QoS-Aware Data Query and Dissemination in Mobile Opportunistic Networks

A Dissertation

Presented to the

Graduate Faculty of the

University of Louisiana at Lafayette

In Partial Fulfillment of the

Requirements for the Degree

Doctor of Philosophy

Yang Liu

Fall 2014

QoS-Aware Data Query and Dissemination in Mobile Opportunistic Networks

Yang Liu

APPROVED:

Hongyi Wu, Chair
Professor of Computer Science
The Center for Advanced Computer Studies

Nian-Feng Tzeng
Professor of Computer Science
The Center for Advanced Computer Studies

Dmitri Perkins
Professor of Computer Science
The Center for Advanced Computer Studies

Magdy Bayoumi
Professor of Computer Engineering
The Center for Advanced Computer Studies

Mary Farmer-Kaiser
Interim Dean of the Graduate School

DEDICATION

I dedicate my dissertation to my father, Xuxin Liu, my mother, Shuqin Yang, and my wife, Meihong Xu.

ACKNOWLEDGMENTS

This dissertation would not have been completed without the help and support of the following people. The following words cannot express the depth of my appreciation.

First, I would like to thank my advisor, Dr. Hongyi Wu. Without his continuous support, advice, and encouragement throughout my Ph.D. study, I could not have finished this work. I would also like to thank Dr. Nian-Feng Tzeng, Dr. Dmitri Perkins, and Dr. Magdy Bayoumi for their valuable supervision of my academic study and research.

Second, I would like to thank all of the CACS faculty for the wonderful knowledge they taught me and all of the CACS staff for their help during my time in CACS as a Ph.D. student. I would also like to thank all of my friends and colleagues in the iWIN lab and CACS for their unselfish help, especially Zhipeng Yang, Ting Ning, Elman Bashar, Baijun Wu, Rui Ning, Mingmin Bai, Yanyan Han, and John Rahat.

Third, I would like to thank my parents. They always supported me and encouraged me with their best wishes.

Finally, I would like to thank my wife, Meihong Xu, for her endless love, support, and encouragement at all times. She accompanies me in overcoming all the difficulties in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: Introduction

A social network is a social structure made up of a set of social actors (such as individuals or organizations) and a set of the dyadic ties between these actors. The social network perspective provides a set of methods for analyzing the structure of whole social entities as well as a variety of theories explaining the patterns observed in these structures [88]. The study of these structures uses social network analysis to identify local and global patterns, locate influential entities, and examine network dynamics. Social networks and the analysis of them is an inherently interdisciplinary academic field which emerged from social psychology, sociology, statistics, and graph theory. Georg Simmel authored early structural theories in sociology, emphasizing the dynamics of triads and "web of group affiliations [72]." Jacob Moreno is credited with developing the first sociograms in the 1930s to study interpersonal relationships. These approaches were mathematically formalized in the 1950s and theories and methods of social networks became pervasive in the social and behavioral sciences by the 1980s [26, 88]. Social network analysis is now one of the major paradigms in contemporary sociology, and is also employed in a number of other social and formal sciences. Together with other complex networks, it forms part of the nascent field of network science [7, 19].

Social networking is among the fastest growing information technologies, as evidenced by the popularity of such online social network sites as Facebook, Twitter, LinkedIn, and Google+ that continue to experience explosive growth. Several features of online social networks are common to each of the more than 300 social networking sites currently in existence. The most basic feature is the ability to create and share a personal profile. This profile page typically includes a photo, some basic personal information (name, age, sex, and location), and extra space for listing your favorite bands, books, TV shows, movies, hobbies, and Web sites. Most social networks on the Internet also let you post photos, music, videos, and personal blogs on your profile page. But the most important feature of online social networks is the ability to find and make friends with other site members. These friends also appear as links on your profile page so visitors can easily browse your online friend network [70].

In contrast to the popular web-based online social networks that rely on the Internet infrastructure (including cellular systems) for communication, this dissertation focuses on mobile opportunistic networks formed by mobile users who share similar interests and connect with one another by exploiting the Bluetooth and/or WiFi connections of their mobile phones or portable tablets. A mobile opportunistic network is often created for a local community where the participants have frequent interactions, e.g., people living in an urban neighborhood, students studying in a college, or tourists visiting an archaeological site. Its size varies from a large group (for instance, all the students in a university) to a small cluster (such as members of a school band). It may serve a community over a long span of years, or be temporary to last for as short as a few hours only (e.g., for social networking among a group of tourists).

An application that can be applied into mobile opportunistic networks is the device-to-device (D2D) communication in 5G network. 5G (5th generation mobile networks or 5th generation wireless systems) denotes the next major phase of mobile telecommunications standards beyond the current 4G/IMT-Advanced standards. 5G is also referred to as beyond 2020 mobile communications technologies. 5G does not describe any particular specification in any official document published by any telecommunication standardization body. 5G is a super-efficient "universal" mobile network that is always attentive to demand and where resources are continuously optimised to deliver a performance that is “always sufficient” - so users will perceive that they are connected to a network of infinite bandwidth; a super-fast “mobile” network comprising the next generation of small cells densely clustered together to give a contiguous coverage over at least urban areas and delivering peak data rates of up to 1 Gb/s; and a converged wireless-fiber network that uses, for the first time for wireless Internet access, the millimeter wave bands (20 - 60 GHz) so as to allow very wide bandwidth radio channels able to support data access speeds of up to 10 Gb/s. Device-to-device (D2D) communication refers to a radio technology that enables devices to communicate directly with each other, that is without routing the data paths through a network infrastructure [25].

In this dissertation, I first study the problem of QoS-aware data query in mobile opportunistic networks. I develop a distributed data query protocol. The proposed protocol is based on two key techniques. First, it employs “reachable expertise” as the routing metric to guide the transmission of query requests. Second, it exploits the redundancy in query transmission, which can effectively improve the query delivery rate in practice if it is properly controlled. To demonstrate the feasibility and efficiency of the proposed data query protocol and to gain useful empirical insights, I carry out a testbed experiment using off-the-shelf Dell Streak tablets. The experiment involves 25 volunteers and lasts for 15 days. Moreover, extensive simulations (based on codes extracted from the prototype implementation) are carried out to learn the performance trend under various network settings, which are not practical to build and evaluate in laboratories.

After the QoS-aware data query in mobile opportunistic networks is discussed, I introduce how to enable efficient QoS-aware data dissemination in mobile opportunistic networks. The QoS-aware delivery probability (*QDP*) is first introduced to reflect the capability of a node to deliver data to a destination within a given delay budget. Each node maintains a set of *QDPs* to make autonomous decisions for QoS-aware data delivery. At the same time, a prioritized queue is employed by each mobile node. In order to support efficient prioritization and redundancy control, the priority is determined by a function of traffic class and data redundancy. The former is pre-determined by the corresponding application, while the latter is dynamically estimated during data delivery. Two experiments are carried out to demonstrate and evaluate the proposed QoS-aware data delivery scheme. The first experiment involves multiple clusters of static Crossbow Micaz sensors that are connected by three mobile nodes carried respectively by a flying vehicle (with high mobility) and two people (with low mobility). All mobile nodes move according to predefined routes and speeds. The second experiment is under a mobile social network setting, where the experimental program is implemented on Dell Streak Android tablets carried by 23 volunteers with arbitrary and diverse mobility patterns for a period of two weeks. Moreover, simulation results are obtained under DieselNet trace and power-law mobility model to study the scalability and performance trend with the increase of network size, traffic load, and nodal mobility.

After the problem of delivering a data packet from a single source to a single destination within a predefined delay budget is studied, I also investigate the problem of delivering a data packet to a set of destinations within a predefined delay budget, which is the delay-constrained least-cost multicast problem in mobile opportunistic networks. While there are a handful of studies on multicasting in mobile opportunistic networks [6, 22, 27, 28, 29, 42, 49, 54, 58, 98], they all deal with unconstrained, best-effort data transmissions. Note that although delay is often considered as a metric in performance evaluation, none of the existing solutions formulate the problem with an explicit delay constraint. I formally formulate the problem and show it is essentially a NP-complete 0-1 integer program. While the 0-1 integer programming model can yield optimal results, it is computationally expensive and thus unpractical for real-world implementation. Given the NP-completeness of the problem, I explore efficient and scalable heuristic solutions. I first introduce a centralized heuristic algorithm which aims to discover a tree for multicasting, in order to meet the delay constraint and achieve low communication cost. While the centralized solution can be adapted to a distributed implementation, *it is inefficient in a mobile opportunistic network, since it intends to apply a deterministic transmission strategy in a nondeterministic network* by transmitting all data packets via a predetermined tree. In mobile opportunistic networks, even if the optimal routing tree can be computed, it is the "best" only on a statistic basis for a large number of data packets. It is not necessarily the best solution for every individual transmission. Based on the above observation, I develop a distributed online algorithm that makes an efficient decision on every transmission opportunity. When a node meets another node, the former needs to decide whether to transmit a packet to the latter. Such a routing decision is made based on a delay/cost-aware multicast routing metric, which indicates if the latter helps reduce the cost to deliver the packet to its destinations while reaching a desired delivery probability within a given delay budget. I prototype the proposed distributed online multicast algorithm using off-the-shelf Nexus tablets and conduct an experiment that involves 37 volunteers and lasts for 21 days to demonstrate its effectiveness. I also carry out simulations to evaluate the scalability of the proposed schemes under large-scale networks.

The rest of this dissertation is organized as follows. Chapter 2 discusses the background, related work and motivation of this dissertation. Chapter 3 presents efficient data query in mobile opportunistic networks. Chapter 4 introduces efficient quality-of-service (QoS) support in mobile opportunistic networks. Delay-constrained least-cost multicasting in mobile opportunistic networks is discussed in Chapter 5. Finally, Chapter 6 concludes the dissertation.

CHAPTER 2: Background and Related Work

In this chapter, I introduce the background and motivation of the proposed research and related work.

2.1 Background

2.1.1 Mobile Opportunistic Network

In recent years, mobile opportunistic networks emerged as a new mechanism of communications in wireless networks. Unlike mobile ad hoc networks (MANETs) that require end-to-end communication paths for message exchange, the communication in mobile opportunistic networks takes place on the establishment of opportunistic contacts among mobile nodes, without availability of end-to-end message routing paths. As the mobile devices can make contact only when humans come into contact, such networks are tightly coupled with human social networks. Therefore, the mobile opportunistic networks exploit the human behaviors and social relationships to build more efficient and trustworthy message dissemination schemes [47].

It is reported that the estimated number of mobile phone users are 3.3 billion worldwide, which is more than half of the world's population [15]. Most of the mobile phones in the current era are equipped with Wi-Fi, Bluetooth, cameras, sensors, and numerous other components. Moreover, most of the modern vehicles are also installed with communication interfaces and sensory equipment. Such a widespread use and availability of mobile communication devices create a huge number of contact opportunities among humans, and are key to the establishment of opportunistic mobile social networks [45]. The human mobility is the key factor in opportunistic communications, and there could be delays in message transfers as long as the humans carrying mobile devices do not come into each other's transmission range. Therefore, several research projects are conducted in various parts of the world to analyze the human mobility and social interaction patterns, and on the basis of that to build efficient message routing models that incur minimum message delays.

2.1.2 Quality of Service

Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. For example, a required bit rate, delay, jitter, packet dropping probability, and/or bit error rate may be guaranteed. Quality of service guarantees are important if the network capacity is insufficient, especially for real-time streaming multimedia applications such as voice over IP, online games and IP-TV, since these often require fixed bit rate and are delay sensitive, and in networks where the capacity is a limited resource, for example in cellular data communication.

Mobile opportunistic networks are characterized by intermittent and nondeterministic connectivity, often due to interruptible wireless links, sparse network deployment, and/or nodal mobility. Such opportunistic networking has been discussed in the context of delay/disruption-tolerant networks, sporadically connected sensor networks, vehicular networks, and peer-to-peer mobile social networks [2, 9, 20, 27, 86, 95, 96]. How to discover and utilize opportunistic communication resources for efficient data transmission has been one of the central research issues in such networks as evidenced by extensive discussions in the literature [2, 4, 9, 10, 17, 20, 24, 27, 32, 37, 38, 39, 43, 44, 46, 48, 50, 55, 56, 60, 61, 62, 64, 65, 66, 73, 74, 77, 81, 84, 86, 87, 89, 90, 95, 96, 99]. However, limited prior work has addressed Quality-of-Service (QoS). While long data delivery delay is generally unavoidable given the unique intermittent connectivity, QoS, especially the guarantee for end-to-end delivery delay, is highly desired in a variety of applications. For example, the dissemination of a data message (such as an advertisement or coupon [27, 63, 67, 96]) in a mobile social network must meet a delay budget no longer than its expiration date, and different data messages are often associated with different delay budgets. Separately, in wildlife tracking applications, interactive control and event report must be delivered within a short end-to-end delay bound, as opposed to routine transmissions of ambient environmental data that can tolerate long delay [2]. Data delivered beyond their delay budgets often lead to reduced or completely forfeited value.

2.2 Related Work and Motivation

Data query in Mobile opportunistic networks is a very unique, interesting, and challenging problem, rendering not only conventional data query schemes for well-connected computer systems but also distributed solutions for mobile ad hoc networks [3, 12, 35, 75, 82, 92] and mobile (online) social networks [53] inapplicable here. Only a handful of works have considered data query in opportunistic network settings. For example, Osmosis [41] employs an epidemic approach to perform file lookup in pocket switched networks. While it is simple and reliable, the communication overhead is very high due to the flooding-like epidemic routing. DelQue [23] aims to query geo-location-based information. It assumes each node moves according to a given schedule and adopts a semi-Markov model to predict nodal meeting events, in order to identify a proper relay to carry the query to the target location and bring the interested information back to the source. [94] proposes a distributed database query framework based on several communication and computing techniques specifically tailored for RFID networks. Neither of them efficiently supports data queries in moible opportunistic networks. On the other hand, although several routing algorithms have been proposed for opportunistic networks by exploiting social relations among mobile users to achieve efficient routing [16, 27, 40, 52, 59, 100], they are developed for generic communications, without consideration of the unique needs and constraints in data query. Among them, [100] is the most recent one, which exploits a distributed community partitioning algorithm to divide a DTN into smaller communities. For intra-community communication, a utility function convoluting social similarity and social centrality with a decay factor is used to choose relay nodes. For inter-community communication, the nodes moving frequently across communities are chosen as relays to carry data to destinations efficiently. Although [100] introduces a solution for DTNs which leverages social properties and mobility characteristics of users, it is not truly applicable for the data query in mobile opportunistic networks, because it does not capture the inherent features for the query delivery in mobile opportunistic networks, hence the nodes are not helpful for each other by making the correct decisions to carry queries to satisfactory nodes.

Quality-of-Service (QoS) is a critical issue in mobile networks, with a diversity of approaches proposed recently [1, 18, 31, 51, 78, 91]. However, none of them are developed for QoS support in mobile opportunistic networks. Given the unique characteristics of mobile opportunistic networks (especially the intermittent, nondeterministic network connectivity), solutions for QoS support in conventional networks are not applicable here. In general, QoS provisioning in mobile opportunistic networks is a less-studied area with limited existing solutions.

In opportunistic networks, the delivery of a single copy data message is often subject to high loss rate or extremely long delay. Therefore, redundancy (achieved by duplication or coding [86]) is commonly employed for desired communication performance. However, redundancy increases overhead, and worse yet, excessive redundancy may degrade overall network performance due to congested channels and frequent buffer overflow [84, 97]. For example, Epidemic routing [84, 97] simply replicates data messages at each communication opportunity. It is essentially a store-and-forward flooding, able to achieve minimum delay under the assumption of unlimited buffer space and link bandwidth. However, in a practical IPN setting, such naive flooding wastes resources and can severely degrade performance. To this end, a series of approaches [4, 10, 43, 56, 77, 81, 89] have been developed to limit redundancy for efficienct resource utilization. However, none of them truly support traffic differentiation or QoS. Among them, [4] is most relevant to this dissertation, where a node sorts its packets in its queue in decreasing order of the ages and makes routing decision according to a marginal utility function to improve the probability of delivering packets within their deadlines. While this scheme maximizes the overall deadline-satisfied delivery rate, it does not differentiate traffic and is not equivalent to the support of QoS. For instance, a data flow with high QoS priority can be submerged by a large volume of background low-priority data that have stayed in the network long enough and thus occupied the head of queue. As a result, the transmission of high-priority data is delayed, resulting in poor QoS provisioning. Moreover, a "control channel" is required in [4] to timely share global information among nodes, but such a channel is not always available in a practical network. Separately, there are a handful of studies dedicated to QoS support in delay-tolerant networks [57, 68, 83]. However, they are either based on simplified settings or operate at the level of individual links only. For example, both [68] and [57] are based on epidemic routing, which itself is inefficient. The former adapts forwarding probabilities and time-to-live parameters to control the usage of network resources, while the latter chooses appropriate transmission probability to enable QoS differentiation in epidemic routing. Despite their value as initial efforts to develop QoS-aware DTN protocols, they are not yet ready for practical application due to the inefficient underlying epidemic routing scheme, and neither of them can be readily extended to other DTN routing algorithms. On the other hand, a few works concentrate on QoS-aware scheduling at individual links for point-to-point space communication [14, 79, 83]. The DTN architecture [11] centers on a new end-to-end message-oriented overlay protocol called Bundle Protocol, which is a middleware between the application layer and the transport layer. The bundle protocol developed for delay-tolerant networks supports class-of-service [2], which is reliability-centric, aiming to ensure correct data transmission but does not support delay constraints. In addition, delay budget is considered in [74], which mainly concerns incentive provisioning based on pairwise tit-for-tat (TFT). Note that very few of the existing works on DTN routing consider the problem of providing QoS to ensure the ongoing communications. [68] presents a method called Delay-Differentiated Gossiping to assure a certain probability of meeting the packets delay requirements while using as little network resources as possible. The idea adapts a set of forwarding probabilities and time-to-live parameters to control the usage of network resources based on how the delay requirements are being met. The DTN routing scheme considers only one base node, while the other nodes have only one destination. [74] proposes the use of pair-wise tit-for-tat (TFT) to develop an incentive-aware routing protocol that allows selfish nodes to maximize their own performance while conforming to TFT constraints. Due to different networking and application settings, they are not readily applicable in this dissertation.

CHAPTER 3: Efficient Data Query in Mobile Opportunistic Networks

In this chapter, I address the problem of how to enable efficient data query in a mobile opportunistic network, formed by mobile users who share similar interests and connect with one another by exploiting Bluetooth and/or WiFi connections. The data query in mobile opportunistic networks faces several unique challenges including opportunistic link connectivity, autonomous computing and storage, and unknown or inaccurate data providers. The goal is to determine an optimal transmission strategy that supports the desired query rate within a delay budget and at the same time minimizes the total communication cost. To this end, I develop a distributed data query protocol for practical applications. To demonstrate the feasibility and efficiency of the proposed scheme and to gain useful empirical insights, I carry out a testbed experiment by using 25 off-the-shelf Dell Streak tablets for a period of 15 days. Moreover, extensive simulations are carried out to learn the performance trend under various network settings, which are not practical to build and evaluate in laboratories.

3.1 System Overview

An individual mobile opportunistic network is incomparable with online social networks in terms of the population of participants, the number of social connections, and the amount of social media. However mobile opportunistic networks gain significant value by serving as a supplement and augment to online social networks and by effectively supporting local community-based ad-hoc social networking. For example, it helps discover and update social links that are not captured by online social networks and allows a user to query localized data such as local knowledge, contacts and expertise, surrounding news and photos, or other information that people usually cannot or do not bother to report to online websites but may temporarily keep on their portable devices or generate upon a request.

This chapter addresses the problem of how to enable efficient data query in mobile opportunistic networks. Consider a mobile opportunistic network with N nodes. Each node can be a query issuer or a data provider, or more commonly act in both roles for different query requests. The queries fall into C categories. Each node has certain expertise to answer a query. Let E denote the expertise matrix, where E_i^c indicates the expertise of Node i to answer a query in Category c, i.e., the probability that Node i can provide a satisfactory answer to a query in Category c. A query is created by a query issuer. It is delivered by the network toward the nodes that can successfully provide an answer (i.e., data providers). If a data provider receives the query, it sends the query reply to the query issuer.

The goal is to determine an optimal transmission strategy that supports the desired query rate and at the same time minimizes the total communication cost.

3.2 Unique Challenges

The use of free, short-range radio is highly desired for a diversity of mobile opportunistic network applications. At the same time, however, it results in a distinctive communication paradigm characterized by intermittent link connectivity and autonomous computing and storage. More specifically, the data query in mobile opportunistic networks faces the following unique challenges.

(1) Opportunistic link connectivity: The connectivity of mobile opportunistic networks is very low and intermittent, forming a sparse network where a node is connected to other nodes only occasionally. This is in a sharp contrast to online social networks, where users always have reliable Internet connections. The data delivery delay in mobile opportunistic networks is potentially long, due to the loose connectivity among nodes. Fortunately, such delay, thought not desirable, is usually tolerable by many data query applications in mobile

opportunistic networks that are often delay insensitive.

(2) Autonomous computing and storage: Central servers are employed to store and process user data in online social networks. Such servers are, however, no longer available in mobile opportunistic networks, where individual portable devices must perform distributed data storage and computation. It is well known that portable devices have limited computing, storage and energy capacity. Nevertheless, such constraints are particularly disadvantageous to mobile opportunistic networks, because a node must process data in a distributed manner and store them locally for a much longer time before sending them to another node, due to intermittent connectivity.

(3) Unknown or inaccurate expertise: When a node issues a query, it is often unaware of the nodes that have sufficient expertise to answer the query, since the cost is prohibitively high to construct a structure to index data and data providers like P2P networks. It is obviously inefficient either to frequently flood queries, which are expensive and often considered spams. Worse yet, in practice, a mobile node hardly knows its probability to answer queries in each category precisely. It may initially claim its expertise based on the mobile user's social roles and available resources. But such initially claimed expertise is often inaccurate.

3.3 Proposed Data Query Scheme

While mobile opportunistic networks offer interesting opportunities to support ad hoc data query, its capacity is unsurprisingly low compared to many other data networks due to its extremely limited and nondeterministic communication opportunities. A distributed data query protocol is proposed, aiming to enable highly efficient ad hoc query under practical mobile opportunistic network settings.

3.3.1 Protocol Design

I introduce a distributed protocol for the data query in mobile opportunistic networks. It is based on two key techniques. First, it employs "reachable expertise" as the routing metric to guide the transmission of query requests. Second, it exploits the redundancy in query transmission.

3.3.1.1 Routing Metric

The delivery of query depends on a routing metric, which is updated routinely and maintained separately from the routing algorithm itself. I first introduce such a metric, i.e., reachable expertise, that guides query transmission.

Each node has certain expertise to answer a query. Let E_i^c denote the expertise of Node i to answer a query in Category c. In practice, it is nontrivial to properly define the expertise, because a mobile node hardly knows precisely its probability to answer queries in each category. It may initially claim its expertise based on the mobile user's social roles (e.g., professions), interests, and available resources. But such initially claimed expertise is often inaccurate. Therefore, after initialization, the expertise should be updated according to the feedbacks from other nodes, especially the query issuers.

In this dissertation, I adopt the exponentially weighted moving average (EWMA) to maintain and update expertise. More specifically, I have

$$E_i^c \leftarrow (1-\mu)[E_i^c] + \mu F_i^c, \tag{3.1}$$

where $0 \leq \mu \leq 1$ is a constant weight to keep partial memory of historic status, $[E_i^c]$ is the expertise before it is updated, and F_i^c is the feedback score for queries that Node i has answered in Category c. Various feedback rating schemes can be adopted to determine F_i^c. In this dissertation, I employ a simple scheme, which supports quick convergence as to be discussed in Sec. 3.4 and shown in Fig. 3.1h.

The expertise indicates the capability of a node to answer queries, but itself is insufficient to guide query transmission. For example, a node may have high expertise, but is not reachable by the query issuer and thus becomes less helpful to answer the query. To this end, I define k-hop reachable expertise. As discussed earlier, the mobile opportunistic network users can often tolerate long delay. Thus, the delay random variables may be reduced to simple nodal contact probabilities. Let p_{ij} denote the probability that Nodes i and j meet. The maintenance and update of p_{ij} have been discussed extensively in DTN networks [4, 21, 44, 76]. The k-hop reachable expertise is calculated as follows:

$$E_i^c(k) = 1 - \prod_{j \in \Phi}(1 - p_{ij} \cdot E_j^c(k-1)), \tag{3.2}$$

where Φ is the set of nodes that Node i meets frequently. $E_i^c(k)$ indicates the probability that Node i can deliver the query within k hops to a node that can answer the query. Clearly, $E_i^c(0) = E_i^c$ and $E_i^c(1) = 1 - \prod_{j\in\Phi}(1 - p_{ij} \cdot E_j^c)$. Node i collects $\{E_j^c(k-1) | \forall\ j \in \Phi\ and\ 0 < k \leq N\}$ whenever it meets other nodes, and periodically makes an update on $E_i^c(k)$ according to Eq. (3.2).

Based on the k-hop reachable expertise, I define the aggregated reachable expertise,

$$AE_i^c = 1 - \prod_{k=1}^{N}(1 - E_i^c(k)), \tag{3.3}$$

which indicates the overall ability of Node i to help a query in Category c to be answered. It serves as the routing metric to guide query delivery as to be discussed next.

3.3.1.2 Routing with Dynamic Redundancy Control

Based on the routing metric, i.e., reachable expertise, I now introduce the routing algorithm. The delivery of a query is guided by the aggregated reachable expertise, where the query is generally forwarded from the node with a lower aggregated reachable expertise to the node with a higher one. In contrast to the conventional store-and-forward data transmission where a single copy of data is transmitted across the network, redundancy is often employed in opportunistic networks. Generally speaking, the higher the redundancy, the higher probability the query is answered successfully. However, redundancy must be properly controlled as excessive redundancy may exhaust network capacity and thus degrade the performance.

A naive approach is to create a fixed amount of redundancy for each query. For example, a predetermined number of copies of the query can be created and distributed to other nodes in the network. This approach, however, is often inefficient, because the effectiveness of redundancy depends on the nodes that receive, carry and forward the query. In an extreme case, all redundant copies of the query may be transmitted and carried by the nodes that have little chance to meet the node(s) that can answer the query and thus become ineffective. As a matter of fact, the effectiveness of redundancy highly depends on the reachable expertise of the nodes that carry the redundant copies. To this end, I introduce a parameter to dynamically reflect the effective redundancy.

More specifically, the proposed routing algorithm with dynamic redundancy control is outlined below. Let R_i^q denote the redundancy of Query q as observed by Node i. The parameter is the estimated probability that at least one copy of the query is answered by any other nodes in the network. It is maintained and updated in a distributed way. Assume Query q is in Category c. R_i^q is initialized to zero when the query is created and subsequently updated during its transmission. Since communication opportunity is low, transmission is often between two nodes only. If more than two nodes are within communication range, I assume an underlying medium access control protocol that randomly selects one node as the sender. Therefore I consider a general scenario where Node i meets Node j in the following discussions.

First, Nodes i and j exchange their k-hop reachable expertise and update their aggregated reachable expertise according to Eqs. (3.2) and (3.3).

Then, Node i fetches the query with the lowest redundancy in its queue. The queue holds the queries that Node i creates or receives from other nodes. It is sorted according to the redundancy level such that the query with the lowest redundancy (denoted as Query q in Category c) is at the head of the queue. If Node j has a high expertise for queries in Category c (i.e., $E_j^c \geq \alpha$ where α is a predefined constant), it directly answers the query by creating and sending a query reply to the query issuer. Since the destination of the query reply (i.e., the query issuer) is known, it can be delivered via any existing routing protocol for opportunistic networks [4, 5, 10, 17, 20, 21, 27, 33, 39, 43, 44, 46, 52, 55, 59, 60, 61, 77, 80, 81, 85, 86, 87]. I adopt the scheme proposed in [44] in the implementation. Note that the answer of Node j is not always satisfactory. It has a probability of E_j^c to be satisfied by the query issuer. Therefore, Node i removes the query from its queue with a probability of E_j^c.

Otherwise, if Node j cannot answer the query directly (i.e., $E_j^c < \alpha$), Node i checks the redundancy of Query q. If $R_i^q \geq \beta$ where β is the desired query delivery probability, it implies that high enough redundancy has been created for the query. Thus no action will be taken. Node i simply holds the query until it meets a node that can directly answer the query or the query must be dropped due to queue overflow. An overflow happens when a new query is added into the queue which is already full. In this case, the query with the highest redundancy (i.e., at the end of the queue) is dropped.

If $R_i^q < \beta$, the query should be further propagated. But it is transmitted to Node j only if $AE_i^c < AE_j^c$ (i.e., the latter has a better chance to deliver the query). This transmission creates two copies of the query, each sharing partial responsibility to get the answer. The redundancies for the two copies are assigned as follows:

$$R_j^q \leftarrow 1-(1-R_i^q)(1-E_i^c(0)), \tag{3.4}$$

$$R_i^q \leftarrow 1-(1-R_i^q)(1-E_j^c(0)). \tag{3.5}$$

In both formulas, $(1-R_i^q)$ denotes the estimated probability that none of other nodes (except Nodes i and j) can get the answer for Query q, and $(1-E_i^c(0))$ and $(1-E_j^c(0))$ give the probability that Node i and Node j cannot directly answer the query. Therefore the updated R_i^q (or R_j^q) indicates the probability that at least one copy of the query can be answered by other nodes except Node i (or Node j).

The transmission of Query q continues upon future communication opportunities until, as discussed earlier, the query is answered by a node or dropped due to queue overflow. Upon receiving the query reply, the query issuer evaluates it and constructs a feedback packet, which is delivered to the node that answers the query, again, via an existing routing protocol for opportunistic networks. The latter then updates its expertise according to Eq. (3.1).

3.4 Prototype and Experiment

To demonstrate the feasibility and efficiency of the proposed data query protocol and to gain useful empirical insights, I have carried out a testbed experiment using off-the-shelf Dell Streak tablets. In this section, I first introduce the testbed setup and then present experimental results.

3.4.1 Prototype and Testbed Setup

I have developed a prototype system by using Dell Streak 5 and 7 tablets that are of the smartphone/tablet PC hybrid operating on Android 2.2. The communication between the tablets is enabled via Bluetooth. A Streak tablet has 16 GB internal storage adequate to keep large amounts of experimental data. I have implemented the proposed data query protocol by using standard Android APIs, closely following the description in the previous section. In order to save power, each node initiates neighbor discovery once every a random interval (between 5 to 10 minutes).

The experiment involves twenty five volunteers including faculty members and students. They are marked as Node 0 to Node 24. In the experiment, I define three categories of queries, i.e., history, science, and arts (which are named Category 0, 1, and 2, respectively). Each participant has a claimed initial expertise for answering queries in each category and generates twelve queries per day in randomly chosen categories. Note that the initial expertise is not accurate. The true expertise is arbitrarily set by letting a small set of nodes to have an expertise of 1 to answer queries in each category. More specifically, Nodes 3, 13, and 19 can answer queries in Category 0; Nodes 4, 14, and 20 can answer queries in Category 2; and Node 1 can answer queries in any category. Other nodes initialize expertise to be (0-1) randomly and learn and update their aggregated expertise during the experiment. The experiment had run for fifteen days, starting from Monday 16:00 p.m. in the first week to Monday 17:00 p.m. in the third week.

I compare several variations of the proposed scheme and related schemes. In the following discussions, 0-hop, 1-hop, 2-hop, and 3-hop stand for the proposed scheme that allows up to 0 hop, 1 hop, 2 hops, and 3 hops relaying, respectively; No Feedback means the

Table 3.1: Results under Experiment.

	Query Rate	Delay	Ave. Replies	Ave. Copies
0-hop	0.34	4.56h	10	1
1-hop	0.73	7.52h	30	2
2-hop	0.96	8.83h	42	2
3-hop	0.86	9.42h	56	4
No Feedback	0.83	9.12h	50	3
Flooding	0.39	4.68h	13	7
Gossip	0.52	7.28h	21	5
Willingness	0.51	6.23h	17	6
Spray and Wait	0.46	5.67h	12	6
Social-based	0.68	9.68h	58	5

proposed scheme (with 2-hop relay) but without the feedback mechanism to rectify expertise; Flooding is the simple flooding scheme for query delivery; Gossip [34] considers multiple categories and assigns the queries in each category a transmission probability for data transmission; Willingness [36] is a scheme that a query is delivered based on willingness, which is the degree to which a node actively engages in trying to re-transmit a query; Spray and Wait [80] is considered as a baseline opportunistic delivery protocol; Social-based [100] is a social-based routing scheme.

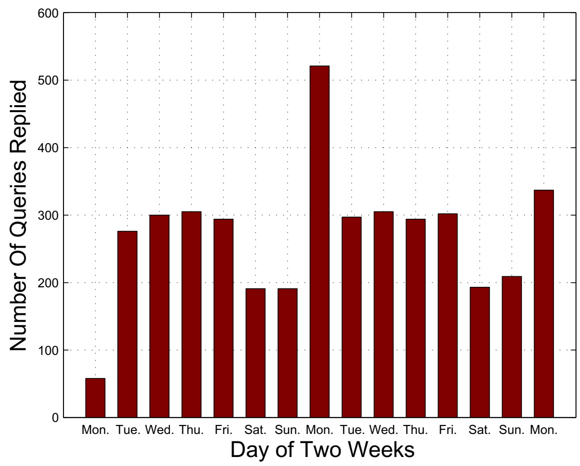
I am primarily interested in two parameters: (1) the success query rate, i.e., the ratio of successfully answered queries to the total generated queries, and (2) the query delay which is the period from the time when a node generates the query to the time when it receives the answer.

3.4.2 Experimental Results

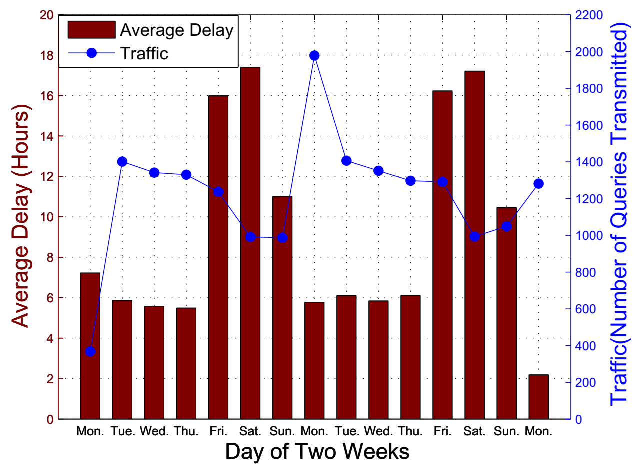
Table 3.1 shows the overall performance of different schemes. The 2-hop scheme achieves the highest query rate. It is not surprising to find the 1-hop scheme with a lower query rate since a node merely tries to answer the queries via up to one hop relay. The 0-hop scheme has the lowest query rate as a query can be answered only when the query issuer meets the data provider directly. On the other hand, it seems anti-intuitive that allowing a longer relay path (e.g., the 3-hop scheme) leads to a negative gain. But this is reasonable because excessive redundancy is created when too many nodes are involved in relaying queries, subsequently overloading the network and resulting in degraded performance. For a similar reason, the Flooding scheme has a even lower query rate given its extremely high redundancy. Under the No Feedback scheme, the inaccurate expertise is not rectified, resulting in misleading reachable expertise and thus lower query rate. Gossip [100] considers multiple categories and assigns the queries in each category a transmission probability for data transmission. However, as a gossiping approach, its data transmission is randomized. Therefore a query is often answered and carried by nodes with insufficient expertise, thus inducing many non-satisfactory replies. Willingness [36] is a scheme that a query is delivered based on willingness, which is the degree to which a node actively engages in trying to re-transmit a query. The willingness does not reflect the expertise based on which a node replies queries, therefore the nodes are not helpful for each other to carry queries to nodes with sufficient expertise. I also compare with Spray and Wait [36] which is considered as a baseline opportunistic delivery protocol. [36] fixes the number of copies for each query which limits the queries to go through correct paths to be replied by nodes with sufficient expertise, making query rate even lower. Social-based [100] exploits a distributed community partitioning algorithm to divide a DTN into smaller communities. For intra-community communication, a utility function convoluting social similarity and social centrality with a decay factor is used to choose relay nodes. For inter-community communication, the nodes moving frequently across communities are chosen as relays to carry data to destinations efficiently. Although [100] introduces a solution for DTNs which leverages social properties and mobility characteristics of users, it is not truly applicable for the data query in mobile opportunistic networks, because when a node issues a query, it is often unaware of the nodes that have sufficient expertise to answer the query. The cost is prohibitively high to construct a structure to index data and data providers like P2P networks. It is obviously inefficient either to frequently flood queries, which are expensive and often considered spams. It is not surprising that the proposed scheme has better performance than Social-based, since Social-based does not capture the inherent features for the query delivery in mobile opportunistic networks, hence the nodes are not helpful for each other by making the correct decisions to carry queries to satisfactory nodes.

In general, when more hops are allowed in relaying queries, the overhead increases, because a query is more aggressively propagated. As a result, more copies of the query are transmitted in the network and the query issuer often receives more replies. At the same time, since a query may potentially go through a longer path to reach the data provider, the average delay also increases. Compared with the 2-hop scheme, No Feedback has longer delay and more number of replies because incorrect expertise often leads the queries to wrong routes.

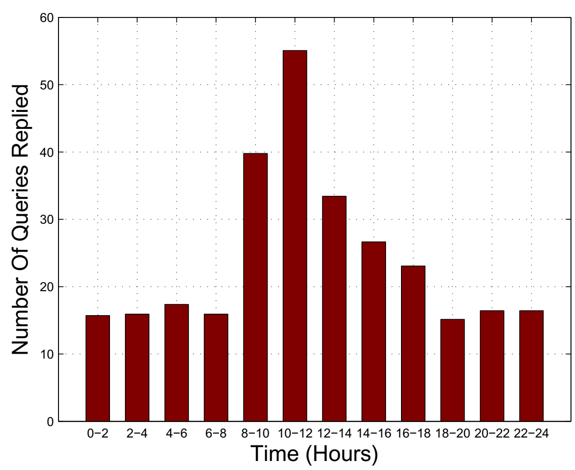
More than 96% queries are answered successfully. The unanswered queries are all generated during the final hours of the experiment. Fig. 3.1a illustrates the number of queries answered on each day of the experiment. As can be seen, the results vary among days, reflecting the moving patterns of the participants. More queries are answered during weekdays than weekends due to the lower interactive activities of students and faculty on Saturday and Sunday. In fact, many queries cannot be answered during weekends and have to wait until Monday of the next week. This explains the peak on the second Monday. It is worth mentioning that the first and the third Monday are not the whole days, hence the number of



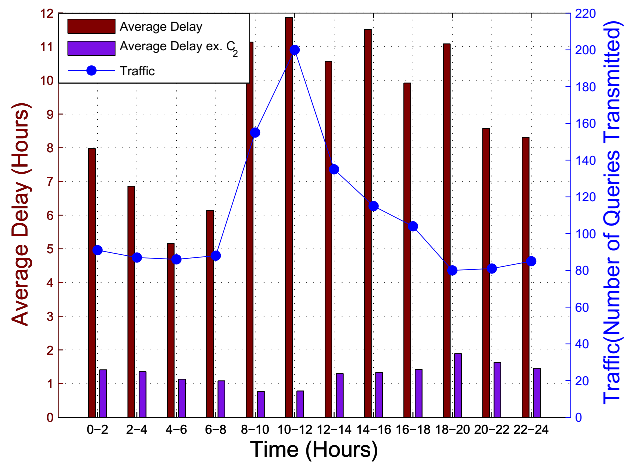
(a) Query replies (daily).



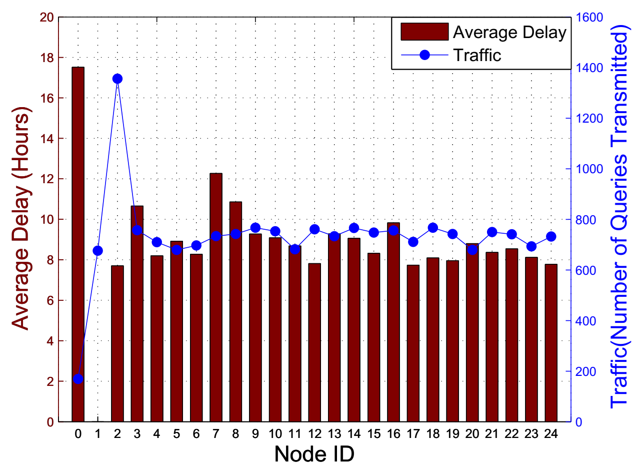
(b) Delay and traffic (daily).



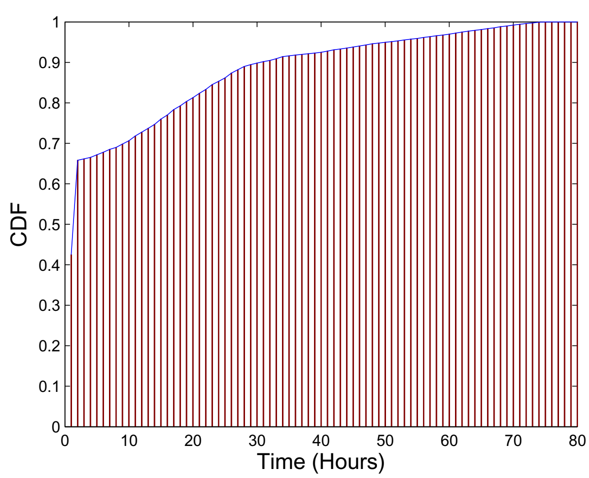
(c) Query replies (hourly).



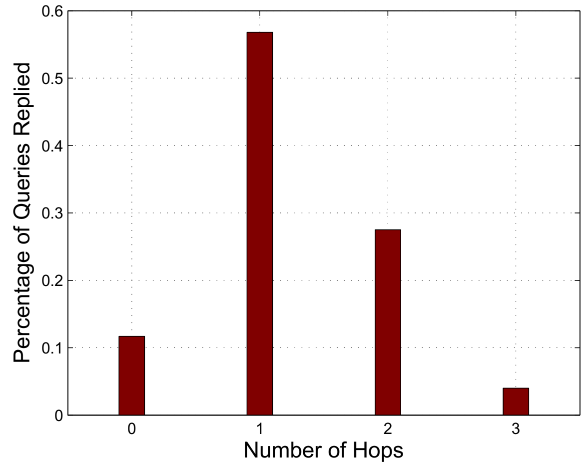
(d) Delay and traffic (hourly).



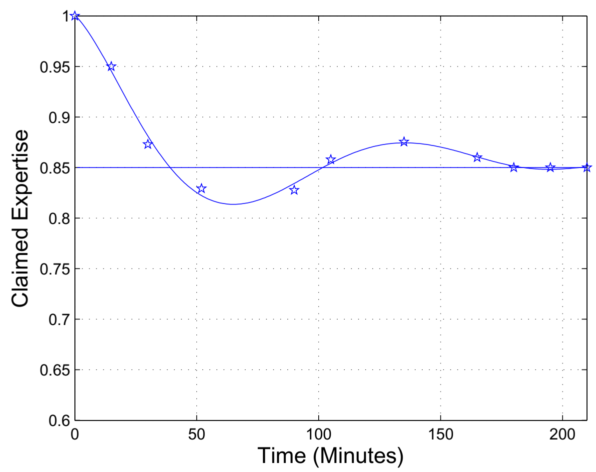
(e) Delay and traffic (nodes).



(f) Delay distribution.



(g) Path length distribution.



(h) Convergence of expertise.

Figure 3.1: Experimental results.

answered queries is less than the second Monday. The activity pattern is also evidenced by the delay variation shown in Fig. 3.1b. Queries generated in weekends have longer delay compared with those in weekdays. The delay of queries generated on Friday is also high because no classes are scheduled on Friday afternoon and many offices are closed after 1:00 p.m.. Fig. 3.1b also shows the total traffic in the network, which follows a similar pattern of nodal activities.

Figs. 3.1c and 3.1d further zoom in to show the results in each hour of a day. The data are averaged over 15 days. Both the network traffic and the number of answered queries are high during daytime and low at night, which again shows the query heavily depends on the activity of students and faculty who carry the tablets. Likewise, I expect lower delay during daytime. However, the results are just the opposite (as depicted in Fig. 3.1d). Such anti-intuitive observation is due to the queries from a few nodes, which experienced extremely long delay that dominates the overall performance. In fact, most queries generated during daytime indeed have short delay. But a set of nodes (including Nodes 3-12) rely on a single node (Node 2) to carry their queries in Category 2 to corresponding data providers. Such delivery happens around 9:00 a.m. daily. The queries generated after 9:00 a.m. must wait until the next day, thus inducing unusually long latency that significantly elevates the overall average. If I exclude such queries (see the lower purple bars in Fig. 3.1d), the average delay becomes much lower, and the daytime delay is generally shorter than that during night.

The average delay and traffic of different nodes are illustrated in Fig. 3.1e. In general, delay and traffic vary among different nodes due to the randomness in nodal mobility and query generation and transmission. Node 0 has extremely poor connectivity (either directly or indirectly) to the nodes with high expertise, resulting in very long delay compared with other nodes. Contrarily, since Node 1 is able to answer all the queries of three categories, it has the minimum delay. In addition, Node 2 has the heaviest traffic load because it frequently meets other nodes, while Node 0 carries the least traffic due to few interactions between it and other nodes.

The delay distribution is shown in Fig. 3.1f. More than 65% queries are answered within two hours. The queries with longer delays are either generated by Nodes 3-12 as discussed above, or generated during weekends and thus cannot be replied until the next Monday.

Fig. 3.1g illustrates the distribution of path length. All queries are answered within three hops. Fig. 3.1h shows the convergence of the claimed expertise to the ground truth. A node is chosen as an example, while similar results are observed in other nodes as well. As can be seen, the feedback mechanism effectively adjusts the node's expertise, gradually approaching to the true value within a few hours.

3.5 Simulation Results

Besides the experiment discussed above, extensive simulations are carried out to learn the performance trend of the proposed data query algorithm under various network settings, which are not practical to evaluate by using lab equipments. The simulation codes are extracted from the prototype implementation, and the simulation results are obtained under real-world traces and power-law mobility model. Each node maintains a maximum queue size of $1,000$.

3.5.1 Simulation under Haggle Trace

I have evaluated the proposed scheme under several real-world traces available at CRAWDAD. Table 3.2 shows the results based on Haggle trace [71], which includes 98 participants carrying small devices (iMotes) during Infocom 2006. I run the simulation for a period of $342,916$ seconds (or about 4 days). Each node generates 1.08 queries per hour. The queries fall into five categories, and each category is associated with three expert nodes that can provide satisfactory replies. Similar to the results in Table 3.1, Table 3.2 shows the results under Haggle trace.

The distributions of query rate and delay are illustrated in Fig. 3.2. About 90% of nodes can achieve a query rate of 80% or higher under the proposed scheme. At the same time, more than half of the queries are answered within an hour.

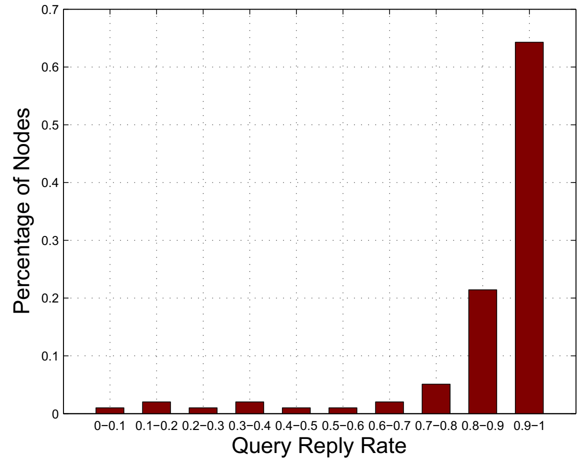
Table 3.2: Results under Haggle Trace.

	Query Rate	Delay	Ave. Replies	Ave. Copies
0-hop	0.31	1.58h	9	1
1-hop	0.68	2.62h	26	2
2-hop	0.88	3.06h	39	3
3-hop	0.79	3.25h	52	4
No Feedback	0.76	3.16h	46	3
Flooding	0.36	1.61h	12	6
Gossip	0.46	2.56h	20	5
Willingness	0.45	2.16h	16	5
Spray and Wait	0.39	1.98h	11	5
Social-based	0.62	3.32h	56	4

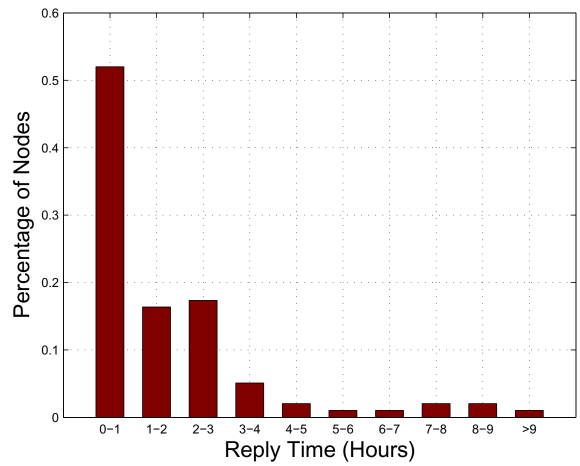
3.5.2 Simulation under Power-Law Mobility Model

Besides the above results based on Haggle trace, I have carried out a simulation under power-law mobility model, which enables convenient study of performance trend with the variation of several network parameters. More specifically, I simulate an area that is partitioned into a grid of 20×20 cells. Each node is associated with a randomly-chosen home cell, in which it initially resides. In a time slot, it may move in one of the four directions, i.e. up, down, left, and right, or stay in its home cell. Let $P_i(x)$ denote the probability for Node i to be at Cell x. $P_i(x) = k_i(\frac{1}{d_i(x)})^{\sigma}$ where k_i is a constant, σ is the exponent of the power-law distribution and $d_i(x)$ denotes the distance between Cell x and Node i's home cell. The default network parameters include a network of 100 nodes, a σ of 2, 10 categories of queries, 5 experts per category, and a generation rate of 0.02 queries per time unit per node.

In an opportunistic network, the communication capacity highly depends on the meeting opportunities among mobile nodes. As shown in Fig. 3.3a, query reply rate grows with the increase of the network density, because the nodes have more opportunities to meet each other and exchange their queries. Fig. 3.3b depicts the impact of the power-law factor σ. When σ is large, the nodes tend to stay in their home cells, i.e., have low mobility, resulting in small



(a) Query rate distribution.



(b) Delay distribution.

Figure 3.2: Distributions under Haggle trace.

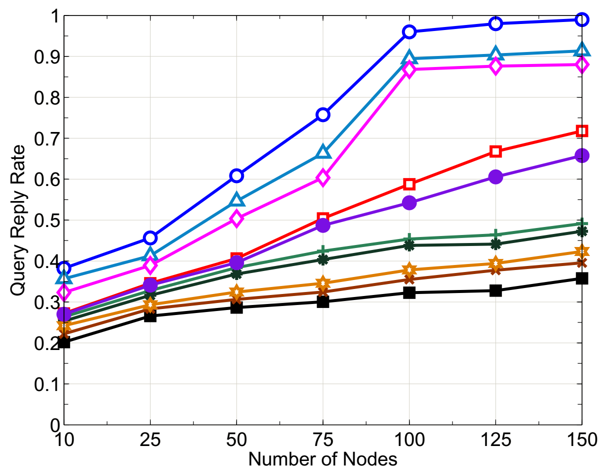
probabilities to meet each other and consequently small network capacity. Therefore the query reply rate is low. When σ is extremely large, the query reply rate may approach as low as zero. On the other hand, when σ is small, the nodes have uniform mobility, i.e., similar probabilities to visit all cells and accordingly similar routing metric (i.e., k-hop reachable expertise), rendering routing ineffective. Under the simulated network setting, $\sigma = 2$ results in the best performance.

The impact of traffic load is illustrated in Fig. 3.3c. While the query reply rate keeps stable at the beginning under all schemes, it starts to drop when the generation rate exceeds 0.03. In general, with a higher query generation rate, the overall traffic load increases, resulting in more frequent queue overflow and consequently lower query reply rate.

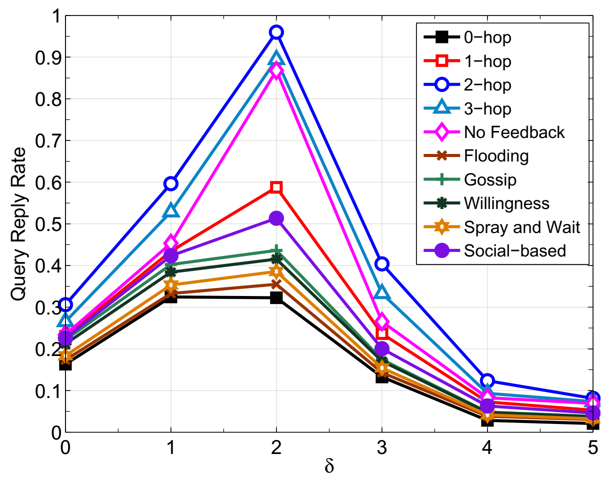
As discussed in Sec. 3.3.1, a threshold β is employed for dynamic redundancy control. A larger β allows more redundancy to be created, aiming to achieve a higher query reply rate. However, if β is too large, the excessive redundancy degrades the utilization of communication and storage resources and lowers the overall performance accordingly (see the Fig. 3.3d). Fig. 3.3e shows that a higher query reply rate is achieved with the increase of queue size, because more queries and replies can be kept in the queue until they are delivered. The number of experts for each category is also studied in this dissertation. As shown in Fig. 3.3f, more experts for a category result in higher query reply rate because more nodes can answer the queries in this category.

3.6 Conclusion

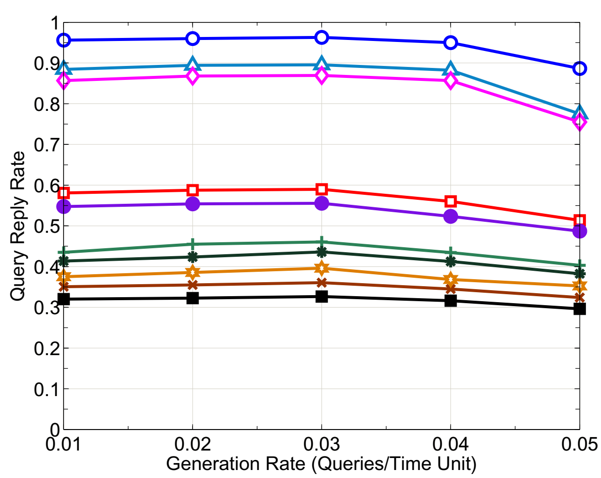
I have studied the problem of data query in a mobile opportunistic network, aiming to determine an optimal transmission strategy that supports the desired query rate within a delay budget and at the same time minimizes the total communication cost. I have developed a distributed data query protocol for practical applications. To demonstrate the feasibility and efficiency of the proposed scheme and to gain useful empirical insights, I have carried out a testbed experiment by using 25 off-the-shelf Dell Streak tablets for a period of 15 days. Moreover, I have run extensive simulations to learn the performance trend under various network settings, which are not practical to build and evaluate in laboratories.



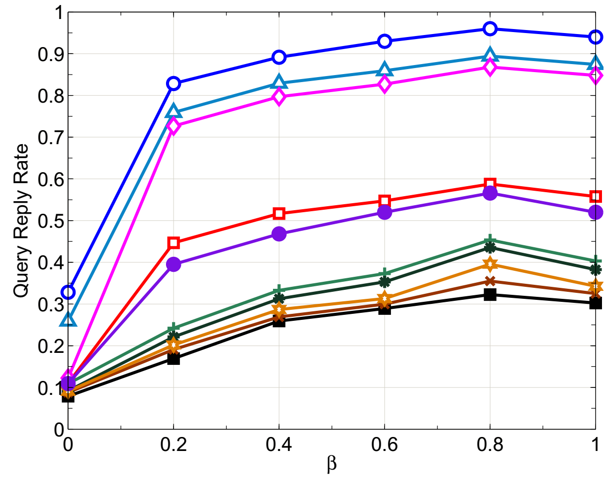
(a) Network density.



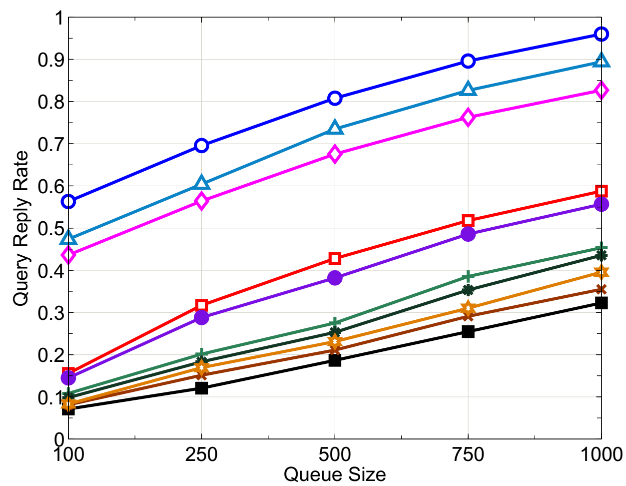
(b) Nodal mobility (σ).



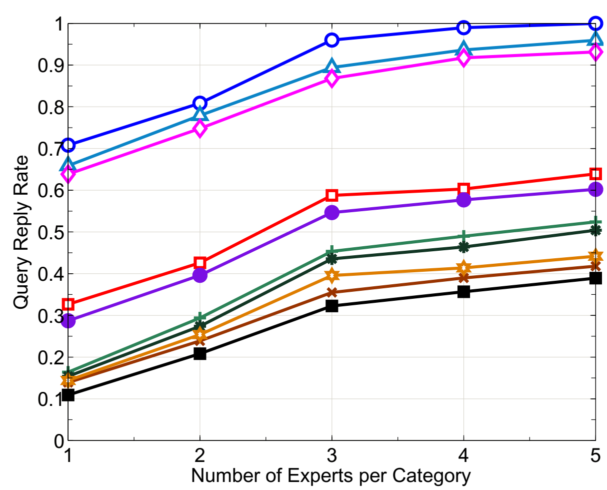
(c) Traffic load.



(d) Redundancy threshold.



(e) Queue size.



(f) Number of experts/category.

Figure 3.3: Performance trend under power-law mobility model.

CHAPTER 4: Efficient QoS Support in Mobile Opportunistic Networks

In this chapter, I aim to support Quality-of-Service (QoS) provisioning, especially the guarantee for end-to-end data delivery delay, in mobile opportunistic networks. The QoS-aware delivery probability (QDP) is introduced to reflect the capability of a node to deliver data to a destination within a given delay budget. Each node maintains a set of QDPs to make autonomous decisions for QoS-aware data transmission. At the same time, a prioritized queue is employed by each mobile node. In order to support efficient prioritization and redundancy control, the priority is determined by a function of traffic class and data redundancy. The former is pre-determined by the corresponding application, while the latter is dynamically estimated during data delivery. Two experiments are carried out to demonstrate and evaluate the proposed QoS-aware data delivery scheme. The first experiment involves multiple clusters of static Crossbow sensors that are connected by air and ground mobile nodes with controlled mobility. The second experiment is under a mobile social network setting, where 23 Dell Streak Android tablets are carried by volunteers with arbitrary and diverse mobility patterns during a period of two weeks. Moreover, simulation results are obtained under DieselNet trace and power-law mobility model to study the scalability and performance trend. The experiments and simulations demonstrate that the proposed scheme achieves efficient resource allocation according to the desired delay budget, and thus supports effective QoS provisioning.

4.1 Introduction

Mobile opportunistic networks are characterized by intermittent and nondeterministic connectivity, often due to interruptible wireless links, sparse network deployment, and/or nodal mobility. Such opportunistic networking has been discussed in the context of delay/disruption-tolerant networks, sporadically connected sensor networks, vehicular networks, and peer-to-peer mobile social networks [2, 9, 20, 27, 86, 95, 96]. How to discover and utilize opportunistic communication resources for efficient data transmission has been one of the central research issues in such networks as evidenced by extensive discussions in the literature [2, 4, 9, 10, 17, 20, 24, 27, 32, 37, 38, 39, 43, 44, 46, 48, 50, 55, 56, 60, 61, 62, 64, 65, 66, 73, 74, 77, 81, 84, 86, 87, 89, 90, 95, 96, 99]. However, limited prior work has addressed Quality-of-Service (QoS). While long data delivery delay is generally unavoidable given the unique intermittent connectivity, QoS, especially the guarantee for end-to-end delivery delay, is highly desired in a variety of applications. For example, the dissemination of a data message (such as an advertisement or coupon [27, 63, 67, 96]) in a mobile social network must meet a delay budget no longer than its expiration date, and different data messages are often associated with different delay budgets. Separately, in wildlife tracking applications, interactive control and event report must be delivered within a short end-to-end delay bound, as opposed to routine transmissions of ambient environmental data that can tolerate long delay [2]. Data delivered beyond their delay budgets often lead to reduced or completely forfeited value.

4.1.1 Challenges in QoS Provisioning in Mobile Opportunistic Networks

QoS has been extensively studied in wireless networks [13]. However, there are unique challenges to support QoS in an opportunistic communication setting. First of all, due to the nondeterministic connectivity, it is intrinsically infeasible to provide hard guarantee of end-to-end delivery delay. Thus, a probability-based delay budget is introduced in this dissertation. More specifically, let $Q_m(\delta, \gamma)$ denote the desired QoS of Message m, which must be delivered to its destination within δ time units with a probability no less than γ.

Second, since end-to-end paths often do not exist in the network, a routing decision must be made based on predicted future connections. To this end, temporal and/or spatial information in nodal contacts is exploited by mobile nodes to estimate their probabilities to deliver data to corresponding destinations [4, 27, 86]. Such delivery probability serves as a routing metric to guide data transmission, where a data message is always forwarded to nodes with higher delivery probabilities. However, most prior studies do not consider delay budget. Therefore the delivery probabilities may become misleading for QoS support. For example, a node with a high delivery probability to a destination may in fact experience long average delay, thus deceptively attracting many data messages by following the routing scheme described above but frequently failing to meet the desired QoS requirements.

Third, the QoS priority associated with a data message is static, i.e., does not change during its transmission, in conventional networks. However redundancy is commonly employed in opportunistic networks for dealing with high data loss probability and achieving desired delivery rate. Consequently, the importance of a message varies during its transmission depending on the amount of redundancy created. For example, a newly generated message is the sole copy, and should be processed with high priority and protected from being lost. When multiple copies of the message are produced during its transmission, deferring or losing a copy would not significantly degrade the delivery probability of the message. In general, data messages that are in the same traffic class may have diverse redundancy (even if they were created by the same node at the same time) and accordingly should be associated with different priority levels. Therefore, data messages must be prioritized not only by QoS requirement but also according to their dynamically changing redundancy.

4.2 Proposed QoS-Aware Delivery Scheme

To address the unique challenges in QoS provisioning in opportunistic networks with intermittent and nondeterministic network connectivity, I propose a QoS data communication scheme based on QoS-aware delivery probability and adaptive queue prioritization as outlined below. The former serves as the QoS routing metric, which guides a data message through the best routing path that meets the desired QoS requirement with high probability. The latter supports efficient resource utilization by proper redundancy control.

4.2.1 QoS-Aware Delivery Probability (*QDP*)

As discussed in Sec. 4.1.1, I adopt a probability-based delay budget, denoted by $Q_m(\delta,\gamma)$, as the QoS metric, demanding Message m to be delivered to its destination within δ time units with a probability no less than γ. A node often has a number of messages in its data queue. It may transmit a message directly to the destination or to an intermediate node which subsequently continues to forward the message directly or indirectly to the destination. When a node meets another node, the former needs to decide whether to transmit a message to the latter. Such a routing decision must be made based on a QoS-aware routing metric, which indicates if the latter has a higher probability to deliver the message to its destination within the delay budget. To this end, I introduce a new routing metric for QoS provisioning in mobile opportunistic networks, dubbed QoS-aware delivery probability (*QDP*).

Since a data message may be associated with any arbitrary delay budget and any destination, it is imperative for a node to maintain a set of *QDPs* to make autonomous decisions for QoS provisioning. Let $P_i = \{p_i^k(t)|0 \leq t \leq \infty, k \in \Phi\}$ denote the *QDPs* of Node i, where $p_i^k(t)$ is the probability that a message can be delivered from Node i to Node k within t time units, and Φ is the set of mobile opportunistic network nodes. For a given k, $p_i^k(t)$,

$0 \leq t \leq \infty$, is intrinsically the cumulative distribution function of delivery delay, which is ideal to support QoS data delivery, but impractical to maintain in continuous time. Thus, a finite set of discrete delays, denoted by Υ, is employed, arriving at $P_i = \{p_i^k(t) | t \in \Upsilon, k \in \Phi\}$.

While *QDP* is defined above, it is obviously challenging to be obtained in a distributed manner, since a node is connected to other nodes only occasionally. With no end-to-end connections, it is extremely difficult, if not impossible, to gain up-to-date global knowledge to compute accurate *QDPs*. But at the same time, I notice that the accurate *QDPs* are, though desired, not imperative. As a matter of fact, the *QDP* can be over- or under-estimated across the network, due to the approximation in *QDP* update. The approximate *QDPs* can effectively support QoS routing as long as they are proportional to the real QoS-aware delivery probabilities, i.e., a node with a truly higher (or lower) QoS-aware delivery probability maintains a higher (or lower) approximate *QDP*. Despite such *QDPs* are inaccurate, they efficiently guide data messages through the best routing paths for QoS provisioning. This observation is verified by the simulations.

To this end, I propose a light-weight distributed algorithm to learn approximate *QDPs*. The overall idea is to let individual nodes maintain their approximate *QDPs*, which are updated based on locally learnt information upon meeting events. Initially, a node only knows the *QDP* with itself as the destination, which is obviously one. It learns the *QDPs* to other destinations via recursive information exchange, during which the QoS delivery probabilities are updated in a ripple manner propagated from the corresponding destinations. More specifically, Node i initializes $p_i^k(t)$ as follows

$$p_i^k(t) = \begin{cases} 1, & i = k \\ 0, & i \neq k, \end{cases} \tag{4.1}$$

and updates them autonomously according to its transmission history, in both direct and cascaded deliveries, as outlined below. Each node divides time into windows. The size of a window can be chosen to be the maximum delay budget interested by the node. The windows of different nodes do not have to be synchronized. This is because each node updates its *QDP* autonomously. It can choose any arbitrary window size and start its window at an arbitrary time instance. The process does not require synchronization among different nodes.

QDPs are updated based on time windows. Let's consider Node i. It maintains a parameter $\xi_i^k(t)$, which is used to calculate the *QDP* of Node i to Node k in each window, for each $k \in \Phi$ and $t \in \Upsilon$. It intrinsically indicates the probability that the message is failed to be delivered to the destination within the delay budget in a time window. In each time window, $\xi_i^k(t)$ is initialized to 1. When Node i meets Node j, it compares $p_i^k(t)$ and $p_j^k(t)$ for every $k \in \Phi$ and $t \in \Upsilon$. If $p_i^k(t) < p_j^k(t)$, the former transmits the corresponding message with Destination k and Delay Budget t to the latter and at the same time updates $\xi_i^k(t)$ as follows:

$$\xi_i^k(t) \leftarrow \xi_i^k(t)(1 - p_j^k(t)), \tag{4.2}$$

where $1 - p_j^k(t)$ is the probability that Node j cannot deliver the message to Destination k within the required delay budget of t. If $j = k$, it is a direct delivery. According to Eq. (4.1), $p_j^k(t) = 1$ and thus $1 - p_j^k(t) = 0$. Otherwise, it is an indirect delivery where Node j may or may not successfully transmit the message to its destination. Consequently, Node i cannot receive a confirmation immediately from Node j. Therefore, it estimates the probability that Node j delivers the message by $p_j^k(t)$.

By the end of the window, Node i calculates its window-based *QDP* as

$$\hat{p}_i^k(t) = 1 - \xi_i^k(t), \tag{4.3}$$

which essentially equals $1 - \Pi(1 - p_j^k(t))$, i.e., the probability that at least one of such transmissions delivers the message to its destination by its delay budget. Clearly, once Node i delivers the message directly to its destination (i.e., Node k), $\hat{p}_i^k(t)$ becomes one and there is no need to further transmit the message.

$\hat{p}_i^k(t)$ is a window-based *QDP*. Its value often varies from window to window, exhibiting undesired instability. It is highly preferable to keep *QDPs* stable, since they are employed to guide data transmission. In this dissertation, I adopt the exponentially weighted moving average (EWMA) to maintain and update *QDPs*. More specifically, I have

$$p_i^k(t) \leftarrow (1-\mu)p_i^k(t) + \mu\hat{p}_i^k(t), \tag{4.4}$$

where $0 \leq \mu \leq 1$ is a constant weight to keep partial memory of historic status. It has been shown in [93] that the above EWMA-based average converges to a constant under statically distributed mobility. $p_i^k(t)$ indicates the probability that Node i delivers a message to Node k within a delay budget of t. Node i performs similar calculation for all k and t to yield the *QDP* matrix $P_i = \{p_i^k(t) | t \in \Upsilon, k \in \Phi\}$.

Every node follows the above algorithm to learn its *QDPs*. I observe in the simulations that, since Node i is only aware of $p_i^i(t)$ (i.e., with itself as the destination) during initialization, *QDPs* are updated in a ripple manner starting from the corresponding destinations. While the simulation details are deferred to Sec. 4.4, Fig. 4.1 illustrates the converged *QDP* indeed reflects the true QoS-aware delivery probability after the warmup period of simulations, thus serving as an efficient routing metric for QoS provisioning.

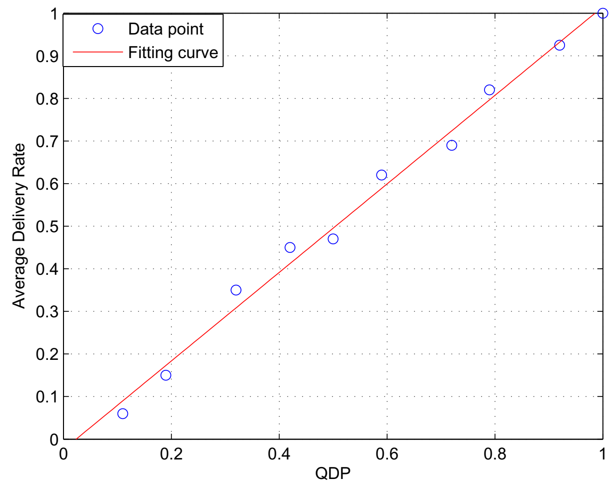


Figure 4.1: *QDP* vs. Delivery Rate (average of $t \in \Upsilon$).

4.2.2 Adaptive Data Queue Prioritization

For the sake of low complexity in queue management, most QoS-aware systems (e.g., IEEE 802.11e) employ multiple FIFO queues, one for each traffic class. In opportunistic networks, however, it is often inevitable and at the same time affordable to deal with more complicated queuing strategies. In contrast to conventional store-and-forward networks where only a single copy of a data message is actively transmitted at a time, redundancy (by either simple duplication or coding) is commonly employed for desired communication performance, especially for QoS support, in opportunistic networks. However, the creation and distribution of redundancy depend on nondeterministic nodal meeting events, thus exhibiting high dynamics. At a given time, data messages in the same traffic class may have very different redundancy. For example, multiple copies may have been created for a message that has been circulated in the network for some time, and thus its preferred delivery probability can be maintained even if a copy is delayed or dropped. On the other hand, a newly created message may be the sole copy that should be absolutely protected from being lost. Consequently, data messages in the same traffic class must be prioritized according to their redundancy level, naturally leading to prioritized queues with a logarithm time complexity for message insertion and removal. Although such computing time is undesired and may become performance bottleneck in conventional networks, it is affordable in opportunistic networks, where a node has plenty of time to manipulate its queue before the next transmission opportunity becomes available.

Based on the above observations, a single prioritized queue is employed by each node. The queue is sorted according to the priority of data messages, which is a function of traffic class and redundancy. The former is pre-determined by the corresponding application and remains unchanged during the transmission of the messages, while the latter is dynamically estimated as to be discussed next. More specifically, let q_i^m denote the priority of Message m in the message queue of Node i. A smaller q_i^m indicates a higher priority. q_i^m is calculated as follows:

$$q_i^m = (1 - \lambda) C_i^m + \lambda F_i^m, \tag{4.5}$$

where C_i^m denotes the traffic class of Message m, F_i^m is the redundancy level of Message m estimated by Node i, and $0 \leq \lambda \leq 1$ is a constant to balance the weight of traffic class and redundancy for queue prioritization. λ is a tunable parameter that can be determined according to specific application needs. I simply set it to $1/2$ in the implementation.

As discussed earlier, while the traffic class (i.e., C_i^m in Eq. (4.5)) is known, the redundancy of a message (i.e., F_i^m) needs to be dynamically estimated. In a typical store-and-forward network, messages are deleted from a node's buffer after they are transmitted to the next hop successfully. In opportunistic networks, however, multiple copies of the data message are often created and stored by different nodes in the network, in order to maintain necessary redundancy for achieving the desired QoS. In general, the higher the redundancy, the higher the message delivery probability when network capacity is not a concern. In this dissertation, I define the redundancy of a message to be the estimated probability that at least one copy of Message m is delivered to its destination within t time units, where t is updated during transmission to reflect the up-to-date remaining delay budget.

The redundancy of a message (F_i^m) is initialized to zero and updated during its transmissions. I consider a general scenario where Node i has an opportunity to communicate with Node j. It fetches Message m that is destined to Node k. First, the delay budget of Message m is updated as $t = t - \tau$ where τ is the time for which the message has stayed in the queue. Node i simply transmits the message to Node j and removes it from its queue if $j = k$. Otherwise, if $F_i^m \geq \beta$ where β is a predefined desired delivery probability, no action will be taken, because there is already sufficient redundancy. Node i will hold the message until it encounters the destination directly or the delay budget expires. If the delay budget expires, Node i simply discards the message from its queue. Generally, the more the redundancy, the higher the probability. $F_i^m \geq \beta$ means current redundancy (e.g., number of copies of Message m) is large enough to ensure a delivery probability no less than β. Note that, even with $F_i^m \geq \beta$, there is no guarantee that Message m will be delivered to the destination within the delay budget. However, if I look at a large number of such data messages, the protocol delivers them with an overall probability of β, thus achieving the goal. If $F_i^m < \beta$ and $p_i^k(t) < p_j^k(t)$, the message is transmitted to Node j. This transmission creates two copies of Message m, each sharing partial responsibility to deliver the data to its destination. Appropriate redundancy needs to be assigned to them, i.e.,

$$F_j^m \leftarrow 1 - (1 - [F_i^m])(1 - p_i^k(t)), \tag{4.6}$$

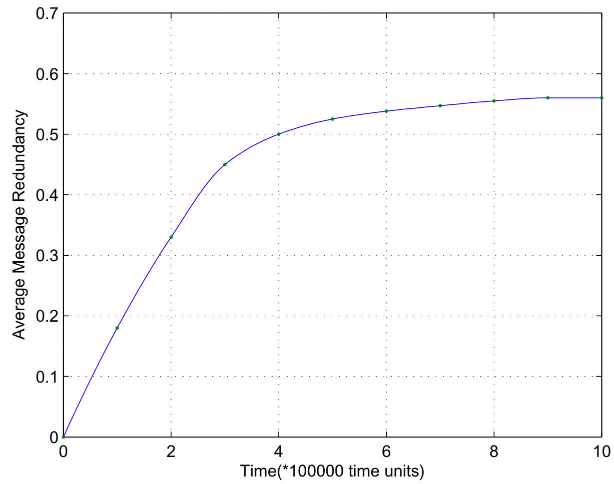


Figure 4.2: Evolvement of redundancy over time.

$$F_i^m \leftarrow 1 - (1 - [F_i^m])(1 - p_j^k(t)). \tag{4.7}$$

In both formulas, $(1 - [F_i^m])$ gives the probability that none of other nodes (except Nodes i and j) can deliver the message, where $[F_i^m]$ is the value before it is updated due to this transmission. Therefore the updated F_j^m (or F_i^m) indicates the probability that at least one copy of Message m can be delivered by other nodes except Node j (or Node i). In general, the more times a data message is forwarded, the more redundancy is created. For example, Fig. 4.2 shows that average message redundancy grows with the message's life span (where the simulation details are to be discussed in Sec. 4.4). At the same time, a node often holds a set of data messages in its queue. Fig. 4.3 illustrates that their redundancies largely fall into a normal distribution.

The above scheme is efficient in redundancy control and queue management, limiting redundancy to be just enough to achieve the desired delivery probability, β. How to choose the optimal β still remains an open issue. It is affected by such parameters as nodal contact probabilities, maximum queuing capacity, and traffic load and patterns. In a specific scenario, I can run simulations to identify the approximate optimal β.

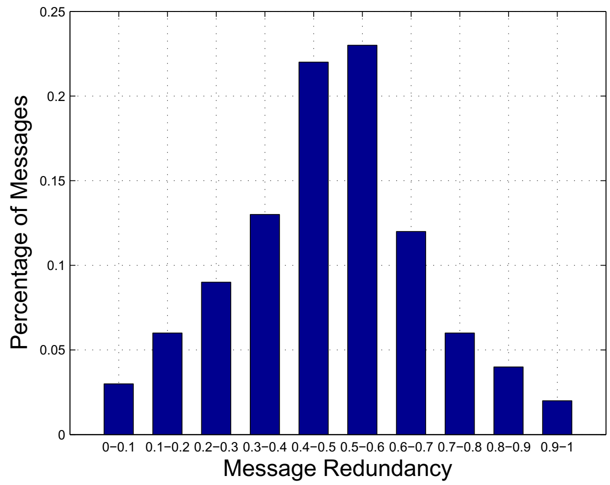


Figure 4.3: Distribution of message redundancy.

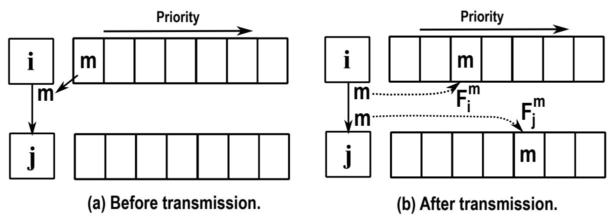


Figure 4.4: An example of transmission between Nodes i and j.

4.2.3 QoS-Aware Data Delivery

Since communication opportunity is low, transmission is often between two nodes only. If more than two nodes are within communication range, I assume an underlying medium access control protocol (e.g., IEEE 802.11) that randomly selects one node as the sender and another as the receiver. Therefore I focus on the scenario where Node i transmits a data message to Node j in the following discussions. Node i first learns the *QDPs* of Node j (i.e., P_j) via a two-way handshaking. Then it fetches the first message in its queue, denoted by Message m to Destination k and with a remaining delay budget of t. If $t = 0$, Node i simply drops the message. Otherwise, if Node j is the destination (i.e., $j = k$), Node i transmits the message to Node j and removes it from its queue. If Node j is not the destination, Node i transmits Message m to Node j if and only if $F_i^m < \beta$ and $p_i^k(t) < p_j^k(t)$.

Upon a message transmission, two copies of the message are created, with their redundancies calculated according to Eqs. (4.6) and (4.7) and their priorities updated by Eq. (4.5), respectively. Then both nodes insert their copies into their data queues according to the updated priorities (see Fig. 4.4). If a queue is full, the message at the end of the queue (i.e., the one with the lowest priority) is dropped.

The above process repeats with a randomly chosen node as the sender, until the communication link is broken (e.g., due to nodal mobility) or no messages are available for transmission.

4.2.4 Complexity Analysis

In general, the computational complexity at individual nodes is linear to the network size. More specifically, the communication between two nodes has a computational complexity of $O(k)$, where k is the buffer size of the message queue of each node. Update of *QDPs* for each node at the end of each time window has a complexity of $O(nl)$, where n is the total number of nodes in the network and l is the number of delay budget levels. Therefore, the overall time complexity of the proposed QoS-aware delivery algorithm is $O(k+nl)$. In addition, the overall space complexity of the proposed QoS-aware delivery algorithm is $O(nl)$. Note that k and l are often constant values. Therefore the time complexity is essentially $O(n)$, and the space complexity is $O(n)$ as well.

4.3 Prototype and Experiments

To demonstrate the feasibility of the proposed QoS-aware data delivery scheme and gain useful empirical insights, I have carried out two sets of experiments using the off-the-shelf Crossbow Micaz motes and Dell Streak Android tablets, respectively. The first experiment involves multiple clusters of static sensors that are connected by a small set of mobile nodes

Table 4.1: Experiment Parameters.

	Experiment I	Experiment II
Number of nodes	39	23
Buffer size	50	1000
Duration	3.5 mins	2 weeks
Message size	10 KB	1 KB-1 MB
Message generation rate	1 per 5 seconds	20 per day
Delay budget	3-90 seconds	4 hours-3.5 days

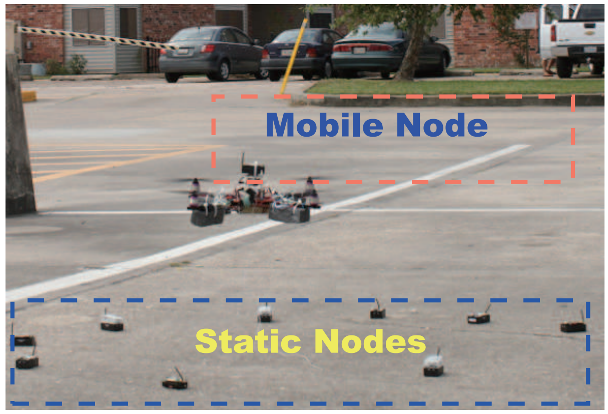
with regular movement patterns, while the second experiment is under the setting of a mobile social network where the nodes have diverse and uncontrolled mobility. The reason I carry out the two experiments is that they are representative examples in mobile opportunistic networks that need QoS support. For both of the experiments and the simulation presented in Sec. 4.4, the first fraction of system time is used to warm up for nodes to accumulate *QDPs*. Only source address, destination address and delay budget are generated during this period, data is generated and forwarded during the remaining part of time. Table 4.1 summarizes the experimental settings and configuration parameters. Sec. 4.3 is organized as follows. I present two experiments in Sec. 4.3.1 and 4.3.2, respectively. In each subsection, I first introduce testbed setup and configuration, and then discuss experimental results and related observations.

4.3.1 Experiment I: Opportunistic Sensor Network

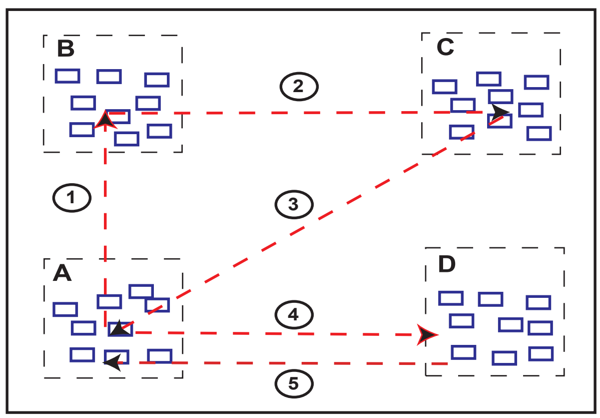
4.3.1.1 Testbed Setup

The sensor network testbed consists of 36 static Crossbow Micaz sensors and 3 mobile nodes. The static sensors are randomly deployed at four corners of a parking lot, forming four isolated clusters (see Clusters A, B, C and D in Fig. 4.5b). The sensors in a cluster are well-connected. I created such an experiment setting because it is common under practical applications. For example, when biologists study animals in a field, it is obviously too costly to cover the entire field with sensors. However, they are often interested in several target spots, and thus can deploy a cluster of sensors at each spot. The sensors in a cluster are closely located, within radio communication range. Therefore they are well-connected. At the same time, the distance between any two clusters is farther than the maximum radio transmission range of Micaz, and hence the clusters are isolated, calling for mobile nodes to carry data between them.

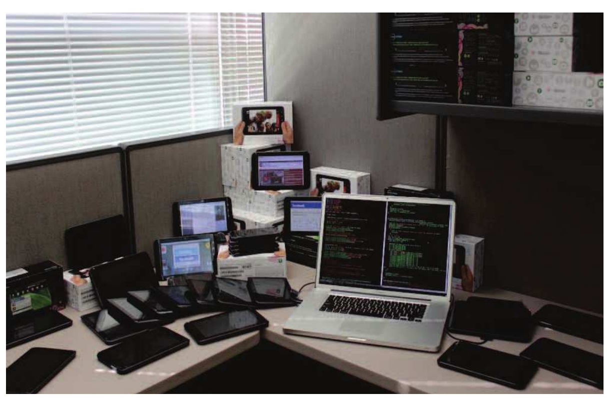
The mobile sensors act as Datamule [73] for data transmission among the sensor clusters. They are carried by a quadrotor and two students, with high and low mobility, respectively. The quadrotor is built upon the Mikrokopter platform. It can fly up to several hundred meters high and at a speed between 0 to 40 kilometers per hour, and thus is well suitable for remote sensor fields. Fig. 4.5a illustrates the ground Micaz motes and the mobile node on a quadrotor. In the experiment, the quadrotor is controlled by an independent remote controller and commutes among the four sensor clusters. It flies according to a pre-determined route as shown in Fig. 4.5b. More specifically, it begins its journey from Cluster A, then sequentially visits Clusters B, C, A, and D, and finally returns to Cluster A. It repeats the above routine flight during the experiment. The average flying height is about 2.2 meters based on the barometer readings on the flying platform. I say the quadrotor visits a cluster if its onboard sensor can communicate with any sensor in the cluster. The average time between it visiting two adjacent clusters is 12 seconds. The average waiting times (i.e., the average interval for a cluster to meet the quadrotor) at Clusters A, B, C, and D are 10, 20, 10, and 15 seconds, respectively. The other two mobile nodes are carried by two students. One student moves in clockwise direction and the other counter-clockwise. It takes about 3.5 minutes to complete a



(a) A cluster of ground sensors in the experimental field and the flying quadrotor that carries a sensor.



(b) Topology of the testbed and the moving pattern of the quadrotor.



(c) The Android tablets used in the mobile social network experiment.

Figure 4.5: Testbed setup. (a) and (b) illustrate Experiment I, while (c) shows the Android tablets used in Experiment II.

round of visits to the four clusters. Note that, although the routes are pre-determined, the communication is opportunistic due to dynamic mobility (e.g., unavoidable dynamics in moving speed and height) and varying channel conditions.

Each ground sensor generates a data message of 10 KB every 5 seconds to a randomly selected destination. A message is associated with a delay budget between 3 and 90 seconds.

4.3.1.2 Experimental Results

For performance comparison, I have implemented four schemes, dubbed QoS-aware, Best Effort, DDG, and Incentive, respectively. QoS-aware is the proposed QoS-aware delivery scheme; Best Effort is a delivery protocol without QoS support [86], which makes the decision on when and where to transmit data messages only according to the delivery probability. It is not a surprise to find the Best Effort approach results in low delivery rate because it does not differentiate traffic at all. DDG is the Delay-Differentiated Gossiping approach [68], which considers multiple traffic classes, and dynamically assigns the packets in each class a transmission probability and a Time-To-Live which together govern the total overhead for data transmission. Although DDG supports QoS provisioning, its data transmission is randomized. Therefore a packet is often delivered via a long path and consequently subject to high dropping probability due to the expiration of its delay budget. Incentive is the pair-wise tit-for-tat approach [63], which adopts an incentive-aware routing protocol that allows selfish nodes to maximize their own performance while conforming to TFT constraints. The Incentive scheme exhibits unsatisfied delivery rate because of its “candidate path generation” process, which results in high overhead under the experiment setting. At the same time, it does not consider different delay budgets for different messages when formulating the linear programming model. For fair comparison, trace data are collected to run comparable schemes.

Table 4.2: Results under Experiment I (Opportunistic Sensor Network)

	QoS-aware	Best Effort	DDG	Incentive
Avg. Del. Rate (overall)	72%	38%	42%	58%
Avg. Del. Rate (3 Sec)	2%	2%	2%	2%
Avg. Del. Rate (5 Sec)	8%	3%	5%	6%
Avg. Del. Rate (10 Sec)	32%	5%	16%	26%
Avg. Del. Rate (60 Sec)	98%	96%	97%	97%
Avg. Del. Rate (90 Sec)	100%	100%	100%	100%

I am primarily interested in data delivery rate. A message is delivered if it reaches its destination within its delay budget. The data delivery rate is defined as the ratio of the total number of delivered messages to the number of generated messages. Table 4.2 shows the overall average delivery rate and the delivery rates for messages with delay budgets of 3, 5, 10, 60, and 90 seconds, respectively. As can be seen, the proposed QoS-aware scheme achieves an overall delivery rate of 72%, significantly higher than other approaches. It is not a surprise to find the Best Effort approach results in low delivery rate because it does not differentiate traffic at all. DDG considers multiple traffic classes, and dynamically assigns the messages in each class a transmission probability and a Time-To-Live which together govern the total overhead for data transmission. However, as a gossiping approach, its data transmission is randomized. Therefore a message is often delivered via a long path and consequently subject to high dropping probability due to the expiration of its delay budget. The Incentive scheme exhibits unsatisfied delivery rate because of its "candidate path generation" process, which results in high overhead under the experiment setting. At the same time, it does not consider different delay budgets for different messages when formulating the linear programming model.

I also observe that the proposed QoS-aware data delivery scheme achieves the best performance compared with other schemes under moderate delay constraints. When the delay budget is extremely low, the QoS-aware scheme does not improve the performance because there is simply no way to deliver the data message within its time budget. As the delay budget increases, the QoS-aware scheme shows superior performance since it prioritizes messages and allocates resource to deliver more urgent and possibly deliverable data messages. It is obvious that, if the delay budget is very high, most data messages can always be delivered, no matter which scheme is employed.

4.3.2 Experiment II: Mobile Social Network

4.3.2.1 Testbed Setup

The second experiment is carried out under a mobile social network setting that involves twenty three volunteers including faculty members, senior Ph.D. students (who do not have classes), and graduate students at M.S. level (who go to classrooms regularly). A mobile social network is often created for a local community where the participants have frequent interactions, e.g., people living in a neighborhood, students studying in a college, or tourists visiting an archaeological site. It exploits Bluetooth and WiFi connections to form a sparse ad hoc network to support social networking. This is in a sharp contrast to web-based online social networks that rely on the Internet infrastructure (including cellular systems) for communication.

Unlike the first experiment where the mobile nodes move under regular patterns, the volunteers in this experiment have arbitrary and diverse mobility. Every volunteer carries a Dell Streak 5 or Streak 7 tablet (See Fig. 4.5c for a photo of the tablets used in the experiment), which operates on Android 2.2. The mobile nodes are paired and ready to communicate with each other via Bluetooth. In order to save power, a Service is created which runs on background to adaptively adjust the scanning frequency of the Bluetooth interface. The default scanning interval is set to 10 minutes during night and 1 minute during daytime. A node creates 20 data files everyday with the file size varying from 1 KB to 1 MB and time budget ranging from 4 hours to 3.5 days. The destination for a data file is randomly selected. The experiment lasts two weeks.

4.3.2.2 Experimental Results

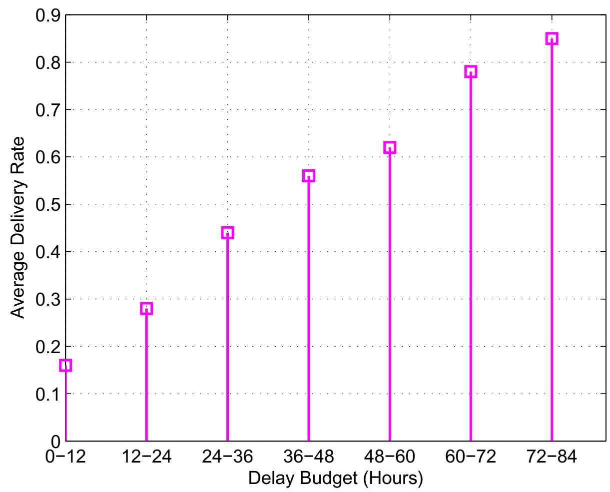
Similar to Experiment I, the proposed QoS-aware scheme outperforms other comparable schemes, i.e., Best Effort, DDG, and Incentive, in this mobile social network experiment. The results are omitted here for conciseness. Instead, I focus on investigating the impact of human activity on QoS-based data delivery. Fig. 4.6a illustrates the average delivery rate for data files with different delay budgets. Clearly, the larger the delay budget, the higher the delivery rate. When the delay budget reaches 3 days, an average delivery rate of more than 80% can be achieved.

Fig. 4.6b presents a detailed look of daily experimental data. More specifically, it illustrates the delivery rate of data files generated on different days during a week. As can be seen, data files generated during weekend always have lower delivery rate than those on weekdays. This is due to the low interactive activities of students and faculty on Saturday and Sunday. As a result, many data files cannot be delivered timely and eventually must be dropped due to their limited delay budgets. The data delivery rate on Friday is lower than other weekdays because no classes are scheduled on Friday afternoon and many offices are closed after 1:00 p.m..

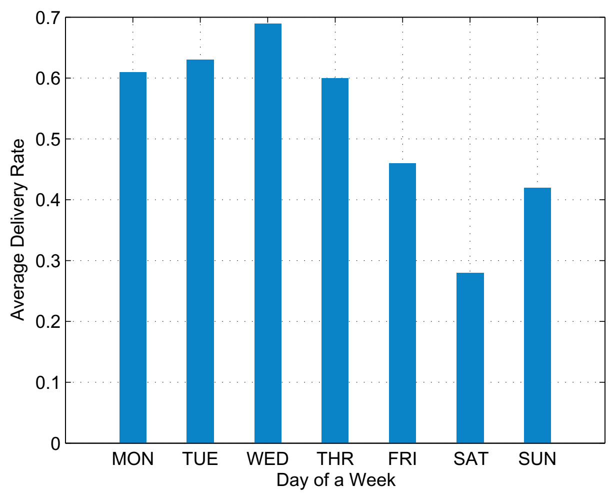
Fig. 4.6c further zooms in to show the delay of data files generated from the first to the 24th hour of a day. The second Tuesday of the experiment is chosen as an example, while similar results are observed on other days as well. The delivery rate is high during the day time and low at the night, which again shows the QoS-aware delivery scheme heavily depends on nodal mobility.

4.3.3 Further Discussion

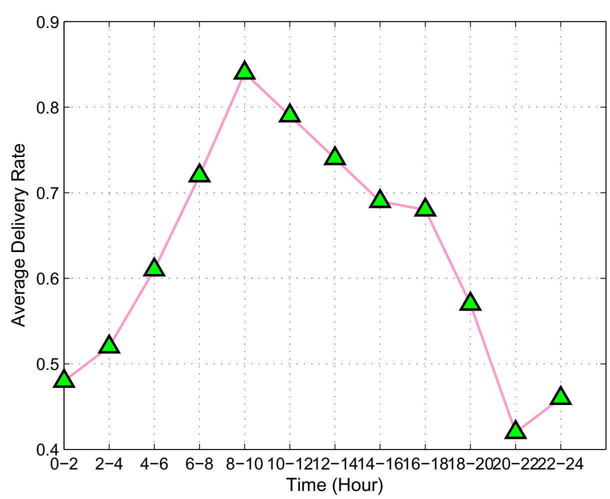
The buffer size indicates the maximum messages each of the nodes in both of the experiments can have in its message queue. The buffer size of Experiment I is much smaller than that of Experiment II, because Crossbow Micaz sensors are used in Experiment I which is a tiny wireless measurement system with a low-power microcontroller with very limited storage, while Experiment II use Dell Streak Android tablets with very large internal and external storage. The second experiment is carried out under a mobile social network setting that involves twenty three volunteers including faculty members, senior Ph.D. students (who do not have classes), and graduate students at M.S. level (who go to classrooms regularly). Based on the mobile node's social roles (e.g., professions), interests, and available resources and messages' utilities, each node creates messages with the size varying from 1 KB to 1 MB. For example, a professor may deliver a homework to students which needs larger message size, while a student may send a sport news to his friends which requires smaller message size. The messages are randomly generated, so there is no average message size. The second experiment is under a mobile social network setting, where 23 Dell Streak Android tablets are carried by volunteers with arbitrary and diverse mobility patterns during a period of two weeks. Unlike the first experiment where the mobile nodes move under regular patterns, the volunteers in this experiment have arbitrary and diverse mobility. Therefore, the messages generated in Experiment II will experience longer time than that in Experiment I to be delivered to the destinations. All the parameters I set for the two experiments are all based on



(a) Delivery rate under different delay budgets.



(b) Delivery rate distribution during a week.



(c) Hourly delivery rate on the second Tuesday.

Figure 4.6: Results under Experiment II (Mobile Social Network).

the charateristics of the two experiments.

Overall, the experimental results demonstrate that the nodal mobility, and accordingly the contact opportunity among nodes, is decisive for the overall network performance as already revealed in earlier works [2, 4, 9, 27, 86]. The proposed scheme efficiently allocates resources according to the delay budgets of data messages, and thus supports effective QoS provisioning, achieving significant higher delivery rate in comparison with other schemes.

4.4 Simulation Results

Besides the experiments presented above, I carry out two separate simulations. The first is based on DieselNet trace to study QoS provisioning in vehicular networks. The second simulation is performed under power-law mobility model for evaluating the scalability of the proposed scheme with the increase of network size, traffic load, and nodal mobility.

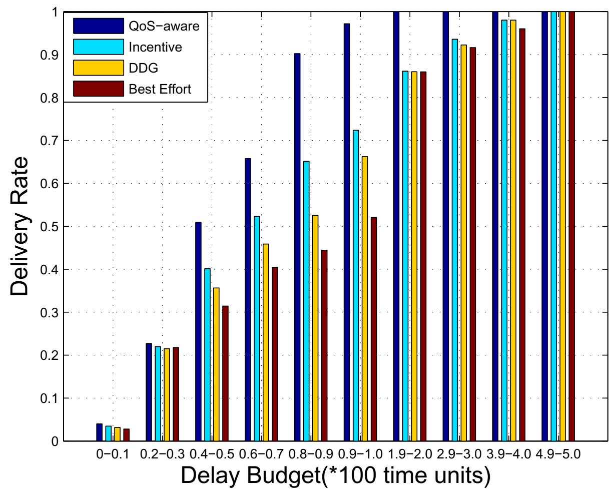
4.4.1 Simulation under DieselNet Trace

The DieselNet testbed comprises 33 buses, serving an area of approximately 150 square miles. Each bus carries a node with WiFi. The simulation is based on the trace data obtained in 2008 [8]. As shown in Fig. 4.7a, the proposed QoS-aware scheme achieves the highest data delivery rate under all delay budgets. For example, it delivers more than 90% of the messages with delay budget between 90 to 100 time units (minutes), in comparison with 52% in the Best Effort approach, 66% in the DDG approach, and 72% in the Incentive approach. At the same time, it well controls transmission overhead as illustrated in Fig. 4.7b. To deliver a data message, an amount of redundancy (i.e., a number of copies) of the message are generated, hoping that at least one of them can reach the destination. The overhead is the average number of such copies per message. The high efficiency of the QoS-aware scheme is attributed to the fact that the estimated *QDP* enables efficient use of communication resource (i.e., the capacity of nodes and their meeting opportunities) and that the adaptive prioritization scheme supports effective queue management and redundancy control. With the increase in delay budget, the delivery rate increases accordingly under all approaches, because data messages have more time and better chance to reach their destinations. In addition, I observe similar average delay under all schemes (see Fig. 4.7c). But note that the average delay is calculated for delivered messages only. Due to the low delivery rate in Best Effort, DDG, and Incentive, many messages that in fact experience long delay are not included in the calculation.

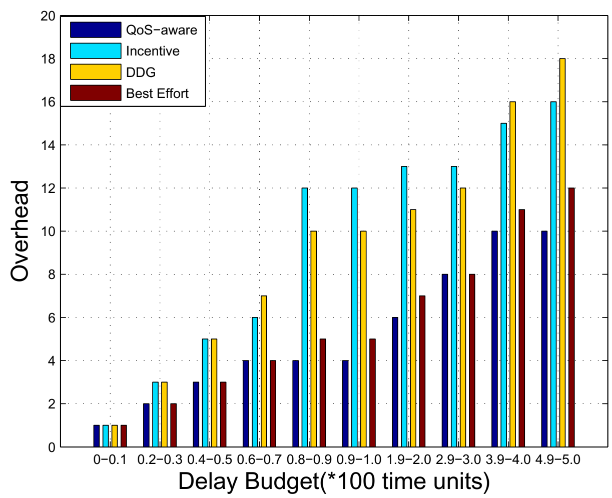
4.4.2 Simulation under Power-Law Mobility Model

The experiments and trace-based simulation have provided a comprehensive evaluation of the proposed scheme in several practical settings including sensor networks, vehicular networks, and mobile social networks. I next present simulation results based on power-law mobility model, which offer valuable performance trend by scaling several network parameters.

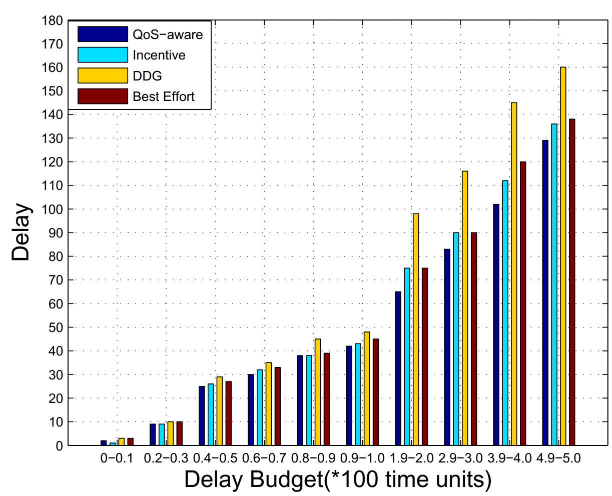
The simulated area is divided into a grid of 10×10 cells. Each node has a home cell where it initially locates, and moves according to power-law distribution, which is deemed a realistic model for human mobility. Two nodes communicate only if they are in the same cell. Let $P_i(x)$ denote the probability for Node i to be at Cell x. $P_i(x) = k_i(1/d_i(x))^{\delta}$ where k_i is a constant, δ is the exponent of the power-law distribution, and $d_i(x)$ denotes the distance between Cell x and Node i's home cell. Under this model, δ is the key parameter governing node behavior. When δ is large, nodes tend to move among a very small subset of cells. With the decrease of δ, the moving range becomes wider. By default, I set the maximum queue size to be 500. The message generation of each node follows a random process with an average interval of 30 time units out of 100 time units. The FTD threshold is set to be $\beta = 0.7$. The



(a) Delivery rate distribution.



(b) Overhead distribution.



(c) Delay distribution.

Figure 4.7: Performance comparison under DieselNet trace.

results are illustrated in Fig. 4.8 for the data messages with delay budget of 90-100 time units. The results under other delay budgets exhibit similar trend and thus are omitted here.

Fig. 4.8a shows the performance of different schemes by varying the number of nodes in the network. With the increase of network size, nodes have more opportunities to meet each other and to reach the destinations. Thus, the messages have better chances to be delivered within their delay budgets. This explains why the delivery rate of all schemes increases.

With the increase of message generation rate, the proposed QoS-aware delivery scheme exhibits a graceful degradation of data delivery rate (see Fig. 4.8b), because it differentiates traffic and makes efficient use of communication and storage resources to meet the QoS needs. For example, when more messages with low delay budgets are generated, the protocol postpones the transmission of some messages with long delay budgets, such that more messages are delivered within their delay budgets in total. On the other hand, Best Effort, DDG, and Incentive do not effectively support traffic differentiation, thus suffering dramatic decrease of delivery rate.

The power-law factor δ determines the mobility patterns of nodes. As illustrated in Fig. 4.8c, if δ is small, all nodes tend to have similar, wide mobility, and thus almost identical *QDPs*, which consequently result in ineffective data transmission and low delivery rate. With a large δ, on the other hand, a node stays close to its home cell, i.e., can hardly reach any remote cells. Lower mobility leads to lower network capacity, and thus lower delivery rate.

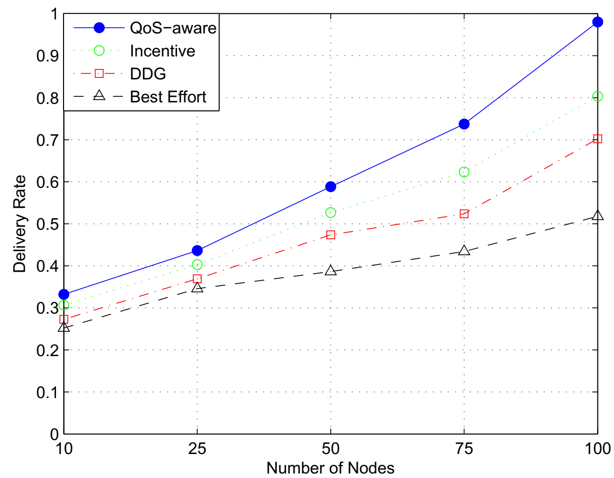
Fig. 4.8d shows the impact of queue size. With an increase in queue size, all schemes enjoy higher delivery rate because more messages can reside in the queue without being dropped.

I have also studied nodal scanning frequency. A node has a duty cycle. It wakes up to explore possible communication opportunities by scanning nearby nodes. The lower the scanning frequency, the less the meeting events, which leads to lower communication capacity. As a result, a lower delivery probability is observed in Fig. 4.8e. Clearly, a higher scanning frequency is at the cost of higher energy consumption.

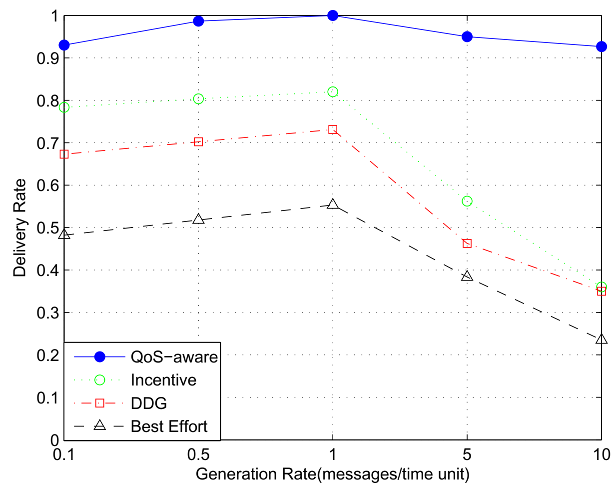
The FTD (Fault Tolerant Degree) threshold β is introduced for redundancy control. In Fig. 4.8f, I first observe a higher delivery rate with the increase of β, because a larger β permits more redundancy and accordingly increases data delivery probability. However, when β is greater than 0.7, the delivery rate starts to decrease. Due to the given constraints on communication bandwidth and nodal queue size, the excessive redundancy created under high β often does not contribute to improving delivery rate. Worse yet, it leads inefficient use of communication opportunities and storage space, resulting in degraded overall performance. It still remains an open problem to find optimal β. As a rule of thumb, the highest delivery rate is achieved when β is around 0.7-0.8 in the simulations and experiments.

4.5 Conclusion

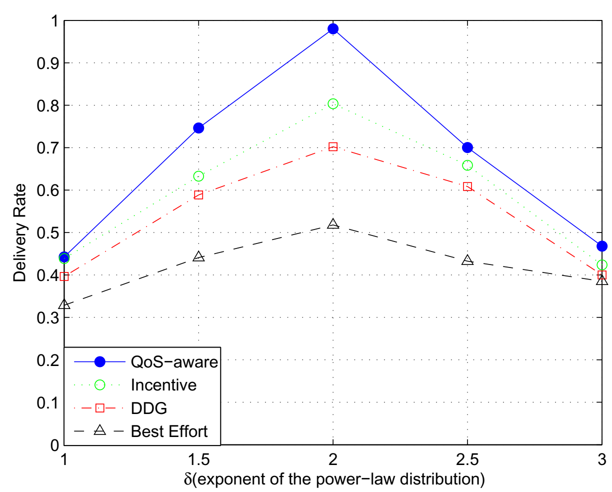
In this chapter, I have proposed a QoS-aware data delivery scheme for mobile opportunistic networks. It employs QoS-aware delivery probability (*QDP*) to reflect the capability of a node to deliver data to a destination within a given delay budget, and maintains a prioritized queue, where the priority is determined by a function of traffic class and dynamic redundancy in order to support efficient prioritization and redundancy control. Two experiments have been carried out to demonstrate and evaluate the proposed QoS-aware data delivery scheme. The first experiment involves multiple clusters of static Crossbow sensors that are connected by air and ground mobile nodes with controlled mobility. The second experiment is under a mobile social network setting during a period of two weeks, where the



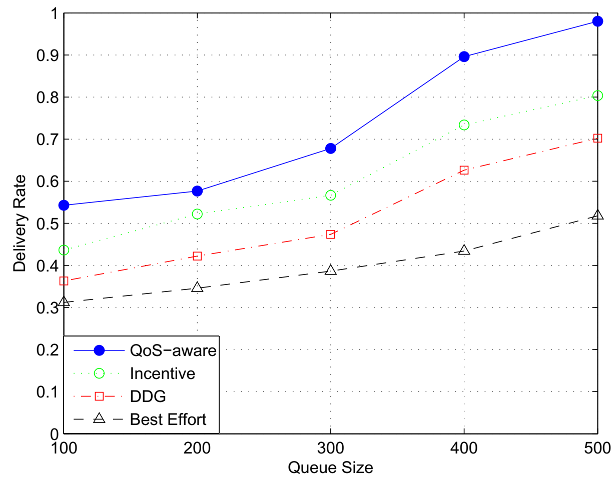
(a) Network size.



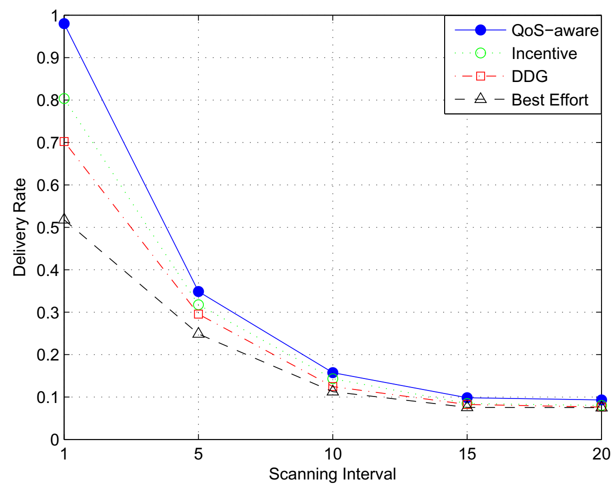
(b) Traffic load.



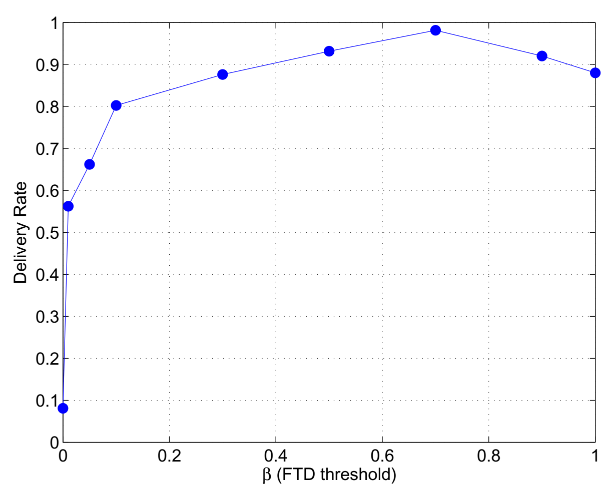
(c) Nodal mobility.



(d) Queue size.



(e) Scanning frequency.



(f) FTD (Fault Tolerant Degree) threshold.

Figure 4.8: Performance trend under power-law mobility model. The results are obtained for data messages with delay budget of 90-100 time units.

prototype is implemented on Dell Streak Android tablets carried by 23 volunteers with arbitrary and diverse mobility patterns. Moreover, simulation results have been obtained under DieselNet trace and power-law mobility model to study the scalability and performance trend. The experiments and simulations have shown that the proposed scheme achieves efficient resource allocation according to the desired delay budget, thus supporting effective QoS provisioning.

CHAPTER 5: Delay-Constraint Least-Cost Multicast in Mobile Opportunistic Networks

In this chapter, I study the problem of delay-constrained least-cost multicast in mobile opportunistic networks. I formally formulate the problem and show it is NP-complete. Given its NP-completeness, I explore efficient and scalable heuristic solutions. I first introduce a centralized heuristic algorithm which aims to discover a tree for multicasting, in order to meet the delay constraint and achieve low communication cost. While the centralized solution can be adapted to a distributed implementation, it is inefficient in a mobile opportunistic network, since it intends to apply a deterministic transmission strategy in a nondeterministic network by delivering all data packets via a predetermined tree. Based on such observation, I develop a distributed online algorithm that makes an efficient decision on every transmission opportunity. When a node meets another node, the former transmits the packet to the latter if the latter helps reduce the cost to deliver the packet to its destinations while reaching a desired delivery probability within a given delay budget. I prototype the proposed distributed online multicast algorithm using Nexus tablets and conduct an experiment that involves 37 volunteers and lasts for 21 days to demonstrate its effectiveness. I also carry out simulations to evaluate the scalability of the proposed schemes.

5.1 System Overview

Efficient multicasting is indispensable for supporting a variety of applications in mobile opportunistic networks. In a multicast event, a source node intends to deliver data to a set of destinations. Each node in a mobile opportunistic network can be a source node or a receiver or more commonly both. Multicasted data span a range of categories, including, among many others, advertisements, coupons, deals, newsletters, product catalogs, and event invitations.

While long data delivery delay is generally unavoidable given the intermittent connectivity in mobile opportunistic networks, the constraint on end-to-end delivery delay is highly desired in a variety of applications. For example, the dissemination of advertisements or coupons must meet a delay budget no longer than its expiration date [27, 67, 96]. Separately, in wildlife tracking applications, interactive control commands must be multicasted within a short end-to-end delay bound, as opposed to the routine transmissions of ambient environmental data that can tolerate long delays. Data delivered beyond their delay budgets often lead to reduced or completely forfeited value. However, due to nondeterministic connectivity, it is intrinsically infeasible to provide a hard guarantee of end-to-end delivery delay. Thus, a probability-based delay budget is adopted in this dissertation, which concerns the probability to deliver a data packet to its destinations within a predefined delay budget.

Besides the delay constraint, it is obviously desired to minimize the total cost for multicasting in mobile opportunistic networks. To this end, I study the delay-constrained least-cost multicast problem in this dissertation, aiming to minimize the overall communication cost and at the same time achieve a desired probability to deliver data to every destination within a predefined delay budget. The formal problem formulation will be given in Sec. 5.2.

5.2 Problem Formulation and 0-1 Integer Programming

In this section, I formally formulate the problem of delay-constrained least-cost multicast in mobile opportunistic networks. I show it is essentially a NP-complete 0-1 integer program, and present numeric results and discuss useful insights for developing heuristic and distributed solutions.

Assume there are n nodes in the network and they form k opportunistic links. The delay of each link is a random variable denoted by T_l, $\forall 1 \leq l \leq k$. To formulate the delay-aware multicast problem, I define a $1 \times k$ binary transmission vector, X, for a data delivered from a source s to a given set of destinations Φ. Each element of the vector is a 0-1 variable to be optimized. If $X_l = 1$, the link l is employed for data dissemination; otherwise, the communication opportunity will not be utilized. A transmission strategy, i.e., X, induces a total communication cost (defined as C_X) and a random variable that represents the delay to deliver the data to each destination (denoted as $\tau_X^d, \forall d \in \Phi$). For analytic tractability, I assume communication delay is dominated by nodal meeting intervals. In addition, I assume a node receives and forwards the same data packet only once.[1]

Therefore the optimization problem is formulated as follows:

$$\begin{aligned} Minimize: &\quad C_X, \\ S.t.: &\quad Pr\{\tau_X^d \leq \delta\} \geq \gamma, \forall d \in \Phi, \end{aligned} \tag{5.1}$$

aiming to minimize the overall communication cost and at the same time reach a desired probability γ to deliver data to every destination within a delay budget δ. Note that, due to nondeterministic connectivity, it is intrinsically impossible to provide a hard guarantee of end-to-end delivery delay. Thus, a probability-based delay budget is adopted in this dissertation to achieve a desired probability of delivering data within a predefined delay budget.

While the problem formulated above appears simple, it is nontrivial to be solved, since the nondeterministic network setting dramatically increases the complexity to derive C_X and $Pr\{\tau_X^d \leq \delta\}$. To derive the delay constraint in Eq. (5.1), I define a $k \times 1$ vector Y^d for each destination d, where an element of the vector is a 0-1 variable. I also define two matrices, A

[1] If other delay factors (e.g., communication delay) become significant or redundant copies of data are introduced in transmissions, the problem formulation and the 0-1 integer programming still hold, but with difference in the calculation of delay, i.e., τ_X^d.

and B. A is a $n \times k$ network topology matrix, where $A_{ij} = 1$ if Node i is on Link j. B is a $n \times |\Phi|$ source-destination matrix, where $B_{ij} = 1$ if Node j is the source node s, $B_{ij} = -1$ if Node j is the i-th destination, and $B_{ij} = 0$ otherwise. Then I introduce two constraints:

$$AY^d = B^d, \tag{5.2}$$

and

$$Pr\{\sum_{l=1}^{k} T_l Y_l^d \leq \delta\} \geq \gamma, \forall d \in \Phi. \tag{5.3}$$

The former ensures the links with $Y_l^d = 1$ form a valid end-to-end path from s to d. The latter enforces a desired probability of delivering data within a given delay budget. While it is difficult to derive a close-form solution for $Pr\{\sum_{l=1}^{k} T_l Y_l^d \leq \delta\}$ under an arbitrary distribution of $T_l, \forall 1 \leq l \leq k$, it can be numerically calculated. More specifically, given the known delay distribution of $T_l, \forall 1 \leq l \leq k$, the distribution of $\sum_{l=1}^{k} T_l Y_l^d$ can be derived by convolution. Of course, under special distributions, the calculation can be dramatically simplified. Finally, I create a set of constraints,

$$Y_l^d \leq X_l, \forall 1 \leq l \leq k, \tag{5.4}$$

such that X captures all links used for data transmissions. The communication cost is often proportional to the number of transmissions. Thus I simply let

$$C_X = \sum_{l=1}^{k} X_l. \tag{5.5}$$

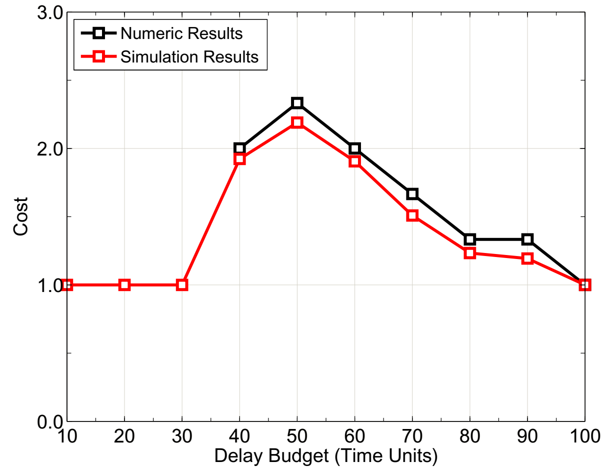
With such manipulation by plugging Eqs. (5.2)-(5.5) into Eq. (5.1), I have arrived at a 0-1 integer program. It is known NP-complete, but existing tools (such as Matlab [30]) can be employed to determine X when the network is small.

I have carried out simulations to validate the above optimization model. The network is deployed in an area of 10×10, with 6 nodes following random walk mobility. The T_l of each opportunistic link is obtained via online learning, and used in optimization calculation for determining the transmission strategy X. The desired delivery probability is 0.6. Based on the optimal X, I run a simulation and compare the simulation results with the numeric results obtained from Eqs. (5.2)-(5.5). Fig. 5.1 depicts the results. In general, I observe a good match in both cost and delivery ratio between the simulation and numeric calculation. Under very small δ, no paths can be used to deliver packets to destinations within the desired delay budget by following the optimization model. Therefore, no cost and delivery ratio are obtained by the numeric calculation. In the simulation, packets are not transmitted according to any strategy but only delivered when the source meets the destination directly, thus the cost is 1. With a longer delay budget, cost increases too because more transmissions with longer delay are aggressively attempted. At the same time, more packets can reach the corresponding destinations and thus the delivery ratio naturally increases. However, when δ is sufficiently large, many options of routing paths become available (that all satisfy the delay budget), allowing the optimization model to choose the one with the lowest cost (i.e., the one that involves the least transmissions). This explains why cost decreases under large δ.

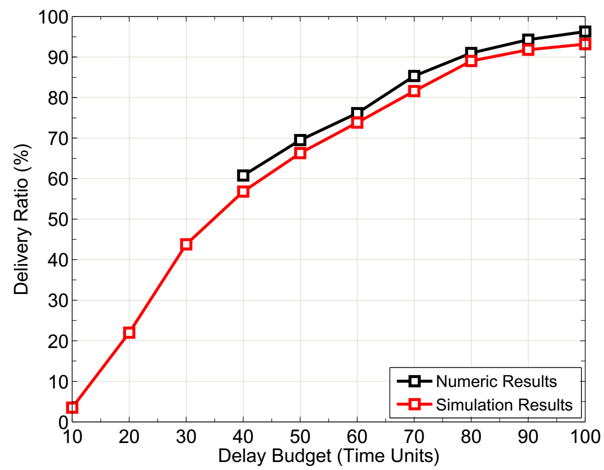
5.3 Centralized Heuristic Algorithm

While the above 0-1 integer programming model can yield optimal results, it is computationally expensive and thus unpractical for real-world implementation. Given the NP-completeness of the problem, I explore efficient and scalable heuristic solutions.

I first introduce a centralized heuristic algorithm which aims to discover a tree for multicasting (denoted by $\mathbb{T}$), in order to meet the constraint in Eq. (5.1) and achieve low communication cost. $\mathbb{T}$ can be considered as an approximation of the optimal X yielded from



(a) Cost.



(b) Delivery ratio.

Figure 5.1: Comparison of the simulation and numeric results based on the 0-1 integer programming model.

the 0-1 integer programming.

Initially, the tree $\mathbb{T}$ includes the source node only and all destinations are put into the set Φ. The algorithm runs in iterations. Each iteration includes the following steps.

- First, it computes a path from every destination in Φ to the current tree, which satisfies the constraint in Eq. (5.1) and at the same time introduces the least additional cost (e.g., the fewest links in addition to the current tree). How to efficiently determine such a path is to be discussed below.

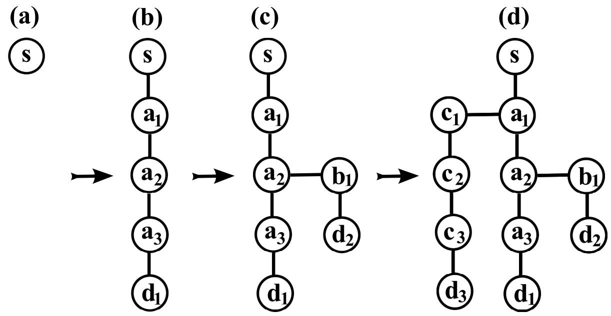


Figure 5.2: An example of the centralized heuristic algorithm.

- Second, the above step essentially creates $|\Phi|$ hypothetical new trees, each augmenting the current tree by a path. A metric, named radiation, is computed to describe the goodness of each hypothetical tree. The destination that results in the smallest radiation is chosen. It is removed from Φ, and the corresponding hypothetical tree replaces the current tree.

- The above steps repeat until Φ is empty.

Fig. 5.2 shows an example of augmenting the tree under the algorithm, until it covers all destinations. The algorithmic details are elaborated below.

5.3.1 Delay-Constrained Least-Cost Single Path Construction

The multicast tree $\mathbb{T}$ is initialized to include the source node s only and then augmented to cover the destinations. In each iteration of the algorithm, I first discover a delay-constrained least-cost (DCLC) path from each node in Φ (i.e., the set of remaining destinations) to the current tree $\mathbb{T}$. The DCLC path for a destination node d is the path with the least additional cost while meeting the delay constraint. Note that, here I am concerned about the additional cost to reach the destination d. The path from s to d may utilize some existing links in the current multicast tree $\mathbb{T}$. Such links do not contribute to the additional cost. By minimizing the additional cost, I essentially encourage the reuse of existing links. The additional cost to reach destination d is denoted by $C_{\mathbb{T}}^{d}$. Similar to the previous discussion, the delay constraint is represented by $Pr\{\tau_{\mathbb{T}}^{d} \leq \delta\} \geq \gamma$, where $\tau_{\mathbb{T}}^{d}$ is the delay from the source node s to the destination node d via the chosen path. Again, since the distribution of any individual opportunistic link delay (i.e., T_l) is known, the distribution of a path delay can be derived by convolution. Thus once the path to d is determined, $Pr\{\tau_{\mathbb{T}}^{d} \leq \delta\}$ can be calculated accordingly.

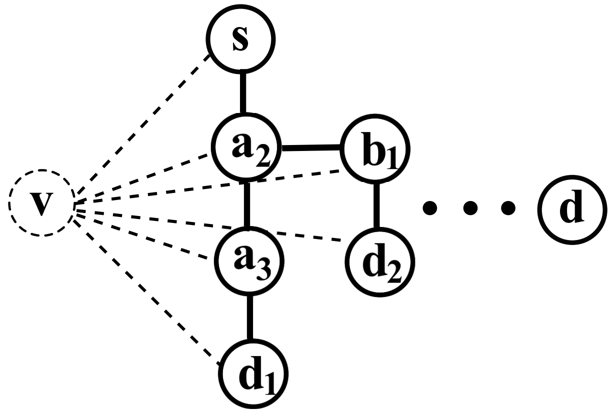
To solve the above DCLC problem between a node and a tree, I convert it to a standard DCLC problem between two nodes. More specifically, I create a virtual node v and connect it to every node in $\mathbb{T}$ via a virtual edge (as shown in Fig. 5.3a). Each virtual edge has a cost of zero. Thus the DCLC path between v and a node d in Φ is equivalent to the DCLC path from d to the multicast tree $\mathbb{T}$.

The DCLC problem between two nodes is NP-hard in conventional networks with stable links (unless all link costs are equal or all link delays are equal) [69]. It obviously remains NP-hard in opportunistic networks. Thus I adopt a heuristic approach based on the idea introduced in [69]. More specifically, from the current node x (which is initialized as v), I recursively determine the next hop node, in order to achieve low cost while meeting the delay constraint.

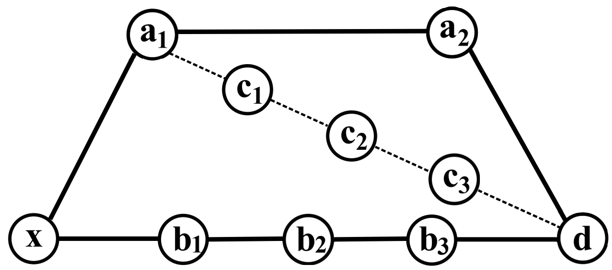
More specifically, I identify two paths between x and d by using Dijkstra's algorithm: a path with the least cost (see $x, a_1, a_2, ..., d$ in Fig. 5.3b) and a path with the highest delivery probability within delay budget δ, i.e., the highest $Pr\{\tau_{\mathbb{T}}^{d} \leq \delta\}$ (see $x, b_1, b_2, ..., d$ in Fig. 5.3b).

Then, I discover the path from a_1 to d such that the path from x through a_1 to d has the highest delivery probability within delay budget δ (see $x, a_1, c_1, c_2, ..., d$ in Fig. 5.3b). If such probability is greater than γ, I choose a_1 as the next hop node; otherwise, b_1 is chosen as the next hop node.

Whenever the next hop is determined, I update x by the next hop node and repeat the



(a) To discover the delay-constrained least-cost (DCLC) path.



(b) An illustration of the DCLC algorithm.

Figure 5.3: DCLC problem.

above steps, until $x = d$, i.e., the destination is reached.

The above heuristic algorithm obviously results in a path that satisfies the delay constraint, i.e., $Pr\{\tau_{\mathbb{T}}^{d} \leq \delta\} \geq \gamma$. It intends to reduce the path cost by always using the least cost path whenever it is possible. Of course, it does not guarantee the final cost ($C_{\mathbb{T}}^{d}$) is minimized.

5.3.2 Selection of Best Hypothetical Tree Based on Radiation

The above step establishes a DCLC path from each node in Φ to the current tree $\mathbb{T}$. If a DCLC path does not exist for a destination, it is marked unreachable. It essentially creates up to $|\Phi|$ hypothetical new trees, denoted by $\{\mathbb{T}_d | \forall d \in \Phi\}$. Next I introduce a metric, named *radiation*, to choose the best hypothetical tree.

Each node d in Φ induces a hypothetical tree. Its *radiation* is defined as

$$R_d = \frac{1}{|\Phi|-1} \sum_{i \in \Phi, i \neq d} C^i_{\mathbb{T}_d}, \tag{5.6}$$

which is intrinsically the average least cost from the remaining destinations to the hypothetical tree $\mathbb{T}_d$.

The hypothetical tree with the lowest radiation is selected, since it minimizes the average cost for future destinations to join the tree. Accordingly, $\mathbb{T}$ is replaced by $\mathbb{T}_d$ with minimum R_d, and the corresponding d is removed from Φ.

The algorithm repeats the above process until all destinations have been added into the tree, i.e., $\Phi = \emptyset$.

5.4 Distributed Online Algorithm

The above centralized algorithm can be adapted to a distributed implementation. For example, each node can run it in a distributed manner according to its best-known knowledge of the network. However, such algorithm, no matter in a centralized or distributed implementation, is essentially an offline solution. It intends to discover an optimal routing tree based on the network graph and transmits data according to the tree. This approach is well accepted in conventional, deterministic networks. However, *it is inefficient in a mobile opportunistic network, since it intends to apply a deterministic transmission strategy in a nondeterministic network* by transmitting all data packets via a predetermined tree. In mobile opportunistic networks, the optimal routing tree is the "best" only on a statistic basis when I consider a large number of data packets. It is not necessarily the best solution for every individual transmission.

For example, assume that under the optimal routing tree, Node A should transmit data packets to Node B, which is the statistically optimal strategy. But when Node A intends to transmit a particular packet, it might not be able to establish a link with Node B within the delay budget. Therefore, the transmission would fail if it is determined to wait for Node B. Instead, it is obviously favorable to deliver the packet via other nodes it meets opportunistically. In general, Node A may meet a sequence of nodes, similar to a stochastic process. It must make an adaptive, online decision on which communication opportunity should be exploited to deliver the packet, in order to achieve the optimization goal given in Eq. (5.1).

Based on the above observation, I propose a distributed algorithm, where the online routing decision is made when two nodes meet. A node often has a number of packets in its data queues. It may transmit a packet directly to the destination or to an intermediate node which subsequently continues to forward the packet directly or indirectly to the destination. When a node meets another node, the former needs to decide whether to transmit a packet to the latter. Such a routing decision must be made based on a delay/cost-aware multicast routing metric, which indicates if the latter helps reduce the cost to deliver the packet to its destinations while reaching a desired delivery probability within a given delay budget.

The proposed algorithm consists of two components, which respectively establish an approximate multicast tree for each node and make appropriate online routing decisions, as outlined below.

5.4.1 Approximate Multicast Tree

It is straightforward to establish an approximate multicast tree. Briefly, each node discovers a set of opportunistic links with its direct neighbors and maintains the corresponding delay distributions. In the implementation, I adopt discrete time slots for constructing approximate delay distributions, where a slot is Δ minutes. The delay distribution of a direct link between Nodes i and j can be represented by a vector $[P_{ij}^1, P_{ij}^2, ..., P_{ij}^K]$, where P_{ij}^k is the probability that their inter-meeting time is greater than $(k-1)\Delta$ and less than $k\Delta$. Such an approximate delay distribution can be built via a trivial online learning algorithm according to historical inter-meeting times. The nodes exchange such information when they meet, to learn the remote opportunistic links up to a certain number of hops. Each node can thus employ the heuristic algorithm introduced in Sec. 5.3 to compute a transmission strategy, i.e., a multicast tree, in a distributed manner according to its best-known knowledge of the network.

Note that, for the sake of low communication overhead and computation complexity, the approach is based on partial network information, and thus does not guarantee a complete multicast tree. As a matter of fact, it often results in a tree that covers partial destinations only.

5.4.2 Online Dynamic Routing

To facilitate the discussion, I assume that each multicast data packet is associated with a descriptive metadata, which includes a source (i.e. s), a set of multicast destinations (i.e. Φ), and a sequence number (i.e. m).

After the packet is created by the source, it will be transmitted to a set of intermediate nodes based on the routing scheme to be introduced below. Each node carries a responsibility to deliver the packet to a subset of destinations. For example, let's assume Node i currently holds a multicast packet. It is responsible to deliver the packet to a set of destinations, $\Phi_i \subseteq \Phi$. Initially, $\Phi_i = \Phi$ if Node i is the source, and $\Phi_i = \emptyset$ for all other nodes. Let $\mathbb{T}_i(\Phi_i)$ denote the multicast tree at Node i that intends to cover the destinations in Φ_i. It is built according to the algorithm discussed in Sec. 5.3. The cost of the tree is denoted by $C_{\mathbb{T}_i(\Phi_i)}$. If $\mathbb{T}_i(\Phi_i)$ covers all destinations in Φ_i (i.e., satisfies the constraint that $Pr\{\tau_{\mathbb{T}}^d \leq \delta\} \geq \gamma, \forall d \in \Phi_i$), $C_{\mathbb{T}_i(\Phi_i)}$ is simply the sum of all link costs. Otherwise, the cost is set to be infinity. In other words, I have

$$C_{\mathbb{T}_i(\Phi_i)} = \begin{cases} \sum_{l \in \mathbb{T}_i(\Phi_i)} C_l, & \Phi_i \subseteq \mathbb{T}_i(\Phi_i), \\ \infty, & \Phi_i \not\subseteq \mathbb{T}_i(\Phi_i), \end{cases} \tag{5.7}$$

where C_l is the cost of an opportunistic link $l \in T_i(\Phi_i)$, which is set to 1 for simplicity in the implementation.

Since communication opportunity is low, transmission is often between two nodes only. If more than two nodes are within communication range, I assume an underlying medium access control protocol (e.g., IEEE 802.11) that randomly selects one node as the sender and another as the receiver. Therefore I focus on the scenario where Node i transmits a data packet to Node j in the following discussion.

When Node i meets Node j, the former instructs the latter to compute a multicast tree, aiming to cover the destination set Φ_i. Node j may or may not be able to cover the entire Φ_i. Let $\mathbb{T}_j(\Phi_j)$ denote the tree constructed by Node j, where $\Phi_j \subseteq \Phi_i$.

Node i transmits the packet to Node j, if and only if the following condition is satisfied:

$$C_{\mathbb{T}_i(\Phi_i - \Phi_j)} + C_{\mathbb{T}_j(\Phi_j)} + 1 \leq C_{\mathbb{T}_i(\Phi_i)}. \tag{5.8}$$

The above condition indicates that the cost can be reduced by splitting the delivery responsibilities between Nodes i and j.

If Node i does transmit the packet to Node j, it updates its destination set to be

$$\Phi_i \triangleq \Phi_i - \Phi_j. \tag{5.9}$$

Node i stops transmitting the multicast packet when either $\Phi_i = \emptyset$ or the delay budget expires.

A key advantage of the online algorithm is to exploit nondeterministic communication opportunities. This is in a sharp contrast to the centralized algorithm that relies on a precomputed static routing tree.

5.5 Prototype and Experiment

To demonstrate the feasibility and efficiency of the proposed algorithms and to gain useful empirical insights, I have carried out a testbed experiment using off-the-shelf Nexus tablets. In this section, I first introduce the testbed setup and then present experimental results.

5.5.1 Prototype and Testbed Setup

The experiment is carried out under a mobile social network setting that involves 37 volunteers including faculty members, senior Ph.D. students (who do not have classes), and graduate students at M.S. level (who go to classrooms regularly). A mobile social network is often created for a local community where the participants have frequent interactions, e.g., people living in a neighborhood, students studying in a college, or tourists visiting an archaeological site. It exploits Bluetooth and WiFi connections to form a sparse ad hoc network to support social networking. This is in a sharp contrast to web-based online social networks that rely on Internet infrastructure (including cellular systems) for communication.

The volunteers in this experiment have arbitrary and diverse mobility. Every volunteer carries a Nexus 7 tablet powered by Android 4.4, KitKat. The mobile nodes are paired and ready to communicate with each other via direct WiFi. In order to save power, a Service is created which runs on background to adaptively adjust the scanning frequency. The default scanning interval is set to 10 minutes during night and 1 minute during daytime. A node generates 24 multicast events everyday with delay budgets randomly distributed between 0.5 to 5 days. The number of destinations and the destination nodes for each multicast are randomly selected. The experiment had run for 21 days, starting from Tuesday 11:00 a.m. in the first week to Tuesday 1:00 p.m. in the third week. The first week is used as a warm-up period for nodes to accumulate network information.

I use the following metrics to evaluate the performance of the proposed schemes: (1) *cost*, i.e., the average number of hops used for each destination to receive a data packet; (2) *delivery ratio*, i.e., the ratio of the number of delivered packets to the total number of packets generated; (3) *success rate*, i.e., the ratio of multicast jobs that meet the delay constraints to the total number of multicast jobs; and (4) *delay*, i.e., the average delay for a destination to receive a data packet.

5.5.2 Experimental Results

Fig. 5.4 compares the performance of different schemes, including “Direct Delivery” where a data packet is only delivered from the source to the destinations directly; “Epidemic” where data packets are transmitted to the destinations via epidemic routing [84]; “Social-Aware” multicast [29] which selects relays according to social-based metrics for forwarding data to the destinations; and “Centralized” and “Distributed” that stand for the proposed centralized and distributed schemes, respectively. Under the centralized algorithm, each source collects network information to compute a multicast tree.

As illustrated in the figure, the proposed distributed algorithm performs the best. Fig. 5.4a shows it achieves significantly lower cost than other schemes. More specifically, its average cost is 72% of that of Social-Aware and 83% of that of the centralized algorithm. Although it appears anti-intuitive to observe the lower performance of the centralized scheme, it is actually reasonable. As discussed earlier, the centralized solution intends to apply a deterministic transmission strategy in a nondeterministic network by delivering all data packets via a predetermined tree. This is inefficient in a mobile opportunistic network. As shown in Fig. 5.4b, the distributed algorithm shows 21% increase in the overall delivery ratio compared with the centralized algorithm and 36% increase compared with Social-Aware. Similar results are illustrated in Fig. 5.4c in terms of delay. The distributed algorithm shows over 20% decrease compared with the centralized algorithm and Social-Aware.

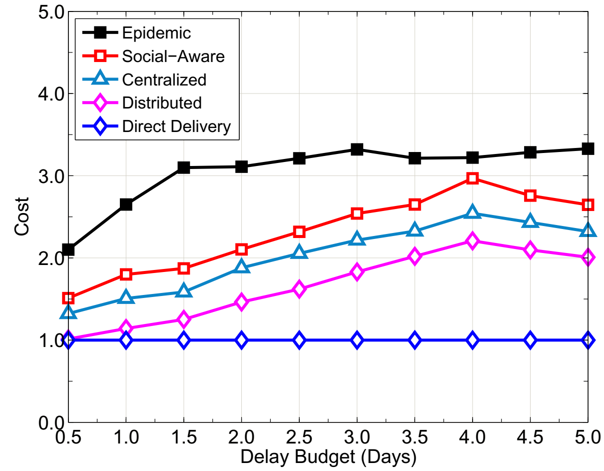
Figs. 5.5a-5.5i depict the performance variation among different days. In general, there are more communication opportunities during weekdays than weekends, due to the lower interactive activities between students and faculty on Saturday and Sunday. As a result, more packets are received during weekdays than weekends. The packets generated in weekends have longer delay compared with those in weekdays. The delay of packets generated on Friday is also high because no classes are scheduled on Friday afternoon and many offices are closed after 1:00 p.m. In addition, the cost is also higher during weekends, because there are less options to deliver data packets.

Figs. 5.5d-5.5f further zoom in to show the results in each hour of a day. Delivery ratio is high during daytime and low at night, which again shows the packet delivery heavily depends on nodal mobility. Likewise, as shown in Fig. 5.5f, the daytime delay is generally shorter than that during night, and in Fig. 5.5d the daytime cost is generally less than that during night.

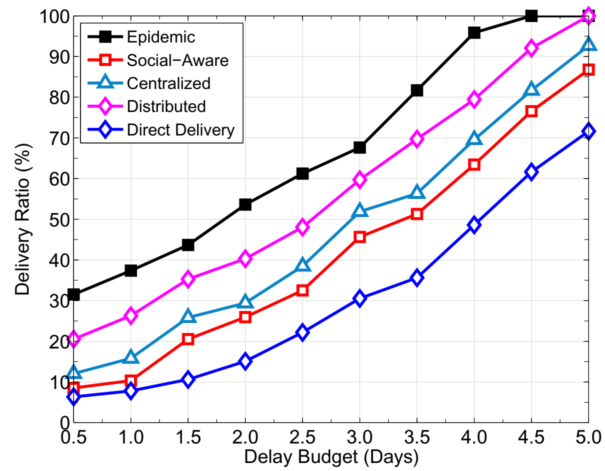
The cost, delivery ratio, and delay distributions are illustrated in Fig. 5.5g-5.5i. In general, the performance of different multicast packets varies due to the randomness in nodal mobility. As can be seen, more than 65% packets reach destinations within eight hours, and all packets are delivered within four hops.

5.6 Simulation Results

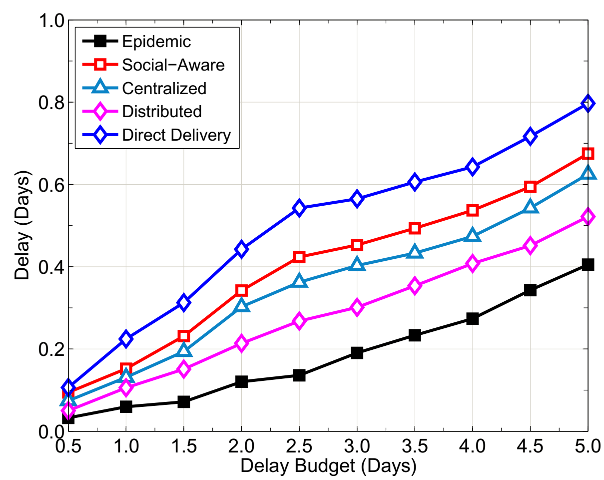
Besides the experiment discussed above, extensive simulations are carried out to learn the performance trend of the proposed algorithms under various network settings, which are not practical to evaluate by using lab equipments. The simulation codes are extracted from the



(a) Cost.

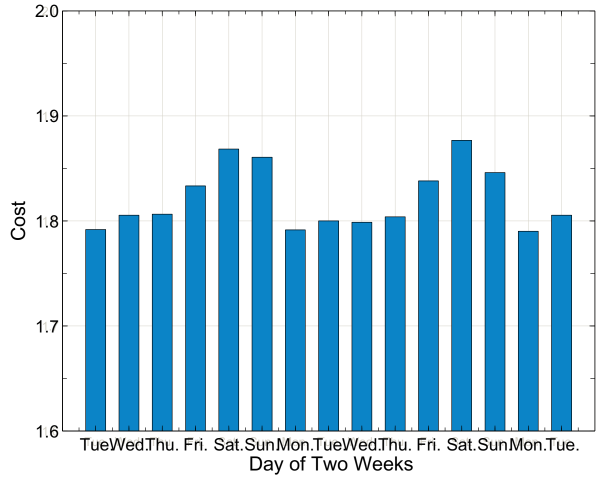


(b) Delivery ratio.

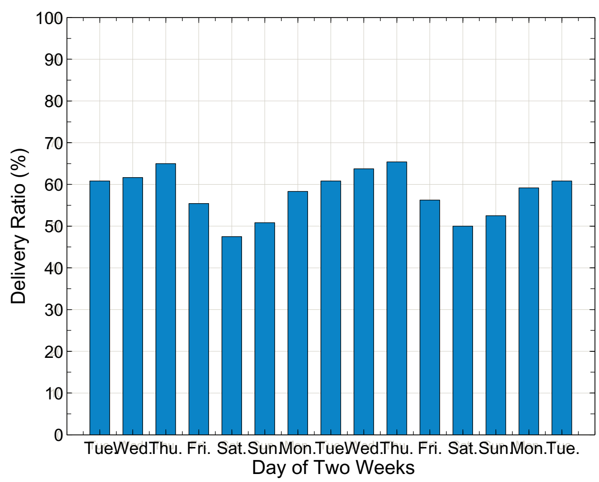


(c) Delay.

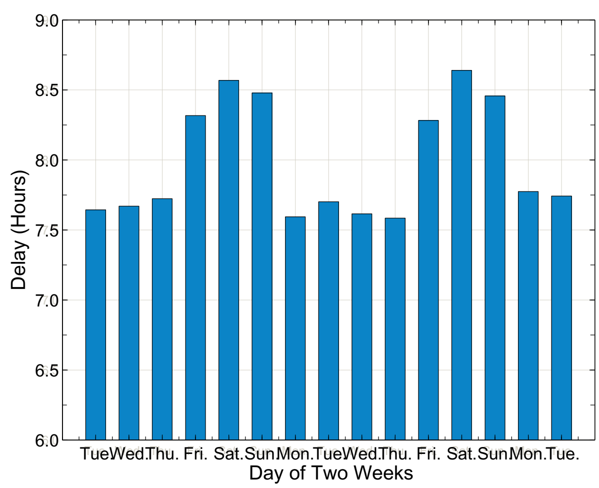
Figure 5.4: Performance comparison under the testbed experiment.



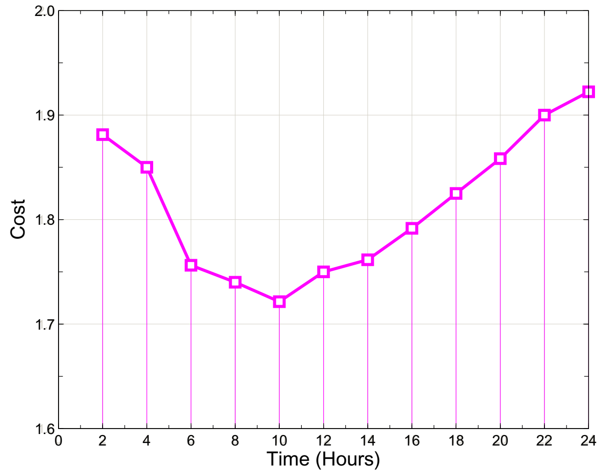
(a) Cost variation during last two weeks.



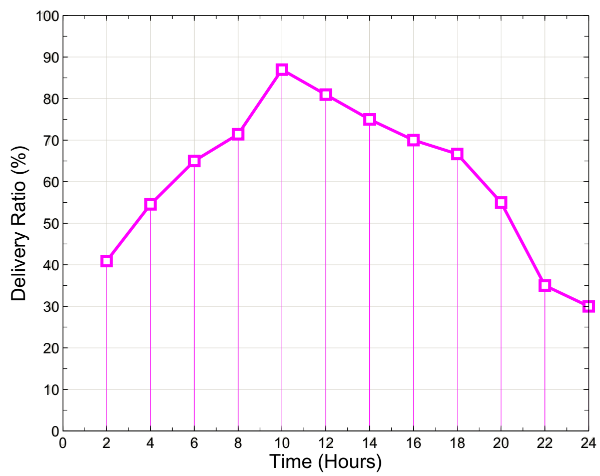
(b) Delivery variation during last two weeks.



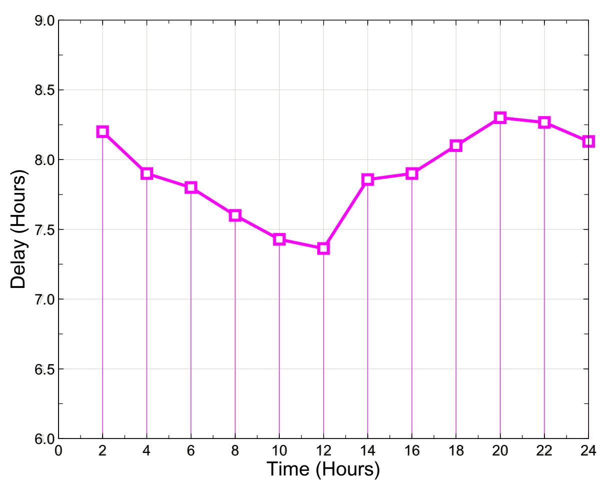
(c) Delay variation during last two weeks.



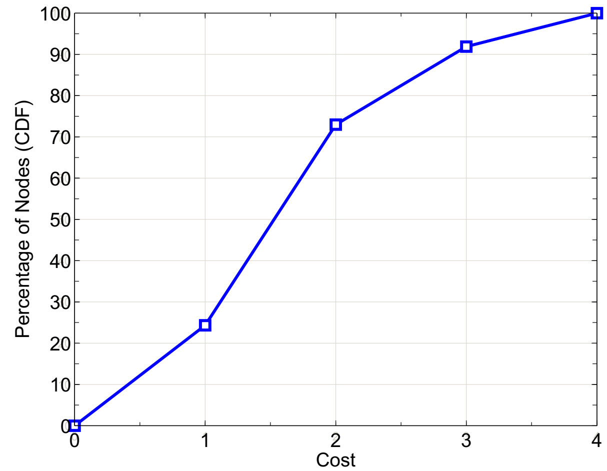
(d) Cost on the third Tuesday.



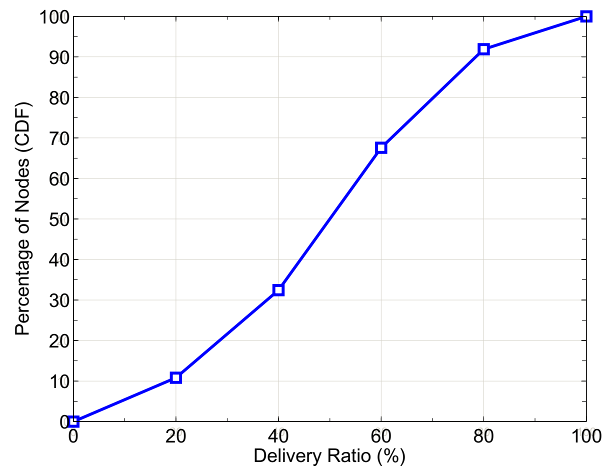
(e) Delivery ratio on the third Tuesday.



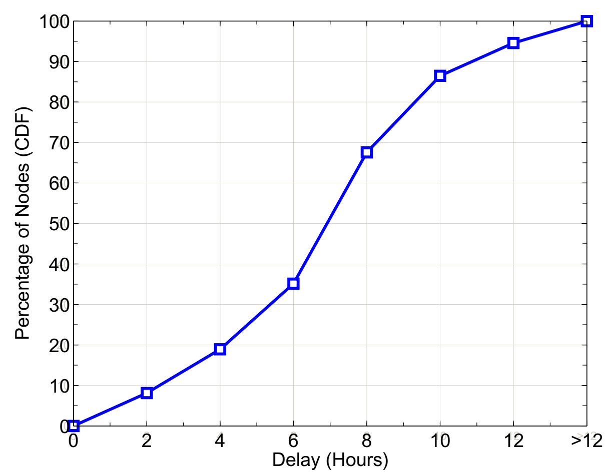
(f) Delay on the third Tuesday.



(g) Cost distribution.



(h) Delivery ratio distribution.



(i) Delay distribution.

Figure 5.5: Experimental results based on the proposed distributed scheme.

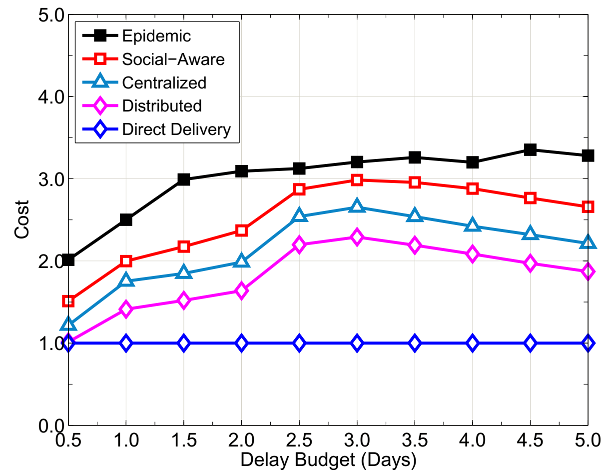
prototype implementation, and the simulation results are obtained under real-world traces and random walk mobility model. Each simulation with a delay constraint is repeated 100 times with a random source node and a fixed number of randomly selected destinations for statistical convergence. The desired delivery probability is 0.8.

5.6.1 Simulation under DieselNet Trace

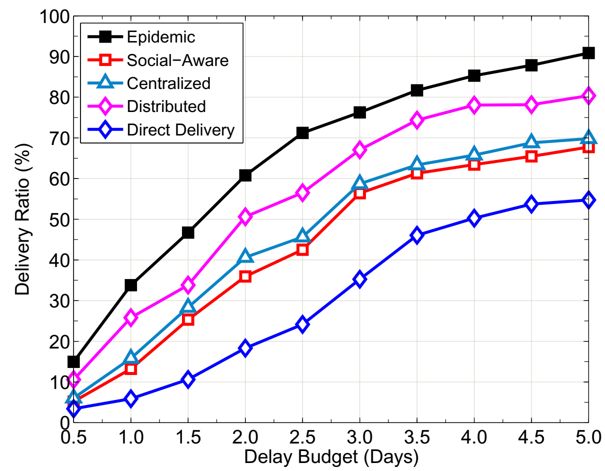
I have evaluated the proposed schemes under several real-world traces. Fig. 5.6 shows the results based on DieselNet trace [8], which comprises 33 buses, serving an area of approximately 150 square miles. Fig. 5.6 shows the simulation results of different schemes, which demonstrate a similar trend as the experimental results in Fig. 5.4.

Fig. 5.7 illustrates the performance of the proposed distributed online algorithm under different delay budgets. Increasing the delay budget results in more aggressively attempted transmissions, including longer paths, thus leading to higher average cost. At the same time, the delivery ratio and delay naturally increase with larger delay budget. However, when the delay budget is sufficiently large (e.g., larger than 3 days in these simulations), there are more options of data delivery paths. As a result, the algorithm is able to choose the one with lower cost. Accordingly, the overall cost decreases. In addition, higher probability threshold γ generally results in higher cost, delay, and average delivery ratio, because it enforces the nodes to adopt more aggressive approaches for data delivery. However, I would like to point out that the success rate (i.e., the faction of multicast jobs that meet the delay requirements) decreases when γ increases as shown in Fig. 5.7d. This is because it becomes more difficult to achieve $Pr\{\tau_X^d \leq \delta\} \geq \gamma, \forall d \in \Phi$, when γ is large.

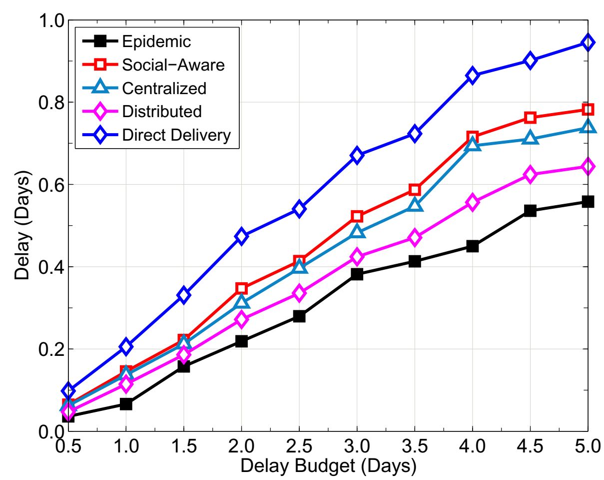
Fig. 5.8 illustrates the results when I vary the size of destination set. In general, it is more challenging to achieve a delay-constrained multicasting for a larger destination set, thus



(a) Cost.

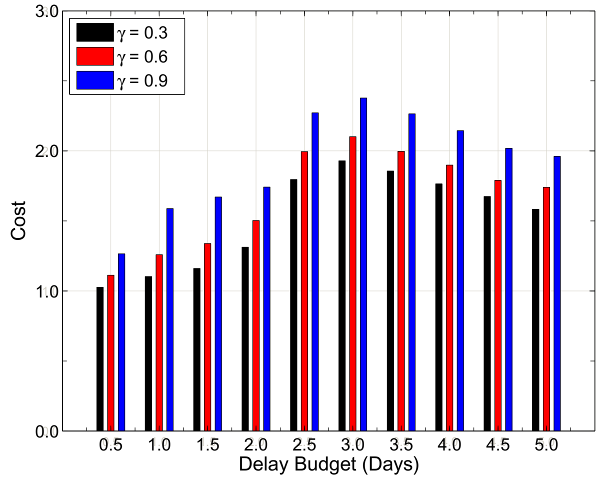


(b) Delivery ratio.

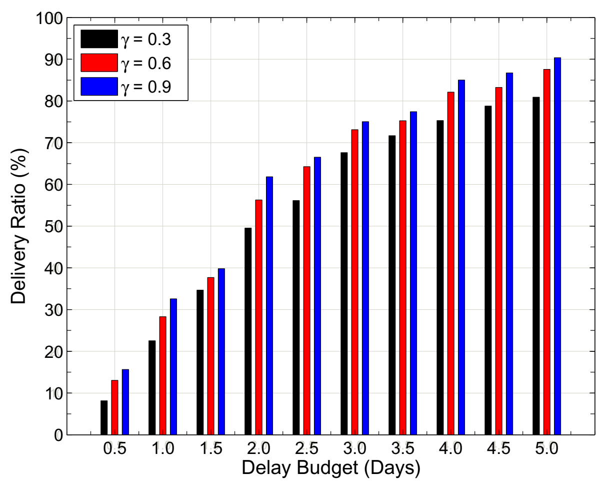


(c) Delay.

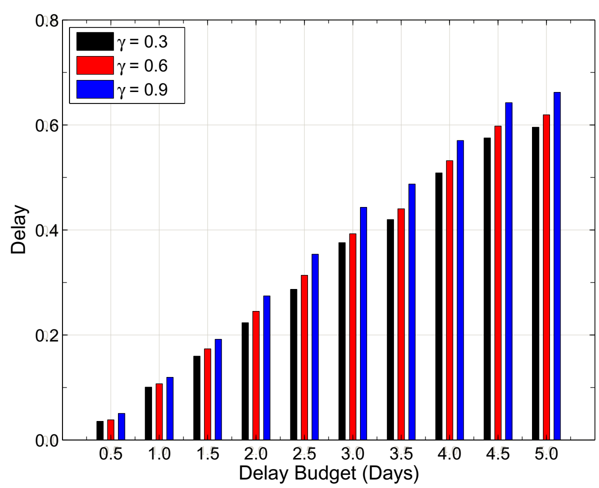
Figure 5.6: Performance comparison under DieselNet trace.



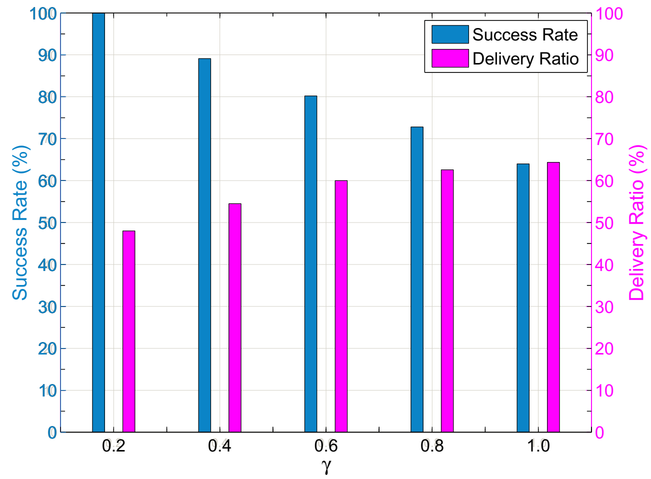
(a) Cost.



(b) Delivery ratio.



(c) Delay.



(d) Success rate vs. delivery ratio.

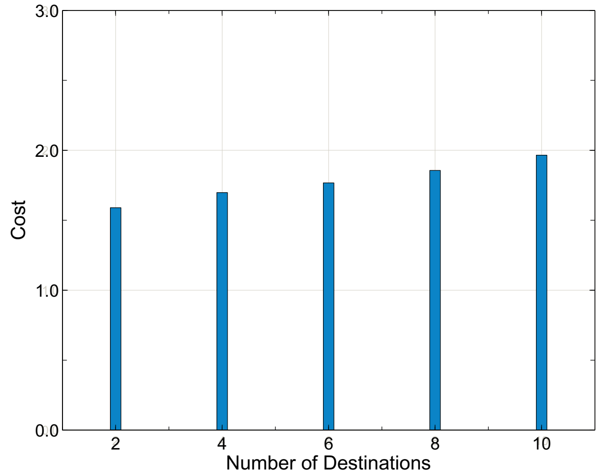
Figure 5.7: Simulation results with different delay constraints under DieselNet trace.

leading to higher cost and longer delay. At the same time, the average delivery ratio and success rate both decrease as shown in the figure.

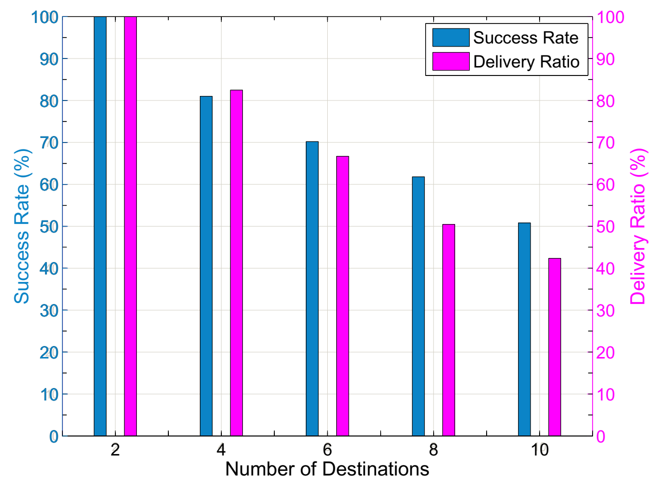
5.6.2 Simulation under Random Walk Mobility Model

Besides the above results based on traces, I have carried out a simulation under random walk mobility model, which enables convenient study of performance trend with the variation of several network parameters. More specifically, the network is deployed in an area of 20×20. The default network parameters include a network of 100 nodes and a generation rate of 0.02 (or one packet per 50 time units) per node.

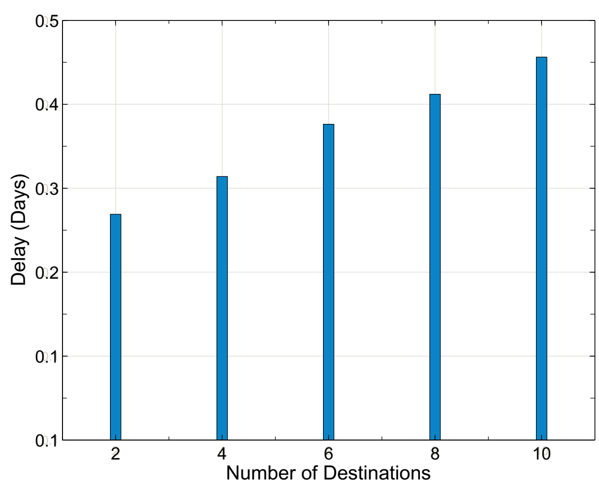
In an opportunistic mobile network, the communication capacity highly depends on the meeting opportunities among mobile nodes. As shown in Fig. 5.9a, the delivery ratio grows with the increase of network density, because the nodes have more opportunities to meet each other and exchange their packets. The impact of traffic load is illustrated in Fig. 5.9b. While the delivery ratio keeps stable at the beginning under all schemes, it starts to drop when the generation rate exceeds 0.03. In general, with a higher packet generation rate, the overall



(a) Cost.



(b) Success rate vs. delivery ratio.



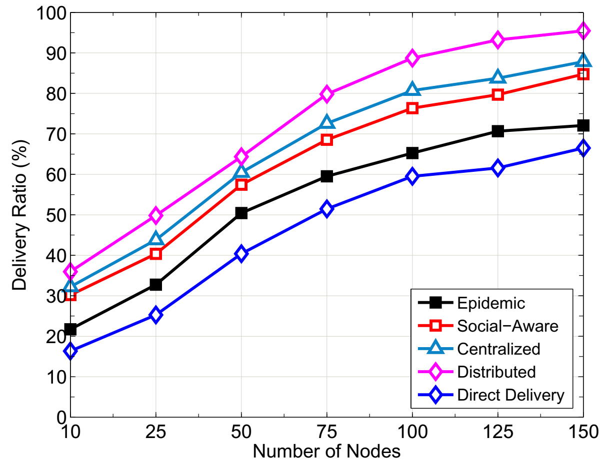
(c) Delay.

Figure 5.8: Simulation results with different size of destination sets under DieselNet trace.

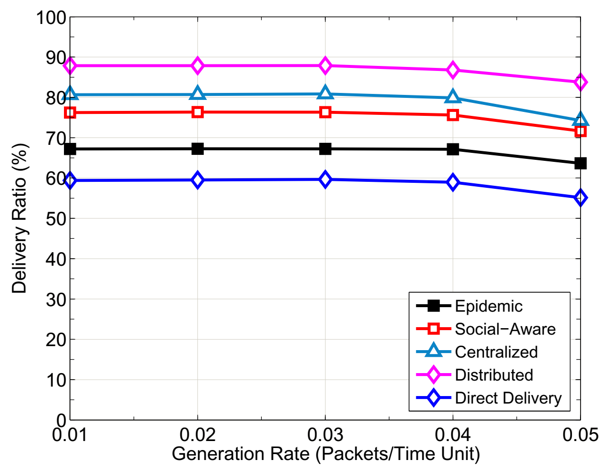
traffic load increases, resulting in more frequent data overflow and consequently lower delivery ratio. Fig. 5.9c shows that a higher delivery ratio is achieved with the increase of queue size, because more packets can be kept in the queue until they are delivered.

5.7 Conclusion

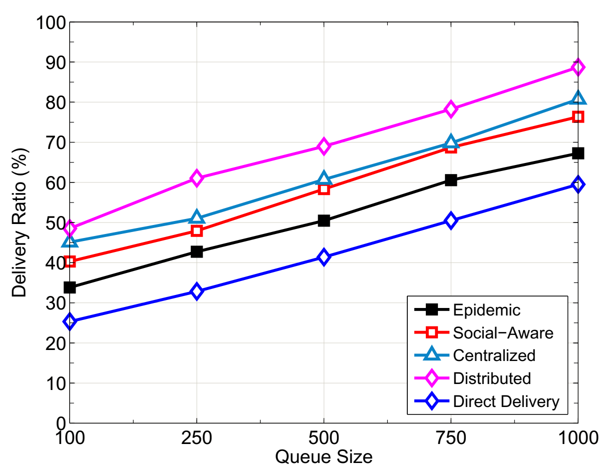
In this chapter, I have studied the problem of delay-constrained least-cost multicast in mobile opportunistic networks. I have formally formulated the problem and shown it is NP-complete. Given its NP-completeness, I have first introduced a centralized heuristic algorithm which aims to discover a tree for multicasting, in order to meet the delay constraint and achieve low communication cost. While the centralized solution can be adapted to a distributed implementation, it is inefficient in a mobile opportunistic network, since it intends to apply a deterministic transmission strategy in a nondeterministic network by delivering all data packets via a predetermined tree. Based on such observation, I have developed a distributed online algorithm that makes an efficient decision on every transmission opportunity. When a node meets another node, the former transmits the packet to the latter if the latter helps reduce the cost to deliver the packet to its destinations while reaching a desired delivery probability within a given delay budget. I have prototyped the proposed distributed online multicast algorithm using Nexus tablets and conducted an experiment that involved 37 volunteers for 21 days. I have also carried out simulations to evaluate the scalability of the proposed schemes under large-scale networks.



(a) Network density.



(b) Traffic load.



(c) Queue size.

Figure 5.9: Performance comparison under random walk mobility model.

CHAPTER 6: Conclusion

In this dissertation, I have studied QoS-aware data query and dissemination in mobile opportunistic networks. The major contributions are summarized as follows.

- I have studied the problem of data query in mobile opportunistic networks, aiming to determine an optimal transmission strategy that supports the desired query rate within a delay budget and at the same time minimizes the total communication cost. I have developed a distributed data query protocol for practical applications. To demonstrate the feasibility and efficiency of the proposed scheme and to gain useful empirical insights, I have carried out a testbed experiment by using 25 off-the-shelf Dell Streak tablets for a period of 15 days. Moreover, I have run extensive simulations to learn the performance trend under various network settings, which are not practical to build and evaluate in laboratories.

- I have proposed a QoS-aware data delivery scheme for mobile opportunistic networks. It employs QoS-aware delivery probability (*QDP*) to reflect the capability of a node to deliver data to a destination within a given delay budget and maintains a prioritized queue, where the priority is determined by a function of traffic class and dynamic redundancy in order to support efficient prioritization and redundancy control. Two experiments have been carried out to demonstrate and evaluate the proposed QoS-aware data delivery scheme. The first experiment involves multiple clusters of static Crossbow sensors that are connected by air and ground mobile nodes with controlled mobility. The second experiment is under a mobile social network setting during a period of two weeks, where the prototype is implemented by Dell Streak Android tablets carried by 23 volunteers with arbitrary and diverse mobility patterns. Moreover, simulation results have been obtained under DieselNet trace and power-law mobility model to study the scalability and performance trend. The experiments and simulations have shown that the proposed scheme achieves efficient resource allocation according to the desired delay budget, thus supporting effective QoS provisioning.

- I have studied the problem of delay-constrained least-cost multicast in mobile opportunistic networks. I have formally formulated the problem and shown it is NP-complete. Given its NP-completeness, I have first introduced a centralized heuristic algorithm which aims to discover a tree for multicasting, in order to meet the delay constraint and achieve low communication cost. While the centralized solution can be adapted to a distributed implementation, it is inefficient in a mobile opportunistic network, since it intends to apply a deterministic transmission strategy in a nondeterministic network by delivering all data packets via a predetermined tree. Based on such observation, I have developed a distributed online algorithm that makes an efficient decision on every transmission opportunity. When a node meets another node, the former transmits the packet to the latter if the latter helps reduce the cost to deliver the packet to its destinations while reaching a desired delivery probability within a given delay budget. I have prototyped the proposed distributed online multicast algorithm using Nexus tablets and conducted an experiment that involved 37 volunteers and lasted for 21 days. I have also carried out simulations to evaluate the scalability of the proposed schemes under large-scale networks.

BIBLIOGRAPHY

[1] Navid Abedini, Swetha Sampath, Rajarshi Bhattacharyya, Suman Paul, and Srinivas Shakkottai. Realtime Streaming with Guaranteed QoS over Wireless D2D Networks. In *Proc. of MobiHoc*, pages 197-206, 2013.

[2] Ian F. Akyildiz, O. B. Akan, Ozgur B. Akan, Chao Chen, Jian Fang, and Weilian Su. InterPlaNetary Internet: State-of-the-Art and Research Challenges. *Computer Networks*, 43:75-112, 2003.

[3] C. Avin and C. Brito. Efficient and Robust Query Processing in Dynamic Environments Using Random Walk Techniques. In *Proc. of IPSN*, pages 277-286, 2004.

[4] Aruna Balasubramanian, Brian Neil Levine, and Arun Venkataramani. DTN Routing as A Resource Allocation Problem. In *Proc. of SIGCOMM*, pages 373-384, 2007.

[5] Nilanjan Banerjee, Mark D. Corner, Don Towsley, and Brian Neil Levine. Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure. In *Proc. of MobiCom,* pages 81-91, 2008.

[6] Chiara Boldrini, Marco Conti, and Andrea Passarella. ContentPlace: Social-Aware Data Dissemination in Opportunistic Networks. In *Proc. of MSWiM*, pages 203-210, 2008.

[7] Stephen P. Borgatti, Ajay Mehra, Daniel J. Brass, and Giuseppe Labianca. Network Analysis in the Social Sciences. *Science*, pages 892-895, 2009.

[8] 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), 2008.

[9] 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*, 41(6):128-136, 2003.

[10] Brendan Burns, Oliver Brock Brian, and Neil Levine. MV Routing and Capacity Building in Disruption Tolerant Networks. In *Proc. of INFOCOM*, pages 398-408, 2005.

[11] Vinton G. Cerf, Scott C. Burleigh, Adrian J. Hooke, Leigh Torgerson, Robert C. Durst, Keith L. Scott, Kevin Fall, Eric J. Travis, and Howard S. Weiss. *Delay-Tolerant Network Architecture: The Evolving Interplanetary Internet*. draft-irtf-ipnrg-arch-01.txt, 2002.

[12] N. B. Chang and Mingyan Liu. Controlled Flooding Search in a Large Network. *IEEE/ACM Transactions on Networking*, 15(2):436-449, 2007.

[13] Long Cheng, Jiannong Cao, Canfeng Chen, Jian Ma, and S. K. Das. Exploiting Geographic Opportunistic Routing for Soft QoS Provisioning in Wireless Sensor Networks. In *Proc. of MASS*, pages 292-301, 2010.

[14] L. Clare, B. Clement, J. Gao, J. Hutcherson, and E. Jennings. Space-Based Networking Technology Developments in the Interplanetary Network Directorate Information Technology Program. In *Proc. of Space Mission Challenges for Information Technology*, pages 20-27, 2006.

[15] M. Conti and M. Kumar. Opportunities in Opportunistic Computing. *Computer*, 43(1):42-50, 2010.

[16] Elizabeth M. Daly and Mads Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In *Proc. of MobiHoc*, pages 32-40, 2007.

[17] Ha Dang and Hongyi Wu. Clustering and Cluster-Based Routing Protocol for Delay-Tolerant Mobile Networks. *IEEE Transactions on Wireless Communications*, 9(6):1874-1881, 2010.

[18] Qinghe Du and Xi Zhang. Base-Station Selections for Qos Provisioning over Distributed Multi-User MIMO Links in Wireless Networks. In *Proc. of INFOCOM*, pages 3038-3046, 2011.

[19] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*, chapter Overview, pages 1-20. Cambridge University Press, 2010.

[20] Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell. The BikeNet Mobile Sensing System for Cyclist Experience Mapping. In *Proc. of SenSys*, pages 87-101, 2007.

[21] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proc. of SIGCOMM*, pages 27-34, 2003.

[22] Jialu Fan, Jiming Chen, Yuan Du, Wei Gao, Jie Wu, and Youxian Sun. Geocommunity-Based Broadcasting for Data Dissemination in Mobile Social Networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):734-743, 2013.

[23] Jialu Fan, Jiming Chen, Yuan Du, Ping Wang, and Youxian Sun. DelQue: A Socially Aware Delegation Query Scheme in Delay-Tolerant Networks. *IEEE Transactions on Vehicular Technology*, 60(5):2181-2193, 2011.

[24] D. Fischer, K. Herrmann, and K. Rothermel. GeSoMo - A General Social Mobility Model for Delay Tolerant Networks. In *Proc. of MASS*, pages 99-108, 2010.

[25] Gabor Fodor. D2D Communications - What Part Will It Play in 5G? http://www.ericsson.com/research-blog/5g/device-device-communications/, 2014.

[26] Linton Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.

[27] Wei Gao and Guohong Cao. User-Centric Data Dissemination in Disruption Tolerant Networks. In *Proc. of INFOCOM*, pages 3119-3127, 2011.

[28] Wei Gao, Qinghua Li, Bo Zhao, and Guohong Cao. Multicasting in Delay Tolerant Networks: A Social Network Perspective. In *Proc. of MobiHoc*, pages 299-308, 2009.

[29] Wei Gao, Qinghua Li, Bo Zhao, and Guohong Cao. Social-Aware Multicast in Disruption-Tolerant Networks. *IEEE/ACM Transactions on Networking*, pages 1553-1566, 2012.

[30] Abebe Geletu. Solving Optimization Problems using the Matlab Optimization Toolbox - a Tutorial. 2007.

[31] M. Geva and A. Herzberg. QoSoDoS: If you can't beat them, join them! In *Proc. of INFOCOM*, pages 1278-1286, 2011.

[32] D. J. Goodman, J. Borras, N. B. Mandayam, and R. D. Yates. INFOSTATIONS: A New System for Data and Messaging Services. In *Proc. of IEEE Vehicular Technology Conference (VTC)*, pages 969-973, 1997.

[33] Dinan Gunawardena, Thomas Karagiannis, Alexandre Proutiere, Elizeu Santos-Neto, and Milan Vojnovic. Scoop: Decentralized and Opportunistic Multicasting of Information Streams. In *Proc. of MobiCom*, pages 169-180, 2011.

[34] Zygmunt J. Haas, Joseph Y. Halpern, and Li Li. Gossip-Based Ad Hoc Routing. *IEEE/ACM Transactions on Networking*, 14(3):479-491, 2006.

[35] T. Hara. Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility. In *Proc. of INFOCOM*, pages 1568-1576, 2001.

[36] Khaled A. Harras and Kevin C. Almeroth. Controlled Flooding in Disconnected Sparse Mobile Networks. *Wireless Communications and Mobile Computing*, 9(1):21-33, 2009.

[37] Dan Henriksson, Tarek F. Abdelzaher, and Raghu K. Ganti. A Caching-Based Approach to Routing in Delay-Tolerant Networks. In *Proc. of ICCCN*, pages 69-74, 2007.

[38] Ying Huang, Yan Gao, K. Nahrstedt, and Wenbo He. Optimizing File Retrieval in Delay-Tolerant Content Distribution Community. In *Proc. of ICDCS*, pages 308-316, 2009.

[39] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket Switched Networks and Human Mobility in Conference Environments. In *Proc. of WDTN*, pages 244-251, 2005.

[40] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble Rap: Social-Based Forwarding in Delay Tolerant Networks. In *Proc. of MobiHoc*, pages 241-250, 2008.

[41] Pan Hui, J. Leguay, J. Crowcroft, J. Scott, T. Friedmani, and V. Conan. Osmosis in Pocket Switched Networks. In *Proc. of ChinaCom*, pages 1-6, 2006.

[42] S. Ioannidis, A. Chaintreau, and L. Massoulie. Optimal and Scalable Distribution of Content Updates over a Mobile Social Network. In *Proc. of INFOCOM*, pages 1422-1430, 2009.

[43] S. Jain, M. Demmer, R. Patra, and K. Fall. Using Redundancy to Cope with Failures in a Delay Tolerant Network. In *Proc. of SIGCOMM*, pages 109-120, 2005.

[44] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proc. of Annual Conference on Architectural Support for Programming Languages and Operating Systems*, pages 96-107, 2002.

[45] Osman Khalid, Samee Ullah Khan, Joanna Kolodziej, Limin Zhang, Juan Li, Khizar Hayat, Sajjad Ahmad Madani, Lizhe Wang, and Dan Chen. A Checkpoint Based Message Forwarding Approach For Opportunistic Communication. In *Proc. of European Council for Modeling and Simulation*, pages 512-518, 2012.

[46] Predrag Klasnja, Beverly L. Harrison, Louis LeGrand, Anthony LaMarca, Jon Froehlich, and Scott E. Hudson. Using Wearable Sensors and Real Time Inference to Understand Human Recall of Routine Activities. In *Proc. of Ubicomp*, pages 154-163, 2008.

[47] J. Kolodziej, S.U. Khan, Lizhe Wang, N. Min-Allah, S.A. Madani, N. Ghani, and Hongxiang Li. An Application of Markov Jump Process Model for Activity-Based Indoor Mobility Prediction in Wireless Networks. In *Proc. of Frontiers of Information Technology*, pages 51-56, 2011.

[48] Jason LeBrun, Chen-Nee Chuah, and Dipak Ghosal. Knowledge Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks. In *Proc. of IEEE Vehicular Technology Conference (VTC)*, pages 1-5, 2005.

[49] Uichin Lee, Soon Young Oh, Kang-Won Lee, and M. Gerla. RelayCast: Scalable Multicast Routing in Delay Tolerant Networks. In *Proc. of ICNP*, pages 218-227, 2008.

[50] Ilias Leontiadis, Paolo Costa, and Cecilia Mascolo. Extending Access Point Connectivity through Opportunistic Routing in Vehicular Networks. In *Proc. of INFOCOM*, pages 486-490, 2010.

[51] Yifan Li, Ping Wang, D. Niyato, and Weihua Zhuang. A Dynamic Relay Selection Scheme for Mobile Users in Wireless Relay Networks. In *Proc. of INFOCOM*, pages 256-260, 2011.

[52] Z. Li and H. Shen. SEDUM: Exploiting Social Networks in Utility-Based Distributed Routing for DTNs. *IEEE Transactions on Computers*, 62(1):83-97, 2011.

[53] Ze Li, Haiying Shen, Guoxin Liu, and Jin Li. SOS: A Distributed Mobile Q&A System Based on Social Networks. In *Proc. of ICDCS*, pages 627-636, 2012.

[54] K. C.-J. Lin, Chun-Wei Chen, and Cheng-Fu Chou. Preference-Aware Content Dissemination in OpportunisticMobile Social Networks. In *Proc. of INFOCOM*, pages 1960-1968, 2012.

[55] A. Lindgren, A. Doria, and O. Scheln. Probabilistic Routing in Intermittently Connected Networks. In *Proc. of MobiHoc*, pages 19-20, 2003.

[56] Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic Routing in Intermittently Connected Networks. In *Proc. of International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR)*, 2004.

[57] H. Liu, A. Srinivasan, Y. Chen, K. Whitehouse, and J. A. Stankovic. Melange: Supporting Heterogeneous QoS Requirements in Delay Tolerant Sensor Networks. In *Proc. of INSS*, pages 93-96, 2010.

[58] Jiajia Liu, Xiaohong Jiang, Hiroki Nishiyama, and Nei Kato. Multicast Capacity, Delay and Delay Jitter in Intermittently Connected Mobile Networks. In *Proc. of INFOCOM*, pages 253-261, 2012.

[59] Alessandro Mei, Giacomo Morabito, Paolo Santi, and Julinda Stefa. Social-Aware Stateless Forwarding in Pocket Switched Networks. In *Proc. of INFOCOM*, pages 251-255, 2011.

[60] Emiliano Miluzzo, Nicholas D. Lane, Kristof Fodor, Ronald A. Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *Proc. of SenSys*, pages 337-350, 2008.

[61] Prashanth Mohan, Venkata Padmanabhan, and Ramachandran Ramjee. Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. In *Proc. of SenSys*, pages 323-336, 2008.

[62] M. Musolesi and C. Mascolo. CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks. *IEEE Transactions on Mobile Computing*, 8:246-260, 2009.

[63] Ting Ning, Zhipeng Yang, Xiaojuan Xie, and Hongyi Wu. Incentive-Aware Data Dissemination in Delay-Tolerant Mobile Networks. In *Proc. of SECON*, pages 539-547, 2011.

[64] J. Ott, E. Hyytia, P. Lassila, T. Vaegs, and J. Kangasharju. Floating Content: Information Sharing in Urban Areas. In *Proc. of PerCom*, pages 136-146, 2011.

[65] A. Passarella, M. Kumar, M. Conti, and E. Borgia. Minimum-Delay Service Provisioning in Opportunistic Networks. *IEEE Transactions on Parallel and Distributed Systems*, 22:1267-1275, 2011.

[66] Agoston Petz and Christine Julien. An Adaptive Middleware to Support Delay Tolerant Networking. In *Proc. of ARM*, pages 17-22, 2008.

[67] D. Quercia, G. Di Lorenzo, F. Calabrese, and C. Ratti. Mobile Phones and Outdoor Advertising: Measurable Advertising. *IEEE Pervasive Computing*, 10:28-36, 2011.

[68] P. Ramanathan and A. Singh. Delay-Differentiated Gossiping in Delay Tolerant Networks. In *Proc. of IEEE ICC*, pages 3291-3295, 2008.

[69] Douglas S. Reeves and Hussein F. Salama. A Distributed Algorithm for Delay-Constrained Unicast Routing. *IEEE/ACM Transactions on Networking*, pages 239-250, 2000.

[70] Dave Roos. How Online Social Networks Work. http://computer.howstuffworks.com/internet/social-networking/information/how-online-social-networks-work.htm.

[71] James Scott, Richard Gass, Jon Crowcroft, Pan Hui, Christophe Diot, and Augustin Chaintreau. Crawdad trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29), 2009.

[72] W. Richard Scott and Gerald F. Davis. *Organizations and Organizing*, chapter Networks In and Around Organizations. Pearson Prentice Hall, 2003.

[73] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks. In *Proc. of International Workshop on Sensor Network Protocols and Applications*, pages 30-41, 2003.

[74] U. Shevade, H. Song, L. Qiu, and Y. Zhang. Incentive-Aware Routing in DTNs. In *Proc. of ICNP*, pages 238-247, 2008.

[75] Addisalem Negash Shiferaw, Vasile-Marian Scuturici, and Lionel Brunie. Interest-Awareness for Information Sharing in MANETs. In *Proc. of MDM*, pages 342-347, 2010.

[76] Tara Small and Zygmunt J. Haas. The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way). In *Proc. of MobiHoc*, pages 233-244, 2003.

[77] Tara Small and Zygmunt J. Haas. Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks. In *Proc. of WDTN*, pages 260-267, 2005.

[78] R. Southwell, Xu Chen, and Jianwei Huang. QoS Satisfaction Games for Spectrum Sharing. In *Proc. of INFOCOM*, pages 570-574, 2013.

[79] W. Spearman, J. Martin, and J.L. Gao. Adaptive QoS in 802.11e Wireless Networks for Lunar Communications. In *Proc. of IEEE Aerospace Conference*, pages 1-13, 2008.

[80] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proc. of WDTN*, pages 252-259, 2005.

[81] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case. *IEEE/ACM Transactions on Networking*, 16(1):77-90, 2008.

[82] Wesley W. Terpstra, Jussi Kangasharju, Christof Leng, and Alejandro P. Buchmann. BubbleStorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search. In *Proc. of SIGCOMM*, pages 49-60, 2007.

[83] Philip Tsao, Shin-Ywan (Cindy) Wang, and Jay L. Gao. Space QoS Framework over a Delay/Disruption Tolerant Network. In *Proc. of IEEE Aerospace Conference*, pages 1-5, 2010.

[84] A. Vahdat and D. Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks, Technical report, Duke University, 2000.

[85] Yu Wang and Hongyi Wu. DFT-MSN: The Delay Fault Tolerant Mobile Sensor Network for Pervasive Information Gathering. In *Proc. of INFOCOM*, pages 1-12, 2006.

[86] Yu Wang and Hongyi Wu. Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A New Paradigm for Pervasive Information Gathering. *IEEE Transactions on Mobile Computing*, 6(9):1021-1034, 2007.

[87] Yu Wang, Hongyi Wu, Feng Lin, and Nian-Feng Tzeng. Cross-layer Protocol Design and Optimization for Delay/Fault-Tolerant Mobile Sensor *Networks. IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Delay and Disruption Tolerant Wireless Communication*, 26(5):809-819, 2008.

[88] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications,* pages 1-27. Cambridge University Press, 1994.

[89] J. Widmer and J. Y Le Boudec. Network Coding for Efficient Communication in Extreme Networks. In *Proc. of WDTN*, pages 284-291, 2005.

[90] Hongyi Wu, Yu Wang, Ha Dang, and Feng Lin. Analytic, Simulation, and Empirical Evaluation of Delay/Fault-Tolerant Mobile Sensor Networks. *IEEE Transactions on Wireless Communications*, 6(9):3287-3296, 2007.

[91] J. Yallouz and A. Orda. Tunable QoS-Aware Network Survivability. In *Proc. of INFOCOM*, pages 944-952, 2013.

[92] Bo Yang and A. R. Hurson. A Content-Aware Multimedia Accessing Model in Ad Hoc Networks. In *Proc. of ICPADS*, pages 613-619, 2005.

[93] Zhipeng Yang, Ha Dang, and Hongyi Wu. Online Estimation of Effective Delivery Capability in Delay-Tolerant Mobile Sensor Networks. In *Proc. of ICDCS Workshops*, pages 228-233, 2008.

[94] Zhipeng Yang, Ting Ning, and Hongyi Wu. Distributed Data Query in Intermittently Connected Passive RFID Networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(10):1972-1982, 2013.

[95] Zhipeng Yang and Hongyi Wu. FINDERS: A Featherlight Information Network With Delay-Endurable RFID Support. *IEEE/ACM Transactions on Networking*, 19(4):961-974, 2011.

[96] Boying Zhang, Jin Teng, Xiaole Bai, Zhimin Yang, and Dong Xuan. P3-Coupon: A Probabilistic System for Prompt and Privacy-Preserving Electronic Coupon Distribution. In *Proc. of PerCom*, pages 93-101, 2011.

[97] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance Modeling of Epidemic Routing. *Elsevier Computer Networks*, 51(10):2867-2891, 2007.

[98] W. Zhao, M. Ammar, and E. Zegura. Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms. In *Proc. of WDTN*, pages 268-275, 2005.

[99] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *Proc. of MobiHoc*, pages 187-198, 2004.

[100] K. Zhu, W. Li, and X. Fu. SMART: A Social and Mobile Aware Routing Strategy for Disruption Tolerant Networks. *IEEE Transactions on Vehicular Technology*, 63(7):3423–3434, 2014.

Liu, Yang Bachelor of Engineering, Harbin Engineering University, China, Spring 2008; Master of Engineering, Harbin Engineering University, China, Fall 2010; Master of Science, University of Louisiana at Lafayette, Spring 2012; Doctor of Philosophy, University of Louisiana at Lafayette, Fall 2014.
Major: Computer Engineering
Title of Dissertation: Qos-Aware Data Query and Dissemination in Mobile Opportunistic Networks.
Dissertation Director: Dr. Hongyi Wu
Pages in Dissertation: 119; Words in Abstract: 327

ABSTRACT

Mobile opportunistic networks are formed by mobile users who share similar interests and connect with one another by exploiting Bluetooth and/or WiFi connections. Such networks not only re-assemble the real-world interaction between people, but also can effectively propagate data among mobile users. This dissertation focuses on QoS-aware data query and dissemination in mobile opportunistic networks.

Firstly, I develop a distributed data query protocol for practical applications. To demonstrate the feasibility and efficiency of the proposed scheme and to gain useful empirical insights, I carry out a testbed experiment by using 25 off-the-shelf Dell Streak tablets for a period of 15 days. Moreover, extensive simulations are carried out to learn the performance trend under various network settings, which are not practical to build and evaluate in laboratories.

Secondly, the QoS-aware delivery probability (QDP) is introduced to reflect the capability of a node to deliver data to a destination within a given delay budget. Two experiments are carried out to demonstrate and evaluate the proposed QoS-aware data delivery scheme. Moreover, simulation results are obtained under DieselNet trace and power-law mobility model to study the scalability and performance trend. Our experiments and simulations demonstrate that the proposed scheme achieves efficient resource allocation

according to the desired delay budget, and thus supports effective QoS provisioning.

Finally, I study the problem of delay-constrained least-cost multicast in mobile opportunistic networks. I formally formulate the problem and show it is NP-complete. Given its NP-completeness, I explore efficient and scalable heuristic solutions. I first introduce a centralized heuristic algorithm which aims to discover a tree for multicasting, in order to meet the delay constraint and achieve low communication cost. I develop a distributed online algorithm that makes an efficient decision on every transmission opportunity. I prototype the proposed distributed online multicast algorithm using Nexus tablets and conduct an experiment that involves 37 volunteers and lasts for 21 days to demonstrate its effectiveness. I also carry out simulations to evaluate the scalability of the proposed schemes.

BIOGRAPHICAL SKETCH

Yang Liu received a Bachelor of Engineering in Electrical Engineering and its Automation from Harbin Engineering University, China, in Spring 2008, a Master of Engineering in Control Theory and Control Engineering from Harbin Engineering University, China, in Fall 2010, and a Master of Science in Computer Engineering from the University of Louisiana at Lafayette in Spring 2012. He earned a Doctorate of Philosophy in Computer Engineering at the Center for Advanced Computer Studies, University of Louisiana at Lafayette, in Fall 2014. His current research interests include delay-tolerant networks, radio frequency identification (RFID) systems, wireless sensor networks and integrated heterogeneous wireless systems.