2012

Data Dissemination in Delay Tolerant Networks

Zhenxin Feng
University of Wollongong

Data Dissemination in Delay Tolerant Networks

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

Doctor of Philosophy

from

THE UNIVERSITY OF WOLLONGONG

by

Zhenxin Feng
Masters of Engineering (Computer Science),
Bachelor of Engineering (Computer Science)

SCHOOL OF ELECTRICAL, COMPUTER
AND TELECOMMUNICATIONS ENGINEERING
2012

To my parents and my wife

Abstract

Delay Tolerant Networks (DTNs) are characterized by large delays, frequent disruptions and lack of contemporaneous paths between nodes. Moreover, nodes may have limited computational power, storage, and battery capacity. Despite these challenges, DTNs have found applications in areas such as animal tracking, surveillance and facilitating education in remote areas. In these applications, a critical problem is routing or data dissemination. Indeed, any routing protocols must aim to transmit data efficiently and maximize data delivery. The challenge, however, is that nodes are only connected intermittently, and do not have a contemporaneous path. Moreover, any established path may not remain valid after a transmission, and nodes may experience large delays before encountering one another. As a result, existing routing protocols cannot be retrofitted to work in DTNs.

To date, a primary solution for routing in DTNs is epidemic-based routing protocols because of their simplicity, low delays and little to no reliance on specific nodes. A key observation of past research on epidemic routing protocols is that they are not evaluated on a unified framework. Specifically, no work has compared the performance of epidemic routing protocols using both the Random Way Point (RWP) model and trace files. Henceforth, this thesis has conducted a comprehensive study that employs both RWP and trace files. In particular, this thesis is the first to compare the performance of epidemic-based routing protocols using a custom-built simulator that moves nodes according to a trace-file and the RWP model. Moreover, it compares these protocols using the same set of parameters; e.g., node numbers, load and buffer space. The extensive simulation studies reveal the following limitations: high buffer occupancy level, premature discard of bundles, inefficient use of immunity tables to purge redundant bundles, low delivery ratio at high loads, and poor adaptivity to changing network parameters. This thesis also outlines three novel enhancements to address the limitations of epidemic-based protocols. First, it introduces a mechanism that adapts the Time to Live (TTL) of bundles dynamically according to a node's encounter interval. The intuition here is that bundles should be buffered according to the interval between two encounters. That is, when nodes experience long inter-contact intervals, bundles will have a large TTL value, whilst short intervals result in a small TTL value. Simulation results show that epidemic with dynamic TTL improves delivery ratio by more than 20%. The second enhancement combines Encounter Count (EC) with TTL to reduce buffer occupancy level and increase bundle delivery ratio. The resulting combination is able to reduce buffer occupancy level by 40%. More importantly, it dramatically improves the delivery ratio by at least 40% at high loads. The last enhancement uses immunity tables to carry cumulative acknowledgments. This has the effect of facilitating buffer management, and more importantly, allows a node to delete multiple bundles upon receiving one immunity table. This is an improvement over past approaches as nodes need to receive N immunity tables in order to delete N bundles. Extensive experiments using both the RWP model and trace-file confirm the superiority of these enhancements over existing epidemic routing protocols.

Epidemic-based routing protocols can also be used for transmitting multicast bundles. This thesis presents three key findings concerning the delivery ratio of epidemic-based multicast protocols. Specifically, (i) subscriber nodes must act as relays. Simulation results show that bundle delivery ratio is only 57% as compared to 100% when they act as relays, (ii) the use of anti-entropy contributes to a rise in delivery ratio to 100%, which is 57% higher than when nodes do not use anti-entropy, and (iii) higher number of relay nodes improve delivery ratio. From extensive simulation studies, it can be seen that when the number of relay nodes increases from 10 to 30, the delivery ratio increased from 43% to

98%. This thesis also presents new findings related to multicast group sizes and delivery ratios. The presented results show that the impact of multicast group size on bundle delivery ratio is dependent upon whether subscribers forward bundles - i.e., whether they are relays or sinks. Specifically, when subscribers work as sinks, epidemic with Time-to-Live (TTL), epidemic with immunity, epidemic with Encounter Count (EC) threshold, and epidemic with cumulative immunity, have poor bundle delivery ratio. On the other hand, for epidemic with immunity, it has a low bundle delivery ratio when subscribers act as relays. In regards to buffer occupancy level in multicast scenarios, simulation results show that multicast group size, anti-entropy session and subscribers forwarding policies have a significant impact on the buffer occupancy level of relay nodes. In particular, all protocols experience high buffer occupancy level. To address this problem, each bundle is assigned an EC quota. In other words, EC quota bounds the number of times in which a bundle is exchanged by relay nodes.

Lastly, this thesis focuses on routing requests and replies in DTN based information retrieval systems. Specifically, this thesis presents the first data centric information retrieval system called Distributed Data-Centric Information Retrieval (DDC-IR) system. Nodes only process a query if its similarity value matches that of the query. This ensures only nodes that have a high probability of resolving a query process and transmit the query. Experiment results show that DDC-IR is able to resolve 50% more queries and has 80% lower buffer occupancy level than prior work. More importantly, DDC-IR is able to support four query types: complex, unique, aggregated and continuous. This is a significant contribution over past approaches that only targeted one query type. Apart from that, DDC-IR supports a novel caching policy, in which nodes cache popular queries to improve the retrieval success ratio. This thesis also investigated the influence of the number of sub-queries and the number of querying nodes on query resolution time. When the number of querying nodes increases, the retrieval success rate reduces from 100% to 20%. When the number of sub-queries in a complex query increases from five to nine, DDC-IR requires 50% more time to resolve queries. In comparison, previous IR systems are unable to resolve any queries.

Statement of Originality

This is to certify that the work described in this thesis is entirely my own, except where due reference is made in the text.

No work in this thesis has been submitted for a degree to any other university or institution.

Signed

Zhenxin Feng
20^{th} February, 2012

Acknowledgments

I would like to thank all the people who gave their support and encouragement during my three-year Ph.D study.

To my supervisor, A/Prof. Kwan-Wu Chin. His attitude towards knowledge, research and truth deeply influenced me. His strict guidance has enriched my life.

To my co-supervisor, Dr. Raad Raad for helping in time of needs.

To my parents, their endless love gave me the encouragement to confront any difficulties and tough situations.

To my wife, Jing. Because of her sacrifice, I could study abroad without any worries.

Contents

List of Figures

List of Tables

List of Abbreviations

DTNs	Delay Tolerant Networks
RAM	Random Access Memory
SWIM	Shared Wireless Infostation Model
WSNs	Wireless Sensor Networks
OSPF	Open Shortest Path First
DSDV	Destination-Sequenced Distance Vector
AODV	Ad-Hoc On-Demand Distance Vector
IR	Information Retrieval
RWP	Random WayPoint
TTL	Time To Live
EC	Encounter Count
DDC-IR	Distributed Data-Centric Information Retrieval
IETF	Internet Engineering Task Force
IRTF	Internet Research Task Force
DFs	Data Ferries
QoS	Quality of Service
PQERPV	(p, q)-epidemic with vaccination routing protocol
SERAC	Epidemic Routing with Active Curing
ACK	acknowledgment message
DLE	Drop-Least-Encountered
PREP	Prioritized Epidemic
AA	Average Availability
MF	Message Ferrying

NIMF	Node-Initiated Message Ferrying
FIMF	Ferry-Initiated Message Ferrying
SIRA	Single Route
MURA	Multiple Route
NRA	Nodes Route
FRA	Ferries Route
MDDV	Mobility-Centric Data Dissemination Algorithm for Vehicle Network
GPS	Global Position System
MV	Meeting-Visit routing protocol
FCFS	First Come First Serve
LMMF	Local Max-Min Fair Algorithm
GMMF	Global Max-Min Fair
WGMMF	Weighted Global Max-Min Fair
LET	Least Energy Tree
MHT	Minimum Hop Tree
APs	access points
LEO	Low Earth Orbit
IPN	Inter-Planetary Network
MRP	Mobile Relay Protocol
CAR	Context-Aware Routing
UDM	Utility-based Distributed routing algorithm with Multi-copies
VoRo	Vector Routing
SD-MPAR	Similarity Degree-based Mobility Pattern Aware Routing
EBR	Encounter-Based Routing protocol
SEPR	Shortest Expected Path Routing protocol
ME/DLE	most Encountered/Drop Least Encountered algorithm
RAPID	Resource Allocation Protocol for Intentional DTN
TM	Temporal Membership
TD	Temporal Delivery Model
CMD	Current-Member Delivery Model
UBR	Unicast-Based Routing
STBR	Static Tree-Based Routing

GBR	Group-Based Routing
DTBR	Dynamic Tree-Based Routing
DPSP	DTN Pub/Sub Protocol
UoW	Univerisity of Wollongong

Chapter 1

Introduction

1.1 Delay Tolerant Networks

Delay Tolerant Networks (DTNs) are characterized by large delays, frequent disruptions and lack of contemporaneous paths between nodes. In addition, nodes may have limited computational power, storage, and battery capacity [5] [6] [7] [8]. To illustrate some of these characteristics, consider forming a network using the vehicles of Fig. 1.1. All vehicles, e.g., buses and cars, are equipped with a radio transceiver, which allows them to communicate with each other, and also to access points, which have connectivity to the Internet and are planted strategically in different parts of the city. In this network, all vehicles will help each other forward messages and also to access points. Given the limited transmission range, the intermittent connectivity of vehicles and location of access points, messages will experience significant delays.

Another key characteristic of DTNs is the so called *store-carry-forward* model used to propagate messages. That is, a vehicle may have to store and carry a message for some distance before encountering and passing the message onto another vehicle or access point. In this regards, a key mobility pattern that can be exploited by routing protocols is the predictable mobility pattern and schedule of buses. In addition, any routing protocols will have to consider the link capacity and duration of each connection, which is governed by channel condition and vehicle speed.

A DTN can also be formed using people. This can be easily realized given the ubiquity of smart phones equipped with a plethora of sensors and transceivers. Hence, they can be used to monitor traffic, crowd, air pollution and spread of diseases to name a few. Unlike vehicle based DTNs, smart devices have resource constraints; e.g., limited battery life. Moreover, people will have varying contact duration and frequency. That is, their movement pattern will be less predictable than vehicles. Consider User-A in Fig. 1.1 who wants to send a file to one or more students attending the University of Wollongong (UoW). Also shown is a possible transmission path, which depends on encounters with other users of the DTN. Inevitably, the resulting topology or path taken will be random in nature and changes with space and time. More specifically, it is difficult to predict as it depends on following three factors. Firstly, the interval time between two encounters is large. Interestingly, the authors of [9] showed that nodes/students who are attending the University of Cambridge are not always connected, and hence they experience large delays between meetings. For example, nodes/students may connect during a class, and disconnect when the class finishes. The next class may be eight hours away. Secondly, the movement of nodes and encounter duration are random. For instance, two good friends may remain in contact for a lot longer as compared to other students. Thirdly, nodes exhibit a mobility pattern that coincides with meeting times, e.g., lectures, and path to a given classroom.

Despite these challenging conditions, as shown in Tab. 1.1, researchers have proposed various DTNs and applications. For example, in ZebraNet [10], zebras have custom sensors that track their movement patterns and locations. A device carried by a person or a mobile base station is then used to collect the said tracking data. As mobile base stations have limited communication range, zebras exchange information with other zebras until they encounter a mobile base station. Given that zebras and a mobile station rendezvous randomly, i.e., they may not meet each other for days or weeks, tracking data incurs significant delays before scientists are able to collect them. Moreover, as the movements of

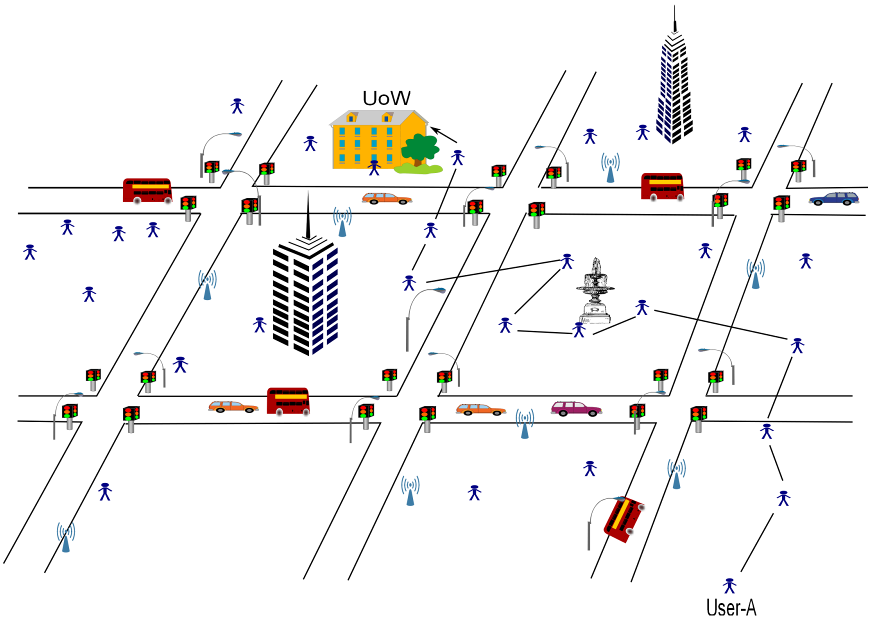

Figure 1.1 An example comprising of DTNs formed by vehicles and people.

zebras are unpredictable, links are established intermittently and hence there are no end-to-end paths from zebras to mobile base stations. In addition, ZebraNet also has storage, bandwidth and energy constraints. Specifically, the sensors on each zebra have a lifetime of only one month, are equipped with a 1MB flash Random Access Memory (RAM) and have a data transmission rate ranging from 2.4 to 19.2 kilobits per seconds. Another wildlife example is SWIM [11], where a sensor network is used to monitor whales. SWIM combines two kinds of nodes: (i) sensors, and (ii) infostations. The sensors are attached to whales, and the infostations are used to collect data from passing whales. In [12], sensors networks are used to monitor water pollution and noise level in urban areas.

There are also a number of applications that involve special nodes. For example, DakNet [13] uses vehicles or data ferries to provide low-cost data delivery between rural villages. In each village, a kiosk is used by villagers to store messages and send data to visiting data ferries, which then uploads the data they have collected onto the Internet. As data is carried by data ferries, it experiences a much higher delay than conventional networks. To clarify, the delays incurred by messages are affected by several data ferry parameters: (i) routes taken to reach a village, (ii) schedules, (iii) speed, (iv) number of ferries, and (v) distance between kiosks.

In a different work, the Pollen network [14] uses humans as data ferries, where mobile devices carried by humans exchange information with each other, and/or with a fixed network. Other projects that use data ferries include KioskNet [15], MotoPost [16], Wizzy Net [17], Widernet [18], Digital Study Hall [19], DigitalGreen [19], Body Sensors [20], and Digital Polyclinic [19]. Interestingly, in [21], the authors showed that sending a 32 Gigabytes message in the same city using a pigeon with a portable disk to have higher bandwidth and shorter delivery time than transmitting the same message via the Internet.

1.2 Motivation

A key problem in DTNs is data dissemination. Specifically, the aforementioned characteristics require data dissemination protocols that can address the following challenges:

- *Stochastic and dynamic topologies.* Nodes are mobile and can engage in various mobility patterns [23]. For example, nodes may be vehicles on a freeway or wild animals roaming in a national park. The resulting topology is therefore unpredictable, characterized by uncontrolled node movements, large delays and arbitrary disconnections.

- *Limited resources.* This challenge requires data dissemination to be efficient. In other words, nodes must utilize their limited hardware resources such as CPU, memory and battery efficiently. For example, in WSNs, nodes can be located in an open environment for years before data are collected, and hence requires nodes to carefully manage their energy usage. Additionally, a good data dissemination scheme will leverage the resources of multiple nodes. For example, nodes may choose to shift some of their stored messages to other nodes to free up memory or to reduce transmission cost.

Table 1.1 A sample of DTN applications.

DTN Applications	**Purpose**	**DTN Nodes**	**Delay**	**Data Ferries Routes**
SeNDT [12]	Water pollution and noise monitor	Chemical sensors, noise detectors and people with mobile devices	Days or months	Random, depending on data requirement
DakNet [13]	Digital communications for rural areas	Coaches, motorbikes, ox carts, kiosks, Access Points (APs)	Minutes or hours	Semi-random, depending on the transport vehicle used
KioskNet [15]		Buses, people with hand-held devices, kiosks, desktop computers with a dial-up connection	Hours or day	Scheduled as per bus timetable or random according to people movement
MotoPost [16]			Hours	
Pollen [14]	Personal communications	People with mobile devices or PDA	Hours or days	Random, as per the environment; e.g., in an office
Wizzy Net [17]	Facilitate education in rural schools	People with memory sticks	Hours or days	Semi-random. Data ferries that visit villages as needed to disseminate information on agriculture or healthcare
Digital Study Hall [19]		People with DVDs and players		
Digital Green [19]	Disseminating agricultural information to rural areas		Days or months	
Digital Polyclinic [19]	Providing healthcare information to rural areas			
Widernet [18]	To improve educational communication systems in Africa	Desktop computers with sufficient storage to store web sites with rich educational contents		
TrainNet [22]	To transport massive amount of non real-time data over large geographical areas	Trains, stations	Minutes or hours	Fixed, as per railway lines

- *Limited topological information*, which compounds the difficulty of finding routing metrics that accurately reflect network conditions. A path from a source to a destination can be static or dynamic. However, as pointed out in [24], without topology information, static routes are not suitable for dynamic topologies. Another challenge is the lack of up to date topological information which can be used to calculate the best path to a given destination. Therefore, as will be evident later, dynamic routing protocols tend to use local metrics, such as the number of times nodes have encountered each other.

- *Variable and uncertain connection duration*. In data dissemination, nodes need to decide whether to transmit all or a subset of stored messages when they encounter each other. For example, as zebras in ZebraNet [10] meet for a limited time period, the routing protocol has to decide which data to forward in order to maximize delivery probability.

Fig. 1.2 demonstrates the difficulties of disseminating data in DTNs. For example, the movement of mobile nodes or students in Fig. 1.2 is uncontrolled, and exhibits varying temporal and spatial characteristics. For example, when students are on campus, they can stay in the library or classroom for a long period of time, and may encounter other nodes readily to exchange messages. Moreover, students may attend lectures and tutorials and hence exhibit group like movement patterns. On the other hand, when students are off campus, the sojourn time at each location is likely to be different to those on campus. As students pass each other, the number of messages exchanged will depend on how long they remain in each other's transmission range - as determined by the communication technology and mobility speed. More importantly, they do not have any topological information which they can use to disseminate a packet efficiently. For example, in Fig. 1.2, there is no contemporaneous path between node-A and node-B, and hence, any developed routing protocol must rely on contacts between nodes. A challenging issue here is that nodes have independent movement patterns. Hence, nodes experience variable and uncertain rendezvous periods and frequencies. For instance, nodes A, B and

C in Fig. 1.2 have overlapping physical locations as they move along their respective path. However, they may not encounter each other, and should they meet, the transmission duration is unpredictable. Note, from here on, the term "bundle" is used to denote messages in DTNs.

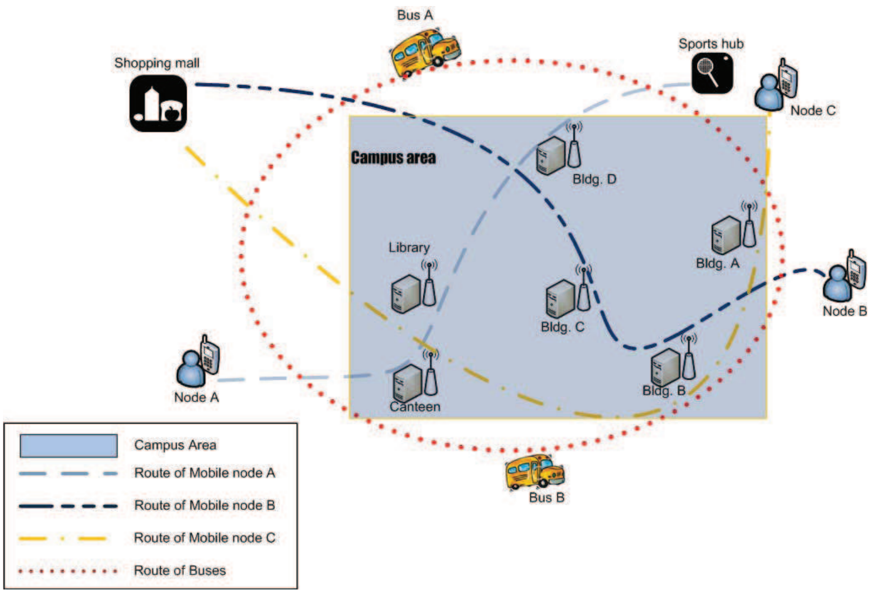

Figure 1.2 An example DTN operating on a university campus.

There are three data dissemination patterns: one-to-one, one-to-many and information retrieval. According to the definition of Internet Engineering Task Force/Internet Research Task Force(IETF/IRTF), bundles are the metadata that is used for wrapping information from other layers and to compress them into a data block. The main goal of data dissemination in one-to-one or unicast is to maximize delivery ratio from a source to a receiver. However, the aforementioned characteristics of DTNs mean existing unicast protocols developed for conventional networks cannot be applied directly. Firstly, some conventional protocols need link states to be propagated throughout the network; for example, nodes using Open Shortest Path First (OSPF) [25] and Destination-Sequenced Distance Vector (DSDV) [26] require a map of the topology in order to construct a shortest path tree. Secondly, unlike conventional networks with stable topologies, DTN topologies change frequently. As a result, nodes are required to send updates and re-calculate routing paths frequently, which leads to convergence issues. Thirdly, existing protocols assume end-to-end delays are in the order of a few hundreds of milliseconds. For example, Ad-Hoc On-Demand Distance Vector (AODV) [27] assumes route construction can be carried out within tens of milliseconds.

The second data dissemination pattern is multicast, in which data is transmitted to multiple nodes or destinations. Current multicast protocols can be classified into three categories. The first, and most popular, is tree-based multicast protocols, where they form a multicast spanning tree, and all multicast bundles are dispatched along the tree. Parent nodes usually have ample energy, many connections to other nodes or have the least cost path to other parent nodes. However, due to the characteristics of DTNs, it is impractical to maintain a stable spanning tree, and hence it is only suitable for networks where nodes encounter each other frequently. The second category relies on data ferries that have higher mobility and connectivity to other nodes and large energy and storage capacity. Data ferries collect multicast bundles from a source and forward them to subscribers upon encountering them. However, to transmit bundles to all subscribers, data ferries need to deliver a bundle to each subscriber one by one, which becomes impractical with increasing multicast group size. Indeed, this approach is also influenced by the number of ferries and their mobility patterns. The last category is epidemic based protocols. These protocols are simple, have low latency and do not rely on special nodes. The downside, however, is their high buffer occupancy level.

The last data dissemination pattern is information retrieval, in which nodes require data from other nodes according to queries. To date, only two works [28] [29] have considered information retrieval in DTNs. These works, however, suffer from several critical problems. First, the transmission of queries and replies are separate to the query resolution process. This may result in query resolution failure because a node with the highest connection to other nodes may not necessarily have the required data to resolve the query. Secondly, all queries are regarded as one-shot single queries, i.e., the query can be resolved only by a single node. However, to resolve other types of queries such as aggregated and complex queries, prior IR systems require access to data from multiple nodes as opposed to one specific node. Thirdly, although past address-centric IR systems avoid flooding by limiting the number of duplicates, there is no buffer policy to remove resolved or staled queries and duplicates.

1.3 Research Statement

To date, researchers have proposed a number of routing protocols for DTNs. As explained in Chapter 2, they can be divided into three main categories: (i) epidemic (ii) data ferries and (iii) statistical. In the first category, once nodes encounter a peer, they will exchange bundles with each other. Data ferries protocols can also be divided into two sub-categories: ferries-initiated, in which data ferries approach nodes to fetch bundles, and nodes-initiated, in which nodes intentionally move toward data ferries. In the last category, nodes record their encounter history and calculate a suitable next-hop for bundles that maximizes bundle delivery ratio.

A key observation of existing routing protocols research is the lack of research that evaluates epidemic routing protocols using a unified framework. In particular, epidemic routing protocols have been tested in different scenarios in terms of number of nodes, network area, buffer size and bundle size; see Tab. 1.2. As a result, it is very difficult to compare epidemic routing protocols objectively. Besides that, no work has compared the performance of epidemic routing protocols using both the Random Way Point (RWP) model [30] and trace files.

Table 1.2 Experiment parameters used in studies such as [1] [2] [3] [4].

Number of Nodes	≤100
Mobility Pattern	Random Waypoint
Network Area	≤50km^2
Transmission Range	≤300m
Metircs	Delivery ratio, average delay, time to deliver all bundles
Buffer Size	Infinite or up to 5 MB
Bundle Size	≤14MB

Other than that, there is also little focus on epidemic based multicast routing protocols. That is, previous research, e.g., [31] and [31], has not studied using epidemic based protocols for delivering bundles to multicast subscribers. In particular, they have not investigated (i) epidemic variants that use immunity messages to purge bundles, (ii) the impact of anti-entropy session on bundle delivery, (iii) whether subscribers are required to act as relays, (iv) the buffer occupancy level of relays, and (v) how to reduce the high buffer occupancy level of relay nodes.

Another observation from works in DTNs is the lack of protocols for information retrieval. In particular, little work has addressed the following query types [32]: (i) *continuous versus one-shot*. If a query requires source nodes to continuously respond and transmit information, the said query is called "continuous", otherwise, it is a "one-shot" query. (ii) *aggregate vs. non-aggregate*. Aggregate queries allow data from different source nodes to be summarized by an intermediate node. (iii) *complex vs. simple*. A complex query is comprised of sub-queries or simple queries. Specifically, complex queries are represented as a conjunctive normal form. (iv) *replicated vs. unique data queries*. The problem addressed in this thesis is ostensibly different from past works [33] [29] [28], which have thus far considered address centric forwarding, whereby one or more bundles are addressed to a single node. In contrast, this thesis proposes a data centric information system that supports all four query types and do not use nodes' addresses.

1.4 Contributions

Henceforth, in light of the aforementioned observations and limitations of existing data dissemination works, this thesis makes the following contributions:

- It uses a unified framework to compare the performance of epidemic-based routing protocols. Specifically, it compares all epidemic routing protocols using a custom-built simulator that moves nodes according to a trace-file and the RWP model. Moreover, it compares these protocols using the same set of parameters; e.g., node numbers, load and buffer space. From extensive simulation studies, it identifies the following limitations with existing epidemic based routing protocols: high buffer occupancy level, premature discard of bundles, inefficient use of immunity tables to purge redundant bundles, low delivery ratio at high loads, and poor adaptivity to changing network parameters.

- It outlines three novel enhancements to address the limitations of epidemic-based protocols. First, it introduces a mechanism that adapts the Time to Live (TTL) of bundles dynamically according to a node's encounter interval. The intuition here is that bundles should be buffered according to the interval between two encounters. That is, when nodes experience a long inter-contact interval, bundles will have a larger TTL value, whilst a short interval results in small TTL values. Simulation results show that epidemic with dynamic TTL improves delivery ratio by more than 20%. The second enhancement combines Encounter Count (EC) with TTL to reduce buffer occupancy level and increase bundle delivery ratio. The resulting combination is able to reduce buffer occupancy level by 40%. More importantly, it dramatically improves the delivery ratio by at least 40% at high loads. The last enhancement uses immunity tables to carry a cumulative acknowledgment. This has the effect of facilitating buffer discard policy, and more importantly, allows a node to delete multiple bundles upon receiving one immunity table. This is an improvement over past studies as nodes need to receive N immunity tables in order to delete N bundles. Extensive experiments using both the RWP model and trace-file confirm the superiority of these enhancements over existing epidemic routing protocols.

- It presents three key findings concerning the delivery ratio of epidemic-based multicasting protocols. Specifically, (i) subscriber nodes must act as relays. Simulation results show that bundle delivery ratio is only 57% as compared to 100% when they act as relays, (ii) the use of anti-entropy contributes to a rise in delivery ratio to 100%, which is 57% higher than when nodes do not use anti-entropy, and (iii) higher number of relay nodes improve delivery ratio. In experiments, it can be seen that when the number of relay nodes increases from 10 to 30, the delivery ratio rises from 43% to 98%.

- It presents new findings relating to multicast group sizes and delivery ratios. The presented results show that the impact of multicast group size on bundle delivery ratio is dependent upon whether subscribers forward bundles - i.e., whether they are relays or sinks. Specifically, when subscribers work as sinks, epidemic with TTL, epidemic with immunity, epidemic with EC threshold, and epidemic with cumulative immunity, have poor bundle delivery ratio. On the other hand, for epidemic with immunity, it has a low bundle delivery ratio when subscribers act as relays.

- It presents new findings in regards to buffer occupancy level in multicast scenarios. Simulation results show that multicast group size, anti-entropy session and subscribers forwarding policies have a significant impact on the buffer occupancy level of relay nodes. For example, large multicast group sizes have an influence on the buffer occupancy level of relay nodes. This too is affected by subscribers' forwarding policies. Specifically, when subscribers are sinks, the buffer occupancy level of nodes is lower with increasing multicast group sizes, while there is no impact when subscribers are relays. Apart from that, this thesis found that, assigning each bundle with an EC quota can effectively reduce nodes' buffer occupancy level. In other words, EC quota bounds the number of times in which a bundle is exchanged by relay nodes.

- It presents the first data centric information retrieval system. The proposed system is able to resolve 50% more queries and has 80% lower buffer occupancy level. Apart from that, all four types of queries are supported. This thesis also investigated the influence of the number of sub-queries on query resolution time. That is, when the number of sub-queries in a complex query increases from five to nine, the new system uses 50% more time to resolve the query. In comparison, previous IR systems are unable to resolve any queries.

1.5 Publications

This thesis has resulted in the following papers:

1. **Zhenxin Feng** and Kwan-Wu Chin. *A unified study of Epidemic routing protocols and their enhancements.* In the proceedings of The 13th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing(PDSEC), Shanghai, China, May, 2012.

2. **Zhenxin Feng** and Kwan-Wu Chin. *On the Performance of Epidemic Based Routing Protocols for Delivering Multicast Bundles in Delay Tolerant Networks.* IEEE Transactions on Parallel and Distributed Systems. Under second review.

3. **Zhenxin Feng** and Kwan-Wu Chin. *State-of-the-art routing protocols for challenged networks.* Journal of Elsevier Computer Networks. Under second review.

4. **Zhenxin Feng** and Kwan-Wu Chin. *A Novel Data Centric Information Retrieval Protocol for Queries in Delay Tolerant Networks.* Journal of Elsevier Ad Hoc Networks. Under review.

1.6 Thesis Structure

The remainder of the thesis is organized as follows:

1. *Chapter 2.* This chapter reviews routing protocols for DTNs. Specifically, it provides an extensive qualitative comparison of all protocols, highlight their experimental setup and outline their deficiencies in terms of design and research methodology.

2. *Chapter 3.* This chapter presents the analysis of epidemic-based routing protocols. In addition, it proposes three enhancements.

3. *Chapter 4.* This chapter analyses epidemic-based multicasting protocols. In particular, it proposes a new multicasting protocol - epidemic with EC Quota.

4. *Chapter 5.* This chapter presents a novel information retrieval system for DTNs. Specifically, this chapter proposes Distributed Data-Centric Information Retrieval (DDC-IR), a data centric IR system that supports all query types.

5. *Chapter 6.* This chapter concludes the thesis, and provides a summary of research outcomes and future research directions.

Chapter 2

Data Dissemination

Routing or data dissemination protocols are critical to the operation of DTNs. To this end, this chapter reviews unicast and multicast routing protocols that are designed specifically to run in delay tolerant or challenged networks. It provides an extensive qualitative comparison of all protocols, highlight their experimental setup and outlines their deficiencies in terms of design and research methodology.

The remainder of this chapter is structured as follows. Section 2.1 first presents an overview and taxonomy of routing protocols before delving into the details of epidemic routing protocols, and their variants in Section 2.1.2. This is then followed by data ferries based protocols in Section 2.1.3. That is, routing protocols that assume the existence of special nodes with ample resources and deterministic trajectories. After that in Section 2.1.4, this chapter reviews protocols that dynamically maintain historical information of past encounters to aid their future forwarding decisions. Then Section 2.2 and Section 2.3 review multicast routing protocols and information retrieval system respectively. Finally, Section 2.4 concludes the chapter.

2.1 Unicast Routing Protocols

2.1.1 Overview

Current DTN routing protocols can be divided into three categories: (i) epidemic, (ii) data ferries, and (iii) statistical. Tab. 2.1 summarizes the key characteristics of each category. We see that the three categories have different features, advantages and disadvantages.

Nodes in category (i) are assumed to have uniform resource and movement patterns. Moreover, they cooperate to route bundles for their neighbors. For example, in Fig. 2.1, at $t = 1$, assume node B wants to forward a bundle to node E. One approach that node B can adopt is to simply forward the bundle to any nodes it encounters; i.e., floods the bundle as widely as possible. The bundle arrives at node E following the path B–C–A–E at $t = 3$. Unlike conventional networks, this path, however, is likely to change in subsequent bundle transmissions. In this respect, the key research objective, as discussed in Section 2.1.2, is to design an efficient flooding based protocol that meets the following goals: a) message delivery rate is maximized, b) message latency is minimized, and c) the total resource, especially memory or energy expenditure, consumed is minimal. In general, epidemic routing protocols have low delays, but high resource consumption. Hence, designing a buffer management policy that balances resource consumption and delivery ratio is a fundamental problem. Moreover, said policy needs to purge staled bundles reliably. Otherwise, prematurely removing bundles may have a negative impact on delivery ratio.

Routing protocols in category (ii) take advantage of resource rich, mobile nodes called Data Ferries (DFs), which act as a communication channel between nodes or disparate networks. These networks could be located on different planets or represent rural villages. Examples of DFs based DTNs are shown in Fig. 2.2. Four networks are serviced by a DF (bus), which tours each network periodically. Note that one can reduce delays further by adding more DFs or buses. Each network, e.g., Net Y, can be serviced by an independent DF. That is, instead of

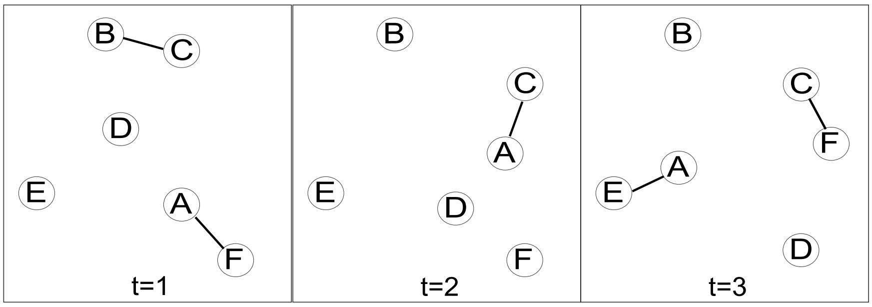

Figure 2.1 A DTN with homogeneous nodes. A solid line shows connectivity between nodes.

networks, a DF (car) is tasked with collecting data from nodes directly. Apart from that, nodes can rely on wireless communications, i.e., Net Z, and only use a DF for inter-cluster communications; see Section 2.1.3.5 for more details. The figure also illustrates two kinds of DF movements: active and passive. In the former, data ferries actively approach source or destination nodes; e.g., the bus. In the latter model, nodes intentionally move toward data ferries, as illustrated by nodes in "Net W". In both types of data ferries, they have a predictable schedule which a routing protocol can then exploit to provide some form of Quality of Service (QoS) guarantees.

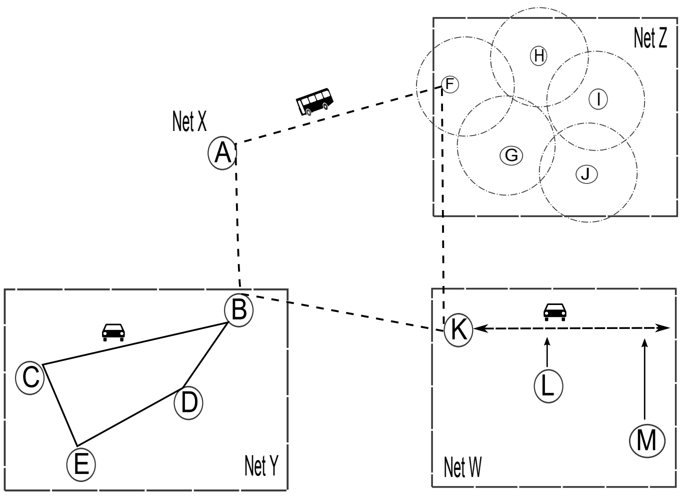

Figure 2.2 Example of DF based DTNs.

The last category, i.e., (iii), of routing protocols utilizes statistical methods to avoid arbitrary flooding. The key assumption is that nodes in a DTN will always encounter one another. Moreover, nodes are homogeneous in that they have similar resources. Hence, each node can compile a set of statistics or metrics regarding its rendezvous time with other nodes. For example, in Figure 2.1, node A has a high connectivity. That is, it meets other nodes frequently, and thereby, making it ideal as a bundle carrier. As shown in Section 2.1.4, nodes may forward a bundle to a neighbor based on statistical information such as next hop forwarding probability. Other metrics include the number of times a node has encountered a given node, and the duration in which a node remains connected to a given neighbor. Moreover, routing protocols may consider the storage capacity, energy, bandwidth and/or type of nodes. As a result, routing protocols in this category result in nodes with lower buffer occupancy consumption. The downside, however, is gathering invariant properties of a DTN which a node can then exploit to forward its bundles.

In the next section, epidemic routing protocols and their optimizations related to buffer management are first outlined. Then Section 2.1.3 and 2.1.4 survey works that consider DFs and statistical respectively. Lastly, Section 2.1.5 expounds works that exploit mobility patterns.

2.1.2 Epidemic

2.1.2.1 Pure

Vahdat et al. [4] proposed the first epidemic routing protocol for DTNs, which is based on an epidemic algorithm originally developed for updating databases [34]. Each node maintains a summary vector describing each bundle's destination, length and hop count. Whenever two nodes encounter each other, they begin an anti-entropy session, where they compare their bundle summary vector to ascertain missing bundles. Nodes stop their anti-entropy session when they have the same bundle summary vector; i.e., same set of bundles. Each bundle also contains a hop count that corresponds to its priorities, and is also used to constrain flooding. Apart from that, nodes allocate a dedicated buffer space for non-local bundles. As shown in Fig. 2.3, node A and B exchange bundles

Table 2.1 A comparison between DTNs routing protocols categories

	Epidemic	**Data Ferries**	**Statistical**
Forwarding Strategy	Pure or limited Flooding	Reactive/Proactive	Proactive
Node Types	Homogeneous	Heterogeneous	Homogeneous
Mobility	Random	Controlled	Semi-random
Delay	Lowest	Highest	Moderate
Bundle Duplication	Every node encounter	Upon encountering a DF	Only to neighbors that meet a given history criterion
Energy Expenditure	Highest	Lowest	Moderate
Maintain Encounter Information	No	Partially	Yes
Location Information	No	Yes	No
Complexity	Simple	Moderate	Highest
Remarks	(i) Strong theoretical support, (ii) flooding of bundles increases buffer occupancy level and energy expenditure, (iii) buffer management policies trade off delivery ratio and buffer occupancy level.	(i) Data ferries have unlimited storage, (ii) finding a tour that minimizes delay is NP hard, (iii) buffer occupancy is dependent on DFs' tour length	(i) Changes in topological properties affect convergence time, (ii) nodes are required to calculate or record statistics or historical data at every encounter

which are different to those in their buffer. The downside of pure epidemic is that the buffer occupancy of each node increases with each encounter. Hence, it needs an appropriate buffer management policy that frees up buffer space whilst ensuring high bundle delivery ratio.

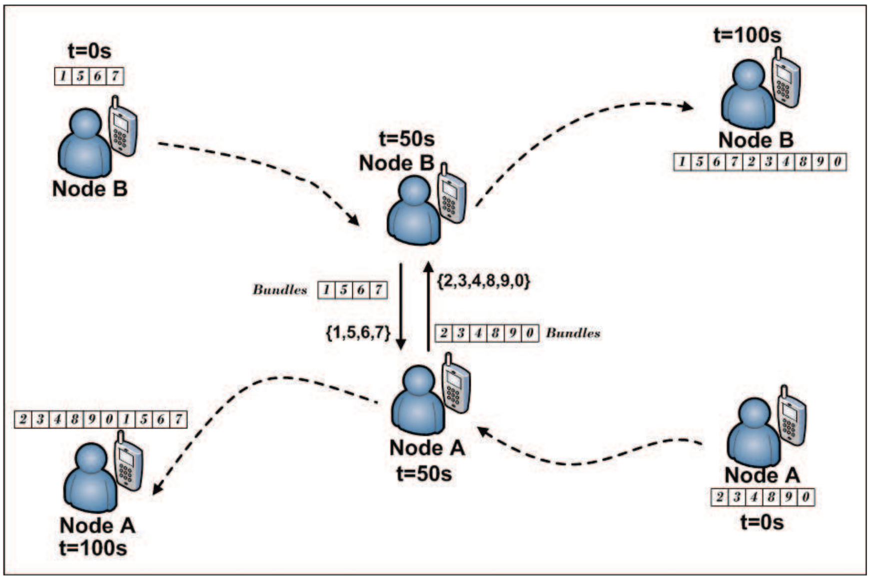

Figure 2.3 Pure epidemic transmission. At t= 0, node A and B carry six and four bundles respectively. At t=50, they encounter each other, and begins by transmitting their respective summary vector: {1, 5, 6, 7}and {2, 3, 4, 8, 9, 0}. After that, both nodes transmit "new" bundles to each other.

In their experiments, the authors compared the delivery rate when nodes are allocated varying buffer sizes, and when bundles have different hop counts. Their results show that using an infinite buffer size results in the fastest message delivery, and achieves 100% bundle delivery with an average time of 147.3 seconds. With a buffer size of 1MB, 500KB and 200KB, they only observe up to 0.4% and 10% degradation in average delivery rate and maximum delivery times respectively. Interestingly, when the buffer size shrinks to 2.5% of its original capacity, the delivery rate reduces to 79.7%, and only 29.3% when nodes have a 10KB buffer size. Their results also show that protocol delivery rate is affected by bundles hop count. When hop count is changed from eight to one, delivery rate drops from 100% to 80%. In general, their results show that buffer size has a non-negligible impact on bundle delivery ratio. Although these results confirm epidemic routing has good performance in DTNs, it suffers from the following problems. Firstly, in scenarios where nodes have a large summary vector or number of bundles, a short contact period may prevent nodes from exchanging their summary vector successfully and complete an anti-entropy session. Secondly, nodes do not preferentially discard bundles during congestion. Thirdly, its performance is limited when nodes have memory constraints.

2.1.2.2 Optimizations

To date, researchers have proposed three ways to reduce the memory consumption of nodes: anti-packets, limited copies, and metrics.

Anti-packets

An anti-packet is generated by a destination node once it has received a bundle successfully. In addition, anti-packets are used to eliminate duplicated bundles. That is, each anti-packet is paired with a bundle. Note, anti-packet is analogous to vaccination [1], [2], immunity [3] or cure [35]. Hence, upon receiving an anti-packet, nodes check their bundle list and delete the corresponding bundle. In Fig. 2.4, nodes first exchange anti-packets. Accordingly, node A determines that bundle 2, 3 and 4 have been delivered to their destination. Therefore, node A deletes them from its buffer and only deliver bundle 8, 9, 10 to node B. The transmission from node B to A follows the same process. Compared to pure epidemic in Fig. 2.3, nodes with immunity tables are able to reduce their buffer occupancy level from 10 bundles to five bundles; see Fig. 2.3 and 2.4. Moreover, the use of anti-packets, which are usually small in size, reduce the number of bundles that are exchanged in each encounter. Example protocols that employ this optimization include epidemic with immunity [3], epidemic with active curing (SERAC) [35] and P-Q epidemic [1].

The (p, q)-epidemic with vaccination routing protocol or PQERPV [1] transmits bundles with varying probability and avoids duplicating bundles. Specifically, nodes store bundles from their encounters with probability p and q, where $p = q = 0$ means nodes do not store any bundles, and $p = q = 1$ indicates nodes are to receive all bundles. Nodes that are neither sources nor destinations are called relay nodes. Moreover, nodes that carry bundles on behalf of

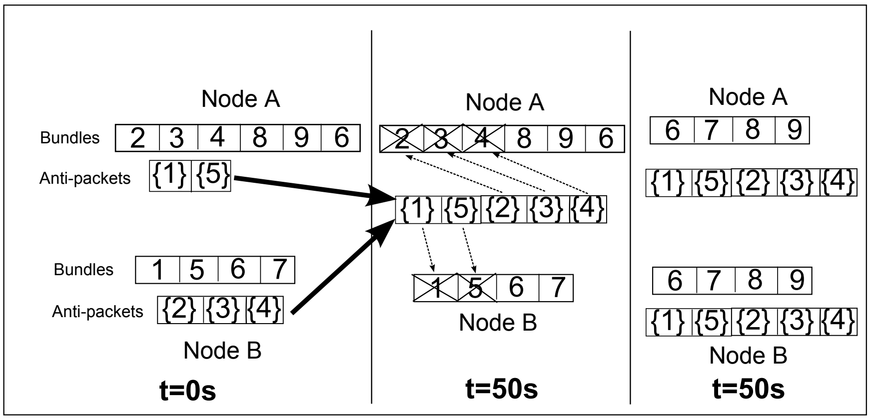

Figure 2.4 Epidemic with anti-packets. Node A has bundles 2, 3, 4, 8, 9, 6 and anti-packets for bundles 1 and 5; shown as filled squares. Node B stores bundle 1, 5, 6, 7 and anti-packets for bundles 2, 3, 4. At t=50, nodes A and B meet each other and exchange their anti-packets. As a result, node A deletes bundles 2, 3, 4 and receives bundles 6 and 7 from node B. Node B deletes bundle 1, 5 and receives bundles 6, 8, 9. After t=50s, both nodes have the same set of anti-packets.

other nodes are termed "infected relay nodes". The value of p and q is set according to node types. Specifically, relay nodes store bundles from sources with probability q and store bundles from infected relay nodes with probability p. Destination nodes initiate a vaccination process once they have received a bundle correctly. Specifically, they flood anti-packets to delete bundles buffered at relay nodes. This protocol has the following limitations. Firstly, the flooding of anti-packets does not consider resource consumption. Secondly, the speed in which source nodes send new bundles is governed by the dissemination of anti-packets. In other words, source nodes will not send new bundles until they have the corresponding anti-packets for bundles sent previously. In an effort to reduce resource consumption, in [36], the authors combine PQERPV with an additional metric called TTL to control and eliminate delivered bundles. Their results show that when TTL is set to 10 minutes, nodes store on average 27% of transmitted bundles as compared to 19% when their TTL value is set to five minutes. This is because nodes are more likely to remove expired bundles.

Another protocol that utilizes anti-packets is Epidemic Routing Protocol with Immunity [3] [37]. This protocol delivers and drops bundles according to two lists: m-list and i-list. m-list is similar to the summary bundle vector in epidemic routing protocol [4], where it records a bundle's ID and destination.

i-list maintains the bundle IDs that have arrived at their respective destination. When two nodes encounter each other, they combine their i-list and exchange bundles that are not in their i-list. Each bundle that successfully arrives at its destination triggers nodes to update their i-list. A key concern, however, is that the size of i-list will increase following the successful delivery of bundles. Moreover, the authors did not specify any i-list management policies.

Scheme for Epidemic Routing with Active Curing (SERAC) [35] proposes a faster anti-packet transmission and efficient buffer management scheme. The main idea, called active curing, is to prioritize the transmission of acknowledgment (ACK) messages, and thereby propagate them quicker throughout the network. Additionally, SERAC recalculates a new route when forwarding ACKs so that they follow the "best" path given the current network state. This, however, consumes more resources in terms of memory and CPU computation. Besides that, to minimize the size of ACKs, SERAC uses two bytes to represent the sequence of bundles that has arrived at a destination. However, when bundles are fragmented or small in size, these two bytes overhead will be significant.

Limited Copies

Spyropoulos et al. [38] [39] propose the Spray and Wait routing protocol. Its routing process can be split into two parts: (i) spray phase, where L copies of a bundle are initially forwarded by a source to N neighbors, and (ii) in the wait phase, these N neighbors relay a copy of the message only when they encounter the destination. The same authors also introduce an improvement called Binary Spray and Wait scheme, in which each node transmits half of their bundles they have to any encountered nodes. For example, a source node with $L = 10$ will transmit five bundles to another node A, and keeps five bundles for itself. This process is then repeated for any nodes that the source and node A meet in the future. Their experiments involving 100 nodes show that with L increasing from five to 20 the delivery delay decreases by approximately 42%. Under the same condition, Binary Spray and Wait has a higher performance, where delivery delay ranges from 3500 to 1500 seconds. The main limitation with this protocol is that a maximum of two hops is used to deliver bundles. Hence, in large DTNs, a bundle may incur a significant delay as it can only be delivered when a relay or source node encounters the destination.

In [40], Bulut et al. consider how bundle copies are distributed to relays given a time constraint. The main approach is to have a number of periods, each with increasing "urgency" corresponding to a bundle's deadline. Initially, a source sends out a small number of copies, and waits for an acknowledgment. If delivery fails, the source *sprays* additional copies to nodes that have yet to receive a copy of the bundle. Hence, with each passing period, more copies are generated to ensure a bundle is delivered. The authors show via analytical and simulation studies that multiple periods reduce the number of bundle copies required to meet a given deadline. Their work, however, assumes acknowledgments are forthcoming in each period to facilitate bundle transmission. In addition, acknowledgments are forwarded using epidemic routing, which incurs high overheads.

Energy is an important issue for DTNs comprising of battery constrained devices; e.g., sensors attached to animals [10]. To this end, Li et al. [41] and Altman et al. [42] study energy efficient forwarding policies for these types of DTNs. That is, limit bundle copies so that nodes incur minimal energy expenditure associated with transmission and reception. In [41], the authors consider a two-hop forwarding model, whereas the latter work also considers probabilistic epidemic forwarding. In both works, the goal is to design policies that improve bundle delivery whilst adhering to a given energy budget or bounded transmission times. The key control parameter is the probability of transmission. For example, in two-hop forwarding, a source forwards a message to another node at time t with probability $p(t)$. Using an extensive analytical framework, Li et al. showed that setting $p(t)$ according to the following policy to be optimal in terms of energy efficiency: given a time threshold t_0, set $p(t) = 1$ if $t \leq t_0$, and $p(t) = 0$ otherwise.

Metrics

In this optimization, nodes compute a metric that reflects the current network state, such as the number of times a node has encountered another node or contact duration, in order to evaluate a neighbor's ability in delivering a bundle successfully.

James et al. [43] developed an epidemic routing protocol that uses a buffer management strategy called Drop-Least-Encountered (DLE). The authors utilize Encountered Count (EC) as a metric to decide which bundles to drop when a node's buffer is full. Specifically, nodes drop bundles with the smallest EC. Apart from that, the value of EC degrades with time. For example, if two nodes meet only once, their EC value will decline from one to zero gradually. Additionally, nodes can also learn the EC metric of a neighboring node's past encounters. Consider node A, B and C. When node A encounters node C after meeting node B, node C can also learn the EC metric of node B and A. However, this protocol still uses flooding to transmit bundles, and it does not consider bundles priority. Consequently, high priority bundles may be dropped when a node's buffer is full.

EC value is also used to delete those bundles which have more duplicated copies than other bundles. For example, Fig. 2.5 shows how bundles with the highest EC value are replaced by newly received bundles. Each bundle has an attached EC value which is stored in the EC table. Once bundles are exchanged, their EC value is increased by one. In the figure, node A transmits bundle 4, 8 and 9 to node B, which results in them having a new EC value of 4, 3 and 7 respectively. Given that node A and B's buffer is only capable of storing five bundles, when node B's buffer is full, bundle 3 and 6 are discarded and replaced by bundle 8 and 9 as they have the highest EC value. Note, undelivered bundles have higher priority even though they have a higher EC value. For example, in Fig. 2.5, because node B has never received bundle 9, node B replaces bundle 6 with bundle 9, which has a higher EC value.

Prioritized Epidemic (PREP) routing protocol [44] gives preference to bundles

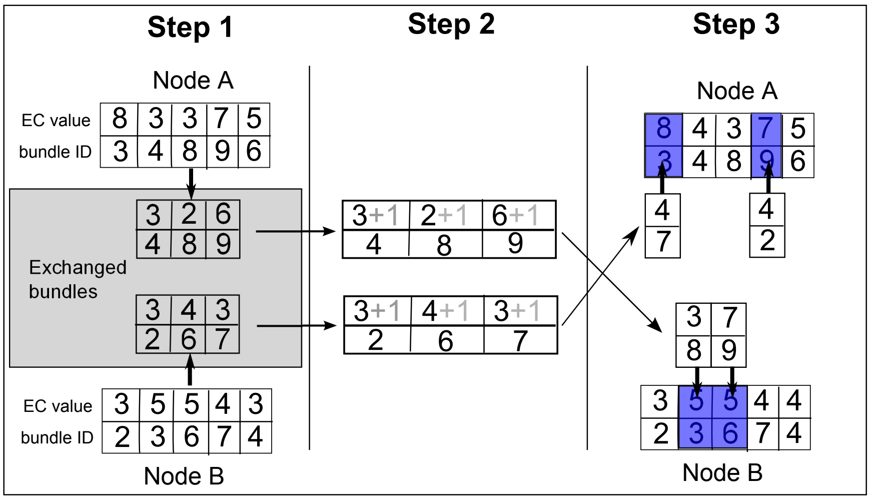

Figure 2.5 Epidemic with EC. Each bundle has an EC value. In Step 1, nodes exchange bundles in a similar manner to pure epidemic; i.e., they conduct an an-entropy session. In Step 2, the EC value of exchanged bundles is increased by one - as reflected by the number next to each bundle. In Step-3, as nodes only have room for five bundles, they replace bundles with a high EC value with newly received bundles. E.g., node A removes bundle 3 and 9 as they have the highest EC value. In their place, node A inserts bundle 7 and 2 respectively. Similarly, node B removes bundle 3 and 6, and inserts bundle 8 and 9 into its buffer.

according to a cost value that is computed according to their respective destination, source and expiry time. In addition, PREP maintains a high replication density when bundles approach their destination. It consists of two main components: topology awareness and bundle drop/transmit processing. The topology awareness component updates a link metric called Average Availability (AA), which is defined as $\frac{T_{up}}{T_i}$, where T_{up} is the total time when the link is up, T_i is the time when the link is available. A bigger AA value means a link has a higher utility and stability. PREP marks bundles that are further away from their destination as low priority. This means nodes maintain a high bundle density as bundles approach their destinations. The main problem with PREP is that in highly dynamic topologies, nodes will announce updated AA frequently, which slows route convergence.

In Epidemic with TTL [36], nodes discard bundles according to bundles' TTL value. Every bundle has the same TTL, and once they are transmitted and stored in a buffer, their TTL begins to reduce by one every second. If a bundle is transmitted to other nodes before its TTL expires, the bundle's TTL value is renewed. As shown in Fig. 2.6, the bundles stored in node A and B are removed after t=50s as both nodes fail to forward these bundles to another node.

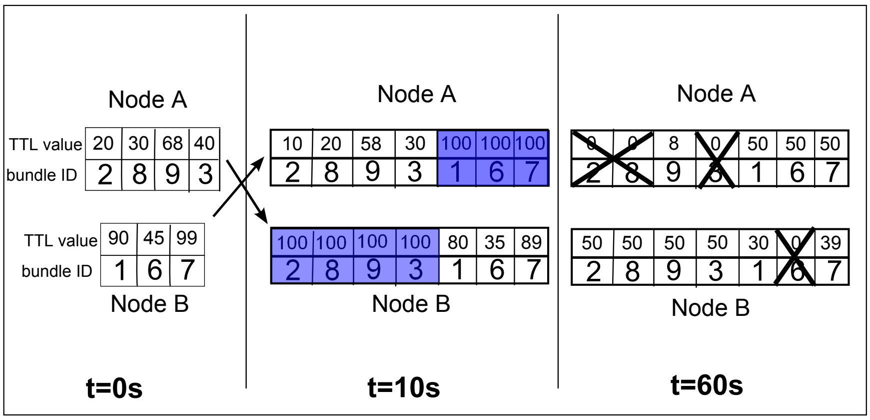

Figure 2.6 Epidemic with TTL. Each bundle has a TTL value. For example, bundles 2, 8, 9, 3 carried by node A has a TTL value of 20, 30, 68, and 40 respectively. At t=10, notice that the TTL value of each bundle has reduced by 10 second. However, after the exchange at t=10s, each bundle's TTL is reset to 100 second. At t=60, bundles 2, 8, 3 expire, and are deleted by node A. At the same time, node B deletes bundle 6 for the same reason.

2.1.2.3 Discussion

Tab. 2.2 summarizes epidemic based routing protocols. We see that since Vahdat et al. [4]'s seminal work, a number of works have been proposed to address many of its limitations. They range from the use of probabilistic metrics to "intelligently" forward bundles to the removal of packets via anti-packets or expiration time. The main limitations of these protocols are that they assume all nodes have the same capabilities or properties, and consume a significant amount of resources. The next section outlines protocols that exploit resource rich nodes that have specific trajectories to help propagate bundles.

2.1.3 Data Ferries

The second category of routing protocols involves the use of DFs. To date, there are 15 protocols that employ DFs. In general, any nodes can be DFs; viz. hand-held devices, vehicles, animals, and satellites. DFs follow a predictable or unpredictable pattern. For example, buses move periodically but with a period that ranges from a few minutes to several hours. On the other hand, wild animals have stochastic movement patterns. Lastly, a DTN can have one to hundreds of DFs [45] [46].

Table 2.2 A summary of epidemic routing protocols.

Protocol	**Techniques to Minimize Flooding**	**Key Feature**
Pure Epidemic [4]	Transmit randomly	Duplicate bundles until they are delivered
PQERPV [1]	Transmit according to probability p and q	Use of anti-packets to erase duplicated bundles
Epidemic with Immunity [3]	Transmit bundles randomly	Use of anti-packets to erase duplicated bundles
SERAC [35]		Prioritized ACK delivery
Epidemic with DLE [43]		Combines buffer management with encounter times
PREP [44]	Relies on the Open Shortest Path Forwarding [25] routing protocol to compute a least cost path	Paths have a cost value, and bundles are forwarded along the path with the least cost
Epidemic with TTL [36]	Transmit according to a probability value	Bundles have an expiration time that is determined by their TTL value
Spray and Wait [38] [39]	Limited copies.	Bounded bundle copies. Two-hops delivery as relays only forward bundles directly to a destination.
Stochastic [41] [42]	Dynamic adjustment of forwarding probabilities	Considers the energy expenditure of bundle transmission and replication.

Section 2.1.3.1 reviews works that investigate the general benefits of DFs. Then, Section 2.1.3.2 presents works that make use of vehicles as DFs. After that, Section 2.1.3.3 surveys the works that use DFs to reduce the energy consumption of nodes. Section 2.1.3.4 outlines the use of DFs in deep space. Lastly, Section 2.1.3.5 reviews works that consider group of nodes, each of which is served by one or more DFs.

2.1.3.1 General

The Message Ferrying (MF) [47] scheme classifies MF movement and routing into two categories: (i) Node-Initiated MF (NIMF), in which DFs move around a deployed area according to known routes and communicate with other nodes they meet, and (ii) Ferry-Initiated MF (FIMF), in which DFs move pro-actively to meet nodes. The authors evaluate the performance of MF according to two metrics: data delivery rate and energy consumption. Their results demonstrate that MF has a higher delivery rate as compared to epidemic routing protocol, and advantageously consume less energy. Moreover, MF transmits 30 times more data per Joule of energy than epidemic routing protocol.

In [45], the authors investigate the relationship between the number of DFs and routing performance. Specifically, they report on the use of a single and multiple DFs. They used the following equation to calculate the delay incurred by a DF, which is nominated as j to collect data from other nodes [47]:

$$D = \frac{\sum_{0<i,j\leq n} w_{ij} d_{ij}}{\sum_{0<i,j\leq n} w_{ij}} \tag{2.1}$$

where n denotes the number of stationary nodes, for each node i, w_{ij} is the data transmission rate between node i and DF j, and d_{ij} is the delivery delay of each transmitted data fragment from node i to DF j. Note that the problem of computing the DF path that minimizes delay is effectively the well known traveling salesman problem. Their approach, however, has only been applied in DTNs with static nodes.

In their subsequent work [46], the authors investigate the use of multiple DFs, and the following routing schemes: single route (SIRA), multiple routes (MURA), relaying between ferries indirectly (NRA), and directly (FRA). The DFs in SIRA travel on the same trajectory, but are on different trajectories in MURA. The difference between NRA and FRA is whether stationary nodes are used as relay nodes between DFs. Their experimental results show that MURA is able to take advantage of routes with the least delay. Also, increasing the number of DFs is also beneficial for reducing the buffer consumption of DFs. Their experiment only considers the scenario where data ferries operate in semi-static topology. However, in high mobility scenarios, data ferries are required to track nodes, and coordinate the delivery process with each other – a task that incurs high signaling overheads.

2.1.3.2 Vehicular Networks

The authors of [48] propose Mobility-Centric Data Dissemination Algorithm for Vehicle Network (MDDV), which they then apply to Vehicle-to-Vehicle (V2V) networks with the following characteristics: (1) predictable and high mobility, (2) dynamic and rapidly changing topology, (3) constrained and mostly one-direction movements, (4) potentially large-scale, (5) frequent disconnections, (6) vehicles that are not completely reliable, and (7) no significant power drain. MDDV exploits the mobility of vehicles to deliver bundles and combines opportunistic and trajectory forwarding as vehicles only encounter each other occasionally, move along streets or roads, and are aware of their geographical location – as provided by the Global Position System (GPS). Nodes using MDDV send bundles to neighboring nodes that are geographically closest to the region that contains the destination. These nodes then flood the bundles upon entering the designated region. This research has two key issues. Firstly, there is no route recovery mechanism. That is, the authors have not considered the case when bundles fail to be delivered due to data ferries running out of patrol. Secondly, the use of flooding is prohibitive when nodes have a high mobility.

Another routing protocol is Meeting-Visit (MV) [49]. Nodes build a geographical information profile whenever they meet peers and visit different geographical locations. MV works as follows. When two nodes meet each other, they will first exchange a list of bundles they are carrying as well as their corresponding destination. In addition, they will annotate these bundles with a delivery probability. After that, the pair of nodes will sort bundles according to their delivery probability. Bundles that are headed in the opposite direction will be allocated the lowest probability. They then select the top n bundles with the highest delivery probability. The delivery probability is calculated as follows:

$$P_n^k(i) = 1 - \prod_{j=1}^{N}(1 - m_{jk}P_{n-1}^j(i)) \tag{2.2}$$

where $P_n^k(i)$ denotes the success probability that node k transmit n bundles, N is the number of nodes, m_{jk} represents the probability that node j and k visit the same location simultaneously.

MV also introduces four kinds of controllers: (i) total bandwidth, which chooses the peer which has not encountered other nodes for the longest period of time, (ii) unique bandwidth, which chooses the peer that has the largest number of messages not present anywhere else in the network, (iii) delivery latency, which chooses the peer whose average delivery time is the largest, (iv) peer latency, which chooses the location least recently visited by a peer. A multi-objective controller then allocates a different priority to the aforementioned controllers. Their experiments show that although nodes experience higher latencies, MV can deliver messages with a high success ratio – i.e., 83% of the maximum achievable delivery rate with minimal duplicated bundles. However, their experiments show that MV has marginal performance as delivery rate does not improve significantly with increasing node density.

Zarafshan-Araki et al. [22] propose TrainNet, a system that uses trains as DFs to transport massive amount of data between stations on a single railway track. A key problem observed is storage optimization, where the storage capacity of hard disks being loaded and unloaded from each station becomes a bottleneck given that trains only stop for a relatively short period of time at each station. They propose four scheduling algorithms to manage the said hard disks: (i) First Come First Serve (FCFS), in which data is loaded onto trains according to their arrival time; (ii) Local Max-Min Fair Algorithm (LMMF), in which the farthest stations are allocated more capacities on train's hard disks; (iii) Global Max-Min Fair (GMMF), in which equal storage capacities are allocated to downstream stations; (iv) Weighted Global Max-Min Fair (WGMMF), in which data is prioritized by weight. Their results show that WGMMF has the best hard disk utilization. Note that WGMMF also has the highest average throughput, which is significantly more than GMMF. This work suffers from the following limitations: (i) intermediate stations cannot be used for storing data, (ii) the protocol does not support train-to-train communication, and (iii) the work only considers trains travelling in a single railway track.

2.1.3.3 Energy

Another key issue in DTNs is the energy constraint of nodes, which has a significant impact on a node's ability to deliver bundles. Zhu et al. [50] present two algorithms, called Least Energy Tree (LET) and Minimum Hop Tree (MHT), to provide energy efficient message ferrying in wireless sensor networks. Both LET and MHT are based on the minimum cost spanning-tree algorithm. These algorithms construct a spanning tree rooted at each node. For LET, each branch of the tree is allocated a weight corresponding to the energy needed to deliver a bundle. On the other hand, all branches are set to one for MHT. Note that, the spanning tree needs to be recomputed if there are topological changes. Their results show that LET consume slightly less energy than MHT. In addition, increasing the number of nodes also has an impact on the energy consumed by mobile nodes.

The authors of [51] consider conserving the energy of DFs. They propose ferry replacement protocols where different nodes take turns to be DFs. The first protocol requires the current ferry to designate a successor that takes over its functionality upon failure. Specifically, it appoints the first node that it meets that has a higher capability than a given threshold. In the second protocol, nodes conduct an election. Each node computes a backoff delay according to their capability. Nodes with a shorter backoff delay will become the next DF. Their results show 15% reduction in overheads when data ferries have a nominated successor. However, in the second protocol, nodes spend more than 1000 seconds electing the next ferry, which delays bundle delivery by at least 2000 seconds. However, the protocol can fail to designate or elect a successor in two scenarios: (1) when all DF candidates have the same backoff delay, and (2) the capability of the current DF falling below a given threshold before it is able to appoint a new DF.

In [52], DFs, also known as MULEs, are used to provide cost-effective connectivity in sparse sensor networks and to reduce the power requirement of sensors. The authors propose a three tier architecture comprising of access points (APs), MULEs and sensor nodes. APs are connected to the Internet, and are used for storing and analyzing data from sensor nodes. MULEs have large storage capacities, renewable power and connect asynchronously to the APs and sensor nodes. In addition, they can communicate with each other. The authors proved that i) a high density of MULEs improves system robustness, and ii) sensor buffer requirements are inversely proportional to the number of MULE nodes. Specifically, when the authors increase the percentage of MULEs from 0.1% to 10%, each with an infinite-buffer, they show a 99.5% reduction in buffer consumption. Nevertheless, to guarantee 100% delivery, their scheme requires duplicated bundles.

2.1.3.4 Space Communications

The advantages of using DFs in deep space networks are as follows [53]. Firstly, DFs cut down the long distance between two planets into relatively shorter ones, which allows each segment of the route to have a high data rate. Secondly, all relays share the energy expenditure of transmitting data on a given route. Lastly, DTNs provide better connectivity as other planetary objects can be used as relays. For example, one can set the moon to be a DF which helps relay messages when Mars is not visible from Earth. In particular, these messages can contain commands that control a spacecraft.

Henceforth, researchers have devised numerous routing protocols for Inter-Planetary Network (IPN). The Interrogation-Based Relay Routing (IBRR) [54] protocol is designed for free space data transmission between Low Earth Orbit (LEO) satellites. When two satellites are in the vicinity of each other, they engage in an interrogation process to exchange orbital information and routing tables. Each satellite will then decide the next best hop according to the following metrics:

- Spatial location and orbital information
- Link bandwidth
- Relative velocity/mobility
- Vicinity of a satellite to other satellites and ground stations
- Memory capacity
- Data rate

IBRR also provides multipath routing. If there are more than one disjoint route to the destination, the source stores all these routes in its cache and selects the best available route. Upon a path failure, it selects an alternate path and routing resumes. However, there are several problems: (a) the authors have not compared the impact of the aforementioned metrics. They assume that if one node has a lower bandwidth but with a higher mobility, nodes will not be able to decide which metric is best, (b) the protocol does not provide a route when a destination is more than one hop away, and (c) each satellite does not consider whether a neighbor has any success in delivering bundles to a given destination.

2.1.3.5 Inter-Cluster Communications

To date, there are several cluster-based routing protocols. In [55], nodes in a cluster employ the Destination Sequenced Distance Vector (DSDV) [26] routing protocol, whereas a single DF is used to provide a communication channel between clusters. Each cluster has multiple gateway nodes that transmit and receive bundles to/from a DF. Nodes in a cluster send their bundles to gateway nodes using one of the following transmission policies: (a) *random* – nodes uniformly pick and transmit their bundles to a gateway, (b) *proportional* – nodes send bundles to the gateway with the highest storage capacity, and (c) *nearest* – nodes choose the gateway with the shortest hop count. Upon meeting a gateway, a DF first transmits any outgoing traffic before accepting incoming traffic or vice-versa. Alternatively, it could transmit bundles in a round-robin manner.

Mobile Relay Protocol (MRP) [56] combines the advantages of traditional and DTN routing protocols. MRP assumes that every node is mobile, and nodes are connected by a traditional routing protocol, i.e., DSDV [26], and also form a virtual cluster. Relay nodes store bundles until they can be transmitted; i.e., they meet a node with a valid route to the destination. If traditional routing fails to find a route, bundles are handed over to the DTN layer. Once relay nodes find a valid route, it reverts back to conventional IP routing. The authors evaluated the delivery rate and latency of MRP using three mobility models: (i) *random* – each node selects its destination randomly within an allowed area, (ii) *soccer player* – a node has a higher probability of picking a nearby destination, and (iii) *homing pigeon* – each node has a "home" location, and its speed is distributed uniformly between 0 and 1 m/s, and nodes choose a random point as the destination with a probability of $\frac{1}{r^2}$, where r is the distance from the "home" to a given destination point, before moving back to its "home" base. In each mobility model, MRP is able to deliver over 95% of the bundles. Nodes experience the best latency when they move according to the homing pigeon model. In addition, they show that MRP is unaffected by node density. There are two problems with MRP. Firstly, transmission between clusters is inefficient because bundles are only handed to their respective destination when a DF arrives at the corresponding cluster. Hence, bundles experience variable delays. Secondly, as MRP relies on DSDV, it is not suitable for sparse DTNs that have no contemporaneous paths between nodes.

In [57], the authors propose a protocol that consists of the following messengers/DFs: regional and independent. A regional messenger only carries bundles for the region it belongs to, while the latter one is used for delivering bundles to all regions. The authors also propose the following strategies: (i) *periodic* – DFs depart regions according to pre-determined schedules, (ii) *on-demand* – DFs leave a region as soon as any nodes require transmission, and (iii) storage-based – DFs leave its region once its buffer is full. Their results show that in terms of average delay, on-demand outperforms storage-based by 50%. On the other hand, the storage-based scheme performs best in terms of transmission cost. That is, in the storage-based scheme, DFs travel one tenth of the trips required by the on-demand scheme. However, the problem is that the delivery strategies of DFs are fixed and are not adaptive to changing network parameters. It postulates that combining all three strategies to be a worthwhile approach. That is, a DF can modify its movement based on memory utilization, transmission requirements from nodes, and allocated paths.

The Context-Aware Routing (CAR) [58] protocol elects nodes as DFs if they have a high number of connections and energy within a cloud or a given geographical area. These DFs are then responsible for ferrying messages between clouds. Their experiments compare CAR's performance with pure flooding and epidemic [4]. Their results show that CAR achieves a 70% delivery rate – assuming nodes move according to the random way-point model. This is 10% higher than pure flooding, but 10% lower than epidemic. However, their experiments only involve one cloud. Therefore, the performance of CAR in multi-clouds scenarios is unclear. Moreover, the authors did not specify any protocols for communications between DFs.

2.1.3.6 Discussion

Tab. 2.3 summarizes the key features of the aforementioned DF-based routing protocols. These protocols can be characterized by (i) the type of nodes that are designated as DFs, (ii) routes taken by DFs, and (iii) the number of DFs. In particular, various DF types are found; e.g., buses and satellites. Apart from that, DFs have a variety of mobility patterns, and have ample resources. Moreover, depending on the application, multiple DFs can be deployed to promote better energy efficiency, delivery rate, and reliability. The fundamental assumption of these protocols is that there exists one or more nodes/DFs with well known properties. However, this assumption is not valid in DTNs where all nodes have pseudo-random movements; e.g., humans.

2.1.4 Statistical

Statistical routing protocols rely on past or historical information such as temporal, spatial and nodes performance. Section 2.1.4.1 outlines work that use temporal information. Then Section 2.1.4.2 discusses work that exploits the spatial location and trajectory of nodes. Lastly, Section 2.1.4.3 presents routing protocols that seek out nodes with good delivery ratios.

2.1.4.1 Temporal

In this category, a time-related function is used to aid forwarding decision. For example, the authors of [59] use the encounter times of nodes. This information is then used by nodes when forwarding bundles, where they preferentially select nodes with the smallest interval between rendezvous periods. In [24], Jain et al. assume the availability of future contact periods. They designed a metric called Minimum Expected Delay (MED) and modified Dijkstra algorithm to compute the path with minimal end-to-end delay. As pointed out by Jones et al. [60], MED is only suitable for certain types of DTNs where future contact times are available, e.g., satellites. To address this limitation, they proposed a new metric called Minimum Estimated Expected Delay (MEED), which is calculated using past contact history. Each node then floods this metric throughout a DTN using a link-state routing protocol, which unfortunately, incurs a significant amount of overheads. Moreover, it is unclear whether such an approach works in sparse DTNs. Another example is Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) [61], which uses a delivery predictability metric P. The metric has three features. Firstly, P is computed iteratively using prior values. Specifically, the P value of node a to b is defined as,

$$P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) \times P_{init} \tag{2.3}$$

Table 2.3 A summary of data ferrying routing protocols.

Protocol	DF Type	Nodes Movement	Ferry type (NIMF/FIMF)	Number of DFs	Target Applications
MF [47]	Designated nodes	Random waypoint	FIMF	1	
MF with single ferrying node [45]		Stationary nodes with deterministic DFs movement			
MF with multiple ferrying nodes [46]				> 1	Generic
MV [49]		Nodes periodically visit known locations	FIMF		Exchange of personal data
MULE [52]	Nodes with large storage capacities and renewable energy	On demand movements between sensors and APs between sensors	FIMF		Tracking, and monitoring
DF and body sensor [20]	Sensor	As per human movement	FIMF	≥ 1	
Energy-efficient MF [50]		As dictated by a minimum spanning tree			
MDDV [48]	Vehicles	According to vehicle routes	Both		V2V
Ferry replacement protocol [51]	Wireless nodes	Nodes move along routes	NIMF	$1^{(1)}$	WSN
IBRR [54]	Satellites	Geosynchronous	FIMF	> 1	Inter-planetary Internet
DSDV [55]	Wireless nodes	Move among groups or clusters		1	
MRP [56]				> 1	
Combined DF scheme [57]					
CAR [58]			Both	$1^{(1)}$	General communications
TrainNet [22]	Train	Between stations according to track layout and timetable	FIMF	1	Data backup and connectivity to rural areas

(1) with load balancing, multiple nodes get to become a DF.

where P_{init} is the initial value, and $P_{(a,b)old}$ is the old value. Secondly, the value of P degrades if nodes do not encounter each other within a given time interval. In other words,

$$P_{(a,b)} = P_{(a,b)_{old}} \times \gamma^{\kappa} \tag{2.4}$$

Here, $\gamma \in [0, 1]$ is a fixed constant, and K is the time units that have elapsed since the last update. Lastly, the metric P can be computed using those from other nodes. That is, if node A and B first meet each other, followed by node B meeting C, then

$$P_{(a,c)} = P_{(a,c)_{old}} + (1 - P_{(a,c)_{old}}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \tag{2.5}$$

where $0 \leq \beta \leq 1$ is set manually to control the impact of transitivity.

In [62], the authors propose PROPHET$^+$, where a "deliverability" value, which is used to determine an appropriate routing path, is based on the following time-varying parameters: buffer size, power, bandwidth, and popularity. Note that, popularity is defined as $1 - \frac{N_t}{M_t}$, where N_t is the transmission rate and M_t is node's capacity in a given time interval t respectively. Their results show PROPHET$^+$ is able to reduce packet loss and delays because it only transmits bundles to nodes that have ample storage and power. However, as pointed out by the authors, PROPHET$^+$ does not consider frequency of encounters.

The authors of [63] showed via experimental studies that cumulative contact characteristics, e.g., probability of node contacts, do not capture transient contact patterns and connectivity. For example, students at a university may meet for a prolonged period during class, and they may have a different set of friends when they are at home. Any routing decision must therefore identify the correct subset of nodes when delivering bundles. Moreover, when students are in class, they may form a connected network. These students are said to have *indirect contacts*, and hence, present an ideal setting for bundle dissemination. To this end, the authors propose a new metric that captures both direct and transient contacts within a fixed time interval. From trace-based simulation studies, their proposed protocols achieved up to 50% improvement in bundle delivery ratio.

In [64], the authors propose a protocol that is analogous to heat transfer. Specifically, the delivery probability from nodes to sinks corresponds to the exchange of heat where nodes with a higher temperature transfer bundles to those with a lower temperature. The temperature of sinks is a constant T, which is set to a much higher value than other nodes. When nodes pass by a sink, they will be "heated", and hence, nodes with higher temperature has a higher probability of receiving bundles from other nodes. The temperature of each node depends on their mobility and the frequency in which they visit sink nodes. When nodes encounter each other, the one with the higher temperature will decrease in value, and vice-versa. This means nodes with a lower temperature will send bundles to those with a higher temperature as they have recently visited a sink. The authors deploy five nodes that move according to the random waypoint mobility model. The results in [64] show that nodes that are geographically closer to the sink have a higher delivery probability. However, the protocol has only been tested in a small network with five nodes.

2.1.4.2 Spatial

The nodes in this category record the speed, direction and mobility pattern of other nodes. For example, Utility-based Distributed routing algorithm with Multi-copies (UDM) [65] prioritizes nodes according to the number of connections a node has to their home communities. Here, "home communities" is defined as locations that nodes passed by and stayed close to most frequently. This means these nodes are more likely to deliver bundles destined to nodes in a given home community. Apart from that, UDM uses binary transmission, where nodes send half of their bundles to another node as long as they have more than one bundle. Their experiments compare UDM with the epidemic and spray-and-wait routing protocol in terms of delivery rate and average delay. Their results show that UDM decreases nodes' average delay by 500 seconds when transmitting 50 bundles, which is half that of spray-and-wait and a third of the average delivery rate achieved by epidemic routing protocol. It is unclear whether similar results can be obtained for a different mobility model.

Both [66] and [67] propose to exploit movement vectors. They assume that nodes have position awareness but do not know each other's movement pattern. The authors of [66] propose the Motion Vector (MoVe) routing protocol, which uses node velocity and angle to calculate the shortest distance to a given destination. Specifically, when they encounter each other, they compare their trajectory, and bundles are forwarded to nodes that are headed to the corresponding destination. Their results, from experiments comprising of 70 nodes with fixed source and destination nodes, show that increasing node numbers improves delivery rate and delay. Their experiments, however, do not consider the impact of duplicated bundles.

Vector Routing (VeRo) [67] uses the trajectory of nodes when forwarding bundles. Specifically, nodes record their position and angle changes, and preferentially exchange a bundle with a node that is moving away from it. The main limitation of VeRo is that stationary nodes tend to receive the most bundles because every other nodes is effectively moving away from them. In a different work, the authors of Similarity Degree-based Mobility Pattern Aware Routing (SD-MPAR) [68] assume that nodes with the same mobility patterns will tend to have similar movement angle and shorter distance between them; i.e., have stable relative positions. When two nodes encounter each other, they compare their similarity degree, which is a function related to the angle and distance between them. Nodes transmit bundles to those with a higher similarity degree. Their results show communication range to have an impact on delivery rate and average delay. Moreover, their results show nodes that have a higher similarity leads to better delivery probability. This, however, is at the expense of additional computational and storage capabilities.

The authors of [69] exploit the mobility patterns of nodes when making routing decisions. For example, a node will prefer to route bundles to nodes that are closer or headed toward the destination. They propose to identify mobility patterns according to four functions. That is, given the Cartesian coordinate of node i and j, (x_i, y_i) and (x_j, y_j), they calculate their (i) Euclidean distance, (ii) Canberra distance, (iii) Cosine angle separation, and (iv) matching distance, where nodes are considered to have a similar distance if $x_j - x_i < \theta, y_j -$

$y_i < \theta$, where θ is a predefined value. Their results show that the Euclidean distance and Cosine angle separation function have the best performance in terms of average delay. As pointed out by the authors, the protocol can be improved further by incorporating temporal factors such as encounter duration or frequency.

Lastly, Nelson et al. [70] propose an Encounter-Based Routing (EBR) protocol. Every node maintains a metric that reflects the average number of contacts within a given time interval. This metric is then used to determine the number of bundle copies that is to be transmitted in each contact. Specifically, a node with a high contact value will receive more copies of a bundle because it has a better chance of propagating a message, which leads to a higher bundle delivery ratio. A key feature of the proposed routing protocol is that it bounds the maximum number of copies of a given bundle to L, where L is either a fixed or probabilistic value. This helps to keep resource consumption low, unlike works such as MaxProp [71] and RAPID [72] that require a high resource consumption in order to achieve comparable delivery ratio.

2.1.4.3 Stochastic

The protocols in this category maintain a time varying network topology that is updated whenever nodes encounter each other. For example, nodes using the Shortest Expected Path Routing (SEPR) [73] protocol maintain a stochastic model of the network. Each node constructs a time varying graph comprising of nodes they have encountered, and links that reflect the connection probability between nodes. This also means nodes will exchange a link probability table containing past encounters whenever they meet other nodes. Applying the Dijkstra algorithm on this graph, each node then calculates the expected path length to a given destination. Their experiments demonstrate that SEPR has an improved delivery gain of 35% with a 50% reduction in resource consumption as compared to epidemic routing and its variants. The key limitation of SEPR is that it has poor scalability due to its reliance on the Dijkstra algorithm. Moreover, it is not suitable for DTNs with high node mobility.

MaxProp [71] improves MV in the following manner. Firstly, MaxProp uses the hop count of bundles to better manage network resources. Secondly, acknowledgments are propagated through the network to erase expired bundles. Lastly, MaxProp deploys a mechanism to eliminate duplicated bundles. Similar to MV, MaxProp utilizes delivery probability during bundle transmission, in which each node initially defines their delivery probability as

$$f = \frac{1}{|s| - 1} \tag{2.6}$$

where f is the initial delivery probability, $|s|$ is the number of nodes in the network. Upon encountering each other, nodes will update their delivery probability. For example, given node A and B that exist in a network with n nodes, where f_A and f_B represent their prior delivery rate, their new delivery rate f'_A and f'_B is calculated as

$$f'_A = f'_B = \frac{f_A + 1}{\sum_{n=1}^{n-2} f + f_A + 1} \tag{2.7}$$

where $\sum_{n=1}^{n-2} f$ represents the sum of other nodes' delivery rate. Each node then maintains a sorted list of nodes delivery rates. Bundles are then sent to nodes with the best delivery rate. Apart from that, MaxProp also employs the following prioritized bundle delivery schemes. First, all bundles destined to neighbors are transferred first, followed by routing information, which includes the probability of meeting every other node. After that, nodes deliver ACKs, followed by bundles that have not traversed far in the network. Finally, nodes transmit the remaining packets and a hop-count list that reflects the list of nodes a bundle has already traversed. MaxProp is evaluated using a real-world testbed called UMassDieselNet. The authors compared MaxProp to three routing algorithms. Namely, (i) random transmission, (ii) Dijkstra algorithm, and (iii) most Encountered/Drop Least Encountered algorithm (ME/DLE) – where bundles are transmitted to nodes that they encounter frequently, and are dropped if a bundle is destined for a node that the receiving node rarely meets. Their results show that bigger buffer size leads to an increased delivery rate. The key limitation of this protocol is that the initial delivery probability is a function of the number of nodes in the network. Unfortunately, without global knowledge, this probability is inaccurate.

Context-aware Adaptive Routing in Delay Tolerant Mobile Sensor Networks (SCAR) [74] calculates the delivery probability of a neighboring node according to the number of (i) encounters it has with sink nodes, (ii) connectivity with other nodes, and (iii) battery capacity. In addition, SCAR is able to replicate a bundle to R nodes with the highest delivery probability. The authors, however, have not evaluated SCAR against any existing DTN routing protocols.

Balasubramanian et al. [72] propose a routing protocol that considers network resources such as bandwidth and storage when optimizing a given route metric. This is especially critical when nodes have resource constraints. The said protocol, called Resource Allocation Protocol for Intentional DTN (RAPID), considers the utility of replicating a bundle at each rendezvous. Here, the utility of a bundle models the benefits of replicating a bundle whilst taking into account resource constraints. They also propose a control channel that allows nodes to collect information such as past transfers and encounters, which are then used to determine the utility $\frac{\delta U_i}{s_i}$ of a bundle i, where the numerator represents the increase in utility if the bundle is replicated, and s_i is the bundle size. Each node only forwards bundles with the highest utility amongst those in its buffer. RAPID's performance is sensitive to how much and frequent information is distributed over the control channel. Their trace-based studies show RAPID is able to deliver 88% of packets with an average delay of 91 minutes. These results, however, are achieved over one set of traces [75], and it is unclear whether similar performance can be achieved over other traces and indeed theoretical models.

2.1.4.4 Social Networks

Another approach taken by researchers to improve the performance of routing protocols is by identifying invariant properties in social networks. Briefly, information diffusion in social networks is a well studied problem and bears resemblance to data transmissions. As shown by Milgram in [76], individuals tend to be separated by six degrees of separation. In his famous experiment, 60 letters destined to a stockbroker were handed to a different person that only knows the stockbroker's name. The study found that the median chain length of intermediate letter holders was approximately six. This experiments demonstrates the "small-world phenomenon", and the ability to propagate information in a seemingly random network.

Henceforth, the authors of [77] propose a routing protocol that exploits the fact that nodes in the real world routinely move and stay in several well known places. This means nodes that visit these places have a contact probability and hence, will have more success delivering bundles. Another example is [78], where Daly et al. propose three relevant properties of information flow: centrality, ties and predictors. Centrality reflects a node's degree of importance, where a node with good centrality has a robust relationship with other nodes in the networks. In particular, centrality can be further measured by three sub-degrees: Freeman's degree, closeness and betweenness. Freeman's degree is the number of directly connected nodes. Closeness refers to the path length or hop count between two nodes. Finally, betweenness corresponds to the number of times a node is used to relay bundles. The ties property evaluates the robustness of the connection between two nodes, where nodes with the following properties are deemed to have a high delivery rate: frequently connect at the same location, long encounter duration, and are well known to other nodes. Lastly, the predictor property indicates whether two nodes are likely to be connected to each other. For example, as proven in [79], based on an analysis of the evolution of scientific collaborations, two co-authors of previous works tend to have a high probability of working together again in the future. In other words, previous connection is a good indicator of future ones. In another work [80], Gao et al. consider user interests during routing. They too exploit centrality, where they seek to exploit relays with high centrality in order to increase user satisfaction. In [81], the authors consider nodes with high centrality as well as their contact duration with others. This is advantageous as popular nodes may only have brief encounters with a high number of nodes, and thereby, have limited communication capacity.

2.1.4.5 Discussion

Tab. 2.4 summarizes the categories and metrics used by proactive routing protocols. To determine the next hop used for delivering bundles, protocols collect information such as time-varying metrics or geographical information. Note that the latter information is easily obtained when nodes are equipped with a GPS unit.

Works that exploit properties of social networks are summarized in Tab. 2.5. We can see all routing schemes utilize the notion of groups or communities where nodes are classified according to their common locations or hobbies. Other than that, researchers also seek out popular nodes, as determined by their connectivity to other nodes. These nodes, therefore, serve as a "good" next-hop when forwarding bundles.

Table 2.4 A summary of proactive routing protocols

Protocol	Category	Metric
TIR [59]	Temporal	Encounter times
PROPHET [61]		Time-varying delivery probability
PROPHET^{+} [62]		
HEAT [64]		
UDM [65]	Spatial	Location visit frequency
EBR [70]		Encounter frequency
MoVe [66]		Movement vector
VeRo [67]		
SD-MPAR [68]		Similarity of mobility pattern
MobySpace [69]		Shortest distance to destination
SEPR [73]	Stochastic	Shortest path calculated by delivery probability
SCAR [82]		Connectivity change with other nodes and remaining battery capacity
RAPID [83]		Network bandwidth and node storage
MaxProp [71]		Delivery probability

2.1.5 Discussion

An important observation is that the works this chapter has reviewed thus far, most if not all, use the RWP mobility model. Tab. 2.6 lists the experiment parameters of all protocols. There is thus a need to experiment with more realistic models. Particularly, this issue is critical given that DTNs are usually based on the movements of both inanimate and animate objects; e.g., vehicles and ani-

Table 2.5 A summary of social networks research

Studies	Definition of a group	Properties/Metrics	Example
[84]	Common location	Connections with other nodes	People that work in the same office or pass each other frequently
[85]			
[86]		Selfishness and altruism	
[87]	Same hobbies and locations	Social profile of encountered nodes	Club members
[88] [89]	Social relationship	Delivery probability and frequency to a community	Family members or classmates or popular individuals/communities
[90]	Locations	Visit frequency	People visiting the same shopping mall
[80]	Mobile users	User satisfcation	Individuals with high centrality; e.g., postmen

mals. Apart from that, all experiments contain a small number of nodes with limited data rate in large areas for simulating sparse nodes density. Additionally, nodes are also allocated limited transmission range and buffer size. Given that all existing works use varying simulation methodologies, it is therefore very difficult to determine the "best" performing routing protocol for a given DTN. Therefore, there is a critical need for a unified research methodology that compares all routing protocols comprehensively.

To date, only a handful of works have proposed alternative mobility models. Bai et al. [30] introduce several mobility models; namely, group, freeway and Manhattan model. The group model is also called the Reference Point Group Mobility Model (RPGM) [91]. In RPGM, nodes move together as a group and every group has a central node called group leader. The group leader determines the movement of group members. In the freeway model, nodes move according to predetermined routes. Lastly, the Manhattan model simulates nodes movement in a metropolitan scenario. Other than that, Leung et al. [92] describe a highway model where nodes enter and leave a highway through multiple entrances and exits. Lastly, the Homing-Pigeon-Based (HoP) model [93] models a scenario where each community has a designated message deliverer that periodically carries bundles from their home community to various destinations before returning home. These works, however, have not comprehensively compared routing protocols designed for DTNs, and thus is an important future work.

Recently, researchers have begun using trace based simulation studies. That is, instead of using theoretical models such as RWP, they use traces of node movements at a given location; e.g., at a conference or city. It is important to note that trace files are specific to a given environment, and cannot be readily generalized to other scenarios. Moreover, they have limited number of nodes; i.e., they cannot be scaled readily to thousands of nodes. Their advantages, however, include the availability of contextual information, and group or community membership. Also, based on nodes' pre-recorded movements, one can easily determine nodes with high *centrality*, and the optimal forwarding path. Indeed, this is the key observation that motivated Hui et al. [89] to develop a forwarding algorithm, called BUBBLE. They identified via trace-files analysis that people based DTNs are characeterized by popular individuals or groups. In effect, individuals and communities have a ranking that denotes their "popularity" in a given DTN. To this end, their algorithm fowards or "bubbles" bundles to increasingly popular nodes or communities.

Past works usually use traces that capture a node's location, encounter times and durations. For instance, in [94], researchers traced the positions of 500 taxis in San Francisco over 30 days using GPS and roadside servers with wireless transmitter. In a different work, the authors of [95] collected the ID of Bluetooth devices carried by students on a university campus. By far, the following are the most popular trace files: (i) Dartmouth/campus [96], (ii) Haggle [97], (iii) MIT/reality [95], (iv) National University of Singapore (NUS) [98], and (v) UMass/diesel [75]. However, thus far, there has been a lack of work that compare the performance of DTN routing protocols using a variety of traces. Lastly, a critical issue is that no works have used a mixed of theoretical and trace based simulation studies to evaluate proposed protocols.

Table 2.6 A summary of experiment parameters

Protocols	Epidemic routing	DFs	Statistical
Number of Nodes	$\leq$100	$\leq$ 120	$\leq$ 240
Mobility pattern	Random waypoint	Random waypoint, along streets	Random waypoint
Area	$\leq$ 50km^2	$\leq$100km^2	$\leq$25km^2
Data rate	$\leq$10KBps		$\leq$250KBps
Transmission	$\leq$300m		
Evaluation metrics	Delivery rate, average delay, time to deliver all bundle	Delivery rate, average delay, KB/J, number of times DFs move from sources to destination nodes	Delivery rate, average delay
Buffer size	Infinite or up to 5MB		
Bundle size	$\leq$14MB		

2.2 Multicast

Applications may need to deliver bundles to a group of users. For example, the dissemination of software patches and targeted advertisements. Supporting multicast in DTNs is non trivial as there are frequent link partitions and nodes experience unpredictable transmission delays. This means it is unlikely that nodes or subscribers will receive bundles at the same time, nor for source(s) to receive all acknowledgments. In addition, group membership changes, varying node speeds and density further add to the complexity of designing multicast protocols that run well in DTNs [99].

Multicast protocols for DTNs can be classified into two categories: (i) unicast-dependent, and (ii) statistical.

2.2.1 Unicast Dependent

In 2005, Zhao et al. [100] define three kinds of multicast receivers according to different semantic models. The first is called Temporal Membership (TM), where nodes that are connected in a given time period are viewed as members belonging to the same group. The second is called Temporal Delivery Model (TD), which combines TM with a delivery threshold for group members. That is, within a given time period, nodes are considered members if they have connectivity to each other and can transmit bundles to their subscribers within a delay threshold. The last one, called Current-Member Delivery Model (CMD), considers the stability of group members – i.e., nodes that remain static for the duration of a multicast session. Zhao et al. also classified multicast routing algorithms into four categories:

1. **Unicast-Based Routing (UBR)** – as in Fig. 2.7, where unicast is used to emulate multicast, whereby a source node sends bundles via unicast to each group member. A source node works as a data ferry, and carry multicast to each group member. Hence, if a DTN has a multicast group with n members, the source node needs to encounter all n subscribers individually in order to deliver a bundle. Note that all data ferries based multicast routing schemes are UBR.

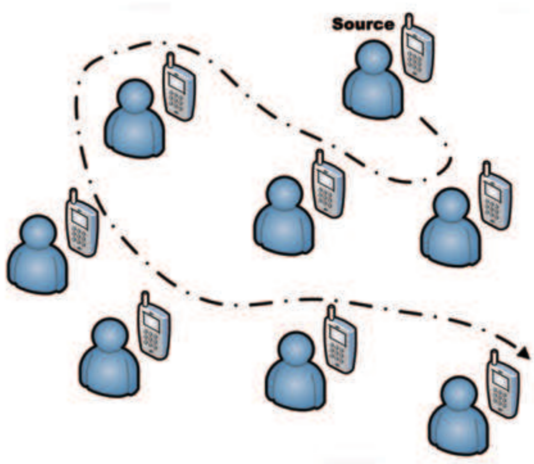

Figure 2.7 Demonstration of UBR.

2. **Static Tree-Based Routing (STBR)** – as in Fig. 2.8, a spanning tree is constructed by a source node, and the tree remains fixed for the duration of the multicast session. This means bundles are delivered along a predetermined spanning tree. Consider a tree where node A, B and C are children of the source or root node, and also, node A, B and C are parent of D to G. During bundle delivery, children only receive bundles from their respective parent. For example, if node B fails to receive the

bundle, both E and F cannot receive the bundle, and will have to wait for node B to recover.

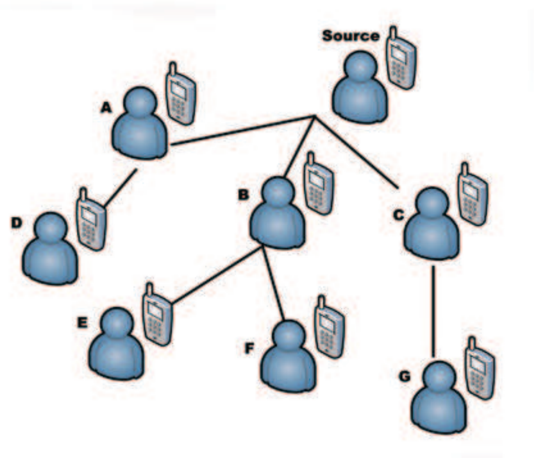

Figure 2.8 Demonstration of STBR.

3. **Group-Based Routing (GBR)** – as in Fig. 2.9, nodes with contemporaneous paths are designated as a group, and a source multicasts bundles to each group using STBR. Within a group, flooding is used to deliver bundles. For example, in a DTN with three groups: 1, 2 and 3, bundles will be disseminated to each group using static paths. Hence, any breaks in these paths lead to delivery failure to the corresponding group. Once a bundle reaches a group, a node is then responsible for flooding the bundle to other group members.

4. **Dynamic Tree-Based Routing (DTBR)** – as in Fig. 2.10, nodes and bundle are allocated a group ID, and bundles are only received by group members with the same group ID. DTBR chooses the shortest route to each group member using Dijkstra's algorithm. This requires the generation of spanning trees that meet different criteria. The parent nodes in the spanning tree generally have higher battery capacity, or lower transmission cost. When a node, say B, runs out of battery, and can no longer work as a parent node, a newly generated spanning tree is constructed,

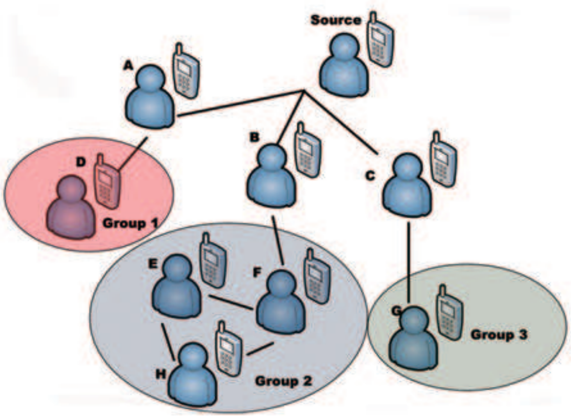

Figure 2.9 Demonstration of GBR.

where node B is pruned from the old spanning tree and becomes a child node.

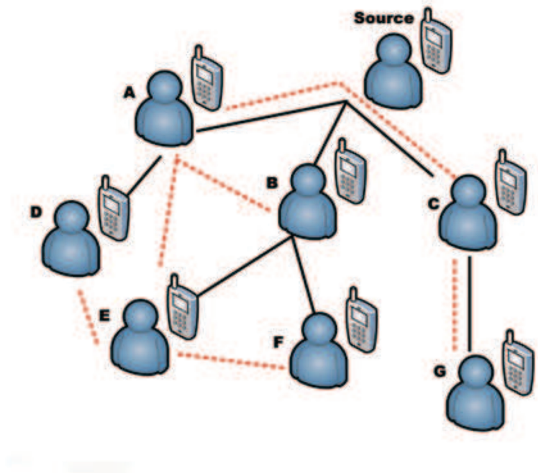

Figure 2.10 Demonstration of DTBR.

All four categories of multicast routing protocols have several shortcomings. UBR requires ferries to meet every subscriber. This means UBR does not adequately utilize the mobility of all non-ferry nodes to help improve the delivery of bundles. For example, in [101], Chen et al. propose a Message Ferrying (MF) routing scheme whereby a single ferry works as a central node to record all encountered nodes and their group membership, and deliver multicast bundles to each group member. Their experiments show that MF has a lower delivery ratio than epidemic routing protocols. Both STBR and GBR have poor flexibility. In STBR, all routing paths are static, and if the path connecting node A and B breaks, both nodes will not be able to communicate. In GBR, bundles are delivered to a group member, which then floods the bundles to other group members. However, GBR still uses STBR to deliver bundles. As DTBR uses Dijkstra's algorithm, it requires a DTN to have contemporaneous paths between nodes.

On-demand Situation-aware Multicast (OS-Multicast) [102] and Context Aware Multicast Routing (CAMR) [103] are examples of DTBR. In OS-Multicast, nodes build a dynamic multicast tree according to their encounter history. That is, nodes maintain encounter records and duration with other nodes, and use this information to calculate the shortest path to a node. In CAMR, the authors propose a multicast routing scheme with a recovery mechanism. Nodes broadcast neighbor discovery messages and if the numbers of neighbors fall below a threshold, nodes set the "sparsely connected" flag and increase their transmission power to recruit more neighbors.

There are also multicast protocols that are designed for DTNs that comprise of nodes with different movement patterns, or capabilities. For example, in [104], Greifenberg et al. propose a multicast protocol that assumes two types of nodes: publishers and subscribers. Nodes designated as publisher store and classify bundles, and flood them to subscribers. Each node maintains a subscription list to record the multicast group they belong to, and floods this list to all other nodes. If nodes receive a bundle destined to a group they have not subscribed to, they will store and forward the bundle to nodes that are subscribed to the corresponding group.

The authors of [101] focus on deploying a proper number of data ferries for multicasting. More specifically, unlike GBR, where each group has a static number of data ferries, the number of data ferries deployed in [101] is dependent on the multicast group size. Moreover, when the multicast group size is small, the protocol uses UBR to deliver bundles, i.e., data ferries deliver bundles to subscribers one by one, otherwise, GBR is utilized where a subscriber floods received bundles transmitted by a data ferry to all group members.

A critical problem for all the aforementioned multicast schemes is the reliance on a spanning tree. However, in a network where topologies change frequently, nodes experience long disconnection times, and may have no acknowledgments from other nodes. Hence, generating and maintaining a spanning tree in such a network becomes impossible. In epidemic routing protocol and its variants, bundles transmissions are based only on node encounters, and does not rely on a spanning tree. Moreover, bundles are propagated at a higher rate, and hence nodes experience a better delivery ratio as all nodes participate in the forwarding process. This is in contrast to data-ferries based multicast routing schemes, where bundles are delivered by designated nodes.

2.2.2 Statistical

In [31], Abdulla et al. argue that the aforementioned approaches are not readily applicable in DTNs due to the lack of knowledge regarding node connectivity and mobility. In other words, it is difficult to form a multicast tree that adapts to the vagaries of DTNs. To this end, they propose Controlled Epidemic Routing for Multicast (CERM). Source nodes multicast bundles to all encountered nodes until they are received by all subscribers. The key problem is duplicated bundles. In this respect, the authors propose two ways to eliminate these bundles. First, they propose the use of synchronization servers, which keep track of TTL value of bundles. Once a bundle expires, these servers flood control messages to all nodes to purge the bundle. The second proposal involves embedding a TTL value in each bundle. Their results show that delivery rate is proportional to multicast group size and simulation time. The group size, however, has no impact on the average delay. This work, however, assumes all nodes have synchronized clocks. In addition, it is unclear how TTL can be adapted to the varying dynamics of a DTN.

The fundamental problem addressed by Gao et al. [105] is to select the minimum number of relays that ensures a given bundle delivery ratio to multicast receivers. Interestingly, this is effectively a variant of the well known knapsack problem, where each item k or relay has a weight w_k. To calculate w_k, the authors employ two key social characteristics, centrality and communities. That is, popular relays or those that are part of the same community as multicast subscribers have a higher weight. Using trace-based simulation, the proposed protocols, Single-Data Multicast (SDM) and Multiple-Data Multicast (MDM), are shown to have 20% and 50% higher bundle delivery ratio respectively than pure epidemic [4] and PROPHET [61], and requires fewer relay nodes.

In the DTN Pub/Sub Protocol (DPSP) [104], there are two types of node: publisher and subscriber. Nodes designated as publisher store and classify bundles, and flood them to subscribers. Each node maintains a subscription list to record the multicast group they belong to, and floods this list to all other nodes. If nodes receive a bundle destined to a group they have not subscribed to, they will store and forward the bundle to nodes that are subscribed to the corresponding group. Their results show that DPSP has a better delivery rate but worse delay than epidemic routing. The limitation of DPSP is that nodes use flooding to announce their subscription. Hence, it is not scalable with increasing node numbers.

In order to support multicast, the authors of [101] propose to use either epidemic routing protocol or DFs depending on group numbers. Source nodes first send bundles to be multicast to DFs, which then decide, according to group size, whether to deliver these bundles one by one or to use the epidemic routing protocol. In terms of delivery rate and delay, their experiments show the proposed protocol reaches a delivery rate of 75%. However, they also showed that the average delay decreases when the group size increases.

2.2.3 Discussion

Tab. 2.7 summarizes the mechanisms and metrics used by each of the aforementioned multicast routing protocols. Additionally, it indicates their delivery rate and delay. Thus far, these protocols assume the ability to build a spanning tree. Hence, there are only applicable to DTNs with semi-static nodes.

Moreover, little to no work has investigated the advantages and disadvantages of using epidemic and DFs routing protocols. Chapter 4 elaborates on these observations further.

2.3 Information Retrieval Systems for DTNs

To date, there are only two works that have systematically studied information retrieval in DTNs. Yang et al. [33] propose a data centric information retrieval system. Each node broadcasts (a) the number of encountered nodes in a given time duration, which they called Friendliness Metric (FM), and (b) a data index, which indicates its stored data. To reduce the duplication rate of queries and replies, the authors predefined a FM threshold. This means nodes that have a "high" FM value can be used to cache queries and replies. Nodes maintain a data index table, which records the node that has a given data. In their system, for a given query, W replicas are made by querying nodes. When a querying node encounters other nodes, it transmits half of its query replicas to its counterpart. Nodes that have queries also transmit half of their stored query replicas until the number of stored replicas is one. They also propose another scheme to avoid flooding queries, in which each query is given a TTL value. The TTL value of a bundle is reduced by one in each exchange. Lastly, the authors used a probabilistic routing protocol to forward replies back to querying nodes. That is, when two nodes encounter each other, the one with the higher delivery probability takes ownership of the reply.

Similar to Yang et al.'s work, Chuah et al. [28] propose another data centric information retrieval system. However, in their system, they propose a special type of nodes to assist in the dissemination of queries and replies. These nodes are called Index and Storage Points (ISPs). They are responsible for storing and collecting stored data indexes from other nodes. Note that, ISPs do not generate their own data, but rather serve to facilitate the exchange of data by providing the address of source nodes. Other than that, Chuah et al. pointed out that there are three types of information: data index, query and response data. They

Table 2.7 A summary of multicast routing protocols

Protocols	**Multicast Mechanism**	**Metrics**	**Limitations**	**Group Definition**
UBR [100]	Calculates a separate routing path from a source to each group members	As per the cost used by unicast protocol	Consume significant resources if multicast group is large	Nodes which require the same bundles
STBR [100]	Constructs a static spanning tree	Bandwidth or transmission rate	Static spanning tree is not suited for dynamic DTNs	
DTBR [100]	Uses Dijkstra algorithm	Transmission cost between nodes in the same group	Bundles cannot be transmitted between groups	Same group ID
GBR [100]	Uses STBR between groups and flooding within groups	Transmission cost between groups	Uses flooding within group, which consumes significant amount of resources	Nodes which require the same bundles
OS-Multicast [106]	Uses DTBR with historical encounters	As per node's destination list	High signaling overheads associated with multicast tree maintenance	
CERM [31]	Employs epidemic routing, and is able to eliminate duplicated bundles	Expiration time	Difficult to synchronize all nodes in DTNs due to frequent disconnections	
DPSP [104]	Relies on flooding and group member filtering	Subscription list	Uses flooding to announce nodes' subscription	Nodes with the same subscription that describes required bundles' categories
CAMR [103]	Uses DTBR with a recovery mechanism	Neighbor counts	Does not work well in sparse networks due to the cost of neighbor search	Nodes in the same location
MEPDF [101]	Combines epidemic routing and DFs, and adaptively chooses suitable protocols according to group size	Group size	Static threshold value for group size	Nodes that require the same bundles
SDM, MDM [105]	Forward bundle to relays with highest centrality and part of a community with multicast subscribers.	Cummulative probability to multicast subscribers	Relays must be in contact with source by the given deadline	Nodes that require the same bundle

also compared the influence of different duplication policies. For example, in Predetermined ISP Advertisement (PISA), nodes only replicate indexes but not queries and data. Specifically, only ISPs store duplicated indexes in PISA. In another scheme, i.e., Opportunistic Regular Node Advertisement with Index Duplication (ORNA-ID), all nodes including ISPs and other non-ISPs nodes are able to store index replicas. The third scheme is called Opportunistic Regular Node Advertisement with Data Duplication (ORNA-DD), in which data are duplicated and stored in any encountered nodes.

Most of probabilistic protocols except for [87] are address centric and not designed for IR systems, which are data centric. This is important because, firstly, address centric systems assume a source node knows the target node's address. However, this assumption is not necessary because the corresponding data may be stored at multiple nodes. Secondly, address centric systems do not support the resolution of complex queries. This is because a complex query needs to be resolved by one or a group of nodes. For example, the query "today's newspaper" can include more than one node that stores different newspapers. In this respect, current IR systems do not provide methods for resolving complex queries. To resolve complex queries, nodes need to estimate the similarity between a query and stored data at encountered nodes. However, in current IR systems, only queries that match stored data will result in source nodes replying back to a querying node. However, for complex queries, when a node has data that partly matches a query, it will be regarded as a source node, and thereby, is obligated to transmit a response. In contrast, in current IR systems, given a node that has data "A" and "B", and if it receives a query "A AND B OR C", it will ignore the query even though its stored data partly matches the query. Apart from that, previous studies [33] showed that, by duplicating data stored at source nodes, queries have a higher probability of being resolved. However, source nodes replicate their data to other nodes before receiving queries can lead to significant buffer occupancy level. Additionally, in current IR systems, as queries are transmitted arbitrarily, nodes may experience significant delays waiting for their query to be resolved.

Boldrini et al. in [87] propose a data centric probabilistic routing protocol - History-based routing protocol for opportunistic networks (HiBOp), in which smart devices worn by every people are regarded as nodes. The nodes record its users' movement and encounters. And then, nodes use HiBOp to calculate forwarding probability for each message according to their stored data. However, HiBOp cannot be applied in IR system. This is because, the data stored by nodes in HiBOp reflects nodes' encounter probability, but not the probability to retrieve a query. Thus, nodes that do not have corresponding data to resolve queries may store the query, and this can increase the retrieval delay and reduce the retrieval efficiency of IR system.

2.4 Conclusion

This chapter has classified and discussed data dissemination from three aspects: unicasting, multicasting and information retrieval protocols respectively. In addition, this chapter has provided a comprehensive qualitative comparison of key features within each category. In particular, epidemic routing protocols constitute a main category of routing protocols because of their simplicity, low delays, and little to no reliance on special nodes. However, to date, there is no work that studies epidemic routing protocols using a common framework to evaluate their performance objectively. Apart from that, high buffer occupancy level and high data duplication rate are key challenges in epidemic routing protocols.

Another key area that is lacking in focus is multicast routing protocols. Indeed, there are ample opportunities to design new ones that incorporate the various strategies used by epidemic based routing protocols. Apart from that, only a few work has studied the efficacy of epidemic routing protocols in delivering messages to multicast group members. In particular, the influence of some critical factors, such as subscribers' group size, anti-entropy session and forwarding policies, has never been evaluated.

Last but not least, IR is an important service in DTNs. However, currently, there are only a few protocols that support IR in DTNs. Other than that, current protocols are address-centric, which cannot resolve all types of queries. Specifically, in address centric systems, queries such as complex and aggregated cannot be resolved because a complex query needs to be resolved by one or a group of nodes. Moreover, as data and queries in current IR systems are duplicated arbitrarily, nodes in current IR systems may have high buffer occupancy level. Also, in scenarios where the required data has a low duplication rate, the resolution may have a long retrieval delay or may result in failure.

Chapter 3

Epidemic Protocols and Their Enhancements

This chapter focuses on epidemic routing protocols. They are ideal for use in a variety of DTNs. For example, those based on the random mobility of humans [14] [107]. This is because they have a simple bundle transmission procedure that only relies on nodes encounters. Moreover, they do not assume the existence of special nodes. In particular, they do not rely on nodes with ample resources or pre-determined movement patterns. These properties are particularly suited for DTNs that use resource constrained sensor nodes; e.g., [108]. Epidemic routing protocols are also critical to one-to-all communication schemes, which can be used to disseminate advertisements or events [109] [110]. More importantly, according to [111], epidemic routing protocols are able to achieve minimum delivery delay, but at the expense of higher resource usage - a key issue addressed in this chapter.

An important issue addressed in this chapter is the lack of research that evaluates epidemic routing protocols using a unified framework. In particular, epidemic routing protocols have been tested in different scenarios in terms of number of nodes, network area, buffer size and bundle size; see Tab. 1.2. As a result, it is very difficult to compare epidemic routing protocols objectively. Besides that, no work has compared the performance of epidemic routing protocols using both RWP model [30] and trace files. This issue remains true for other categories of DTN routing protocols; i.e., DF and statistical. The study of these protocols using a common platform is an immediate future work.

Henceforth, this chapter aims to shed light on the performance of epidemic routing protocols under a unified framework. Specifically, this chapter compares all epidemic routing protocols using a custom-built simulator that moves nodes according to a trace-file and the RWP model. Moreover, this chapter also compares these protocols using the same set of parameters; e.g., node numbers, load and buffer space. From extensive simulation studies, the results identified the following limitations with existing epidemic based routing protocols: high buffer occupancy level, premature discard of bundles, inefficient use of immunity tables to purge redundant bundles, low delivery ratio at high loads, and poor adaptivity to changing network parameters.

This chapter also contains three key enhancements to address the aforementioned limitations. First, set the TTL parameter of bundles dynamically according to a node's encounter interval. The intuition here is that bundles should be buffered according to the interval between two encounters. That is, when nodes experience a long inter-contact interval, bundles will have a larger TTL value, whilst a short interval results in small TTL value. The results show epidemic with dynamic TTL improves delivery ratio by more than 20%. Second, combined EC with TTL to reduce buffer occupancy level and increase bundle delivery ratio. The resulting combination is able to reduce buffer occupancy level by 40%. More importantly, it dramatically improves the delivery ratio by at least 40% at high loads. Third, this chapter improves the use of immunity tables to carry a cumulative acknowledgment. This has the effect of facilitating buffer discard policy, and more importantly, allows a node to delete multiple bundles upon receiving one immunity table. This is an improvement over past studies as nodes need to receive N immunity tables in order to delete N bundles.

The rest of this chapter is organized as follows. Section 3.1 presents shortcomings of current epidemic-based protocols. Section 3.2 describes the enhancements to each epidemic routing protocol. Section 3.3 presents the research methodology and Section 3.4 shows the experiments results. Section 3.4.3 and

3.5 outline the discussion and conclusion respectively.

3.1 Limitations of Existing Epidemic Routing Protocols

In P-Q epidemic, setting P and Q to be less than one may increase transmission delay and decrease delivery ratio. Specifically, in an encounter, assume two nodes can transmit up to 10 bundles. If $P = Q = 0.5$, neither nodes can transmit all 10 bundles. This means they are not taking full advantage of their encounter. This also means nodes are required to encounter each other more often in order to deliver bundles. Unfortunately, in DTNs, nodes are not guaranteed to encounter each other frequently. Every encounter is important, and a missed opportunity will likely result in long delays and low delivery ratio.

Epidemic with fixed TTL values is poorly suited for use in DTNs. This is because setting a large TTL value can result in nodes storing bundles that have arrived at their respective destination. On the other hand, small TTL values lead to bundles being discarded prematurely. This is especially critical if nodes are discarding bundles that have a low duplication rate as doing so leads to transmission failure. The primary problem in epidemic with EC is that nodes experience high buffer occupancy levels and longer delivery delays than other protocols; see Section 3.4. This is because nodes delete their bundles that have the highest EC value from their buffer only when it is full. Moreover, discarding bundles before they are received by the destination reduces bundle delivery ratio.

In epidemic with immunity, a destination node generates an immunity table whenever it receives a bundle successfully. That is, each immunity table identifies one bundle. Hence, the number of immunity tables transmitted is proportional to the load. As a result, they may cause congestion and consequently, result in the discard of bundles. This is particularly detrimental if the discarded bundles have not been forwarded to other nodes.

3.2 Enhancements

This section proposes three enhancements that address the limitations presented in Section 3.1.

- **Epidemic with TTL**. To prevent bundles from being discarded prematurely or buffered unnecessarily due to improper TTL values, the value of TTL is set dynamically; see Algo. 1. More specifically, a bundle's TTL value is set to double the interval time between the last two encounters. This means longer interval results in larger TTL values, and vice-versa. The intuition here is that a longer interval means a DTN is sparse, and hence, bundles should be buffered for a longer period of time to ensure successful delivery.

```
SetDynamicTTL(Bundle b) {;
/* Get the interval of the last contact.  That is, if the last contact was at time t_k, and t is
   the current time, then ttl_time = t − t_k.                                                  */
ttl_time = GetLastInterval(t);
b.TTL = 2.0 × ttl_time ;
}
```

Algorithm 1: Pseudo-code used to set the TTL value of each bundle

- **Epidemic with EC**. A minimum EC value is defined before nodes are allowed to delete a bundle; see Algo. 2. In addition, when the EC value of bundles exceeds a given threshold value, bundles will be given a TTL value. The TTL value will depend on a bundle's EC values. In particular, the TTL of a bundle is proportional to the number of times it has been transmitted. In the experiments, when bundles are transmitted over eight times, bundles will be given a TTL value of 300. For each additional transmission, their TTL value will be reduced by 100 seconds.

```
SetECandTTL(Bundle b, int ECThreshold) {;
/* Store a bundle until its EC value exceeds ECThreshold                                       */
if b.ECvalue≤ECThreshold then
 |   Store(b);
else
 |   b.TTL=300-(b.ECvalue-ECThreshold)×100;
end
}
```

Algorithm 2: Pseudo-code used to set the EC and TTL value of each bundle

- **Epidemic with Immunity**. A cumulative immunity table is introduced; see Fig. 3.1. For example, an immunity table with a bundle ID of 30 means the destination node has received bundles 1 to 30. Note that, destination generates a cumulative immunity table only when it has received one or more bundles successfully. The destination transmits an immunity table for each node that it meets. In terms of buffer policy, a node removes any immunity tables that are redundant. That means, if there are two immunity tables that cover bundles with ID up to 30 and 50, the node will delete immunity table that covers the first 30 bundles.

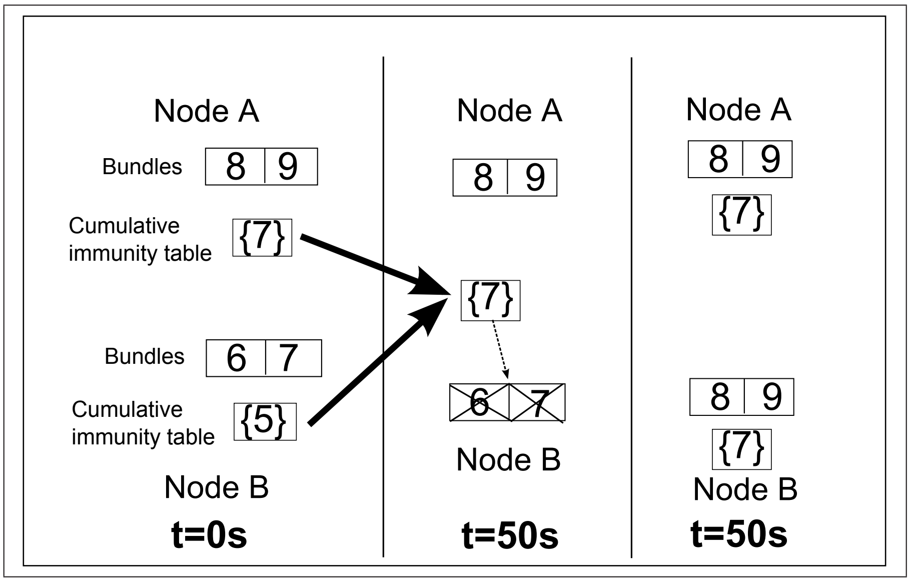

Figure 3.1 Epidemic with cumulative immunity table. Node A has two bundles: 8, 9 and a cumulative immunity table with identity 7. Node B has two bundles: 6, 7, and cumulative immunity table with identity 5. When they encounter at t=50, they compare their cumulative table. As the immunity table of node A has a higher identity, node B updates its immunity table to 7, and deletes all bundles with ID less than or equal to 7. Consequently, node B has no bundles to transmit to node A. Node A then transmits bundles 8, 9 to node B.

3.3 Research Methodology

A customized simulator is used to evaluate all DTN routing. The simulator focuses on the network layer unlike existing simulators such as NS-2 [112]. Apart from that, the developed DTN simulator takes as input a text file containing the movement patterns of nodes. In particular, it works with different trace files that are available for download from CRAWDAD [113]. Also the third simulator is not driven by time but by contact patterns. This allow investigation into routing protocols that may operate over large time scale; i.e., years. Such scenarios are difficult, and consume a lot of processing time, to simulate using existing simulators because nodes may not encounter one another for weeks or rarely. Lastly, the simulator compiles results automatically. Specifically, after each simulation run, it outputs metrics such as delivery ratio of each node, average delivery ratio, buffer occupancy level of each node, average buffer occupancy level via a Graphical User Interface(GUI).

The experiments are based on the data collected by Scott et al. [97]. They collected data over a five day period for the following scenario. Students were asked to carry short range communication devices at the University of Cambridge. In total, there are 12 devices. Each device has a unique ID and records the following information for every node it encounters: begin times, duration and number of encounters.

The data reflects students' rendezvous behaviours. In particular, the trace file shows nodes/students are not always connected, and hence they experience large delays between meetings. For example, nodes/students may connect during a class, and be disconnected when the class finishes. The next class may be eight hours away. Secondly, nodes' movement and encounter duration are random. For instance, two good friends may remain in contact for a lot longer as compared to other students. Thirdly, nodes exhibit a mobility pattern that coincides with meeting times, e.g., lectures, and path to a given classroom.

Briefly, the trace file, see Tab. 3.1, loaded into the simulator specifically records all node movements and encounter history. The file contains each node's trace event. Specifically, each event describes an encountered node's ID, the begin and end time of an encounter, number of encounters and the time interval time from the last to current encounter. Note that, because each encounter involves two peers, there are two trace events for both sides of the encounter. For example, item #1 and #2 in Tab. 3.1 record the same encounter event between Node 3 and 9, which begins at 3568 second and ends at 3882 second. Moreover, Node 3 and 9 have encountered each other six times thus far, and their fifth encounter ended 1350 seconds ago.

At the start of each simulation, the trace file is processed event by event. The transmission of bundles begins and finishes at the start and end each encountered node respectively. Bundles are generally much bigger than messages in conventional networks. For example, bundles in [22] range from several hundreds of Megabytes to Terabytes. Consequently, in the experiments, the transmission time is fixed to 100 seconds. For example, event #1 and #2 record indicate that Node 3 and 9 have an encounter duration of 3882-3568= 314 seconds, and in this duration, Node 3 sends $\lfloor 314/100 \rfloor = 3$ bundles to Node 9.

The said datasets are then used to conducted trace base simulation studies of different DTN routing protocols. Each node is set to hold 10 bundles. The transmission rate, which is the number of bundles that can be transmitted in one time unit, is set to 1 bundle/second. This means the number of bundles exchanged is directly proportional to the rendezvous duration. In each transmission, there is only one source and destination node.

Each protocol is also studied and evaluated when nodes move according to the RWP model [30]. Specifically, 12 nodes moving according to the RWP model within a 600,000 seconds period in the simulations. Nodes randomly choose a destination point, and moves at a speed ranging from 0 to 10 m/s. Nodes may be in contact, whilst on the move or stationary, for a maximum 500 seconds. In both scenarios, once nodes encounter each other, they begin to exchange bundles.

Note that RWP has two fundamental problems [114]. First, any experiments employing RWP may result in nodes having odd movements such as circular or zig-zag patterns. Second, an improper velocity value can lead to all nodes becoming stationary after some period of time. To avoid these problems, a RWP trace-file is generated that ensures nodes move continuously along rendezvous points until the end of the simulation. Specifically, there are less than 100 subscriber points in a one square kilometre area, and nodes encounter and exchange bundles at each point. When nodes reach one subscriber point, they will randomly stop for less than 1000 seconds and move to the next subscriber point, which is also chosen randomly. Note that, the speed that nodes move from one subscriber point to another is dependent on the distance between points, and the interval between contacts, as recorded in the trace file. The speed is then calculated as distance/interval time. In the experiments, the distance between any two subscriber points is less than 1000 meters. For example, event #3 and #4 of Tab. 3.1 indicate that Node 1 encounters Node 3 at one subscriber point before meeting Node 8 at a different point. Given the distance between these two subscriber points is 500 meter, Node 1's speed between these two points is 500 meter / (1311-258)=0.47 m/s. In the experiments, the maximum distance between any two subscriber points is 1,000 meter, and the minimum interval time is 100 seconds, therefore, the velocity of nodes in the experiments ranges from 0 to 10 m/s. Note that, 0 is the speed when nodes encounter and exchange bundles at subscriber points.

In the experiment, a source node is chosen randomly, and transmits k bundles to a destination node. The value of k is increased by five after each experiment, and the maximum number of bundles is set to 50. For each k value, the simulation is run by 10 times and average the results; note, additional simulation runs did not yield any discernible changes in the results. The source and destination node are also changed after each run. Moreover, to avoid collision, the node with the lower ID will send first. After the destination received all bundles, the simulation ends. Also, the maximum recorded time from the trace file is 524,162s. This means if the simulation exceeds this time, the destination node may not have a chance to receive all bundles. In this case, the transmission is marked as failed, and no delays will be recorded.

In experiments where nodes use pure epidemic, they transmit bundles according to their encounter duration as determined by the trace file. More specifically, if two nodes meet each other, the number of bundles that will be transmitted is dependent on nodes' transmission rate and their encounter duration. In P-Q epidemic, recall that a source node sends bundles according to probability P, while other nodes transmit their bundles as per probability Q. The following P

Table 3.1 Trace file format and example of trace items.

	Node ID	Encounter ID	Encounter begin time	Encounter end Time	Number of Encounters	Time interval from the last encounter
#1	3	9	3568	3882	6	1350
#2	9	3	3568	3882	6	1350
#3	1	3	234	258	5	130
#4	1	8	1311	1341	3	50
⋮	⋮	⋮	⋮	⋮	⋮	⋮

and Q values are used in experiments: 0.1, 0.5 and 1. In epidemic with TTL, TTL values are set to 50, 100, 150 and 200 seconds respectively. In all the experiments, the following metrics are recorded:

- **Buffer occupancy level** - the average buffer utilization of all nodes.
- **Bundle duplication rate** - the number of nodes in the network that has a copy of a given bundle over the total number of nodes in the network. For example, a bundle duplication rate of 50% means half of the nodes in the network have a copy of a given bundle.
- **Delivery ratio** - a metric that reflects how many bundles have been delivered successfully to their destination. More specifically, the ratio of received bundles over the total number of bundles sent by the source.
- **Delay** - the time taken for all bundles to arrive at their respective destination.

3.4 Results

The following section presents two studies on epidemic variants. In particular, the studies compare existing epidemic routing protocols using both RWP and trace-file simulation. As pointed out in Section 3.1, such comparison has never been carried out in past studies. From the experiments, the key limitations of each epidemic routing protocol are highlighted, which serve to justify the enhancements proposed in Section 3.2. After that, Section 3.4.3 evaluates the effectiveness of the enhancements in addressing these limitations.

3.4.1 Existing Epidemic-Based Protocols

3.4.1.1 Pure Epidemic

Fig. 3.2 demonstrates the average buffer occupancy level of nodes. With increasing load, the average buffer occupancy level in "Drop-oldest" policy increases from 41% to 100%, whilst in "discard transmitted bundle", the average buffer occupancy level increases from 3% to 10%. Both buffer policies require nodes to consume at least 5% more buffer space. Comparing the two buffer policies, the "discard transmitted bundle" policy consumes the least buffer space, whereas the "Drop-Oldest" policy consumes more than 80% of nodes' buffer when load is more than 10 bundles. This is because discarding a bundle after transmission frees up space for incoming bundles. As a result, nodes observed a bundle occupancy level of 20%. On the other hand, "drop oldest" only deletes bundles when a node's buffer is full.

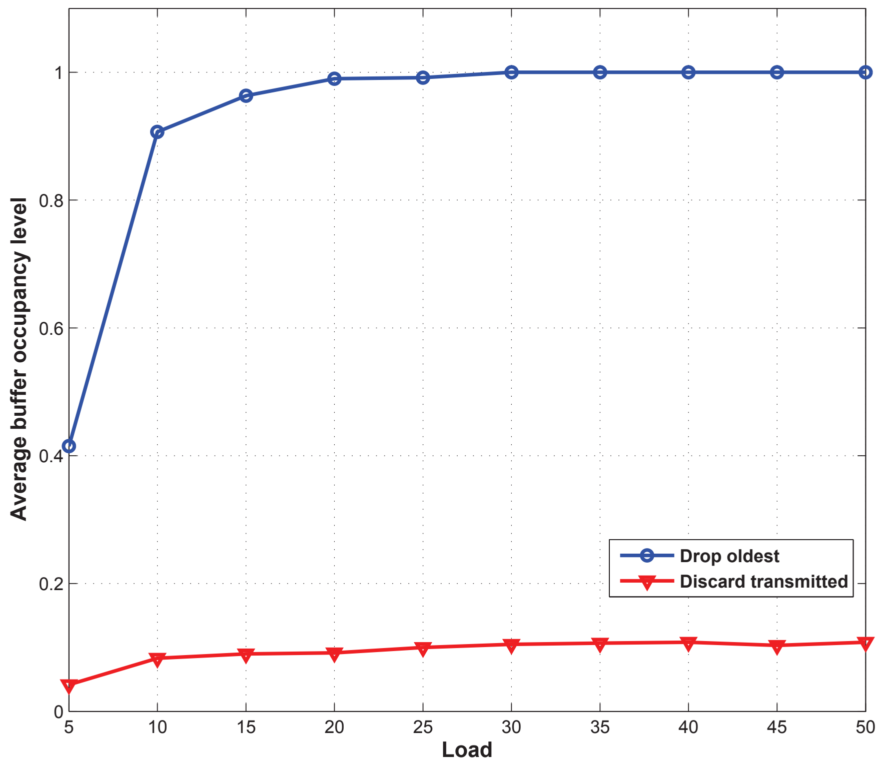

Figure 3.2 Buffer occupancy level with different buffer policies.

Fig. 3.3 shows the average bundle duplication rate, which indicates both buffer policies have similar average bundle duplication rate when a source node increases its load. In "drop-oldest" policy, the average bundle duplication rate is ranging from 81% to 93%, whilst in "discard transmitted" policy, the average bundle duplication rate is ranging from 54% to 61%. Indeed, compared to the "discard transmitted" policy, "drop oldest" policy results in 40% more nodes holding a copy of a bundle. However, with the "discard transmitted" policy, as nodes delete a bundle after a successful transmission, there is only one copy of each bundle in the network. That means, each bundle only exist in one node's buffer. As the probability of two nodes encountering each other obeys a uniform distribution, the probability that a node encounters another node with a bundle is 1/N-1, where N is the number of nodes in the network. However, the same probability for the "drop oldest" policy is n/N-1, where n is the number of bundle copies.

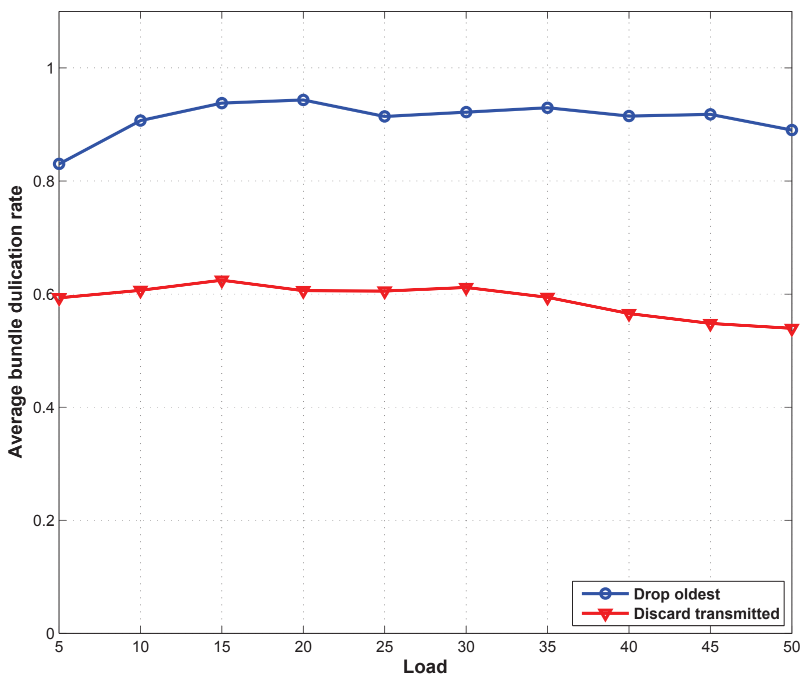

Figure 3.3 Bundle duplicate rate with different buffer policies.

Fig. 3.4 shows that delay follows a step-wise trend. In "drop oldest" policy, the minimum delay is 130s and the maximum delay is 300,000s. In "discard transmitted" policy, the minimum delay is 250,000s and maximum delay is 522,508s. This is due to nodes' mobility characteristic such as encounter frequency and duration, and transmission rate. For example, given two nodes that first encounter each other for 10 seconds, they are able to transmit 5 and 10 bundles in this encounter, and the delay difference between transmitting 5 and 10 bundles is 5s. However, when the load is increased to 15 bundles, they will have to wait for the next rendezvous time. This means, with increasing load, more encounter times are required to transmit all bundles.

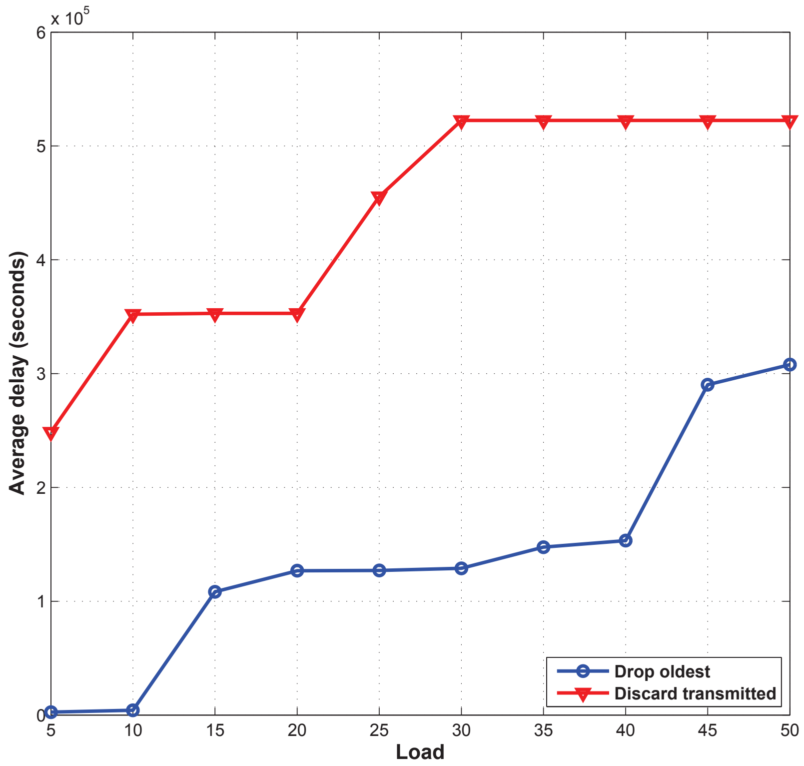

Figure 3.4 Delay with different buffer policies.

The delay in DTN is also influenced by nodes' encounter sequence. For example, in the experiment, when node 5 and 3 is the source and destination respectively, and the load is set to 20 bundles, the delay is 5,804s. However, when the source node is changed to node 3 and retain the load at 20 with the same buffer policy, the transmission fails, and node 5 cannot receive all the bundles. This is because different encounter sequence can result in no bundles being exchanged. For example, assume node A transmits bundles to node B under the "discard transmitted" policy, and node A and B have a separate encounter sequence: node B, E, F, and C, D, A respectively. Consider two transmissions, A to B and from B to A. If a bundle originates from node A and destined for node

B, the bundle will be transmitted in their first encounter. However, in the latter case, the bundle will be sent to node C first, which requires one further encounter before reaching node-A.

The experiment results reveal that delivery ratio is influenced by buffer policy. Fig. 3.5 shows the 'drop oldest' policy has the best performance for load of 10, which achieved 100% delivery rate. The minimum delivery ratio is 92%. Whilst, the "discard transmitted" policy never reaches 100% delivery rate, which is ranging from 73% to 91%.

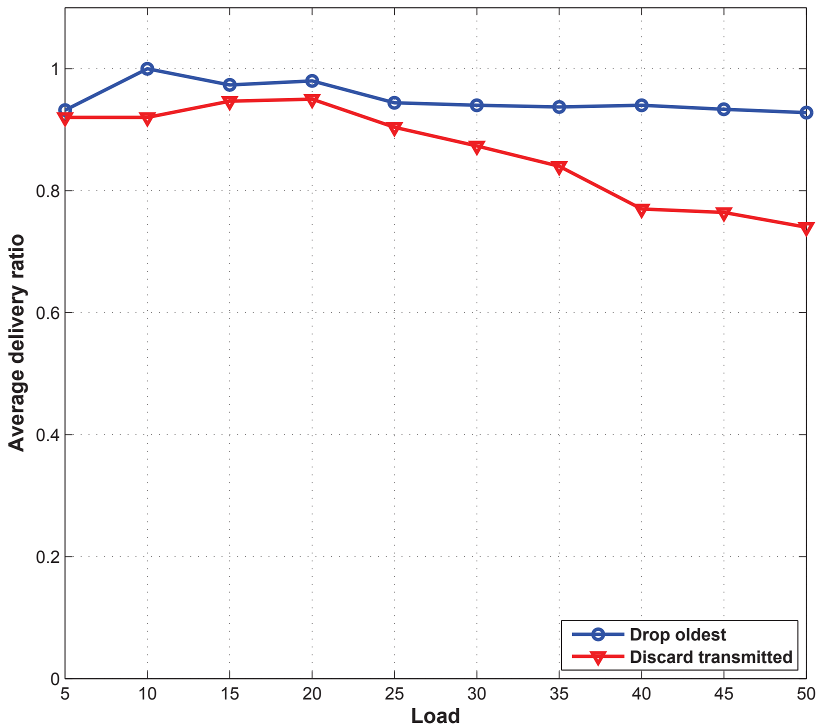

Figure 3.5 Delivery rate with different buffer policies.

Fig. 3.3 and 3.5 confirm that bundle duplication in DTN helps transmission, although the disadvantage of flooding bundles in the network is that each node has a higher buffer occupancy level. Comparing the delivery rate of sending 50 bundles with "drop oldest" and "discard transmitted" policies, the former one delivers 4% more bundles while occupying 5% more of a node's buffer than the latter one. Delivery is increased by multiple bundle copies. This is because higher bundle duplication means nodes are more likely to receive these bundles. On the other hand, if there is no bundle duplication, the bundle can be stored by nodes that do not have many connections with other nodes. Thus, due to the lack of connection, the nodes may fail to transmit the bundles.

3.4.1.2 P-Q Epidemic

Fig. 3.6 demonstrates the impact of P and Q on delivery rate, which is ranging from 9% to 100%. Specifically, when $P > 0$ and $Q < 1$ and the load is more than 35, the nodes fail to transmit all bundles. This is because in a given period, the delivery rate is determined by two aspects: encounter frequency and the number of bundles transmitted in one encounter. When $P > 0$ and $Q < 1$, nodes do not transmit all bundles in their buffer, thereby, to transmit all bundles, nodes need to wait until the next encounter. If nodes' encounter times do not increase, nodes delivery rate will reduce if the number of bundles transmitted in one encounter is not increased. When nodes' buffer overflows, old bundles will be overwritten by newly received bundles, and thereby, reduces nodes' delivery rate. Moreover, the highest delivery rate is reached when $P = Q = 1$. In other words, delivery ratio improves if the number of bundles that are exchanged in each rendezvous increases.

Fig. 3.7 and 3.8 compare the delay experienced by bundles using different P and Q values for trace-based and RWP simulation studies, respectively. In trace-based scenario, when change the value of P and Q, the minimum delay is less than 200s, whilst the maximum delay reaches 522,508s. In RWP scenario, the delay is ranging from 137s to 241,000s. In both studies, when $P = Q = 1$, bundles experience minimum delay. Specifically, in the trace-file based study, when the load is more than 30 bundles and $P > 0$ and $Q < 1$, the delay is at least 50% higher than when $P = Q = 1$. Additionally, when the load is 25 and $P = Q = 0.1$, the delay is 50 times larger than the other four epidemic-based protocols. The delay experienced by a bundle is due to the following factors: interval time between node rendezvous, transmission time between nodes, and nodes' processing time. Indeed, as nodes encounter each

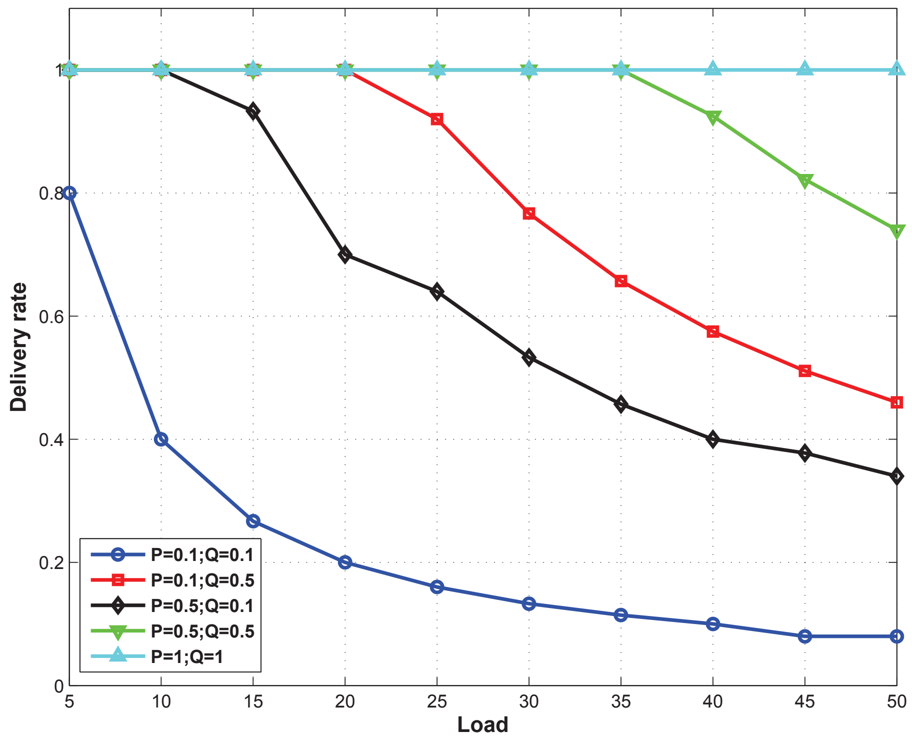

Figure 3.6 Delivery ratio comparison with different P and Q values.

other randomly and do not have a contemporaneous path, the interval duration between each rendezvous is longer than the transmission time. Hence, when $P = Q = 1$, all bundles can be transmitted in one encounter. Whilst, when $P > 0$ and $Q < 1$, nodes do not transmit all their bundles and hence, these bundles have to wait for the next rendezvous time. Note, the curve is not plotted for when $P = Q = 0.1$ in Fig.3.7 as nodes cannot transmit all bundles in any load scenarios.

Fig. 3.9 and 3.10 show the average bundle duplication rate under two experimental studies. In trace-based studies, the average bundle duplication rate is ranging from 19.5% to 100%, whilst in RWP scenario, the average bundle duplication rate is ranging from 41% to 100%. Specifically, the average bundle duplication rate is different depending on the type of experiments. For trace-based simulation studies, the bundle dupliction rate reduces with increasing load, whilst duplication rate goes up under the RWP model. In particular,

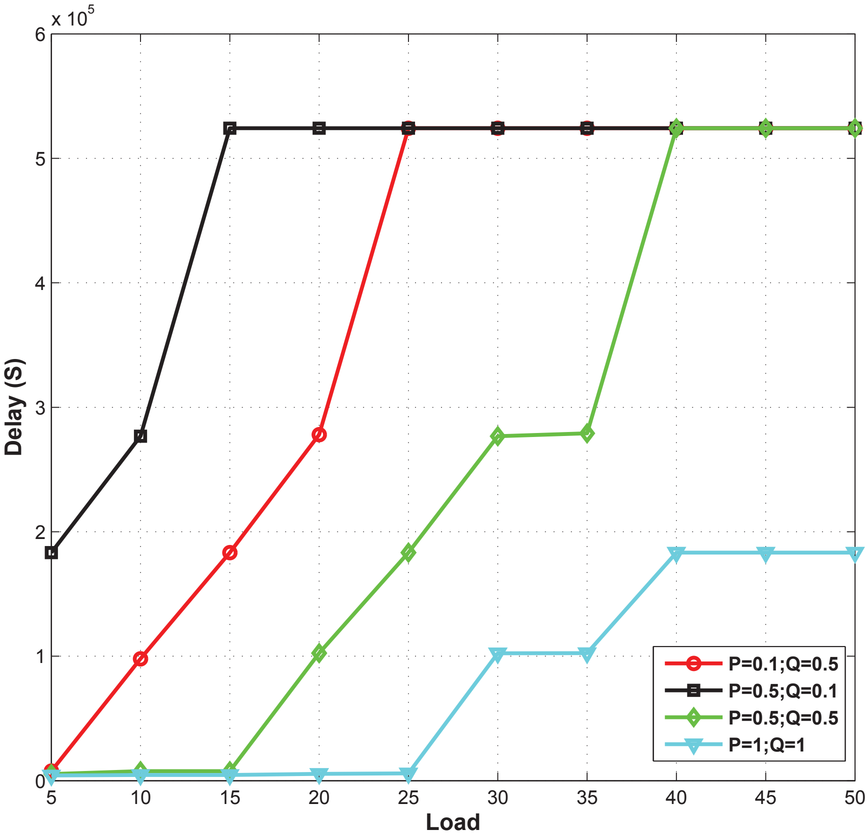

Figure 3.7 Delay comparison with different P and Q values for trace-based.

when $P = Q = 0.1$, and load is more than 10, the bundle duplication rate in the trace-based simulation studies is less than 40%. However, that value soars to over 80% in RWP model. This difference is due to different encounter times and duration. Specifically, increased encounter times and duration lead to higher bundle duplication rate. In the RWP studies, nodes have a maximum 500 seconds encounter duration, and more than 50 encounters, whilst in the trace-based studies, nodes have less encounter times and duration. The delivery probability of a bundle is proportional to the frequency of node encounters and bundle dispatch rate. In trace-based experiments, reducing P and Q values also causes bundles duplication rate to decrease. This is because when $P < 1$ and $Q < 1$, not all bundles in nodes' buffers are transmitted and stored. This means, as there are fewer bundles in each exchange, bundle duplication rate

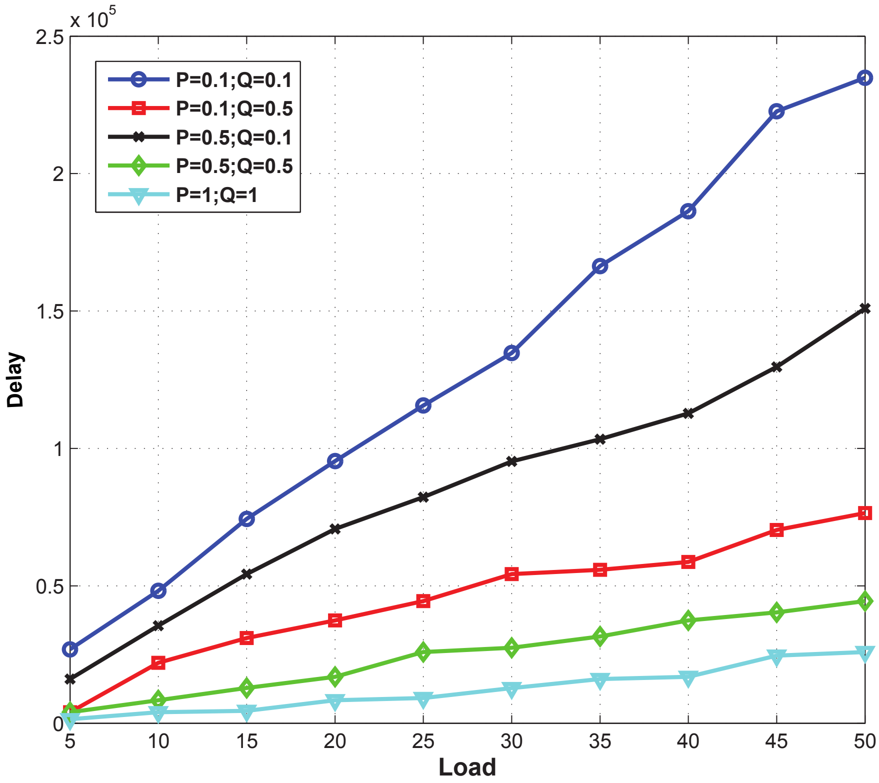

Figure 3.8 Delay comparison with different P and Q values for RWP studies.

decreases accordingly.

As shown in Fig. 3.11 and 3.12, the average buffer occupancy level is different in trace-based and RWP studies. The average buffer occupancy level in trace-based scenario has maximum and minimum value as 25% and 100% respectively, whilst in RWP scenario, the maximum and minimum average buffer occupancy level are 20% and 100%. For example, in the trace-based simulation studies, when nodes transmit more than 10 bundles, the average buffer occupancy level exceeds 90% when $P = 1$ and $Q = 1$. However, the P and Q values have no significant influence on buffer occupancy level in the RWP model when the load is over 15. The primary reason is because in the RWP model, nodes have more encounter times than nodes in the trace file, and hence, more transmitted bundles. Hence, the service rate will be less than the incoming rate, and thereby, any P and Q values lead to increase nodes' buffer occupancy. Note that in trace-

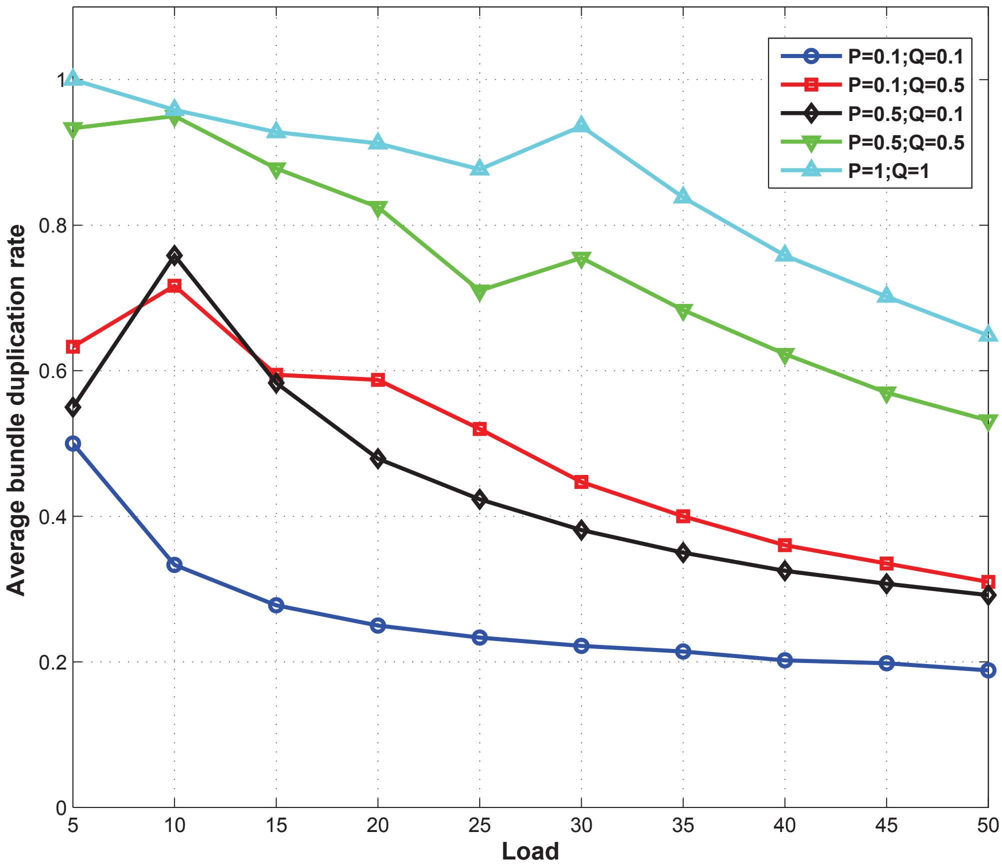

Figure 3.9 Average bundle duplication with different P and Q values for trace-based simulation.

based simulation when $P = 0.1$ and $Q = 0.1$, nodes' buffer occupancy level remains stable, which is less than 1. This can be explained as nodes' service rate is equal to incoming rate.

3.4.1.3 Epidemic with TTL

This section now focuses on epidemic with TTL. Fig. 3.13 and 3.14 show the bundle duplication rate with different TTL values for both studies. In trace-based simulation studies, the average bundle duplication rate is ranging from 28% to 89%. When the load goes up, bundle duplication rate reduces, and the duplication rate are different with each change in TTL. Specifically, the duplication rate is the lowest when TTL is set to 100 seconds. In other words,

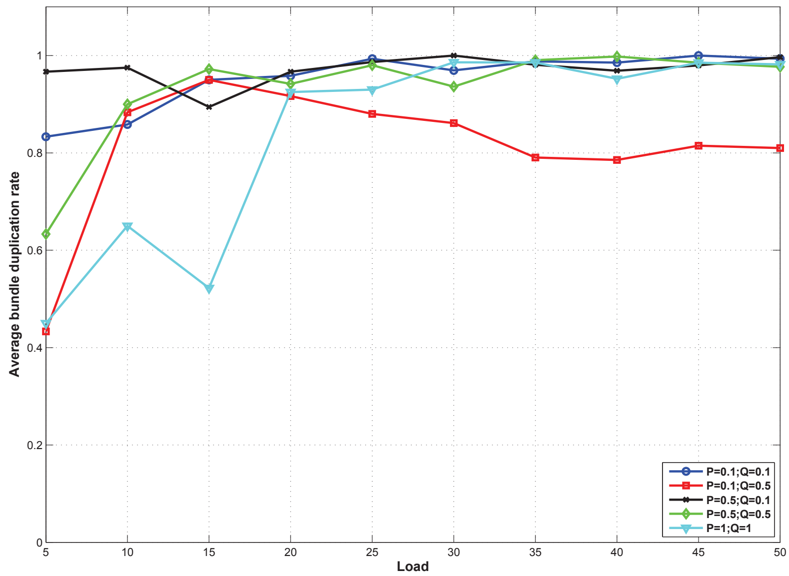

Figure 3.10 Average bundle duplication rate comparison of different P and Q values for RWP studies.

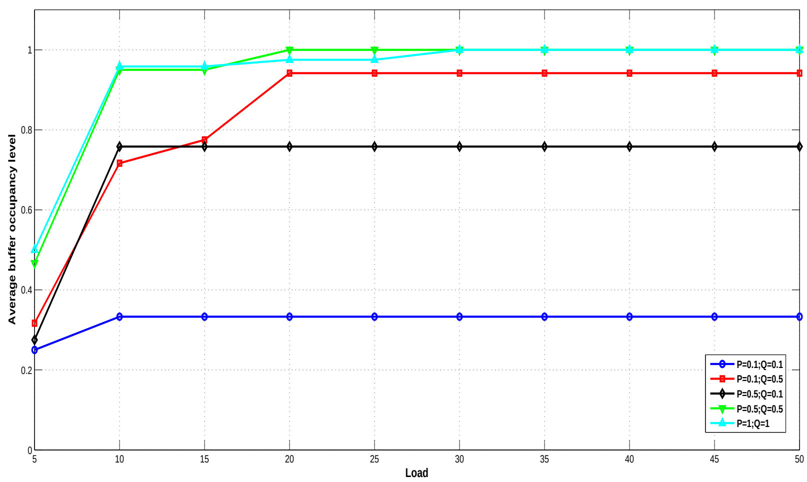

Figure 3.11 Buffer occupancy level of P-Q epidemic with different P and Q values for trace-based simulation.

a smaller TTL value means the bundle has a shorter storage time in nodes' buffer. As the number of stored bundles decreases, the probability that nodes will receive a bundle also reduces. As for RWP studies, the maximum and minimum average bundle duplication rate are 95% to 35% respectively. The bundle duplication rate is no longer decreasing. For example, when load is less

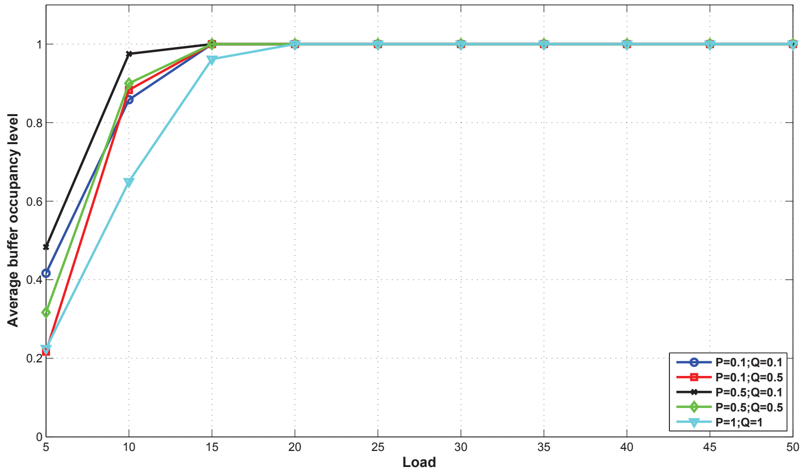

Figure 3.12 Average buffer occupancy level of different P and Q values in RWP.

than 20 and the TTL value is set to 200, the average bundle duplication rate increases. This is because nodes encounter each other frequently, and thereby, renewing bundles' TTL value accordingly. The net effect of this is that there are more bundle transmissions, which lead to higher bundle duplication rate.

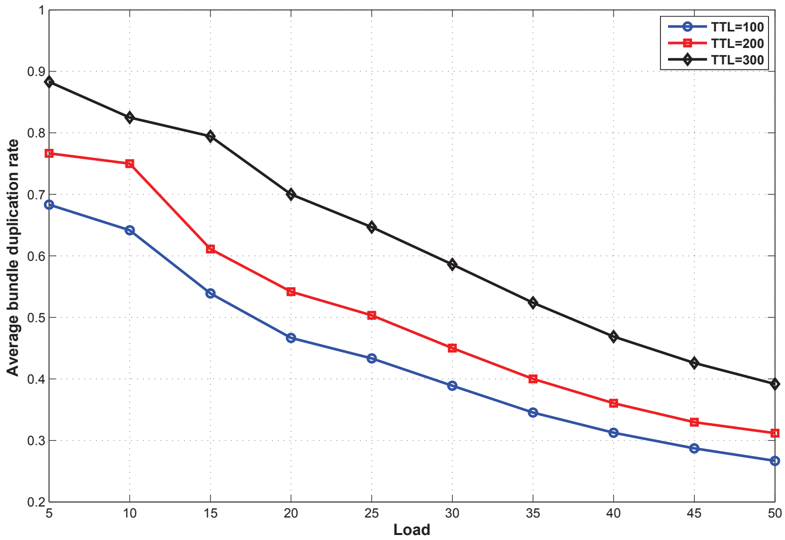

Figure 3.13 Average bundle duplication rate with different TTL values in trace-based simulation studies.

Fig. 3.15 shows the delays experienced by bundles with different TTL values.

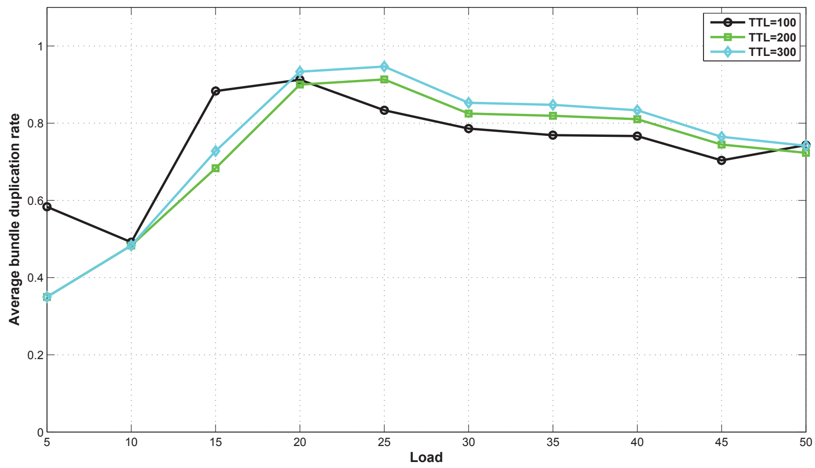

Figure 3.14 Average bundle duplication rate with different TTL values for RWP studies.

We see that for TTL values ranging from 0 to 300, the delays are the same when the load is at 5, 15 and 25. In other words, different TTL values can also lead to the same delay because the interval between transmissions is longer than all TTL values. As bundles with expired TTL are deleted, a node may discard bundles before it has a chance to encounter another node. This means if a bundle's TTL is shorter than nodes' encounter interval, the delivery rate will be low. Moreover, if the TTL remains fixed for different networks, then delivery rate will vary widely. For example, students on a campus may meet each other once per day, while buses will have encounter each other several times in an hour. This means bundles that are transmitted between students need longer storage time as compared to those in a DTN comprising of buses. In other words, TTL should be adaptive according to nodes encounter frequency; i.e., when nodes rendezvous frequently, TTL has a small value, and vice versa. Note that, the results are similar if nodes move according to the RWP model, see Fig. 3.16.

In RWP experiments, all bundles are received by their respective destination. Therefore, Fig. 3.17 only compares the delivery rate for trace-based simulation. Fig. 3.17 shows that delivery rate is ranging from 59% to 100%, and can take advantage of increasing TTL value. Specifically, when TTL increases from 100

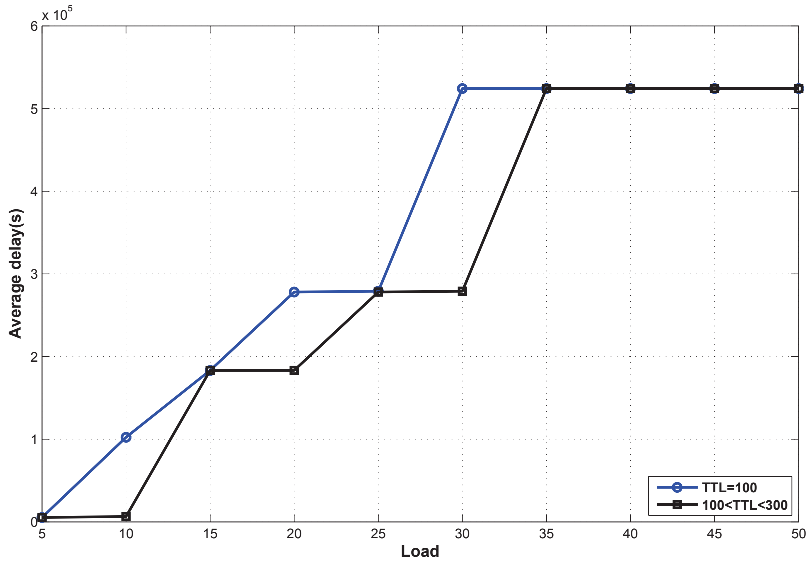

Figure 3.15 Delay with different TTL ranges in trace-based simulation study.

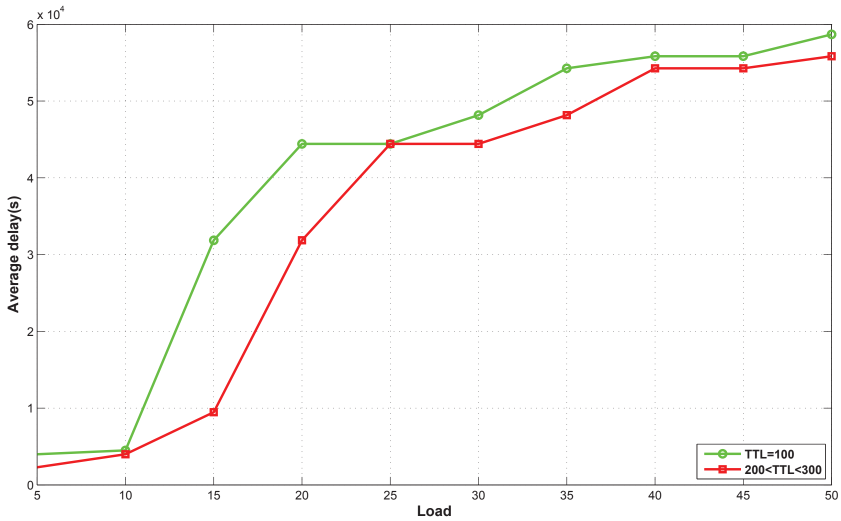

Figure 3.16 Delay comparison with different TTL values in RWP.

to 200, the average delivery rate improves to a maximum of 9%. That is because compared to small TTL values, nodes store bundles for a longer period of time with increasing TTL value, and thereby, improves bundles delivery probability.

The experiments also compare the buffer occupancy level when TTL takes on

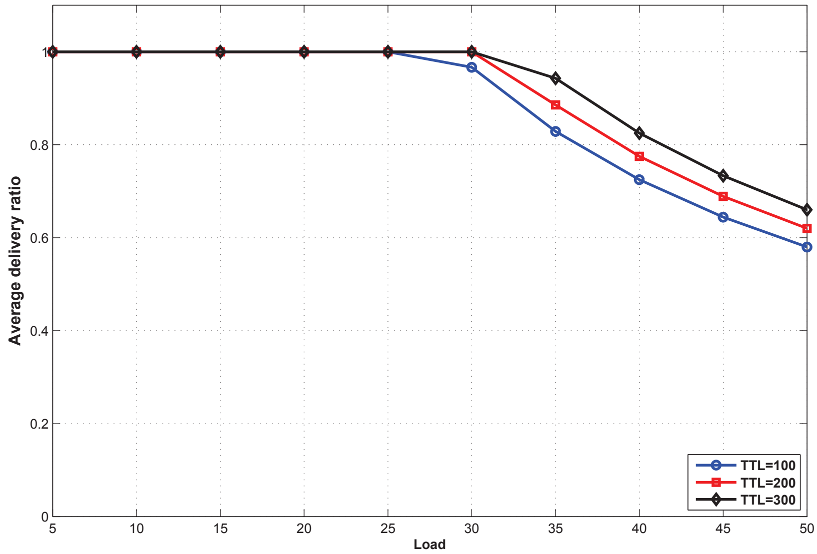

Figure 3.17 Delivery rate with different TTL values in trace-based study.

different values under the RWP model. Fig. 3.18 shows the buffer occupancy level is ranging from 8.7% to 24.5%, and no discernible impact when the number of bundles is between 25 and 45. This is because an improper TTL value has no influence on nodes' buffer occupancy level, especially since every node has a maximum 500 seconds encounter interval, meaning any TTL values that are less than 500 seconds will delete nodes' stored bundles. Moreover, the buffer occupancy level is influenced by the load. At low loads, nodes are able to transmit all bundles before their respective TTL value expires, keeping the buffer occupancy level low. Longer TTL values mean nodes store bundles for a longer period of time, and thereby they experience a higher buffer occupancy level.

3.4.1.4 Epidemic with EC

Fig. 3.19 compares the delays experienced by bundles when nodes use epidemic with EC for both trace-based and RWP studies. The delay is ranging from 247s to 6,000s in RWP scenario, and is ranging from 314s to 522,508s in trace-based scenario. Bundles experience higher delays with increasing load because nodes need more encounter times to transmit all bundles. However, in trace-based

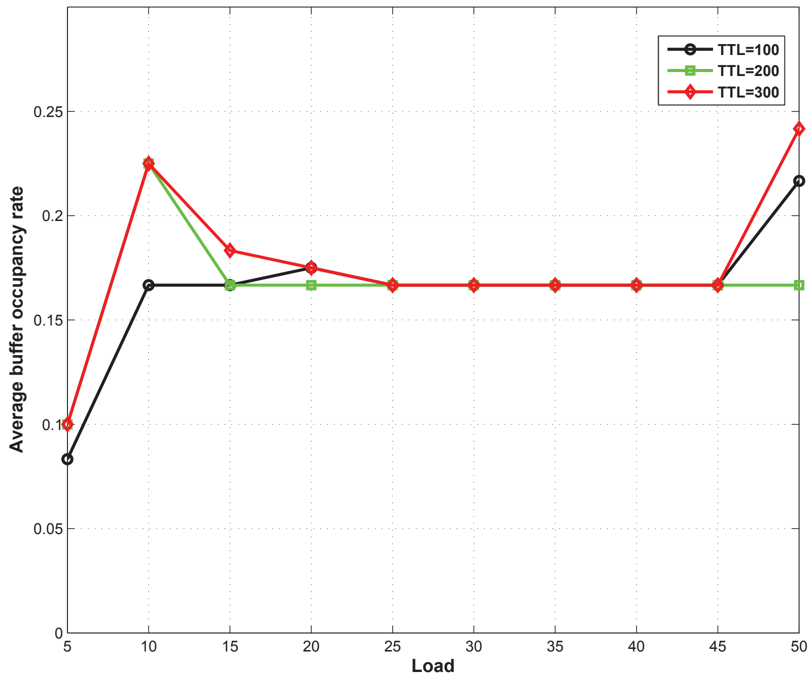

Figure 3.18 Average buffer occupancy level with different TTL values in RWP.

studies, the delay is much higher than that in the RWP model. This is because both studies use a different mobility scenario and thus have distinct encounter times and duration. In particular, the trace file records students' movement in campus, which may be discontinuous as students may move outside the campus. On the other hand, in the RWP model, nodes move continuously within a given area. This means there are always 12 nodes moving about in a given time period. As a result, nodes moving according to the RWP model have a higher probability of encountering each other, which results in bundles experiencing less delay.

In Fig. 3.20, epidemic with EC is effective in reducing the bundle duplication rate when the load exceeds 15. This is because nodes delete bundles with high EC values. Specifically, the bundle duplication rate for trace file experiments decreased from over 60% to less than 30% when the load ranges from five to 50. When load increases from five to 15, the bundle duplication rate experiences a dramatic increase in the RWP model. More specifically, the buffer occupancy

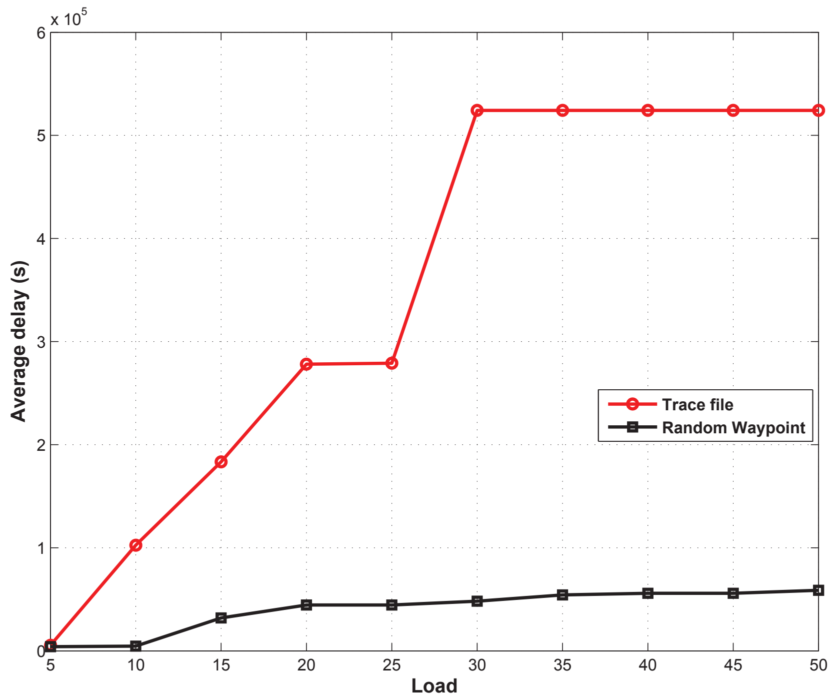

Figure 3.19 Delay comparison of trace-based and RWP studies.

level rises significantly from 6% to 80%. There are two reasons: first, as the load is low, nodes do not invoke any buffer policy; i.e., they are able to store all received bundles. Second, as nodes encounter each other frequently, bundles are transmitted to more nodes as compared to nodes that move according to the trace file.

3.4.1.5 Epidemic with Immunity

Fig. 3.21 shows that epidemic with immunity has a different trend in terms of average bundle duplication rate. This is because the bundles duplication rate is determined by the time in which the immunity table is dispatched. If all bundles are received by their respective destination in a short time and the corresponding immunity table is not dispatched to all nodes during that time, bundles duplication rate will increase. In the RWP model, nodes encounter each other more than 50 times, which far exceeds that of the trace file. Hence, bundles propagate a lot quicker in the RWP model. Moreover, the number of

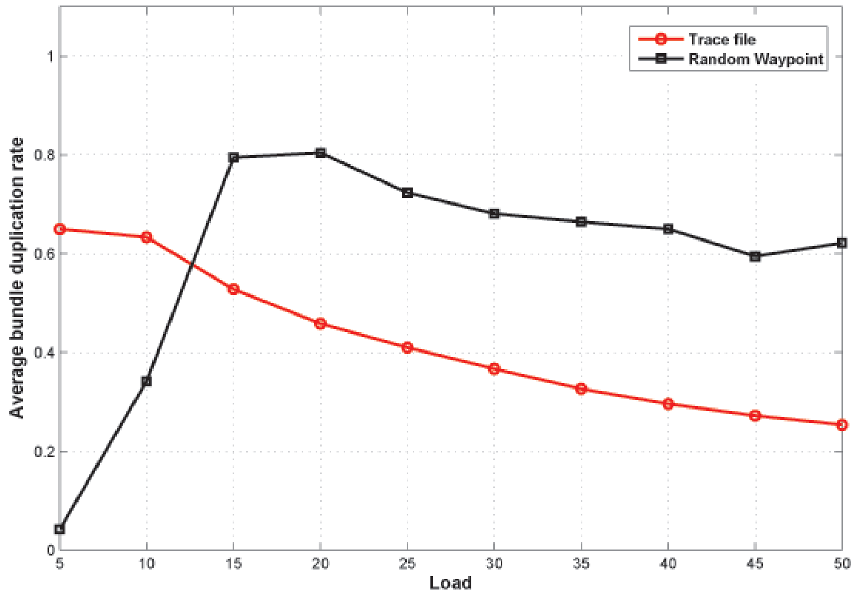

Figure 3.20 Average bundle duplication rate comparison of trace-based and RWP studies.

immunity tables dispatched is proportional to the number of bundle arrivals. For example, if there are N bundles, the destination will transmit N immunity tables. However, this requires all nodes to receive N immunity tables in order to purge all N bundles from the network. Another reason for the high buffer occupancy level is because a destination node or nodes with immunity tables experience a significant delay before encountering other nodes.

Fig. 3.22 and 3.23 show the difference between delay and buffer occupancy level in the two movement scenarios. First, as the load increases, so does the delay in both scenarios. In RWP scenario, the delay is ranging from 732s to 25,000s, whilst in trace-based scenario, the delay is ranging from 5,000s to 522,508s. Specifically, the delay experienced by nodes moving according to the trace file is much greater than the RWP model. In fact, it is at least five times greater when load is more than 30. Second, the buffer occupancy levels also increase when the load rises in both scenarios. The buffer occupancy level is ranging from 16% to 83%. Note that the immunity table maintains buffer occupancy level below 90% for all load values.

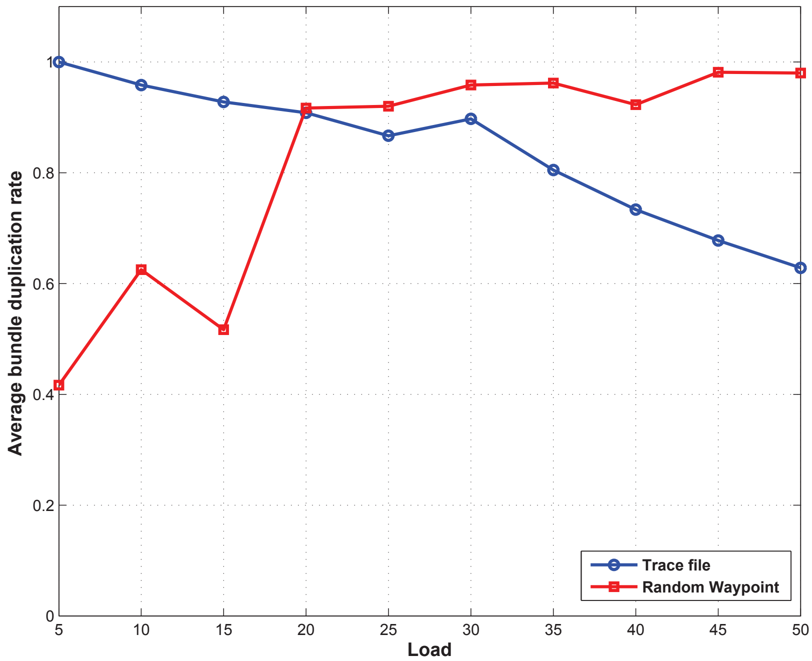

Figure 3.21 Average bundle duplication comparison between trace-based and RWP studies.

3.4.1.6 Comparison of Existing Epidemic-Based Protocols

The delay experienced by epidemic routing protocols are reported in Fig. 3.24 and 3.25. The experiments use the parameters that results in the best delay for all protocols. For P-Q epidemic, the values of P and Q are set to one, and for epidemic with TTL, the TTL is set to 300 seconds. Note that, because P-Q epidemic and epidemic with immunity have the same delay in trace-based experiments when P=Q=1, Fig. 3.24 only plots the delay curve of P-Q epidemic. With increasing load, the delay of epidemic with EC grows the quickest, and P-Q epidemic has the slowest growth. The reason is because epidemic with EC is able to delete bundles before they are received by their destination, which leads to higher delays. Furthermore, Fig. 3.25 shows that, epidemic with TTL has a higher delay than epidemic with immunity. This is because the value of TTL is fixed, whilst epidemic with immunity discards bundles as soon as nodes receive an immunity table. A key observation is that nodes frequently delete

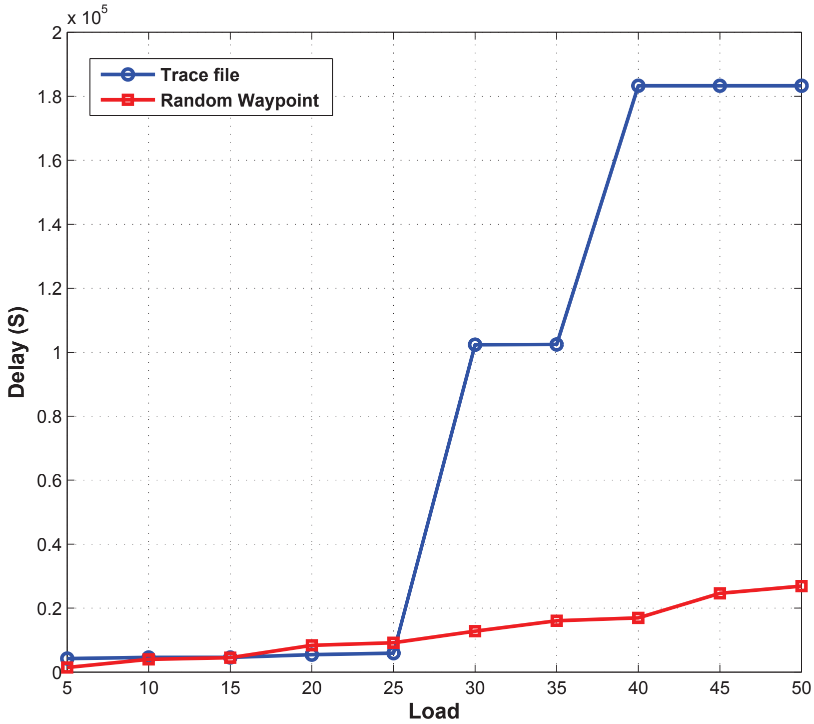

Figure 3.22 Delay comparison between trace-based and RWP studies.

bundles as the TTL value of bundles is shorter than their encounter interval. As a result, the network will have fewer duplicated bundles, and consequently, have a low bundle delivery ratio because destination nodes are less likely to meet nodes with the required bundles.

Fig. 3.26 and 3.27 show epidemic with EC has a lower bundle duplication rate than other protocols. However, epidemic with EC fares better as it discards bundles frequently, which unfortunately lowers bundle delivery ratio. Moreover, epidemic with immunity has a high bundle duplication rate, which reaches over 60%. This is due to the following reasons. First, dissemination of immunity tables relies on the frequency of nodes encounters, and the number of immunity tables, which is equal to the number of bundles or load. The results show that immunity tables are propagated slowly. As a result, nodes that have not received any immunity tables will continue to transmit the corresponding bundles. Second, when nodes free up their buffer, they are able to store more

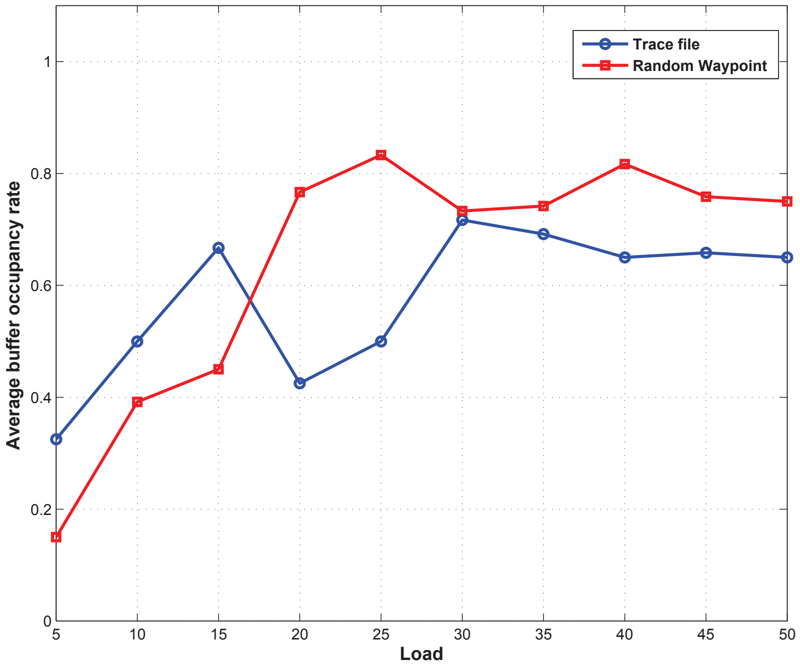

Figure 3.23 Average buffer occupancy level in trace-based and RWP studies.

undelivered bundles and exchange them when they encounter each other. Consequently, freeing up buffer space also plays a role in increasing duplication rate. Apart from that, P-Q epidemic also has a high bundle duplication rate. This is because when $P = Q = 1$, P-Q epidemic is similar to pure epidemic. In other words, the bundle duplication rate is proportional to nodes' encounter frequency.

From Fig. 3.28 and 3.29, P-Q epidemic consumes more than 80% of nodes' buffer when the load is higher than 10 in both trace-based and RWP studies. This is because after bundles are received by the destination, the protocol does not have any mechanism to purge these bundles. Other protocols, however, have such a mechanism. For example, epidemic with immunity table discards transmitted bundles according to its immunity table, which results in over-20% decrease in buffer occupancy level. Furthermore, because epidemic with TTL discards bundles before they are received by their respective destination, its low buffer occupancy level has a negative effect on transmission. Note, when the

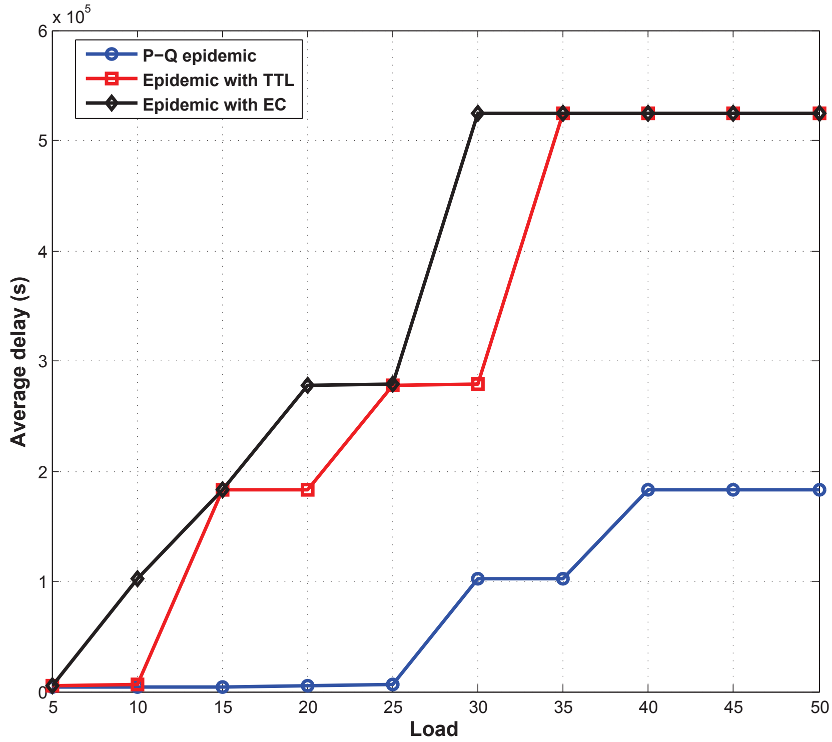

Figure 3.24 Delay comparison of epidemic-based protocols when nodes move according to the trace file.

load is more than 20, the buffer occupancy level of epidemic with immunity experience sudden drops and rises in both trace file and RWP model. For example, when load is increased from 25 to 30 in RWP scenario, and the load is increased from 15 to 20 in trace file, nodes' buffer occupancy level decreases. This is because epidemic with immunity only discards bundles after they reached their respective destination. Therefore, the buffer occupancy level of nodes is dependent on immunity tables stored in each node.

The trace-based experiments only compare the delivery ratio of epidemic with EC and TTL because other protocols have a 100% delivery ratio. As shown in Fig. 3.30, when the load increases, the delivery ratio of all protocols reduces accordingly. The TTL is not suitable for use in DTNs because of the following reason. Nodes experience large encounter intervals, much more than the delays experienced by nodes in conventional networks. In practice, each network will have specific encounter characteristics, where nodes rendezvous interval may

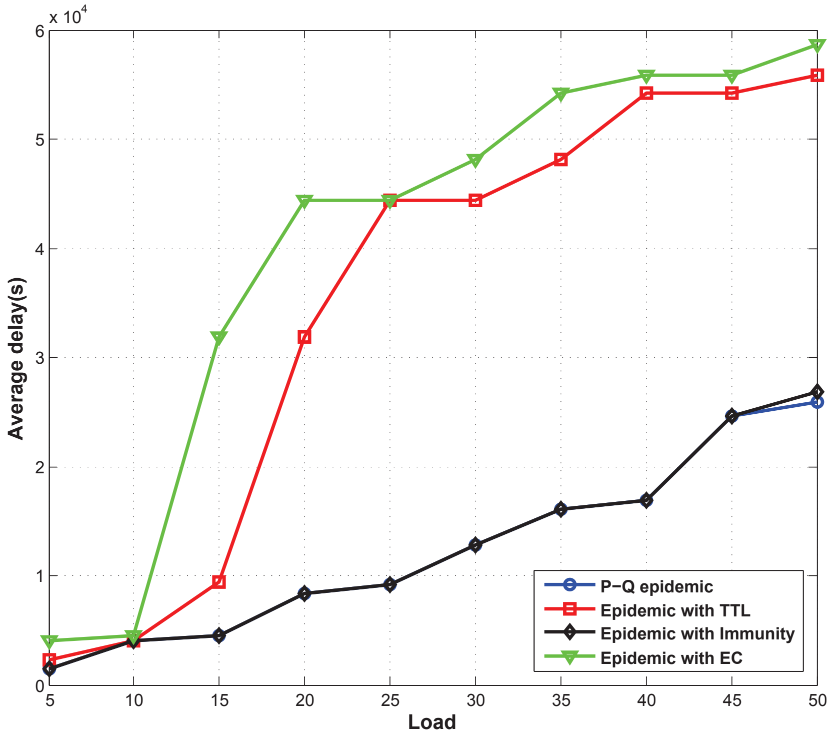

Figure 3.25 Delay comparison of epidemic-based protocols under RWP.

range from a few seconds to days. Consequently, delays may be unbounded, and hence, it is challenging to select a TTL value that can be used to safely discard bundles.

The experiments yield the following findings: first, a high duplication rate leads to short delays. For example, epidemic with immunity has a shorter delay and higher bundle duplication rate than other protocols, whilst epidemic with EC has a longer delay and a lower bundle duplication rate as compared to other protocols. This is because as more nodes have the required bundles, more bundles will be delivered to their respective destination. Second, deleting bundles that are enroute to their destination may result in the deletion of bundles that have a low duplication rate, which unfortunately leads to increased delay or low delivery ratio. Ideally, if the bundles have arrived safely, then any duplicates of the bundles can be deleted without sacrificing delay or delivery ratio. This helps reduce buffer occupancy level, and enable nodes to store bundles that

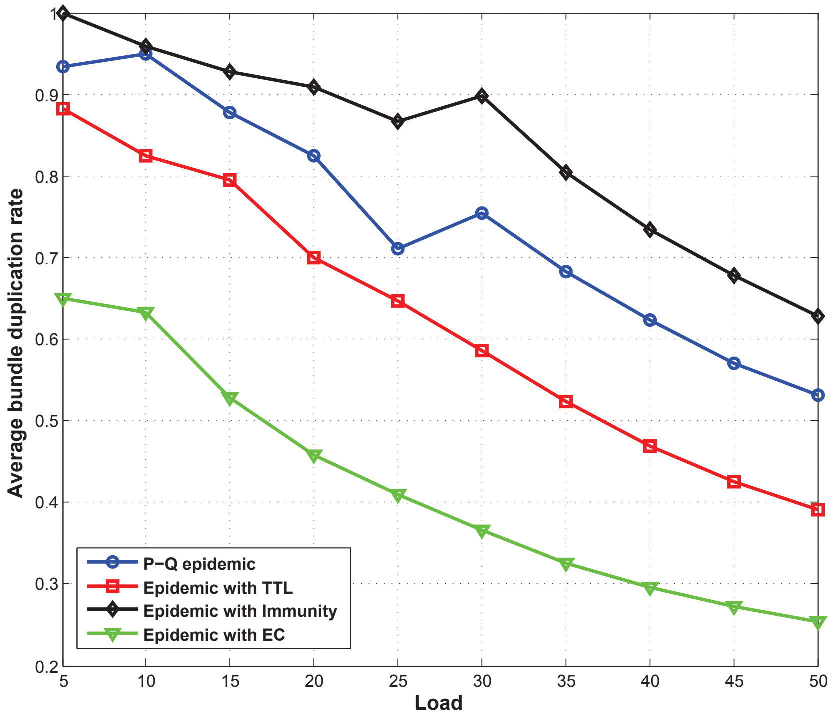

Figure 3.26 Average bundle duplication rate of epidemic-based protocols when nodes move according to the trace file.

have not reached their destination. Unfortunately, the propagation of feedback generated by a destination is also governed by the contact characteristics of nodes.

3.4.1.7 Influence of Nodes density

This section studies the impact of node density on bundle transmission. In the previous section, there are 12 nodes in the RWP model, which correspond to the number of nodes in the trace file. In this section, the number of nodes are varied from 10 to 50, and use the same parameters as the earlier experiments. Note, in these experiments, to distinguish P-Q epidemic and flooding, P and Q values are set to 0.5. Moreover, TTL is set to 300. All experiment results are an average of 10 simulation runs.

Buffer duplication rate

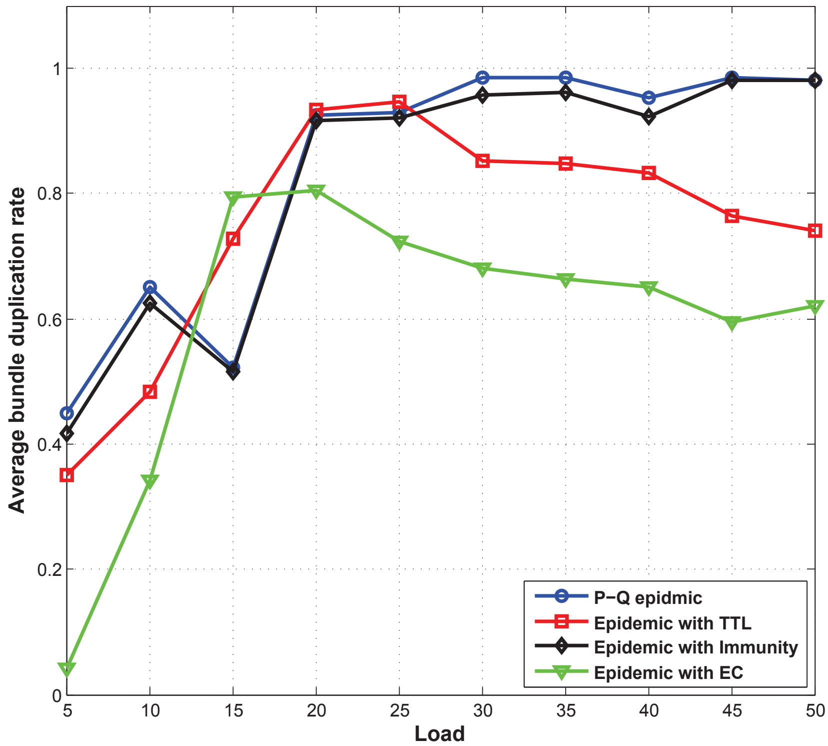

Figure 3.27 Average bundle duplication rate comparison of epidemic-based protocols when nodes move according to the RWP model.

There are three factors that can influence buffer occupancy level: buffer policies, nodes encounter frequency and load. A good buffer policy can free up buffer space in a timely manner and reduce transmission delay. Moreover, a high encounter frequency and load means more bundles are exchanged, which leads to higher buffer occupancy level.

Fig. 3.31 illustrates the buffer occupancy level when nodes use flooding. Note, as there is no buffer policy, when the load exceeds 20, nodes' buffer level is at its maximum regardless of node densities. From here on, the figures only illustrate the difference in buffer occupancy levels. Fig. 3.31 shows that before nodes occupy their entire buffer space, higher node density increases buffer occupancy level. More specifically, in the experiment, when the load is 10, nodes density of 50 has the highest buffer occupancy rate, at more than 90%. This is because higher node density leads to more opportunities to transmit bundles.

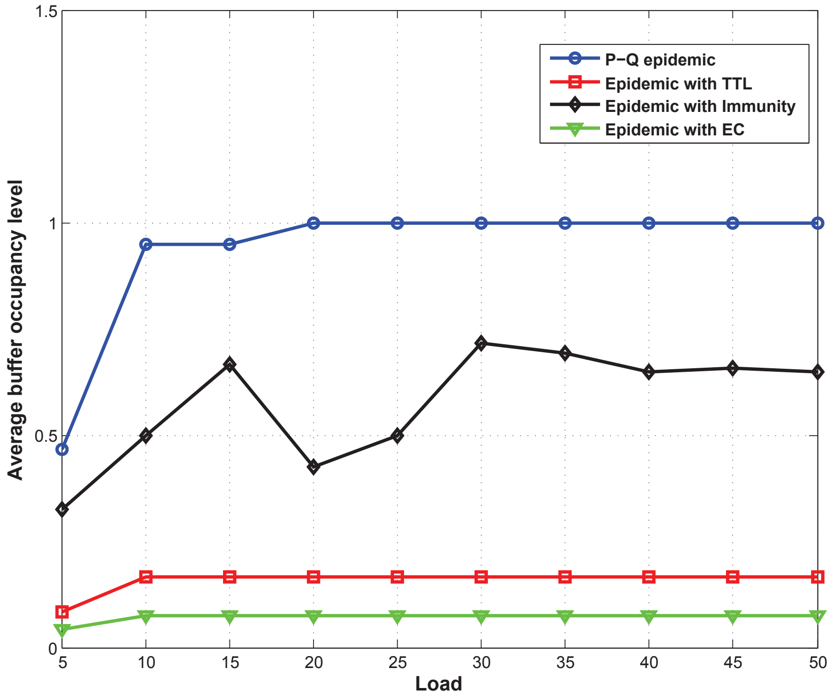

Figure 3.28 Buffer occupancy level comparison of epidemic-based protocols in trace-based studies.

Fig. 3.32 shows how the number of nodes affect nodes' buffer when they use P-Q epidemic. Note that, altering the number of nodes does not have much effect on nodes buffer occupancy level. In particular, when the load is less than 15, nodes density as high as 50 results in less than 5% buffer occupancy level as compared to when there are 10 nodes. The main reason is because when there are a high number of nodes, the encounter frequency increases proportionally. As a result, nodes are able to clear their buffer quicker. However, if nodes use P-Q epidemic, the increase in node density does not directly increase the number of bundles that are exchanged by nodes, but instead, it is governed by the value of P and Q value. For example, if $P = Q = 0.5$, each bundles will need two encounters before it is transmitted, as opposed to being transmitted at each encounter when nodes use flooding.

Fig. 3.33 demonstrates that in epidemic with TTL, smaller number of nodes results in higher buffer occupancy level. As the number of nodes increases from 10 to 50, nodes' buffer occupancy level decreases from 10% to around 5%. This

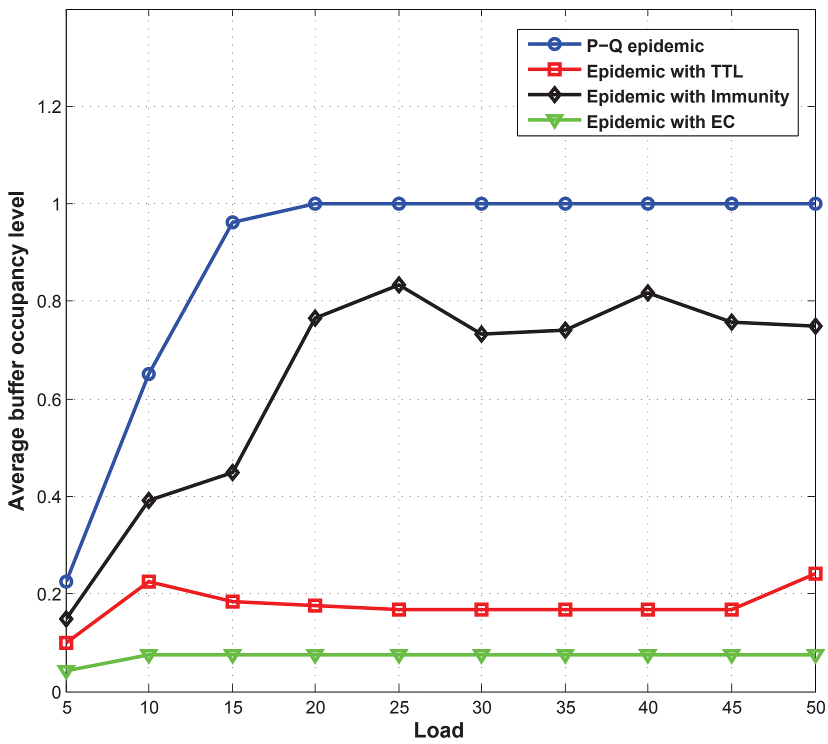

Figure 3.29 Average buffer occupancy level comparison of epidemic-based protocols in RWP studies.

is because of the overall increase in the aggregate buffer space. For example, in a five nodes network, assume the buffer occupancy level is at 5%. This means, in a 10 nodes network, the same set of buffer will only occupy 2.5%.

Fig. 3.34 shows the buffer occupancy level of epidemic with EC in networks with varying number of nodes. There are a number of key observations. First, changing nodes density only leads to less than 10% difference on buffer occupancy level. Second, regardless of nodes density, nodes will eventually occupy 90% of their buffer. Nodes density is not the only factor that influences buffer occupancy level, as it is influenced by nodes encounter frequency and their buffer policy too. In other words, the higher encounter frequency leads to more bundles transmissions, which has the effect of increasing the number incoming bundles. At the same time, there will be a proportional number of bundles leaving a node, which means they have a higher probability of being deleted.

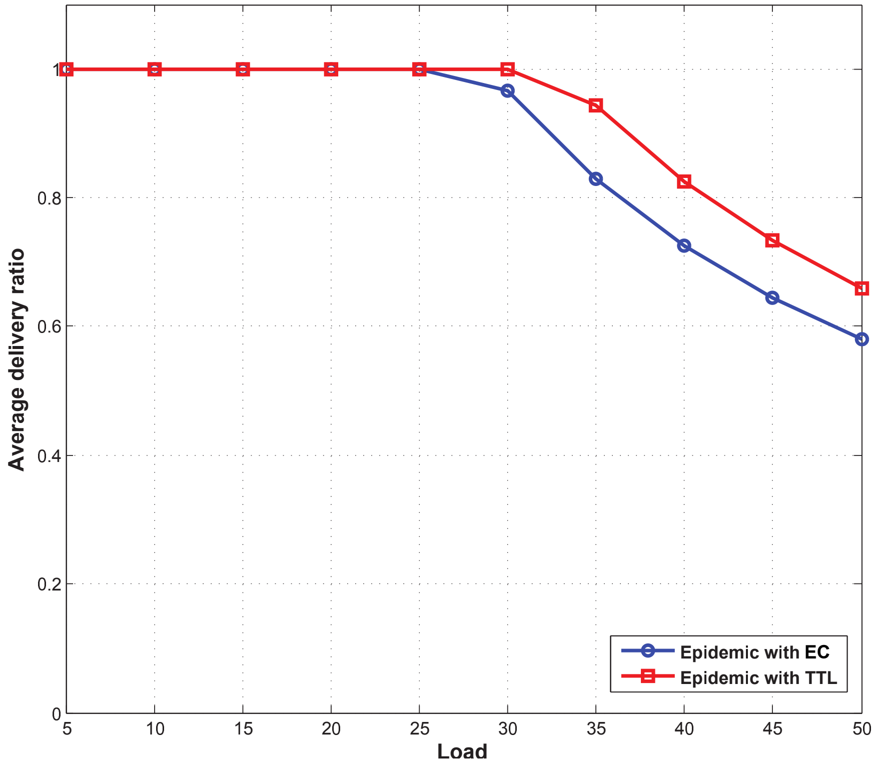

Figure 3.30 delivery ratio comparison of epidemic with TTL and EC.

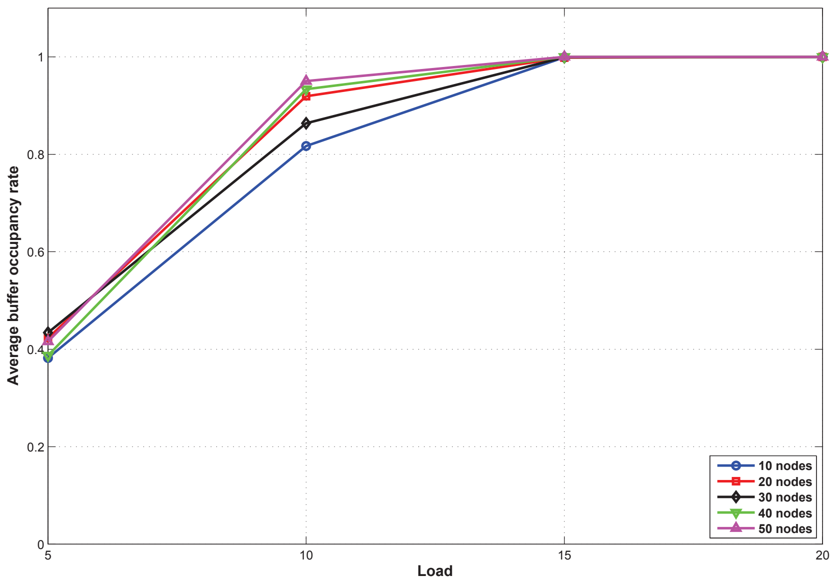

Figure 3.31 Buffer occupancy level when nodes use flooding.

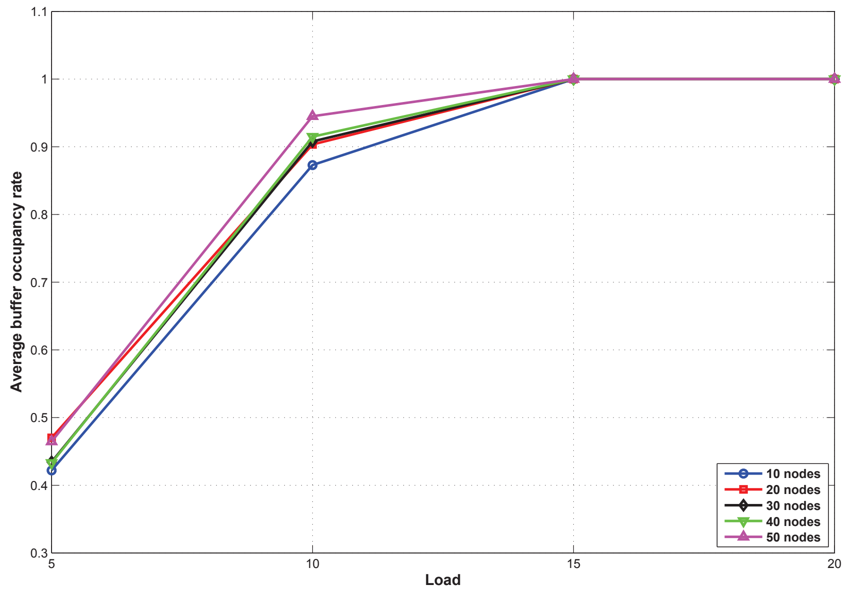

Figure 3.32 Buffer occupancy level of P-Q epidemic versus number of nodes.

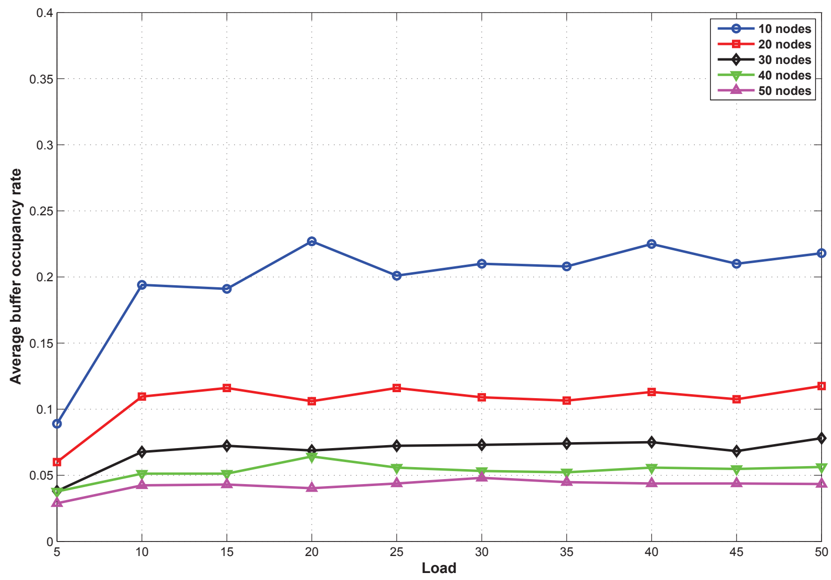

Figure 3.33 Buffer occupancy level of epidemic with TTL with varying node numbers.

Fig. 3.35 displays the buffer occupancy level of epidemic with immunity. The buffer occupancy level curves cross and overlapped each other, and fewer nodes results in a higher probability of achieving the lowest occupancy level. The

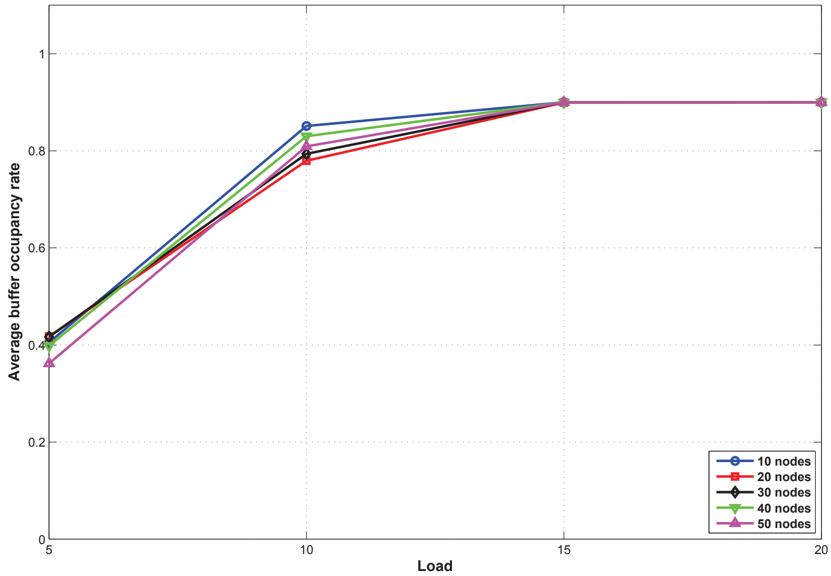

Figure 3.34 Buffer occupancy level of epidemic with EC with different number of nodes.

results imply that having a small number of nodes is beneficial as immunity tables can be propagated faster, and hence, remove bundles that have already reached their destination.

The findings are summarized as following. First, increasing node density effectively creates more opportunities for nodes to encounter each other. For example, when nodes use flooding, a higher encounter frequency leads to an increase in buffer occupancy level.

Second, the impact on nodes' buffer occupancy level depends on whether any buffer policies are invoked at each node encounter. For example, in epidemic with EC and epidemic with immunity table, encounters directly influence nodes' buffer policies, e.g., nodes delete bundles more frequently. Another example is epidemic with TTL, where frequent encounters mean a continually renewal of TTL values. As a result, bundles are more likely to be stored by nodes.

Bundle Duplication Rate

Fig. 3.36 to 3.40 show the bundle duplication rate of all tested protocols. All

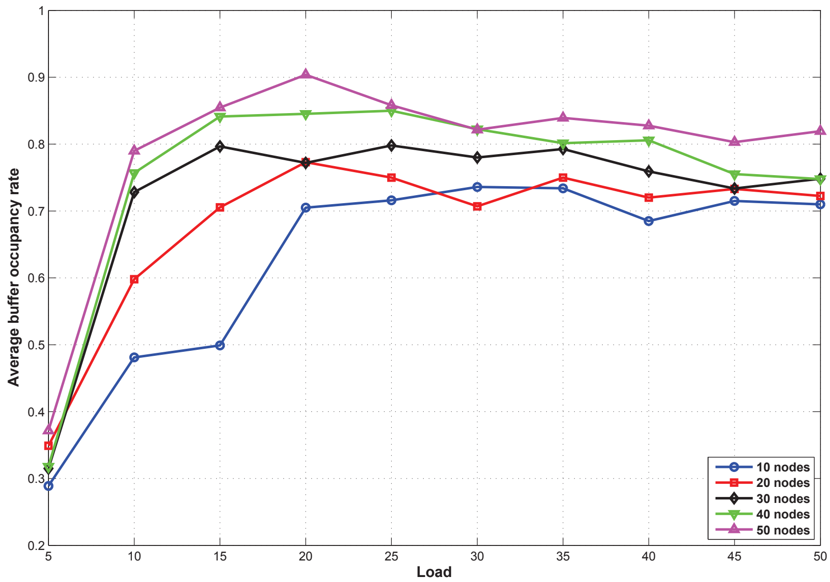

Figure 3.35 The impact of number of nodes on the buffer occupancy level of nodes when they use epidemic with immunity.

figures show the same conclusion. That is, when the number of nodes increases, the bundle duplication rate reduces. Specifically, when the number of nodes is 10, the bundle duplication rate of all routing protocols is over 75%, while at 50 nodes, the bundle duplication rate reduces to below 30%. This implies that when nodes density decreases, bundles have a higher probability of spreading the bundles to all nodes in the network.

Delay

Fig. 3.41 to 3.45 show the delay experienced by each protocol when the number of nodes ranges from 10 to 50. The observation covers the following aspects. First, node density has different influence on delays for all epidemic based protocols. Referring to Fig. 3.43 and 3.44, the delay of epidemic with TTL and EC shows significant differences when the node density varies from 10 to 50. The largest delay difference is more than 100,000 seconds, while other protocols only observe a maximum of 10,000 seconds delay difference. The reason for this large discrepancy is because when there are more nodes, they tend to encounter each other frequently. This allows the TTL of bundles to be renewed before they

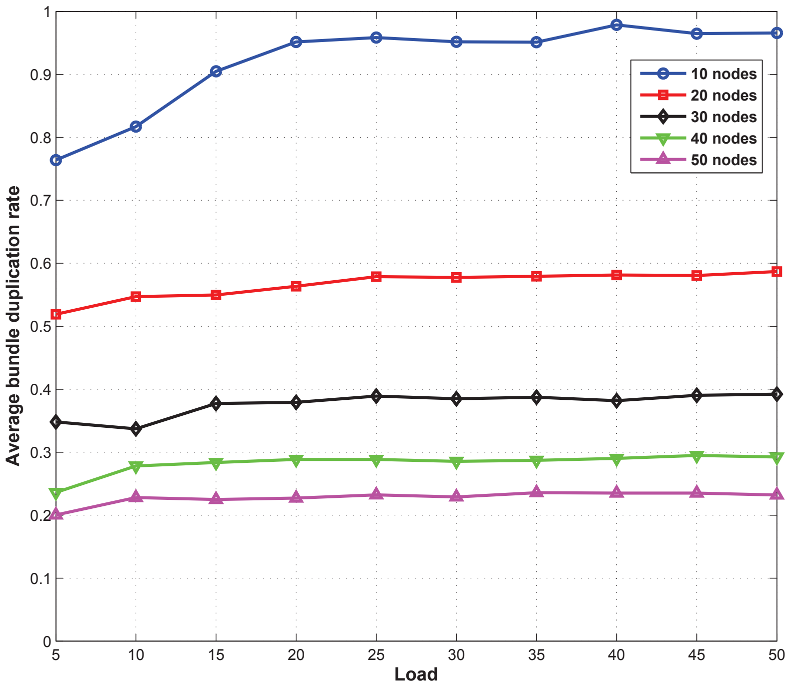

Figure 3.36 Bundle duplication rate when nodes use flooding in networks with different number of nodes.

expire. This also means bundles are more likely to be discarded as a result of the "drop oldest" policy. Hence, a higher delay will be recorded if nodes delete rare bundles.

Secondly, nodes that use epidemic with EC may experience a sudden increase or drop in delay. For example, in Fig. 3.44, when node density is at 40, there is a spike in delay when the load reaches 35. The main reason is because nodes experience a buffer overflow due to short and frequent encounters, which is likely when the node density is high. In these encounters, a node will receive a burst of bundles, which may have small and similar EC values. Once the node's buffer is full, it may result in the deletion of rare bundles.

3.4.2 Enhanced Protocols

This section focuses on the enhancements presented in Section 3.2. Again, both the RWP and trace-file are used in all experiments. These enhancements are

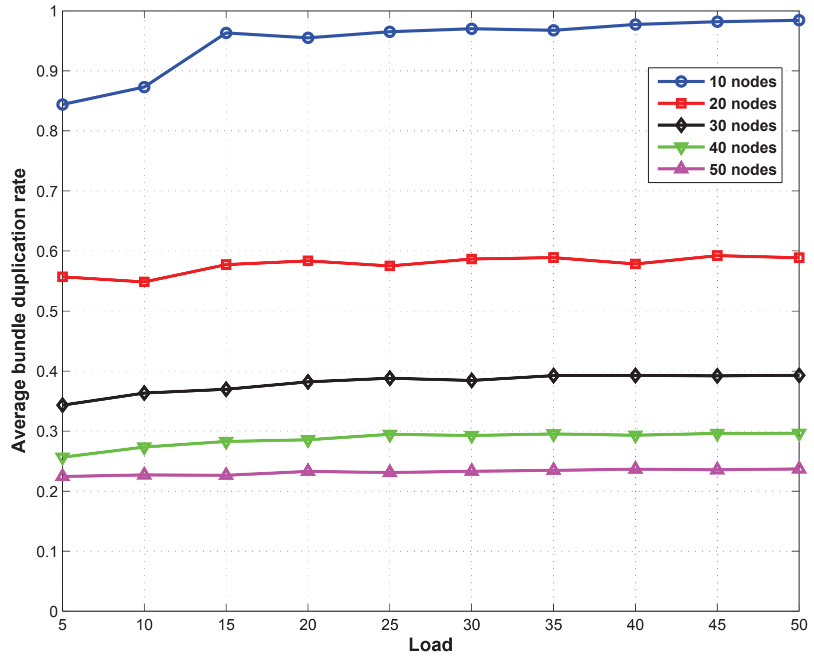

Figure 3.37 Bundle duplication rate when nodes use P-Q epidemic in networks with varying number of nodes.

compared against their corresponding un-modified version. Additionally, in all scenarios that deploy epidemic with constant TTL, the TTL value is set as a constant to 300 seconds.

3.4.2.1 Delivery Ratio

Firstly, the encounter interval time has a significant effect on the delivery ratio of epidemic with constant TTL. For example, the experiments deployed two network scenarios to evaluate the influence of encounter interval on epidemic with TTL. Both scenarios include 20 nodes, each of which has at most 20 encounters with other nodes. The only difference between these two scenarios is that the interval time between two successive encounters is set to a maximum of 400 and 2000 seconds respectively.

Fig. 3.46 shows the delivery ratio achieved by epidemic with constant TTL value. As the figure showed, when the interval between encounter increases,

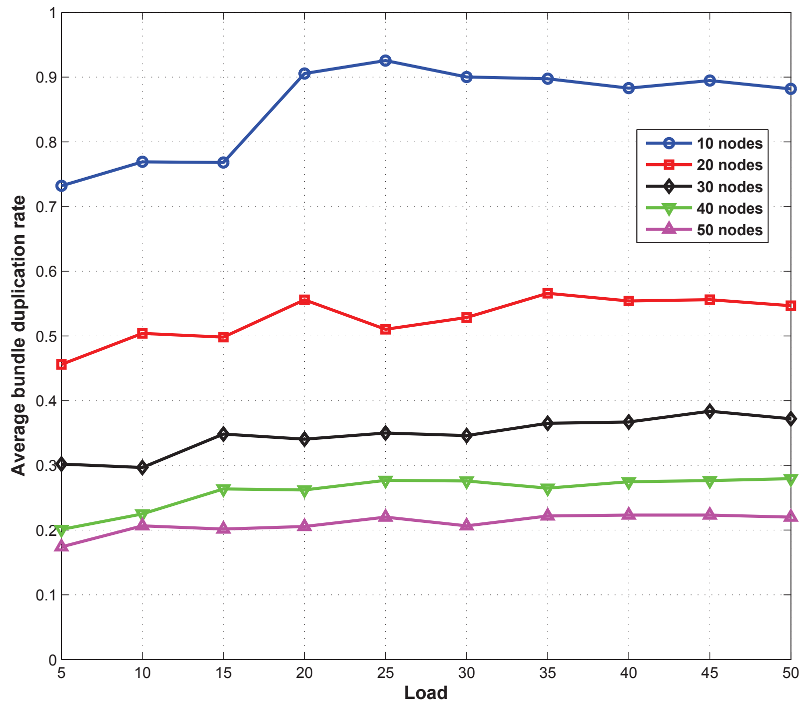

Figure 3.38 Bundle duplication rate when nodes employ epidemic with TTL in networks with different number of nodes.

delivery ratio drops dramatically - viz. an interval time of 2000 has a 20% lower delivery ratio than an interval time of 400. The main reason is because nodes delete bundles before they are transmitted - i.e., the average interval time is longer than the TTL of bundles.

Fig. 3.47 and 3.48 show the delivery ratio of all modified and un-modified protocols. As shown in the figures, dynamic TTL has a higher delivery ratio than epidemic with constant TTL values. In particular, the delivery ratio of dynamic TTL is ranging from 20% to 100%, which significantly increases the delivery ratio over constant TTL by 40% in trace file and 20% in RWP model respectively. The higher delivery ratio is due to bundles having a dynamic TTL value corresponding to the intervals between encounters, which reduces the likelihood of nodes discarding bundles that have not arrived at their respective destination. Apart from that, with dynamic TTL, nodes also can effectively discard bundles after those bundles are exchanged. This is because the TTL set

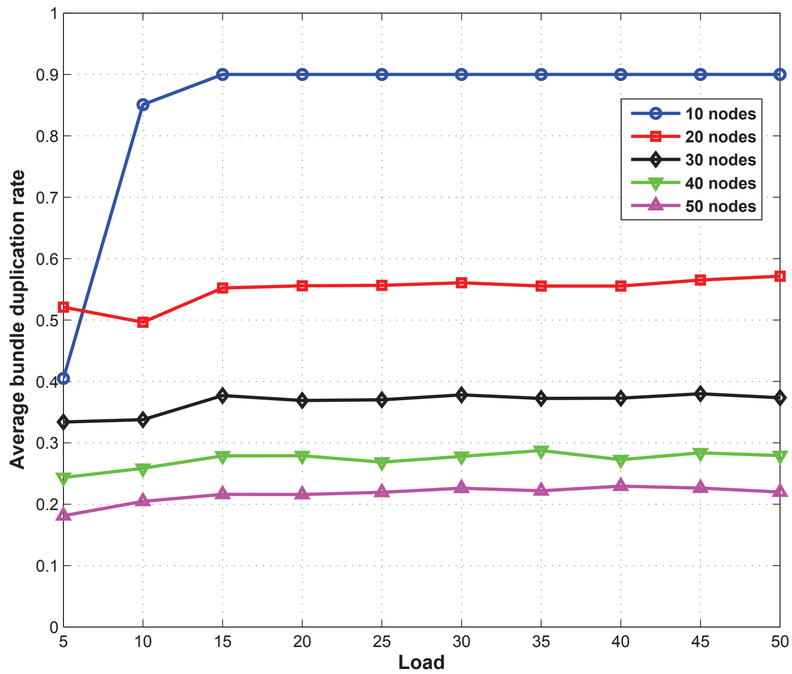

Figure 3.39 Bundle duplication rate when nodes use epidemic with EC in networks with different number of nodes.

for bundles changed according to nodes encounters, therefore, once bundles are exchanged, old bundles are discarded from buffers in a short time, and nodes have more buffer size for new undelivered bundles.

The epidemic with EC+TTL increases delivery ratio and has over 80% delivery ratio in both RWP and trace file scenarios. In particular, in trace file experiments, epidemic with EC+TTL has a much higher delivery ratio than epidemic with EC when the load is 30. For example, when the load is 45, epidemic with EC+TTL results in more than 85% delivery ratio, whilst epidemic with EC has less than 60% delivery ratio. The reason for the higher delivery ratio is due to, on one hand, by the EC threshold of bundles, which encourages bundles duplication and hence, increases their delivery probability. On the other hand, with the use of TTL, nodes are able to free up their buffer and thus, store more undelivered bundles. Lastly, the delivery ratio of epidemic with cumulative immunity is similar to epidemic with immunity. This is because cumulative

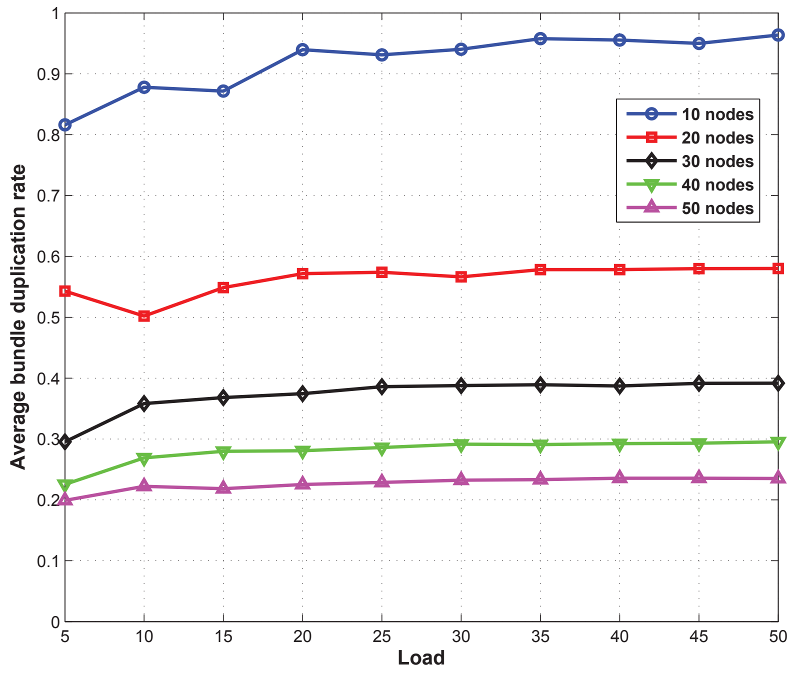

Figure 3.40 Bundle duplication rate when nodes use epidemic with immunity in networks with different number of nodes.

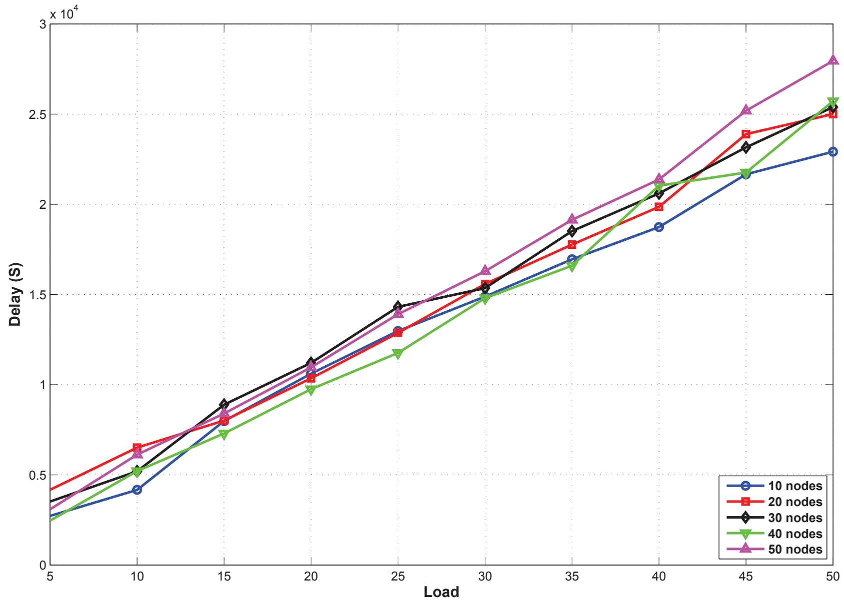

Figure 3.41 Delay experienced by nodes using flooding in networks with different number of nodes .

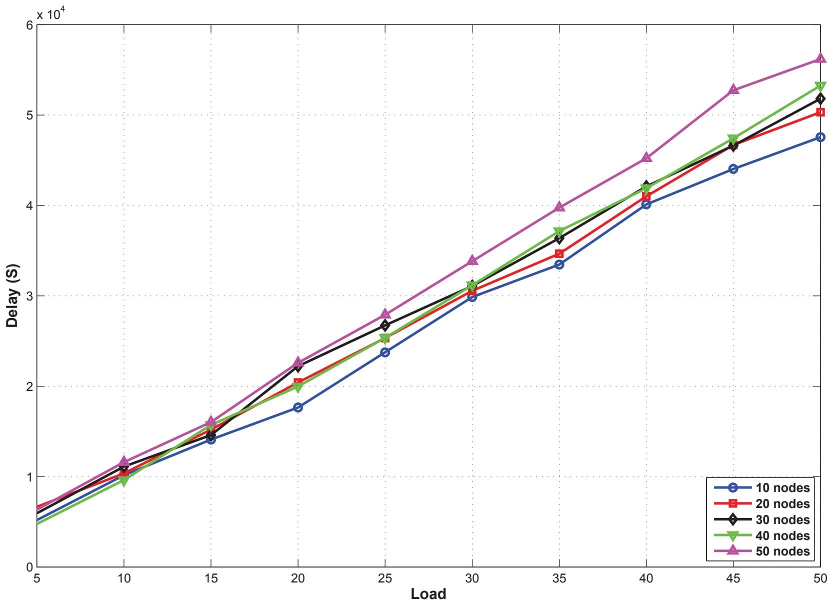

Figure 3.42 Delay resulting from the use of P-Q epidemic in networks with 10 to 50 nodes .

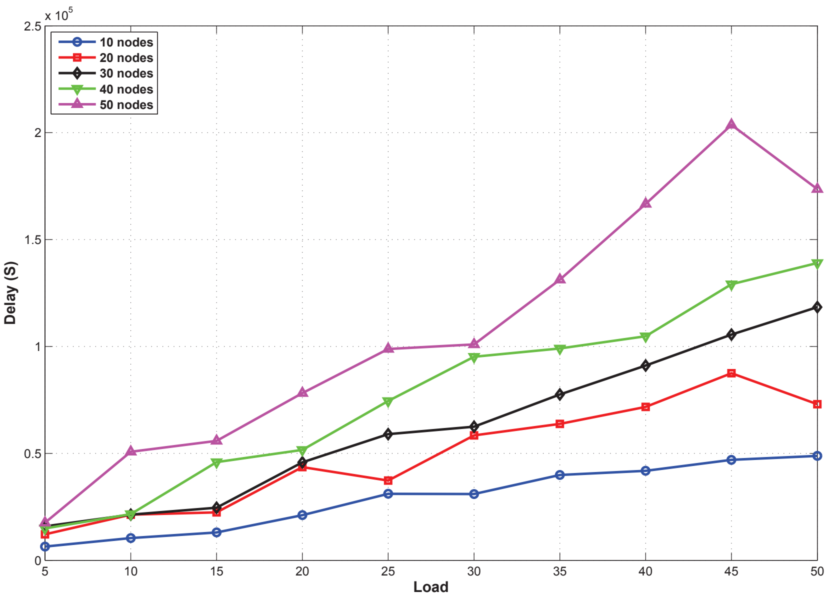

Figure 3.43 Delay resulting from the use of epidemic with TTL in networks with different number of nodes.

immunity is a buffer policy. It has no influence on the transmission of bundles before they are received by their respective destination.

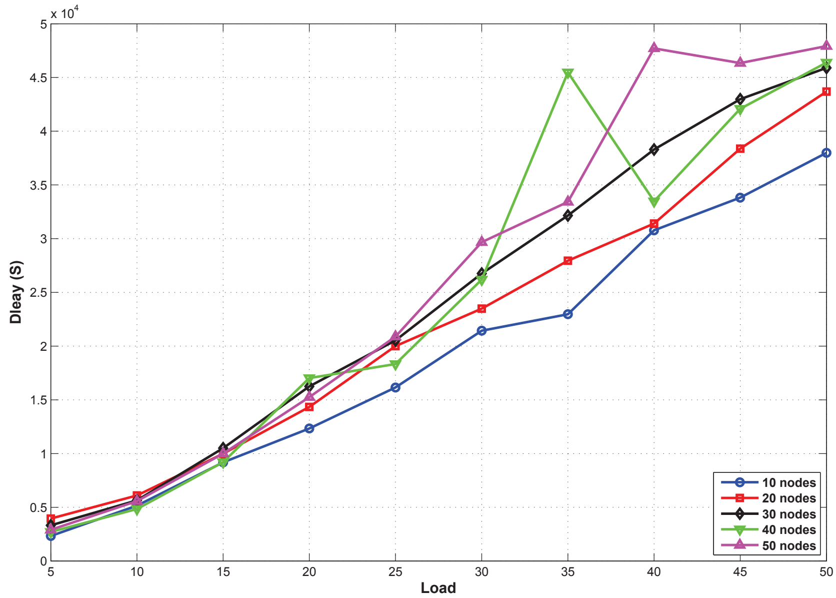

Figure 3.44 Delay resulting from the use of epidemic with EC in networks with varying number of nodes .

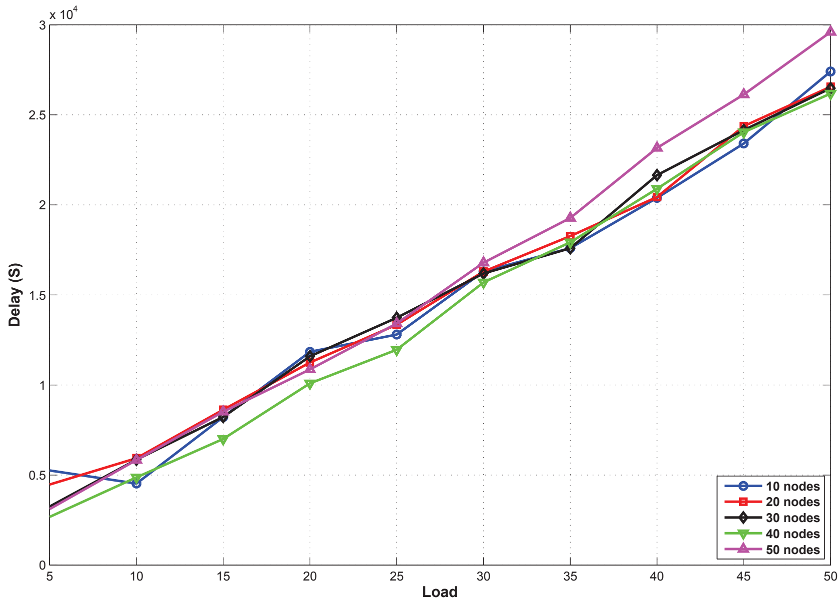

Figure 3.45 Delay comparison of epidemic with immunity in networks with different number of nodes.

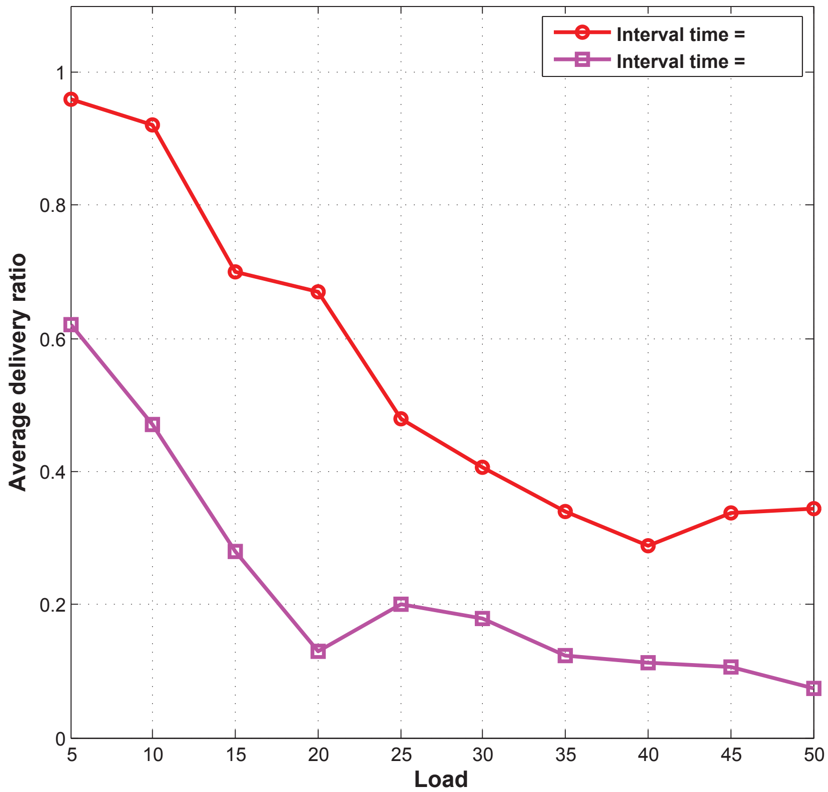

Figure 3.46 Delivery ratio comparison of epidemic with TTL =300 in two scenarios with different interval times.

3.4.2.2 Buffer Occupancy Level

Fig. 3.49 and 3.50 illustrate the buffer occupancy level of all modified protocols. The figures show that epidemic with dynamic TTL increases buffer occupancy level, but remains less than 20%. The main reason is because dynamic TTL values lead to longer buffering time, and consequently, higher buffer occupancy level. This can be seen from Fig. 3.49, in which the highest buffer occupancy level is reached in the scenario with 2000 seconds interval time. When the interval time is reduced, the buffer occupancy level decreases accordingly.

By comparing EC+TTL and epidemic with EC, the results show that, epidemic with EC+TTL reduces the buffer occupancy level of nodes. In the RWP model, the buffer occupancy level of epidemic with EC+TTL is ranging from 20% to 68%, which is 10% less than that of epidemic with EC only. However, given

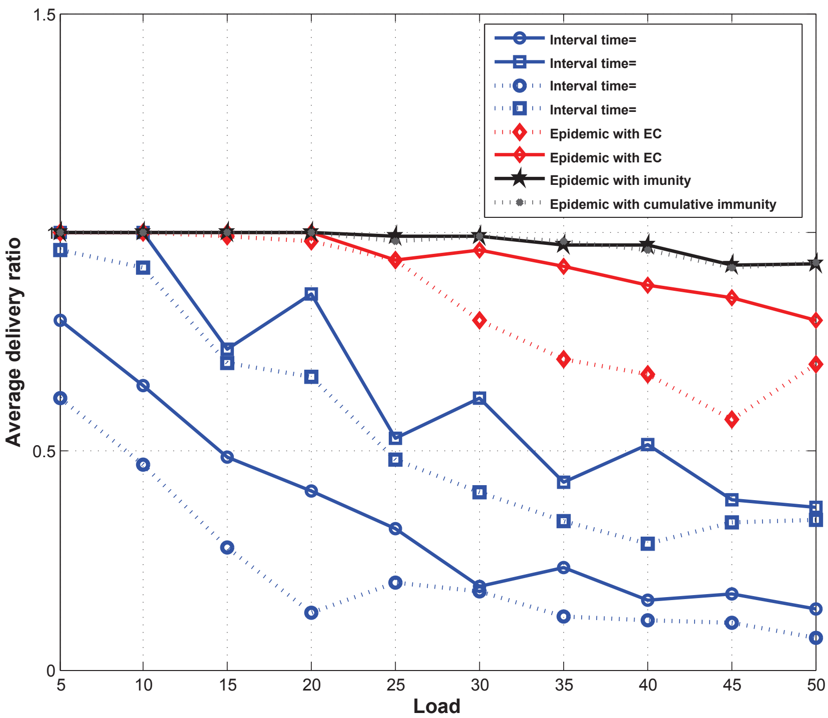

Figure 3.47 Delivery ratio comparison of modified and un-modified protocols in RWP model.

that nodes have fewer encounters in the trace file, the EC threshold is never reached, and hence, the TTL of bundles remains constant. As a result, nodes have a higher buffer occupancy level than in the RWP model, which is ranging from 25% to 87%. For example, when the load is at 20, epidemic with EC+TTL costs nodes less than 50% of their buffer in the RWP model, but is over 60% in trace file based experiments. Finally, the results highlight the effectiveness of using cumulative immunity tables in reducing nodes' buffer occupancy levels - as shown in RWP and trace file scenarios.

3.4.2.3 Duplication Rate

Fig. 3.51 and 3.52 show that dynamic TTL has a different performance in terms of duplication rate. In the RWP model, dynamic TTL has a maximum 10% higher duplication rate as compared to experiments where nodes use a constant TTL. In trace file experiments, the maximum difference in duplication rate between dynamic and constant TTL is 20%. The increased duplication

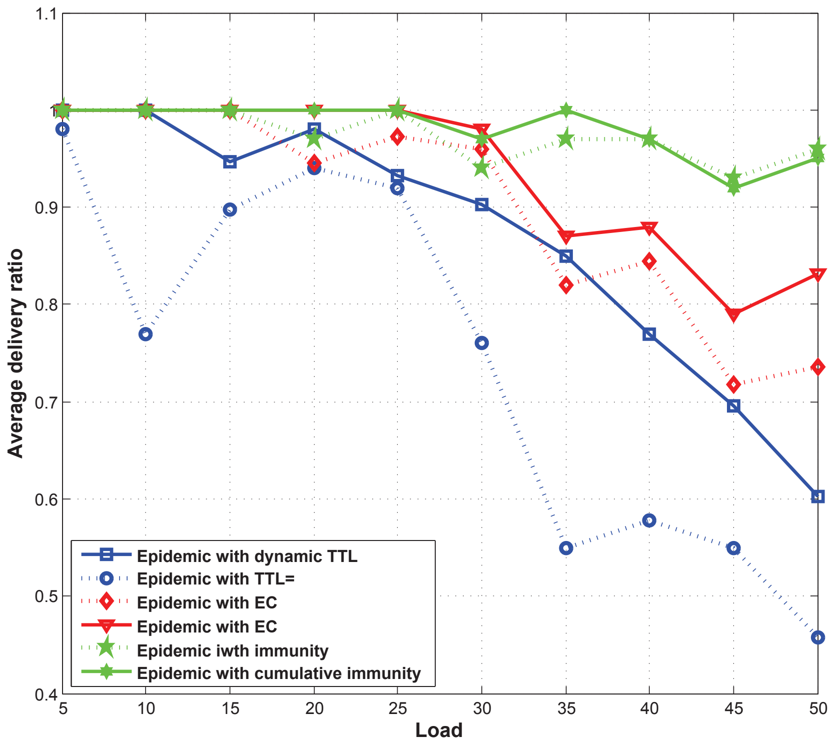

Figure 3.48 Delivery ratio comparison of modified and un-modified protocols in trace-file based study.

rate is due to bundles having a lower probability of being discarded, and this is attributed to dynamic TTL values that let nodes store bundles until they encounter other nodes. As a result, bundles have a higher chance of being transmitted to more nodes, which increases duplication rate.

The results shown in both Fig. 3.51 and 3.52 also imply that epidemic with EC+TTL has a similar bundle duplication rate, in which the difference is less than 10%. Note that, in both RWP and trace file experiments, when the load is greater than 30, epidemic with EC+TTL has a higher bundle duplication rate. This is because, unlike epidemic with EC, epidemic with EC+TTL sets a transmission count threshold values for each bundle. That is, before each bundle is deleted, it must have been transmitted a given number of times. As a result, bundles are dispatched to more nodes. On the other hand, when the load is less than 30, as the total number of bundles is small, the destination is able to receive all bundles in a short period of time. Consequently, there are fewer

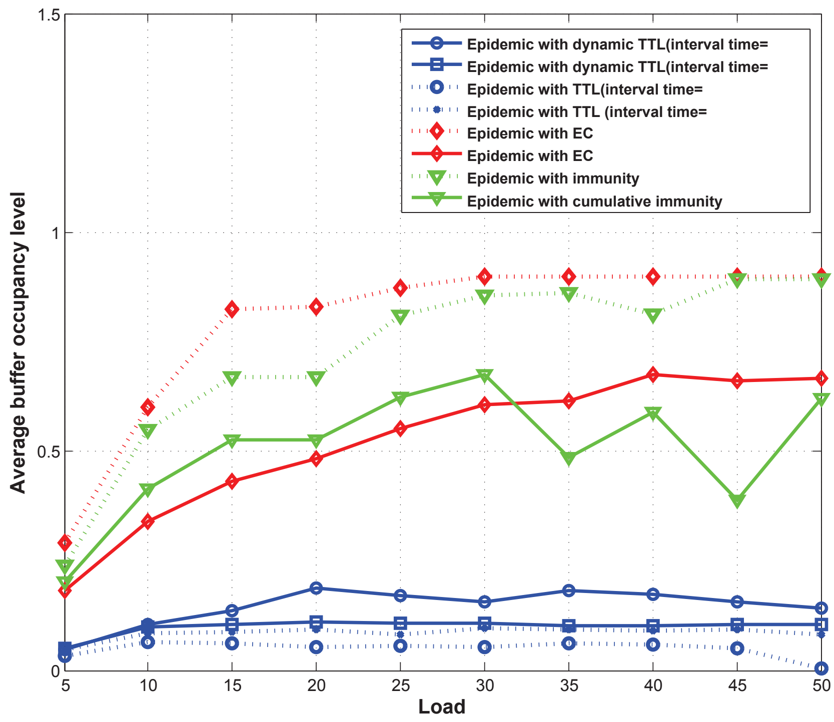

Figure 3.49 Buffer occupancy level comparison of modified and un-modified protocols in RWP model.

redundant bundles, and hence, two protocols have similar bundle duplication rate.

The use of cumulative immunity tables reduces bundle duplication rate in both RWP and trace file experiments. Note, the bundle duplication rate is lower in the RWP model, which is ranging from 36% to 49%. This is because bundle duplication rate is closely related to the dissemination of the immunity table. In the trace file scenario, nodes have fewer encounters than those that move according to the RWP model. As a result, the immunity table is propagated more slowly to nodes, which leads to higher bundle duplication rate.

3.4.3 Discussion

Tab. 3.2 compares original epidemic-based protocols and their enhanced counterparts. Note, all the values in the table are average values. First, except epidemic with immunity and cumulative immunity, enhanced protocols have a

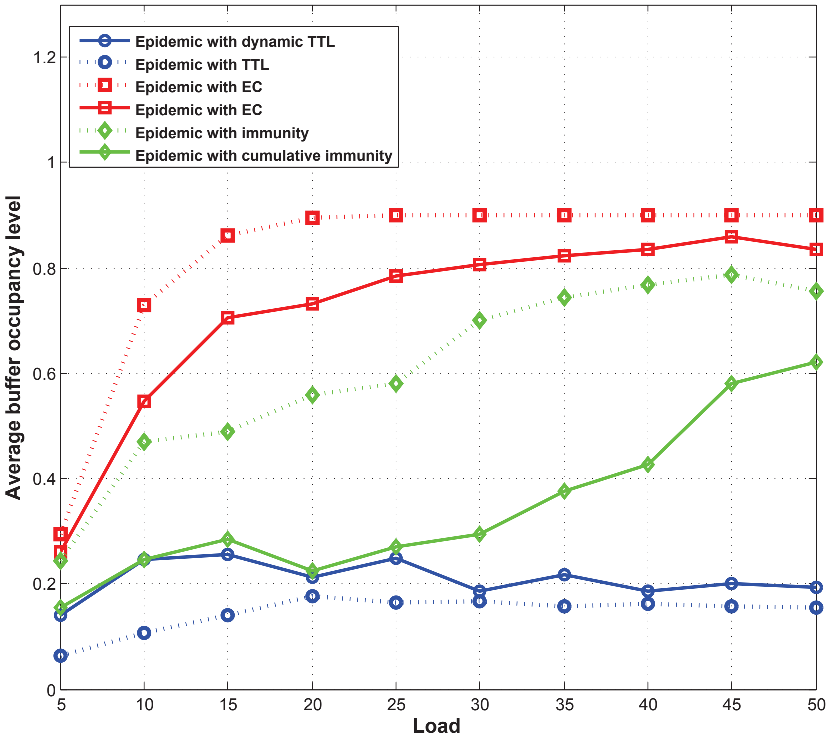

Figure 3.50 Buffer occupancy level comparison of modified and un-modified protocols in trace-based study.

higher average delivery ratio. In particular, the biggest difference is between dynamic and constant TTL, where dynamic TTL improves delivery ratio in both trace-file and RWP experiments by 12% and 40% respectively. The main reason for this significant improvement is that nodes are able to adapt the TTL of bundles in accordance with varying contact duration. Similarly, epidemic with EC+TTL avoids discarding bundles prematurely. Both epidemic with immunity and cumulative immunity have the same delivery ratio. However, as will discuss later, they have a low buffer occupancy level.

Second, dynamic TTL has a higher buffer occupancy level - in fact the lowest recorded buffer occupancy level is 12% and 11% higher than its original counterpart in trace-file and RWP experiments respectively. This, however, leads to superior bundle delivery ratio as bundles are stored by nodes for a longer period time, especially when the frequency of contact is low. The experiments demonstrates that, epidemic with EC and EC+TTL have the highest buffer

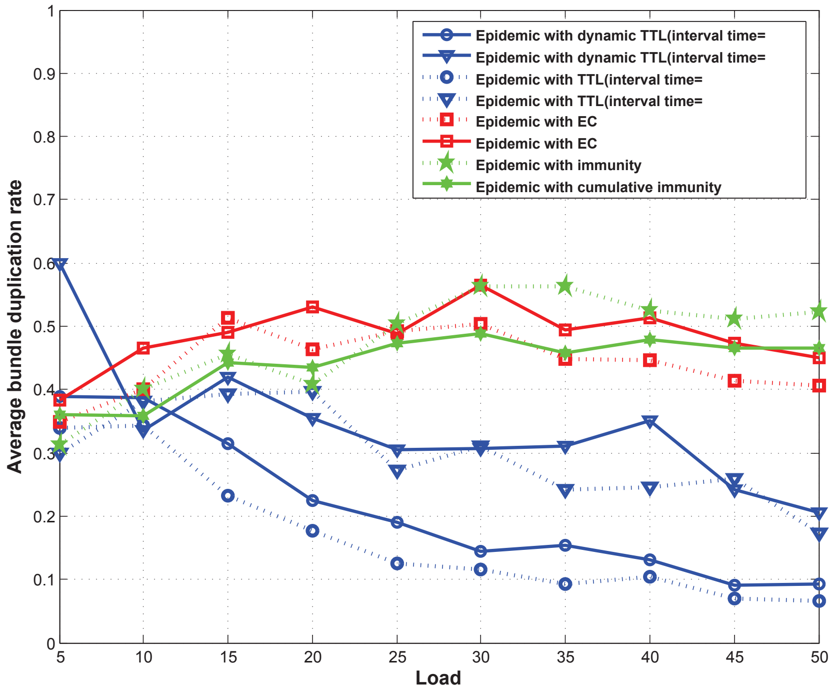

Figure 3.51 Bundle duplication rate comparison of modified and un-modified protocols in RWP model.

occupancy level. However, by incorporating a TTL value, the buffer occupancy level reduces from 79% to 57% and from 74% to 59.5% in trace file and RWP studies respectively. In other words, EC+TTL has approximately 20% lower buffer occupancy level than its counterpart.

Third, except for epidemic with cumulative immunity, the enhancements have slightly higher bundle duplication rate. In particular, dynamic TTL increases the lowest duplication rate from 66% to 69% and from 13.8% to 22.8% respectively in trace file and RWP experiments. Note that, epidemic with immunity has the highest duplication rate – 82% in trace file, and 48% in RWP experiments. Advantageously, a high bundle duplication rate means better delivery ratio - as demonstrated by epidemic with dynamic TTL and EC+TTL. This is a fundamental feature of epidemic-based protocols as they are highly dependent on contact frequency. This means when bundle duplication rate is high, bundles can be forwarded quickly throughout the network, and thereby, lead to high bundle delivery ratios. Apart from that, the results show that epidemic

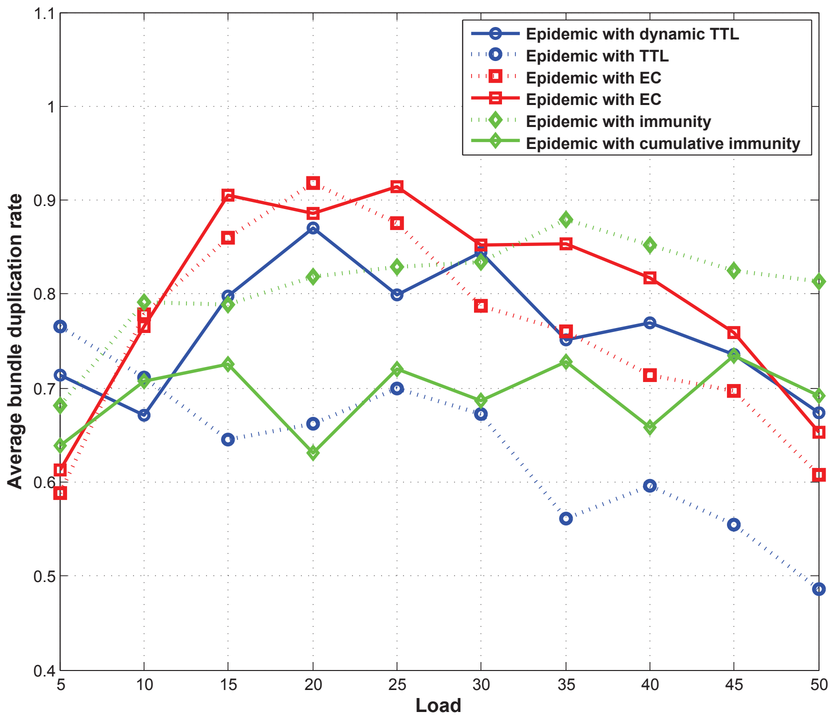

Figure 3.52 Bundle duplication rate comparison of modified and un-modified protocols in trace-file study.

with a cumulative immunity table is able to maintain a high delivery ratio with low duplication rate. This is primarily due to the effectiveness of the cumulative immunity table in purging received bundles from nodes.

3.5 Conclusion

This chapter has compared epidemic-based protocols using a unified framework. Specifically, the same mobility models, i.e., trace-file and RWP and performance metrics are implemented to compare key epidemic-based protocols. The results show that P-Q epidemic increases transmission delay and has poor bundle delivery ratio. Epidemic with immunity table has the highest delivery ratio at the expense of higher buffer occupancy level. In addition, the use of a constant TTL value results in poor performance as nodes in DTNs have wide ranging contact intervals. In addition, epidemic with EC experiences high buffer occupancy level

Table 3.2 comparison of original and enhanced protocols.

	Delivery rate (%)		**Buffer occupancy level (%)**		**Duplication Rate (%)**	
	RWP	Trace file	RWP	Trace file	RWP	Trace file
Epidemic with TTL	24.6	74.4	5.1	11.3	13.8	66.3
Epidemic with Dynamic TTL	64.7	86.8	16.3	23.3	22.8	75.4
Epidemic with EC	76.4	88.2	74.6	79.7	45.7	79.2
Epidemic with EC+TTL	92.5	93.6	59.5	57.1	49.3	80.3
Epidemic with Immunity table	97.7	95.3	72.5	58.2	48.5	82.4
Epidemic with Cumulative Immunity table	98.4	98.6	45.8	32.8	35.5	69.4

and long delivery delay. Accordingly, this chapter proposes three enhancements to address these limitations: epidemic with dynamic TTL, EC+TTL and cumulative immunity table. The extensive experiments show these enhancements to have high delivery ratio. Moreover, the use of cumulative immunity tables helps reduce duplication rate and buffer occupancy level significantly.

Chapter 4

Multicasting

4.1 Introduction

This chapter focuses on epidemic-based multicast routing protocols. First, to date, little research, e.g., [31] and [3], has studied epidemic-based protocols for delivering bundles to multicast subscribers. Specifically, only a few works have studied the efficacy of epidemic routing protocols in delivering bundles to multicast group members. Thus far, only the author of [31] investigated epidemic with TTL. The authors, however, have not considered multicast sessions with a large number of subscribers, where a small TTL value may lead to bundles expiring prematurely. This means epidemic with EC or epidemic with immunity are better suited for multicasting in DTNs. However, currently, no work has conducted any investigation on these two buffer protocols.

Second, no researchers have studied the effect of anti-entropy in multicast scenarios. In [4], the authors pointed out that "comparing before exchanging" is critical because it avoids duplicated bundles. This is important because once a node receives a bundle, it will never receive the bundle again even though said bundle has been discarded from its buffer. However, in multicast, given that there may be more than one subscriber, it is unclear whether anti-entropy will result in lower bundle delivery ratio. Intuitively, a node that has delivered a bundle, and subsequently deleted the bundle before meeting another subscriber should be given another copy of said bundle to improve bundle delivery ratio.

Third, no research has analyzed the impact of having multicast subscribers act as relay nodes to forward bundles. In every epidemic variant, e.g., [31], [43], [3], subscribers are effectively sink nodes, and are not required to forward bundles to others. However, in multicast, it is unclear how this influences multicast delivery. This is an important consideration when there are more subscribers than non-subscribers or relay nodes. If there are only a few relay nodes, the probability of a subscriber meeting a relay node becomes less, which increases end-to-end delays. In the worst case, only the source is the relay and all other nodes are subscribers. This means the source will have to meet each subscriber in order to deliver bundles.

Fourth, the size of a multicast group affects the delivery ratio of epidemic with TTL [31]. Specifically, the experiments conducted by [31] showed that when the multicast group size increases, the delivery ratio drops accordingly. In particular, when the multicast group size increases from two to 60, the delivery ratio reduces from 95% to 50%. However, it is unclear whether multicast group size has any impact on delivery ratio if subscribers do not forward bundles, or when nodes employ EC or immunity packets to discard bundles. This is because the ratio of multicast subscribers to non-subscribers is a critical issue as larger multicast group sizes mean fewer relay nodes, and vice-versa. Moreover, in epidemic with immunity, the number of immunity packets corresponds to the multicast group size. Increasing the multicast group size translates to proportionally more immunity packets. This, however, corresponds to faster bundle deletion despite some subscribers not having received the deleted bundles.

Henceforth, this chapter makes the following contributions. Firstly, this chapter presents three key findings concerning delivery ratio: (i) subscriber nodes must act as relays. The results show that bundle delivery ratio is only 57% as compared to 100% when they act as relays, (ii) the use of anti-entropy contributes to a rise in delivery ratio to 100%, which is 57% higher than when nodes do not use anti-entropy, (iii) higher number of relay nodes improve delivery ratio. In the experiments, when the number of relay nodes increases from 10 to 30, the delivery ratio rises from 43% to 98%.

Secondly, this chapter investigates the influence of varying multicast group sizes. The results reflect that the impact of multicast group size on bundle delivery ratio is dependent upon subscribers' forwarding policies - i.e., whether they are relays or sinks. Specifically, when subscribers work as sinks, epidemic with TTL, epidemic with immunity, epidemic with EC threshold, and epidemic with cumulative immunity, have poor bundle delivery ratio. On the other hand, for epidemic with immunity, it has a low bundle delivery ratio when subscribers act as relays.

Lastly, this chapter studies the buffer occupancy level of relay nodes, and finds that multicast group size, anti-entropy session and subscribers forwarding policies have a significant impact on the buffer occupancy level of relay nodes. For example, large multicast group sizes have an influence on the buffer occupancy level of relay nodes. This too is affected by subscribers' forwarding policies. Specifically, when subscribers are sinks, lower buffer occupancy level is observed with increasing multicast group size, whilst there is no impact when subscribers are relays. Apart from that, all protocols experience high buffer occupancy level. Specifically, pure epidemic and epidemic with EC incur 100% buffer utilization. To address this problem, each bundle is assigned an EC quota. In other words, EC quota bounds the number of times in which a bundle is exchanged by relay nodes. The simulation results show that when the EC quota is 10 in a network with 20 subscribers, the buffer occupancy level of epidemic with dynamic TTL reduces from more than 95% to less than 15% whilst maintaining similar bundle delivery ratio.

The rest of this chapter is organized as follows. Section 4.2 describes the research methodology. Section 4.3 and 4.4 present the results and conclusions respectively.

4.2 Research Methodology

The experiments in this chapter use the same research methodology as that of Chapter 3; see Section 3.3. There is only one source, and one subscriber group. The nodes that are neither the source nor subscribers are termed *relay* nodes. The source node is chosen randomly, and has k bundles. Subscriber nodes have an infinite buffer size. Subscribers is also allowed to act as relay nodes, in other words, subscribers can store bundles for each other. The value of k is increased by 10 after each experiment, and stops at 50. For each k value, 100 simulation runs are conducted and the results are averaged. All results are at 95% percentile. The source and subscriber nodes are also changed after each run. Moreover, to avoid collision, the node with the lower ID will send first. Once all subscribers received k bundles, the simulation ends. Also, the maximum simulation time is 500,000 seconds. Bundles are generally much bigger than messages in conventional networks. For example, bundles in [22] range from several hundreds of Megabytes to Terabytes. Consequently, the transmission time is fixed to 100 seconds.

The experiments also consider different multicast group sizes. Specifically, the multicast group size is varied from 10 to 30 at an increment of 10. In each simulation, the multicast group size is fixed, and nodes do not change their membership. Each node has a finite buffer size of 10 bundles. Furthermore, the transmission rate of each bundle is one bundle/second. This means the number of bundles exchanged is directly proportional to the rendezvous duration.

Anti-entropy is implemented as follows. Each node has an anti-entropy table that records received bundles' ID. When two nodes encounter each other and are ready to transmit bundles, they check each other's anti-entropy table to determine missing bundles. Note that, with anti-entropy, nodes receive a bundle only once, even though a node may have deleted it. In experiments related to epidemic with EC quota, it limits a bundle's hop count in order to reduce the average buffer occupancy level. For example, when the hop count limit is two, this means the bundles can only be stored by two relay nodes. The first two relay nodes that encounter the bundle will store it in their buffer.

In the experiments, the following metrics are recorded:

- **Delivery ratio** – the average percentage of received bundles. This metric is calculate as $\frac{\sum x_i}{a}$, where a is the multicast group size, x_i is subscriber i's delivery ratio.
- **Buffer occupancy level of relay nodes** – the average buffer utilization of all relay nodes.
- **Delay** – the average time when all subscribers receive all bundles. This metric is collected only when protocols have 100% delivery ratio. Specifically, delay is calculated as $\frac{\sum d_i}{a}$, where d_i is the delay recorded at subscriber i.

4.3 Results

This section presents the experimental results in terms of delivery ratio, delay and buffer occupancy level. Section 4.3.1 focuses on the factors that can influence delivery ratio. Namely, anti-entropy, subscriber forwarding polices, EC value of bundles and multicast group size. Section 4.3.2 compares transmission latency or delay differences when protocols have a delivery ratio of 100%. Lastly, Section 4.3.3 investigates the buffer occupancy of relay nodes for different forwarding protocols.

4.3.1 Delivery Ratio

4.3.1.1 Influence of anti-entropy session

Fig. 4.1 shows the delivery ratio of epidemic-based protocols when nodes use or do not use anti-entropy in RWP. The results show that, on one hand, removing anti-entropy session results in protocols having lower delivery ratio. For example, pure epidemic and epidemic with TTL have a delivery ratio of 78% and 64% respectively. For pure epidemic, removing anti-entropy means nodes can repeatedly receive old bundles. Moreover, old bundles are exchanged frequently, i.e., as more nodes store old bundles than those store new bundles, therefore, the old bundles have higher probability to be exchanged, and new bundles are likely to be replaced by old bundles. For epidemic with TTL, repeatedly receiving an old bundle implies the bundle's TTL value cannot be used for discarding bundles. This is because receiving an old bundle has the effect of renewing a bundle's TTL. As a result, bundles remain in nodes' buffer forever and result in reduced buffering capacity. On the other hand, epidemic variants such as epidemic with EC, epidemic with dynamic TTL and epidemic with cumulative immunity are not influenced by the lack of anti-entropy session as their delivery ratio remains higher than 95%. This is because their respective buffer management policy works similarly to an anti-entropy session. Recall that, the main function of anti-entropy session is to avoid duplicated bundles being transmitted and exchanged at each contact. In this regards, the buffer policy epidemic with EC, epidemic with dynamic TTL and epidemic with cumulative immunity are able to delete duplicated bundles, and hence, have the same impact as anti-entropy session. For example, epidemic with EC discards bundles that have been transmitted the most times. Note that, when nodes receive the same bundle twice, the second one must have a higher EC value than the first one. Therefore, the most transmitted bundles are always deleted from nodes' buffer.

Removing anti-entropy session does not lead to low delivery ratios when nodes use epidemic with immunity and epidemic with EC threshold. Referring to Fig. 4.1, regardless of anti-entropy session, both epidemic with immunity and epidemic with EC threshold have a declining delivery ratio. Specifically, at a load of 50, their delivery ratio is only 43% and 70% respectively. The main cause is immunity packets. That is, subscribers prematurely disseminate immunity packets before all subscribers received a bundle. For example, given two subscribers, node A and B, and assume node A has received bundle k. Consider the scenario where node B did not receive bundle k. As node A has received bundle k, it sends an immunity packet to purge bundle k from the network. If node A is successful, node B will never receive bundle k. Note, these immunity packets also remove any redundant bundles caused by the removal of anti-entropy session. In epidemic with EC threshold, nodes delete bundles only when their EC value exceeds the EC threshold. Consequently, if nodes' buffer is occupied by bundles that have smaller EC values than said threshold, nodes will have no space left for new or duplicated bundles, and thus experience low delivery ratios.

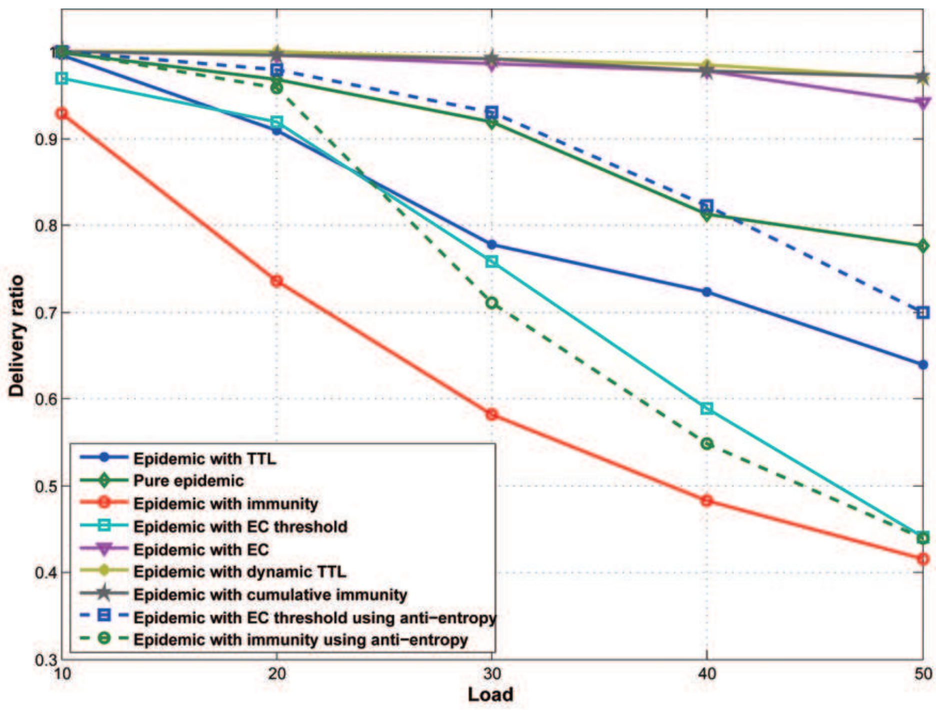

Figure 4.1 Delivery ratio for with and without anti-entropy sessions in RWP.

Fig. 4.2 demonstrates what happens when anti-entropy is removed for the trace file scenario. All protocols can be classified into two groups depending on whether anti-entropy has any influence on delivery ratio. On one hand, protocols such as pure epidemic, epidemic with TTL, epidemic with cumulative immunity and epidemic with dynamic TTL have reduced delivery ratio when they operate without anti-entropy. For example, transmitting 50 bundles in pure epidemic using anti-entropy result in 40% higher delivery ratio. This is because without anti-entropy sessions, old bundles are received repeatedly by nodes. Thus, new bundles are replaced by old bundles. Moreover, anti-entropy also has an impact on buffer policy. For example, removing anti-entropy from epidemic with TTL result in nodes continuously receiving redundant bundles with reset TTL value. Therefore, redundant bundles lead to network congestion, and this reduces delivery ratio. On the other hand, the delivery ratio of protocols such as epidemic with EC and epidemic with immunity is not influenced by anti-entropy. This is because their buffer policy are able to remove duplicated bundles. For example, in epidemic with EC, bundles are removed when they have the highest EC value. In epidemic with immunity, redundant bundles are deleted by immunity tables. Therefore, nodes are able to receive and store new bundles, which lead to higher delivery ratios.

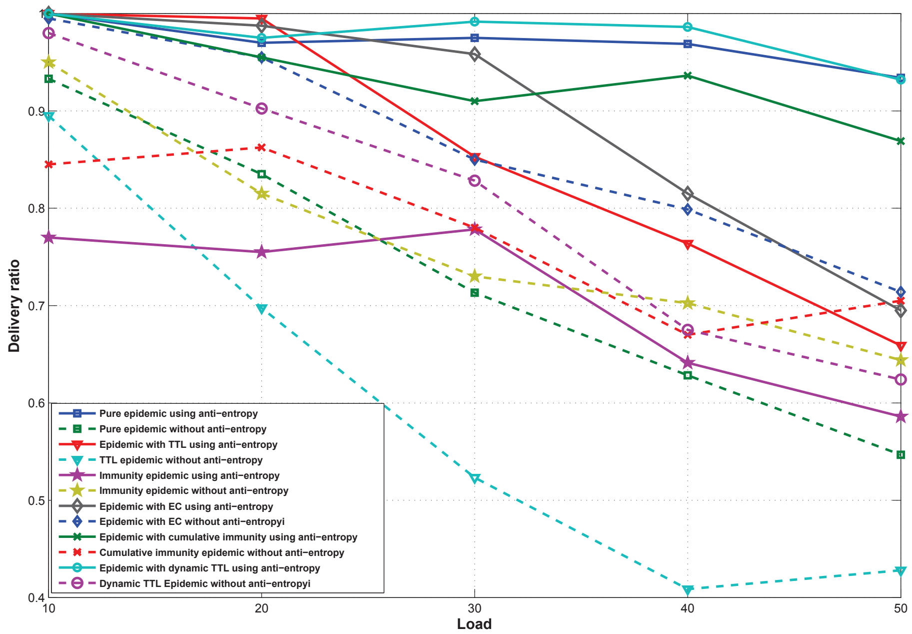

Figure 4.2 Delivery ratio for with and without anti-entropy sessions in trace-file.

4.3.1.2 Should subscribers forward bundles?

The results imply that subscribers forwarding policies have different impacts on routing protocols. Specifically, there are two subscribers forwarding policies, namely, (i) subscribers as relays, in which subscribers store and exchange bundles with any nodes, and (ii) subscribers as sinks, in which subscribers only receive bundles from relay nodes, and cannot exchange bundles.

On one hand, protocols such as pure epidemic, epidemic with EC and epidemic with dynamic TTL are not influenced by any subscriber forwarding policies. Specifically, even though subscribers do not forward bundles, these protocols have 100% delivery ratio in RWP scenario. Note that, the key difference between the two subscribers forwarding policies is the number of opportunities in which nodes are able to exchange bundles. Specifically, if subscribers work as relays, then there are effectively more relays as compared to subscribers working as sinks. In protocols in which subscribers forwarding policies have no impact, the frequencies in which bundles are exchanged have no impact on bundles storage and discard, and hence, suffer less bundle loss. For example, in pure epidemic and epidemic with EC, given that nodes exchange fewer bundles, their buffer has a lower probability of experiencing an overflow, and hence, suffer from fewer bundle drops. Epidemic with dynamic TTL also experiences a high delivery ratio regardless of subscribers forwarding policies for a similar reason. Recall that, epidemic with dynamic TTL sets the TTL value of bundles according to contact intervals. Therefore, bundles have a larger TTL value and are retained by nodes for a long period of time, which played a key role in maintaining high delivery ratio.

On the other hand, four epidemic variants, i.e., epidemic with TTL, epidemic with cumulative immunity, epidemic with EC threshold, epidemic with immunity, have a low delivery ratio when subscribers do not relay bundles. Fig. 4.3 shows their delivery ratio when subscribers are either relays or sinks. In this experiment, the number of relay nodes is equal to the number of subscribers; i.e., 20 relays and subscribers. When subscribers also work as relays, the delivery ratio of these four protocols is 100%. However, when subscribers do not act as a relay, the delivery ratio reduces significantly. In particular, epidemic with immunity has the lowest delivery ratio at 60% when delivering 50 bundles. This is because when subscribers are not relays, there are fewer nodes in the network to help propagate a bundle along. In particular, as there are 40 nodes, getting subscribers to forward bundles means each subscriber node will be able to receive bundles from the other 39 nodes. However, if subscribers are not relays, the larger the multicast group size, the fewer the number of relays, meaning lower delivery ratio because every contact may be between subscribers. The aforementioned problems also exist in epidemic with cumulative immunity. The advantage, however, is that cumulative immunity packets are transmitted less frequently, which provides subscribers with more opportunities to receive bundles. Epidemic with TTL and EC threshold also have reduced delivery ratio. Recall that, when bundles are exchanged, their TTL is set to a new TTL duration. However, when subscribers cannot exchange bundles, the TTL of bundles expire sooner as they are refreshed less frequently. In epidemic with EC threshold, bundles are discarded from buffer when they have a higher EC value than the given EC threshold. This means bundles are less likely to reach the EC threshold when subscribers do not forward bundles. Accordingly, delivery ratio reduces.

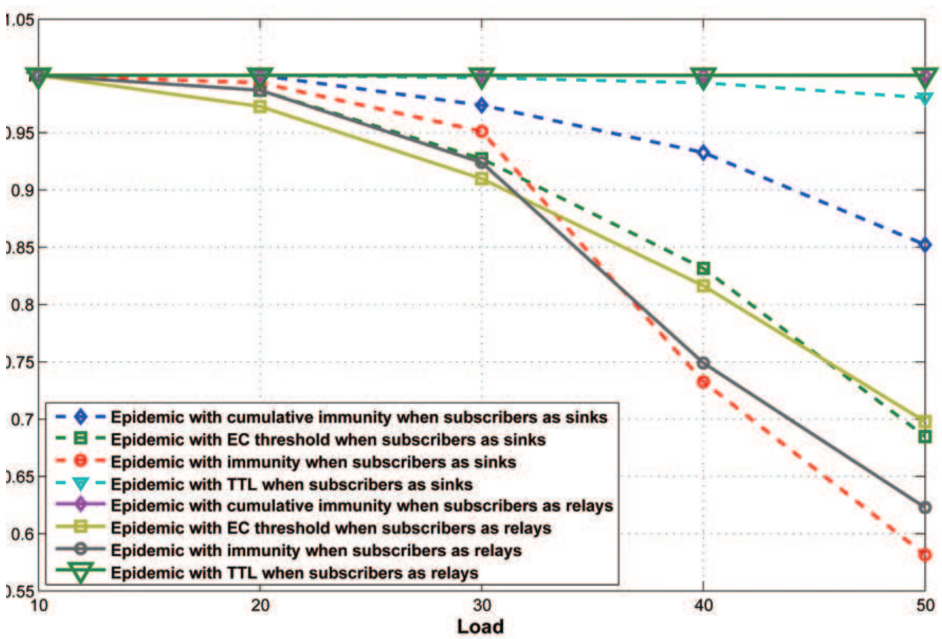

Figure 4.3 Delivery ratio when subscribers are relays vs. sinks in RWP scenario.

Fig. 4.4 shows the influence of subscriber forwarding policies on delivery ratio in the trace-file scenario. There are two key observations. First, subscribers forwarding policies have a different impact depending on the epidemic variant. For example, in epidemic with EC, the delivery ratio is similar regardless of subscriber forwarding policies. The delivery ratio of epidemic with EC is over 80%, whilst in epidemic with TTL, subscribers acting as sinks experience a delivery ratio reduction from 76% to 28%. The main reason for this performance is because there are more nodes, i.e., subscribers, that can help exchange bundles. This increased in exchanges, however, has different impacts on the following protocols. In epidemic with EC, regardless of the number of bundle exchanges, nodes only discard one bundle, which is the bundle with the highest EC value. Thus, the forwarding policy has less impact on epidemic with EC. However, in epidemic with TTL, an increase in the number of exchanges means the TTL value of bundles is refreshed more frequently. Thus, bundles are more likely to be stored in nodes' buffer due to their refreshed TTL value, and thus leading to a higher delivery ratio.

The second observation is that load also has an impact on subscriber forwarding policies. In pure epidemic, when the load is less than 30, subscriber forwarding policies have little influence, whereby the difference between the two policies is less than 10%. However, when load exceeds 30, the gap between the two policies reaches 32%. This is because when subscribers forward bundles, there are more relay nodes. Therefore, the bundles are transmitted faster to all group members. However, if subscribers do no forward bundles, these bundles are stored in nodes' buffer. Thus, when the load increases, there is a high probability that nodes' buffer will overflow, which leads to low delivery ratio.

The influence of multicast group size on delivery ratio is also related to subscriber forwarding policies. This is because changing the multicast group size has a significant impact on performance. Specifically, when subscribers are relays, all nodes collaborate to exchange bundles. On the other hand, if subscribers work as sinks, the bigger the multicast group size, fewer nodes will be able to store and deliver bundles.

4.3.1.3 Influence of Group Size when Subscribers are Relays

In the experiments, except for epidemic with immunity, the delivery ratios of other epidemic variants are not influenced by different multicast group sizes. Fig. 4.5 shows the changes in delivery ratio when epidemic with immunity is used to deliver bundles to different multicast group sizes in the RWP scenario.

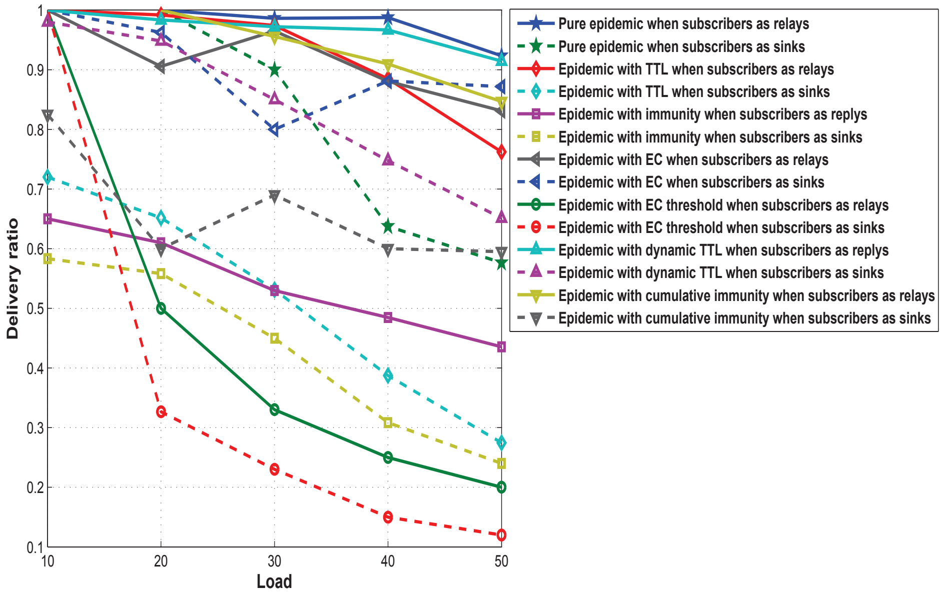

Figure 4.4 Delivery ratio when subscribers are relays vs. sinks in trace-file scenario.

The changed group sizes lead delivery ratio ranging from 43% to 100%. Note that, the figure excludes protocols that are not influenced by multicast group sizes. When the multicast group sizes increase, the delivery ratio of epidemic with immunity reduces accordingly. More specifically, when the multicast group size is 30, the delivery ratio is less than 50%, which is half that when the multicast group size is at 10. This is because a large multicast group size means more nodes will disseminate immunity packets after receiving a bundle, which causes duplicated bundles to be discarded with a higher probability as compared to small multicast group sizes. Therefore, bundles are discarded faster, meaning not all subscribers will have received the bundles, which leads to low delivery ratios.

In the trace-file scenario, when subscribers work as relays except epidemic with immunity, other protocols have similar delivery ratios regardless of their subscriber forwarding policies. As shown in Fig. 4.6, when the group size increases, the delivery ratio of epidemic with immunity reduces. Specifically, when the group size is 30, the lowest delivery ratio of epidemic with immunity is 43%.

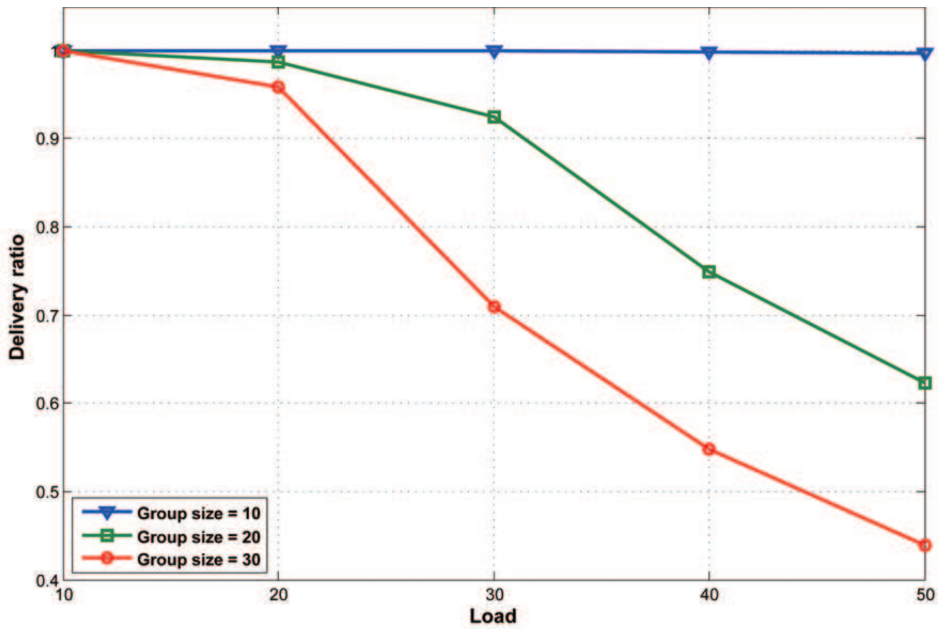

Figure 4.5 Delivery ratio of epidemic with immunity when subscribers are relays in different multicast group sizes in the RWP scenario.

This is because increasing group size leads more nodes to disseminate immunity packets. Therefore, bundles have a higher probability of being deleted before transmitting to all subscribers.

4.3.1.4 Influence of Group Size when Subscribers as Sinks

Multicast group size has a different effect on routing protocols. On one hand, multicast group size has a significant impact on epidemic with TTL, epidemic with immunity, epidemic with EC threshold and epidemic with cumulative immunity. Specifically, all four protocols have low delivery ratios with increasing multicast group sizes. As shown in Fig. 4.7, when the multicast group size is increased from 10 to 30, in epidemic with TTL, the delivery ratio drops from 100% to less than 60%; epidemic with immunity observed a drop in delivery ratio from 99% to 44%; the delivery ratio of epidemic with cumulative immunity drops to less than 60%; and epidemic with EC threshold has the lowest delivery ratio at 44%. The reason for this poor delivery ratio is because large multicast group size reduces the number of relay nodes, and correspondently,

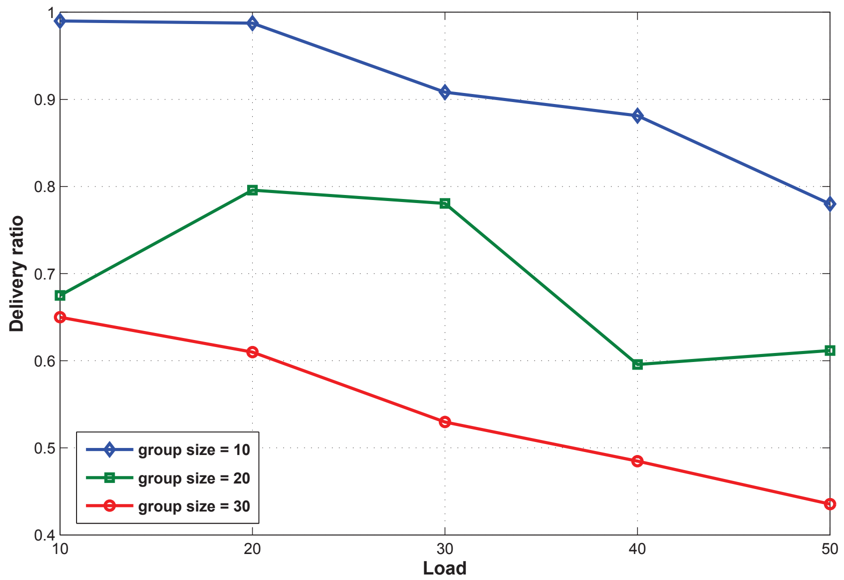

Figure 4.6 Delivery ratio of epidemic with immunity when subscribers are relays in different multicast group sizes in trace-file scenario.

reduces the number of bundle exchanges. Therefore, fewer nodes contribute to the forwarding process. For example, in epidemic with TTL, when the number of nodes is constant, a larger multicast group size implies fewer bundles are exchanged due to fewer relay nodes. Hence, bundles have a higher probability to be discarded due to TTL expiration. In epidemic with EC threshold, fewer bundle exchanges mean the EC value of bundles are likely to be less than the EC threshold, which leads to high buffer occupancy level.

Multicast group size has little impact on the delivery ratio of pure epidemic and epidemic with dynamic TTL. Fig. 4.8 demonstrates the delivery ratio of these two protocols, which only declined slightly in large multicast group sizes and heavy load scenarios; e.g., when multicast group size and load exceed 20 and 40 respectively, the delivery ratio drops only by 10%. This is because these two protocols store bundles until they are delivered. In both pure epidemic and epidemic with dynamic TTL, bundles are stored by relay nodes, and regardless of the multicast group size, all subscribers are able to receive bundles when they encounter relay nodes.

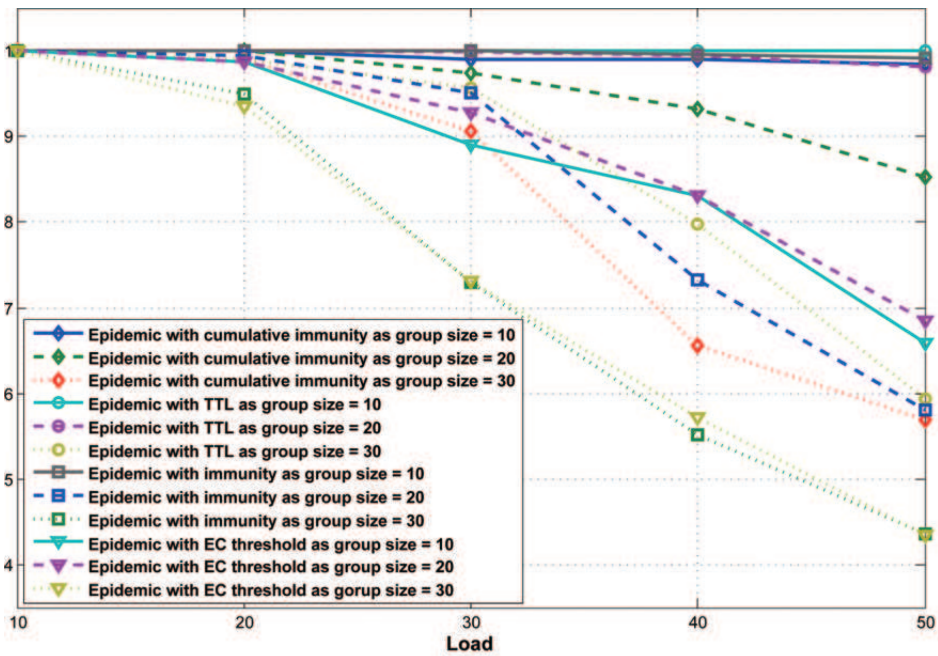

Figure 4.7 Delivery ratio comparison of protocols sensitive to multicast group size changes.

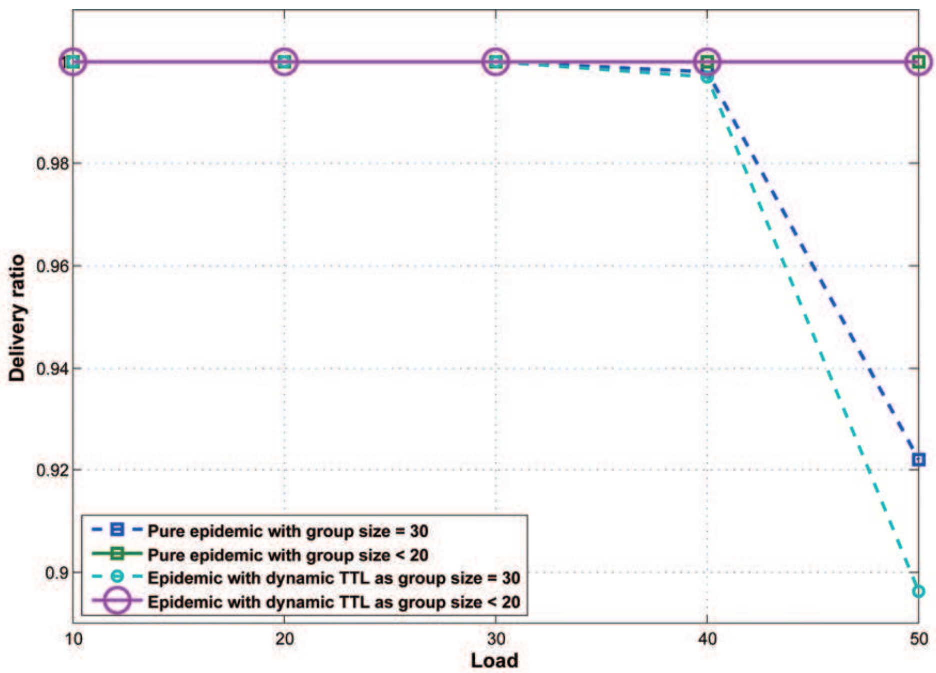

Figure 4.8 Delivery ratio comparison of protocols insensitive to multicast group size changes in the RWP scenario.

Fig. 4.9 and 4.10 demonstrate the delivery ratio of epidemic variants when they have increased group size. Specifically, when subscribers work as sinks in the trace-file scenario, bigger group sizes lead to reduced delivery ratios. For instance, when the group size is increased from 10 to 30, the delivery ratio of epidemic with immunity reduces from 82% to 24%. This is caused by smaller number of reply nodes, which leads to fewer opportunities to exchange bundles. In Fig. 4.10, the difference in delivery ratio for group sizes of 10 and 30 is only 18%. However in epidemic with immunity, when the group size increases from 10 to 30, the difference is over 50%. This is because increased group size reduces bundles dissemination speed, and this has a different impact on routing protocols. For example, in epidemic with TTL, bundles' TTL value is renewed less frequently. Thus, bundles have a higher probability of being deleted. However, in epidemic with EC threshold, as bundles are governed by their EC value, a slower dissemination speed means they are less likely to be discarded, up to a given EC threshold. Conversely at higher speeds, they are more likely to be discarded as their EC value will reach said threshold quicker.

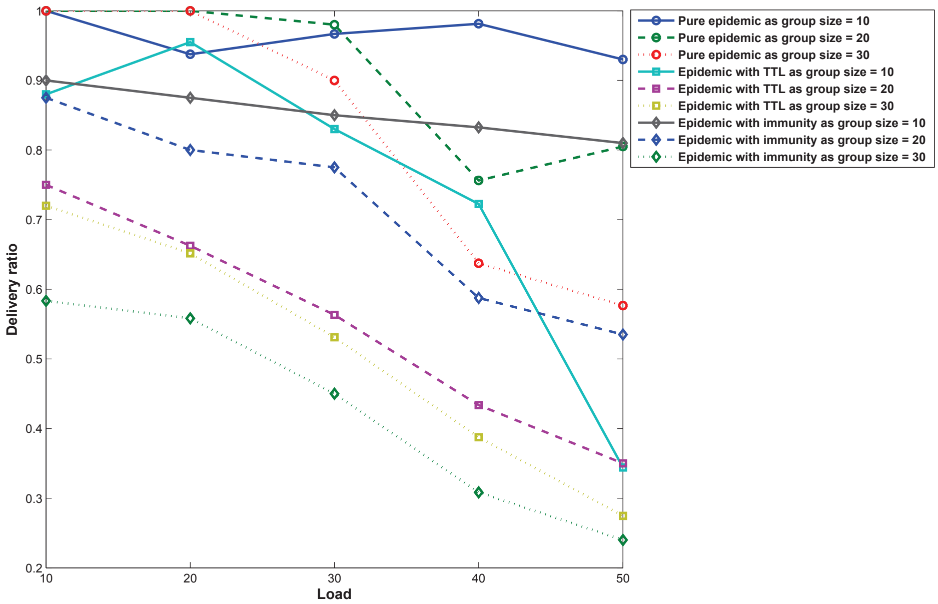

Figure 4.9 Delivery ratio comparison of protocols with increased group size in trace file scenario.

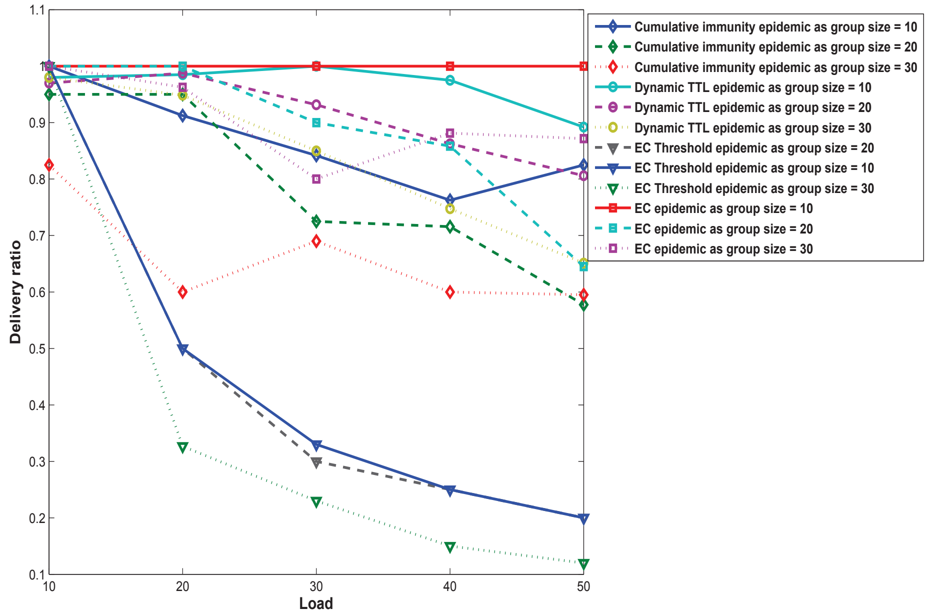

Figure 4.10 Delivery ratio comparison of protocols with increased group size in trace-file scenario.

4.3.2 Delay

This section discusses transmission delay for protocols that have the same delivery ratio in the RWP scenario. For example, as mentioned earlier, pure epidemic, epidemic with EC and epidemic with dynamic TTL have 100% delivery ratio regardless of subscriber forwarding policies. Note that, as all epidemic variants cannot transmit all bundles in the trace-file scenario, no results regarding delays will be presented. Fig. 4.11 shows the delay comparison of pure epidemic in the following scenarios: different multicast group sizes and subscriber forwarding policies. The delay is ranging from 496s to 7,600s. As expected, if subscribers forward bundles, then the average delay reduces. In particular, increasing multicast group size has no influence on delay when subscribers work as relay nodes. However, when subscribers do not forward bundles, smaller multicast group size means less transmission latency. Referring to the figure, when the multicast group size increases from 10 to 20, the delay increases by 50%. For example, at a load of 40, the delay increases from 4000 to 6000 seconds.

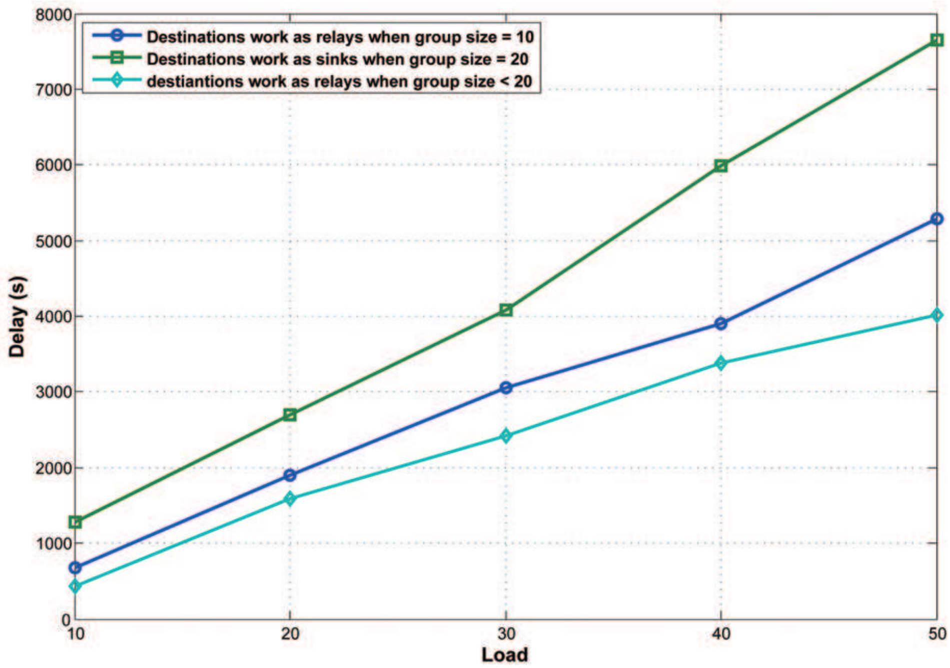

Figure 4.11 Delay comparison of pure epidemic with different multicast group sizes for scenarios where subscribers are sinks and relays.

Fig. 4.12 and 4.13 compare the delay of epidemic with EC with different multicast group sizes and subscriber forwarding policies. As Fig. 4.12 shows, multicast group size has an impact on delays when subscribers do not forward bundles. The influence of multicast group size is also dependent on the load. For example, when transmitting 20 bundles, larger multicast group size means greater delays. However, when transmitting 50 bundles, larger multicast group size implies smaller delay. Moreover, when subscribers forward bundles, the delays are larger than when subscribers act as sinks; see Fig. 4.13. This is because in epidemic with EC, the deletion of bundles is in accordance to their EC value, which in turn is governed by the number of exchanges. If subscribers are sinks, relay nodes are more likely to buffer bundles for a long period of time. Hence, bundles have a higher chance of being forwarded to subscribers. On the other hand, when subscribers are relays, bundles are exchanged more often, and have higher probability of being discarded.

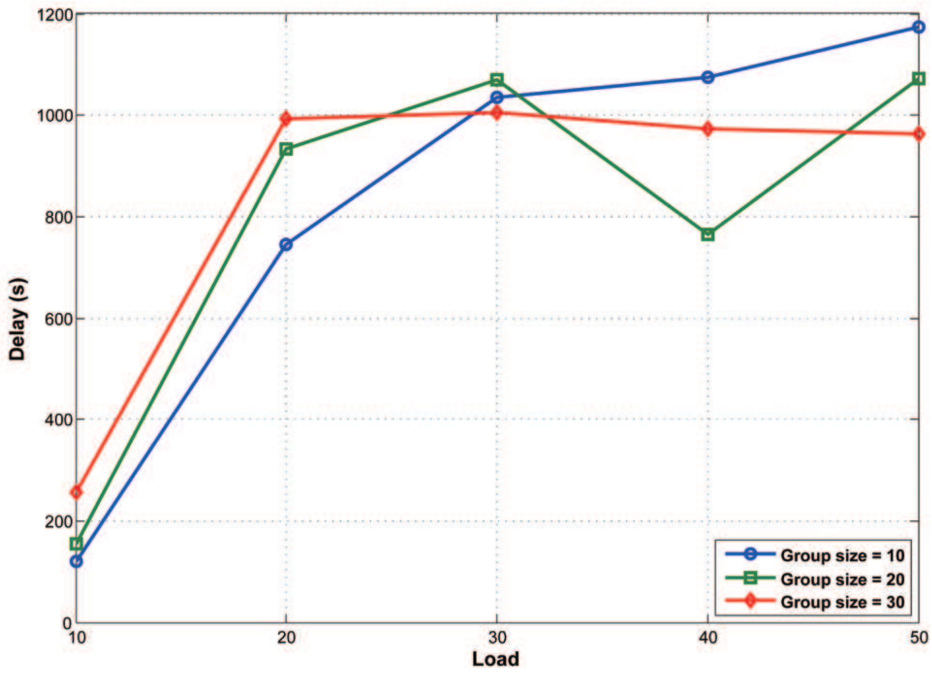

Figure 4.12 Delay comparison of epidemic with EC in different multicast group sizes when subscribers do not forward bundles.

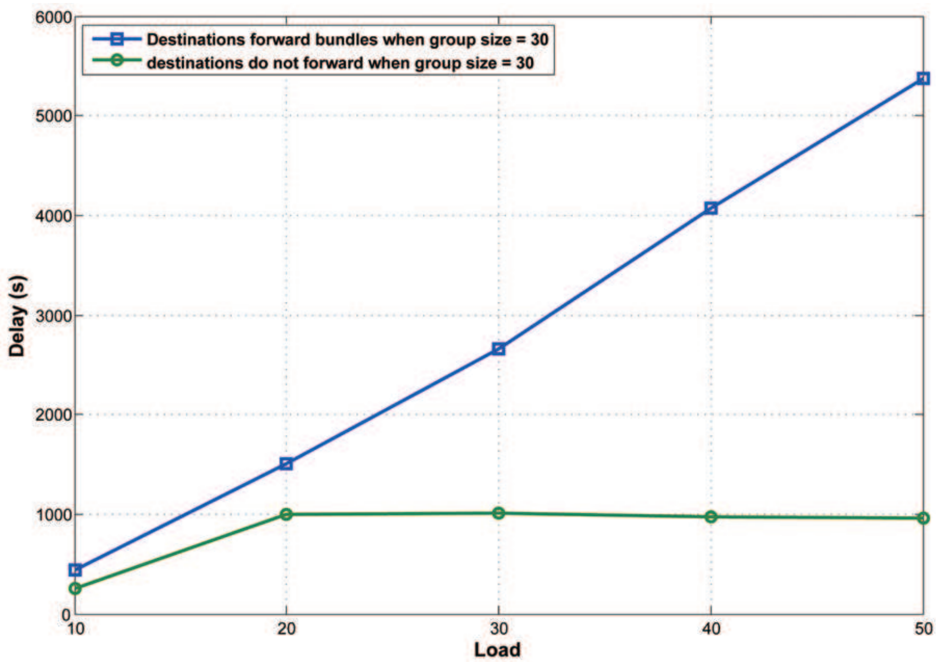

Figure 4.13 Delay comparison of epidemic with EC when subscribers are relays vs. subscribers as sinks when the multicast group size is 30.

4.3.3 Buffer Occupancy Level of Relay Nodes

Fig. 4.14 compares the buffer occupancy level of relay nodes for epidemic variants with and without anti-entropy session in the RWP scenario. Note that, the figure excludes pure epidemic and epidemic with EC because anti-entropy session has no influence on the buffer occupancy level of these two protocols. The figure shows that the nodes' buffer occupancy level in epidemic with cumulative immunity is significantly increased when nodes do not use anti-entropy session. Here, the buffer occupancy level increased from less than 30% to more than 90%. This is because immunity packets are dispatched at a low frequency, where an immunity packet is only generated after a node receives ten bundles. This means before the arrival of an immunity packet, without anti-entropy session, bundles are flooded to all nodes. However, interestingly, after removing anti-entropy session from epidemic with TTL and dynamic TTL, nodes' buffer occupancy level reduces. Specifically, the buffer occupancy level of relay nodes in epidemic with TTL is 0 due to the expiry of bundles' TTL value. Moreover, epidemic with EC threshold and immunity have similar buffer occupancy level before and after removing anti-entropy session. Specifically, the difference in buffer occupancy level when nodes deploy or do not deploy anti-entropy session is less than 10%. This is because anti-entropy session has little effect on their bundle discard policy. Moreover, their buffer policies can discard duplicated bundles. For example, duplicated bundles have higher EC values than other bundles as they are exchanged more than other bundles. In other words, for epidemic with EC, duplicated bundles are discarded due to their high EC value whilst in epidemic with immunity, duplicated bundles are deleted by immunity packets.

Fig. 4.15 shows the buffer occupancy level comparison of relay nodes for epidemic variants with and without anti-entropy in the trace-file scenario. Note that, this figure does not include the results of pure epidemic and epidemic with EC because their buffer occupancy level is at 100% regardless of whether anti-entropy is used. Specifically, without anti-entropy, epidemic with cumulative immunity has the highest buffer occupancy level. This is because removing anti-entropy causes bundles to be flooded and thus congest the network. Consequently, subscribers are unable to receive bundles. Recall that in epidemic with EC threshold, the subscribers dispatch an immunity table after receiving ten bundles. However, because of network congestion, the frequency of immunity tables being propagated reduces, which leads to higher buffer occupancy level.

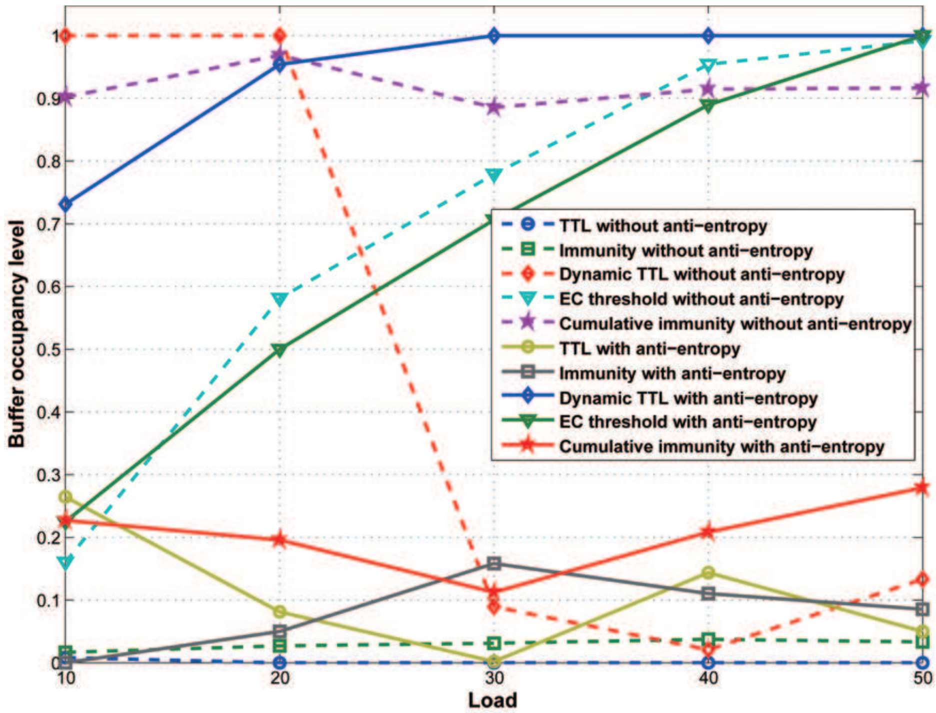

Figure 4.14 Buffer occupancy level of protocols with vs. without anti-entropy session in RWP.

Fig. 4.16 shows the buffer occupancy level of epidemic variants in terms of subscriber forwarding policies. The difference in buffer occupancy level for epidemic with EC, EC threshold, immunity, cumulative immunity, TTL is less than 15%. However, in epidemic with dynamic TTL, the buffer occupancy level is directly related to the forwarding policy used by subscribers. In epidemic with dynamic TTL, if subscribers do not forward bundles, the buffer occupancy level reduces from 95% to 0. The reason is because subscribers cannot exchange bundles with each other. Apart from that, if subscribers are sinks, there will be fewer relay nodes, and accordingly, bundles have less probability to be exchanged and delivered, which lead to the expiration of bundles.

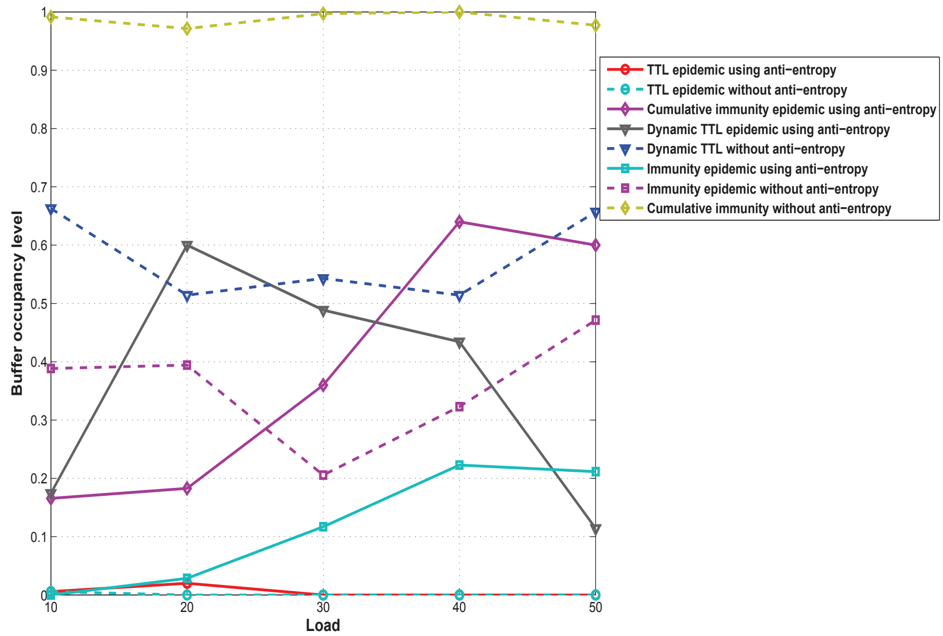

Figure 4.15 Buffer occupancy level of protocols with vs. without anti-entropy session in trace-file scenario.

Fig. 4.17 compares the buffer occupancy level of nodes when subscribers are relays or sinks for the trace-file scenario. Note that, pure epidemic and epidemic with EC have 100% buffer occupancy level regardless the forwarding policies. Therefore, this figure does not include their buffer occupancy level. The results show that if subscribers work as relays, the buffer occupancy level of nodes is higher. This is because bundles can be transmitted more times when subscribers are relays.

4.3.4 Discussion

4.3.4.1 Does Anti-entropy Session Have An Effect On Delivery Ratio?

The answer to this question is dependent upon whether removing anti-entropy session influences the buffering policy of routing protocols. Recall that, the advantage of removing anti-entropy session is that, relay nodes can repeatedly receive bundles that they have previously deleted from their buffer. However,

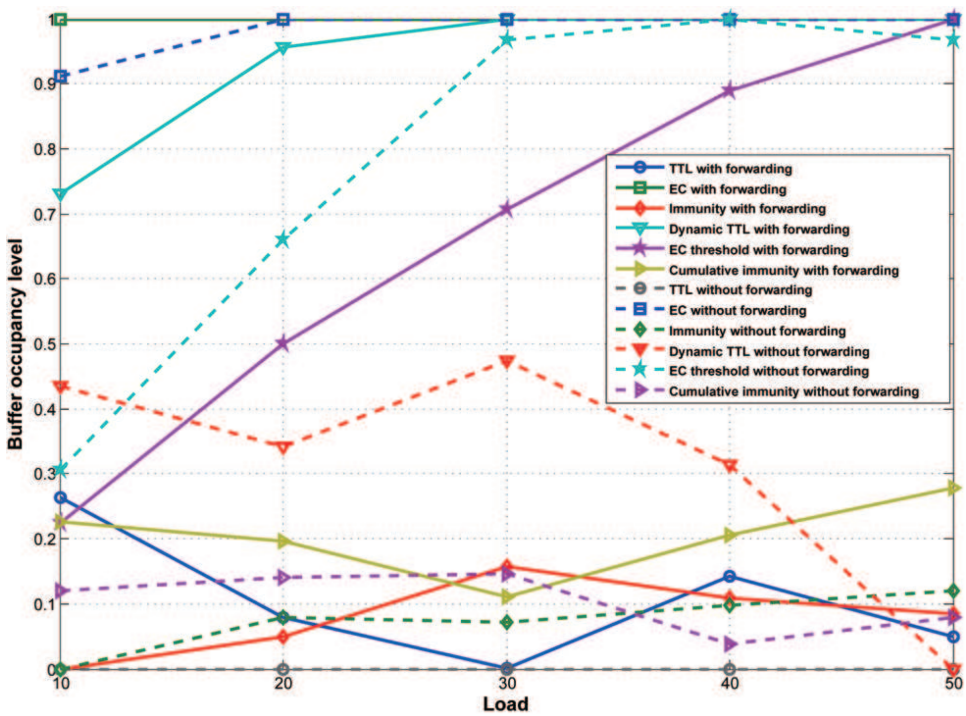

Figure 4.16 Buffer occupancy level of subscribers as relays versus subscribers as sinks in RWP scenario.

the main problem caused is redundant/old bundles occupying nodes buffer and congesting the network, which prevent new bundles from being delivered. On one hand, if protocols use buffer policies that discard redundant bundles, the answer to this question is "no", i.e., removing anti-entropy session has no effect on delivery ratio. For example, there is no impact on the delivery ratio of epidemic with EC, epidemic with dynamic TTL and epidemic with cumulative immunity because duplicated bundles are deleted by their buffer policies. One caveat, however, is that if buffer management policies of protocols cannot cope with redundant bundles, no anti-entropy session means reduced delivery ratio. One example is pure epidemic. In particular, the removal of anti-entropy session reduces its delivery ratio from 100% to less than 80%. Note that, although removing anti-entropy session has no effect on delivery ratio in some protocols such as epidemic with dynamic TTL, it is not recommended to remove anti-entropy session in energy constrained environments; e.g., DTNs comprising of mobile devices. This is because, in the experiments, nodes can repeatedly

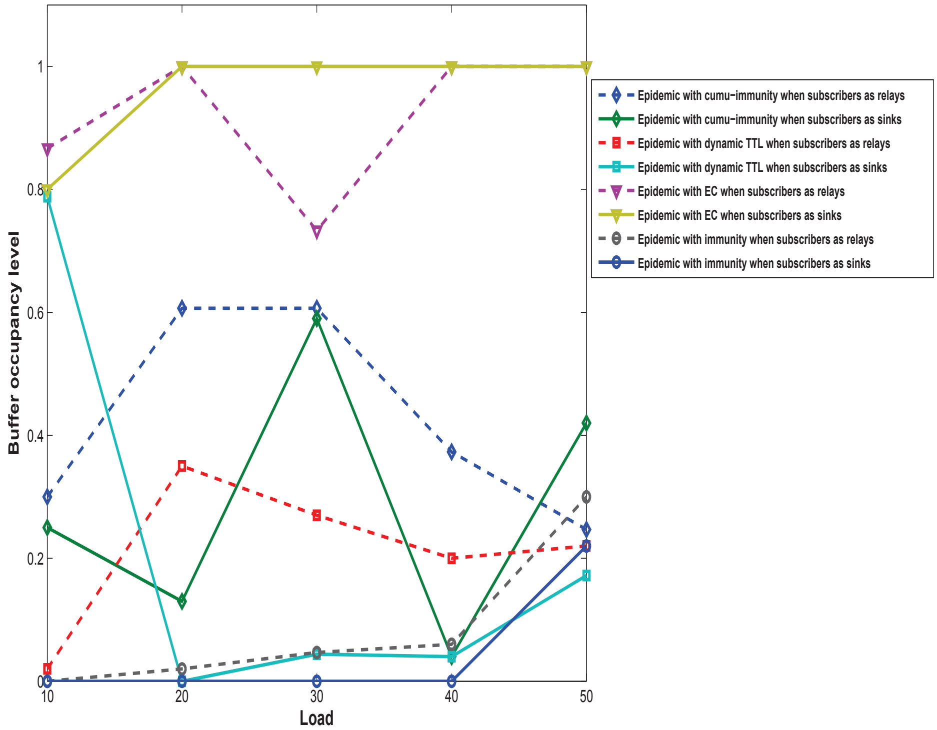

Figure 4.17 Buffer occupancy level of subscribers as relays versus subscribers as sinks in trace-file scenario.

receive and delete the same bundles up to 30 times when they do not use anti-entropy session.

4.3.4.2 Does Multicast Group Size Matter?

First, multicast group size can influence delivery ratio. More protocols have reduced delivery ratio when subscribers act as sinks. Specifically, only one protocol has reduced delivery ratio when subscribers are relays, whilst five protocols have reduced delivery ratio when subscribers are sinks. This is because when subscribers work as sinks, the larger multicast group size means fewer relay nodes can deliver bundles to subscribers, hence, fewer bundles are exchanged, which reduces bundle delivery ratio. Second, because of buffer policies, multicast group size has no impact on delivery ratio. For example, pure epidemic is able to maintain 100% delivery ratio with increasing multicast group sizes. The reason for this is due to nodes not discarding bundles as their buffer is never full, meaning subscribers are able to receive all bundles from relay nodes. If the multicast group size is large, subscribers receive all bundles from a few relay nodes, but experienced higher delays as subscribers are likely to encounter fewer relays.

Group size also has an impact on buffer occupancy level. Specifically, when subscribers are sinks, larger multicast group size means fewer relay nodes. Accordingly, at a given load, fewer relay nodes implies each relay node has to deliver more bundles to subscribers. Moreover, in the experiments, even when the multicast group size and load are small, relay nodes have 100% buffer utilization.

4.3.4.3 How to Reduce the Buffer Occupancy Level of Relay Nodes?

The solution is to simply set an EC quota for each bundle, i.e., each bundle can only be transmitted and stored by k relay nodes, where k is no more than the value of the EC quota. In other words, relay nodes can only exchange a bundle k times.

Nodes record the number of times each bundle has been exchanged, and once bundles' EC value equals the EC quota, they cannot be transmitted amongst relay nodes. Note that, the EC quota only restricts transmissions amongst relay nodes, and do not apply to subscribers. Moreover, EC quota is different to the buffer policy in epidemic with EC threshold. Firstly, EC threshold is used by nodes to discard bundles from their respective buffer, whilst EC quota is a transmission policy that controls which bundle is chosen for delivery. Secondly, EC threshold sets a minimum EC value for each bundle, i.e., a bundle can only be deleted when its EC value exceeds the EC threshold. In contrast, EC quota sets a maximum exchange time for bundles. Additionally, EC quota is not deployed when using epidemic with EC threshold as these two policies conflict with each other. For example, in scenarios where EC quota is five and the value of EC threshold is 10, bundles can only be transmitted five times, meaning they will never be deleted from nodes' buffer since their EC value is less than the EC threshold. Therefore, nodes' buffer are occupied by old bundles, and new bundles cannot be stored and delivered.

The experiments compare protocols performance in terms of delivery ratio and buffer occupancy level of relay nodes with and without EC quota.The EC quota is set to 10 in a 40 nodes network, where there are 20 subscribers. The experiments show protocols that deploy EC quota to have the same delivery ratio. In particular, epidemic with immunity has a higher delivery ratio when it deploys EC quota. As shown in Fig. 4.18, epidemic with immunity has nearly 30% higher delivery ratio, which improved from 62% to 89%.

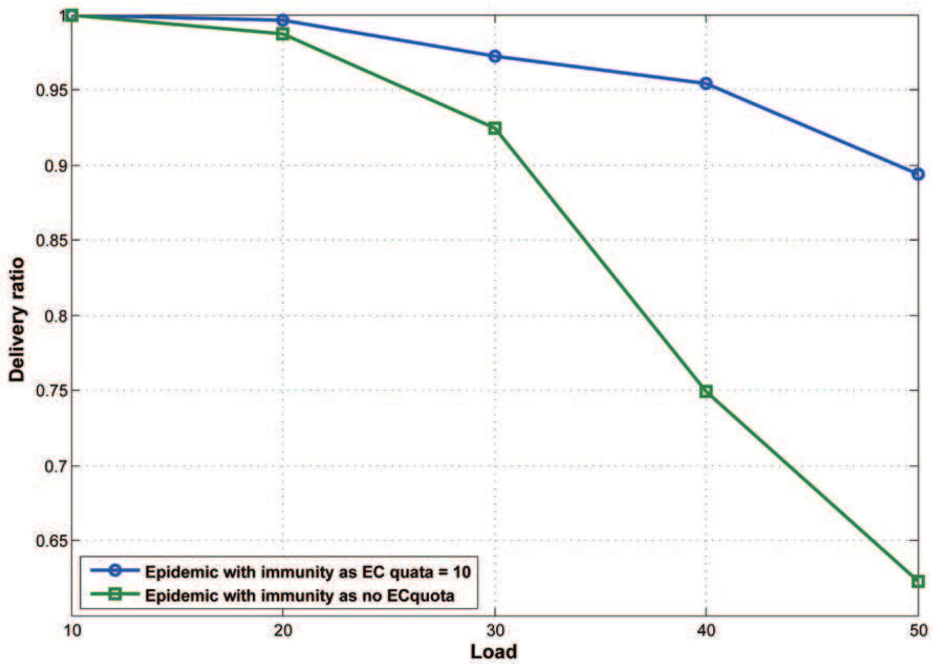

Figure 4.18 Delivery ratio comparison of epidemic with immunity with or without EC quota.

Fig. 4.19 shows that EC quota significantly reduces the buffer occupancy level. Specifically, the buffer occupancy level of epidemic with TTL and epidemic with dynamic TTL reduced from 60% to 0 and from over 74% to less than 15% respectively. This can be explained as follows: due to EC quota, relay nodes reduce the number of bundle exchanges. As a result, bundles delivery relies more on subscribers. Other than that, epidemic with EC also has a significant drop in buffer occupancy level due to EC quota, a reduction from 97% to 12%.

Moreover, epidemic with immunity and cumulative immunity reduce the buffer occupancy level of relay nodes by more than 20%. Note, relay nodes that use pure epidemic still experience a high buffer occupancy level because they do not use any buffer policy.

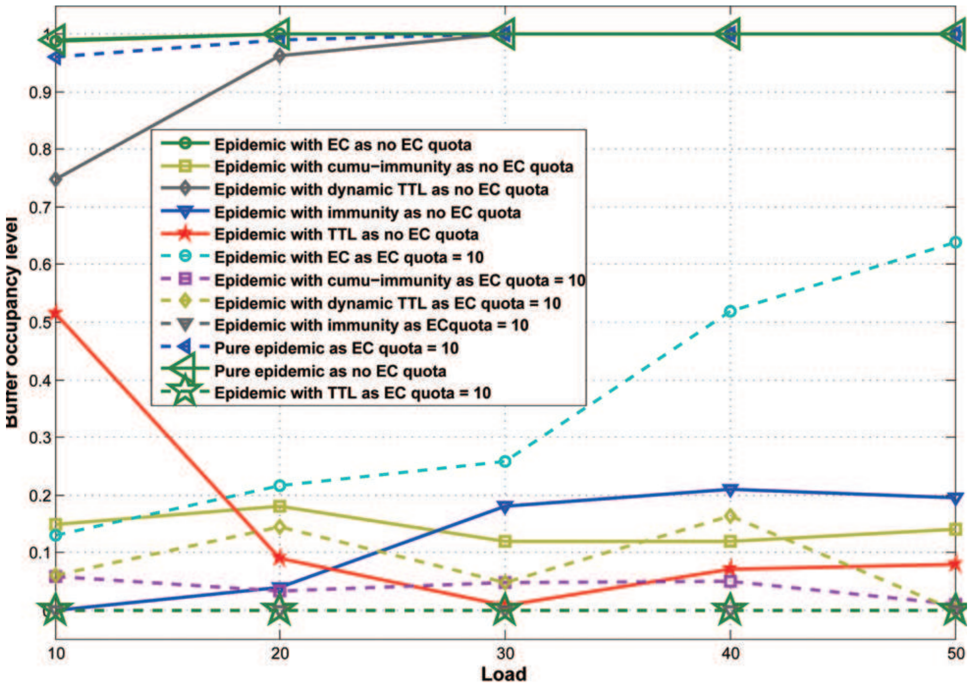

Figure 4.19 Buffer occupancy level comparison of protocols with or without EC quota.

4.4 Conclusion

This chapter has investigated the suitability of using epidemic-based routing protocols to deliver multicast bundles, and evaluated their performance in terms of bundle delivery ratio, delay and buffer occupancy level of relay nodes. Specifically, critical factors that have a significant impact on bundle delivery are studied; namely, anti-entropy session, multicast group size and subscribers' forwarding policies. A key problem with the buffer occupancy level of relay nodes is identified, and propose EC quota, a new mechanism that is shown to reduce the buffer occupancy level of relay nodes significantly and also maintain high delivery ratio.

Chapter 5

Information Retrieval in DTNs

5.1 Introduction

This chapter fouses on Information Retrieval (IR) systems in DTNs. In current IR systems, the data that satisfies a query is assumed to be stored on a single node. Therefore, once a node receives a query in which it has the corresponding data, the query can be resolved completely. However, in scenarios where a query requires data from multiple nodes, these IR systems may fail.

Information retrieval in DTNs consists of three phases, all of which are made difficult by the characteristics of DTNs presented in Chapter 1.1: (i) query transmission, (ii) propagation of source data, and (iii) forwarding of replies to a query node. During the query transmission phase, a querying node transmits a query into a DTN. A routing or query resolution protocol is then responsible for resolving the request/query as quickly and efficiently as possible. In the propagation of source data phase, the aim is to determine a policy that replicates information on source nodes in an efficient manner to facilitate query resolution. Important factors to consider include the number of duplicates or replicas and the buffer occupancy of nodes. Lastly, during the forwarding of replies phase, once a request or query is resolved, the answer must be forwarded back to the querying node. This phase must also consider the factors outlined in phase (ii).

Fig. 5.1 demonstrates the three phases of information retrieval. In phase (i), querying node Q generates a query. The main challenge is to resolve the query quickly and with a high success rate. A strategy here is to propagate the query to as many nodes as possible. However, arbitrary query transmissions may result in unnecessary buffer occupancy. In phase (ii), node N propagates its stored data to node X, Y and Z. This also facilitates query resolution as having more duplicates of the required data ensures a query has a better chance of being resolved. For example, given that the data propagated by node N can answer the query from node Q, meaning after node X, Y and Z receive the duplicated data from node N, all of them can answer the query. However, the problem is that, because nodes have limited buffer capacity, they cannot duplicate indiscriminately all stored data to nodes. Hence, they need to decide which data to replicate whilst ensuring high query resolution. In phase (iii), once node X, Y, Z or N receives the query, it replies to the querying node. In this phase, the main problem is how to transmit the reply to the querying node quickly whilst incurring a low buffer duplication level.

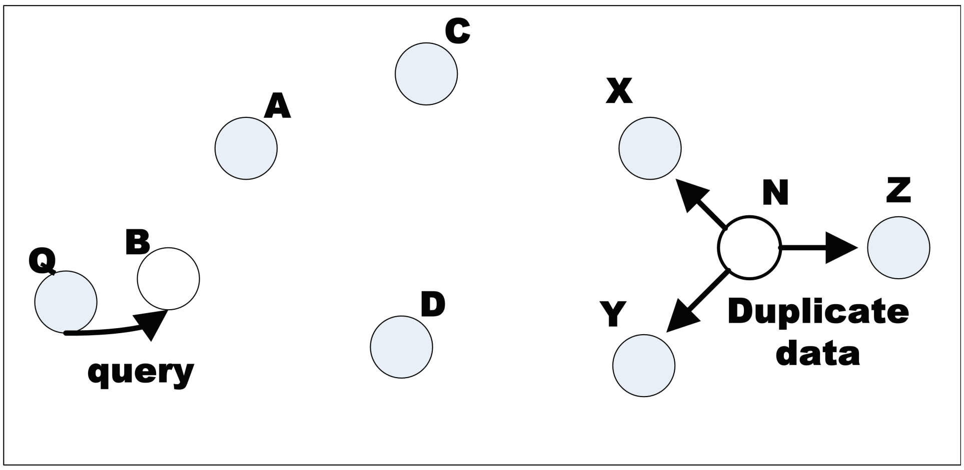

Figure 5.1 An example information retrieval process in DTN.

To date, as described in Section 2.3, only two works have considered information retrieval in DTNs. As both of these two IR systems use PROPHET [61] to transmit replies, these two IR systems will be called PROPHET-IR systems. Note that the problem at hand is ostensibly different from past works on routing, see [29], which primarily deal with address centric forwarding whereby one or more bundles are destined to a given node's address. In these works, the query and stored data are duplicated and transmitted to nodes with a high connectivity to other nodes. These works, however, suffer from several critical problems. First, the transmission of queries and replies are separate to the query resolution process. This may result in a query resolution failure because a node with the highest number of neighbours may not necessarily have the required data to resolve the query. Secondly, all queries are regarded as one-shot single queries, i.e., the query can be resolved only by a single node. However, to resolve other types of queries such as aggregated and complex queries, prior IR systems require access to data from multiple nodes as opposed to one specific node. Thirdly, although past address-centric IR systems avoid flooding by limiting the number of duplicates, there is no buffer policy to remove resolved or staled queries and duplicates.

Henceforth, this chapter proposes Distributed Data-Centric Information Retrieval (DDC-IR), a data centric IR system that supports all query types such as continuous and complex. More importantly, it is designed specifically to operate in DTNs. Moreover, it incorporates a new query and reply packet, aka Query Reply Packet (QRP), that combines both query and one or more replies in order to improve resolution probability and reduce buffer occupancy level. In addition, it uses caching so that nodes store popular queries, which has the effect of speeding up query resolution. This chapter has conducted an extensive simulation study to compare DDC-IR to previous systems using both RWP and trace-file scenarios. The results show that DDC-IR system is able to resolve 50% more complex queries and has 80% lower buffer occupancy level. This chapter also tests the influence of the number of sub-queries on query resolution time. That is, when the number of sub-queries in a complex query increases from five to nine, DDC-IR uses 50% more time to resolve the query. In comparison, prior IR systems are unable to resolve any queries.

The rest of the chapter is organized as follows. Section 5.2 introduces the proposed data-centric IR system. Section 5.3 and 5.4 present the research methodology and the experiments results. Section 5.5 concludes this chapter.

5.2 System Description

In DDC-IR, nodes store both data and queries. Moreover, DDC-IR is *data centric* and has three main phases: (i) query transmission, (ii) propagation of matching data, and (iii) transmission of replies. DDC-IR combines the query transmission and retrieval process. Specifically, when a node receives a query, it inserts any matching data corresponding to each index term into the query. Once a node finds that all index terms have been answered, it forwards the query back to the querying node. The entire retrieval process finishes when a querying node receives a reply in which all index terms have corresponding data. Additionally, to ensure fast query resolution, nodes propagate and cache data according to their frequency or popularity.

The remaining sections will use the network scenario shown in Fig. 5.4 as an example to demonstrate each phase of DDC-IR system over time. There is one querying node Q, and seven other nodes; namely, A to G. All nodes are populated with the corresponding ID Elements (IDEs), explained later, data and index terms shown in Tab. 5.1. At different time epochs, nodes within a circle mean they are within range of one another, and hence are ready to exchange data. Tab. 5.1 also shows each node's IDEs.

5.2.1 Nodes

A node stores: (i) its identification (ID), (ii) data and (iii) received queries. Specifically, different to address-based systems, a node's ID is a tuple of attributes. It refers to these attributes as ID Elements (IDEs). For example, the ID of a portable device in DDC-IR can be <Wollongong, cell phone, University, student, Telecommunication faculty>, which indicates the device is a cell phone located in the Wollongong area and the user is a university student in the Telecommunication faculty. It assumes that nodes with similar IDEs to have a higher probability of encountering one another. In effect, nodes with similar

IDEs are considered to have some relationships with each other and can be considered to be in the same group. This fact has also been used by [65] to improve the delivery ratio of bundles. However, the DDC-IR uses it to ensure queries are propagated to nodes with a high similarity, explained later, of answering a query and also forward replies back to a querying node.

The data stored by a node is associated with a list of index terms. Index terms are words in natural language that describe and represent stored data. For example, if a node has data pertaining to the temperature of a given area, a likely index term would be "temperature". When nodes delete a piece of data, they also remove its corresponding index terms. Note that, index terms may encompass multiple data. For example, the index term "sports" can include all types of sports. Therefore, if a query includes an index term "sport", nodes that have sports related index terms will answer the query. However, the relationship between index terms is beyond the scope off this chapter. Interested readers are referred to [115].

Every node stores past queries in a table called Query History Table (QHT). Specifically, the table is used to record the *popularity* of past index terms. If a node receives a query with the index term (idx), then the corresponding entry in its QHT is increased by one. The structure of QHT is shown in Fig. 5.2. The size of QHT is dependent on a node's buffer capacity. Moreover, when nodes encounter each other, they compare and exchange index terms in their QHT, and also transfer the corresponding matching data for index terms they do not have in their respective QHT. This will be explained further in Section 5.2.2. Apart from that, each index term is assigned a TTL value, which decreases by one after every set time duration, and is discarded once the TTL value reaches zero. Note that, the time duration that controls the speed of TTL decrease can be set to a different value. In this thesis, the TTL value is set to decrease by one after every second. This is elaborated in Section 5.2.4.

Index Term	Matching Data	Popularity Count	TTL

Figure 5.2 QHT structure.

5.2.2 Query+Reply Packet (QRP)

A query is stored in a QRP. As shown in Fig. 5.3, a QRP includes eight parts: (i) query type, (ii) source IDEs, (iii) destination IDEs, (iv) length of the QRP, (v) Query ID (vi) similarity threshold, (vii) query in terms of index terms, and (viii) matching data. Note that, different from address-based systems, a querying node does not specify the address of a node in the "Destination" and "Source" field. Instead, it lists the IDEs of nodes that it wants to process its queries. For example, given a query for the temperature of Wollongong, a querying node will populate the Destination field with "Wollongong". This ensures only nodes that are associated to the Wollongong area reply to the query. Other nodes will ignore the query even though they may have matching data. In this regard, a querying node can use the Similarity Threshold field to allow nodes that partially match the IDEs of the Destination field to process the packet; see next section. It is important to note that it can easily convert this into an addressed based system because an IDE can simply be a node's unique address. Apart from that, each QRP is identified by a Query ID that is 32 or 64 bits in length. The Query ID acts a version control mechanism. This means a QRP with a larger Query ID means it is generated later than those with a smaller Query ID. Hence, it provides a means to remove old QRPs.

5.2.3 QRP Transmission

QRP transmission is based on two processes. The first process checks whether the similarity value of a node exceeds K_1, also called the similarity threshold. Another process checks whether stored data, i.e., index terms, match those in

Table 5.1 IDEs and data of each node in the network scenario shown in Fig. 5.4.

Node	**IDEs**	**Index Terms**
A	Wollongong	Christmas, Party, Gifts, UOW
B	New South Wales, Wollongong	Time, Party, NSW
C	Sydney	Christmas, News, Sports, Clubs
D	Wollongong	Art, Festival, Movies, UOW
E	Wollongong, University	Time, Schedule, Christmas, Party, Gift, Member, UOW
F	Wollongong, ICT, University, Tutor	Telecommunication
G	Wollongong, University, Tutor	Art
Q	Member, ICT, Wollongong, University, Tutor	Class notes

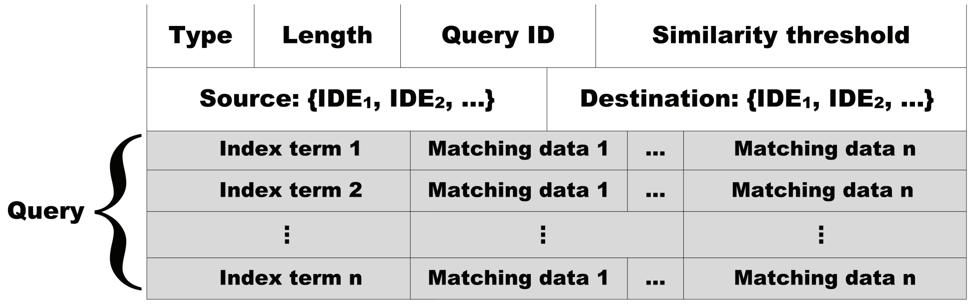

Figure 5.3 Structure of a QRP.

the query.

The resolution of a given query is based on a node having matching IDEs as those contained in the Destination field of a QRP. For example, given a Destination field of "Wollongong University", and a query with the index terms *time, schedule, Christmas, party, and University,* node C will not process the QRP because it does not share the same IDEs. In addition, using the Similarity Threshold field, a querying node is able to control how many of the IDEs in the Destination field are to be matched before a node is allowed to process the query. For instance, a threshold value of 50% means the querying node is happy for a node that matches half of the specified IDEs in the Destination field to process the query. It calls the percentage of nodes' IDEs matching a QRP's Destination field as its similarity value.

If nodes have similar IDEs, meaning a similarity value higher than K_1, they begin the retrieval process. For example, node B's IDEs "New South Wales, Wollongong" match the QRP's Destination field of "Wollongong". Also, node A has one IDE matching the Destination field "Wollongong", therefore, the similarity value of node A is 50%. Additionally, nodes E, F, G and Q have the same IDEs "Wollongong" and "University". Hence, it deems them to have 100% similarity. In other words, these four nodes have a high probability of being associated to the University of Wollongong, and hence, are likely to encounter one another. As a result, these nodes will check their stored data to answer the query. Otherwise, nodes simply ignore the QRP.

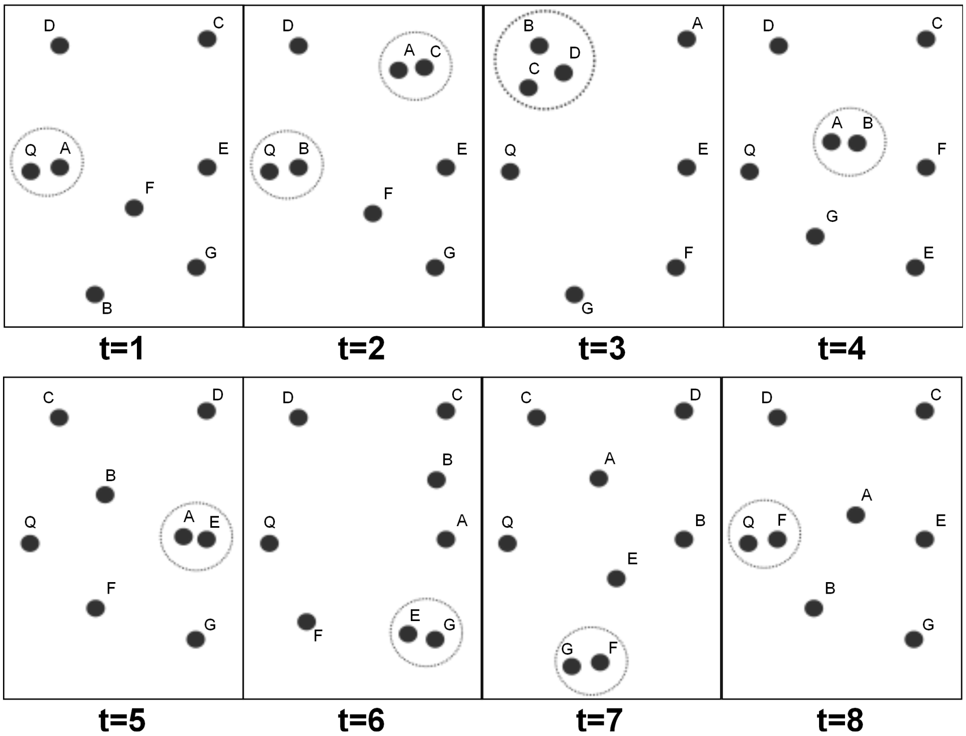

Figure 5.4 Network scenario.

The second aspect of the transmission process that is distinct from past systems is that a QRP contains matching data from past nodes that have processed the QRP. This makes it possible to resolve a query during QRP transmission. Specifically, once all index terms in a QRP have matching data, a QRP effectively becomes a reply packet, meaning any node that receives such a QRP will forward it to a node matching the IDEs in the Source field.

Fig. 5.5 shows nodes resolving a query. As the Destination field of the QRP is "Wollongong", and K_1 is set to 20%, both node A and B have a higher similarity value than K_1, meaning both of them will process the QRP. In particular, at t=1 and t=2, when nodes A and B encounter the querying node Q, they have a 100% similarity value. After receiving the QRP, they include matching data for index terms "Christmas", "party", "UOW" and "time". At this point, the QRP remains unresolved. Thus, nodes A and B store the QRP. At t=3, when node B encounters node D, as node D has a 100% similarity value, it accepts the QRP and checks whether its stored data can answer the query. In this case, it fills the matching data for the index term “UOW”. At t=4, when nodes A and B encounter each other, as the QRP carried by both nodes have the same Query ID and Source field, they are able to consolidate the QRP by synchronizing index terms and matching data. Hence, after the encounter, the QRP has the same set of resolved index terms. The QRP is fully resolved and becomes a reply at t=5, when node A receives the data pertaining to index term “schedule” from node E.

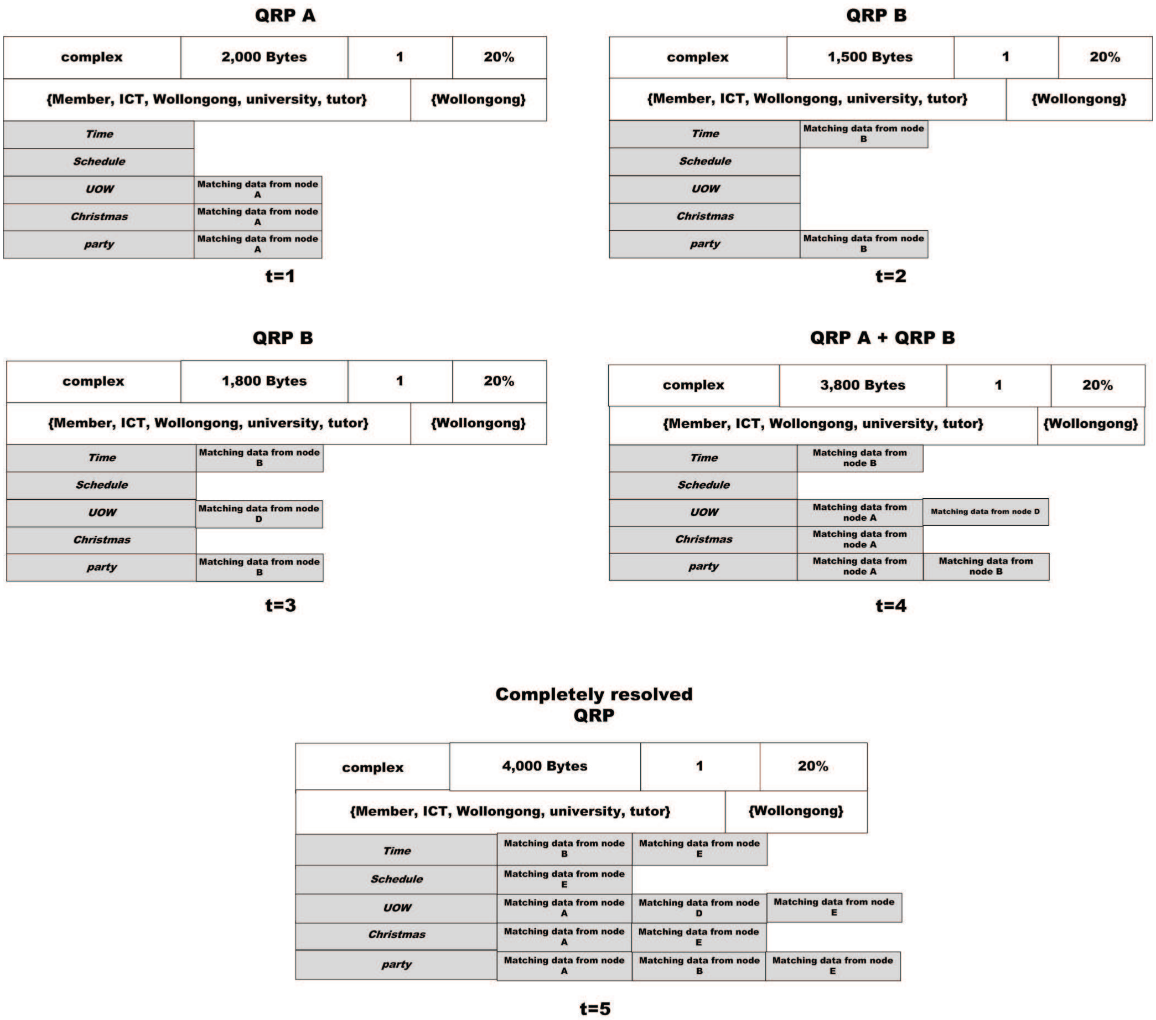

Figure 5.5 Query resolution process.

5.2.4 Reply

Replies are simply a QRP where all index terms have corresponding matching data. Hence, the first node that has a reply packet will be the one that has the last piece of data for a given unfilled index term. Once a node fills in the last missing data, the QRP is considered as resolved and will now be forwarded based on the Source field. In this respect, the reply process is similar to the query transmission process whereby a resolved QRP is only forwarded to nodes matching the IDEs in the source field. This also avoids flooding QRPs throughout the network, and ensures QRPs are processed by nodes that have a high probability of meeting the querying node.

The transmission of QRP from node E to Q is based on the similarity value of a node's IDEs to the Source field. Specifically, the QRP is only transmitted to nodes with a higher similarity value to the IDEs in the Source field. In the example, the Source field has IDEs "Member, ICT, Wollongong, University, tutor", whilst node G's IDEs are "Wollongong, University, tutor". This means the similarity value of node G is 60%. By the same calculation, node F has a similarity value of 80%, and hence will store the reply from node G. Subsequently, the QRP is transmitted from E to G, F and Q respectively.

After a querying node receives a resolved QRP, there are three methods to remove redundant QRPs: (i) timer, (ii) QueryID, and (iii) W-copies. In the first method, once nodes receive a resolved QRP, they will store it for up to a given duration; say TTL_e. After TTL_e time, the QRP is removed. However, an improper TTL_e value may lead to the QRP being removed prematurely. In method (ii), old QRPs are automatically discarded when a querying node issues a new query; i.e., one with a larger Query ID value. When a node receives the new QRP, it proceeds to remove the old QRP from its buffer; note, this is carried out regardless of the resolution status of the old QRP. The last method limits the number of QRP copies. That is, only W QRP copies can exist in the network.

5.2.5 Data Propagation and Caching

In DDC-IR system, nodes are likely to notice that some data are particularly "popular". Here, "popular" means a piece of data or index term has been observed numerous times in past queries. For example, a fire surveillance application operating over a sensor network might want to repeatedly query data from a specific geographical area more times than other areas, meaning nodes will frequently see queries for data from the requested area. Specifically, the index terms corresponding to the data generated from said area appear in queries with a high frequency.

To this end, in DDC-IR, nodes cache and propagate popular data in their QHT. In other words, similar to a web cache, given a cache size, nodes like to maximize cache hits. Upon receiving a query, a node increases the frequency count of the corresponding index term in its QHT. Any new index terms are also added into the QHT. Fig. 5.6 demonstrates how node A updates its QHT, and the process of propagating data according to its QHT. It assumes each QHT can store five index terms and corresponding data. Recall that in a node's QHT, each index term has a corresponding TTL value, which is decremented every **t** time, and re-initialize to TTL_0 whenever a node encounters a query with the same index terms. Once the TTL value of an index term decrements to zero, the index term and corresponding data is removed from a node's QHT. In this example, the initial value for each TTL is set to one hour. When two nodes encounter one another, the frequency of common index terms is updated to the higher value and the TTL of these index terms are reset to one hour. For example, at t=4, when nodes A and B meet, both nodes have the index terms "UOW" and "party". Therefore, they increase their frequency by one. In other words, the frequency of the index term "UOW" becomes 16 and "party" has a frequency of 10. Index terms with expired TTL are removed from the QHT. For this reason, in this example, the index term "Christmas" is removed from node A's QHT at t=6.

In each encounter, nodes then synchronize their QHT in the following manner.

Consider an encounter between node A and B with the following index terms and corresponding frequency: A $\rightarrow$ (idx_1, 5), (idx_2, 2), (idx_3, 10), B $\rightarrow$ (idx_1, 1), (idx_3, 3), (idx_4, 6). Moreover, assume each node can only hold up to *three* index terms in its QHT. In the first step, both nodes will update the frequency value of index terms they have in common. In this case, the frequency value of idx_1 and idx_3 will be updated to five and 10 respectively. The next step is to consider index terms they do not share; namely, idx_2 and idx_4. Given that idx_4 has a higher frequency than that of idx_2, idx_4 will replace idx_2 in node A's QHT. Consequently, both A and B's QHT becomes (idx_3, 10), (idx_4, 6), (idx_1, 5). This means after nodes synchronize their QHT, they share the same popular index terms.

5.2.6 Discussion

The query transmission process has three advantages as compared to past IR systems. First, it incurs a lower buffer occupancy level. This is because queries are only transmitted to nodes with a given similarity value threshold. This means queries are not flooded. The low buffer occupancy level is also due to the TTL mechanism that nodes used to remove redundant and queries. In contrast, current systems such as [33] [28] rely on duplicating queries and cannot cope with redundant and staled queries, which in turn leads to high buffer occupancy level.

Second, the query transmission process has a shorter delay. This is because of the caching and data replication strategy specified in Section 5.2.5. However, in prior IR systems, query transmission is separate to the retrieval process. Thus, data is cached arbitrarily and stored when nodes encounter one another. If a query resolution requires data which has a low duplication rate, the resolution will have a long retrieval delay.

Third, nodes have the option of choosing less popular IDEs, and hence, they are able to determine whether to participate in query processing. This is advantages as nodes have the option to opt out of processing queries. Similarly, a querying node can adjust the threshold K_1 to determine how many nodes will process a

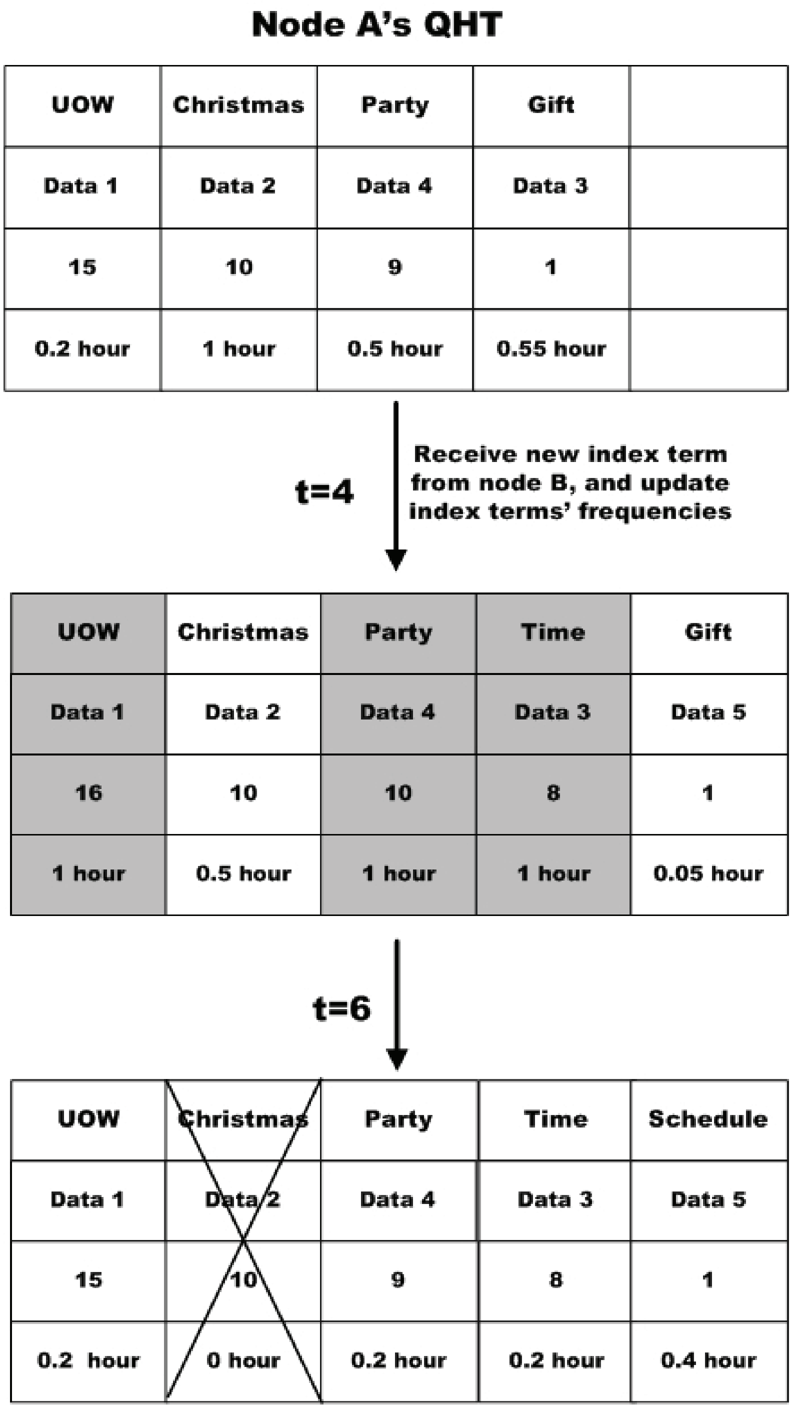

Figure 5.6 QHT updating and propagation.

given query. This flexibility allows a node to restrict its query to a few nodes if the required data is known to be located at multiple nodes and hence a network flood is not necessary.

Lastly, DDC-IR supports all types of queries. Table 5.2 demonstrates how DDC-IR can be used to resolve different queries. For example, complex queries require a conjunctive reply that can answer all indexes in the QRP, whilst a unique query requires data from a node that match all IDEs in the Destination field. In addition, to fulfill a continuous query, the QRP has two extra fields: (i) time duration, which is used to indicate the time constraint of a query, and (ii) time label, which is used to indicate the current time. These two fields thus help a node determine the time interval in which a data is valid. Lastly, nodes manage their QHT differently depending on query types. In particular, for a complex query, nodes store the complete query and corresponding answer in their QHT. On the other hand, for unique queries, nodes do not store any replies; i.e., data. For continuous queries, nodes only store *valid* data, as determined by the time duration and label field.

5.3 Research Methodology

The experiments are based on the trace file collected by Scott et al. [116] and RWP model [30]. Simulation studies are then conducted using these mobility models. There are w nodes chosen randomly as the querying nodes. Each querying node generates m different queries. The value of w and m are varied after each experiment. For example, in some experiments, see Section 5.4, w is set to one and m is increased from one to 10 in each experiment. Another example is to evaluate the influence of w. Here, m is set to three, and w is varied from one to 10.

The simulator is also modified to support all query types, as explained in Section 5.2. Each query has k index terms, where k is initialized to five. After each experiment, the value of k is increased by one. The maximum value of k is set to nine. Each node in the experiment initially has five pieces of data/index terms. Moreover, every node has a buffer that can maximally store five other pieces of data/index terms and corresponding data from other nodes. That is, every node has a QHT that stores the five most popular index terms and corresponding data. The TTL value of each index term in QHT is set to 1000 seconds.

For comparison purposes, the simulator is also modified to support PROPHET IR system used by both [33] and [28], which includes transmission of queries and replies, and data caching. Specifically, nodes in PROPHET-IR employ the First-In-First-Out (FIFO) buffer policy. Whilst in DDC-IR, nodes will remove index terms with the lowest popularity. Every query can only be transmitted and stored by L=5 nodes. In other words, the query can be transmitted up to L-hops. When the query is resolved, a reply is transmitted according to the PROPHET protocol [61]. Briefly, in PROPHET, every node records their encounter times with other nodes. This frequency or probability of encounters is then used to decide next-hop forwarding nodes. In the simulation, when two nodes encounter one another, they record each other's node ID and increase the corresponding frequency of encounters. Replies are then forwarded to nodes that have a high encounter frequency with the corresponding querying node.

Every experiment is comprised of 10 simulation runs. The querying node is also changed after each run. Note that, the maximum recorded time from the trace file is 524,162s. This means if the simulation exceeds this time, the querying nodes may not have a chance to receive all resolved queries. In this case, the retrieval delay of the unsolved query is not recorded.

The following metrics are used to compare DDC-IR and PROPHET-IR:

- **Retrieval Success Ratio** - the number of queries resolved successfully.
- **Buffer occupancy level** - the average buffer or QHT utilization of all nodes.
- **Retrieval Delay** - the time taken to resolve a query.

5.4 Results

This section first presents a comparison of DDC-IR and PROPHET-IR in terms of retrieval success ratio, buffer occupancy level and retrieval delay. Then Section 5.4.4 considers other factors, such as query length and number of querying nodes. Lastly, Section 5.4.5 discusses and summarize the results.

5.4.1 Retrieval Success Ratio

Fig. 5.7 shows that, as nodes move according to the RWP model, all four types of queries have higher retrieval success ratios when using DCC-IR as compared to PHOPHET-IR. Specifically, the retrieval success ratio of PHOPHET-IR is less than 60%. This is because in DCC-IR, the QRP is stored by nodes that have data for the corresponding index terms specified in a query. Whilst, in PROPHET-IR, the transmission of QRP is limited to L hops or copies. Given that query transmission is separate to the retrieval process, nodes receiving a QRP may not be able to resolve a query. As a result, PROPHET-IR experiences a lower retrieval success ratio. For example, continuous queries have a better retrieval success ratio than in PROPHET-IR. This is because in PROPHET-IR, all queries are one-shot queries. This means in order to receive replies continuously, querying nodes need to transmit queries to source nodes continuously. However, given that the query transmission and retrieval process is separate in PROPHET-IR, source nodes have a low probability of receiving all queries from a querying node. This leads to a low retrieval success ratio. However, in DDC-IR, nodes transmit queries to nodes with a high similarity value. In other words, nodes are likely to encounter and transmit the query to source nodes. Moreover, in DDC-IR, with the help of time duration field, source nodes are able to reply continuously to a querying node without having to receive additional QRPs.

Different query types lead to varying retrieval success ratios. For example, continuous queries have the highest retrieval success ratio. Interestingly, when the number of queries, i.e., the value of m is not higher than four, DDC-IR is able to reach 100% retrieval success ratio, which is 5-10% higher than other query types. In PHOPHET-IR, continuous query reaches a 60% retrieval success ratio, which is 20% higher than other types of queries. Continuous queries have the highest retrieval success ratio. This is due to the operation of IR systems in DTNs. Specifically, if a querying node receives a reply that satisfies all requirements in the query, the retrieval process is successful. Take continuous query as an example. If a querying node receives a reply that completely resolved all index terms and satisfies the given time label, the retrieval is regarded as successful. That is, given that source nodes continuously disseminate replies, if a querying node receives any one of these replies, the retrieval is considered successful.

In contrast, unique queries have the lowest retrieval success ratio for all scenarios. Specifically, in PROPHET-IR, when m is over seven, the retrieval success ratio reduces to zero. In DDC-IR, the retrieval success ratio reduces to 63%. This is because data required to fulfill a unique query is not duplicated to other nodes.

Fig. 5.8 demonstrates the retrieval success ratio of PROPHET-IR and DDC-IR in trace-file experiments. The figure shows DCC-IR has a higher retrieval success ratio for all four query types. Continuous query has the highest retrieval success ratio, which exceeds 40%. All four query types can be resolved in DCC-IR with at least 58% retrieval success ratio. Note, the figure does not include the retrieval success ratio of aggregation, unique and complex query types for PHOPHET-IR because it fails to resolve queries, i.e., the retrieval success ratio is zero. In trace-file experiments, nodes have fewer encounter times, and hence, they have limited opportunities to exchange stored index terms. The only type of queries that can be resolved by PHOPHET-IR is continuous query. This is because compared to other queries, the source nodes continuously disseminate replies, and all these replies are stored in QHT. Therefore, nodes are able to use data from their respective QHT to resolve a continuous query.

In DDC-IR, unique queries have the lowest retrieval success ratio at 58%. The retrieval success ratio of unique queries decreases from 98% to 58% with increas-

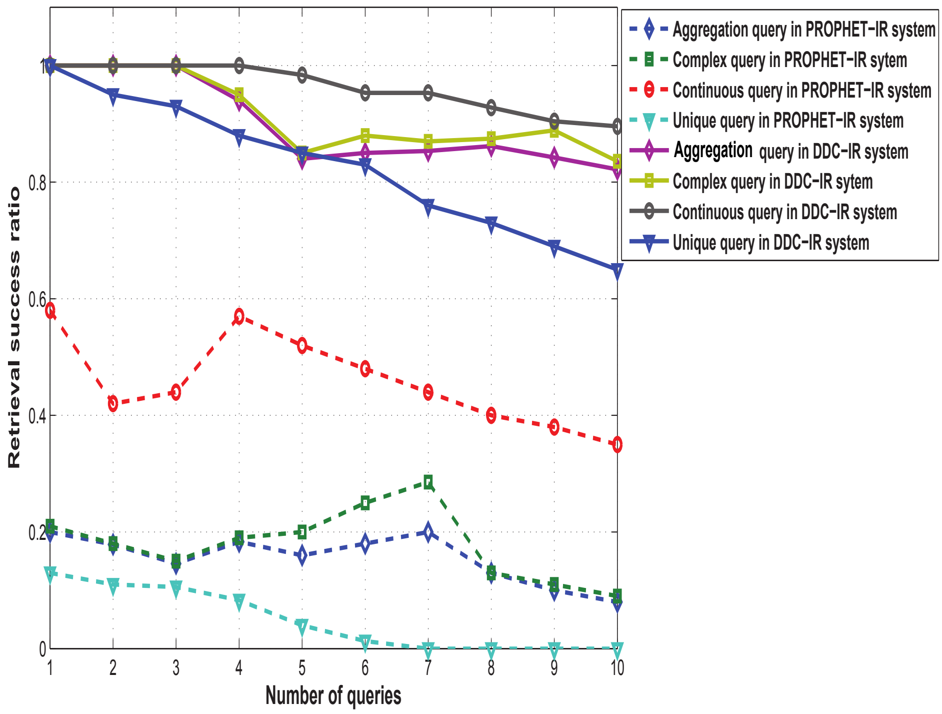

Figure 5.7 Retrieval success ratio between DCC-IR and PHOPHET-IR for the RWP model.

ing m values. The low retrieval success ratio of unique queries in the trace-file scenario is caused by two reasons: (i) no duplicated data, and (ii) fewer encounter times. On one hand, data required to resolve a query is only stored in one node, meaning unique queries have the lowest retrieval success ratio among all query types. On the other hand, nodes that move according to the trace-file have fewer encounter times, and hence, the probability of encountering one specific node becomes lower.

Aggregation and complex queries also experience a declined in retrieval success ratio, from 100% to 70% and 73% respectively when the number of queries increases. However, interestingly, increasing the number of queries can lead to an increase in retrieval success ratio. For example, when the number of queries increases from five to seven, the retrieval success ratio of complex query and aggregation query increases from 80% to 85.71% and to 81.46% respectively.

This is because more queries mean nodes need more encounters to resolve index terms. The increase in retrieval success ratio is due to a higher probability of answering a query from a node's QHT. In other words, higher number of queries means more nodes are likely to cache the corresponding data for a query, which facilitates query resolution.

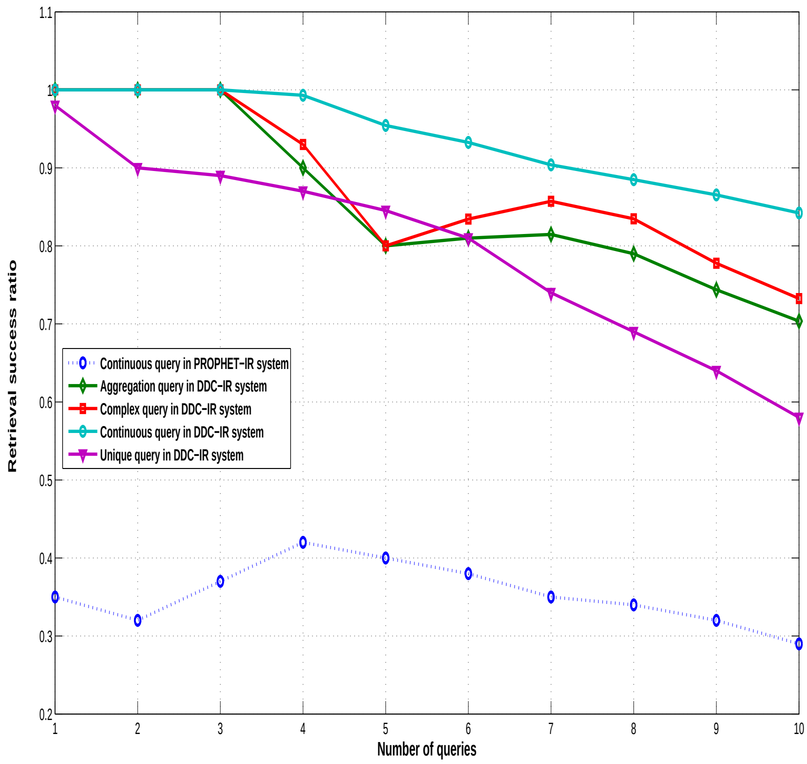

Figure 5.8 Retrieval success ratio comparison between DDC-IR and PHOPHET-IR for trace file study.

5.4.2 Retrieval Delay

Fig. 5.9 shows the retrieval delay for PROPHET-IR and DDC-IR in RWP experiments. The figure shows that all types of queries have a lower retrieval delay in DDC-IR as compared to PHOPHET-IR. For example, in DDC-IR, the retrieval delay of unique queries is less than 400s. However, in PHOPHET-IR, resolving a unique query can take over 900s. The reason for DDC-IR's performance is due to the use of QHT, whereby popular index terms are propagated and cache by other nodes. In the unique queries case, where nodes do not cache data, DDC-IR system also has a shorter retrieval delay. This is because the queries are transmitted to nodes with a higher similarity value to the source node. In other words, these nodes have a higher probability of encountering the source node. In turn, this increases the probability to resolve a query and reduces retrieval delay. Moreover, in both PROPHET-IR and DDC-IR, increasing the number of queries causes higher retrieval delays. For example, in PROPHET-IR, when the number of queries is increased from one to 10, the retrieval delay of aggregation query increases from 259s to 582s. On the other hand, for DDC-IR, the increase in retrieval delay is only from 163s to 347.3s. This is because nodes have random encounters, and thereby, increasing the number of queries means more queries needs to be transmitted, and this leads to increased delay.

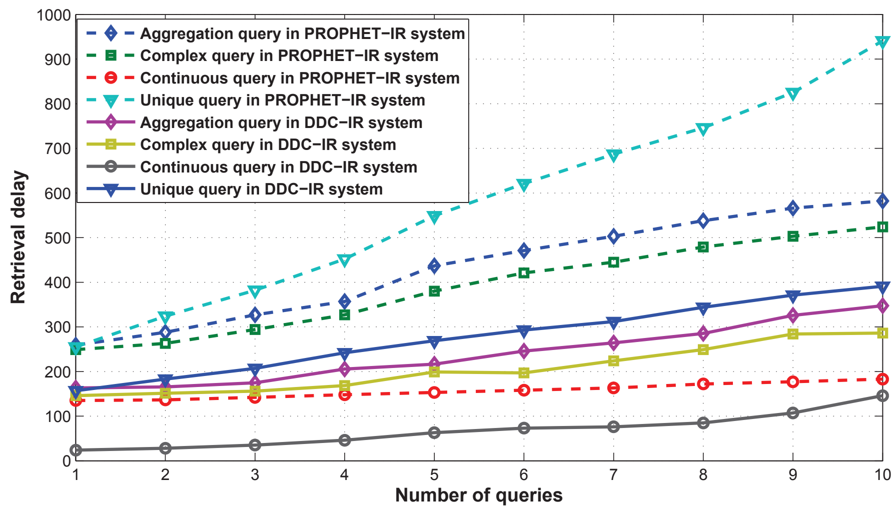

Figure 5.9 Retrieval delay comparison of four types of queries in past and DDC-IR system in RWP.

Fig. 5.10 compares the retrieval delay of four query types in PROPHET-IR and DDC-IR for trace file experiments. Again, as PROPHET-IR system fails to retrieve aggregation, complex and unique queries, the retrieval delay of these three query types are not included in this figure. The figure demonstrates that when increasing the number of continuous query from five to seven, the retrieval delay decreases from 800s to 700s. The same situation also happens with complex queries, whereby the retrieval delay decreases from 917s to 836s. When nodes store the corresponding data for the index terms of continuous and complex queries, retrieval delay decreases. Moreover, the retrieval delay of continuous query using PHOPHET-IR is much higher than DDC-IR. Specifically, to transmit from one to 10 continuous queries in PHOPHET-IR, the retrieval delay of continuous query increases from 700s to 1,333s, whilst the retrieval delay in DDC-IR is less than 800s. This is because of cached data in QHT of nodes that can be used to resolve queries, which in turn reduces retrieval delay. For example, in continuous and complex queries, all data stored in the QHT of nodes can be used repeatedly to resolve queries.

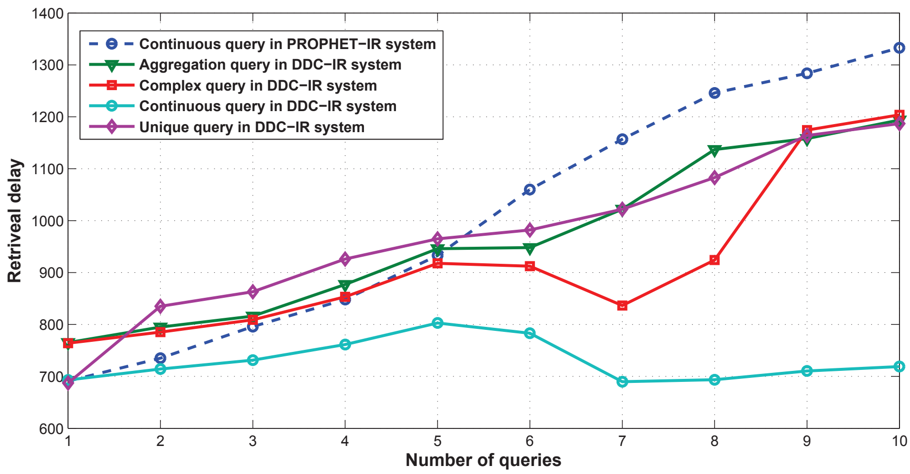

Figure 5.10 Retrieval delay comparison of four types of queries in past and DDC-IR system in trace file.

5.4.3 Buffer occupancy level

Fig. 5.11 shows the buffer or QHT occupancy level of nodes for DDC-IR when the number of queries increases from one to 10 in RWP scenarios. The figure shows that, except for unique queries, when the number of queries exceeds six, the buffer occupancy level of nodes for all query types is over 50%. When the number of queries exceeds nine, all query types except for unique reach 100% buffer occupancy level. This is because when there are more queries, more index terms and data will be stored in QHT. In DDC-IR with aggregation query, when the number of popular index terms exceeds six, the QHT is full; i.e., 100% buffer occupancy level. However, before the number of queries exceeds six, buffer occupancy level is less than 85%. This is because all four query types have high retrieval success ratio when the number of queries is less than six. Therefore, those resolved queries are deleted from nodes' buffer. Note that, the figure only include results for DDC-IR because nodes that use PROPHET-IR have 100% buffer occupancy level regardless of the number of queries. In PROPHET-IR, each node has five pieces of data and a buffer that can store five other pieces of data. Hence, when nodes encounter one another, they will exchange all stored data, and thereby, achieve 100% buffer occupancy.

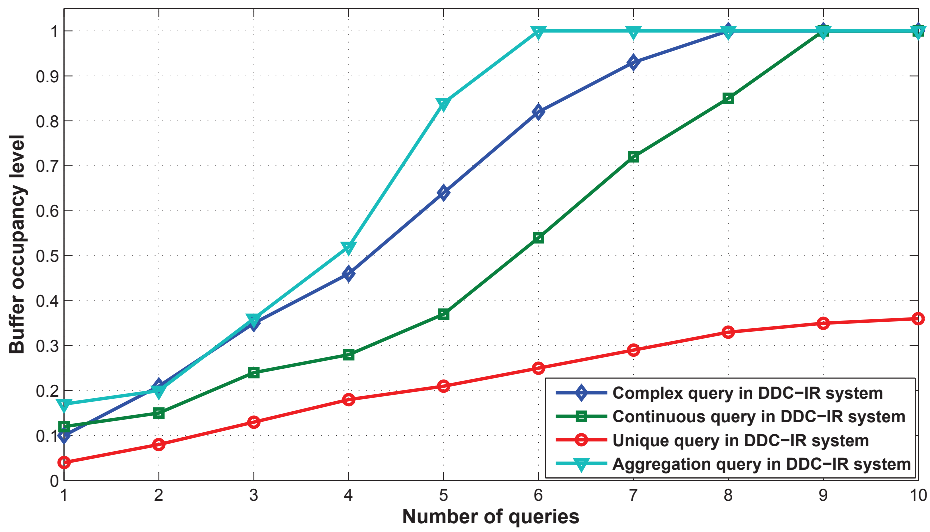

Figure 5.11 Buffer occupancy level comparison of all query types in DDC-IR for the RWP model.

Fig. 5.12 shows the buffer occupancy level comparison of DDC-IR with increasing number of queries for trace-file scenarios. In unique queries, as nodes do not cache data, therefore, it has the lowest buffer occupancy level. The buffer consumption of unique queries is caused by nodes transmitting replies to querying node. Aggregated queries have the highest buffer consumption because the data stored in QHT are from multiple nodes, whose data is to be summarized. This implies that, all nodes in a designated area need to receive the same query until it is resolved. Whilst in other query types such as complex and continuous, once the query is completely resolved, the query is not transmitted any more. Therefore, these queries have a comparatively lower buffer occupancy level.

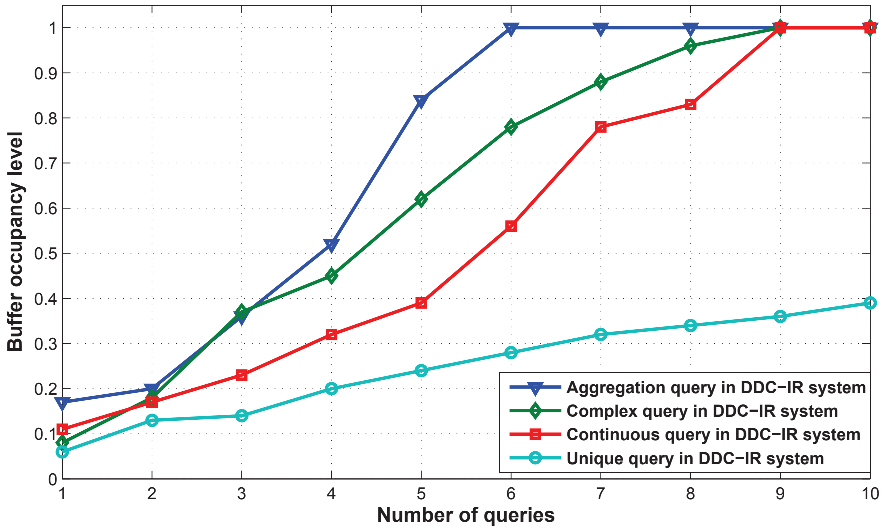

Figure 5.12 Buffer occupancy level comparison of all queries in DDC-IR for the trace file model.

5.4.4 Other factors

The influence of k and w are also tested, which represent the number of index terms in a query and the number of querying nodes respectively. This is important because in an IR system, k and w have an impact on query retrieval delay and success ratio.

5.4.4.1 Influence of value k

Fig. 5.13 demonstrates the relationship between k and retrieval delay. When k increases from five to nine, except for unique queries, the retrieval delay of all query types increases. This is because, as k increases, there are more index terms to be resolved. Therefore, a QRP will have to encounter more nodes before it is resolved, which leads to a longer resolution time. In particular, unique queries have the highest retrieval delays, however, with increasing k values, the retrieval delay is unchanged. This is because for unique queries, they can only be resolved by one node.

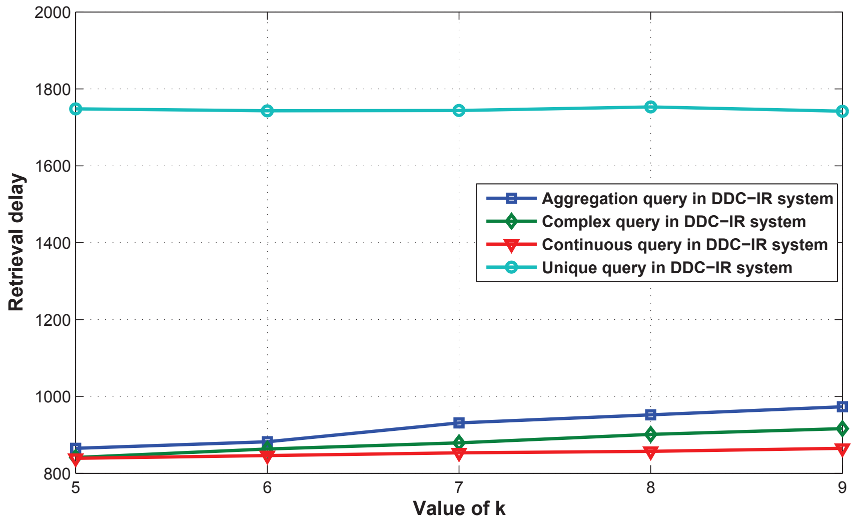

Figure 5.13 The influence of k on retrieval delay

5.4.4.2 Number of Querying Nodes

Fig. 5.14 shows that as w increases, the retrieval success ratio decreases. Specifically, when w increases from one to 10, the retrieval success ratio of complex queries drops from 100% to zero. This is because of the increasing number of replies. In DDC-IR, increasing replies can cause a full buffer, which will be discarded before they are received by the querying node. Interestingly, unique queries have a higher retrieval success ratio than complex queries with increasing w. This is because unique queries do not store duplicated data. Therefore, nodes experience a smaller probability of buffer overflow. Moreover, when the number of querying node increases, the retrieval success ratio of continuous queries can increase. For example, when w increases from three to seven, the retrieval success ratio of continuous queries increases from 57% to 61%. This is because DDC-IR uses a caching mechanism whereby similar queries can be answered by nodes with the corresponding data in their QHT. When w exceeds seven, the retrieval success ratio of continuous queries drops from 61% to 21%. This is because increasing replies leads to buffer overflow.

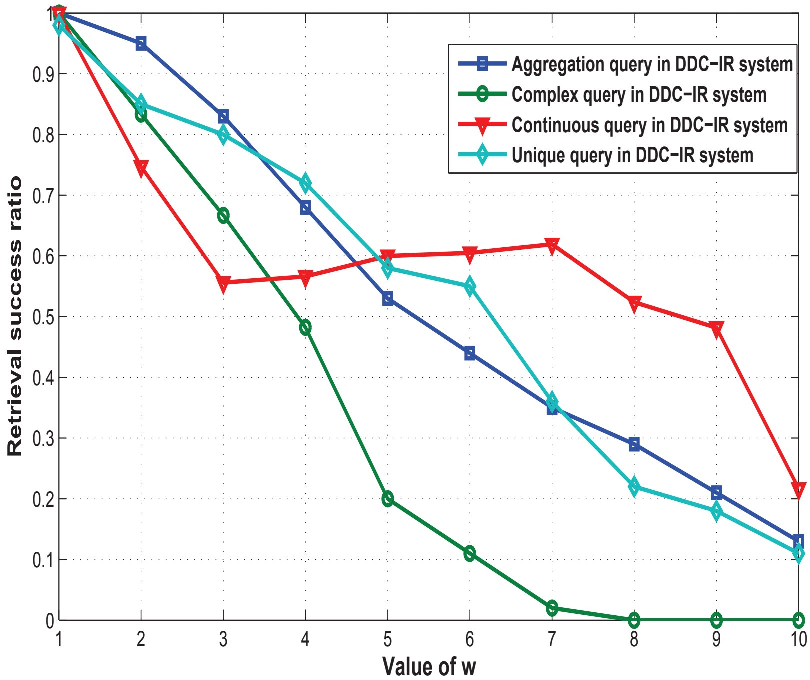

Figure 5.14 the influence of querying nodes' number on retrieval success ratio

5.4.5 Discussion

First, DDC-IR has a higher retrieval success ratio and lower retrieval delay than PHOPHET-IR for all query types. This is because in PHOPHET-IR, all queries are regarded as one-shot queries. That is, PROPHET-IR system assumes that one query can be completely resolved by one node. However, this method ignores the fact that nodes may only be able to answer a query partially. As a result, PHOPHET-IR has a low retrieval success ratio for complex and aggregated queries; both of which may require more than one node to reply. In DDC-IR, the process of resolving a complex and aggregated query relies on nodes distributing a copy of their matching data to answer each index term in queries. Hence, every node that has matching data can contribute to the resolution of queries. Moreover, this process combines the query transmission and retrieval process, which reduces retrieval delay.

Second, DDC-IR has a lower buffer occupancy level than PHOPHET-IR. There are two approaches to ensure low buffer occupancy level in DDC-IR: (i) transmitting query/reply according to similarity value, and (ii) using QHT. On one hand, using similarity value to transmit query/reply avoids high buffer occupancy because nodes do not exchange data arbitrarily. On the other hand, the use of QHT increases the likelihood that nodes have the necessary index terms to resolve queries, and avoid low retrieval success ratio which is caused by low duplication rate of index terms. Apart from that, DDC-IR system also includes a TTL mechanism to discard redundant and resolved queries. This also contributed to the low buffer occupancy level of nodes when they use DDC-IR. In PROPHET-IR, buffer occupancy level is simply controlled by data transmission times. Thus, when nodes have a limited buffer size or a large value of m, nodes will have high buffer occupancy level.

5.5 Conclusion

This chapter has proposed a data-centric IR system called DDC-IR, which can retrieve all types of queries from a DTN. Specifically, different from previous address-based IR systems, DDC-IR combines the query and reply transmission process, which has the effect of reducing retrieval delay. Moreover, nodes transmit queries and replies according to the similarity value of nodes to a query, which improves retrieval success ratio and lower buffer occupancy level. In addition, nodes using DDC-IR cache data based on the popularity of a query. The simulation studies on all four types of queries in both RWP and trace-file scenarios show that DDC-IR yields higher retrieval success ratios, lower retrieval delays and lower buffer occupancy levels as compared to past address-centric IR systems. Additionally, experiments investigating the influence of factors such as the number of queries and querying nodes, and showed that when the value of w and m are large, the IR system has a low retrieval success ratio and high buffer occupancy level. Moreover, due to the low duplication ratio of unique queries, they have a low retrieval success ratio. Henceforth, an immediate future study is to remedy these two problems.

Table 5.2 A comparison of different query types supported by DDC-IR.

Query Types	Objective	Example Query	Implementation
Complex	Retrieve data from nodes that satisfy a conjunctive or disjunctive normal form.	"temperature " AND Wollongong	The Query Type field is set to "Complex". Nodes with matching data transmit a reply. In this example,nodes with data that matches both "temperature" and "Wollongong" will send a reply. Nodes will store index terms with high popularity value and their corresponding data in their QHT.
Unique	Similar to a complex query but without caching. That is, data only exists on one node.	"Temperature AND Wollongong AND NodeID"	The Query Type field is set to "Unique". The key difference to complex queries is that nodes do not cache data or replies. This means nodes ignore the corresponding data in QRP marked as "Unique".
Aggregation	Provides a summary of all data from a designated set of nodes that match index terms.	"Average temperature AND Wollongong"	The Query Type field is set to "Aggregation". QRP stores index terms and corresponding data from nodes in the same area. As the example query, all nodes in Wollongong area transmit their corresponding temperature data to the node with highest similarity value to the Destination IDEs field of QRP. The node calculates an average temperature value and replies to querying node. Nodes store the summary with IDEs that indicate the area in their QHT.
Continuous	Provide a continuous flow of data from a set of nodes that matcha given index terms.	"temperature AND Wollongong"	The Query Type field is set to "Continuous", and there are two extra fields: (i) time duration and (ii) time label. In the example, "2 A.M. to 6 P.M" is stored in the former field. Data that meets the information in these fields will be included in the QRP and stored in the QHT of nodes.

Chapter 6

Conclusions

This thesis focuses on data dissemination in DTNs. Specifically, as pointed out in Chapter 1, this is a difficult problem because nodes experience stochastic and dynamic topologies and have limited resources and topological information. To date, there are three main categories of protocols: unicasting, multicasting and information retrieval. In regards to unicast protocols, this thesis has presented a key observation and limitations of prior works. That is, although there are many epidemic-based protocols, the performance of these protocols has never been evaluated in a unified framework. As for multicast protocols, prior works have not investigated the influence of key factors such as anti-entropy session, subscriber group size or forwarding policies. Lastly, little work has been carried out to evaluate and design new IR-based protocols for DTNs.

This thesis is the first to conduct a comprehensive study on data dissemination protocols in DTNs. In particular, all epidemic-based protocols are evaluated under a common framework comprising of both RWP and trace-file mobility models. The results show that the three enhancements proposed in Chapter 3 can effectively reduce nodes buffer occupancy level and duplication rate. In particular, the results confirm their superiority over existing epidemic routing protocols. The next key contributions are presented in Chapter 4 whereby factors such as subscribers group size, forwarding policy and anti-entropy on the delivery ratio of multicast bundles are evaluated. Moreover, this thesis presents a novel epidemic-based multicast protocol called EC Quota that reduces the buffer occupancy level of nodes when disseminating multicast bundles. The results show that the EC quota can reduce buffer occupancy level of nodes by up to 80% whilst maintaining similar bundle delivery ratio. Last but not the least, this thesis is the first to propose a data-centric IR system called DDC-IR, which supports all query types. DDC-IR incorporates a new query and reply packet, aka QRP, that combines both query and one or more replies in order to improve resolution probability and reduce buffer occupancy level. Additionally, nodes cache popular queries, which have the effect of speeding up query resolution.

A key future research direction is energy consumption. This is an important issue due to the ubiquity of smart devices. According to a recent survey [117], the total carbon emission related to digital devices is nearly one gigaton, which is 2% of total global emissions. However, in current DTNs routing protocols, data transmission is the primary aim and few work has considered energy consumption in DTNs. That is, current DTN routing protocols assume that all nodes are awake at all times.

Another future research is data dissemination in large-scale DTNs with tens of thousands of nodes. On one hand, the mobility pattern in such DTNs is very complex. In particular, each node may have varied mobility patterns in different scenarios or different time duration. On the other hand, to describe a mega-scale DTN needs novel methods to continuously record node movements. However, there is no simulator that can support such a massive trace file.

Social networking is also an important branch of DTNs research. The mobility pattern of nodes or people is influenced by people's relationship, and locations. For example, two good friends provide a reliable communication channel for data transmission, whilst the transmission between two strangers may fail due to fewer encounter times or shorter encounter duration. Another example is people that visit the same location frequently have a better chance to exchange data. These properties can thus be exploited by routing protocols to improve bundle delivery ratio.

Lastly, reducing retrieval delay and redundant data are two crucial problems for future research in DTN IR system. The results shown in Chapter 5 indicate that an increase in the number of sub-queries in a complex query can lead to significantly increased retrieval delay. In particular, when the number of nodes is large, increasing a sub-query in a complex query may imply to retrieve hundreds of nodes. Moreover, success rate can be improved by having a high duplication rate of required data. However, after the retrieval process finishes, this data may remain in the network. Therefore, an effective mechanism that reliably discards redundant data whilst ensuring a high retrieval success is another topic in future DTN IR research.

Bibliography

[1] 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.

[2] Y. Hayel and H. Tembine, "Information dissemination using epidemic routing with delayed feedback," in *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS)*, (Pisa, Italy), pp. 1–4, October 8-11 2007.

[3] P. Mundur, M. Seligman, and J. N. Lee, "Immunity-based epidemic routing in intermittent networks," in *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, (California, USA), pp. 609–621, June 2008.

[4] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," in *Technologies Report CS*, Department of Computer Science, Duke University, 2002.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, (Karlsruhe, Germany), Aug. 2003.

[6] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "RFC 4838: Delay-tolerant networking architecture," April 2007.

[7] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant network architecture: The evolving interplanetary internet." Internet Draft: draft-irtf-ipnrg-arch-01.txt, Feb. 2003.

[8] S. Burleigh, A. Hooke, 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.

[9] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *25th IEEE International Conference on Computer Communication (INFOCOM)*, (Miami, FL, USA.), pp. 1 –13, apr. 2006.

[10] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet," *ACM SIGPLAN NOTICES*, vol. 37, no. 10, pp. 96–107, 2002.

[11] T. Small and Z. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *ACM MobiHOC*, (New York, NY, USA), pp. 233–244, July 2003.

[12] P. McDonald, D. Geraghty, I. Humphreys, S. Farrell, and V. Cahill, "Sensor network with delay tolerance (SeNDT)," in *Proceedings of 16th International Conference on Computer Communications and Networks (IC-CCN)*, (Honolulu, Hawaii, USA), pp. 1333 –1338, August, 2007 2007.

[13] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[14] N. Glance, D. Snowdon, and J.-L. Meunier, "Pollen: using people as a communication medium," *Computer Networks*, vol. 35, no. 4, pp. 429 – 442, 2001.

[15] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *ACM MOBICOM*, (New York, NY, USA), pp. 334–345, May 2006.

[16] S. Naidu, S. Chintada, M. Sen, and S. Raghavan, "Challenges in deploying a delay tolerant network," in *Proceedings of the third ACM workshop on Challenged networks (CHANTS)*, (New York, NY, USA), pp. 65–72, 2008.

[17] "Wizzy project." http://www.wizzy.org.za.

[18] "Widernet." http://www.widernet.org.

[19] "Digital study hall." http://dsh.cs.washington.edu/index.html.

[20] M. Quwaider and S. Biswas, "DTN routing in body sensor networks with dynamic postural partitioning," *Ad Hoc Networks*, vol. 2, no. 5, pp. 124–132, 2010.

[21] BBC, "SA pigeon faster than broadband." http://news.bbc.co.uk/2/hi/africa/8248056.stm, 2009.

[22] M. Araki and K.-W. Chin, "Trainnet: A novel transport system for delivering non real time video," *Elsevier Computer Communications*, vol. 33, no. 15, pp. 1850–1863, 2010.

[23] E. Daly and M. Haahr, "The challenges of disconnected delay-tolerant MANETs," *Ad Hoc Networks*, vol. 8, no. 3, 2009.

[24] R. P. Sushant Jain, Kevin Fall, "Routing in a delay tolerant network," in *ACM SIGCOMM*, (Portland, OR, USA), pp. 120–131, Aug. 2004.

[25] J. Moy, "OSPF version 2." Request for Comments, RFC2178, Apr. 1998.

[26] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM*, (New York, NY, USA), pp. 234–244, June 1994.

[27] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing." Request for Comments: RFC 3561, July 2003.

[28] M. Chuah and J. Han, "Performance evaluation of information retrieval schemes for multi-attribute queries in DTNs," in *IEEE Military Communications Conference (MILCOM)*, (San Diego, CA, USA), pp. 1 –7, nov. 2008.

[29] M. Mitra and B. B. Chaudhuri, "Information retrieval from documents: A survey," *Inf. Retr.*, vol. 2, no. 5, pp. 141–163, 2000.

[30] 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," in *IEEE INFOCOM*, vol. 2, (San Francisco, USA), pp. 825–835, March 2003.

[31] M. Abdulla and R. Simon, "Controlled epidemic routing for multicasting in delay tolerant networks," in *IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems (MASCOTS)*, (Baltimore, USA), pp. 1–10, September 2008.

[32] N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active query forwarding in sensor networks," *Journal of Ad Hoc Networks*, vol. 3, no. 8, pp. 91–113, 2005.

[33] M. Chuah and P. Yang, "Performance evaluation of content-based information retrieval schemes for DTNs," in *IEEE Military Communications Conference (MILCOM)*, (Orlando, Florida, USA), pp. 1 –7, oct. 2007.

[34] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," *Opereration System Review of Special Interest Group on Operating Systems (SIGOPS),*, vol. 22, no. 1, pp. 8 – 32, 1988.

[35] J. Tower and T. Little, "A proposed scheme for epidemic routing with active curing for opportunistic networks," in *22nd IEEE International Conference on Advanced Information Networking and Applications (AINA)*, (GinoWan, Okinawa, Japan), pp. 1696–1701, March 25-28 2008.

[36] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, *Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding in Sparse Mobile Networks*, ch. Lecture Notes in Computer Science, pp. 1180–1192. Springer Berlin / Heidelberg, 2005.

[37] P. Mundur, M. Seligman, and J. N. Lee, "Epidemic routing with immunity in delay tolerant networks," in *IEEE Military Communications Conference (MILCOM)*, (San Diego, CA, USA), Nov. 2008.

[38] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multi-copy case," *IEEE Transactions on Networking*, vol. 16, no. 2, pp. 77–90, 2008.

[39] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACK Transactions on Networking*, vol. 16, no. 2, pp. 63–76, 2008.

[40] 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. 10, pp. 1530–1540, 2010.

[41] Y. Li, Y. Jiang, D. Jin, L. Su, L. Zeng, and D. Wu, "Energy-efficient optimal opportunistic forwarding for delay tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 11, pp. 4500–4513, 2010.

[42] E. Altman, G. Neglia, F. D. Pellegrini, and D. Miorandi, "Decentralized stochastic control of delay tolerant networks," in *IEEE INFOCOM*, (Rio de Janeiro, Brazil), pp. 23–38, April 2009.

[43] J. Davis, A. Fagg, and B. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," in *Proceedings of Fifth International Symposium on Wearable Computers (ISWC)*, (Zurich, Switzerland), pp. 141–148, October 18-21 2001.

[44] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings*

of the 1st international MobiSys workshop on Mobile opportunistic networking (MOBIOPP), (New York, NY, USA), pp. 62–66, Sep. 2007.

[45] W. Zhao and M. Ammar, “Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks,” in *Proceedings of The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, (San Juan, Puerto Rico), pp. 308–314, May 2003.

[46] W. Zhao, M. Ammar, and E. Zegura, “Controlling the mobility of multiple data transport ferries in a delay-tolerant network,” in *IEEE INFOCOM*, (Miami, Florida), pp. 1407 – 1418, March 2005.

[47] W. Zhao, M. Ammar, and E. Zegura, “A message ferrying approach for data delivery in sparse mobile ad hoc networks,” in *ACM MobiHOC*, (New York, NY, USA), pp. 187–198, 2004.

[48] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, “MDDV: a mobility-centric data dissemination algorithm for vehicular networks,” in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks (VANET)*, (New York, NY, USA), pp. 47–56, Sep. 2004.

[49] B. Burns, O. Brock, and B. Levine, “MV routing and capacity building in disruption tolerant networks,” in *IEEE INFOCOM*, vol. 1, (Miami, Florida), pp. 398–408, March 2005.

[50] Y. Zhu, W. Wu, V. Leung, and L. Yang, “Energy-efficient tree-based message ferrying routing schemes for wireless sensor networks,” in *The Third International Conference on Communications and Networking in China (CHINACOM)*, (Zhejiang, China), pp. 844–848, August 2008.

[51] J. Yang, Y. Chen, M. Ammar, and C. Lee, “Ferry replacement protocols in sparse MANET message ferrying systems,” in *IEEE Wireless Communications and Networking Conference*, (Philadelphia, PA), pp. 2038–2044, March 2005.

[52] R. Shah, S. Roy, S. Jain, and W. Brunette, “Data MULEs: modeling a three-tier architecture for sparse sensor networks,” in *IEEE Interna-*

tional Workshop on Sensor Network Protocols and Applications (SNPA), (Anchorage, AK), pp. 30–41, May 2003.

[53] R. Cesarone, D. Abraham, and L. Deutsch, "Prospects for a next-generation deep-space network," *Proceedings of the IEEE*, vol. 95, no. 10, pp. 1902 –1915, 2007.

[54] C. Shen, G. Borkar, S. Rajagopalan, and C. Jaikaeo, "Interrogation-based relay routing for ad hoc satellite networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, (Taipei, China), pp. 2920–2924, November 2002.

[55] Y. Chen, W. Zhao, M. Ammar, and E. Zegura, "Hybrid routing in clustered DTNs with message ferrying," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking (MOBIOPP)*, (New York, NY, USA), pp. 75–82, May 2007.

[56] D. Nain, N. Petigara, and H. Balakrishnan, "Integrated routing and storage for messaging applications in mobile ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 6, pp. 595–604, 2004.

[57] K. Harras and K. Almeroth, "Inter-regional messenger scheduling in delay tolerant mobile networks," in *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, (Washington, USA), pp. 93–102, 2006.

[58] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, (Taormina, Italy), pp. 183–189, June 2005.

[59] H. Lu, L. Yin, W. He, and Y. Cao, "A novel time interval routing protocol in delay tolerant networks," in *International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC)*, (Wuhan, China), pp. 430–433, April 2009.

[60] E. P. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical routing in delay tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–955, 2007.

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

[62] T. K. Huang, C. K. Lee, and L. J. Chen, "PRoPHET+: An adaptive PRoPHET-based routing protocol for opportunistic network," in *24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, (Perth, Australia), pp. 112–119, April 2010.

[63] W. Gao and G. Cao, "On exploiting transient contact patterns for data forwarding in delay tolerant networks," in *IEEE International Conference on Network Protocols*, (Kyoto, Japan), pp. 134–144, October 2010.

[64] M. Kalantari and R. La, "A DTN packet forwarding scheme inspired by thermodynamics," in *42nd Annual Conference on Information Sciences and Systems (CISS)*, (Baltimore, USA), pp. 1216 –1221, April 2008.

[65] Z. Li and H. Shen, "Utility-based distributed routing in intermittently connected networks," in *37th International Conference on Parallel Processing (ICPP)*, (Torino, Italy), pp. 604–611, September 2008.

[66] J. LeBrun, C. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *2005 IEEE 61st Vehicular Technology Conference (VTC)*, vol. 4, (Dallas, USA), pp. 2289–2293, Septemper 2005.

[67] H. Kang and D. Kim, "Vector routing for delay tolerant networks," in *IEEE 68th Vehicular Technology Conference (VTC)*, (Marina Bay, Singapore), pp. 1–5, May 2008.

[68] L. Yin, Y. Cao, and W. He, "Similarity degree-based mobility pattern aware routing in DTNs," in *International Symposium on Intelligent Ubiq-*

uitous Computing and Education (IUCE), (Chengdu, China), pp. 345 – 348, May 2009.

[69] J. Leguay and T. F. V. Conan, "DTN routing in a mobility pattern space," in *ACM SIGCOMM workshop on Delay-tolerant networking (WDTN)*, (Philadelphia, USA), pp. 276–283, July 2005.

[70] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *IEEE INFOCOM*, (Rio de Janeiro, Brazil), pp. 361–374, April 2009.

[71] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *IEEE INFOCOM*, (Barcelona, Spain), pp. 1–11, April 2006.

[72] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication routing in DTNs: a resource allocation approach," *IEEE/ACM Transactions on Networking*, vol. 18, no. 8, pp. 596–609, 2010.

[73] K. Tan, Q. Zhang, and W. Zhu, "Shortest path routing in partially connected ad hoc networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, (San Francisco, USA), pp. 1038–1042, December 2003.

[74] M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile*, vol. 8, no. 2, pp. 246–260, 2009.

[75] J. Burgess, B. N. Levine, R. Mahajan, J. Zahorjan, A. Balasubramanian, A. Venkataramani, Y. Zhou, B. Croft, N. Banerjee, M. Corner, and D. Towsley, "CRAWDAD data set umass/diesel (v. 2008-09-14)." Downloaded from http://crawdad.cs.dartmouth.edu/umass/diesel, September 2008.

[76] S. Milgram, "The small world problem," *Psychology Today*, vol. 2, no. 2, pp. 60–67, May, 1967.

[77] J. G. Joy, H. Ngo, and C. Qiao, “Mobility profile based routing within intermittently connected mobile ad hoc networks (ICMAN),” in *Proceedings of the 2006 international conference on Wireless communications and mobile computing (IWCMC)*, (Vancouver, Canada), pp. 551–556, Jun. 2006.

[78] E. Daly and M. Haahr, “Social network analysis for information flow in disconnected delay-tolerant MANETs,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, pp. 606–621, 2009.

[79] M. Newman, “A measure of betweenness centrality based on random walks,” *Social Networks*, vol. 27, no. 1, pp. 39 – 54, 2005.

[80] W. Gao and G. Cao, “User centric data dissemination in disruption tolerant networks,” in *IEEE INFOCOM*, (Shanghai, China), pp. 173–188, April 2011.

[81] K. Xu, V. O. Li, and J. Chung, “Exploring centrality for message forwarding in opportunistic networks,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, (Sydney, Australia), pp. 143–152, April 2010.

[82] C. Mascolo and M. Musolesi, “SCAR: context-aware adaptive routing in delay tolerant mobile sensor networks,” in *Proceedings of the 2006 international conference on Wireless communications and mobile computing (IWCMC)*, (New York, NY, USA), pp. 533–538, Aug. 2006.

[83] A. Balasubramanian, B. Levine, and A. Venkataramani, “DTN routing as a resource allocation problem,” in *ACM SIGCOMM*, (New York), pp. 134–149, October 2007.

[84] P. Pantazopoulos, I. Stavrakakis, A. Passarella, and M. Conti, “Efficient social-aware content placement for opportunistic networks,” in *IFIP/IEEE Seventh International Conference on Wireless On-demand Network Systemsand Services (WONS)*, (Kranjska Gora, Slovenia), pp. 1228–1239, February, 3-5 2010.

[85] P. Hui and S. Buchegger, "Groupthink and peer pressure: Social influence in online social network groups," in *Proceeding of International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, (Athens, Greece), pp. 256–266, July 2009.

[86] K. Xu, P. Hui, V. O. K. Li, J. Crowcroft, V. Latora, and P. Liò, "Impact of altruism on opportunistic communications," in *Proceeding of First IEEE International Conference on Ubiquitous and Future Networks (ICUFN)*, (Hong Kong, China), pp. 87–99, June 2009.

[87] C. Boldrini, M. Conti, and A. Passarella, "Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution," *Elsevier Pervasive and Mobile Computing*, vol. 4, no. 5, pp. 633–657, 2008.

[88] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: social-aware data dissemination in opportunistic networks," in *IEEE MSWiM*, (Vancouver, Canada), pp. 203–210, Oct. 2008.

[89] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, (New York, NY, USA), pp. 241–250, Aug. 2008.

[90] C. Boldrini, M. Conti, and A. Passarella, "The sociable traveler: human traveling patterns in social-based mobility," in *Proceedings of the 7th ACM international symposium on Mobility management and wireless access (MobiWAC)*, (Tenerife, Spain), pp. 34–41, Oct. 2009.

[91] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, (New York, NY, USA), pp. 53–60, June 1999.

[92] K. Leung, W. Massey, and W. Whitt, "Traffic models for wireless communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 8, pp. 1353–1364, 1994.

[93] H. Guo, J. Li, and Y. Qian, "Hop-DTN: Modeling and evaluation of homing-pigeon-based delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 857 –868, 2010.

[94] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD data set epfl/mobility (v. 2009-02-24)." Downloaded from http://crawdad.cs.dartmouth.edu/epfl/mobility, Feb. 2009.

[95] N. Eagle and A. S. Pentland, "CRAWDAD data set mit/reality (v. 2005-07-01)." Downloaded from http://crawdad.cs.dartmouth.edu/mit/reality, July 2005.

[96] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, "CRAWDAD data set dartmouth/campus (v. 2009-09-09)." Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus, September 2009.

[97] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)." Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

[98] V. Srinivasan, M. Motani, and W. T. Ooi, "CRAWDAD data set nus/contact (v. 2006-08-01)." Downloaded from http://crawdad.cs.dartmouth.edu/nus/contact, August 2006.

[99] P. Jacquet, B. Mans, and G. Rodolakis, "Broadcast delay of epidemic routing in intermittently connected networks," in *IEEE International Symposium on Information Theory (ISIT)*, (Seoul, Korea), pp. 839–843, July 2009.

[100] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in delay tolerant networks: semantic models and routing algorithms," in *Proceedings of the ACM SIGCOMM workshop on Delay-tolerant networking (WDTN)*, (New York, NY, USA), pp. 268–275, May 2005.

[101] Y. Chen, J. Yang, W. Zhao, M. Ammar, and E. Zegura, "Multicasting in sparse MANETs using message ferrying," in *IEEE Wireless Communica-*

tions and Networking Conference (WCNC), (Philadelphia, PA), pp. 691–696, April 2006.

[102] Q. Ye, L. Cheng, M. C. Chuah, and B. Davison, "Os-multicast: On-demand situation-aware multicasting in disruption tolerant networks," in *IEEE VTC*, (Montreal, Canada), pp. 357–358, May 2006.

[103] P. Yang and M. C. Chuah, "Context-aware multicast routing scheme for disruption tolerant networks," in *Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks (PE-WASUN)*, (New York, NY, USA), pp. 66–73, Aug. 2006.

[104] J. Greifenberg and D. Kutscher, "Efficient publish/subscribe-based multicast for opportunistic networking with self-organized resource utilization," in *22nd International Conference on Advanced Information Networking and Applications (AINAW)*, (Ginowan, Japan), pp. 1708–1714, March 2008.

[105] W. Gao, Q. Li, B. Zhao, and G.Cao, "Multicasting in delay tolerant networks: A social network perspective," in *ACM MobiHOC*, (New Orleans, Louisiana, USA), pp. 56–68, May 2009.

[106] Q. Ye, L. Cheng, M. C. Chuah, and B. D. Davison, "Performance comparison of different multicast routing strategies in disruption tolerant networks," *Computer Communications: special Issue of Computer Communications on Delay and Disruption Tolerant Networking*, vol. 32, no. 16, pp. 1731 – 1741, 2009.

[107] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of ACM SIGCOMM Workshops*, (Philadelphia, PA, USA), pp. 134–144, August 22-26 2005.

[108] D. Geraghty, "Sensor networking with delay tolerance (SeNDT)." http://down.dsg.cs.tcd.ie/sendt, 2006.

[109] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi, "Mobeyes: Smartmobs for urban monitoring with a vehicular sensor network," *IEEE Wireless Communications*, vol. 13(5), no. 6, pp. 52–57, (2006).

[110] V. Lenders, G. Karlsson, and M. May, "Wireless ad hoc podcasting," in *IEEE SECON*, (San Diego, California, USA), pp. 273–283, June 18-21 2007.

[111] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks*, vol. 51, no. 7, pp. 2867 – 2891, 2007.

[112] S. McCanne and S. Floyd., "ns network simulator-2." http://www.isi.edu/nsname/ns/.

[113] D. Kotz, T. Henderson, and I. Abyzov, "CRAWDAD data set dartmouth/campus (v. 2004-12-18)." Downloaded from http://www.crawdad.org/dartmouth/campus, Dec. 2004.

[114] G. Resta and P. Santi, "An analysis of the node spatial distribution of the random waypoint model for ad hoc networks," in *ACM Workshop on Principles of Mobile Computing (POMC)*, (Toulouse, France), pp. 97–106, July 2007.

[115] M. A. P. de Cristo, P. P. Calado, M. de Lourdes da Silveira, I. Silva, R. Muntz, and B. Ribeiro-Neto, "Bayesian belief networks for ir," *International Journal of Approximate Reasoning*, vol. 34, no. 2-3, pp. 163 – 179, 2003.

[116] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/intel (v. 2006-01-31)." Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/intel, Jan. 2006.

[117] J. Newton, "Is cloud computing green computing?," *GPSolo*, vol. 27, no. 8, pp. 28–31, 2010.