User Mobility for Opportunistic Ad-Hoc Networking

by

Jing Su

A thesis submitted in conformity with the requirements
for the degree of Masters of Science
Graduate Department of Computer Science
University of Toronto

Abstract

User Mobility for Opportunistic Ad-Hoc Networking

Jing Su

Masters of Science

Graduate Department of Computer Science

University of Toronto

2005

As mobile devices become increasingly pervasive and commonly equipped with short-range radio capabilities, we observe that it might be possible to build a network based only on pair-wise contact of users. By using user mobility as a network transport mechanism, devices can intelligently route latency-insensitive packets using power-efficient short-range radio. Such a network could provide communication capability where no network infrastructure exists, or extend the reach of established infrastructure. To collect user mobility data, we ran two user studies by giving instrumented PDA devices to groups of students to carry for several weeks. We evaluate our work by providing empirical data that suggests that it is possible to make intelligent routing decisions based on only pair-wise contact, without previous knowledge of the mobility model or location information.

Dedication

I dedicate this thesis to Mom, Dad, Jenny, and Juliana for their constant support.

In memory of the Tuft.

> *For God's sake, stop researching for a while and begin to think!*
>
> – `fortunemod`

Acknowledgements

A big *Wes'syde* shout-out to my family on the West Coast.

A big *East-side* shout-out to friends and lab-mates here in TO.

And, of course, last but not least, a big thank you to my Procrastination Buddy.

Contents

List of Figures

Chapter 1

Introduction

Mobile devices have enjoyed tremendous growth in recent years, and this trend of growth is projected to continue. A significant motivation for the adoption of these mobile devices is easy access to wireless network services. Most mobile devices today either come with built-in wireless access technologies or have expansion options for adding this capability.

Wireless access technologies in most mobile devices can be divided into three categories: long range infrastructure, short range infrastructure, and ad hoc. Common technologies used for these types of networks are cellular, WiFi, and Bluetooth, respectively. We consider ad hoc networks where participants are mobile to be Mobile Ad Hoc Networks (MANETs). A more detailed discussion of various radio technologies is covered in Section 2.1.

Mobile devices do not rely solely on long range infrastructure networks, such as cellular, because short range infrastructure networks can provide advantages in terms of speed, cost, and power efficiency. Despite improvements in battery technology, pressure to include more processing power and functionality into slimmer form factors continues to place power constraints on mobile devices. A constant balance must be kept between providing functionality and ensuring acceptable operational life. In particular, radio transmission is a significant consumer of power on mobile devices [53], and the design and management of radio operation remains an active area of research. Though short range radio can provide savings in power (shorter radio range results in less power use), long range networks can provide service when outside of hotspot coverage area. Thus it can be beneficial to have multiple layers of connectivity, to take advantage of each respective medium's strengths.

In this thesis, we postulate that it might be possible to use mobile ad hoc networking to provide extended network availability outside of infrastructure coverage. Furthermore we postulate that there is sufficient regularity in peoples' contact patterns to aid in routing through such a MANET.

Many previous works have suggested that people usually do not move randomly [31, 47], and as a result have regular patterns of meeting people. We hypothesize this may even hold true between pairs of strangers. For example, it may be obvious that John meets Jane for a staff meeting every Monday morning. However, it may also be the case that John and Dave, who are total strangers to each other, happen to take the same bus to work every morning. Though individuals might not be aware of it, their daily routines may create correlated connections with other individuals. If this is the case, then it may be possible to use people as packet carriers, transporting them from one hop to another, until a suitable infrastructure system can be used.

Previous research efforts have looked at tracking mobility to provide location driven services [56] as well as supporting data collection and dissemination applications. Techniques range from using animals such as whales and zebras, to tracking peoples' location via triangulation from base-stations [6]. However, location tracking is a complex and difficult task. Though there have been many efforts on location tracking [12, 26, 32, 51] in general, we observe that there have been few studies to collect trace data of user contact patterns. Contact patterns may be easier to detect, and might provide sufficient information for packet routing.

1.1 Motivation and Hypothesis

We are motivated to explore whether social contact patterns between people can be combined with multi-modal radio technologies to provide improved networking services. Specifically, we explore the possibility of using delay tolerant MANETs [35] to supplement infrastructure systems. Improvements can include increased network service availability, by supplementing available infrastructure, or providing lower cost or lower power communication mediums.

We hypothesize that pairwise contact patterns between individuals can provide sufficient information for making routing decisions across a delay tolerant MANET.

Finally, we are motivated to perform our experiments using available consumer electronics. The experience will provide insights into the readiness of current devices for participating in cooperative networking environments, especially with respect to power management.

1.2 Approach and Challenges

Our approach is mainly composed of two separate phases. In the first phase, we instrument data collection devices and obtain pairwise contact traces. In the second phase, we use these data traces as input in several analysis and network simulation programs to help characterize the data.

For the data collection phase, we instrument twenty popularly available personal digital assistant (PDA) devices, equipped with Bluetooth radios. Power management is the greatest challenge for impoverished mobile devices. Most available consumer devices are unable to operate continuously for a whole day, and regular radio usage significantly increases power demands. Our approach is to start with a device with good battery longevity, and carefully manage radio use. Overly-aggressive radio use can result in premature battery drain, and render the device unable to collect more data until recharged. Conversely, overly-lax radio use can result in many missed data collection opportunities.

Custom software was written for the mobile devices which periodically operated the radio to search for other nearby devices. We used a synchronized radio protocol which allowed the sleep times to be maximized and transmission and listen times to be kept very short. In order to minimize radio usage during the short radio activity period, devices only briefly broadcast their ID number. Devices within radio range of the broadcast record the ID number in a data trace. This provides pairwise contact information, but no details regarding connection quality, connection duration, or potential bandwidth.

We distributed these instrumented devices to individuals to carry for several weeks time. The data trace was then collected and characterized for possible patterns, trends, and potential networking opportunities.

The analysis phase was done offline using the data trace collected from the previous phase. Simulation software was used to determine the kinds of patterns and theoretical capacities of a potential network built on the data trace. In particular, we examined three areas: the role each device played in the network, in particular its importance in bridging between other devices, the latency characteristics of packets routed over a simulated network, and various routing methods. We explored three different routing strategies: epidemic routing, aggregate future knowledge routing, and gossip route learning.

These three routing strategies are chosen to characterize the effects of replication, future knowledge, and online learning. Epidemic propagation always guarantees least-latency from source to destination, but requires unlimited replication. Aggregate future knowledge routing allowed us to explore replication restriction, but routing based on some future knowledge. Finally, we explored a gossiping protocol that allowed the use of restricted replication while requiring no future knowledge. Unfortunately, our collected data trace does not contain sufficient information for testing existing routing protocols. Therefore we chose these three general routing strategies to characterize trends and upper bounds.

1.3 Contribution

We hypothesize that users have regular schedules, destination locations, and social circles, which result in regular and predictable contact patterns. These patterns can be used to provide a supplemental ad hoc networking layer for delay tolerant packets.

This work provides empirical trace data of contact patterns from individuals. We then characterize this data and the potential networking opportunities by using simulators of various routing methods and oracle knowledge levels. This work also provides an account of our insights and experiences with developing and instrumenting devices for the data collection and deploying them in two user studies.

Our results show that despite the sparse community of volunteers, there is promising evidence that contact patterns are regular enough to provide significant networking opportunities. We also show that current consumer devices are lacking in power management technologies and APIs[1]. It is currently not feasible to provide extensive wireless networking on available mobile devices and expect battery life to last a full day.

The rest of this thesis is organized as follows. First we provide some background on various network systems, routing strategies in ad hoc networks, and movement simulation and trace collection efforts in Chapter 2. Then in Chapter 3 we provide details on our experimental setup, the device instrumentation, and user studies. Chapter 4 examines the data trace collected from the experiments, and shows characteristics of various simulated networks using the different routing strategies. Finally, we close with a discussion in Chapter 5 and conclusions and future work in Chapter 6.

[1] Application Programming Interface

Chapter 2

Background

In this section we start by presenting, in general, the three types of radio networks we consider, and the various advantages and disadvantages of each system. Following that we present some related work on the three broad categories of routing strategies that we consider for our analysis. Finally we provide some related works in mobility simulation and trace gathering.

2.1 Radios and Networks

There are typically three types of radio networks that are used with mobile devices. Long range infrastructure, short-range infrastructure, and ad hoc. Abstractly we consider wireless networks to be a collection of antennas, capable of two-way communication. Each independent antenna can be considered a *node*. Concretely, nodes can be radio towers, satellites, laptops, cellular phones, etc. Depending on the situation being examined, nodes can be geographically placed at specific locations, randomly placed, or even mobile.

2.1.1 Long Range Infrastructure

Long range infrastructure systems, such as cellular networks, provide fairly comprehensive network coverage and service. Within urban and suburban areas, cellular networks are expected to have reliable and fairly consistent coverage. With the rising popularity of cellular telephones, it is not uncommon to find reliable coverage in some rural areas as well.

However, cellular radios are hindered by two main limitations. First cellular networks depend on complex infrastructure. For service providers, placing and maintaining transmission towers is a complex and expensive operation [55]. Despite the best efforts of service providers, “dead spots”, where the signal cannot reach, exist. Second, cellular networks work across regulated frequencies. Access to cellular networks typically require subscriptions, and data transmissions are charged.

Recent products have become available to provide indoor repeaters and relays for cellular networks [33, 61, 66]. These devices can help bridge service gaps in many locations, but still rely on infrastructure and subscriptions to service providers. To the best of our knowledge, there is no system available to provide private “mini-cellular” networks, analogous to a private LAN. Due to frequency regulations, it is unlikely such products would become available to consumers.

2.1.2 Short Range Infrastructure

Short range infrastructure systems, such as 802.11 (WiFi) [68], have enjoyed tremendous growth in recent years [64]. They operate at unregulated frequencies and are relatively inexpensive to obtain and install. Locations serviced by a base-station are usually referred to as “hotspots”. Security and access issues aside, it is easy for end users to install base-stations to provide service where needed.

Despite this ease, hostspot availability still relies on the installation of base-stations. Because short range radios naturally have smaller coverage areas, the chances of not being in a hotspot are greater. Even within buildings with fairly comprehensive WiFi coverage, signal strength and network qualities can greatly differ between various locations [14]. Furthermore, because these systems use unregulated frequencies, there is significant interference from other devices such as competing wireless systems, cordless phones, and microwaves. There exist some initiatives to provide comprehensive hotspot coverage within urban centers [18, 57, 65]. However, at the time of this writing, those initiatives are not yet fully implemented.

A major concern with short range infrastructure radios is power consumption [11, 16]. Though base-stations are "plugged in" and have few power concerns, client radios must be conservative with radio power consumption. The greater the coverage distance, the more power is consumed in radio transmissions. Furthermore, as the number of devices in an area grows, additional medium contention and communication overhead will increase power demands.

2.1.3 Ad Hoc

Unlike the other infrastructure systems, ad hoc networks [2, 23, 54] have no a priori hierarchy (though some protocols allow ad hoc formation of hierarchy [9, 28]). There are no base-stations – every member of the network can be both a client and a router. Assuming there are nearby clients, network availability can be achieved by connecting through various other clients; without need for infrastructure near that location.

Since there are no fixed base-stations, providing network availability and routing can be a difficult challenge. The network must constantly cope with peer failures, disconnects, and partitions. Furthermore, in addition to transmitting packets belonging to the client, each node must also transmit packets of other clients routed through them. This additional network traffic can be a significant source of additional power consumption. Providing a mechanism for fair exchange of resources is an area of much active research [4, 5, 19, 22, 25, 52, 63].

2.2 Routing

2.2.1 Epidemic Propagation

Previous works have looked into epidemic algorithms for data propagation [7] using node mobility. Epidemic propagation approximately models the spreading behavior of infectious diseases. The basic premise is that a virus can replicate an unlimited number of times and infect an unlimited number of hosts. All hosts can move and, with a certain level of probability, make contact with other hosts. Infected hosts will infect healthy hosts. Once a host is infected, it cannot be re-infected and never rids itself of the virus.

Instead of grimly modeling infection, epidemic algorithms model a method for dispersing data through a network, whether to reach all nodes or a specific one. Nodes always replicate new data to other nodes they come into contact with, and nodes never accept more than one copy of the same piece of data. Epidemic algorithms have the desirable feature of always guaranteeing to find the least latency path from any source to any destination. Unfortunately, it also requires an assumption of unbounded storage space and time.

Many works have used epidemic algorithms to study the effects of mobility in ad hoc networks. Grossglauser and Tse [27] mathematically show that with random node mobility, unlimited storage, and unbounded time, mobility reduces medium contention and improves successful message delivery to the destination. Davis et al. [20] proposes using wearable computers for carrying and forwarding data between infrastructure points, to bridge highly partitioned groups of users.

Epidemic algorithms have also been used for collecting sensor data. Examples of such mobile ad-hoc networks include ZebraNet [39, 46] and SWIM [59] which have been created and physically deployed in real environments, using zebras and whales for nodes, respectively. Zhao et al. [67] use mobility for data delivery in MANETS, similar to DataMULEs [58]. The objective of the DataMULE system is sensor data collection, and relies on a two tier hierarchy: sensors distributed randomly, and collection devices placed on mobile objects (city vehicles) which have a known mobility patterns (buses) or known possible mobility routes (cars on roads). Zhao et al. also builds an ad hoc sensor data collection system, where the deployed devices are of two separate classes: immobile and cheap sensors, and more sophisticated sensors which have some limited mobility. The sensors of limited mobility can adjust their position in order to maximize their ability to bridge partitions of sensors, and to forward data.

Though the motivation for our work is to potentially provide networking, our study of pairwise contact does not preclude the possibility of building a data collection system. What distinguishes our work is that we have no pre-determined mobility model and have no control over node mobility.

2.2.2 Delay Tolerant Networks

Jain et al. [35] provides a summary of delay tolerant networks (DTN), and the various effects of oracle power and node capacity in coping with DTNs. Their work provides a general characterization of the different classes, and summarizes several directions for future research. In general, they find that least knowledge algorithms tend to perform worse than algorithms with more knowledge. However, limited additional knowledge can still provide a large boost in performance. In this work we examine using aggregate future knowledge oracles for our first-handoff preference routing.

Other works have examined the possible uses of DTNs as middleware, such as in [13, 62]. The PostMANET system uses the postal system for providing internet content delivery by taking advantage of large capacity mass storage devices. The system anticipates and pre-caches content related to the current requests being serviced, allowing PostMANET to hide some of the high latency aspects of the network.

Chen et al. [17] hypothesizes that mobile users can be clustered into groups. Certain mobile users can belong to more than one cluster, and thus can be used as agents for cross cluster transport. Within clusters, standard ad hoc protocols can be used. Nodes share data on recently met nodes, which is used for calculating a “trajectory” for bridging clusters until the destination is reached.

2.2.3 Gossip Protocols

Many works have explored the use of gossip protocols for many distributed applications, for complexity analysis, coverage, resource location, replication coordination, aggregation, and synchronization [29, 36, 41].

Goel [24] explores methods for maintaining view consistency in a distributed and replicated filesystem. Gossiping protocols are used for reconciliation vector maintenance. Because not all replicas undergo direct communication, gossiping information is used for maintaining acknowledgment even across multiple degrees of separation.

Li et al. [44] explore a method for constraining gossip messages to regions, to further reduce the overhead necessary to achieve routing. Their work relies on nodes having some positional information about destination nodes, and forming an elliptical region of gossip constraint. Furthermore their simulations target MANETs on a two dimensional plane with random placement and mobility. It is not clear how Regional Gossip would fare in a DTN, potentially with periodically predictable patterns.

The focus of this work is not the evaluation of advanced routing and information dissemination protocols, so we will only explore the use of simple global flooding gossip algorithms.

2.3 Mobility: Simulation, Modeling, and Tracing

2.3.1 Simulation and Modeling

Most studies of MANETs, whether for data retrieval, distribution, or networking, use simulated movement and theoretical mobility models [30, 34, 43, 45]. Though some of these studies go to great lengths to model the physical and geographical movement of nodes, and to an extent a task and objective based destination selection, they do not capture the sociological aspect of user behavior. Camp et al. [15] shows that different mobility models can have significant impacts on routing performance; and that a model most closely matching the intended scenario should be used.

Other studies have looked at modeling sociological behavior, both for simulations and characterizing traces. Herrmann [31] uses social constraint modeling to characterize simulation behavior, rather than geographical constraints. The method consists of determining cliques for nodes that make contact. It is then assumed that these nodes meet for specific reasons, and thus an anchor point is created to represent a specific meeting task. Nodes are then constrained such that they must be able to visit these anchors without time overlap (i.e. nodes cannot be at two places at once).

Instead of simulating user mobility, we collect empirical data of user contact patterns. Thus our network analysis and simulation tools use empirical traces, rather than artificially generated ones. For future work, we hope that our empirical data can help explore and verify mobility simulation models.

2.3.2 Trace Collection

A more realistic approach is to obtain traces from a real environment and use these traces as a model for simulation. Jetcheva et al. [37] used a fleet of city buses as mobile nodes to obtain mobility trace data. They then simulated potential latency and routing characteristics, assuming various radio coverage models, using the collected data. Our work is unique in that we need not make assumptions about radio coverage or mobility models. Our radio coverage is inherently captured in the radio technology we use in our experiments. Unfortunately, because our data only captures pairwise contact with no location information, it is not possible to extend and apply our data to various other radio coverage models. Furthermore, instead of knowing pre-determined paths (like that of city buses), we collect pairwise contact traces of real people, for whom we have no predetermined mobility model.

Kotz et al. [42] provide an extensive study of large wireless network environments. Their work provides supplemental research on wireless activity and metrics that our study does not address. However, their work focuses on traces of WiFi clients interacting with base stations.

Our work expands on their efforts by focusing on detecting user mobility and peer-to-peer contact patterns. Instead of studying infrastructure, we aim to study user interaction patterns, to potentially form a supplemental networking platform based on user mobility and contact.

Chapter 3

Experiment

We investigate whether real user mobility and opportunistic pair-wise interactions between users can be exploited to provide data communication. The secondary objective of the experiment is to determine the “readiness” of current consumer products for continuous participation in wireless networks.

We also investigate the characteristics of forming peer-to-peer as well as peer-to-infrastructure networks. Thus we deploy two types of instrumented devices: mobile and stationary. Mobile devices will be carried by users, and stationary devices will be hidden in certain high traffic locations. It should be noted that this distinction only refers to the mobility of the devices. All devices are identical in capability, capacity, and functionality.

Deploying data collection devices to real users requires careful design considerations. The following section describes the design requirements of the experiment in terms of data collection and user impact. We then describe the implementation concerns and decisions with respect to the design requirements, followed by the deployment of the experiment in two separate user studies.

3.1 Design Requirements

The primary objective of the experiments is to collect trace data of pair-wise contact. The experiment does not strive to transfer real data, detect connection quality, measure bandwidth, or track user[1] location.

To address the issue of real user mobility, we need to provide users with an instrumented device to carry. The instrumentation must satisfy three requirements: there should be some motivation for the user to carry the device as often as possible, the data collection should work independent of the user's activities, and the device should last at least an eight-hour work-day.

We provided users with a featureful device, to encourage frequent carrying of the device. The instrumentation software runs invisibly in the background with minimal impact on the usability. Though we could have used specialized devices designed for this experiment, we felt that using commodity devices helps highlight our motivation of networking consumer mobile devices in interesting ways.

Our aim is to detect opportunistic pair-wise contact, even when users might not be aware of it. Contact could take place while at a meeting, waiting at an elevator, or even walking by another participant. Users may not be aware of who may or may not be a participant, and they may not be using their devices during that moment of contact. Nevertheless, it is desirable to record such contact since it presents a communication opportunity. It is highly likely that these devices will be carried in pockets or bags most of the time. Therefore we opted to use radio, which is omni-directional and does not require line-of-sight. Infrared would be a more power efficient option, but its line-of-sight requirement and susceptibility to interference from daylight or florescent lighting make it unviable for this experiment.

Power management is an important issue with mobile devices. Inadequate power management can render the device unusable and prevent it from gathering data. Since many mobile devices rely on disk-less storage, an extended power outage can result in lost user and exper-

[1]We refer to participants of the user studies as "users".

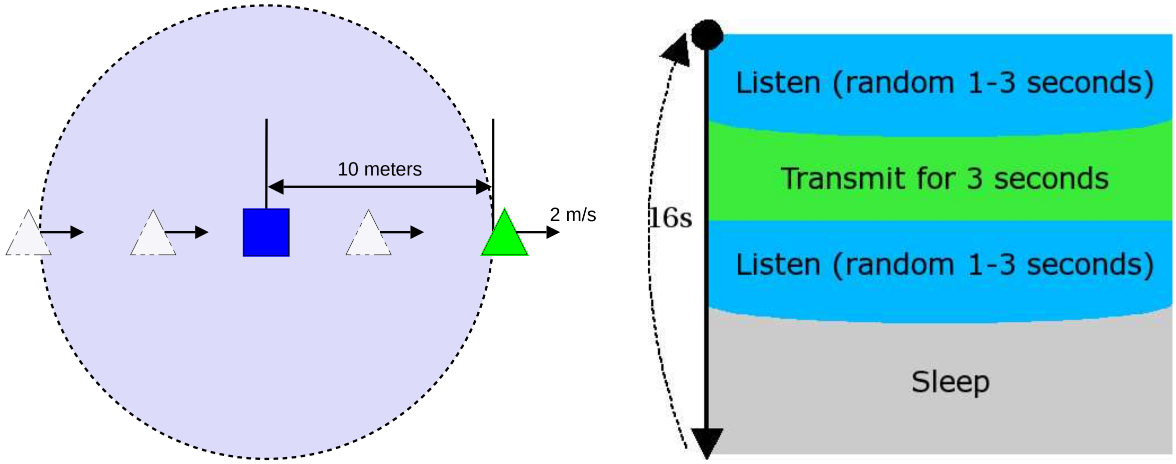
iment data. Most users likely will recharge their device at the end of the day, and many will not have opportunity to recharge them mid-day. Requiring users to recharge the device mid-day would be disruptive to their daily routine and increases the likeliness of the device being forgotten or left behind. To cover a working day, we estimated that the devices should last at least eight hours, including standard usage as well as background radio operation and data gathering.

It should be noted that security and privacy are not issues as far as the experiment is concerned. Devices do not track or share user information, and the mapping of devices to users is kept confidential. The data used for analysis is anonymized before use. At this time, we also do not consider the security and privacy concerns of actually implementing a functioning network using this method. This work is primarily concerned with determining whether such a network is feasible.

3.2 Implementation

We chose to use PDA (personal digital assistant) devices, in particular Palm Tungsten T handhelds (herein referred to as Palm devices) running the Palm Operating System (PalmOS). Because sufficient battery life is a major concern, the PocketPC platform, which usually lasts approximately ten hours under nominal usage, was not a viable option. Similarly, due to power concerns, we chose to use Bluetooth radio instead of WiFi (WiFi can consume between 10 to 50 times more power than Bluetooth in low-usage modes[2] [21]), though WiFi is currently more commonly available. The Tungsten T devices have a built-in Bluetooth radio, which is slightly more power efficient than using an add-on card. They also can be updated with any number of available third-party applications, which helped increase its appeal to the users. To gather data, we developed a custom Palm application [49, 50] to run in the background and periodically use the radio to search for other users. Because PalmOS is a single-threaded event-driven plat-

[2] low usage defined as, on average, 90% of time in sleep mode and 10% in RX and TX.



(a) Walk-By Illustration (b) Radio Protocol

Figure 3.1: Walk-by and Radio Protocol Diagrams

form, we use a self-setting alarm timer to grab background processing time. When the timer is triggered, we asynchronously operate the radio to listen for nearby devices and announce our presence. The application then sets another timer and sleeps. For most applications, this technique produced no observable hindrance to the user experience.

The frequency at which the devices announce and listen on their radios affects battery longevity. However, because we aim to capture serendipitous contact, longer sleep times may result in the device missing brief contacts. We made a best-effort attempt to have the protocol capture what we call the "walk-by", illustrated in Figure 3.1(a). Assuming a 10-meter antenna range, and an average walking speed of 2 m/s, there is a 10-second window of opportunity to detect a user walking directly past another stationary user. We recognize that this scenario does not fully capture the many ways in which individuals move and make contact, and may miss many moments of pairwise contact. However, this simply means our data traces will be conservative. A tighter detection method would capture more data and will likely provide better results.

After several implementation iterations, we developed a minimal protocol, as illustrated in Figure 3.1(b), to stretch the battery life to the target range. At the start of each user study, all Palm devices are NTP (Network Time Protocol) time synchronized so that radio usage can be minimized and the odds of successful communication increased [38]. The Bluetooth radio on the Palm devices are half-duplex, which required a scheme for allowing each device to announce their presence as well as listen for other nearby devices. In other words, while transmitting, the radio cannot "hear" other devices. At an established time epoch, all devices power their radio simultaneously. Each device will then listen on their radios for a random 1 to 3 seconds. Immediately after the random listen interval, the device will broadcast its presence for 3 seconds, followed by another random 1 to 3 seconds of listening. The device will then place its Bluetooth radio in a low power non-listening sleep state, and wait until the next epoch to repeat the cycle. Because transmitting is the dominant consumer of radio power, we chose a broadcast interval which was long enough to cover a reasonable range in the 16-second period, while short enough to be power conscious. The listen intervals are chosen to match the broadcast interval.

The randomized listen intervals provide a crude medium access mechanism, while minimizing the amount of time that the Bluetooth radio must be powered. During the random listen, there is sufficient overlap such that nodes have an opportunity to broadcast their presence as well as detect the broadcasts of other devices. It is possible that a pair of devices might choose the same random intervals (i.e. listen at the same time, and broadcast at the same time, with no overlap), and thus not find each other. However, we expect this to be uncommon.

We experimentally found that under normal user activity, a 16-second period for this protocol achieves approximately 8 to 10 hours of battery life. Though this period means we fall short of catching the "walk-by" window, increasing the frequency resulted in unacceptable sacrifice in battery life. Conversely, decreasing the frequency would likely result in an increase in missed serendipitous contact. Because we believed the achieved battery life would be sufficient, we did not chance reducing the protocol's period length.

Technical issues on platform limitations and power conservation, which led to using such a conservative and carefully managed communication protocol, are discussed in Section 5.2.1.

3.3 User Study

We conducted two separate user studies for our experiment. Each study involved approximately 20 students in total from two separate classes in different departments: Computer Science (CS) and Electrical and Computer Engineering (ECE).

We acknowledge the inherent limitations in the size and selection of our user pool. Twenty participants is not a large number, considering how the students can be anywhere on or off campus. At the very least, they are expected to meet once per week during class times, predisposing them to an a priori pattern. Despite these limitations, these initial user studies helped examine some interesting questions regarding the feasibility of using user mobility for packet transport. Do the users meet more often than just during class time? Is there a bias in which an intermediate node[3] is most successful at delivering packets to a particular destination? Is there robustness in the network or is packet transport reliant on a few nodes? Can this trace data be used to begin formulating better routing decisions that result in more efficient network usage with minimal latency impacts compared to epidemic?

Our first user study involved only graduate students during the autumn of 2003 and lasted for two-and-a-half weeks. Nine students were in a CS graduate course, eight students were in a graduate ECE course, and one student was unrelated to either of those two courses. The second user study involved only undergraduate students during the spring of 2004 and lasted for eight weeks. Ten students were in an undergraduate CS class and ten in an undergraduate ECE class.

In addition, we deployed three stationary devices[4] which were hidden throughout the computer science building. These devices are not base stations and play no special role in the experiment or network. One can think of them as users with very little mobility. The stationary devices were included in the study to help examine the following questions. If we assumed the stationary devices were base stations or stationary people, how often would users pass by

[3]For this study we consider users and Palm devices to by synonymous, and refer to them as *nodes*.
[4]For both user studies, 04, 05, and 09 are stationary Palm m125 devices.

one? Do they play a critical role in distributing packets through the network? For the first user study, the stationary devices were hidden near locations frequented by graduate students. For the second user study, the devices were hidden near undergraduate labs.

Chapter 4

Results & Analysis

In this chapter we explore the characteristics of the data trace gathered from the experiments. The motivation is to simulate a network built on top of the trace data. In other words, the empirical trace data is the mobility model for the MANET. We start by examining the connectivity and reachability of the network. Then we characterize different routing protocols, to examine the tradeoffs between replication and latency, and future knowledge versus on-line learning.

For exploring networking characteristics, we start with epidemic propagation. Epidemic always finds the least latency paths from source to the destination, but requires unbounded replication. We use the epidemic results as the "gold standard" for comparing other routing methods which relax epidemic's replication and storage assumptions.

From the epidemic results, we then relax the replication assumption and limit to no replication. However, we wish to minimize the potentially increased penalty in latency or successful delivery rate. Static first handoff preference routing uses aggregate future knowledge, utilizing the bias information from the epidemic simulation, in order to establish a fixed routing table.

While the static first handoff preference routing allowed us to relax the replication assumption, it required future knowledge. Thus we explore routing using a gossiping protocol. The gossiping routing protocol is an on-line algorithm, requiring no future knowledge and makes no replication assumptions.

4.1 Connectivity and Reachability

In this section, we examine the direct contact and reachability of the nodes. Reachability refers to a node's ability to send a packet, via some path of intermediary nodes, to a selected destination node. The path traversed by the packet must obey the chronological ordering of node contacts found in the trace data.

4.1.1 Connectivity

Tables 4.1 and 4.2 shows the direct contact (adjacency) for each of the nodes in user studies 1 and 2. For any given node, the table lists the set of all nodes directly contacted over the lifetime of the data trace. Note that for the first user study, nodes 21 and 22 were not deployed due to malfunction. Therefore they are not included in the study and omitted from the analysis.

Due to the small size of our user population, it is not unexpected that some nodes come into contact with all other nodes at least once over the lifetime of the trace. As the user population increases, for example if expanded to all workers in an office building, the number of adjacent nodes is expected to decrease relative to the overall population. In the next section, we explore how adjacency relates to reachability.

4.1.2 Reachability

Tables 4.3 and 4.4 show the number of other nodes reachable from any given node via direct contact or through intermediate nodes. We calculated the reachability by exhaustive and complete search over the lifetime of the trace. The column *original* shows the reachability given the original trace data, with no filtering. In *no class*, we remove all traces which take place 15-minutes before, during, and 15-minutes after class times for each of the nodes. The column *no stationary* shows the reachability with all entries involving the three stationary nodes[1] removed. The combined effect of removing class times and stationary nodes is shown in column

[1] 04, 05, and 09 are stationary nodes

node label	adjacent to
01	03 07 11 16 17
02	03 04 05 07 09 11 14 15 16 17 20 23
03	01 02 05 06 07 11 14 15 16 17 20
04	02 07 11 14 15 16 17
05	02 03 07 11 14 15 17 20 23
06	03 07 08 10 11 12 13 14 15 16 17 18 19
07	01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 20 23
08	06 07 09 10 12 13 15 18 19
09	02 07 08 11 14 15 17 20 23
10	06 07 08 11 12 13 14 18 19
11	01 02 03 04 05 06 07 09 10 12 13 14 15 16 17 18 19 20 23
12	06 07 08 10 11 13 18 19
13	06 07 08 10 11 12 18 19
14	02 03 04 05 06 07 09 10 11 15 16 17 20 23
15	02 03 04 05 06 07 08 09 11 14 16 17 20 23
16	01 02 03 04 06 07 11 14 15 17 20
17	01 02 03 04 05 06 07 09 11 14 15 16
18	06 07 08 10 11 12 13 19
19	06 08 10 11 12 13 18
20	02 03 05 07 09 11 14 15 16 23
23	02 05 07 09 11 14 15 20

Table 4.1: User Study 1, Adjacency - Nodes 04, 05, and 09 are stationary nodes

node label	adjacent to
01	06 10 14 20 22
02	03 04 10 14 18 22
03	02 04 10 12 13 14 22
04	02 03 10 12 13 14 20 22
05	12 14 20 22
06	01 07 11 12 13 16 19 21 22 23
07	06 08 15 16 19 21 22
08	07 11 15 16 19 21 22 23
09	22
10	01 02 03 04 12 13 14 20
11	06 08 15 16 19 22 23
12	03 04 05 06 10 14 18 20
13	03 04 06 10 14 22
14	01 02 03 04 05 10 12 13 18 20
15	07 08 11 16 19 21 22 23
16	06 07 08 11 15 19 21 22 23
17	20
18	02 12 14
19	06 07 08 11 15 16 21 22 23
20	01 04 05 10 12 14 17
21	06 07 08 15 16 19 22 23
22	01 02 03 04 05 06 07 08 09 11 13 15 16 19 21 23
23	06 08 11 15 16 19 21 22

Table 4.2: User Study 2, Adjacency - Nodes 04, 05, and 09 are stationary nodes

node label	original 21 nodes total	no class 21 nodes total	no stationary 18 nodes total	no class, no stationary 18 nodes total	only stationary 21 nodes total
01	20	19	17	16	0
02	20	19	17	16	12
03	20	19	17	16	1
04	20	19	0	0	12
05	20	19	0	0	12
06	20	19	17	16	0
07	19	17	16	14	11
08	20	19	17	16	13
09	20	19	0	0	13
10	20	19	17	16	0
11	20	19	17	16	12
12	20	19	17	16	0
13	20	19	17	16	0
14	20	19	17	16	13
15	20	19	17	16	13
16	20	19	17	16	12
17	20	19	17	16	12
18	20	19	17	16	0
19	20	0	17	0	0
20	19	17	16	14	10
23	19	17	16	14	12

Table 4.3: User Study 1, Reachability - number of other nodes reachable by multi-hop paths. Nodes 04, 05, and 09 are stationary nodes

node label	orig 23 nodes total	no class 23 nodes total	no stationarys 20 nodes total	no class, no stationary 20 nodes total	only stationary 23 nodes total
01	21	21	18	18	0
02	19	19	17	16	7
03	19	19	17	17	4
04	19	19	0	0	9
05	20	20	0	0	9
06	21	21	18	18	0
07	21	21	18	17	0
08	21	20	18	16	0
09	20	20	0	0	9
10	19	19	17	17	8
11	21	21	18	17	0
12	21	19	18	17	7
13	21	19	18	16	8
14	19	19	17	17	8
15	21	20	18	16	0
16	21	21	18	17	0
17	20	20	18	18	0
18	17	16	7	6	0
19	21	21	18	17	0
20	20	20	18	18	9
21	21	21	18	17	0
22	21	21	18	17	9
23	21	21	18	17	0

Table 4.4: User Study 2, Reachability - number of other nodes reachable by multi-hop paths. Nodes 04, 05, and 09 are stationary nodes

no class, no stationary. Finally, *only stationary* shows the reachability for an infrastructure setup, where nodes only communicated with the stationary nodes[1].

The *no class* and *no stationary* columns show that the connectivity is not reliant upon class time or the three hidden nodes. We also examined the effects of independently removing each of the nodes, in turn, from the traces, and found that there is no significant drop in reachability.

These results suggest that nodes have significant contact and reachability between one another, and that the network does not necessarily depend on specific "hub" nodes for connectivity. Instead, there are multiple redundant paths available for reaching any particular node. Moreover, this suggests there is a measurable amount of interaction between the devices, which may be sufficient for networking.

Column *only stationary* of Tables 4.3 and 4.4, compared to column *original*, shows the potential for increased reachability if pair-wise communication is utilized. This suggests that there can be potential networking gains if pair-wise communication is utilized to extend the reach of wireless hot-spots.

Surprisingly, the *only stationary* column for the second user study produced much lower reachability numbers than expected. This might suggest that not as many undergraduates utilized the labs as anticipated, or that there is a significant impact from the power loss problems discussed in Section 5.2.

4.2 Epidemic

In this section we will explore the characteristics of the trace data assuming an epidemic propagation model. We developed a software suite to take our data traces as input, and simulate epidemic propagation across the traces. Thus the *simulator* output provides us with an abstract view of a potential network and its topology. The simulator assumes that all nodes have infinite amounts of memory and have infinite instantaneous bandwidth when radio contact is made.

In our MANET, nodes can experience very long periods of disconnect from other nodes;

with disconnect times ranging up to seven days. To capture this characteristic, we must select a packet generation policy which highlights, rather than obscures, this long disconnect nature of the network. An obvious (and simple) policy might be to generate packets at a fixed rate.

Unfortunately, a constant rate generator has two important drawbacks. First, suppose a node experiences a long period of disconnect, much longer than the packet generation rate. During this interval, many packets will be created, but queued at the node. Suppose the next contact this node makes is with the intended destination, and all of the generated packets get delivered. When we examine the latency characteristics of these packets, we see a linear curve between packet creation time and latency to delivery. Of course, this is clearly not the case – there was only one moment of contact. Thus the continuous generation of packets hid the latency characteristic. Second, suppose a node moves quickly through a group of other nodes, faster than the packet generation rate. During this interval, insufficient packets were generated to capture the changing nature of the network at that moment.

Therefore we must select a packet generation policy that captures the long disconnect and changing characteristics of the network. We do this by having the node generate packets before and after a change in contact. We define change in contact as follows: as the simulator runs over the trace data, it keeps track of who each node last saw. Whenever a node meets another node that is different from the one it last saw, it will generate new packets. Two packets are generated: one just before the moment of contact, and one just after. By covering moments just before and just after contact, we can capture, on average, the latency for packets created at any arbitrary moment. At the same time, the contact detection will ensure nodes generate packets when there are changes in contact, implying potentially useful mobility.

4.2.1 Capacity

Our primary interest in this investigation is the feasibility of forming a MANET using user mobility. In particular we focus our efforts on exploring the characteristics of multi-hop routes.

We consider each transfer, from one node to another, as a hop[1]. Therefore a multi-hop route is any route involving three or more nodes. However, it is quite possible that a group of nodes might be close enough together that their radios have sufficient coverage to form a connected ad-hoc network. For example, a group of three users working at adjacent workstations could form a connected network. In such a scenario, even if packets were delivered from one user to another via a multi-hop path, it does not utilize user mobility and thus is not of interest to our study.

To remove these interactions from the simulator output, we ignore any packets delivered (from any node to any other node) with an end-to-end latency of less than two minutes. This two minute cut-off is an educated guess – we assume that packets which take more than two minutes to get delivered are likely influenced by user mobility.

Because epidemic produces unbounded amounts of replication, the destination node may receive more than one copy of any given packet over the lifetime of the simulation. We refer to the first copy to arrive (i.e. the one with least latency) as the first arrival. Subsequent duplicates are ignored. Figure 4.1 shows the cumulative distribution for all first arrivals in the epidemic simulation of user studies 1 and 2. The square brackets in the legend shows the total number of successfully delivered packets. The median for the first user study is two days 15 hours (3798 minutes), and the median for the second user study is five days 20 hours (8401 minutes).

Multi-hop routes (routes which have two or more hops) accounted for approximately 84% of first-arrivals for the first user study, and approximately 77% for the second user study. A stacked bar graph showing the path length distribution (measured in number of hops) of first arrival packets is shown in Figure 4.2. The large proportion of packets delivered with least latency using multi-hop paths provides motivation for investigating multi-hop routing.

[1]If nodes are vertices and packet transfers between nodes are edges, hop count is equivalent to edge count.

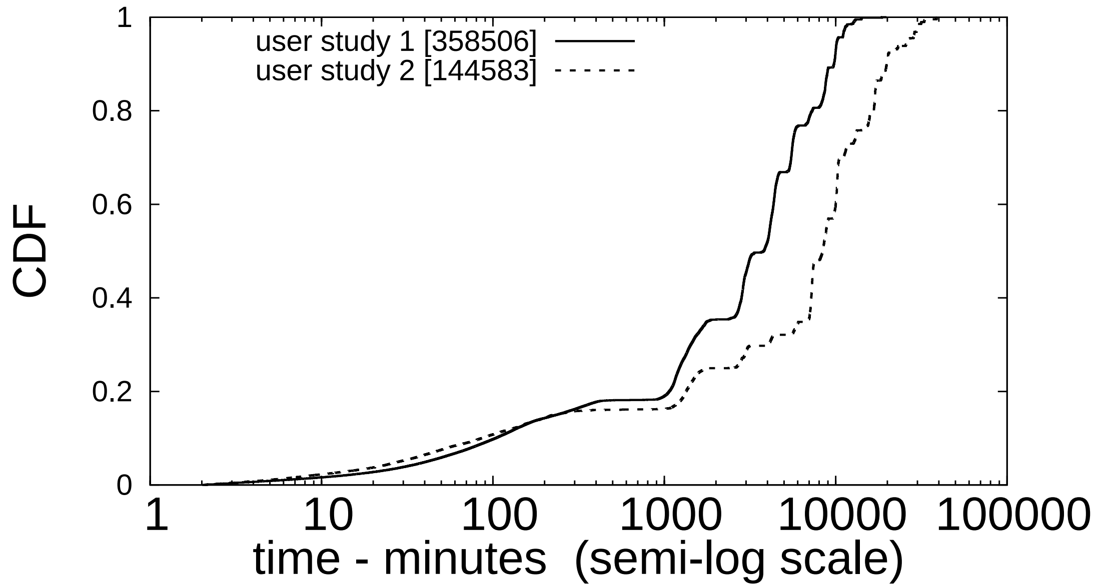
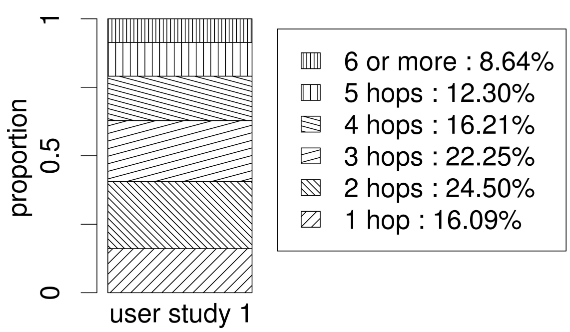
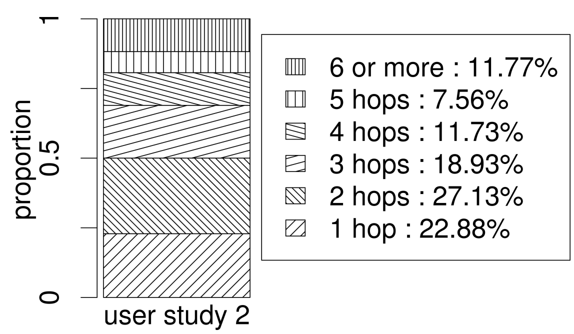


Figure 4.1: CDF of all first-arrivals for user studies 1 and 2



(a) User Study 1 – path length distribution



(b) User Study 2 – path length distribution

Figure 4.2: Path length (hop count) distributions for user studies 1 and 2

4.3 Next Handoff

As an initial step toward finding patterns that would allow us to make better routing and replication decisions, we examine the proportion of first arrival packets delivered from a given source to destination, based on the first intermediary node the source handed the packet to.

Note that a destination node may receive more than one copy of any given packet due to replication in the network. Here we only consider the first arrival of any given packet, ignoring subsequent duplicates. We call the first intermediary node the *first-hand-off node*. We hypothesize if a large proportion of the successful packet deliveries are done by a small number of first-hand-off nodes, then the source node might be able to achieve high rates of successful delivery while minimizing the number of replicas it sends into the network.

We pick six pairs of sender-receiver nodes from user study 1 to examine. From the epidemic output, we selected node pairs which have a high level of communication between each other. Though this is a fuzzy metric for choosing pairs, we believe these pairs reasonably reflect the characteristics of other well-performing pairs.

Three of the pairs make direct contact, but were able to send many packets faster via multi-hop paths instead of waiting for direct contact. Three other pairs never make direct contact, and thus had to send all of their packets via multi-hop paths. The three pairs of nodes that make direct contact are $12 \rightarrow 10$, $15 \rightarrow 02$, and $18 \rightarrow 08$. The pairs that do not make direct contact are $08 \rightarrow 14$, $12 \rightarrow 14$, and $18 \rightarrow 15$.

Tables 4.5 and 4.6 show the proportion of multi-hop first arrivals from the given source to destination, based on the first-handoff node used. The tables show that, for these nodes, a large number of successful least-latency deliveries to a specific destination were done via one or two first-handoff nodes. Note that the percentages do not reflect successful deliveries–they reflect proportion of first-arrivals. It is possible that all of those first-handoff nodes are able to successfully deliver all packets – just with longer latencies.

From these results, we are motivated to explore the efficacy of routing using these preferences determined ahead of time. Thus instead of epidemic, where packets replicate without

first-handoff node	%
18	50.7
08	23.5
11	11.9
13	8.5
06	5.1
19	0.2

(a) 12 to 10

first-handoff node	%
14	62.9
11	14.3
07	10.8
06	6.0
23	3.9
others	2.2

(b) 15 to 02

first-handoff node	%
12	38.4
10	33.6
11	10.9
06	10.7
13	6.2
19	0.2

(c) 18 to 08

Table 4.5: User Study 1, Multi-hop packet delivery proportion by first-hand-off node (pairs that can meet directly. i.e. pairs with adjacency)

first-handoff node	%
15	54.9
10	27.8
18	9.3
12	7.6
13	0.2
06	0.1
09	0.1
07	0.1

(a) 08 to 14

first-handoff node	%
08	46.5
18	16.6
11	14.5
10	13.1
13	6.4
06	2.7
19	0.1
07	0.0

(b) 12 to 14

first-handoff node	%
08	29.1
11	27.5
12	20.6
10	16.2
06	3.4
13	3.0
19	0.2
07	0.0

(c) 18 to 15

Table 4.6: User Study 1, Multi-hop packet delivery proportion by first-hand-off node (pairs that never meet directly. i.e. pairs without adjacency)

bound, we can effectively restrict packet replication, only allowing it to transfer to the set handoff preference.

4.4 Static Preference

In the previous section, we explored the general characteristics of the trace using epidemic propagation. Using aggregate future knowledge provided by the epidemic simulator output, we explored using a static first-handoff preference routing protocol.

To simulate a static preference routing protocol, the simulator must use unicast packets. To address the issue of who creates packets and to whom those packets are addressed, we randomly pair nodes. Pairings are assigned as follows: each node chooses one other node to send packets to. This choice remains fixed for the duration of the simulation. Furthermore, a node cannot be a receiver for more than one other node. Thus, each node sends to exactly one other node, and is a receiver for exactly one other node. It is not necessarily the case that sender/receiver pairings be symmetrical.

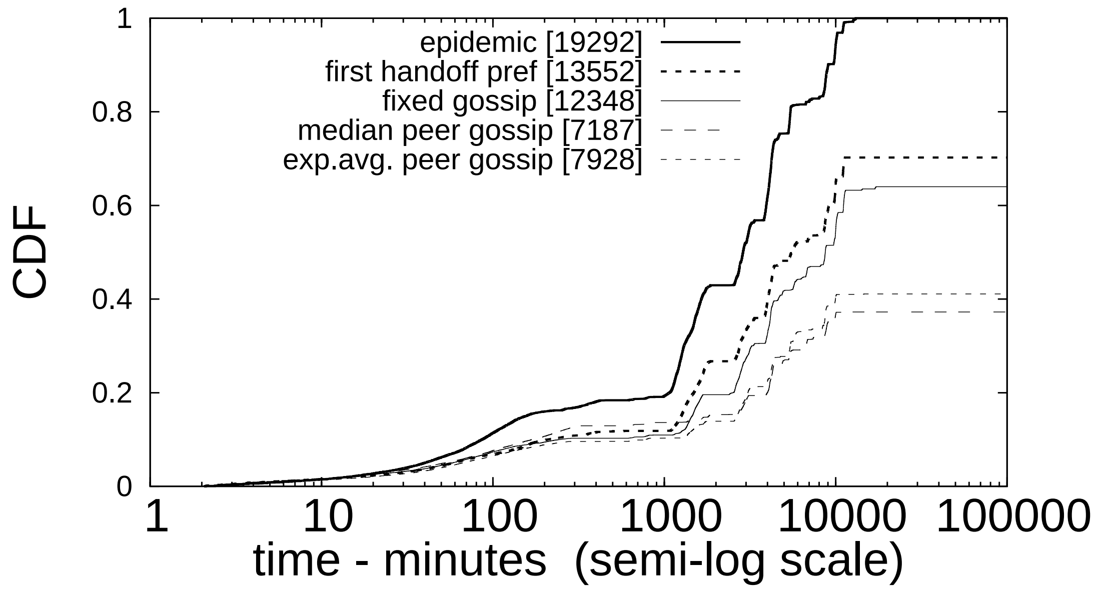
Nodes follow the same packet creation policy as described in Section 4.2. We refer to these unicast packets as *ping* packets. When a receiver receives a ping, it will create another packet to be returned to the ping sender (this is independent of a node's regular packet generating behavior), which we call an *ack*. Ping and ack packets are treated homogeneously; we only use them to find delivery success rate and round-trip times.

Note that because node pairings are random, plots of the resulting characteristics can vary. Some pairings are fortunate and produce low latencies, while other pairings may be more pessimistic. The results shown in this section are for a specific pairing instance which provides a relatively common characteristic; neither optimistic nor pessimistic. Several initial simulation runs were performed with various different pairings, and we found that the general characteristics of the various pairings were quite similar. Though this sampling is far from comprehensive, we believe it supports the belief that the pairings presented in this work captures the general characteristics.

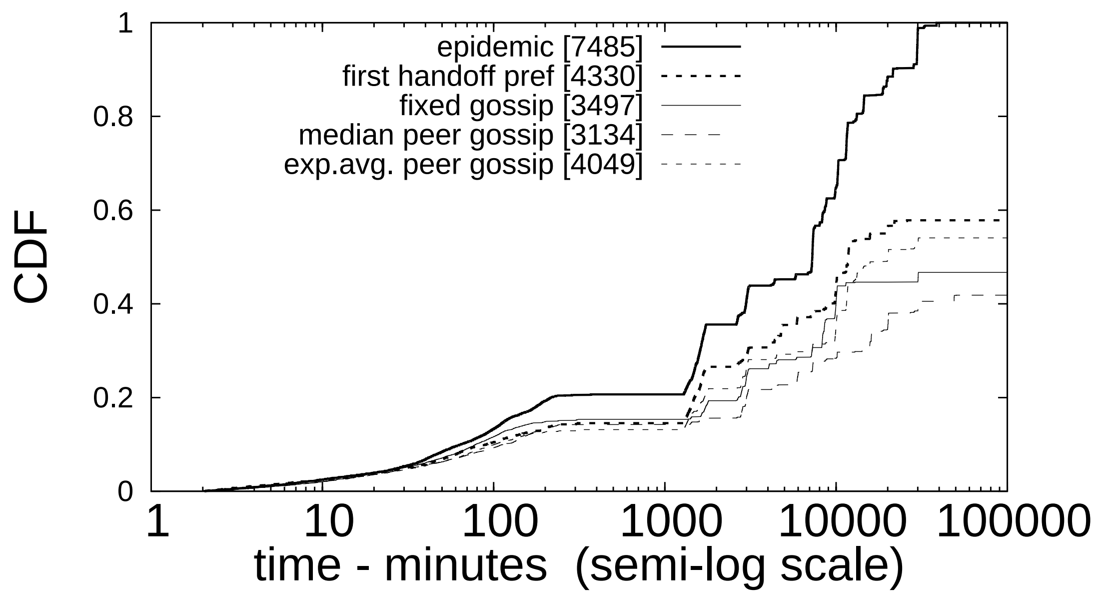
With the first-handoff tables from the previous section, we implement a node routing policy which uses fixed preferences. We use a minimal replication policy, opposite of epidemic, so at any point in time there is only one instance of any given packet in the system. In other words, packets are not replicated. Nodes route packets by looking up a static preference table, indexed on the packet's final destination. Thus this routing strategy is similar to the *Contacts Summary Oracle* in Jain et al. [35].

Figures 4.3 and 4.4 show the packet latency characteristics of the simulation. The one-way time only counts ping packets – counting ack packets would further bias protocols successful at delivering ping packets. The *epidemic* line shows the epidemic results for the selected node pairings. The static preferences simulator is labeled as *first handoff pref*. The other lines, *fixed gossip*, *median peer gossip*, and *exp.avg. peer gossip* are for the gossiping protocols, and will be presented in the next section. The total number of packets successfully delivered is shown between the square brackets in the legend. Because epidemic always produces optimal least-latency and successful delivery rate, all other methods are shown relative to epidemic. Table 4.7 shows the total number of ping and ack packets created over the life of the simulation, including packets which were not successfully delivered. Because ping packets are created based on contact patterns independent of routing protocol, all ping packet creation counts are identical. However, ack packet creation depends on successful ping packet delivery, and hence varies.

We can see that epidemic outperformed the static preference method, which is not surprising considering the significant advantage in replication epidemic enjoys. Because static preferences was less successful at delivering packets (specifically ping packets), it also resulted in fewer ack packets being created. This also contributes to the large disparity between the total count of packets between the two different methods. However, considering that static preference routing delivered 70% as many ping packets as epidemic, with a median latency of four days versus epidemic's two days, using only $1/20^{th}$ as many packets, we believe this routing method fared very well.

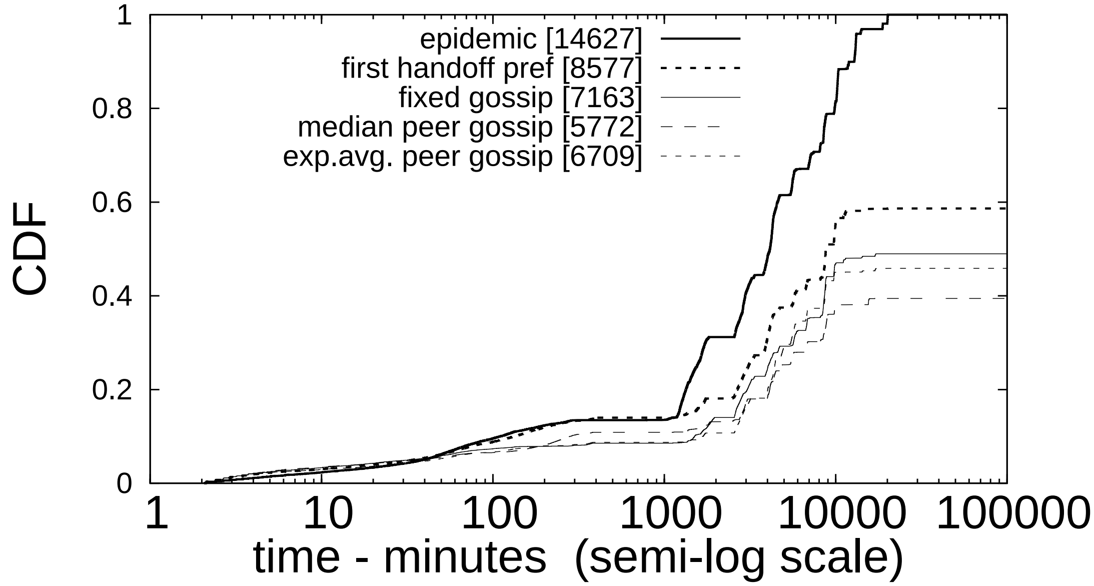


(a) one-way time (ping packets only) - user study 1

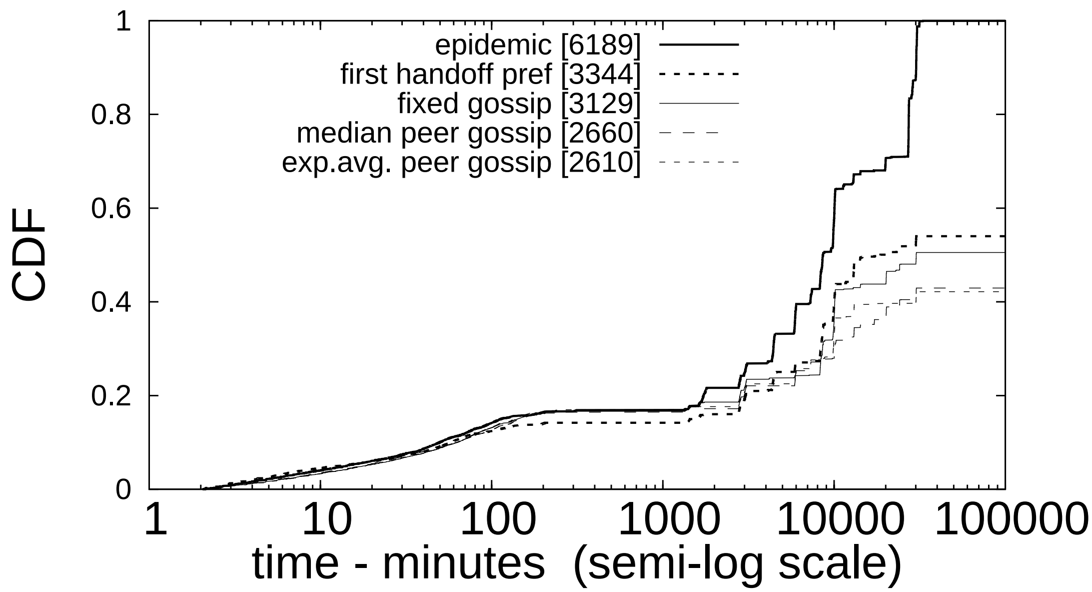


(b) one-way time (ping packets only) - user study 2

Figure 4.3: CDF of one-way packet latency



(a) round-trip time (ping+ack) - user study 1



(b) round-trip time (ping+ack) - user study 2

Figure 4.4: CDF of round-trip packet latency

user study 1	total ping packets created	total ack packets created
epidemic	24228	21429
first handoff pref	24228	15274
median peer gossip	24228	9033
exp.avg. peer gossip	24228	9805
fixed gossip	24228	11467
user study 2	total ping packets created	total ack packets created
epidemic	12062	8116
first handoff pref	12062	4856
median peer gossip	12062	3597
exp.avg. peer gossip	12062	4512
fixed gossip	12062	5379

Table 4.7: Table of total ping and ack packets created. Includes all packets created, both successfully and unsuccessfully delivered.

4.5 Gossip

Unfortunately, static handoff preferences requires aggregate future knowledge, and cannot cope with change. In this section, we will explore the use of gossiping algorithms to learn network patterns and routing paths at runtime. Because of the long disconnected nature of the networks, existing network routing protocols are insufficient. Existing ad hoc protocols, both pro-active (e.g. DSDV[2] [48]) and reactive (e.g. DSR[3] [10]), route for connected subgraphs, and thus fail against a DTN [35]. Extensions can be created to improve these algorithms for DTNs, by tracking time varying connection patterns and stability metrics. However, such extensions and methods for distributing such information are still relatively unexplored.

We explore using a gossiping algorithm for distributing network state beliefs. The belief information includes a cumulative node connectivity graph, with time varying information for each edge. In gossiping algorithms, nodes exchange metadata information which allows them to learn and disseminate knowledge regarding the state of the network. Though there is some overhead in the gossip exchange, this knowledge allows nodes to make more informed decisions for packet routing and responding to network change.

4.5.1 Setup

We wish to examine the effectiveness of using a gossiping protocol to establish suggested routing paths. The times when nodes make contact are constrained by our data trace. Nodes ping and ack in the same manner as described in Section 4.4.

In general, our gossiping protocol works as follows. When nodes make pairwise contact, they exchange gossiping information as part of the communication handshake. This gossip information contains hints and beliefs regarding the state of the network. Nodes then apply this information to forwarding packets.

---

[2]Destination-Sequenced Distance Vector
[3]Dynamic Source Routing

In the following sections we define the terms and data-structures, followed by a brief description of the algorithm. We then show the simulation results of the gossiping protocol, in terms of latency ranking, gossip contents, and gossip convergence rate.

4.5.2 Definitions and Data Structures

First we will describe the data structures kept by nodes for the purpose of performing the pairwise gossiping protocol. We will also define the terms used to describe the algorithm.

For any given packet, we will refer to the originating node as the *source*, the designated receiver as *destination*, and all other nodes that help to transport the packet as *intermediaries*. During contact between any pair of nodes, we will refer to the node that currently holds a given packet as the *sender*, and the node that might potentially receive it as the *receiver*. Sender nodes may also refer to the receiver node as a *handoff*.

Each node maintains a weighted graph of what they believe to be the state of the network. We will refer to this graph as the *Hint Graph*. The hint graph is an undirected graph with vertices representing nodes and edges representing direct contacts between nodes. Edges contain a version number and latency data. Upon pairwise contact, nodes exchange hint graph information and update their beliefs as part of the handshaking process. Nodes then use their hint graph to determine which, if any, packets to exchange.

4.5.3 Algorithm

We will now introduce the algorithm for our gossiping protocol. Sources will only send individual packets once, and intermediaries will only forward packets once. Thus at any time in the network, there will be only one instance of any given packet. For now, we assume there is no packet loss.

When node pairs come into peer contact, they both execute a handshake protocol as illustrated in Figure 4.5. Nodes maintain a hint graph, where each link maintains whatever

```
def HANDSHAKE(NODE self, NODE peer)
    # update our belief of contact with peer
    self.hintgraph.addEdgeIfNotExist(self, peer)
    edge = self.hintgraph[self,peer]
    edge.addContactTime(now())
    edge.version++
    # now for all other hint information
    for each edge in peer.hintgraph do
        self.hintgraph.addEdgeIfNotExist(edge.v1, edge.v2)
        # accept data iff newer version
        if self.hintgraph[edge.v1,edge.v2].version > edge.version then
            # copy version number and expected latency
            self.hintgraph[edge.v1,edge.v2].version = edge.version
            self.hintgraph[edge.v1,edge.v2].latency = edge.latency
```

Figure 4.5: Handshake Algorithm

```
def FORWARDING(self, peer)
    for each packet in queue do
        path = findPath(self, packet.destination)
        # path[0] will be self.
        preference = path[1]
        if preference == peer then # bingo! send it away
            xmitToNode(peer, packet)
```

Figure 4.6: Forwarding Algorithm

timestamp or interval information necessary to infer edge latency. We explore two methods for tracking latency: median latency and weighted exponential averaging. For median latency, a full histogram of all contact intervals must be maintained so that a median value can be selected. For exponential weighted averaging, only the running average value needs to be kept. However, for both median and exponential average methods, peers need only gossip the expected latency value; thus for median latency the entire histogram does not need to be copied.

Nodes are only allowed to update edges for which they are a vertex. Thus nodes can only record what they directly observe, and can only gossip what they have been told. In lines 5 and 6 in Figure 4.5, nodes record their direct observation and increment the version number. The function `addContactTime()` performs whatever computation is necessary to maintain latency information. Then in lines 8 to 14, edge information is copied from the contacted node, but only if the version number is greater. Because edge values can only be updated by direct observation, we never have update conflicts, and only need to gossip version and expected latency values.

Once handshaking has completed, the two nodes will examine their packet queues to determine which, if any, packets to forward to this peer. Figure 4.6 illustrates the forwarding algorithm. For every packet in the queue, the `findPath` algorithm (which is simply Dijkstra's shortest path algorithm) determines what it believes to be the best path for the packet to take. If the current node in contact is on the best path, then the packet is forwarded. Thus, nodes forward to the best handoff node, as determined by their hint graph search result. The hope is that subsequent handoff nodes will have consistent hinting information, so that the packet will efficiently make its way to the destination.

4.5.4 Pairwise Gossip Results

Here we present the results for the gossip protocol based routing. To maintain consistency with previous simulations, we randomly pair nodes as sources and destinations, using the same pairings as before.

To explore the effectiveness of gossip based routing, we use three different hint graph metrics: *median*, *exponential weighted average*, and *fixed*. Median latency hint graphs maintain a full histogram of latency intervals and selects the median value as the edge weight. Exponentially weighted average latency needs only to maintain the current latency value, and adjusts it with each new interval value. Finally, fixed applies a precomputed hint graph using future knowledge, similar to static preference routing in the previous section. However, while static preference routing uses aggregate future knowledge on handoff packet delivery bias, fixed hint graphs use aggregate future knowledge on pairwise contact intervals.

Median and exponentially weighted average hint graphs are “online” algorithms. These hint graphs start with zero knowledge, and must learn the network state as the simulation progresses. We expect the median latency hint graph to be more sensitive to change, while the exponential average latency hint graph to be more stable and adjusts gradually to longer trends. Over the course of the simulation, the hint graphs are free to change as the algorithm deems necessary.

Fixed hint graphs are precomputed using aggregate future knowledge, assigned to all of the nodes at the start of the simulation, and will not change. The fixed hint graph is computed as follows: the data trace is scanned and median and mean values are kept for the contact intervals between all pairings of nodes. At the end of the scan, for each pair, the median and mean latency values are averaged together and used as the edge’s expected latency value connecting that pair of nodes. Using the fixed hint graph allows us to explore two distinct factors for evaluating the effectiveness of the median and average hint graphs: whether "warm-up" time is a limiting factor in the online algorithms, and whether nodes reach sufficient relative consensus regarding network state to route effectively.

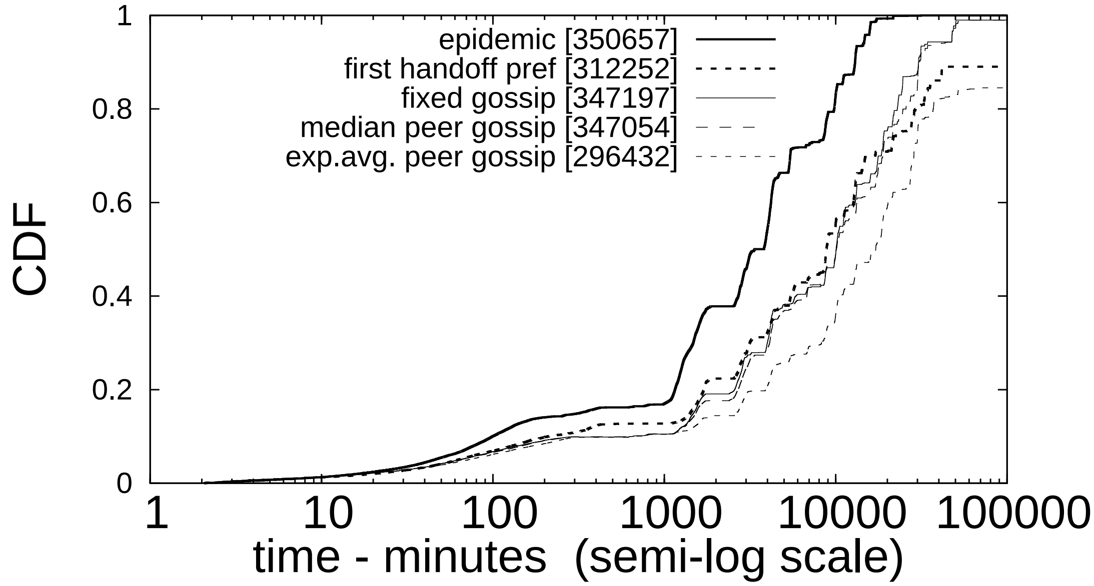
Referring back to figures 4.3 and 4.4, we can see that all of the gossiping methods are relatively comparable to each other. The exponential averaging and median selection hint graphs converge close to the precomputed hint-graph, suggesting that both methods can perform well with extra tweaking. Furthermore, the gossiping algorithms deliver packets with latency curves comparable to first-handoff. This suggests that gossip can in fact learn and take advantage of good paths by relying on past behavior.

4.5.5 Increased Trace Data

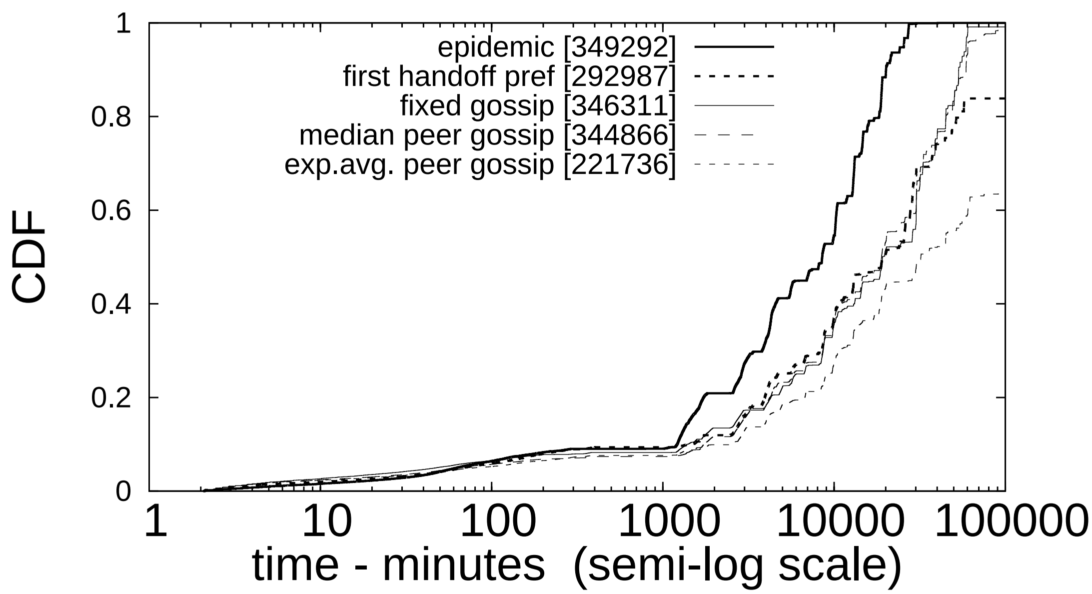
Due to the short length of the trace data, the gossip protocols might not have sufficient time to take advantage of the latency information they have learned. Therefore, we expand our dataset by repeating it sixteen times. Granted this introduces artificial regularity into the resulting trace, but we believe the regularity is a reasonable assumption based on the evaluation of the original data trace.

The trace data is expanded by concatenating multiple copies of the original trace together, with the dates of each subsequent copy adjusted. We hypothesize that users have regular contact patterns based on the time and day of the week. Therefore with every repeat of the data set, we shift the timestamps such that the time and day of the week is preserved.

We will now revisit the various characteristics of the trace, under the repeated dataset. Because we do not introduce new nodes or contact patterns in the repeat copies, connectivity and reachability characters are unchanged. Furthermore, epidemic propagation characteristics are unchanged because the repeats pose no extra advantage or change. Since static first-handoff preferences are set by aggregate future knowledge, we expect it to continue being relatively successful. What we would like to see is whether the gossiping algorithm can improve with



(a) one-way time - user study 1



(b) round-trip time - user study 1

Figure 4.7: CDF of packet one-way and round-trip latency using repeated trace

time, and approach the static preference performance over time (especially after some warm-up period).

Figure 4.7 shows the performance of the different routing algorithms again, over the repeated dataset. Precomputed hint graphs are shown by *fixed gossip*, median latency edge weight hint graphs are shown by *median peer gossip*, and exponential weighted average latency edge hint graphs are shown by *exp.avg. peer gossip*.

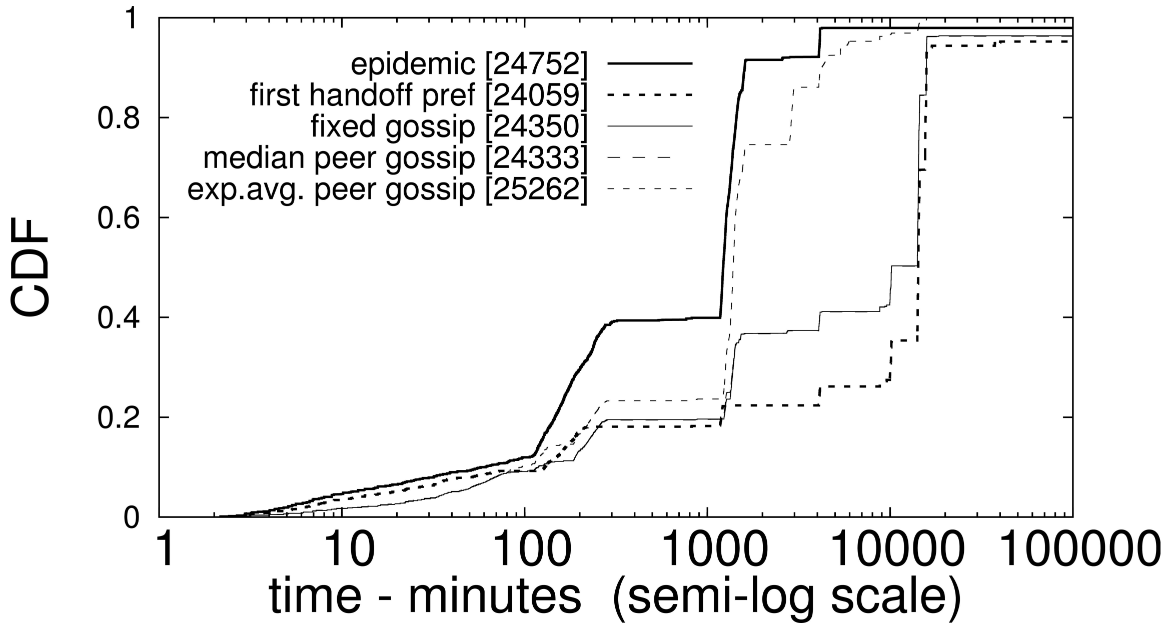
We can see that all of the gossiping algorithms performed with similar success and latency characteristics. The gossiping algorithms have an expected median latency of approximately one week, compared to epidemic's expected median latency of three days. Considering the significantly reduced ($1/20^{th}$) replication assumption compared to epidemic, and no need for future knowledge, we are encouraged that the online gossiping algorithms can be practical. It is difficult to compare the relative strengths of the different routing protocols based on small differences in total packet delivers. In Section 5.2.1, we discuss the limitations of our idealized simulation assumptions.

4.5.6 Results for a Specific Pair

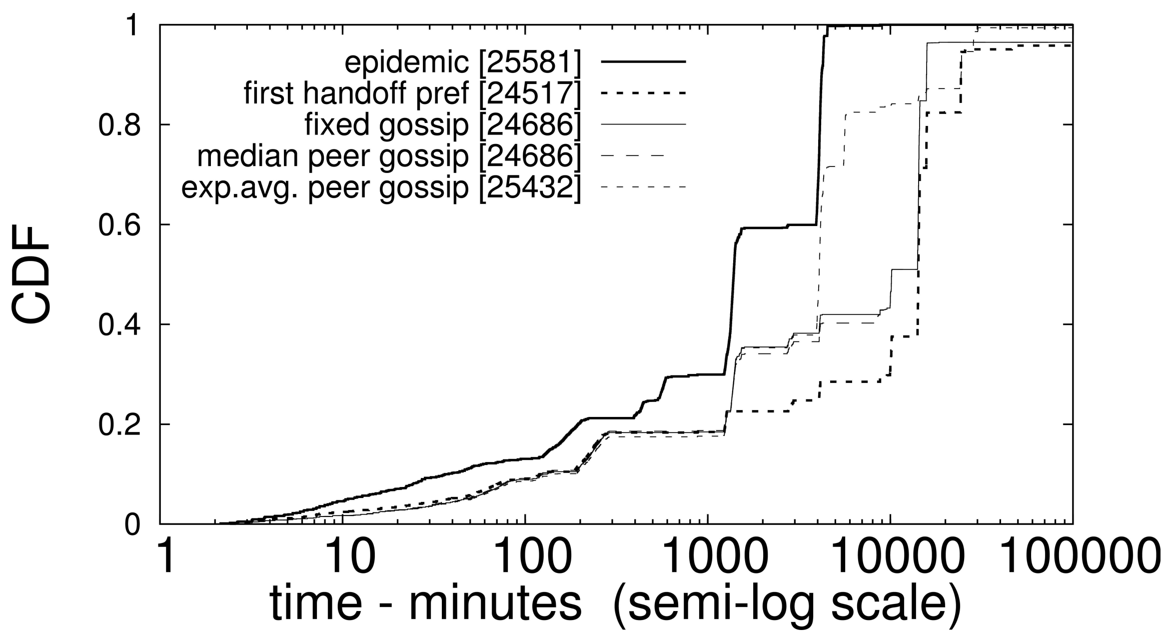
The results shown thus far have been cumulative for all nodes. To get a feel for the characteristics of a specific source to a specific destination, we show the routing results for sending from node 15 to node 02. Recall that this pair was one of the pairings examined in the first handoff analysis in Section 4.3.

Figure 4.8 shows the ping and ack characteristics for packets between nodes 15 and 02, using the repeated trace data. Note that the total number of ping packets as seen in Figure 4.8(a) is often smaller than the total number of ping+ack packets shown in Figure 4.8(b). This is because, as described in Section 4.2.1, we exclude packets which are delivered in less than two minutes from the graphs. It happens that many ping packets were delivered in less than two minutes, but their corresponding ping+ack time was greater than two minutes.

Again we see that aggregate future knowledge helped first handoff preferences deliver a large number of packets, but actually fared poorly in its latency characteristics. The gossiping methods have comparable success in final delivery of packets, but often were able to select better routes to take advantage of local patterns for reduced latency.

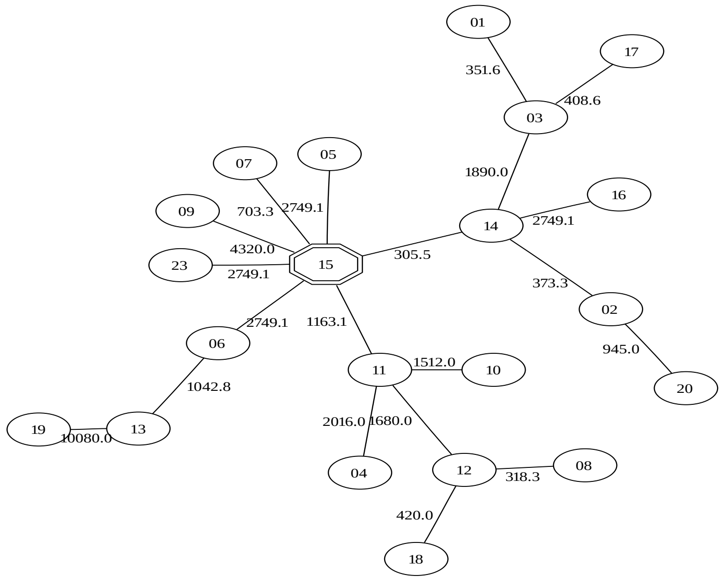


(a) one-way time - $15 \rightarrow 02$ - user study 1

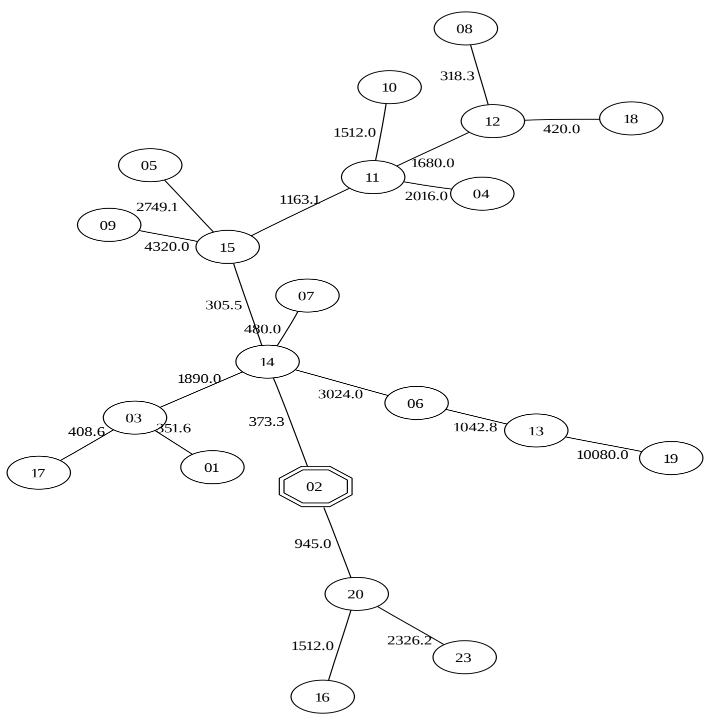


(b) round-trip time - $15 \rightarrow 02$ - user study 1

Figure 4.8: CDF of one-way and round-trip packet latency using repeated trace for pair $15 \rightarrow 02$



(a) Node 15



(b) Node 02

Figure 4.9: Minimum Spanning Hint Graph for Nodes 15 and 02 - edge values are latency time in minutes

user study 1	total ping packets created	total ack packets created
epidemic	387648	384849
first handoff pref	387648	339964
median peer gossip	387648	376412
exp.avg. peer gossip	387648	324199
fixed gossip	387648	235957

Table 4.8: Table of total ping and ack packets created using repeated datatrace. Includes all packets created, both successfully and unsuccessfully delivered.

Just for illustration, we present the minimum cost spanning hint graph for nodes 15 and 02 in Figure 4.9. The actual hint graphs contain all known edges connecting the various known nodes. What is shown is the result of the minimum cost path search, originating at the respective owner node.

4.5.7 Hint Graph Convergence

Finally, we examine the latency for the median and exponential weighted averaging hint graphs to converge. Recall that hint graphs are undirected graphs, containing vertices which represent nodes, and edges which represent direct contacts between nodes. Each edge contains a version number and latency, and nodes are only allowed to update the edge values for which they are a vertex.

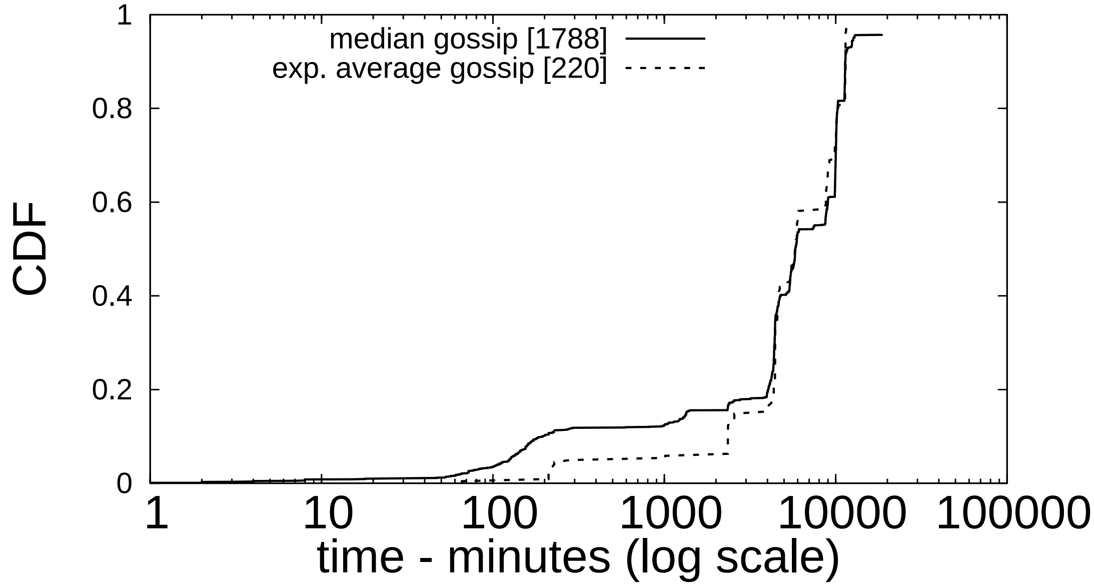
Because the hint graph represents a belief of the state of the network, two hint graphs with different version numbers on edges might still represent the same beliefs of network connectivity and latency estimates. We consider a change in the hint graph which alters the belief of the state of the network to be a *significant change*.

Significant change is determined by the following procedure. When comparing any two hint graphs, there is a significant change if: (1) there is a new edge or vertex that was not previously known, and (2) if an edge has a newer version number AND a different expected latency value. Note that the second conditional thus allows newer version numbers to NOT trigger a significant change, because the newer observations do not shift the currently existing belief about the edge behavior (i.e. the latency is unchanged). Otherwise, if only the version number is checked, then gossip behavior would simply follow epidemic propagation.

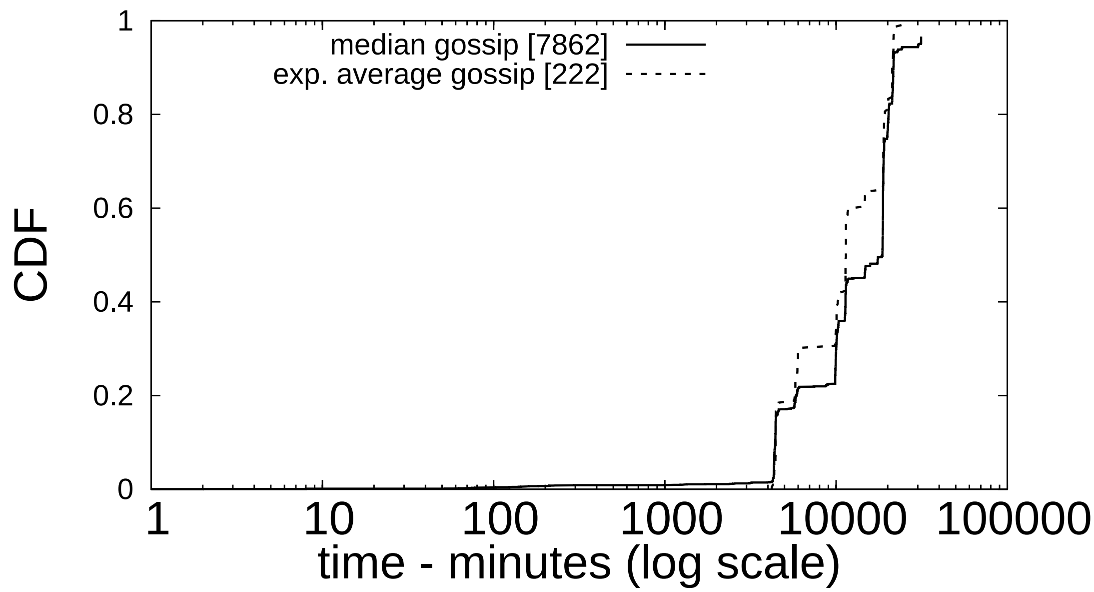
We now describe how hint graph convergence is determined. For each particular node in the trace, we scan through and examine each trace point when the node updates its hint graph via a direct observation. We test the observation and determine whether it caused a significant change, when compared to its previous belief. If no significant change is detected, the entry is skipped and scanning continues. If the direct observation results in a significant change, then this position in the trace is saved, and we begin scanning forward and checking the hint graphs of other nodes. Forward scanning continues until all nodes have learned the direct observation. The time difference from the save point to when the last node learns of the update is a single convergence latency value. We then return to the save point, and repeat the process until all hint graph updates for this node have been examined. The process is then repeated for all other nodes.

Figure 4.10 shows the convergence latency for all nodes, under both the original dataset and for the repeated dataset. The line *median gossip* shows the convergence rate when using histogram and median interval selection for edge latencies, and *exp average gossip* shows the convergence rate when using exponentially weighted averaging over contact intervals for edge latencies. The CDF graphs are scaled to the total number of hint-graph changes detected. Because not all changes could converge before the end of the trace, the plots do not reach 100%. For the first user study, the hint graphs reached 96% and 98% convergence for median and exponential average, respectively.

Because we hold the strict requirement that a change is not considered converged until all nodes have observed the change, it is not surprising that the vast majority of changes require just over a weeks worth of time. However, from a practical standpoint, it is not necessary for a hint to reach all nodes before it becomes useful. For example, should this form of gossiping



(a) original data



(b) repeated data

Figure 4.10: CDF of hint graph convergence latency

be deployed on a city-wide network, it is not necessary for all devices to reach consensus. As long as the hints propagate to members of their social cluster and intermediaries that bridge other clusters, effective routing can be achieved.

We also note that there are fewer hint graph changes under exponential averaging as opposed to median interval selection. Because contact intervals are so varied, for example sometimes between one day and other times between one week, median values tend to "sway" more prominently. Exponential averages tend to be more resilient to these extremes.

Comparing figures 4.10(a) and 4.10(b), we see that exponential averaging is significantly more stable, deciding to trigger only two additional adjustments over the original data trace (220 versus 222). However, the median selection is significantly more sensitive to change, triggering over four times as many change detections as the original trace. However, considering that the trace is repeated 16 times, we can see that some knowledge and stability is retained.

Chapter 5

Discussion

In this chapter we discuss overall factors and considerations in the trace data that was collected. We highlight the limitations of our available data, and our intuition for how we believe the data would scale with a larger sample. Following that we discuss our experiences with the two user studies for collecting pairwise contact data. We present factors which affected our data, and differences between the groups of participants across the two user studies. Finally, we present additional possibilities for exploring gossiping protocols which were not considered for this work.

5.1 Trace Data and Density

Due to the sparse nature of our network from the limited number of participants and potentially large ground area covered, it can be expected that most nodes will require long periods of time to communicate. However, as network density increases, we expect the number of nodes able to find quick multi-hop paths to destination nodes will increase, and the need for high-latency paths will decrease.

As discussed in Section 4.1, the number of direct contacts that a node makes is not necessarily an indicator of its ability to reach other nodes or be an intermediary message carrier. If the number of participants in the network increased, we expect the magnitude of adjacency for nodes to decrease relative to the total number of nodes. However, the data trends suggest that nodes will continue to maintain high reachability. Furthermore, it is not unreasonable to assume that in a real deployment, most nodes will only communicate with a small subset of the total set of nodes in the system. Thus what is more important is the density of nodes and contacts between related clusters of communicating nodes, rather than overall density of the system.

Because this work is a study of empirical trace data, we did not explore artificially increasing the density of the trace by adding more nodes and contact patterns. It is not clear how this can be done while preserving the integrity of the empirical trace. Finding techniques for correlating the trace data with simulations in order to bootstrap more data is a topic of future work.

Another possibility is to simply run more user studies, with more participants over longer periods of time. It would also be beneficial to select a more controlled environment, such as a hospital or elder care facility, instead of random participants. Instrumenting such an experiment is left for future work.

Our analysis of the data trace relies on using idealized packets and infinite bandwidth. This assumption was necessary because the trace data does not contain bandwidth information. As a result, the characterization, which relies on packet counting, can produce exaggerated results. For example, a single moment of contact can result in the delivery of thousands of packets, which might not be possible under less ideal assumptions.

This work has established that there is potential for using opportunistic pairwise contact for ad hoc routing. Future work to further explore the networking possibilities of the trace data include making less ideal assumptions regarding packet size, link bandwidth, and device capacity.

5.2 Experiences

In hindsight, we find that our original estimate of an 8 to 10 hour work-day is insufficient for our user-base. After the first user study with graduate students, we believed our estimate worked well. However, the second user study proved to require even more working battery life. In post-experiment interviews, we found that graduate students kept chargers at their office, and would regularly recharge the devices while at their desk. Thus most graduate students did not fully exercise the eight-hour battery life.

Most undergraduate students cannot recharge their devices mid-day. Indeed, from the onset of the second user study, a significant number of the users could not finish their work-day without draining their devices. Though we established a strict regimen of collecting data on a weekly basis from the students, they often suffered catastrophic data loss from battery exhaustion, losing several days worth of data.

Furthermore, we also found that graduate students were far more conservative with the Palm devices. Few used more than the basic features, and most only carried the devices diligently without much usage. After the first experiment, many participants mentioned that they understood the experimental nature of the software and objective, and treated the device with delicate care.

In contrast, the undergraduate students used the devices quite liberally. Within two weeks of the second user study, we found that most of the participants had downloaded significant numbers of third-party software to use on the Palm devices, including numerous games. Clearly the usage patterns of the undergraduates were significantly more demanding than we had anticipated.

5.2.1 Technical Limitations

The most fundamental and important implementation detail of the experiment is power management on the PalmOS platform. Though there are numerous other oddities of the platform that we work around, power management proves to be the most critical.

Previous research has shown that power consumption on mobile devices is dominated by the radio and display [8]. The fundamental problem with the PalmOS API is that it is designed to be user activity driven, and does not provide interfaces for managing specific resources used in background tasks. Furthermore the API only provides a limited set of common operations available to applications, and reserves full control over resource management. In order for our software to use the Bluetooth radio, it must wake and power the whole Palm device, including the display.

This poses a significant challenge for us when attempting to periodically use the Bluetooth radio. Our software cannot initialize and utilize the radio without first asking the PalmOS to power the rest of the device, including the display. Thus even when the device is in a user's bag, periodically using the Bluetooth radio to search for other devices also means paying for the activation of the display for the duration of the radio communication. Since the Tungsten T devices provide back-lit displays, the power cost is quite significant. We searched for a method to disable the back-lighting while performing the periodic radio communication. Unfortunately, the PalmOS API only provides a method for toggling the back-lighting, but no method for querying the current setting. Again, the API assumes the user will call the toggle through the application until the desired setting is reached.

The limitations of the PalmOS API highlight the need for better hinting mechanisms between the underlying hardware, operating system and applications, as found in Anand et al. [3]. Initiatives such as ACPI [1] have made some progress for notebook computers, but few of these efforts have extended into other mobile devices. Even with current efforts to bridge hardware and software support for power savings, a richer set of hints are needed for efficient operation in a context aware environment. Devices might need to sniff or transmit on their radio device for location specific services and do background processing even when the device is not in active use. Current power management schemes assume devices are in an "on" and functional state, or in a "sleep" and non-functioning state. Supporting hybrid operating modes for varied background tasks, and remaining functional for a sustained work-day, is not yet possible.

The Palm Tungsten T devices keep a reserve of power in the batteries, in case the power levels run too low. Should this happen, the devices refuse to power on, and utilizes the remaining power to maintain memory state. At the start of every communication cycle, our software checks the battery status of the device, and disables further radio usage if the power level runs too low. Though we have the detector set to trigger well before the critical low-battery level, we still experienced numerous total power failures in the second user study. In these cases, we suspect that the reserve amount was insufficient to last the many hours between power failure and when the students finally get home to recharge.

Finally, due to extreme power limitations, we use a clock-synchronized radio protocol for data gathering. To ensure that the clocks do not drift too far, we visit each Palm device at least once per week with a NTP-synchronized laptop to re-synchronize the time.

5.3 Routing and Gossiping: Other Considerations

In this section we discuss various considerations in selecting and implementing gossiping algorithms, and how these considerations relate to potential future work. We start with a brief description of other factors which can be considered for exploring future simulations. Next we compare packet versus peer based gossip, and the relative merits of each method. Finally we discuss possible extensions to the hint graph which might make it more responsive to time factors, node churn, and graph searching.

Relaxing Simulator Assumptions

As mentioned in previous sections, the analysis in this work provides an abstract characterization of the MANET, and motivates future work. To further explore more realistic networking scenarios, incremental increases in replication and retransmission can be implemented. The gossiping protocols explored in this thesis are restricted to no replication. Would an incremental increase of allowing one extra replica significantly increase performance? Furthermore, should more realistic assumptions be made regarding bandwidth, memory, and packet loss, replication and/or retransmits will be necessary to maintain reasonable performance.

Packet Gossip versus Peer Gossip

In this work, we presented an algorithm which used peer gossip. An orthogonal approach would be to use packet gossip. Instead of exchanging gossip information upon pairwise contact, packets can carry gossiping information as part of the metadata payload.

Because peer gossip is packet agnostic, it cannot optimize its decision making for hot paths, flow direction, or common pairs. It is not unreasonable to assume that certain nodes will communicate with only a specific subset of other nodes. In other words, it is unlikely that nodes will decide to randomly communicate with some other node. This affinity of pairing creates hot paths that packet gossiping would be able to take advantage of.

Unfortunately, packet gossiping is influenced by the patterns in which packets are generated and to which senders and receivers they connect. Thus, it can be more difficult to characterize the algorithm with respect to how packet behavior is parameterized. Furthermore, because packet gossip depends upon packet flow, randomization must be introduced to add “jitter” to the network. This occasional injection of entropy is necessary for discovering new and better routes.

Hint Graph Extensions

The implementation presented in this work maintains no temporal ordering information regarding how nodes meet. Thus for example, nodes do not know what day of the week it is, and do not track what day of the week certain intervals appear on. The current implementation is, in effect, at the mercy of circumstance as to whether the chronological order in which nodes make contact will coincide with the path the algorithm chooses. Adding temporal information to the hint graph would give an extra dimension of data, allowing hint graphs to weight their latency expectations depending on the time. This can possibly improve the hint graph's ability to predict potential future contact, and provide improved reduced latency.

In this study, all members of the network are known a priori. However, in a deployed system, new nodes may join and old nodes may leave forever at any moment. Thus a mechanism for penalizing and aging hint graph edges and vertices will be necessary. The simple hint graphs presented in this work never age information, and thus have a limited capacity for dealing with change. Potential extensions might include aging edges and vertices, so that they can be pruned if absent from the network for extended periods of time. Furthermore, penalizing factors might be given to edges if they do not meet expected latency intervals.

Finally, our work implements a naive algorithm which performs path searches for every packet upon demand. Since our work makes idealized assumptions regarding device capacities, this inefficiency does not impact our results. However, real devices will have significant resource constraints, and thus methods for optimizing the hint graph searching must be examined.

Assuming sufficient computing power, one possibility is to have nodes compute path hinting information during long idle disconnect times. Though the decisions made during this disconnect period might not have the latest gossiping information, we assume the network state to be "stable" enough that decisions made during this idle time will still be good. It is likely that many packets in a node's queue will be destined for the same destination node. Multiple path lookups for each of these packets would be a waste of resources. Search result caching and hint graph fingerprinting can be used to determine when new searches are necessary, and reduce computational demand.

Chapter 6

Conclusions and Contributions

We presented an experimental study to test the feasibility of using user mobility and opportunistic pair-wise contact to form an ad-hoc network. Using commodity mobile devices, we instrumented two user studies for experimentally collecting trace data of user contact. Our approach is unique in that we do not have a predetermined model of user mobility, and strive to provide a networking model based only on pair-wise contact.

The results of the experiment are promising, showing that user mobility can potentially be used to form a network. Using this trace data, we simulate an idealized network using epidemic propagation, and observe that nodes exhibit signs of regularity and affinity of contact. From this result we implement simulators to explore the use of aggregate future knowledge for setting static source routing preferences, as well as using gossip to learn network state and route at run-time. Even with a severely limited replication policy for the subsequent routing protocols, they performed comparatively well and with only limited sacrifice in latency (median of seven days) when compared with unlimited epidemic propagation (median of three days). Our results are promising, showing that the gossiping protocols, which are practical protocols with relaxed replication and feature knowledge assumptions, can route effectively across the trace data.

We also describe our experiences developing and deploying instrumented devices to real users. Our experiences show that power management is still an area with much room for improvement, especially for background ambient operation. Many power conservation research projects focus on energy management for devices in use. But power management methods and device peripheral design will need to take into account how devices might be used in two different modes: active use and background operation.

As future work, we plan to instrument another user study, with improved device battery longevity, and collect longer traces. We believe a more focused user group, for example nurses in a hospital or elder care facility. In addition, we plan to correlate the empirical trace data to existing works in characterizing and simulating user mobility, as well as evaluating protocols for ad hoc and delay tolerant networks. Integration of empirical trace data with simulators might allow mobility simulations to produce more detailed and realistic mobility traces, with sufficient detail for exploring networking issues such as bandwidth, power, and congestion.

Bibliography

[1] Advanced configuration & power interface. Advanced Configuration & Power Interface `http://www.acpi.info`.

[2] Micah Adler and Christian Scheideler. Efficient communication strategies for ad-hoc wireless networks. In *Proceedings of the Tenth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 259–268, New York, New York, USA, 1998. ACM Press.

[3] Manish Anand, Edmund B. Nightingale, and Jason Flinn. Ghosts in the machine: Interfaces for better power management. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 23–35, Boston, Massachusetts, USA, 2004. ACM Press.

[4] Luzi Anderegg and Stephan Eidenbenz. Ad hoc-vcg: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 245–259, San Diego, California, USA, 2003. ACM Press.

[5] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371, 2004.

[6] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *IEEE Infocom 2000, volume 2*, pages 775–784. IEEE, March 2000.

[7] N. Bailey. The mathematical theory of infectious diseases and its applications (second edition). Hafner Press, 1975.

[8] K. Barr and K. Asanovic. Energy aware lossless data compression. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, San Francisco, California, USA, may 2003.

[9] Elizabeth M. Belding-Royer. Multi-level hierarchies for scalable ad hoc routing. *Wireless Networks*, 9(5):461–478, 2003.

[10] David B.Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996. edited by Tom Imielinski and Hank Korth.

[11] Luciano Bononi, Marco Conti, and Lorenzo Donatiello. A distributed mechanism for power saving in ieee 802.11 wireless lans. *Mobile Network Applications*, 6(3):211–222, 2001.

[12] Gaetano Borriello, Matthew Chalmers, Anthony LaMarca, and Paddy Nixon. Delivering real-world ubiquitous location systems. *Communications of the ACM*, 48(3):36–41, 2005.

[13] Ioana Burcea, Hans-Arno Jacobsen, Eyal de Lara, Vinod Muthusamy, and Milenko Petrovic. Disconnected operation in publish/subscribe middleware. In *IEEE International Conference on Mobile Data Management*, pages 39–51, Berkeley, California, USA, January 2004. IEEE Computer.

[14] Simon Byers and Dave Kormann. 802.11b access point mapping. *Communications of the ACM*, 46(5):41–46, 2003.

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

[16] Jyh-Cheng Chen, Krishna M. Sivalingam, and Prathima Agrawal. Performance comparison of battery power consumption in wireless multiple access protocols. *Wireless Networks*, 5(6):445–460, 1999.

[17] Xiangchuan Chen and Amy L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Principles of Mobile Computing*, 2001.

[18] Cerritos first us city to offer wireless internet access to all. City Mayors, `http://www.citymayors.com/made4cities/cerritos.html`.

[19] Landon P. Cox and Brian D. Noble. Samsara: Honor among thieves in peer-to-peer storage. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pages 120–132, New York, New York, USA, 2003. ACM Press.

[20] James A. Davis, Andrew H. Fagg, and Brian N. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In *Proceedings of the Fifth International Symposium on Wearable Computers*, pages 141–149, Zurich, Switzerland, October 2001.

[21] Ferro Erina and Potort Francesco. Bluetooth and ieee 802.11 wireless protocols: a survey and a comparison. CMPSCI Tech. Rep. 2003-TR-36, ISTI - CNR (Institute of the National Research Council), Moruzzi 1, 56124 Pisa, Italy, 2003.

[22] Michal Feldman, Christos Papadimitriou, John Chuang, and Ion Stoica. Free-riding and whitewashing in peer-to-peer systems. In *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, pages 228–236, New York, New York, USA, 2004. ACM Press.

[23] J.D. Gibson. The mobile communications handbook. In *The Mobile Communications Handbook*. CRC Press, 1996.

[24] Ashvin Goel. View consistency for optimistic replication. CiteSeer.IST, `http://citeseer.ist.psu.edu/article/goel96view.html`, 1996.

[25] Michel Goemans, Li Erran Li, Vahab S. Mirrokni, and Marina Thottan. Market sharing games applied to content distribution in ad-hoc networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 55–66, Roppongi Hills, Tokyo, Japan, 2004. ACM Press.

[26] W. G. Griswold, P. Shanahan, S. W. Brown, R. Boyer, M. Ratto, R. B. Shapiro, and T. M. Truong. Activecampus - experiments in community-oriented ubiquitous computing. *IEEE Computer*, to appear.

[27] Matthias Grossglauser and David Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proceedings of INFOCOM*, pages 1360–1369, Anchorage, Alaska, April 2001.

[28] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of dht routing geometry on resilience and proximity. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 381–394, New York, New York, USA, 2003. ACM Press.

[29] R. G. Guy, G. J. Popek, and T. W. Page Jr. Consistency algorithms for optimisic replication. In *Proceedings of the First International Conference on Network Protocols*, 1993.

[30] Katrina M. Hanna, Brian Neil Levine, and R. Manmatha. Mobile distributed information retrieval for highly-partitioned networks. In *Proceedings of 11th IEEE International Conference on Network Protocols*, pages 38–49, Atlanta, Georgia, USA, November 2003.

[31] Klaus Herrmann. Modeling the sociological aspects of mobility in ad hoc networks. In *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 128–129. ACM Press, 2003.

[32] Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. *The Sixth International Conference on Ubiquitous Computing*, 2004.

[33] Hristo D. Hristov, Rodolfo Feick, and Walter Grote. Improving indoor signal coverage by use of through-wall passive repeaters. In *Antennas and Propagation Society International Symposium*, pages 158–161, Boston, Massachusetts, USA, July 2001. IEEE.

[34] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, 11(1), February 2003.

[35] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 145–158. ACM Press, 2004.

[36] Mark Jelasity, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. The peer sampling service: Experimental evaluation of unstructured gossip-based implementations. In *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*, pages 79–98, Toronto, Ontario, Canada, 2004. Springer-Verlag New York, Inc.

[37] Jorjeta G. Jetcheva, Yih-Chun Hu, Santashil PalChaudhuri, Amit Kumar Saha, and David B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Proceedings of the Fifth IEEE Workshop on Mobile Computing Systems and Applications*, pages 32–43, Pittsburgh, Pennsylvania, USA, October 2003.

[38] Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.

[39] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, San Jose, California, USA, October 2002.

[40] Aman Kansal, Arun A. Somasundara, David D. Jea, Mani B. Srivastava, and Deborah Estrin. Intelligent fluid infrastructure for embedded networks. In *Proceedings of ACM MobiSYS 2004*, pages 111–123, Boston, Massachusetts, USA, June 2004.

[41] David Kempe, Jon Kleinberg, and Alan Demers. Spatial gossip and resource location protocols. *Journal of the ACM*, 51(6):943–967, 2004.

[42] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, pages 107–118, Atlanta, Georgia, USA, 2002. ACM Press.

[43] Andrés Lagar Cavilla, Gerard Santamaria Baron, Thomas Edward Hart, Lionel Litty, and Eyal de Lara. Simplified simulation models for indoor massachusettsnet evaluation are not robust. In *Proceedings of the First IEEE Conference on Sensor and Ad-Hoc Communications and Networks*, October 2004.

[44] Xiang-Yang Li, Kousha Moaveninejad, and Ophir Frieder. Regional gossip routing for wireless ad hoc networks. *Mobile Network Applications*, 10(1-2):61–77, 2005.

[45] Zongpeng Li, Baochun Li, Dongyan Xu, and Xin Zhou. iflow: Middleware-assisted rendezvous-based information access for mobile ad hoc applications. In *Proceedings of the First International Conference on Mobile Systems, Applications and Services*, pages 71–84, San Francisco, California, USA, May 2003.

[46] Ting Liu, Christopher M. Sadler, Pei Zhang, and Margaret Martonosi. Implementing software on resource-constrained mobile sensors: Experiences with impala and zebranet.

In *Proceedings of the Second Internation Conference on Mobile Systems, Applications and Services (MobiSys 2004)*, pages 256–269, Boston, Massachusetts, USA, June 2004.

[47] Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo. An ad hoc mobility model founded on social network theory. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 20–24. ACM Press, 2004.

[48] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, pages 234–244, New York, New York, USA, 1994. ACM Press.

[49] Pilrc: Pilot resource compiler. Sourceforge.net, `http://sourceforge.net/projects/pilrc`, 2004.

[50] gcc prc-tools. Sourceforge.net, `http://prc-tools.sourceforge.net`, 2003.

[51] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, Boston, Massachusetts, USA, 2000. ACM Press.

[52] Ying Qiu and Peter Marbach. Bandwidth allocation in ad hoc networks: a price-based approach. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 797–807 vol.2. IEEE, 2003.

[53] Vijay Raghunathan, Trevor Pering, Roy Want, Alex Nguyen, and Peter Jensen. Experience with a low power wireless mobile computing platform. In *Proceedings of the 2004 International Symposium on Low Power Electronics and Design*, pages 363–368, Newport Beach, California, USA, 2004. ACM Press.

[54] Rajmohan Rajaraman. Topology control and routing in ad hoc networks: a survey. *Special Interest Group on Algorithms and Computation Theory News*, 33(2):60–73, 2002.

[55] Pablo Rodriguez, Rajiv Chakravorty, Julian Chesterfield, Ian Pratt, and Suman Banerjee. Mar: A commuter router infrastructure for the mobile internet. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 217–230. ACM Press, 2004.

[56] Bill N. Schilit, Anthony LaMarca, Gaetano Borriello, William G. Griswold, David McDonald, Edward Lazowska, Anand Balachandran, Jason Hong, and Vaughn Iverson. Challenge: Ubiquitous location-aware computing and the ”place lab” initiative. In *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, pages 29–35, New York, New York, USA, 2003. ACM Press.

[57] Seattle broadband and telecommunications task force resolution 30684. Seattle Government, `http://www.seattle.gov/btt/res_30684.htm`.

[58] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Datamules: Modelling a three tiered architecture for sparse sensor networks. In *First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, May 2003.

[59] 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 *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 233–244, Annapolis, Maryland, USA, 2003.

[60] Jing Su, Alvin Chin, Anna Popivanova, Ashvin Goel, and Eyal de Lara. User mobility for opportunistic ad-hoc networking. In *Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 41–50, Lake District National Park, United Kingdom, December 2004. IEEE Computer.

[61] Ctia wireless 2005 show report. Tom's Networking, `http://www.tomsnetworking.com/Sections-article108-page3.php`.

[62] Randolph Y. Wang, Sumeet Sobti, Nitin Garg, Elisha Ziskind, Junwen Lai, and Arvind Krishnamurthy. Turning the postal system into a generic digital communication mechanism. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 159–166. ACM Press, 2004.

[63] Weizhao Wang, Xiang-Yang Li, and Yu Wang. Truthful multicast routing in selfish wireless networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 245–259, New York, New York, USA, 2004. ACM Press.

[64] Aaron Weiss. Trends for 2005. *netWorker*, 8(4):20–27, 2004.

[65] Wireless philadelphia. Wireless Philadelphia, `http://www.phila.gov/wireless/index.html`.

[66] Falling costs put indoor coverage in reach. Wireless Review, `http://wirelessreview.com/mag/wireless_falling_costs_put`.

[67] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 187–198. ACM Press, 2004.

[68] IEEE 802.11 Working Group, `http://grouper.ieee.org/groups/802/11/`.