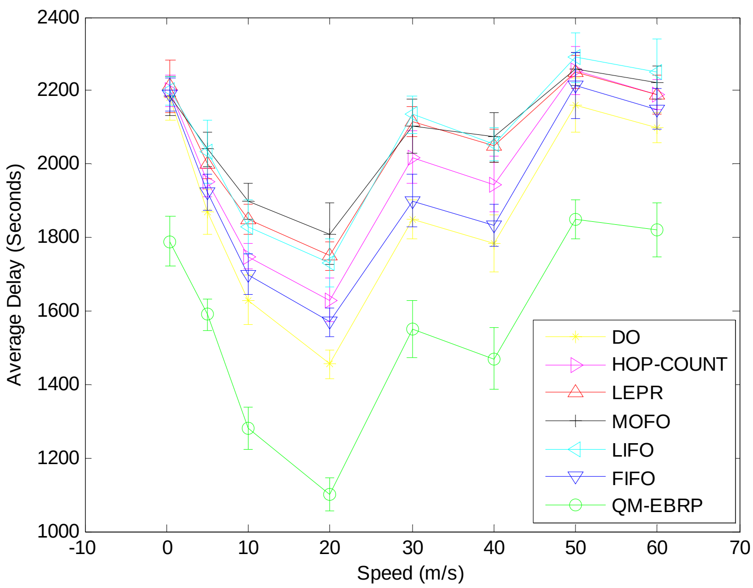
4.4.1 Shortest Map-based Mobility

Figure 4-4 shows the impact of speed and radio range when vehicles have infinite buffer space. Hence, there is no drop policy. Recall that in the first scenario, nodes have different speeds, which help to simulate different contact duration. That is, when vehicles' speed increases, contact periods become shorter and nodes cannot forward all queued bundles during contacts. Figure 4-4 (a) shows that the policies that do not use bundle information such as TTL result in low delivery ratios. For example, FIFO, HOP-COUNT, LEPR, MOFO and LIFO have a delivery ratio between 70.5% and 71.3%. These policies prioritize bundles based on information such as arrival time, nodes' encounter rate and number of relays. Hence, for said policies, nodes may receive old bundles that do not have sufficient lifetime. Recall that the main reason for using bundle lifetime is to avoid forwarding old bundles during contact. For example, DO sends the bundle that has the longest remaining lifetime. As shown, DO has 5% better delivery performance as compared to said policies. Now, consider the scenario where node A has stored a bundle that has a large lifetime but the bundle has no more replicas to be forwarded. Accordingly, if node A meets the bundle's destination, the bundle will be delivered. Otherwise, it will never leave node A until its lifetime expires. In QM-EBRP, a higher forward priority is given to bundles that have a large lifetime and those that will generate a large number of replicas in the future. As shown in Fig4-4 (a), QM-EBRP performs up to 15% better than other policies in terms of bundle delivery. Note that, at speeds of 0.5m/s and 60m/s, all the considered forward/drop policies have similar delivery probability. This is because at low speeds, vehicles are within each other's range for sufficiently long, and thereby, allowing them to drain their queue. On the other hand, at high speeds, a contact may not be sufficient to transmit even one bundle. Consequently, delivery ratio reduces significantly. In terms of delay, as shown in Figure 4-4 (b), policies that forward newly generated bundles or recently transmitted bundles achieve a low delay. For example, DO, FIFO and HO-COUNT have a delay of 1450, 1590 and 1630 seconds respectively. QM-EBRP trades off delivery ratio and delay such that bundles' expected delay reduces and delivery ratio increases. Figure 4-4 (b) shows that QM-EBRP delivers bundles up to 25% quicker as compared to DO. Policies may deliver a small number of bundles quickly using a small number of hops. In this case, the overhead and delay reduces but the network experiences a low delivery ratio. Figure 4-4 (d) shows the trade-off between delivered bundles and delays. QM-EBRP recorded 60% improvement in terms of

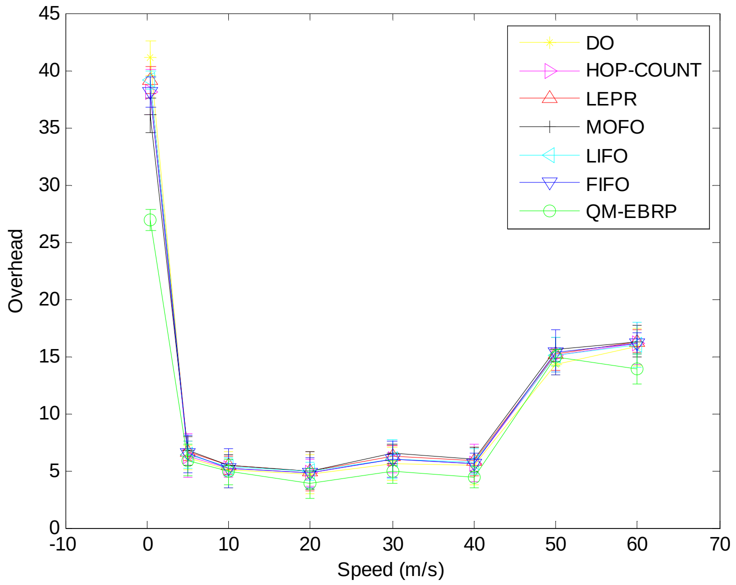
DA. Figure 4-4 (e) shows that QM-EBRP has up to 32% improvement in terms of *DOR*. Also, Figure 4-4 (f) shows that QM-EBRP improves DOA up to 80% .



(a)

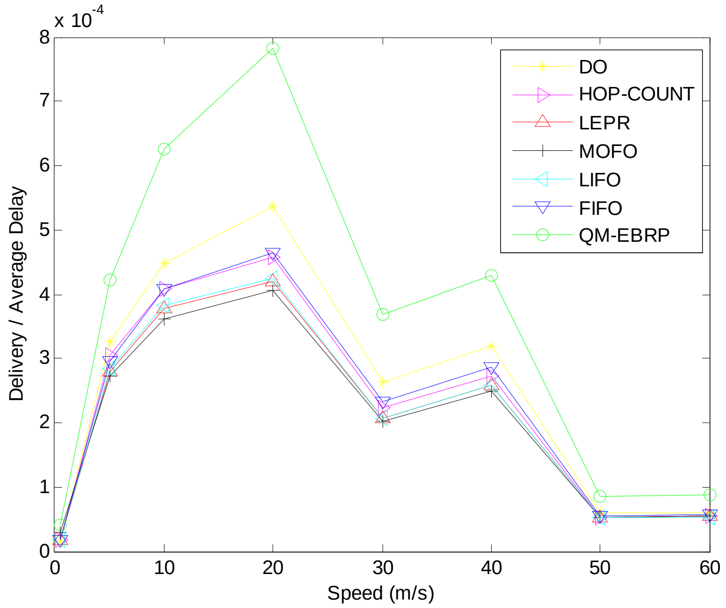


(b)



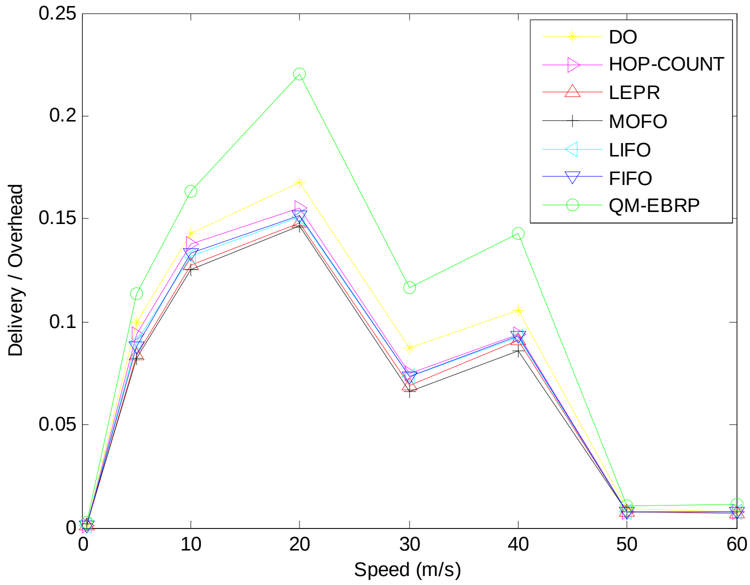
Overhead
Speed (m/s)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(c)

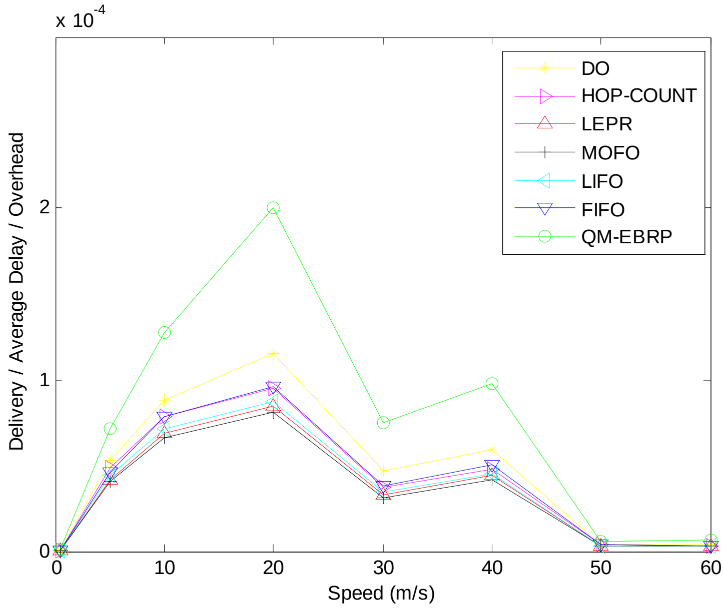


x 10^-4
Delivery / Average Delay
Speed (m/s)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(d)



(e)

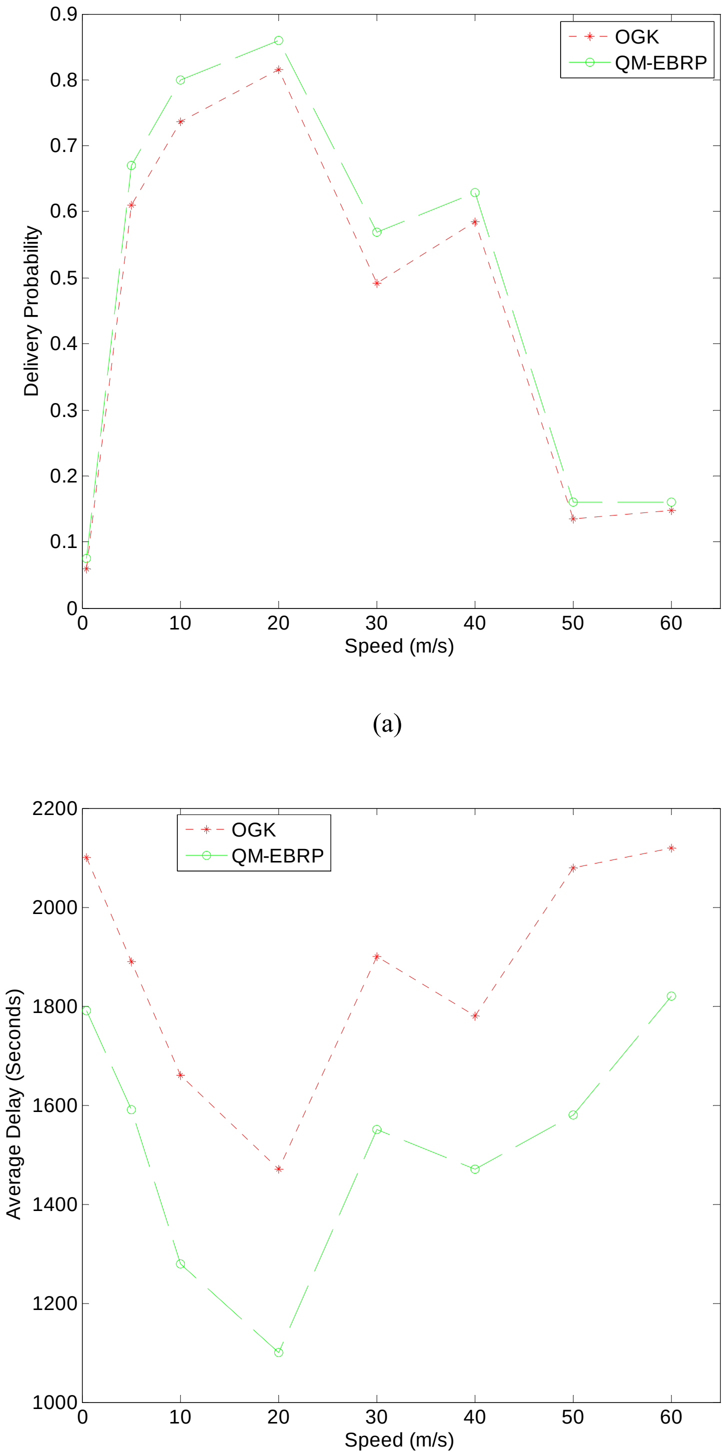


(f)

Figure 4-4 Network performance under the shortest map based mobility with different vehicle speeds, a) delivery probability, b) average delay c) overhead d) DA, e) DOR, and f) DAO

Figure 4-5 shows a comparison of QM-EBRP against OGK. Although OGK does not suffer from inaccurate/obsolete information, it disregards information such as the lifetime of bundles and the encounter rates of nodes. This causes OGK to give a high priority to bundles that have a large number of replicas despite their short lifetime. The results in Figure 4-5 (a) show that QM-EBRP has 10% more delivered bundles.

Also, Figure 4-5 (b) shows that QM-EBRP has up to 25% reduction in delay as compared to OGK.



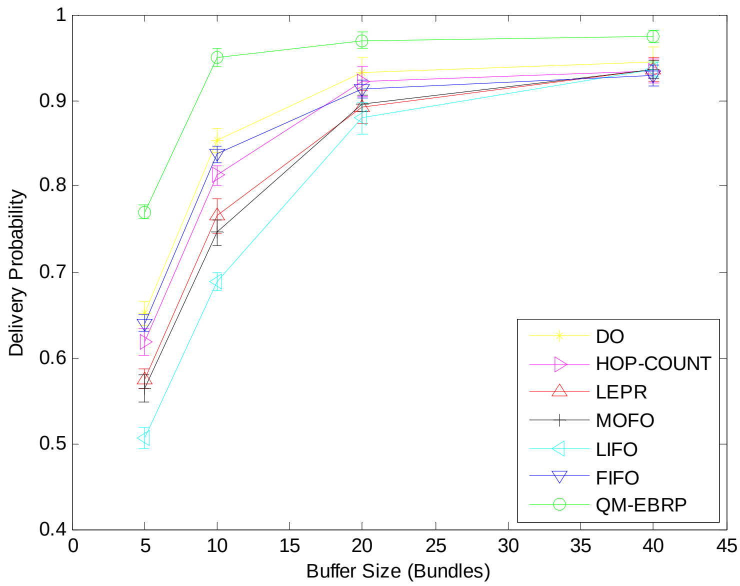
(b)

Figure 4-5 A comparison of QM-EBRP against OGK under the shortest map based mobility with different vehicle speeds, a) delivery probability, b) average delay

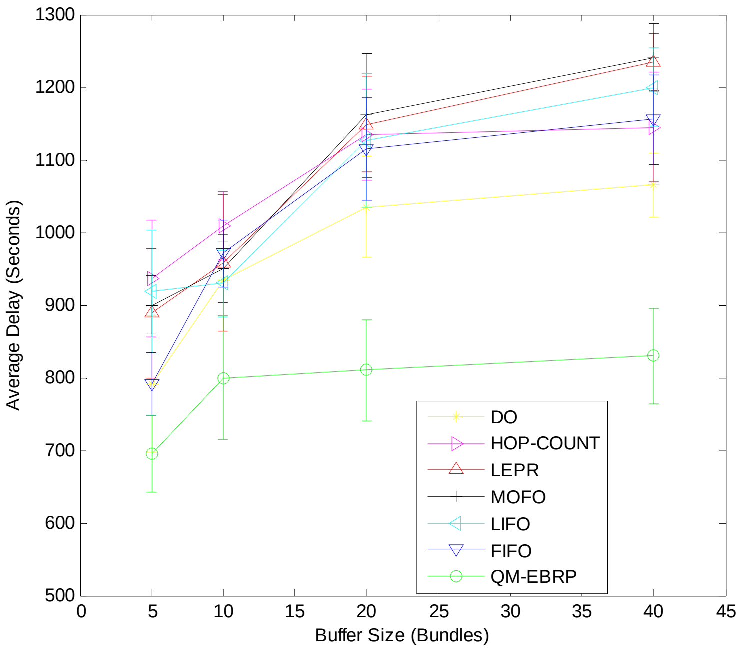
In the next experiment, different buffer sizes are considered. Notice that although increasing nodes' buffer size causes nodes to store more bundles, it can result in a high ratio of dropped bundles when long contacts occur. On the other hand, increasing nodes' buffer size causes nodes to select a larger number of bundles for forwarding over short contacts. QM-EBRP will lower the priority of a bundle with a lower delivery probability and larger delay. Note that a bundle has a low delivery probability if the dissemination rate is low and/or its remaining lifetime is short. Figure 4-6 (a) shows QM-EBRP has up to 12% improvement in terms of delivery ratio as compared to DO. LIFO has the worse delivery ratio with 5% fewer delivered bundles as compared to MOFO and LEPR. This is because LIFO drops recently received bundles. Notice that the delivery ratio gradually increases when nodes' buffer size increases. This is because nodes have the capability to buffer more bundles. This implies that when a node has a small buffer, upon a contact, a majority or whole of buffered bundles will be replaced with received bundles. Hence, with respect to buffer size, as the rate of replacement is high e.g., 90% of the buffer, buffered bundles may not have the chance to remain at a node upon a contact. However, when nodes' buffer size increases, the replacement rate decreases for the following reasons. Firstly, since nodes can store a large number of bundles, the receiver nodes may already have the forwarded bundle. Secondly, since contacts' duration is short, nodes may not be able to transmit all theit forwarding bundles.

Figure 4-6 (b) shows that delivery delay increases when nodes' buffer size increases. This can be explained as follows. Suppose that contacts duration is short. When nodes have a small buffer size, i.e., five bundles, nodes are able to drain their queue. On the other hand, when nodes have a large buffer size, i.e., 20 and 40 bundles, they can only transmit a small portion of queued bundles. In this case, a large number of bundles may not be forwarded for a long time. This results in increased delay. In terms of delay, Figure 4-6 (b) shows that QM-EBRP has up to 16% reduction as compared to DO and up to 23% as compared to FIFO and HOP-COUNT. In terms of overheads, forwarding bundles that have a low delivery probability increases overhead. This is because forwarding these bundles increases the number of relays even though they may not have a chance to be delivered. QM-EBRP addresses this problem by giving a low priority to bundles that have a low delivery probability.

Figure 4-6 (c) shows that QM-EBRP has up to 7% reduction in overhead. To quantify the trade-off between delivery and delay, Figure 4-6 (d) depicts that QM-EBRP has up to 23% improvement in *DA*. Also, Figure 4-6 (e) shows the trade-off between delivery and overhead that QM-EBRP has up to 22% improvement in *DOR*. In terms of the trade-off between delivery, delay and overhead, Figure 4-6 (f) shows QM-EBRP has up to 30% improvement in terms of *DAO*. As mentioned, this is obtained because QM-EBRP simultaneously takes advantage of parameters such as bundle's TTL, number of bundle's replicas, and node's encounter rate that have high impact in predicting bundle's delivery probability. Other methods use one of these metrics as an estimation of delivery probability.

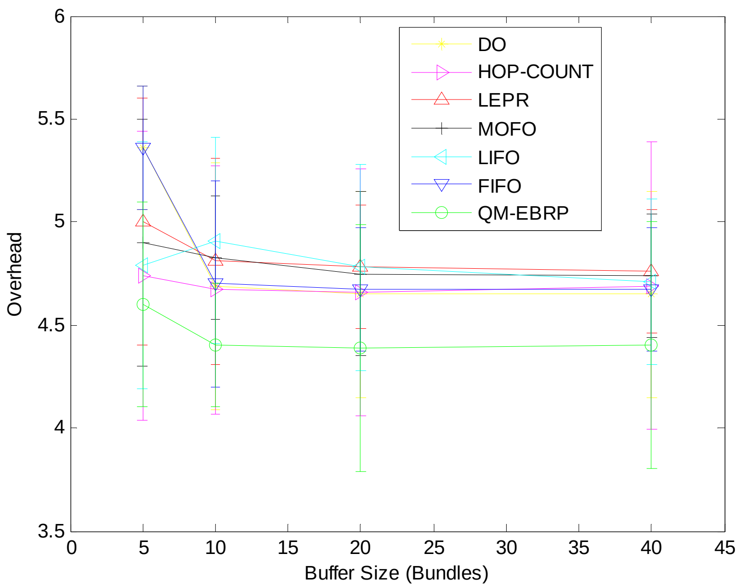


(a)



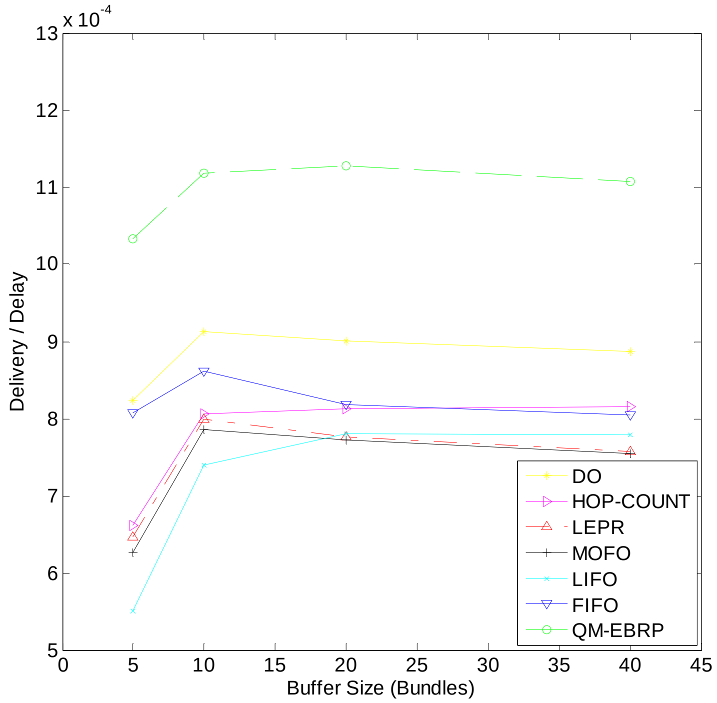
Average Delay (Seconds)
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(b)

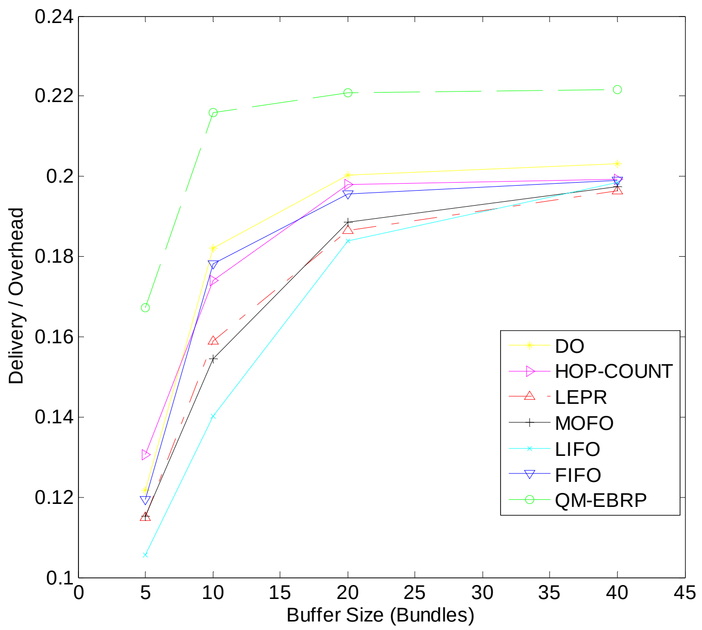


Overhead
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

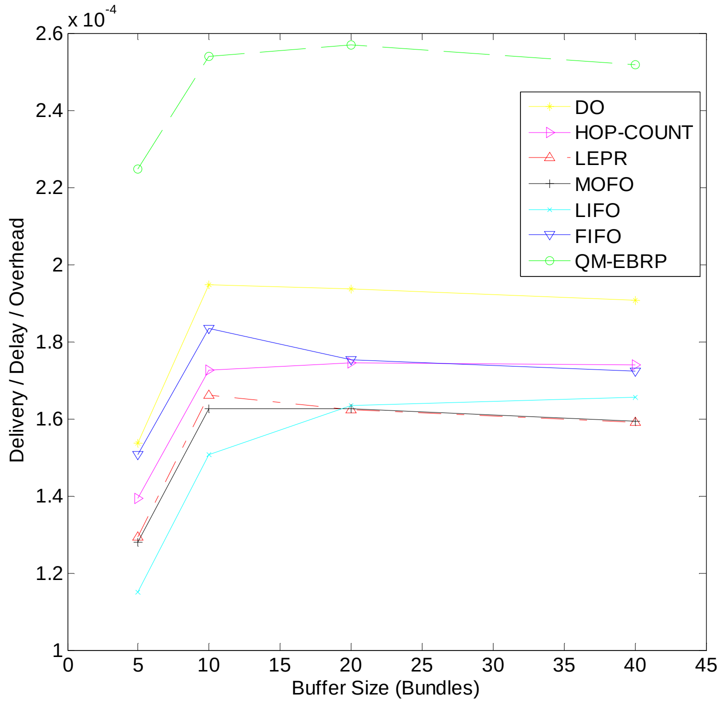
(c)


x 10^-4
13
12
11
10
9
8
7
6
5
Delivery / Delay
0
5
10
15
20
25
30
35
40
45
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(d)


0.24
0.22
0.2
0.18
0.16
0.14
0.12
0.1
Delivery / Overhead
0
5
10
15
20
25
30
35
40
45
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

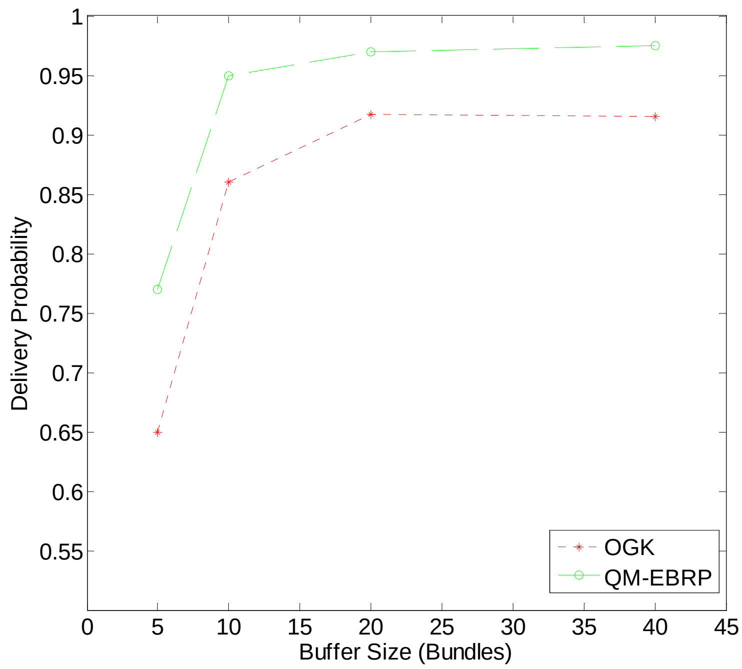
(e)



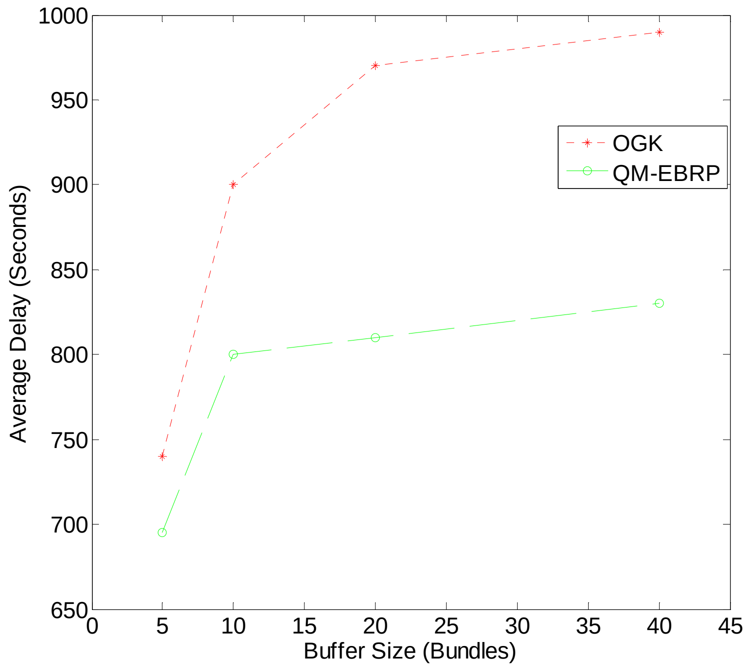
(f)

Figure 4-6 Network performance under shortest map based mobility with different vehicle buffer sizes, a) delivery probabilities, b) average delays c) overheads, d) DA, e) DOR, and f) DAO

Figure 4-7 compares QM-EBRP against OGK. In terms of delivery, Figure 4-7 (a) shows that QM-EBRP has up to 12% improvement. As mentioned earlier, when the buffer size of nodes increases, a large number of bundles may not be forwarded for a long time. However, OGK does not consider the expected delay when forwarding bundles. Hence, bundles experience a large delay of 990 seconds. The performance of OGK versus QM-EBRP exhibit a similar trend for the forthcoming mobility models. We thus omit them from the rest of the simulated scenarios.



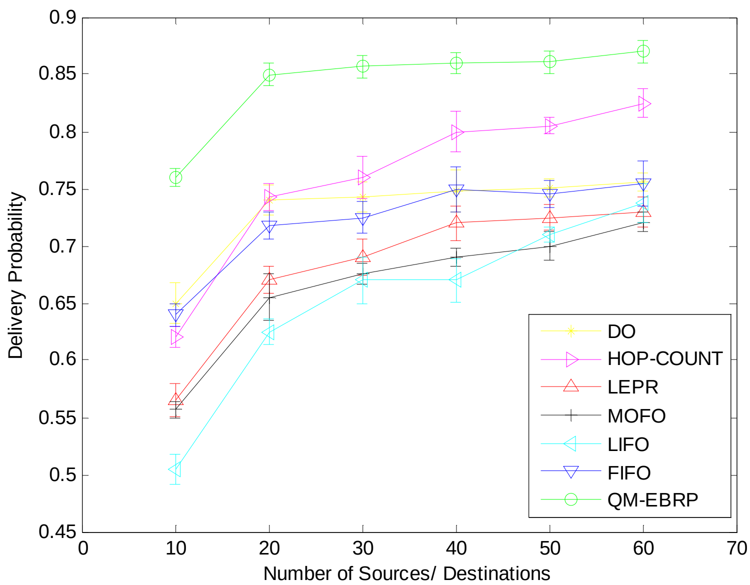
(a)



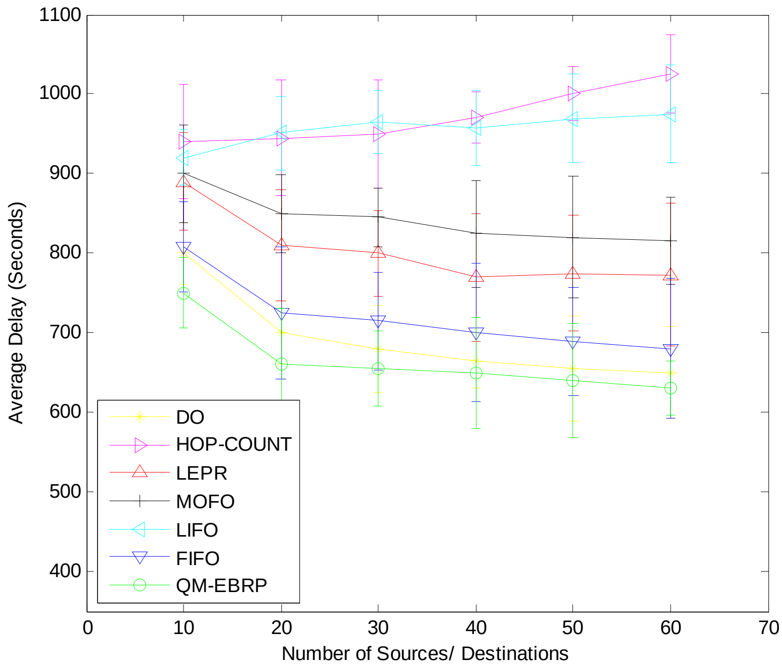
(b)

Figure 4-7 A comparison of QM-EBRP and OGK under the shortest map based mobility with different vehicle speeds, a) delivery probabilities, b) average delays

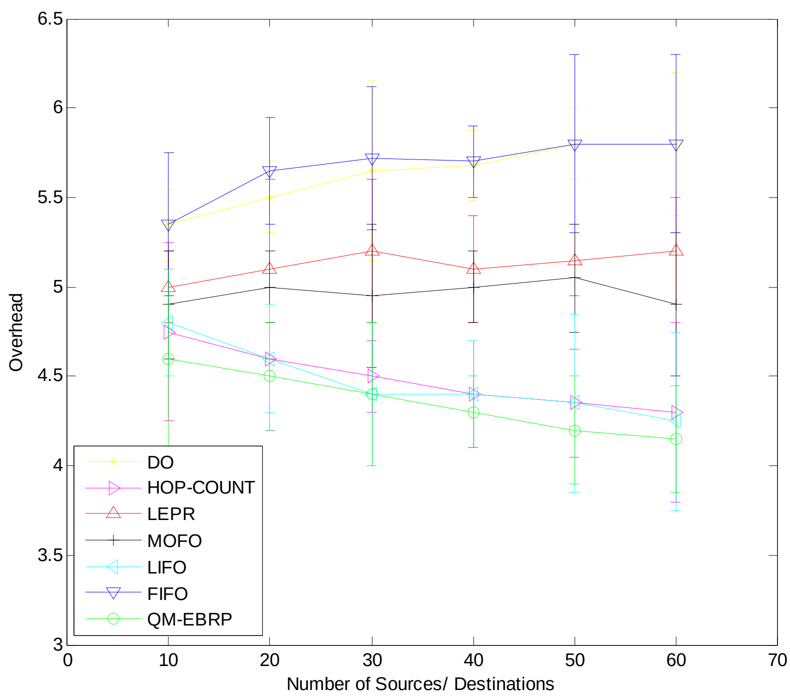
Figure 4-8 shows the impact of different numbers of source/destination nodes. Suppose that only one destination exists in the northern part of a city and the source is in the southern part of the city. Hence, nodes forward bundles towards the northern part of the city and consequently nodes in that area experience a high load, and thus drop bundles frequently. This example illustrates the downside of forwarding all bundles towards a small number of destinations i.e., 10. Indeed, in these experiments, we see protocols have low delivery ratios and large delays. For example, DO, FIFO and HOP-COUNT have a delivery ratio of 65%, 64% and 62% respectively. Now, suppose there are multiple, geographically dispersed destination nodes. This means traffic will be distributed uniformly across the network. Hence, when the number of destinations increases, the drop ratio of bundles decreases, resulting in a higher delivery ratio and smaller delays. Furthermore, destination nodes may not be reachable within a bundle's lifetime. To address the said issues, QM-EBRP takes advantage of nodes' encounter rate, bundle life time and number of bundle replicas to effectively consider how likely one of the bundle's replicas will be delivered within the bundle's lifetime. As shown in Figure 4-8 (a), as compared to HOP-COUNT, DO and FIFO, QM-EBRP has up to 17% improvement in delivery ratio and also up to 7% reduction in delay. In terms of *DA*, Fig. 8(d) shows that QM-EBRP has up to 24% improvement. Also, Figure 4-8 (f) shows that QM-EBRP has up to 60% improvement in terms of *DAO*.



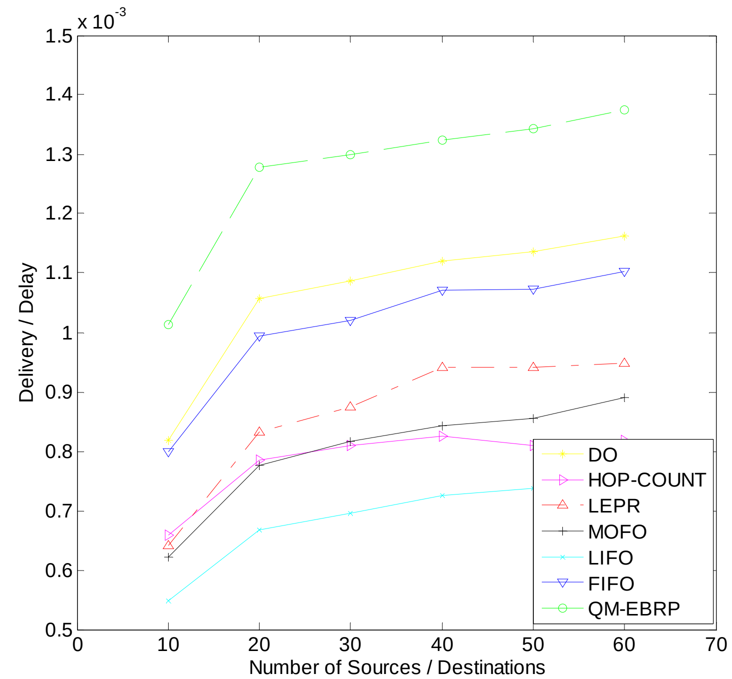
(a)


Average Delay (Seconds)
Number of Sources/ Destinations
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

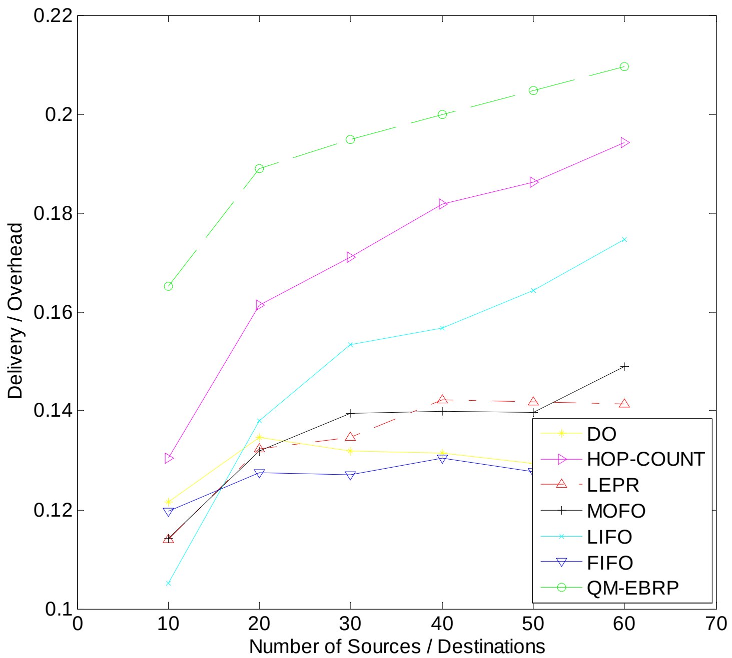
(b)


Overhead
Number of Sources/ Destinations
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

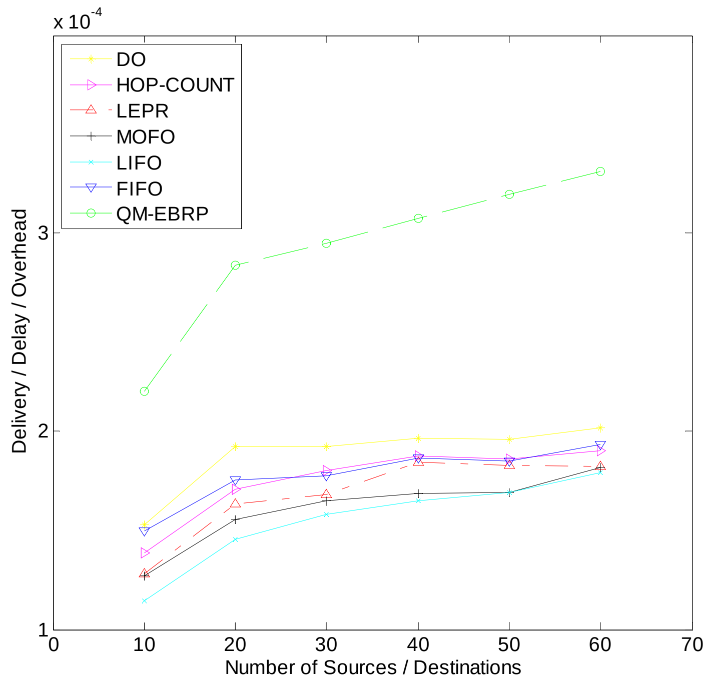
(c)


x 10^-3
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
Delivery / Delay
0
10
20
30
40
50
60
70
Number of Sources / Destinations
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(d)


0.22
0.2
0.18
0.16
0.14
0.12
0.1
Delivery / Overhead
0
10
20
30
40
50
60
70
Number of Sources / Destinations
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(e)



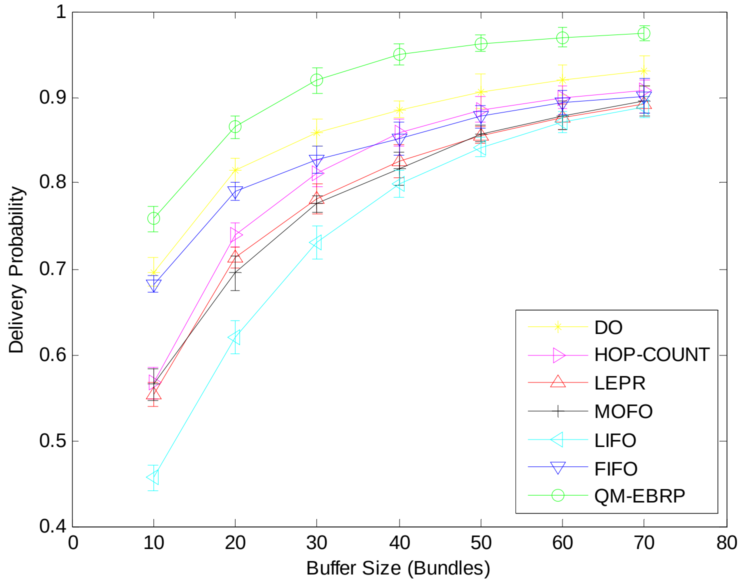
(f)

Figure 4-8 Network performance under shortest map based mobility with different number of source/destination pairs. a) delivery probability, b) average delay c) overhead d) DA, e) DOR, and f) DAO

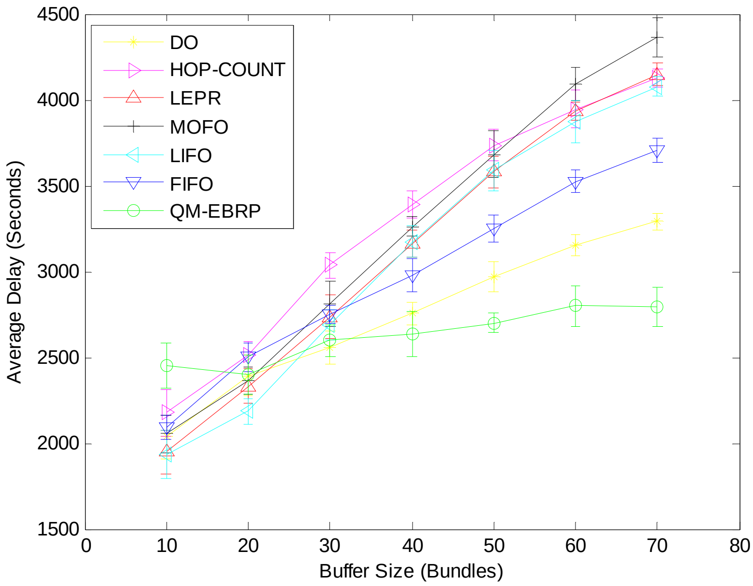
4.4.2 Working Day Movement Model

Figure 4-9 depicts the network performance when nodes have different buffer sizes. The simulation duration and bundles' TTL are increased based on working hours to ensure every bundle has enough time to be delivered. Notice that bundles' lifetime directly impacts delivery ratio. Accordingly, the policies that consider bundles' lifetime have a high delivery ratio. For example, DO delivers 70% of bundles when nodes have a buffer size of 10 bundles. FIFO also indirectly considers bundle's TTL such that new arrival bundles are sent upon contact. The results in Figure 4-9 (a) show that FIFO delivers 69% of the total bundles. Similar to Section 4.4.1, QM-EBRP takes advantage nodes' encounter rate. Figure 4-9 (a) shows that QM-EBRP has up to 10% improvement in terms of delivery ratios. As for delays, Figure 4-9 (b) depicts that QM-EBRP recorded a 20% drop. Figure 4-9 (c) shows QM-EBRP has 10% less overheads. In terms of trade-off between delivered bundles and delays,

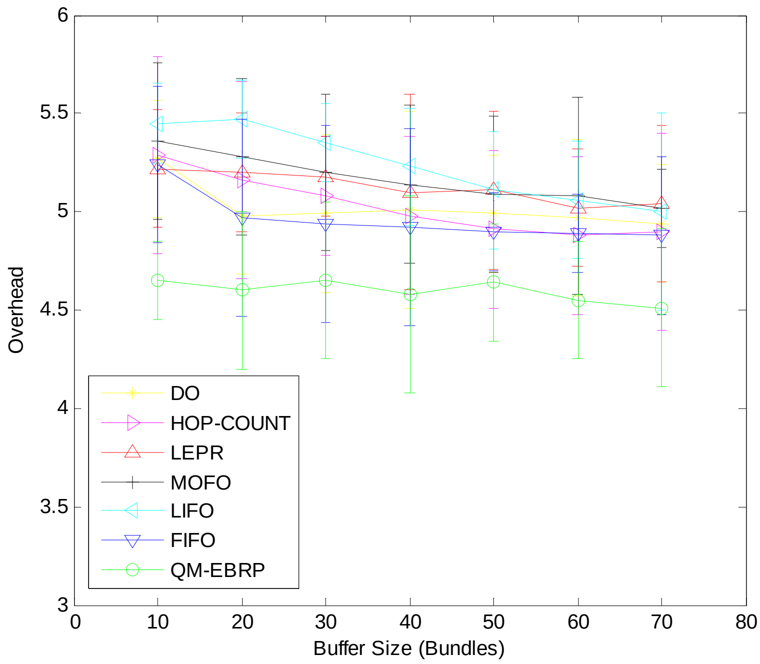
Figure 4-9 (d) shows that QM-EBRP has up to 30% improvement. In total, QM-EBRP achieves up to 35% improvement. Comparing the results of Section 4.4.1with this section show that QM-EBRP outperforms other methods as the estimation of the future network performance is accurate when the nodes' mobility pattern is at least semi-predictable. As an example, when nodes' mobility pattern is predictable, considering history of nodes' encounter rate is a good prediction of future contacts [30].



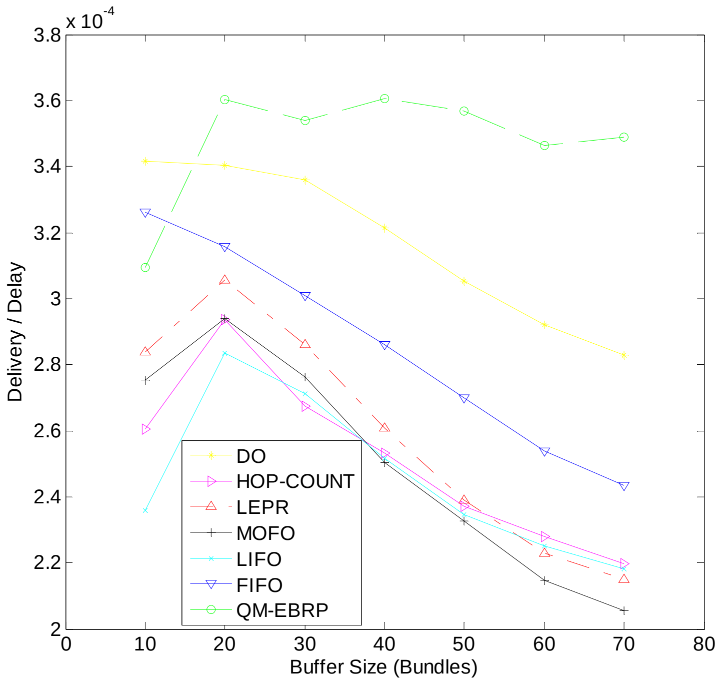
(a)



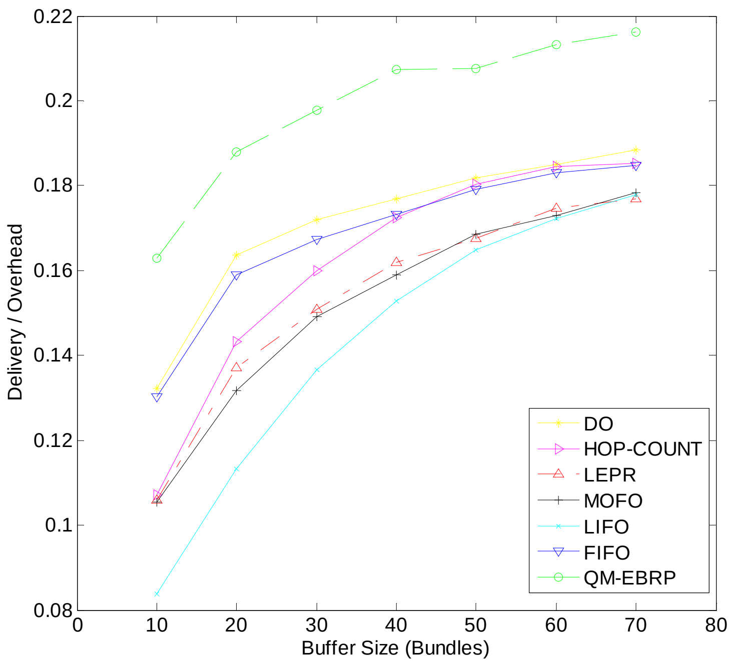
(b)


Overhead
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

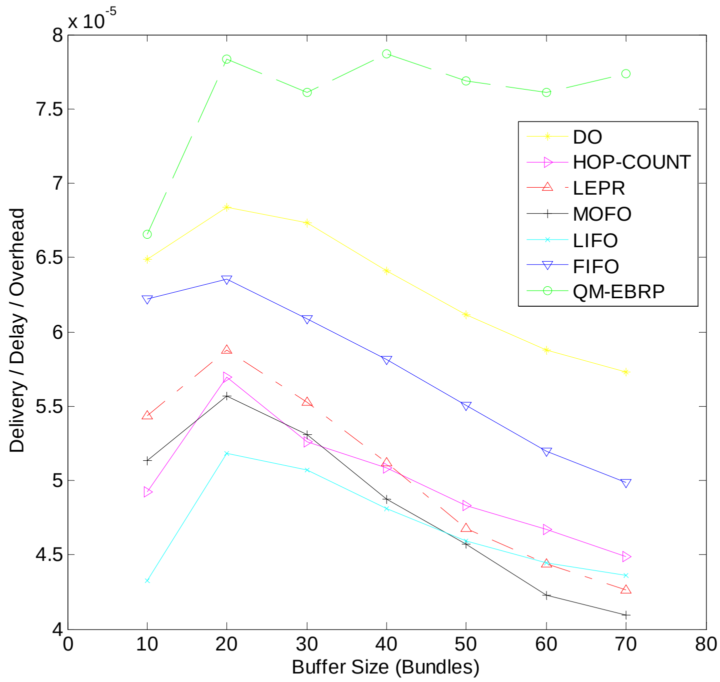
(c)


x 10^-4
Delivery / Delay
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(d)



(e)

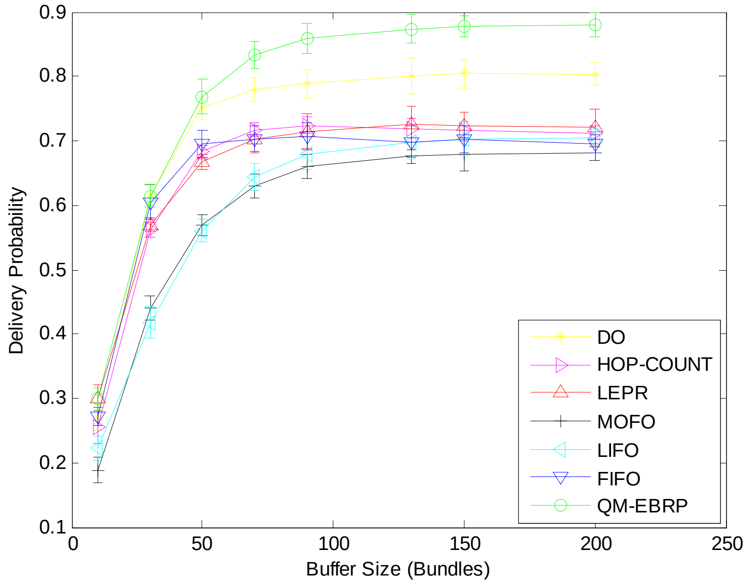


(f)

Figure 4-9 Network performance under working day movement model with different vehicle buffer sizes, a) delivery probability, b) average delay c) Overhead d) DA, e) DOR, and f) DAO

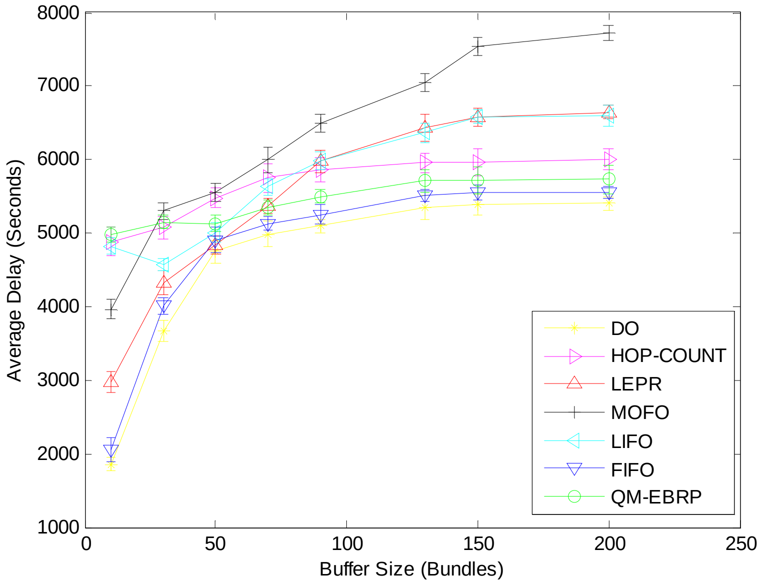
4.4.3 Random Mobility Model

This section considers a mobility model whereby nodes are unable to predict future contacts via their encounter rates. Now, suppose that a large number of nodes are randomly located in an area and meet each other frequently for a short period of time. In this case, the nodes' encounter rate increases but nodes may not meet each other in the future as nodes do not follow any predetermined paths. Hence, nodes' encounter rate will be obsolete/inaccurate for future decisions. In this respect, QM-EBRP relies on other parameters such as number of replicas and their TTL to prioritize bundles. The simulation results in Figure 4-10 (a) show that in terms of delivery, QM-EBRP has up to 10% improvement as compared to DO, and up to 27% improvement as compared to LEPR, HOP-COUNT, MOFO, LIFO and FIFO. In contrast, MOFO has the lowest delivery ratio at 65%. This is because MOFO considers delivery probability of bundles based on nodes' encounters, which is highly inaccurate in this mobility model. In terms of delay, Figure 4-10 (b) shows that QM-EBRP has a delay of 5050, 5400 and 5500 seconds when nodes' buffer size is 30, 90 and 200 bundles respectively. Notice that using nodes' encounter rate under a random mobility model causes inaccurate expected delay calculation. However, QM-EBRP also considers the number of disseminated replicas to estimate how likely a bundle will be delivered. Consequently, as compared to LIFO and LEPR, QM-EBRP has up to 16% reduction in delay and up to 30% reduction in delay as compared to MOFO. In terms of *DAO*, Figure 4-10 (f) shows that QM-EBRP has up to 10% and 36% improvement respectively as compared to DO and FIFO.



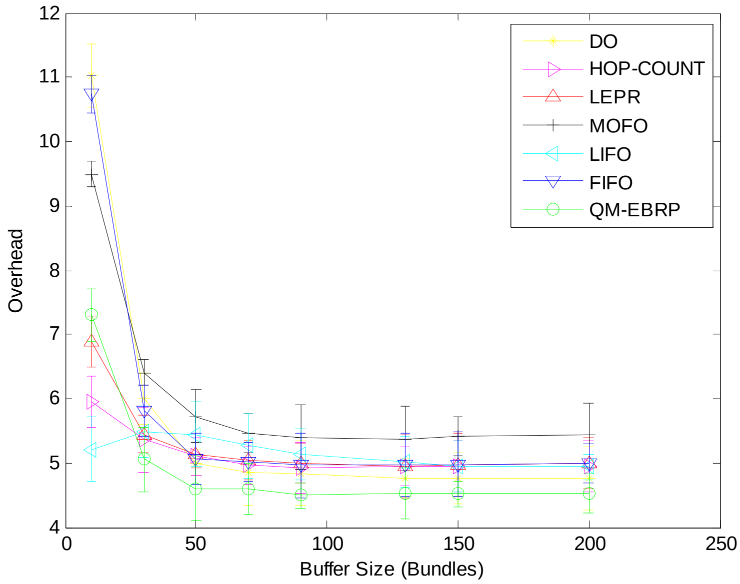
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Delivery Probability
0
50
100
150
200
250
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(a)

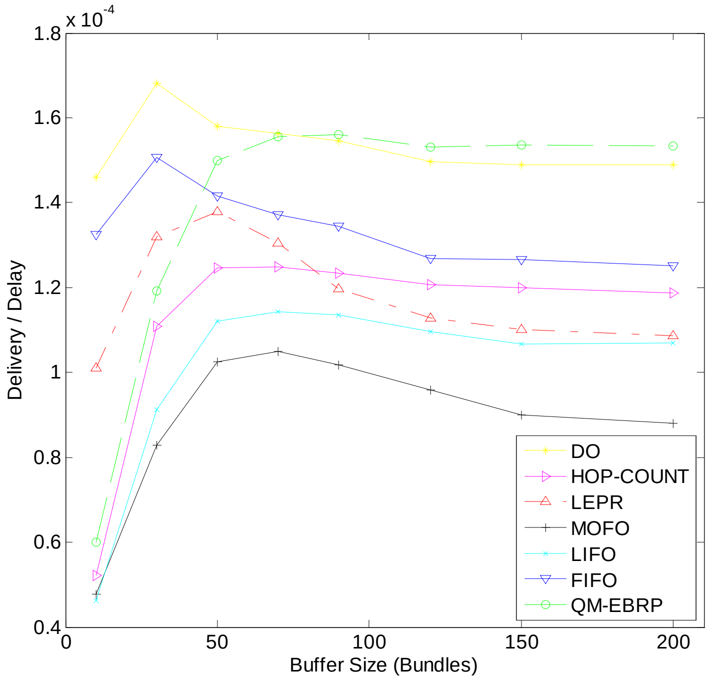


8000
7000
6000
5000
4000
3000
2000
1000
Average Delay (Seconds)
0
50
100
150
200
250
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

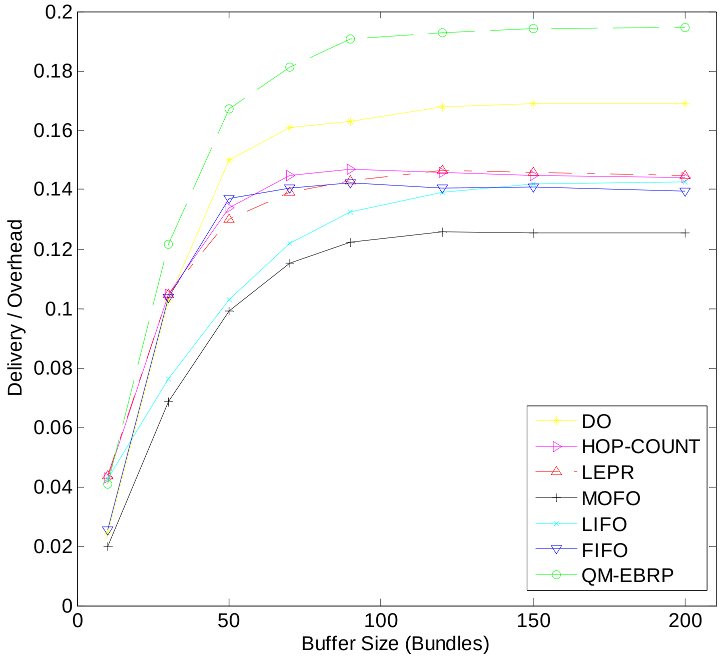
(b)


Overhead
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

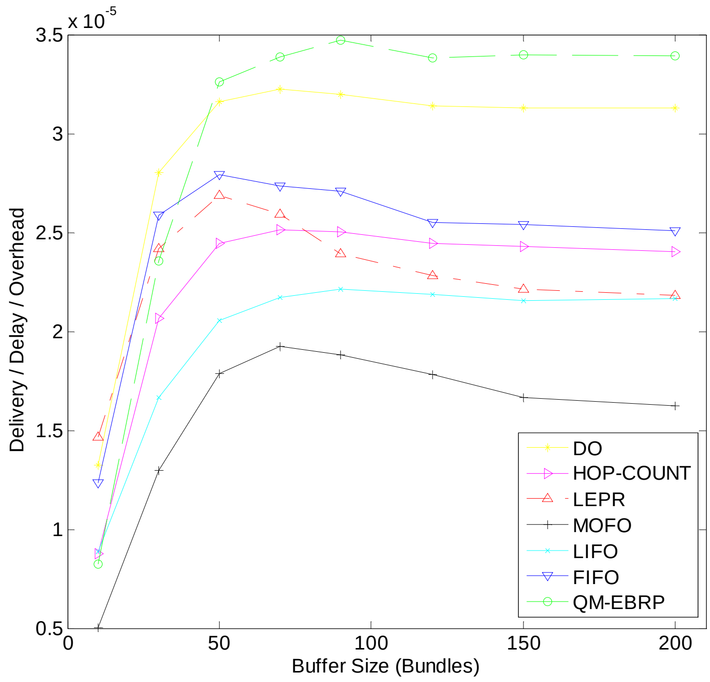
(c)


x 10^-4
Delivery / Delay
Buffer Size (Bundles)
DO
HOP-COUNT
LEPR
MOFO
LIFO
FIFO
QM-EBRP

(d)



(e)



(f)

Figure 4-10 Network performance under random mobility model with different vehicle buffer sizes, a) delivery probability, b) average delay c) overhead d) DA, e) DOR, and f) DAO

4.4.4 Discussion

The obtained results suggest that QM-EBRP performs well across all tested scenarios. They confirm QM-EBRP effectively use the combination of parameters available locally at each node; namely, a node's encounter rate, bundle's lifetime and number of replicas of a bundle. Indeed, QM-EBRP outperforms other tested policies in terms of both delivery ratio and delay. The reasons that policies such as FIFO, LIFO, LEPR and MOFO perform poorly are their reliance on metrics such as encounter rates or arrival time of a bundle only, which cause these policies to (i) forward bundles that may have insufficient remaining lifetime to be delivered, (ii) drop bundles with a long remaining lifetime, or (iii) drop bundles that have a large number of replicas. In terms of the trade-off between delivery ratio and delay, QM-EBRP outperforms other tested policies. This is because, in the calculation of a bundle's utility, delivery ratio and delay are considered together. However, QM-EBRP is not effective in reducing delays under the random mobility model. Recall that QM-EBRP uses nodes' encounter rate in the calculation of a bundle's utility, which helps estimate how likely a bundle will be delivered in the future and also its expected delay. However, in the random mobility model, a node that has a high rate of encounter rate will not necessarily be reachable in the future. It should be noted that approximating the delivery probability via a cumulative distribution results in inaccurate estimation. This is because the delivery probability distribution function is not accurately predictable over time as it is highly dependent on nodes' mobility pattern.

4.5 Conclusion

This chapter has investigated a novel bundle drop/forward policy for encountered-based quota protocols in DTNs. A multi objective function is proposed that estimates the delivery ratio and delay of a bundle based on local network information such as encounter rate, remaining time to live, and number of replicas. This is in contrast to current queue management policies that require global information. Then, the rate of change of both bundle delivery ratio and bundle delivery delay is calculated simultaneously. Finally, the proposed policy, QM-EBRP, which uses the resulting multi-objectives function, optimizes the global delivery ratio and delay by prioritising bundles during contacts.

The performance of QM-EBRP is evaluated over a wide range of scenarios that consider different mobility models and buffer sizes and speeds. The simulation results showed under shortest map based mobility, QM-EBRP achieved up to 40% improvement in *DAO* when vehicles have infinite buffer space and up to 30% when vehicles have different buffer size over current state of the arts policies such as Drop Oldest (DO), Last Input First Output (LIFO), First Input First Output (FIFO), Most Forwarded first (MOFO), LEast PRobable first (LEPR), and drop greatest HOP-COUNT. Also, under a working day movement, QM-EBRP performed up to 35% better in *DAO* when vehicles have different speeds as well different buffer size.

Both Chapters 2 and 3 have investigated protocols designed to improve network performance when contacts are semi-predictable. However, these protocols are not efficient when the network topology is deterministic and contacts are completely predictable. This is because they do not consider time of contacts in their forwarding metric. In the following chapter, a routing protocol is proposed that takes advantage of nodes' mobility pattern in semi-deterministic DTNs. In addition, the proposed method considers space-time graph with expiration time.

Chapter 5

A Mobility Based Routing Protocol (MBRP) for Deterministic DTNs

5.1 Introduction

The previous chapters have investigated DTNs where the network topology is not completely predictable. In this chapter, we investigate a form of DTNs where nodes have scheduled contacts [103] or remain on a given predictable path for some period of time. For example, in transport networks [9], buses and trams have well defined movement, meaning their mobility pattern or trajectory will not change over time. In contrast, for a given time period, taxis have a mobility pattern that is valid for a short time period; e.g., when they are carrying passengers to their destination. After passengers arrive at their destination, the taxis will set a new trajectory or path. Consequently, in these networks, bundles will be forwarded on a predetermined route based on scheduled contacts within a given period of time. Thus, within this period, nodes have a known trajectory that allows other nodes to determine point of contacts and their duration. This information can then be used to compute different paths that meet varying criteria. For example, bundles can be delivered through routes with the minimum delay or number of hop counts. Moreover, it is possible to determine the remaining time until a pair of nodes meets each other again. Also, contact duration can be computed, and thereby, allowing nodes to determine the amount of data that can be transferred in advance.

A key concept employed in the said DTNs is space-time graph. A space-time graph is defined as a graph that shows the sequence of network connectivity over time. Figure 5-1 depicts a DTN comprising of five nodes N={n_1, n_2, n_3, n_4, n_5}. Note that nodes located in the same cell are in contact with each other. Figure 5-1(b) shows the corresponding space-time graph over three time slots, $\boldsymbol{t}=\{1,2,3\}$. In particular, the graph depicts the connectivity of nodes over three time intervals. More importantly, using the resulting space time graph, we can find routes from a sender to a destination. As an example, a bundle at n_1 can be delivered to n_2 via n_4 and n_5 within three time slots. From this example, we see that in order for nodes to construct a space-time graph [36, 58, 59, 104], they will need to learn the mobility pattern of every encountered node. In addition, once nodes have the resulting space-time graph, they can easily compute routes towards destinations that meet criterion such as minimum delay or hop-count.

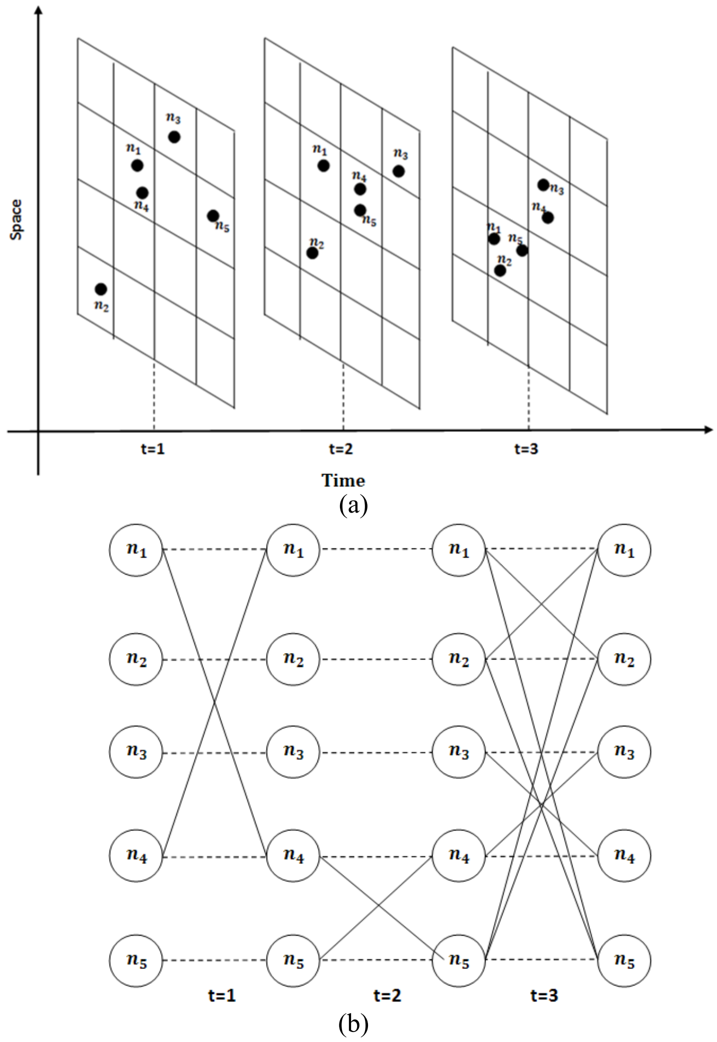


Figure 5-1 A time-evolving DTN, a) time-evolving topologies of a DTN (a sequence of snapshots), b) corresponding space-time graph

To date, the key assumption, see Section 2.1.3, of current routing protocols that rely on space-time-graph is that nodes are aware of the mobility of all nodes. That is, they are pre-installed with a space-time graph. Consequently, the main research question is to compute a suitable route that meets a given criterion. In practice, nodes will have to gradually learn the mobility pattern of nodes upon each contact and update their space-time graph accordingly. Consider Figure 5-2.We see a deterministic DTN where a tram and a taxi move according to a pre-determined path [105]. The tram has a fixed mobility pattern within an unspecified time period. In contrast, the taxi may have a different mobility pattern for each time period. For example, the taxi in Figure 5-2 shows it moving via three point of interests: POI_1, POI_2 and POI_3. In the time period between POIs, its trajectory is fixed. We can assume each POI to be the destination of passengers. Upon arrival, the taxi will form a new trajectory or path. Consequently, when a node builds a space-time graph based on the current mobility pattern of the taxi, the space time graph is valid up to the time that the taxi arrives at a POI. Notice that whenever the taxi sets a new trajectory, the space-time graph needs to be updated. Hence, in the time period in which nodes are still learning this new trajectory, we have an *incomplete* space-time graph. To further illustrate, assume that the tram has the taxi's mobility pattern for the time period between the taxi's current location and POI_1. In this case, when the taxi arrives at POI_1 and selects POI_2, the possible contact points between the taxi and trams are at c_1, c_2, c_3, c_4 and c_5. These new points thus need to be made known to the tram. In general, after any change in a node's mobility pattern, the node has to broadcast its new mobility pattern to update other nodes.

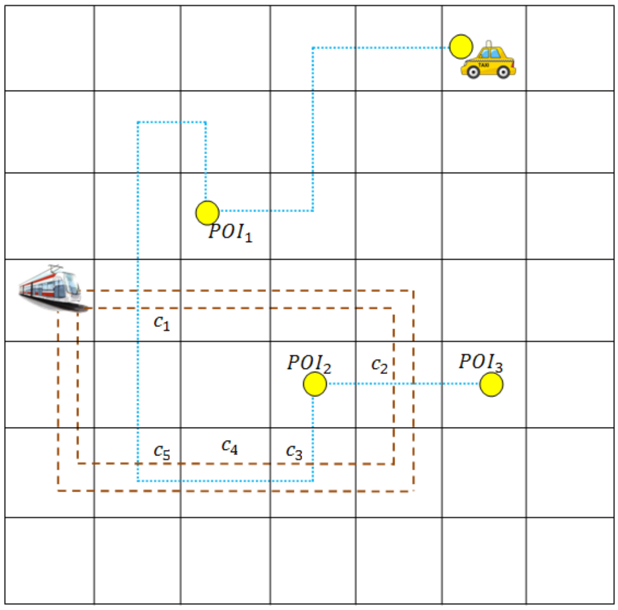


Figure 5-2 A DTN comprising of a taxi and a tram that have heterogeneous mobility patterns

Based on the said scenario, this chapter considers the following limitations. Suppose that while nodes are learning the space-time graph, a source node generates a bundle. In this case the source node may face the following problems. First, the space-time graph may not be complete, meaning there may not be a route towards the bundle's destination. Recall that a space-time graph will become incomplete or staled when nodes such as taxis form a new trajectory or path. Another reason is because a node may not have sufficient contacts to learn the trajectory of all nodes. Notice that if the time period between POIs is short, the space-time graph will become staled quickly. Consequently, unlike prior works that assume a fixed space time graph we consider the issue of learning the space-time graph dynamically. Secondly, when a sender node has a number of routes towards a destination, these routes may not be optimal. Using any of these routes may impose a large delay/overhead as compared to the optimal route. Hence, this chapter also considers route optimality issue when forwarding bundles.

In both the aforementioned problems, many history based routing protocols, such as PROPHET [5], MaxProp [9], EBR [30], can be applied when nodes are learning the space-time graph. However, current history based routing protocols are designed for social based networks where people usually roam in relatively small regions. This is because they assume the mobility pattern of nodes correspond to some form of relationship. In this work, nodes will have independent mobility pattern or may have dependent mobility pattern for a period of time. Under such condition, current history based protocols do not work efficiently. This is because previous encounters may not be a good estimate of future contacts.

To address the aforementioned limitations, the proposed protocol takes advantage of the following observation. Consider vehicle A that is moving from POI_1 to POI_2 in a given hour. Also, vehicle B moves from POI_3 to POI_4 in a two hours period. Now assume 45 minutes have since elapsed and both vehicles have encountered a number of vehicles except destination D. As an example, assume vehicles A and B have learned 20 and 10 mobility patterns respectively. In this case, vehicle A is in a high density area or had contacts with many nodes. Now, assume source vehicle S encounters vehicles A and B. At this time, vehicle B would make a good bundle carrier for destination D if the remaining length of all trajectories learned by vehicle B is longer than those of vehicle A. This implies that vehicle S will be able to predict subsequent contacts reliably. To this end, vehicle S forwards more replicas to vehicle B.

Henceforth, based on aforementioned observations, this chapter presents a mobility based routing protocol (MBRP) that tackles the problem of routing in deterministic DTNs where nodes have scheduled contacts [103] or remain on a given predictable path for *some* period of time. Contrary to current space-time graph protocols, MBRP assumes that each node's trajectory or mobility pattern has an expiration time. In MBRP, while nodes are learning the space-time graph, a number of routes may exist in the current space-time graph. In this case, MBRP runs a forwarding strategy called "space-time phase" that sends a single copy of a bundle towards its destination via the fastest route to date. On the other hand, as the space-time graph may be incomplete, these routes may not be optimal in terms of delay. To overcome this issue, MBRP proposes a forwarding strategy called "heuristic phase" that evaluates the reachability of encountered nodes based on their mobility pattern in order to determine the number of replicas to forward.

The remainder of the chapter is organized as follows. Section 5-2 models the system. Section 5-3 describes the problem. Section 5-4 presents the simulation set-up. This is then followed by obtained experimental results in Section 5-5. Finally, Section 5-6 concludes this chapter.

5.2 System Model

Consider a DTN with *v* mobile nodes represented by the set ***N***=$\{n_1, \ldots, n_v\}$. Every node is equipped with a GPS unit and moves independently with a different speed and has a radio range of R. Nodes are assumed to have unlimited buffer. Also nodes have a semi-deterministic mobility pattern, meaning they visit a sequence of locations in a predictable manner for a given time period. The term "*cycle*" is used to denote nodes that repeat their mobility pattern. For example, a person may leave his/her home at 7:00am, go to work and return home at 10pm every day. He/she then repeats this routine every day; i.e., they have a cycle of 24 hours. Nodes move on a grid with $w \times w$ cells. Each cell size is 2×R. This means if two nodes are located in a cell, they are in communication range of one another. Let C= $\{c_1, \ldots c_i, \ldots, c_m\}$ be the set of all cells, where $m=|C|= w \times w$. As an example, a DTN that is overlayed on a grid of size 4×4 has 16 cells C= $\{c_1, \ldots, c_{16}\}$. Another key assumption is that time is discrete and it is divided into slots of equal length, denoted as $\boldsymbol{t}$ = $\{1, \ldots, T\}$. Moreover, nodes are synchronized in time, which can be achieved via GPS. Based on the space and time information, every node *a* records its mobility pattern MP_a as a sequence of *ordered pairs* (c_i, t), where (c_i, t) denotes cell *i* and time *t*. For example node *a* may have the following mobility pattern within five time slots t=5, $MP_a = \{(c_5, 1), (c_4, 2), (c_6, 3), (c_2, 4), (c_1, 5)\}$. Node *a* is called the "owner" of MP_a. In addition, each mobility pattern of node *i* has an expiration time EXP_i. Let RT_i be the routing table of node n_i. The notation $RT_i.MP_a$ is used to denote the mobility pattern of node *a* in node *i*'s routing table as. Also, let *L(t)* be the set of contacts at time slot t.

To capture node contacts at different points in time as well as represent the routing table maintained by nodes, a space time graph is used, denoted as $G(t) = (N, L(t))$, where $t = \{1, \ldots, T\}$. There are two types of links in a space time graph: *spatial* and *temporal*. A spatial link is a directed link between two nodes if they meet each other at the same time *t*. For example, n_i has a spatial link to n_j in *G(3)* if n_j is located in the same cell as n_i at time slot *3.* This means a bundle can only be forwarded from one node to the other through a spatial link. Temporal links (dotted links) on the other hand capture the connection of the same node n_i across the (t-1)-*th* and t-*th* time slots. Every node is connected to itself in every slot, implying it can carry a bundle over all time slots. Nodes are located in one of the cells over time, i.e., $MP_{n_3} = \{(c_3, 1), (c_4, 2), (c_7, 3)\}$. We see from Figure 5-1(a) that the DTN topology changes over time. Figure 5-1(b) shows the corresponding space-time graph over three time slots, $\boldsymbol{t}$=*{1,2,3}*. Horizontal links (dotted lines) and vertical links represent temporal and spatial links, respectively. From the resulting space time graph, see Figure 5-1(b), we can find routes from a sender to a destination. As an example, a bundle at n_1 can be delivered to n_2 via n_4 and n_5 within three time slots.

5.3 The Problem

In past works such as [58, 59, 104], the authors assume that nodes are pre-loaded with a space-time graph that allow nodes to compute a path that meets a given condition; e.g., the foremost path. However, in practice, nodes will have to construct a space-time graph based on contacts whilst attempting to deliver bundles. Hence, before learning the complete space-time graph, if a source node generates a bundle for a given destination, it is faced with one of the following forwarding problems: (i) there is no route to a given destination. This means a source has to either wait until a route is available, which incurs delays that may exceed a bundle's expiration time, or (ii) there is at least one route to the given destination. Here, a source needs to decide whether to use available routes, which may be sub-optimal or wait for a better route in the future that has less delay. Henceforth, the key challenge is how to forward bundles based on incomplete routing table information while nodes are learning the space-time graph such that the delivery ratio is maximized and delay is minimized.

The key challenges are highlighted using the example shown in Figure 5-3. Six nodes *A, B, C, D, E* and *F* have the following mobility pattern over a grid of size 5×5 cells:

$$\boldsymbol{MP_A}=\{(c_{2,5},1),(c_{2,4},2),(c_{2,3},3),(c_{2,2},4),(c_{2,1},5)\}$$
$$\boldsymbol{MP_B}=\{(c_{5,2},1),(c_{4,2},2),(c_{3,2},3),(c_{2,2},4),(c_{1,2},5)\}$$
$$\boldsymbol{MP_C}=\{(c_{4,1},1),(c_{4,2},2),(c_{4,3},3),(c_{4,4},4),(c_{4,5},5)\}$$
$$\boldsymbol{MP_D}=\{(c_{1,5},1),(c_{1,4},2),(c_{1,4},3),(c_{1,3},4),(c_{1,2},5)\}$$
$$\boldsymbol{MP_E}=\{(c_{3,4},1),(c_{3,3},2),(c_{4,3},3),(c_{5,3},4),(c_{5,3},5)\}$$
$$\boldsymbol{MP_F}=\{(c_{4,1},1),(c_{5,1},2),(c_{5,2},3),(c_{5,3},4),(c_{5,4},5)\}$$

Figure 5-3 also indicates the contact time for each node. Assume at each contact, nodes exchange their routing table if it is new. In this example, nodes *F* and *C* meet each other at t=1 and they exchange their mobility pattern. Nodes *B* and *C* meet each other at t=2 and node *C* sends MP_C and MP_F to node *B*, and node *B* sends MP_B to node *C*. At t=3, node *C* meets node *E*. Node *C* sends MP_c, MP_B and MP_F to node *E,* and node *E* sends its mobility pattern to node *C*. At t=4, nodes *B* and *A* meet each other. Node *A* receives MP_C, MP_B and MP_F, and node *B* receives MP_A. Also, at the same time nodes *E* and *F* meet each other and node F receives MP_E and node *E* does not receive any path vector as there is no new information. At t=5, nodes *B* and *D* meet each other and node *D* receives the path vector of all other nodes except node *E*'s path vector and node B adds MP_D to its routing table.

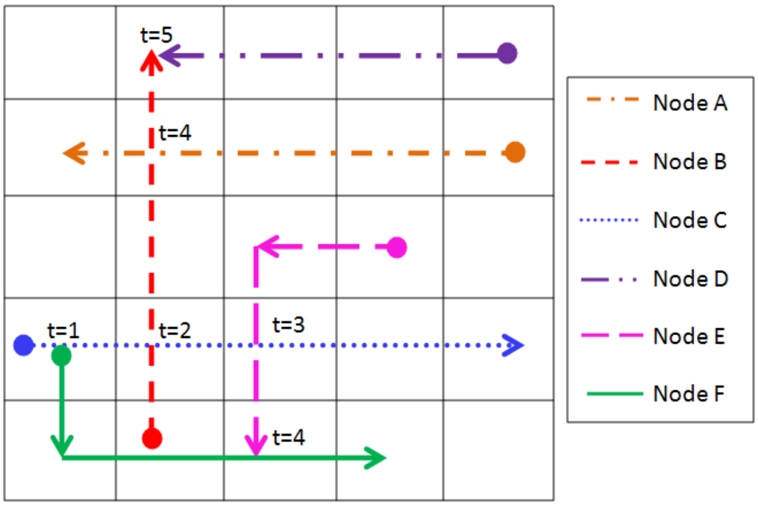


Figure 5-3 Five nodes with predefined paths are moving on a grid of size 5×5

Scenario 1: suppose that source node *A* generates a bundle at t=2 for destination *D*. When node *A* encounters node *B* at t=4, there is no route or path vector that shows any contacts that lead to node *D*. Node *A* has to wait for more contacts in hope of discovering a route. In this case, node A has to wait for five time units i.e., t=9, to discover a path. Then, the bundle is delivered within one time unit i.e., t=10. In addition to the increased delay, also notice that if the bundle lifetime expires in less than 8 time units, the bundle is not going to be delivered. In contrast, if node *A* sends the bundle to node *B* at t=4, then at t=5 the bundle is delivered.

Scenario 2: suppose that source node *A* generates a bundle at t=2 for destination *F*. At t=4 node *A* encounters node *B*, which has a route to destination *F*; a route that goes via node *B* and *C* exists. If the bundle is sent through this path, the bundle is delivered at t=11. However this route is not optimal as there is another route that goes via nodes *B, C* and *E*, which delivers the bundle at t=9. However, at t=4 node *A* has not received information about the optimal route. In this case, node *A* has to wait for more contacts to discover the optimal route to deliver the bundle faster. For example, if node *A* had waited, at t=9, the route discovered is optimal, which enables the bundle to be delivered at t=14. However, waiting for more contacts may increase delivery delay. On the other hand, waiting for a better route ensures we do not use the resources of other nodes unnecessarily. This is particularly critical if nodes have finite buffer size or energy constraint.

In summary, the problem at hand is to route bundles from every sender node i with incomplete routing information i.e., $|RT_i| < |N|$, such that 1) delivery delay is minimized, and 2) delivery ratio is maximized. Notice that the maximum performance is achieved when every node has a complete space-time graph, which they can then use to compute the optimal route to any destination.

5.4 Mobility Based Routing Protocol (MBRP)

MBRP considers the trajectory of nodes and the time of last contact between owners in order to minimize delay and maximize delivery ratio concurrently. In addition, MBRP is a quota protocol that limits the number of replicas for each generated bundle. This reduces the number of relay nodes required to deliver bundles. MBRP consists of the following two routing phases: space-time and heuristic. Briefly, in the former phase, each node constructs a space-time graph based on its recorded mobility pattern and contacts. Then, by applying a modified Dijkstra algorithm on the space-time graph, each node finds the fastest path. In the heuristic phase, nodes use recorded mobility patterns to predict subsequent contacts when their space-time graph is incomplete. Recall that a space-time graph is *incomplete* if a node's space-time graph does not contain the mobility pattern of all nodes. Also, if at least one recorded mobility pattern expires, the space-time graph becomes incomplete.

Nodes maintain the following data structure. A node's MP within a time period t is stored in a one dimensional array of size t. Every element i of the array indicates the geographical location of the node at time slot i. Each geographical location is assigned a unique integer number. Specifically, in a grid of size $w \times w$ where the grid coordinates x and y are between 1 and w, the unique integer number of each cell is calculated as follows.

$$s(x, y) = (y - 1) \times w + x \tag{5.1}$$

The space-time graph can be represented by a three-dimensional matrix M. Each element (i,j,k) of matrix M represents the time of the k-th contact between nodes i and j. For example, if nodes i and j meet each other two times at t=4 and t=10, matrix

M is updated to M($i,j,1$)=4 and M($i,j,1$)=10. Hence, the number of entries in matrix M is dependent of the number of contacts.

5.4.1 Space-time Phase

In this phase, each node uses the space-time graph constructed using learned mobility patterns from each contact to forward bundles via the fastest path. In order to find the fastest path from a source to a destination node, the source node assigns a cost l_i to every link *i* as follows

$$l_i = T_i - T_{i-1} \tag{5.2}$$

where T_i represents the time that the i-th link occurs in the path. For example, node S is connected to node A at t=1 and then node A is connected to node B at t=4. In this case, assuming the current time is zero, the delay of the link is one, and the delay of the link between A and B is three. As a result, any bundles on the route from node S to B will take 1+3=4 time units. Formally, the cost of a route ω is calculated as follows,

$$Cost_\omega = \sum_{i=1}^{|L_\omega|} l_i \tag{5.3}$$

where $|L_\omega|$ represents the number of links on path ω. In order to store the cost of links, a three-dimensional matrix, called cost matrix (CM), is established where each element (*i,j,k*) represents the cost of the *k*-th contact between nodes *i* and *j*. Each discovered path may have a different cost. In order to find the fastest path, nodes use a modified Dijkstra algorithm based on the proposed cost function. Algorithm 1 presents the pseudo-code used by nodes to find the fastest path towards a given destination. As mentioned, node i considers the recorded mobility patterns to find contacts (*line 3*). If a contact is detected, the time of contact is added to matrix M (*line 4*). Based on matrix M and the proposed cost function (See Equation 5.2), a node determines the CM matrix (*line 9*). Then, CM and a bundle's destination ID are fed into $Dijkstra()$ in order to find the fastest path $\mathcal{L}$ towards destination d (lines 12-13). Lastly, a single copy of bundle M_b is forwarded over route $\mathcal{L}$ (*line 14*).

Algorithm 1: the space-time phase

Input: RT_i

Output: *the fastest path* $\mathcal{L}$

Begin

1- **FOR** *every order pair X of recorded mobility patterns in* RT_i **DO**
2- **FOR** *every pair of nodes j and k where* MP_j *and* $MP_k \in RT_i$ **DO**
3- **IF** $MP_j(X) = MP_k(X)$
4- $M(j,k,q) \leftarrow X$;
5- **ENDIF**
6- **ENDFOR**
7- **ENDFOR**
8- **FOR** *every link* l_e *that connects nodes j and k* **DO**
9- $CM(j,k,q) \leftarrow T_e - T_{e-1}$;
10- **ENDFOR**
11- **FOR** every buffered bundle M_b at node n_i **DO**
12- $d \leftarrow M_b.destination$
13- $\mathcal{L} \leftarrow Dijkstra(CM, d)$
14- *send* $(M_b, \mathcal{L})$
15- **ENDFOR**

END

5.4.2 The Heuristic Phase

The aim of this phase is to route bundles when the space time graph is incomplete. The main idea is to take advantage of knowing the number of ordered pairs to estimate the reachability of nodes. Accordingly, the main observation is that when an encountered node has a large number of ordered pairs, it will be a good bundle carrier. Suppose that node *i* has recorded MP_j at time *t*. In this case, node *i* will mark an ordered pair of a mobility pattern MP_j as "*expired*" in RT_i if the second element of MP_j, namely time, is less than or equal to *t*. Node *i* also marks the remaining ordered pairs of MP_j as "*valid*", meaning their second element i.e., time, is greater than *t*. For example, in Figure 5-3, when node *A* meets node *B* at t=4, node *A* is not aware of any new contacts that nodes *C* and *F* have had after t=2 and t=1 respectively. In this example, node C meets node E at t=3 but at t=4 node *A* will not be aware of this contact given that the said contact occurs after the last contact with node C. Hence, when nodes *A* evaluates node *B* based on the number of valid ordered pairs, there are eight valid ordered pairs in node *B*'s routing table. Also, there is one valid ordered pair in node *A*'s routing table. Suppose that node *A* sends a number of replicas to node B. Based on Scenario 1 (see Section 5.3), the bundle is delivered at t=5. Based on the second scenario, when node *B* meets node *C* at t=6, there is one valid ordered pair in node *B*'s routing table; i.e., $(c_{2,1}, 5)$ in MP_A. In contrast, node C finds MP_E in its routing table with has two valid ordered pairs: $(c_{5,3}, 4)$ and $(c_{5,3}, 5)$.

In order to calculate the number of valid ordered pairs, every node *i* establishes a metric called "*Contact Time*" or CT_j for each encountered node *j*. This metric represents the last contact time between nodes *i* and *j*. For example, when nodes *i* and *j* meet each other, they set CT_j and CT_i to the contact time. In addition, they will also exchange MP_j and MP_i. Figure 5-4 shows an example. Nodes *i* and *j* meet each other at t=2 and exchange their mobility pattern and set $CT_i = 2$, $CT_j = 2$. At t=4, when node *j* and *k* meet each other, node *j* receives MP_k and sets $CT_k = 4$. Also node *k* receives MP_i, $CT_i = 2$ and MP_j, and sets $CT_j = 4$. Notice that nodes *i* and *k* have a different CT_j.

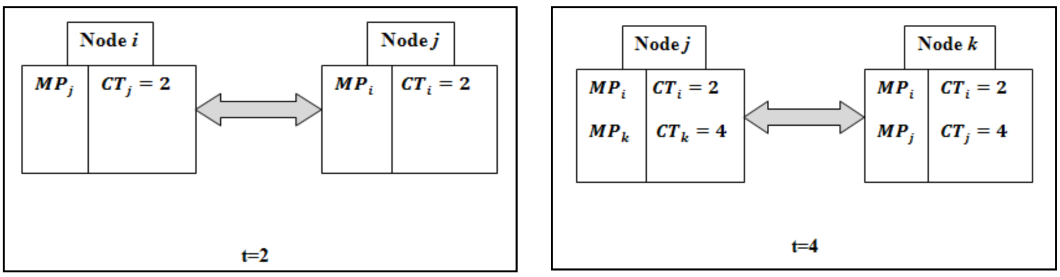


Figure 5-4 An example of mobility patterns exchange.

Upon contact, both connected nodes count the number of valid ordered pairs that belong to nodes with periodic and dynamic mobility patterns. Specifically, in terms of periodic mobility pattern, node *i* counts the number of valid ordered pairs as follows,

$$PMP_i = \sum_{MP_k \in RT_i} (|MP_k| - RT_i.CT_k) \quad (5.4)$$

where $|MP_k|$ indicates the total number order pairs of node k's mobility pattern and $RT_i.CT_k$ represents the last contact time that node i recorded for node k. In words, Eq. 4 counts the number of ordered pair of all periodic mobility patterns in node i's routing table since their last *Contact Time* up to the time that the cycle finishes. Recall that a cycle is a time period in which a node has a known mobility pattern.

As nodes with a dynamic mobility pattern, e.g., taxis, set a new trajectory in each cycle, these nodes will have more valid order pairs as compared to a node with periodic mobility pattern. Hence, the number of valid order pairs in a dynamic mobility pattern is dependent on the summation of all its cycles' length, called *CL*. Specifically, the number of valid ordered pairs for the dynamic case at node i is calculated as follows,

$$DMP_i = \sum_{MP_k \in RT_i} (CL - RT_i.CT_k) \quad (5.5)$$

In other words, Equation 5.5 counts the number of order pairs of all learned dynamic mobility patterns since their last *Contact Time* up to the time that the last cycle finishes. Here, CL is assumed to be equal to the time when the last recorded mobility pattern expires. Based on Equation 5.4 and Equation 5.5, the total number of valid order pairs, VOP_i, in the routing table of node i is computed as,

$$VOP_i = DMP_i + PMP_i \quad (5.6)$$

The next issue is forwarding of bundles. A sender node specifies the number of replicas to be forwarded to an encountered node based on the ratio of the number of valid order pairs in its routing table and the encountered node's routing table. For two nodes a and b, for the i^{th} bundle M_i that is headed to destination d, node a sends the following number of replicas to node b,

$$m_i \times \frac{VOP_b}{VOP_b + VOP_a} \tag{5.7}$$

where m_i is the available number of replicas for the i^{th} bundle at node a. In other words, using Equation 5.7, node a compares the number of valid ordered pairs in its routing table and node b's routing table. If VOP_a is smaller than VOP_b, node a does not need to keep a large number of replicas for itself. As a result, if node b has a larger number of valid ordered pairs, more replicas are forwarded to node b.

For example, assume node a has 10 replicas of a bundle M_1 and meets node b. Furthermore, assume node a with VOP_a = 10 and VOP_b = 90. Node a sends $\frac{90}{90+10} = \frac{90}{100}$ of available replicas of M_1 to node b. Therefore, node a forwards $10 \times \frac{90}{100} = 9$ replicas of M_1 to node b. Now assume VOP_a = 60 and VOP_b = 10, then $\frac{10}{10+90} = \frac{10}{100}$ of replicas of M_1 to node b. In this case, node a forwards $10 \times \frac{10}{100} = 1$ replica of M_1 to node b.

Algorithm 2 presents the steps performed by the heuristic phase. The algorithm is executed by every node i whenever it encounters another node j (line 3). Node i calculates the ratio of VOP_i and VOP_j in order to forward a portion of a bundle's replicas to node j (line 5-6).

Algorithm 1: the heuristic phase

Input: RT_i

1- **FOR** every encountered n_j **DO**

2- **FOR** every buffered bundle M_b at node n_i **DO**

3- $m_b \leftarrow M_b.numOfReplicas$

4- $d \leftarrow M_b.destination$

5- $m_{send} \leftarrow m_b \times \frac{VOP_j}{VOP_j + VOP_i}$

6- send m_{send} replicas of M_b to n_j

7- **ENDFOR**

8- **ENDFOR**

5.4.3 Discussion

Recall that MBRP uses GPS to encode the trajectory of a node whereby at each time slot, a node's location is recorded in the form of an ordered pair. Note that the resolution of mobility patterns is dependent on the length of time slots. This means if the time slot length is short, there will be more samples. In contrast, when the time slot length is long, contacts within a time slot may happen after each other without any overlap. To elaborate, assume that a time slot is 300 seconds in length. Now, suppose that node a and c meet each other in the first 10 seconds and then node c encounters node d for 40 seconds. Although these contacts happened after each other, they are recorded in one time slot. In this case, as the link between nodes a and c occurs before the link between nodes c and d at a given time slot, node d cannot send a bundle to node a in the time slot. To overcome this issue, samples are taken at small time units, e.g., one second, at the expense of a larger buffer size. For example, if a node's trajectory spans five hours and it encodes its trajectory at a resolution of one sample per second, then this amounts to 18000 samples. Now, if each sample is mapped to an integer number (See Eq. 1), then the node will require 72MB of memory to store its mobility pattern.

In order to reduce the size of mobility patterns, every node can encode its trajectory based on the residence time of being in each location. This means the storage required to store a node's mobility pattern is dependent on the number of cells in a grid. In addition, the grid resolution is dependent on nodes' wireless range. Recall that two nodes with a radio range of R are in contact if their distance is less than $2R$. Hence, if two nodes are located in a cell of size $2R$×$2R$, they are in contact. Accordingly, a grid covering an area of $x \times y$ will have $\frac{x}{2R} \times \frac{y}{2R}$ cells. If nodes were to encode their trajectory based on residence time, the length of a node's mobility pattern is dependent on the number of cells. Suppose the DTN is operating in an area of 10000×10000 m^2 and every node in the network has a transmission range of 100m. Then, there are 50×50 cells, each cell is 200×200 m^2 in size. Now, assume that a node has a speed of 10m/s and remains on the grid for 1000 seconds. In this case, the node passes each cell in 20 seconds, meaning that the node will pass

50 cells within 1000 seconds. Hence, only 50 samples are taken rather than 1000 samples.

5.5 Evaluation

MBRP was evaluated in the Java based simulator Opportunistic Network Environment (ONE) [95]. This simulator is able to generate vehicle movements using different mobility models [98-100] where nodes can have different cycle lengths. A deterministic network is created where nodes can have a periodic or dynamic mobility pattern in different cycles. Nodes have a predetermined mobility pattern and move in an area of approximately 5×3 km^2 in downtown Helsinki, Finland. All experiments adopt ONE's default settings, whereby 64% of nodes are pedestrians that move with a speed between 0.5 and 1.5 m/s. The other 32% of nodes are vehicles that move with a speed ranging from 2.7 and 13.9 m/s. Other nodes are trams that move with a speed between 7 and 10 m/s. Note that all nodes have a fixed transmission range of 20m and they also have a buffer size with a capacity of 20 bundles except trams that store 500 bundles. In all experiments, the bundle size is 100 KB. All nodes have a transmission speed of 250 kBps except trams, which has a transmission speed of 10 MBps.

In the first group of experiments, all nodes have a periodic mobility pattern. In this case, nodes repeat their mobility pattern every 12 hours. Each simulation lasts for three simulated cycles i.e., 36 hours, and each data point is an average of 10 runs. In the second group of experiments, nodes have dynamic mobility patterns where every node sets a new trajectory towards a new POI per cycle. Furthermore, nodes experience different cycle lengths. The third experiment models both periodic and dynamic mobility patterns. In this experiment, trams and buses have periodic mobility patterns and, cars/taxis and pedestrians have dynamic mobility patterns. In all these experiments, the number of sources/destinations is varied from 10 to 60 in increments of 10.

MBRP is compared against four well-known protocols. Namely, EBR [30], EPIDEMIC [43], MAXPROP [9], PROPHET [5] and Optimal [58]. Briefly, they operate as follows. EBR limits the number of replicas for each generated bundle. In EBR, every vehicle maintains its past average rate of encounters. Upon contact, a sender node sends a proportional number of replicas of a buffered bundle based on the ratio of its own and the receiver's encounter rate. As for EPIDEMIC, nodes simply broadcast a bundle to every encountered node. In addition, there is infinite number of replicas. MAXPROP assigns a weight to each link and derives a cost for each possible route. In fact, each node keeps track of the probability of meeting other nodes. Upon contact, nodes exchange these values. Then, the cost for possible paths toward destination nodes is calculated and bundles are forwarded via the minimum cost path. PROPHET uses a metric that indicates how likely a node will deliver a bundle to a given destination successfully. In each contact, a sending node passes its buffered bundle if an encountered node has a higher probability of delivering these bundles. Finally, when nodes use the Optimal algorithm, they are preloaded with the space-time graph. Hence, nodes know the network topology and the space-time graph is fixed throughout each experiment. Accordingly, if there is at least one route from a source node to a destination, the Optimal algorithm is guaranteed to find the fastest and shortest path.

The routing protocols are evaluated using three well-known performance metrics, namely 1) *delivery probability*, 2) *overhead*, and 3) *average delay*. Briefly, *delivery probability* is the ratio between the number of delivered bundles and the number of generated bundles. The metric *overhead* is the ratio of the number of delivered bundles and the number of bundles received by a node. Finally, *average delay* is the average time until a bundle is delivered. As mentioned in [30, 97], many protocols optimize one metric at the expense of another. For this reason, this work also uses three composite metrics namely, *DA, DOR*, and *DAO*; all of which are introduced by the authors of [30]. These composite metrics provide a ratio between delivery probability and conventional metrics. For example, *DA* provides a ratio between delivery probability (DP) and latency average (LA). Specifically,

$$DL=DP \times \frac{1}{LA} \tag{5.8}$$

Equation 5.9 defines DO that captures the trade-off between delivery probability and resulting overhead (OR), i.e.,

$$DO = DP \times \frac{1}{OR} \tag{5.9}$$

Lastly, Equation 5.10 defines *DLO* that scales the performance of a protocol based on delivery probability, average delay and overhead.

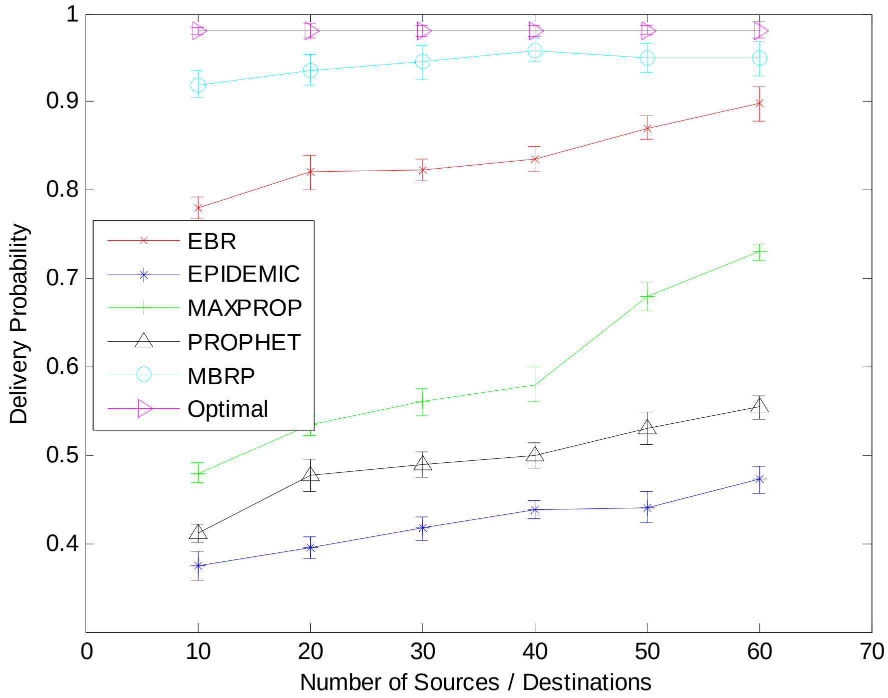
$$DLO = DP \times \frac{1}{LA} \times \frac{1}{OR} \tag{5.10}$$

5.5.1 Periodic Mobility Patterns

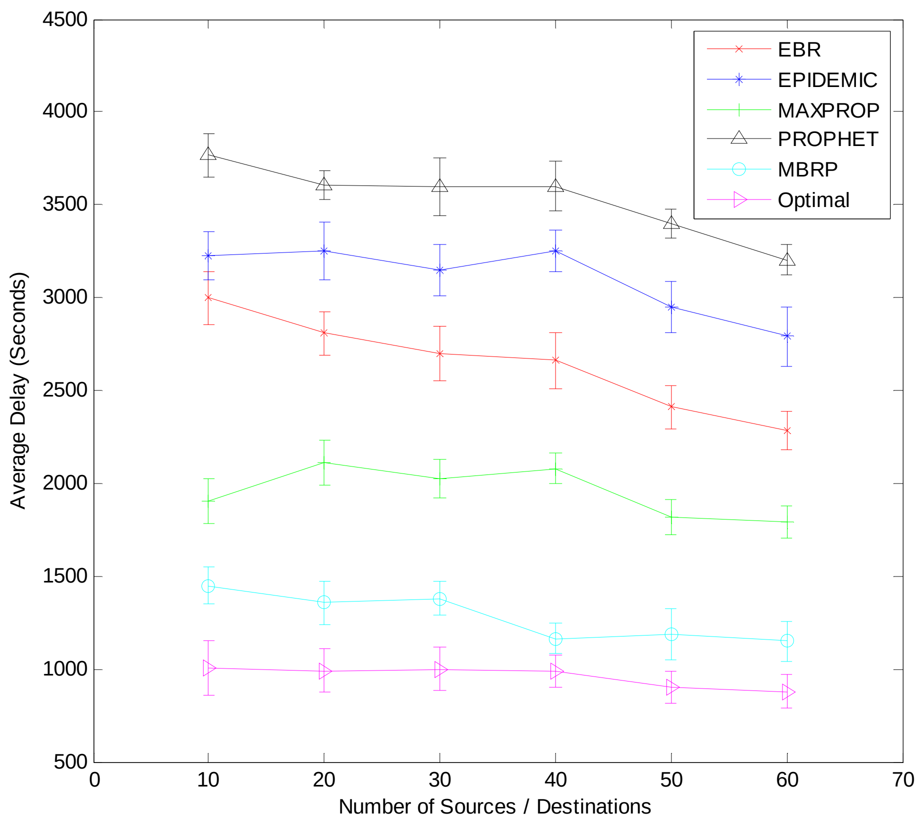
Figure 5-5 shows the performance of a DTN where every node has a fixed mobility pattern for each cycle and contacts occur periodically. In this scenario, as nodes do not change their trajectory, the space-time graph will reach a steady state once nodes record all mobility patterns. Figure 5-5(a) shows that MBRP delivers up to 16% more bundles as compared to EBR. This is because MBRP is guaranteed to deliver a bundle if a route is discovered. In addition, when there is no route towards a destination, MBRP estimates the future reachability of nodes to select a bundle's next hop.We see that EBR outperforms MAXPROP, PROPHET and EPIDEMIC. The reason is because EBR limits the number of replicas and hence, there are fewer number of dropped bundles as compared to flooding protocols. However, EBR may fail to deliver a bundle if the destination is located in a low density area. Figure 5-5(a) also shows that the Optimal protocol has up to 9% improvement as compared to MBRP. This is because nodes using MBRP may have an incomplete space-time graph.

In terms of delay, as shown in Figure 5-5(b), we see that MBRP delivers bundles up to 35% quicker than MAXPROP. Recall that MBRP sends bundles via the fastest discovered path. Consequently, bundles are delivered on a path with much smaller delays as compared to MAXPROP, PROPHET, and EBR. In terms of overhead, Figure 5-5(c) shows that MBRP and EBR use a small number of relays due to the finite number of replicas. Also, Figure 5-5(d) illustrates that MBRP performs 50% better than EBR in terms of *DO*. This is because MBRP uses the space-time phase where only a single copy is forwarded and bundle is guaranteed to be delivered. This results a high delivery ratio and low overhead. In Figure 5-5(e), the *DL* of MBRP is 42% less than the optimal protocol. As mentioned, this is due to nodes running

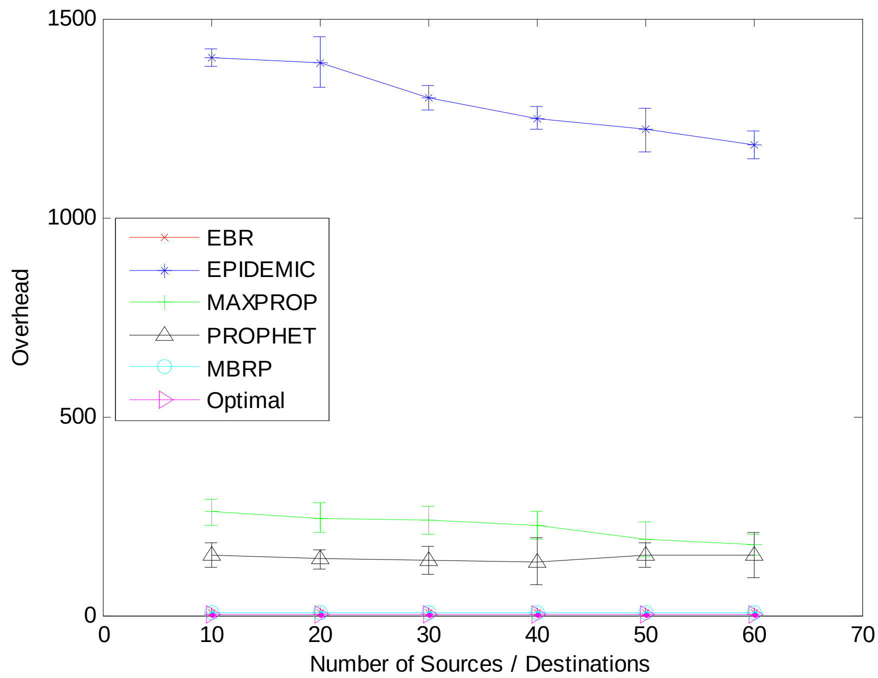
MBRP using an incomplete space-time graph. Lastly, Figure 5-5(f) shows that MBRP performs 150% better than EBR in terms of *DLO*.



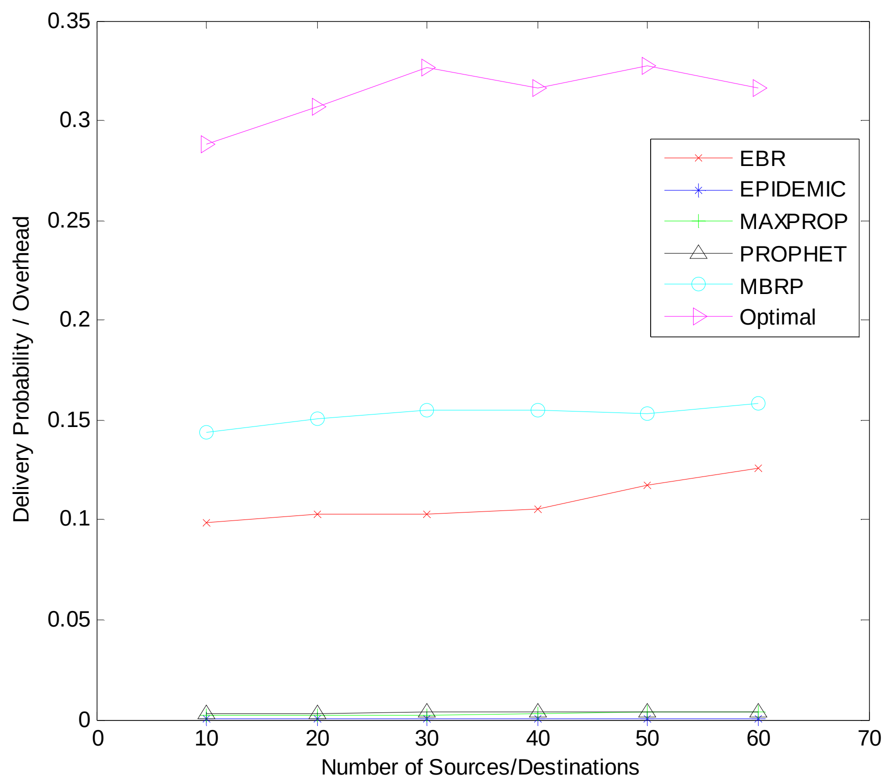
(a)



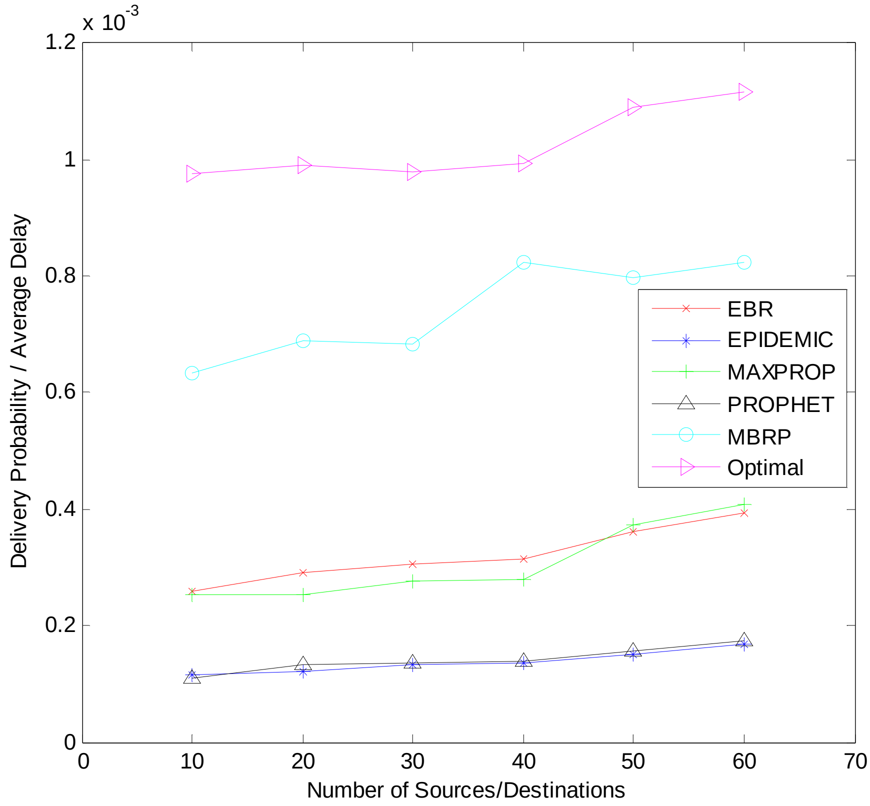
(b)


1500
1000
500
0
Overhead
0
10
20
30
40
50
60
70
Number of Sources / Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

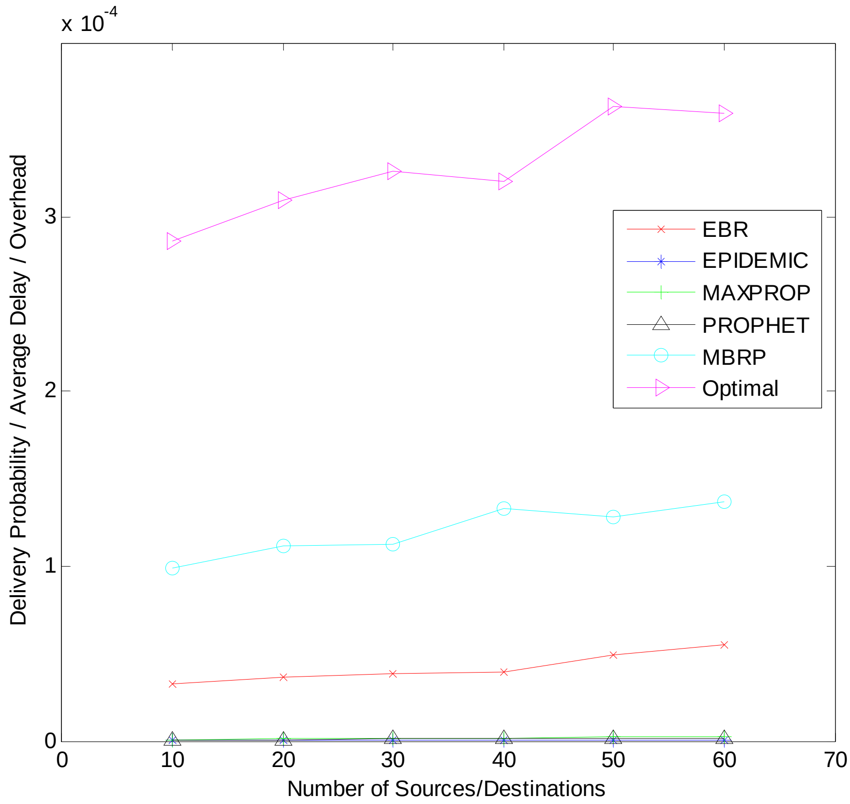
(c)


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Delivery Probability / Overhead
0
10
20
30
40
50
60
70
Number of Sources/Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

(d)



(e)

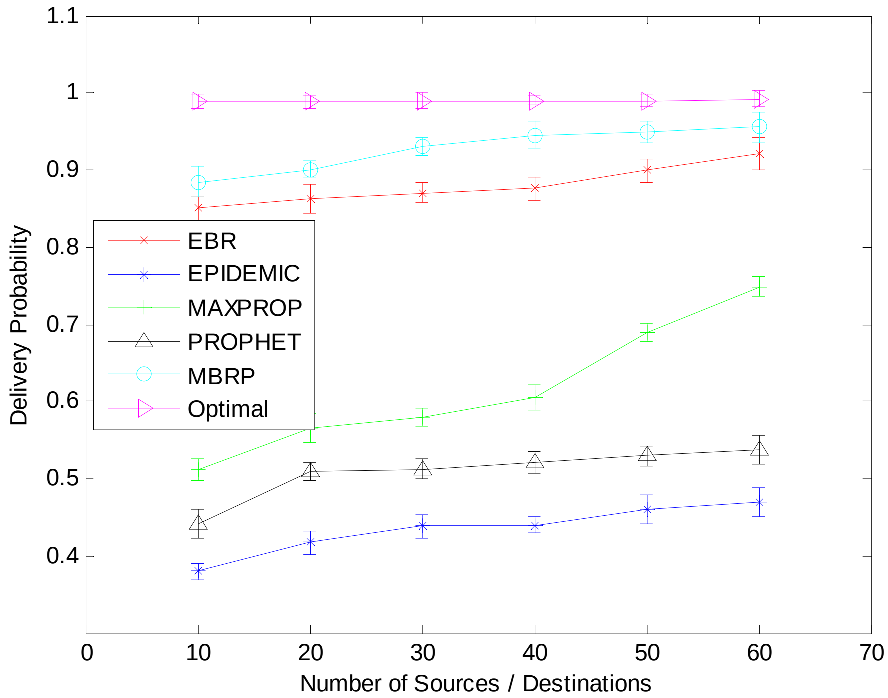


(f)

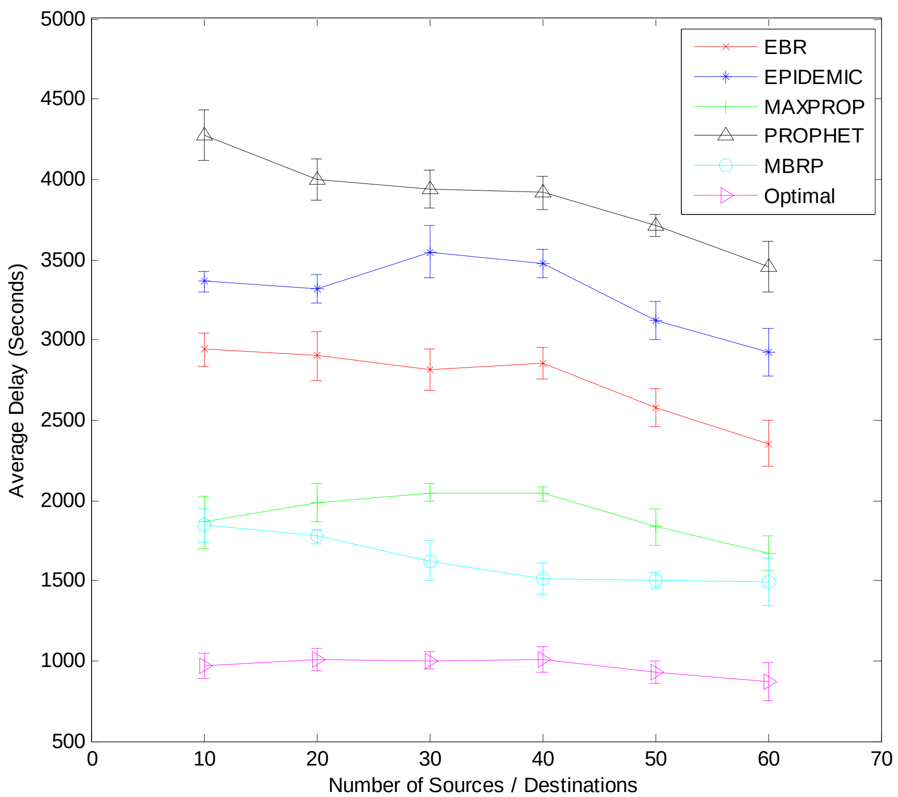
Figure 5-5 Network performance when the number of sources and destinations is varied between 10 and 60, a) delivery probability, b) average delay, c) overhead, d) DO, d) DL, and f) DLO

5.5.2 Dynamic Mobility Patterns

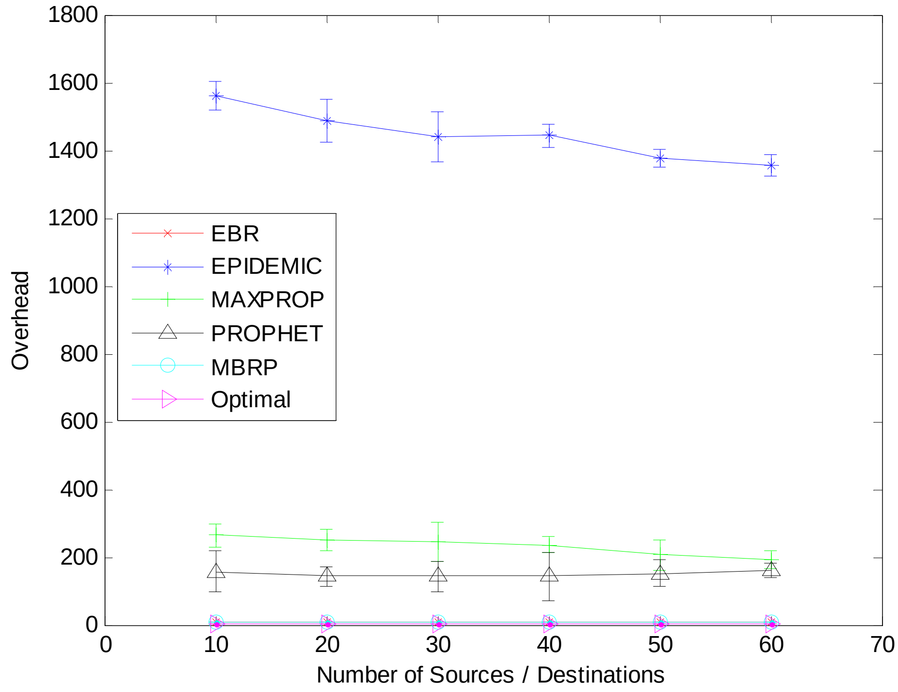
In this set of experiments, every node has a dynamic mobility pattern that changes once the node reaches a random POI. Hence, nodes have different mobility pattern lengths. Figure 5-6(a) shows MBRP is up to 6% better than EBR in terms of delivery ratio. Although nodes only have a valid mobility pattern for a given time period, the space-time phase may find a route towards a destination before their recorded mobility patterns expire. This causes MBRP to outperform EBR in terms of delivery ratio. As we can see from Figure 5-6(a), when the number of source/destination nodes increases, MBRP delivers up to 94% of bundles. This is because when the number of source/destination nodes increases, the probability that a sender node has a destination's mobility pattern increases. In other words, MBRP enters the space-time phase frequently. Figure 5-6(b) shows that MBRP reduces delays by up to 25% as compared to MAXPROP. As mentioned, the space-time phase reduces delays as bundles are forwarded via the fastest discovered path. As shown in Figure 5-6(b), when the number of sources and destinations increases, due to the use of the space-time phase, bundles' delivery delay decreases. In terms of overheads, Figure 5-6(c) shows that MBRP incurs 14% less resources usage as compared to EBR. This is because in the space-time phase of MBRP only a single copy of bundles is forwarded. Also, Figure 5-6(d) shows that MBRP has up to 25% improvement in *DO*. As a trade-off between delivery and delay, Figure 5-6(e) shows that MBRP has up to 100% improvement in *DL*. Finally, Figure 5-6(f) depicts that MBRP performs up to 100% better than EBR.


1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
Delivery Probability
0
10
20
30
40
50
60
70
Number of Sources / Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

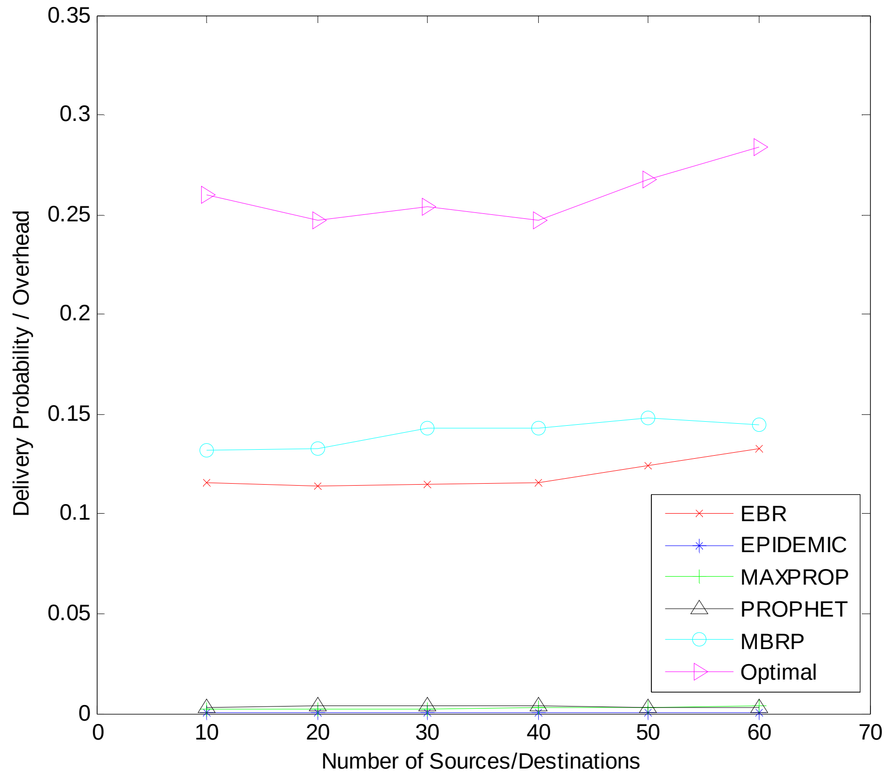
(a)


5000
4500
4000
3500
3000
2500
2000
1500
1000
500
Average Delay (Seconds)
0
10
20
30
40
50
60
70
Number of Sources / Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

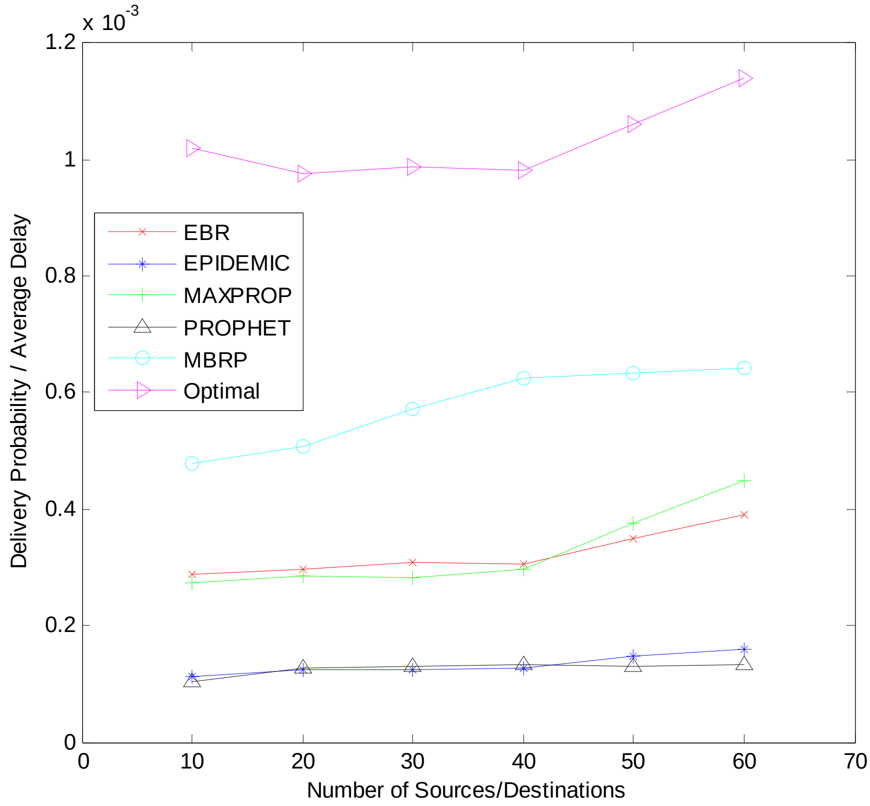
(b)


1800
1600
1400
1200
1000
800
600
400
200
0
Overhead
0
10
20
30
40
50
60
70
Number of Sources / Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

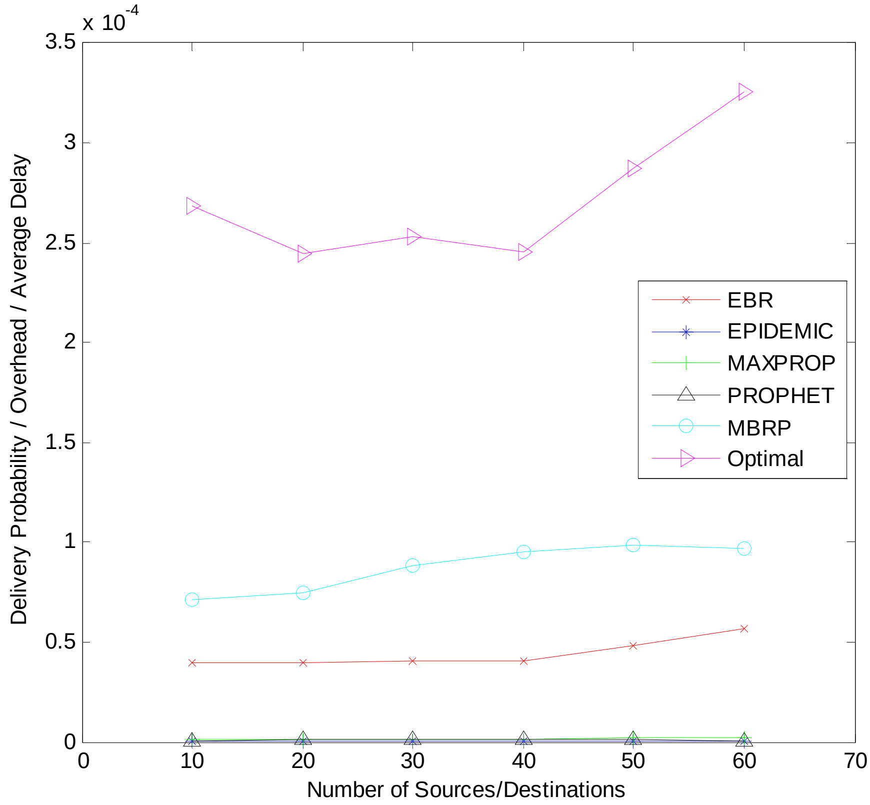
(c)


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Delivery Probability / Overhead
0
10
20
30
40
50
60
70
Number of Sources/Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

(d)



(e)

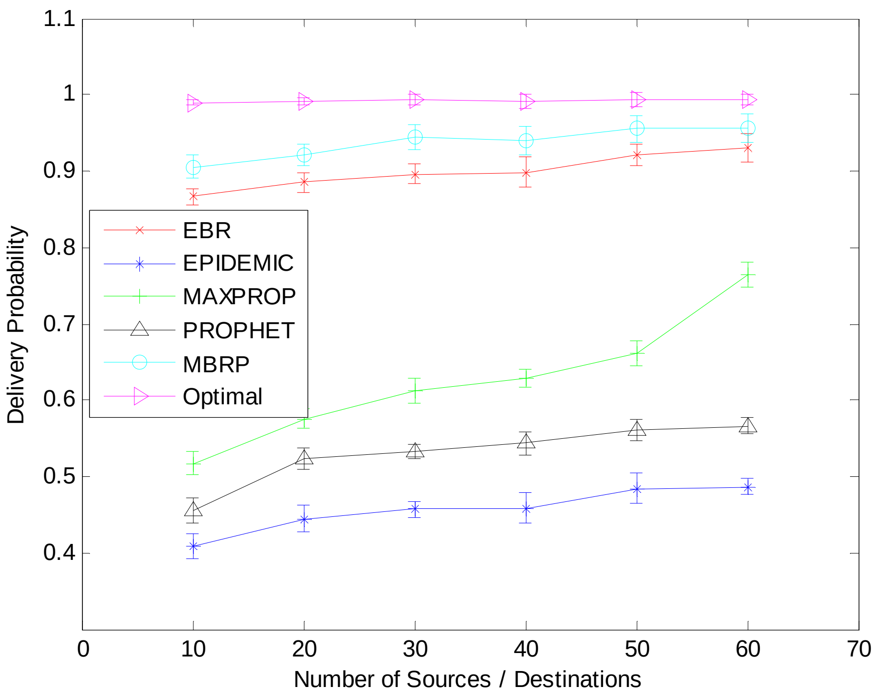


(f)

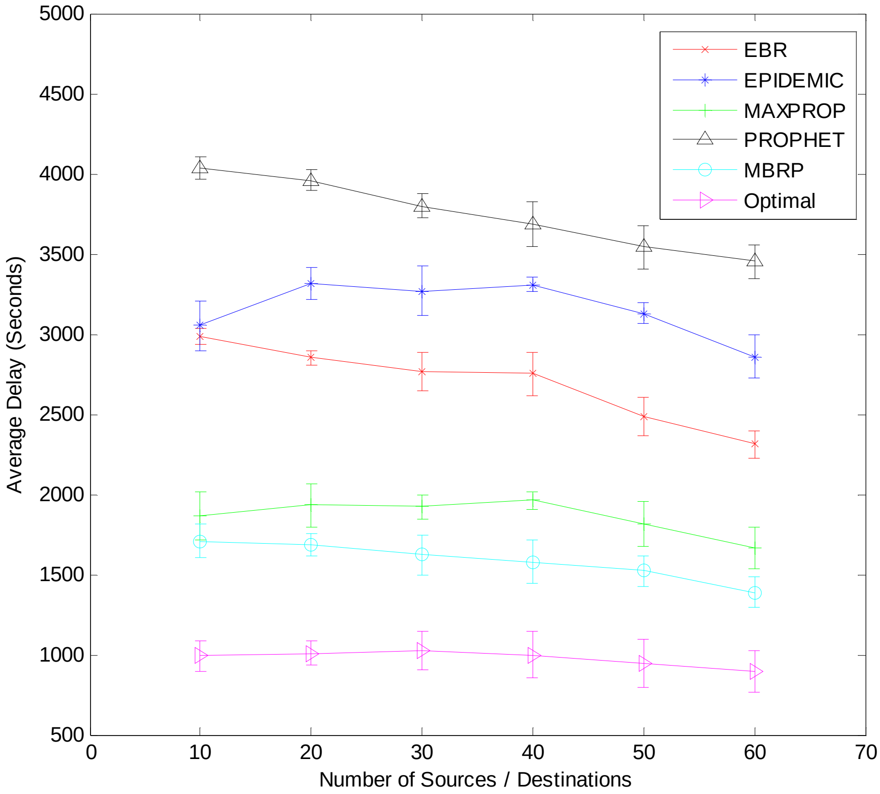
Figure 5-6 Network performance when the number of sources and destinations is varied between 10 and 60, a) delivery probability, b) average delay, c) overhead, d) DO, d) DL, and f) DLO

5.5.3 Mixed Mobility Patterns

Let's us now consider a scenario where 20% of nodes have a periodic mobility pattern and the remaining nodes have dynamic mobility patterns. In other words, 20% of nodes' routing table will remain fixed. Figure 5-7(a) shows that compared to EBR, MBRP achieves 7% improvement in delivery ratios. Also, MBRP's performance is 5% less than the optimal protocol. In terms of delay, Figure 5-7(b) shows that MBRP delivers bundles up to 15% quicker compared to MAXPROP. Figure 5-7(c) shows that MBRP consumes less resource as compared to PROPHET. This is because the number of replicas is limited in MBRP. Compared to EBR, Figure 5-7(c) also shows that MBRP has 21% reduction in overheads. This is due to its use of the space-time phase that forwards a single copy of bundles. Figure 5-7(e) shows the impact of mixing periodic and dynamic mobility patterns on both delivery ratio and delay. We can see that MBRP has 100% improvement as compared to EBR. Also, in terms of *DLO*, Figure 5-7(f) shows that MBRP performs up to 105% better than EBR.

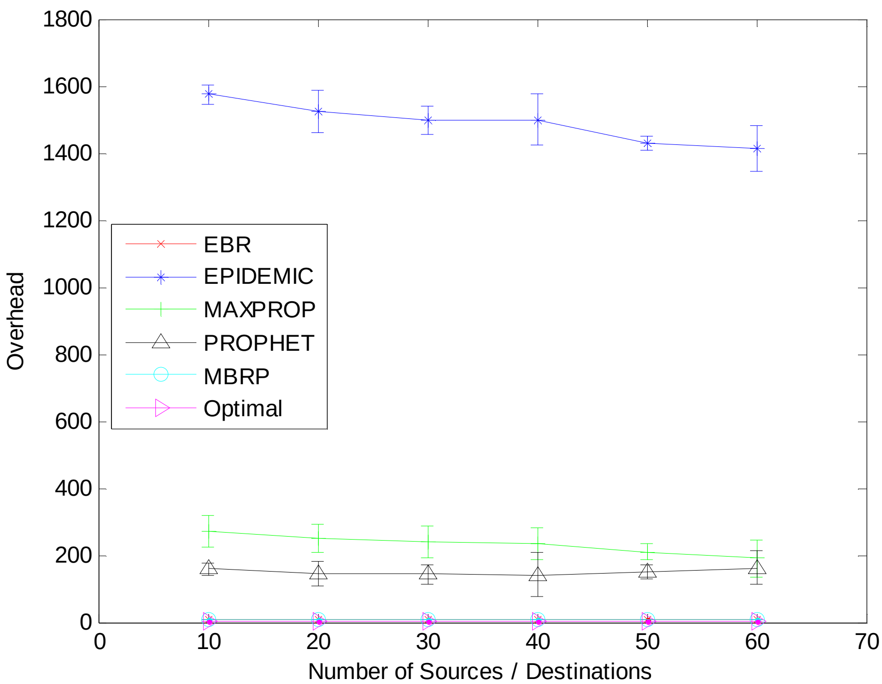


(a)



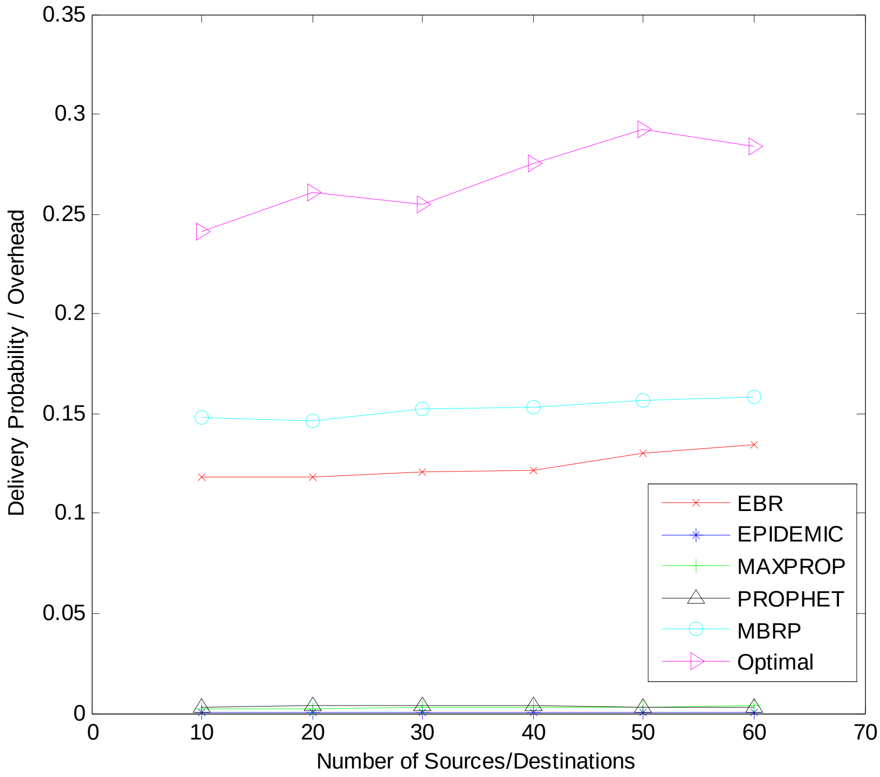
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal
Average Delay (Seconds)
Number of Sources / Destinations
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
10
20
30
40
50
60
70

(b)

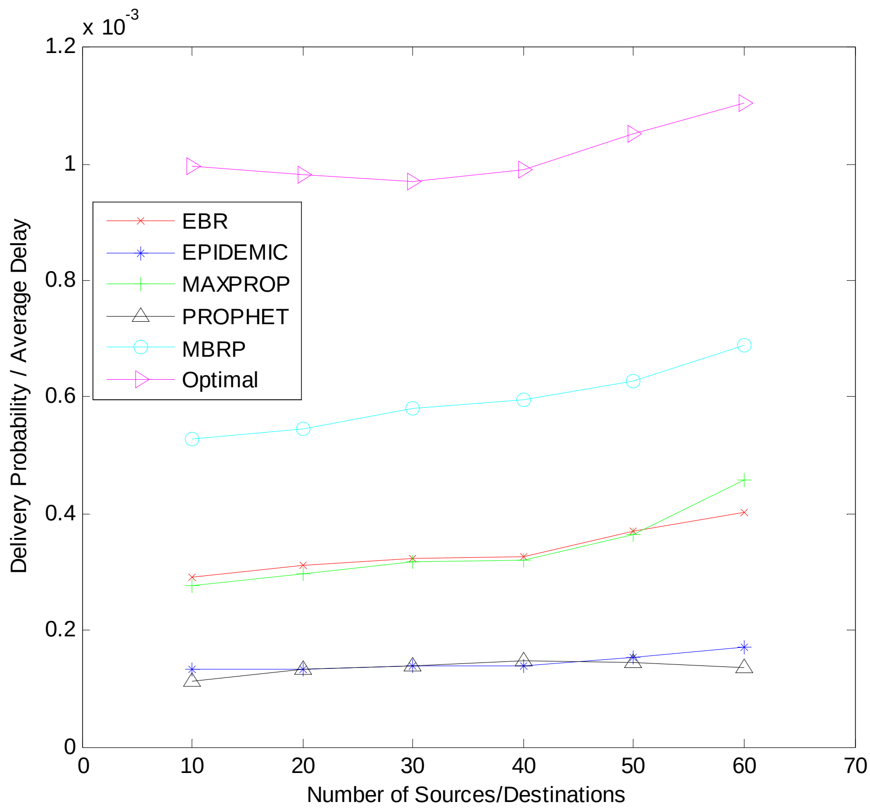


EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal
Overhead
Number of Sources / Destinations
1800
1600
1400
1200
1000
800
600
400
200
0
0
10
20
30
40
50
60
70

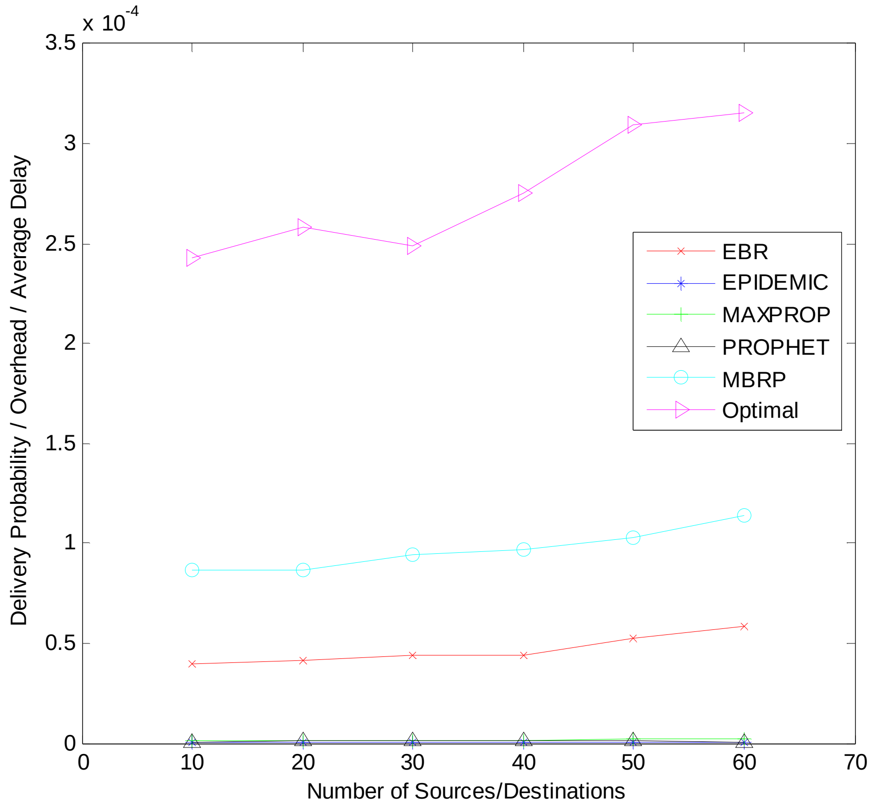
(c)


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Delivery Probability / Overhead
0
10
20
30
40
50
60
70
Number of Sources/Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

(d)


x 10^-3
1.2
1
0.8
0.6
0.4
0.2
0
Delivery Probability / Average Delay
0
10
20
30
40
50
60
70
Number of Sources/Destinations
EBR
EPIDEMIC
MAXPROP
PROPHET
MBRP
Optimal

(e)



(f)

Figure 5-7 Network performance when the number of sources and destinations is varied between 10 and 60, a) delivery probability, b) average delay, c) overhead, d) DO, d) DL, and f) DLO

5.6 Conclusion

This chapter has investigated a novel forwarding strategy for deterministic DTNs in order to increase delivery ratios and reduce delivery delays. In particular, this chapter proposes MBRP, a protocol that takes advantage of a space-time graph to send a single copy of each bundle over the fastest discovered path. In addition, as nodes initially have zero information about the network topology and mobility patterns may be valid only for a time period, the space-time graph may become incomplete or staled. In this case, as a route may not be discovered or the discovered route may not be optimal in terms of delay, MBRP evaluates the reachability of encountered nodes based on their routing table in order to send a proportional number of replicas to them. The simulation results, over a DTN comprising of nodes with dynamic and periodic mobility patterns show that compared to EBR, MBRP achieved up to 105% improvement in a service quality metric called DLO which comprises of delivery, delay and overhead.

Chapter 6

Conclusion

This thesis has studied data management in DTNs where there are no permanent paths between nodes. Consequently, the resulting topology cannot be supported by traditional ad-hoc routing protocols and requires nodes to use store and carry paradigm in order to forward bundles. Due to the properties of DTNs, nodes are faced with the following challenges: (i) sender/source nodes need to decide the next hop node for each bundle. In this respect, flooding protocols are simple and do not limit the number of forwarded replicas for a given bundle. Although flooding protocols provide robust network performance such as good delivery ratios and delays, they have high overheads. In contrast, quota protocols limit the number of replicas at the expense of low delivery ratios, (ii) nodes need an effective buffer management policy during congestion. In addition, as nodes may move at a high speed and/or have short radio range, contacts between nodes may be insufficient to exchange all bundles. In this case, buffered bundles need to be prioritized based on different criteria.

Based on aforementioned challenges, this thesis has investigated the following research questions:

1. How to efficiently forward a finite number of replicas based on the contact history of nodes?

2. How to effectively prioritize bundles in order to yield high delivery ratios and low delays with respect to finite number of replicas?

3. How to exploit mobility patterns of nodes when they are deterministic for a given time period with the goal of maximizing network performance such as delivery ratios and delays?

To address the first question, a detailed literature survey showed that existing dynamic routing protocols are suitable for unpredictable DTNs where nodes movement is random, and unpredictable. Compared to dynamic routing protocols, history based protocols improve network performance in terms of delivery ratios and delays when network resources such as buffer, bandwidth, and energy, are limited. However, these protocols assume a bundle can be replicated infinitely, which incurs high overheads. Also, history based quota protocols do not work efficiently when the network has a low node density. For example, in EBR [30], as replicas are disseminated to area(s) with a high encounter rate, meaning regions with high node density, it will fail to deliver a bundle if the destination is in a low node density area. These findings were exploited by the routing algorithm described in Chapter 3. Specifically, Chapter 3 proposed a Destination Based Routing Protocol (DBRP) that selects a next hop node based on the ratio of its encounters with a bundle's destination compared to other nodes. Hence, upon contact, a sender node forwards a proportional number of finite replicas based on the encounter rate of the sender as well as the encountered node. This observation is a marked departure of other routing protocols that forward to nodes with high encounter rates even though they may never had any contacts with the destination. Chapter 3 also verified this hypothesis using a Markov model to predict contacts between nodes. The model shows that when a contact between an intermediate node and a destination is predictable within a given period of time, the probability of delivery through that intermediate node is maximum.

Chapter 4 addresses the second question. A key finding from the extensive literature survey is that current buffer management policies are designed for flooding protocols. This is because nodes under flooding protocols are allowed to replicate a bundle without any limit. Inevitably, this causes congestion. However, when the number of replicas is finite, and a replica is dropped, the delivery probability of the corresponding bundle reduces. Amongst current policies, many schemes use global information in order to improve network performance. However, due to large delays, collected information may become obsolete. In addition, collecting global information imposes a high signalling overhead. To overcome this issue, a number of protocols approximate global information via a distribution function. However, the resulting estimates are not accurate under different forwarding strategies. Based on these findings, Chapter 4 proposed a queue management policy called QM-EBRP that works under quota protocols. QM-EBRP utilises three bundle properties available locally at each node; namely, a node's encounter rate, a bundle's lifetime and the number of replicas associated with a bundle. These properties enable QM-EBRP to derive the probability that a bundle has been delivered and its likelihood to be delivered in the future. In turn, these probabilities enable QM-EBRP to prioritize the dropping and forwarding of bundles during congestion and at each contact.

To address the last question, the literature showed that routing protocols for deterministic DTNs assume that the space-time graph is loaded at all nodes in advance. However, this is not practical when nodes have a dynamic mobility pattern. In addition, the space-time graph is fixed all the times. Chapter 5 investigated a novel forwarding strategy called MBRP that does not make the said assumptions. MBRP assumes that nodes do not have full knowledge of the network topology. In addition, the space-time graph is dynamic, meaning the trajectory of nodes may only be valid for a given period of time. Base on these assumptions, MBRP takes advantage of space-time graph to send a single copy of each bundle over the fastest discovered path. MBRP also considers the case where the space-time graph is incomplete or staled due to nodes initially having zero information about the network topology. Moreover, mobility patterns may be valid only for a finite time period. In this case, MBRP evaluates the reachability of encountered nodes based on their routing table in order to send a proportional number of replicas to them.

To conclude, unlike existing works, this thesis has identified new ways to exploit the encounter rate of nodes. Consequently, protocols such as DBRP and QM-EBRP are able to exploit the encounter rate of nodes to estimate their utility in delivering bundles. For example, DBRP rates a person *A* that goes to work and meets person *C* every day highly if there are bundles destined to person C. Hence, the mobility pattern of nodes is predictable, meaning that nodes *A* and *C* will meet each other in the future. As a result, person *A* is an ideal bundle carrier for person *C* because delivery is guaranteed. Also, QM-EBRP uses nodes' encounter rate to determine a bundle's utility, which helps estimate how likely a bundle will be delivered in the future and also its expected delay. As mentioned in [91, 92], people usually roam in relatively small regions. Hence if a node has a high encounter rate in its history, it will have a high encounter rate in the future. QM-EBRP takes advantage of this observation to estimate a bundle's future delivery probability and to reduce expected delays. However, in the random mobility model, QM-EBRP will not work efficiently as the history of encounters does not necessarily represent an estimate of future contacts. Hence, QM-EBRP is only suited for semi-predictable/social-based networks. This thesis also investigated DTNs where nodes are semi-predictable. In this respect, Chapter 5 showed that if the mobility pattern of nodes is longer, MBRP has a higher chance to discover a route towards a destination. However, large delays in DTNs cause nodes to record a large number of expired ordered pairs and in the worst case, these mobility patterns may expire before being received by nodes.

An immediate future work is to investigate social based mobility patterns whereby the movement of nodes is dependent on each other in terms of location and time. For example, a person who is waiting for a bus may have an independent mobility pattern before catching the bus. However, when this person is on the bus its mobility pattern is dependent on the bus's mobility pattern. In this case, when the mobility pattern of a node expires, the new mobility pattern can be predicted in advance. Another future work is to use different inference engines in evaluating nodes. For example, a fuzzy inference engine maps routing parameters to linguistic parameters. Then, linguistic parameters are fed into rules to make decision based on human knowledge. This decision can determine the number of replicas to be forwarded.

References

[1] Delay tolerant networking research group, http://www.dtnrg.org.

[2] S. Burleigh, A. Hooke, and L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary Internet", *IEEE Communications Magazine,*vol. 41, no. 6, pp. 128-136, 2003.

[3] A. Chaintreau, P. Hui, and J. Crowcroft, "Impact of Human Mobility on Opportunistic Forwarding Algorithms", *IEEE Transactions on Mobile Computing,* vol. 6, no. 6, pp. 606-620, 2007.

[4] K. Scott and S. Burleigh, "Bundle Protocol Specification", IETF RFC5050, November 2007.

[5] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks", *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 7, no. 3, pp. 19-20, 2003.

[6] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: rethinking connectivity in developing nations", *IEEE Computer,* vol. 37, no. 1, pp. 78-83, 2004.

[7] Z. J. Haas and T. Small, "A new networking model for biological applications of ad hoc sensor networks", *IEEE/ACM Transaction on Networking (TON),* vol. 14, no. 1, pp. 27-40, 2006.

[8] W. Hsu, A. Helmy , "On modeling user associations in wireless lan traces on university campuses", *IEEE 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, April 2006.

[9] J. Burgess, B. Gallaghe, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks", *IEEE INFOCOM*, Barcelona, Spain, April 2006.

[10] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks", *Elsevier Ad Hoc Networks*,vol.1, no. 2-3, 2003.

[11] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet", *SIGARCH Computer Architecture News,* vol. 30, no. 5 2002.

[12] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing", *IEEE Proceedings of the Second Workshop on Mobile Computer Systems and Applications (WMCSA)*, p.90, New Orleans, Louisiana, U**SA** , February 1999.

[13] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", *Springer Mobile Computing,* vol. 353, pp. 153-181, 1996.

[14] Y.-Bae Ko and N. H. Vaidya, "Location Aided Routing (LAR) in mobile ad hoc networks", *Springer Wireless Networks,* vol. 6, no. 4, 2000.

[15] M. Grossglauser and M. Vetterli, "Locating nodes with EASE: last encounter routing in ad hoc networks through mobility diffusion", *IEEE INFOCOM*, March-April 2003.

[16] H. D. Ferriere, M. Grossglauser, and M. vetterli "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages", *ACM Proceedings of 4th international symposium on Mobile ad hoc networking & computing (MobiHOC),* Annapolis, Maryland, USA, June 2003.

[17] S.-Ju Lee, W. Su, and M. Gerla, "Wireless ad hoc multicast routing with mobility prediction", *Springer Mobile Networks Applications,* vol. 6, no. 4, pp. 351-360, 2001.

[18] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges", *IEEE Communications Surveys & Tutorials,* vol. 8, no.1, pp. 24-37, 2006.

[19] M. H. Elizabeth and M. Daly, "The challenges of disconnected delay-tolerant MANETs", *Elsevier Ad Hoc Networks,* vol. 8, no. 2, pp. 241-250, 2010.

[20] M. G. Rubinstein, F. B. Abdesslem, M. D. de Amorim, S. R. Cavalcanti, R. d. S. Alves, L. H. M. K. Costa, O. C. M. B. Duarte, M. E. M. Campista, and U. F. Fluminense, "Measuring the capacity of in-car to in-car vehicular networks," *IEEE Communications Magazine,* vol. 47, no. 11, pp. 128-136, 2009.

[21] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-Based Routing in DTNs", *IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[22] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network", *ACM Proceedings of the 5th annual international conference on Mobile computing and networking (MOBICOM)*, Seattle, Washington, United States, August 1999.

[23] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical Routing in Delay-Tolerant Networks", *IEEE Transactions on Mobile Computing,* vol. 6, no. 8, pp. 943-959, 2007.

[24] V. Erramilli and M. Crovella, "Forwarding in opportunistic networks with resource constraints", *ACM Proceedings of the third workshop on Challenged networks*, pp. 41-48, San Francisco, California, USA, September 2008.

[25] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks", *ACM*

SIGCOMM workshop on Delay-tolerant networking, Philadelphia, Pennsylvania, USA, August 2005.

[26] S. Kapadia, B. Krishnamachari, and L. Zhang, "Data Delivery in Delay Tolerant Networks: A Survey", *Mobile Ad-hoc Networks: Protocol Design*, 2011.

[27] J. Miao, O. Hasan, S. B. Mokhtar, and L. Brunie, "A self-regulating protocol for efficient routing in mobile delay tolerant networks", *IEEE 6th International Conference on Digital Ecosystems Technologies (DEST),* Campione d'Italia, June 2012.

[28] M. A. T. Prodhan, R. Das, M. H. Kabir, and G. C. Shoja, "Probabilistic quota based adaptive routing in Opportunistic Networks", *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim)*, Victoria, BC, August 2011.

[29] S. C. Lo and W. R. Liou, "Dynamic Quota-Based Routing in Delay-Tolerant Networks", *IEEE Vehicular Technology Conference (VTC)*, Yokohama, May 2012.

[30] S. C. Nelson, M. Bakht, R. Kravets, and A. Harris, "Encounter: based routing in DTNs", *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 13, no. 1, pp. 56-59, 2009.

[31] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network", *ACM SIGCOMM Computer Communication Review,* vol. 34, no.4, pp. 145-158, 2004.

[32] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary internet: state-of-the-art and research challenges", *Elsevier Computer Networks,* vol. 43, no. 2, pp. 75-112, 2003.

[33] B. K. Polat, P. Sachdeva, M. H. Ammar, and E. W. Zegura, "Message ferries as generalized dominating sets in intermittently connected mobile networks", *ACM Proceedings of the Second International Workshop on Mobile Opportunistic Networking (MobiOpp)*, pp. 22-31, Pisa, Italy, February 2010.

[34] Q. Li and D. Rus, "Sending messages to mobile users in disconnected ad-hoc wireless networks", *ACM MOBICOM*, Boston, Massachusetts, USA, August 2000.

[35] X. Zhang, J. Kurose, B. N. Levin, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing", *ACM MOBICOM*, Montreal, Quebec, Canada, September 2007.

[36] J. Leguay, T. Friedman, and V. Conon, "DTN routing in a mobility pattern space", *ACM SIGCOMM Proceedings of the workshop on Delay-tolerant networking*, Philadelphia, Pennsylvania, USA, August 2005.

[37] C. Liu and J. Wu, "An optimal probabilistic forwarding protocolin delay tolerant networks", *ACM Proceedings of the tenth international symposium on Mobile ad hoc networking and computing (MobiHoc),* pp. 105-114, New Orleans, LA, USA, May 2009.

[38] E. Altman, G. Neglia, F. D. Pellegrini, and D. Miorandi, "Decentralized Stochastic Control of Delay Tolerant Networks", *IEEE INFOCOM*, 2009.

[39] R. Groenevelt, P. Nain, and G. Koole, "Message delay in MANET", *ACM SIGMETRICS Performance Evaluation Review,* vol. 33, no. 1, pp. 412-413, 2005.

[40] T. Spyropoulos*, K. Psounis*, and C. S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case", *IEEE/ACM Transactions on Networking,* vol. 16, no. 1, pp. 63-76, 2008.

[41] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case", *IEEE/ACM Transactions on Networking,* vol. 16, no. 1, pp. 77-90, 2008.

[42] D. Henriksson, T. F. Abdelzaher, and R. K. Ganti, "A Caching-Based Approach to Routing in Delay-Tolerant Networks", *IEEE Proceedings of 16th International Conference on Computer Communications and Networks(ICCCN)*, Honolulu, HI, USA, August 2007.

[43] A. Vahdat, D. Becker, "Epidemic routing for partially connected ad hoc networks", Technical Report, CS-200006, Department of Computer Science, Duke University, Durham, NC, 2000.

[44] T. Matsuda and T. Takine, "(p,q)-Epidemic routing for sparsely populated mobile ad hoc networks", *IEEE Journal on Selected Areas in Communications,* vol. 26, no. 5, pp. 783-793, 2008.

[45] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks"*, IEEE/ACM Transactions on Networking,* vol. 10, no. 4, pp. 477-486, 2002.

[46] E. Bulut, Z. Wang, and B. K. Szymanski, "Cost-Effective Multiperiod Spraying for Routing in Delay-Tolerant Networks", *IEEE/ACM Transactions on Networking,* vol. 18, no. 5, pp. 1530-1543, 2010.

[47] G. Sandulescu and S. Nadjm-Tehrani, "Opportunistic DTN routing with window-aware adaptive replication", *ACM Proceedings of the 4th Asian Conference on Internet Engineering*, PP. 103-112, Pratunam, Bangkok, Thailand, November 2008.

[48] M. Musolesi and C. Mascolo, "CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks", *IEEE Transactions on Mobile Computing,* vol. 8, no. 2, pp. 246-260, 2009.

[49] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks", *IEEE Proceedings of Fifth International Symposium on Wearable Computers*, Zurich ,Germany, October 2001.

[50] E. C. R. D. Oliveira and C. V. N. D. Albuquerque. "NECTAR: a DTN routing protocol based on neighborhood contact history", *ACM Proceedings of the symposium on Applied Computing*, PP. 40-46, Hawaii, USA, March 2009.

[51] M. Kalantari and R. J. La, "A DTN packet forwarding scheme inspired by thermodynamics", *IEEE 42nd Annual Conference on* in *Information Sciences and Systems*, Princeton, NJ , USA, March 2008.

[52] L. Ze and S. Haiying, "Utility-Based Distributed Routing in Intermittently ConnectedNetworks", *IEEE 37th International Conference on Parallel Processing*, Portland, OR , USA, September 2008.

[53] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility", *IEEE 5th Annual International Conference on Pervasive Computing and Communications Workshops*, White Plains, NY,USA, March 2007.

[54] K. Tan*, Q. Zhang, and W. Zhu*, "Shortest path routing in partially connected ad hoc networks", *IEEE Global Telecommunications Conference* (*GLOBECOM)*, 2003.

[55] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks", *IEEE INFOCOM*, March 2005.

[56] I. Leontiadis and C. Mascolo, "GeOpps: Geographical Opportunistic Routing for Vehicular Networks", *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM)*, Espoo, Finland, June 2007.

[57] V. N. J. G. Soares, J. J. P. C. Rodrigues, and F. Farahmand, "GeoSpray: A geographic routing protocol for vehicular delay-tolerant networks", *Elsevier Information Fusion,* vol. 15, pp. 102-113, 2014.

[58] M. Huang, S. Chen, Y. Zhu, B. Xu, and Y. Wang, "Topology Control for Time-Evolving and Predictable Delay-Tolerant Networks", *IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Valencia, Spain, October 2011.

[59] M. Huang, S. Chen, Y. Zhu, and Y. Wang, "Cost-Efficient Topology Design Problem in Time-Evolving Delay-Tolerant Networks", *IEEE Global Telecommunications Conference (GLOBECOM)*, Miami, FL, USA, December 2010.

[60] B. B. Xuan, A. Ferreira, and A. Jarry, "Computing Sortest, Fastest, and Foremost Journey in Dynamic Networks," *International Journal of Foundations of Computer Science,* vol. 14, no. 2, pp. 267-285, 2003.

[61] A. Ferreira, "Building a reference combinatorial model for MANETs", *IEEE Network,* vol. 18, no. 5, pp. 24-29, 2004.

[62] R. Handorean, C. Gill, G.-C. Roman, "Accommodating Transient Connectivity in Ad Hoc and Mobile Settings", *Elsevier Pervasive Computing,* vol. 3001, pp. 305-322, 2004.

[63] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Transactions of the ASME – Journal of Basic Engineering,* Series D, no. 82, pp. 35-45, 1960.

[64] C. Schwingenschlogl and T. Kosch, "Geocast enhancements of AODV for vehicular networks", *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 6, no. 3, pp. 96-97, 2002.

[65] C. Lochert, H. Hartenstein, J. Tian, H. Fubler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments", *IEEE Proceeding of Intelligent Vehicles Symposium (IV)* Columbus, Ohio, USA, June 2003.

[66] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges", *IEEE, Communications Surveys & Tutorials,* vol. 14, no. 2, pp. 607-640, 2012.

[67] S. Merugu, M. Ammar, and E. Zegura, "Routing in Space and Time in Networks with Predictable Mobility", Georgia Institute of Technology, Technical Report, GIT-CC-04-07, 2004.

[68] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "*Introduction to Algorithms*", The MIT Press, 1990.

[69] R. W. Floyd, "Algorithm 97: Shortest path", *Communication of the ACM,* vol. 5, no.6, pp. 345, 1962.

[70] D. Hay and P. Giaccone, "Optimal routing and scheduling for deterministic delay tolerant networks", *IEEE Sixth International Conference on Wireless On-Demand Network Systems and Services*, Snowbird, UT, February 2009.

[71] C. Liu, and J. Wu, "Practical Routing in a Cyclic MobiSpace", *IEEE/ACM Transactions on Networking,* vol. 19, no. 2, pp. 369-382, 2011.

[72] Q. Ayub, S. Rashid, "T-Drop: An optimal buffer management policy to improve QOS in DTN routing protocols", *Journal of Computing,* vol. 2, no. 10, 2010.

[73] X. Zhang, G. Neglia, J. Kurose, D. Towsley, "Performance modeling of epidemic routing", *Computer Networks: The International Journal of Computer and Telecommunications Networking,* vol. 51, no. 10, pp. 2867-2891, 2007.

[74] A. Lindgren, and K. S. Phanse, "Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks", *IEEE First International Conference on Communication System Software and Middleware*, New Delhi, India, 2006.

[75] D. Pan, Z. Ruan, N. Zhou, X. Liu and Z. Song, "A comprehensive-integrated buffer management strategy for opportunistic networks", *EURASIP Journal on Wireless Communications and Networking,* pp. 103-103, 2013.

[76] J. F. Naves, I. M. Moraes, and C. Albuquerque, "LPS and LRF: efficient buffer management policies for Delay and Disruption Tolerant Networks", *IEEE 37th Conference on Local Computer Networks (LCN)*, Clearwater, FL , USA, October 2012.

[77] G. Fathima, R. S. D. Wahidabanu, "Buffer management For preferential delivery in opportunistic delay tolerant networks," *International Journal of Wireless & Mobile Networks,* Vol. 3, no. 5, p. 15, 2011.

[78] C. Rohner, F. Bjurefors, P. Gunningberg, L. McNamara, E. Nordström, "Making the most of your contacts: transfer ordering in data-centric

opportunistic networks", *ACM Proceedings of the third nternational workshop on Mobile Opportunistic Networks*, pp. 53-60, Zurich, Switzerland, March 2012.

[79] A. Elwhishi, P.-Han Ho, K. Naik, and B. Shihada, "A novel message scheduling framework for delay tolerant networks routing," *IEEE Transactions on Parallel and Distributed Systems,* vol. 24, no. 5, pp. 871-880, 2013.

[80] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks", *5th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Francisco, CA, USA, June 2008.

[81] A. Krifa, C. Barakat and T. Spyropoulos, "Message drop and scheduling in DTNs: theory and practice", *IEEE Transactions on Mobile Computing,* vol. 11, no. 9, pp. 1470-1483, 2012.

[82] L. Yin, H.-Mei Lu, Y.-Da Cao, and J.-Min Gao, "Buffer scheduling policy in DTN routing protocols", *IEEE 2nd International Conference on Future Computer and Communication (ICFCC)*, vol. 2, Wuhan, China, May 2010.

[83] D. Pan, W. Cao, H. Zhang, and M. Lin, "Buffer management and hybrid probability choice routing for packet delivery in opportunistic networks", *Mathematical Problems in Engineering,* vol. 2012, p. 1, 2012.

[84] Y. Li, M. Qian, D. Jin, L. Su, and L. Zeng, "Adaptive optimal buffer management policies for realistic DTN", *IEEE Global Telecommunications Conference (GLOBECOM)*, Honolulu, Hawaii, USA, Novermber-December 2009.

[85] D. Kim, H. Park, and I. Yeom, "Minimizing the impact of buffer overflow in dtn", *Proceeding of International Conference on Future Internet Technologies (CFI)*, 2008.

[86] A. Balasubramanian, B. Levine and A. Venkataramani, "DTN routing as a resource allocation problem," *ACM SIGCOMM Computer Communication Review,* vol. 37, no. 4, pp. 373-384, 2007.

[87] Y. Liu, J. Wang, S. Zhang, H. Zhou, "A buffer management scheme Based on message transmission status in delay tolerant networks", *IEEE Global Telecommunications Conference (GLOBECOM)*, Houston, TX, USA, Dec 2011.

[88] K. Shin and S. Kim, "Enhanced buffer management policy that utilizes message properties for delay-tolerant networks," *IET Communications,* vol. 5, no. 6, pp. 753-759, 2011.

[89] R. Ramanathan, R. Hansen, P. Basu, R. R.-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks", *ACM Proceeding of the 1st international MobiSys Workshop on Mobile Opportunistic Networking (MobiOpp)*, San Juan, Puerto Rico, June 2007.

[90] J. A. Dias, J. N. Isento, V. N. G. J. Soares, and J. J. P. C. Rodrigues, "Impact of scheduling and dropping policies on the performance of vehicular delay-tolerant networks", *IEEE International Conference on* in *Communications (ICC)*, Kyoto, Japan, June 2011.

[91] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks", *IEEE Transactions on Mobile Computing,* vol. 10, no. 11, pp. 1576-1589, 2011.

[92] M. C. Gonzalez, C. A. Hidalgo, and A. L. Barabási, "Understanding individual human mobility patterns", *Nature,* vol. 453, pp. 779-782, 2008.

[93] R. A. Howard, "Dynamic probabilistic system", *International Journal for Numerical Methods in Engineering,* vol. 2, pp. 303-304, 1972.

[94] I. Cardei,C. Liu, J. Wu, and Q.Yuan, "DTN Routing with Probabilistic Trajectory Prediction", *Springer Proceedings of the Third International Conference on Wireless Algorithms, Systems, and Applications*, pp. 40-51, Dallas, Texas, October 2008.

[95] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation", *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pp. 1-10, Rome, Italy, March 2009.

[96] J. M. François and G. Leduc, "Delivery Guarantees in Predictable Disruption Tolerant Networks", *Springer networking, Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, vol. 4479, pp. 167-178, 2007.

[97] S. Iranmanesh, R. Raad, and K-W Chin, "A Novel Destination-Based Routing Protocol (DBRP) in DTNs", *IEEE the International Symposium on Communications and Information Technologies (ISCIT)*, QLD, Australia, October 2012.

[98] A. Keränen, and J. Ott, "Increasing Reality for DTN Protocol Simulations", Helsinki University of Technology, Technical Report, July, 2007.

[99] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks", *IEEE INFOCOM*, San Francisco, CA, USA, March-April 2003.

[100] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research", *Wireless Communications and Mobile Computing,* vol. 2, no. 5, pp. 483-502, 2002.

[101] M. Taboga, "Lectures on probability theory and mathematical statistics", *CreateSpace Independent Publishing Platform*, Second edition, ISBN-13: 978-1480215238, 2012.

[102] F. Ekman, A.Keränen, J. Karvo, and J. Ott, "Working day movement model", *ACM SIGMOBILE Proceedings of the 1st workshop on Mobility models*, Hong Kong, China, pp. 33-40, May 2008.

[103] R. Diana, E. Lochin, L. Franck, C. Baudoin, E. Dubois, and P. Gelard, "A DTN Routing Scheme for Quasi-Deterministic Networks with Application to LEO Satellites Topology", *IEEE Vehicular Technology Conference (VTC)*, Quebec City, QC, September 2012.

[104] L. Huai-En, C. Chen, J. Chang, C. Shen, and R. Jan, "Shortest path routing with reliability requirement in delay tolerant networks", *IEEE First International Conference on Future Information Networks (ICFIN)*, Beijing, China, October 2009.

[105] M. Zarafshan-Araki and K.-W. Chin, "TrainNet: A transport system for delivering non real-time data", *Elsevier Computer Communications,* vol. 33, no. 15, pp. 1850-1863, 2010.