An Empirical Evaluation of the Student-Net Delay Tolerant Network

Jing Su, Ashvin Goel[†], Eyal de Lara
Department of Computer Science
[†]Department of Electrical and Computer Engineering
University of Toronto
http://www.cs.toronto.edu/~{jingsu,delara}, http://www.eecg.toronto.edu/~ashvin

Abstract— **Radio equipped mobile devices have enjoyed tremendous growth in the past few years. We observe that in the near future it might be possible to build a network that routes delay-tolerant packets by harnessing user mobility and the pervasive availability of wireless devices. Such a delay-tolerant network could be used to supplement wireless infrastructure or provide service where none is available. Since mobile devices in a delay-tolerant network forward packets to nearby users, the devices can use short-range radio, which potentially reduces device power consumption and radio contention.**

The design of a user mobility based delay-tolerant network raises two key challenges: determining the connectivity of such a network, and determining the latency characteristics and replication requirements of routing algorithms in such a network. To determine realistic contact patterns, we collected user mobility data by conducting two user studies. We outfitted groups of students with instrumented wireless-enabled PDAs that logged pairwise contacts between study participants over a period of several weeks. Experiments conducted on these traces show that it is possible to form a delay-tolerant network based on human mobility. The network has good connectivity, so that routes exist between almost all study participants via some multi-hop path. Moreover, it is possible to effectively route packets with modest replication.

I. INTRODUCTION

Mobile devices, which enable entire new classes of applications, have enjoyed tremendous growth in the past few years. As the number of radio-equipped mobile devices increases, we observe that it might be possible to build a network that routes delay-tolerant packets based on pairwise contact between users. Such a delay-tolerant network (DTN) [1] could be used to supplement wireless infrastructure or provide service where none is available. For packet delivery, the delay-tolerant network uses two transports: user mobility (no radio needed!) and packet forwarding when users meet. In the latter case, mobile devices only need short-range radio, which potentially reduces device power consumption and radio contention.

Typically, delay-tolerant networks exhibit long periods of disconnection where nodes seldom have end-to-end instantaneous connectivity. As a result, designing effective routing protocols is challenging. In particular, existing routing algorithms for ad hoc networks such as DSR [2] and DSDV [3] assume reasonable connectivity and are thus not well suited for delay-tolerant networks. Furthermore, the disconnected nature of the network results in incomplete routing information, and hence replication may be needed to improve successful packet delivery as well as packet delivery times.

The main goal of this work is to determine whether real user mobility patterns can be used to build a delay-tolerant network. In particular, we wish to address the following key questions: 1) what are the connectivity characteristics of this network, and 2) what are the latency characteristics and replication requirements of the routing algorithms that can be used in such a network.

As a first step towards our goal, we conducted two user studies to collect traces of user mobility. In each study, we outfitted groups of 20 students with Bluetooth-enabled Palm PDA devices. We configured the PDAs to periodically search for other participants and logged all pairwise meetings between users. While this data does not provide precise information about user movement, it captures all opportunities for communication in our network.

We then used the trace data to determine network connectivity and experimented with two types of routing algorithms to evaluate the inherent latency vs. replication trade-off in our network. Our first algorithm uses epidemic propagation [4] to forward packets. While this algorithm can deliver packets with the least latency, it requires making a large number of packet replicas. As a result, we explore link-state algorithms [5], under varying degrees of source-based replication, that use past behavior of contact patterns to determine routing paths.

Our results show that even though our network is sparse it has good connectivity. In particular, while most participants come into direct contact with only a small subset of other participants, they are able to indirectly contact almost all other participants via some multi-hop path. Furthermore, even participants that come into direct contact can generally route packets to each other with lower delay using multi-hop paths. In our network, the median one-way delivery time is approximately three days. While this is not practical, we stress that our population size was very small compared to the area of the city. When comparing epidemic propagation with the link-state algorithms, we find that the link-state algorithms require a small fraction (about 1/10th) of packets compared to epidemic but incur only twice the latency.

This paper makes two contributions. First we show that it is feasible to build delay-tolerant networks based on real traces [6] of human mobility. Second, we characterize the performance of different classes of delay-tolerant routing algorithms in terms of their latency characteristics and replication requirements.

The rest of this paper is organized as follows. The following section describes the experiment we conducted to collect traces of pairwise contacts between users. Section III characterizes the mobility data trace and motivates the need for multi-hop routing strategies. Next, Section IV describes the routing algorithms that we experiment with. Section V describes our results, and Section VI discusses our experiences in conducting mobility studies and some technical limitations in our current study. We cover related work in Section VII, and present our conclusions and avenues for future work in Section VIII.

II. EXPERIMENT

Since we wish to use real user mobility data to evaluate the feasibility of human-based delay-tolerant networks, we conducted an experiment to collect traces of pairwise contact between users [6].[1] Our experiment identified when any two users met, but did not transfer real data or measure connection bandwidth. This approach was easier to implement and provided sufficient data to evaluate our routing algorithms.

To collect traces of pairwise user meetings, we outfitted users with instrumented mobile devices. The instrumented devices had to satisfy three requirements: 1) there needed to be motivation for the user to carry the device as often as possible; 2) the data collection had to work independent of the user's activities; and 3) the device had to operate for at least an eight-hour period, i.e. a work day. Below, we describe how these requirements were met in our experiments.

We provided users with a featureful device to encourage frequent carrying, and implemented our instrumentation software to have minimal impact on usability. Though we could have used specialized devices for our experiment (e.g., motes [7]), we used commodity PDAs (a Palm device with a short-range Bluetooth radio) because it helps highlight our motivation to network consumer mobile devices in interesting ways.

Our aim was to detect opportunistic pairwise contact, even when users might not be aware of it. Contact could take place while at a meeting, waiting for an elevator, or even walking by another participant. Users might not be aware of who may or may not be a participant, and they might not be using their devices during that moment of contact. Nevertheless, it is desirable to record such contact since it presents a communication opportunity. As a result, our instrumentation software ran continuously and invisibly in the background.

We expected that most users would not have an opportunity to recharge their device until the end of the day. So the devices had to operate for at least an eight-hour work-day. Requiring mid-day recharges would be disruptive of the user's routine and increase the likelihood of the device being forgotten or left behind. Unfortunately, our experience showed that it is not possible to meet the eight-hour battery requirement when the Palm devices, even with short-range radio, are continuously powered and transmitting on their radios. Therefore, judiciously managing device activity, and in particular radio transmissions, was essential to achieving our data gathering objectives.

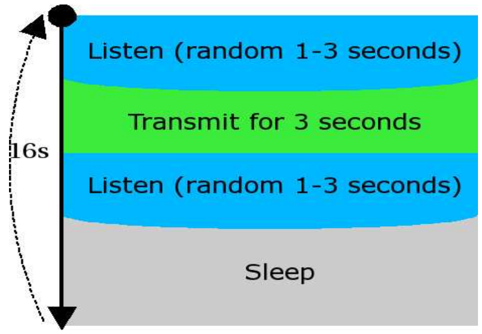


Fig. 1: Radio protocol

The design of our radio protocol was influenced by two main factors: 1) catching opportunistic contact; and 2) ensuring the devices operated for at least eight hours between recharges. 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. Therefore we expect 10-second intervals between device searches to be sufficient for catching most opportunistic contacts.

Unfortunately, a 10-second period consumed too much power and devices fell short of the eight-hour work-day goal. As a result, our protocol searches for peer devices once every 16 seconds. We recognize that the 16 second search interval can miss certain instances of pairwise contact. However, this simply means our data traces are conservative. A shorter sleep time will capture more data, but requires more battery power.

To maximize power conservation under our radio protocol, device radios are active for a short period of time within the 16 second period and sleep the rest of the time. To increase the odds of successful detection we time-synchronized all devices [8] at the start of the user study using Network Time Protocol (NTP).

Our protocol is illustrated in Figure 1. All devices start the protocol cycle at the same time, where they enter into an active radio mode. Within this mode, devices listen for peers as well as transmit their presence for three seconds at a randomized time. Because Bluetooth devices are half-duplex, the randomization provides a necessary form of contention avoidance. Devices then sleep for the rest of the period until the start of the next cycle. Under normal user activity, this radio protocol gave approximately 8 to 10 hours of battery life. To ensure that clock drift does not hinder the protocol's effectiveness during the user study period, we re-synchronized each Palm device at least once per week using NTP.

---

[1] We refer to participants (subjects) in our user studies as "users".

	User Study 1	User Study 2
subjects	21 grad students 3 stationary	23 undergrad students 3 stationary
length of study	2.5 weeks	8 weeks
trace length	30486 tuples	11161 tuples

TABLE I: Summary of User Studies

A. *Data Collection Prototype*

For our experiments, we used Palm Tungsten T PDA devices, running the Palm Operating System (PalmOS). Because sufficient battery life is a major concern, the PocketPC platform, which usually lasts approximately ten hours without radio usage, was not a viable option. Similarly, due to power concerns, we use Bluetooth radio instead of WiFi (WiFi can consume between 10 to 50 times more power than Bluetooth in low-usage modes[2] [9]).

Each Palm device recorded tuples of contact data. At the end of the user study, the logs of each of the Palms were merged together to form a single trace. This trace is a list of tuples of the form: `(timestamp, node id, node id)`.

It should be noted that for the experiment, devices do not track or share user information, and the mapping of devices to users is kept confidential. The trace data for analysis is anonymized before use. At this time we do not consider the security and privacy concerns in such a network.

B. *User Studies*

We conducted two separate user studies. Each study involved approximately 20 students in total from two separate classes in two different departments at the University of Toronto: Computer Science (CS) and Electrical and Computer Engineering (ECE).

The first user study involved only graduate students 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. In addition we hid three stationary devices in several locations to simulate an always available stationary user. The second user study involved only undergraduate students and lasted for eight weeks. Ten students were in an undergraduate CS class and ten in an undergraduate ECE class. Again, three stationary devices were hidden in various locations to simulate stationary users. A summary of the user studies is shown in Table I.

III. INITIAL CHARACTERIZATION

In this section, we present an initial analysis of the data traces we collected. Specifically, we show that routing packets through intermediate nodes improves network connectivity and reduces latency.

A. *Connectivity*

Figure 2 shows the *adjacency* and *reachability* of all nodes in the two user studies over the full length of the traces. Adjacency refers to the number of other peers that a node comes into contact directly. Reachability refers to the number of other peers that a node comes into contact indirectly, via some causal path of intermediate nodes. Along the X-axis is an enumeration of all nodes in each study, sorted by their connectivity. Connectivity (shown on the Y-axis) refers to either adjacency or reachability, i.e. the number of other peers that the node comes into contact directly or indirectly.

We see that multi-hop paths provide a significant increase in connectivity. On the far left of Figure 2(a), the node with the smallest adjacency (it comes into direct contact with 5 other peers), is able to reach almost all other peers (19 other peers) via some multi-hop path. Over half of the nodes in the first user study come into direct contact with less than half of the other peers; and yet reachability for all of the nodes is nearly perfect. We see a similar trend in Figure 2(b) for the second user study, where most nodes are adjacent to less than half of the other peers, and yet are able to reach most or even all peers via multi-hop paths.

Since many participants attend the same lectures, we examine whether class time has an effect on the connectivity of the trace. We remove moments of contact which take place 15 minutes before, during, and 15 minutes after scheduled class times of the participants. The effects are illustrated by the lines labelled “no class” in Figure 2. Though the class time removal shows some loss in adjacency, reachability remains consistently high.

We also examined the degree to which specific nodes were critical in forming the connectivity of the network. The adjacency and reachability of nodes were re-examined multiple times, each time independently removing a node. In all cases network reachability was not significantly affected. This shows that there is robustness in the contacts between nodes, and alternative paths for reachability can often be found. Thus while nodes with high adjacency are beneficial, they are not vital to the connectivity of the network as a whole.

Furthermore we also examined the degree to which the hidden stationary nodes played a role in providing connectivity. We performed the above analysis after removing all contacts involving the three hidden nodes. We found no significant loss from removing these stationary devices. We believe the stationary devices were not effective as intermediaries due to the short radio range of Bluetooth.

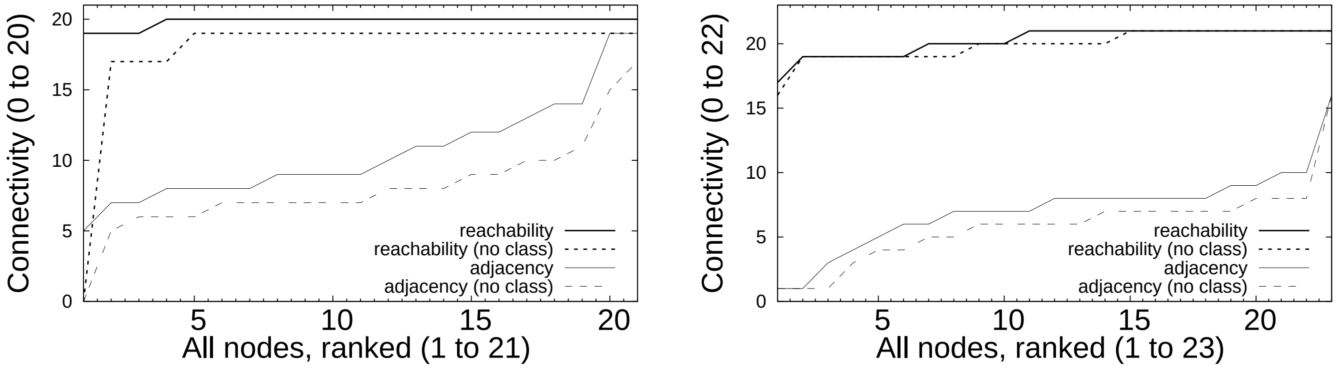
These results are significant in that they show it is possible to create a network based on pairwise meetings and node mobility. While most nodes only meet a small fraction of all nodes in the study, forwarding packets over intermediate nodes enables communication between almost all nodes. Moreover, these results show that there is significant robustness in the network with many alternative multi-hop paths between nodes.

B. *Latency*

To understand whether multi-hop paths can provide lower latency communication between nodes compared to peers meeting directly, we ran the traces through an experiment that implements epidemic propagation. For all of our experiments, we randomly group nodes together as senders and receivers.

---

[2] Low usage is defined as, on average, 90% of time in sleep mode and 10% of the time in receive and transmit modes.



(a) User Study 1

(b) User Study 2

Fig. 2: Adjacency and reachability plots

Every node acts as a sender, and is randomly assigned a specific receiver. No two nodes are assigned the same receiver, and it is not necessarily the case that sender-receiver pairings are bidirectional. Because of the nature of epidemic propagation and almost full connectivity of our network (over time), we expect that multiple replicas of a packet will likely arrive at its destination. We call the first successful delivery of every packet the *first arrival*. We provide a more detailed description of the epidemic propagation in Section IV-A.

Figure 3 shows the distribution of hop counts for all successfully delivered first arrival packets. The graph shows that approximately 81% (100%-18.37%) of packets found lower latency paths to their destination via multiple hop paths. This result shows that not only are multi-hop paths necessary for connecting nodes, but they also provide lower-latency packet delivery.

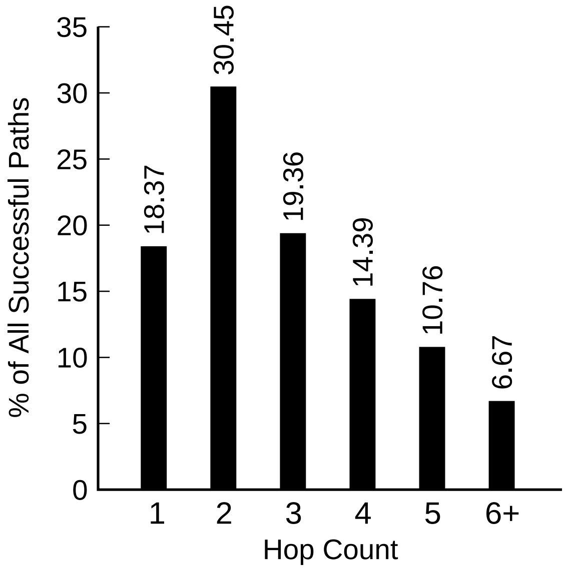


Fig. 3: Hop counts for all first arrivals; graph sums to 100%

IV. ROUTING PROTOCOLS

In this section, we describe the two routing protocols used in our experiments. Our objective is to explore the latency vs. replication trade-off in these routing algorithms. We start with epidemic propagation, which can deliver packets with the least latency but can result in a large number of packet replicas. Next, we explore two variations of a link-state algorithm that use past behavior to determine routing paths at runtime. These algorithms limit packet replication but increase packet delivery latency. We then explore the effect of incrementally increasing source-based replication in our link-state algorithms.

A. Epidemic

In epidemic propagation, every packet transmitted by a source node eventually arrives at every node reachable from the source. When two nodes meet (as indicated by an entry in the trace) they transmit to each other copies of all their packets. Once a node has a packet, it is kept indefinitely and does not receive new copies of it. Epidemic, by its nature, provides the lowest latencies and highest success rate for packet delivery. Since all nodes (including intermediaries) replicate the packet, up to N replicas can be created, where N is the number of nodes in the network.

By itself, epidemic propagation contains no method for packet removal. Techniques exist for removing packets from queues, such as time-to-live restriction and probabilistic delivery detection. However, in the worst case, the entire network must be flooded with any given packet to ensure successful delivery.

In our epidemic experiment, nodes keep packets until a global oracle has determined that a packet has been successfully delivered. Clearly, such an oracle is highly idealized. Even so, our experiment showed that, on average, epidemic created 11 copies per packet in a network with only 21 nodes. The median is 10 copies, with a maximum of 21. This is an intuitive result based on our findings in Section III. Because almost all nodes have a causal path from itself to all other nodes, on average we expect half the network to be flooded for a given source to reach a given destination.

As described in Section II, power management is an important factor for mobile devices, especially with respect to radio use. Each additional replica in the experiment represents additional packets to be transmitted over radio. Thus there is an inherent relationship between replication and increased radio use. Though short range radio technologies can have very high throughput speeds, the primary concern is power consumption. This result motivates exploring routing algorithms that require fewer replicas.

B. Link-State Protocol

Unlike connected ad hoc networks, delay tolerant networks are mostly disconnected and partitioned. In such a network, route querying on-demand would be impractical. Instead, we assume that past user mobility patterns are a good predictor of future patterns. The intuition is that people have regular schedules and meeting patterns, leading to regular pairwise contact patterns.

We explore the use of a link-state based protocol to determine routing paths at runtime in delay-tolerant networks. In a link-state protocol, each node maintains state about the connectivity of the network. Each node shares this state with other nodes that it meets and re-evaluates its own state based on its own observations combined with the state of other nodes. Intuitively, this means nodes track who they meet, and learn with whom other nodes meet.

In our link-state protocol, each node stores a graph of its perceived state of the network, which we refer to as the *link-state graph*. This graph is stored as a table of edges of the form `(node id, node id, weight, version)`. The table, in the worst case, can have $N \times (N-1)$ edges, where N is the number of nodes in the system.

Edge weights in the link-state graph provide an estimate of delay between pairwise contacts, and are a function of the inter-contact time intervals. For example, the edge may be assigned the average time interval between pairwise contacts.

We explore two methods for maintaining edge weights: median latencies (which we refer to as median weighting), and exponentially weighted latencies (which we refer to as average weighting). With median weighting, each edge entry in a link-state graph maintains an unbounded array of contact intervals. When computing weights or exchanging link-state graph entries, the median value is selected from this array, and used as the weight. The contents of the unbounded array are never shared. With average weighting, edge weights are updated using the formula: $weight_{new} = (1-\alpha) \times I + \alpha \times weight_{old}$, where I is the time interval since last contact with the given peer and α is the weighting parameter. We choose a large value of α (0.9) to give weight to the time interval history.

Over time, nodes will make updates to their link-state graph table and share the updated entries with other nodes. We use a single-writing/multi-reader model where only the owner of an edge entry can update it and increment the version number. Next, we describe our link-state protocol, which consists of three phases, and how the graph structure is used. Note that the link-state protocol only runs upon radio contact.

State Update: In this mode, each node updates its edge entry in the link-state graph for this node pair by adjusting the edge weight and incrementing the entry's version number. The edge weight is derived from the time interval from last contact between this pair, as described earlier.

State Sharing: Next, the pair of nodes will share their link-state graphs with one another. Each node will transmit "newer" graph entries to its peer: entries that are unknown to its peer, or entries that have a higher version number. The recipient node creates new entries or replaces existing entries with newer versions. In this state sharing mode, $N \times (N-1)$ entries may have to be exchanged in the worst case.

State Lookup: Once state sharing has completed, the pair of nodes (still in contact) enter state lookup mode. Each node performs a min-path search over its link-state graph, resulting in a minimum spanning tree from itself to all other known nodes. An example of the resulting spanning tree generated at node 02 in our experiment is shown in Figure 4. For every packet in its queue (which may include packets generated by it as well as handed to it by other nodes), the node pairs communicate to compare the expected latency from themselves to the destination. If the expected latency from the peer is less than from itself, the packet is transmitted to the peer. Otherwise, the packet remains queued.

C. Idealized Link-State Graph

Because our link-state protocol must learn the network state at runtime, there is a considerable "warm-up" cost. We experimented with a link-state graph with preset read-only weights. This idealized link state graph has no warm-up cost and does not suffer from making conflicting decisions due to frequent state changes and transitions, thus providing an upper bound for how well our live algorithms will work in practice. In the idealized link state experiments, we first precomputed the link-state graph structure by walking the full length of the trace. We then provide all nodes with a copy of this data structure before the start of the experiment.

Note that epidemic propagation has no warm-up cost because it always replicates packets upon contact. The link-state protocols have a fixed number of replicas per packet. Thus idealized link-state protocol provides us with a comparison of replication versus latency trade-off.

D. Scalability Considerations

Before continuing, we take a moment to discuss some limiting factors in our link-state protocol. In this work we present a protocol capable of learning new nodes, new paths, and edge weight updates online. However, for simplicity, there is currently no mechanism for eventual removal of old and stale nodes or edges. The link-state graph can be extended to support aging, penalizing, and entry pruning. We leave these extensions for future work.

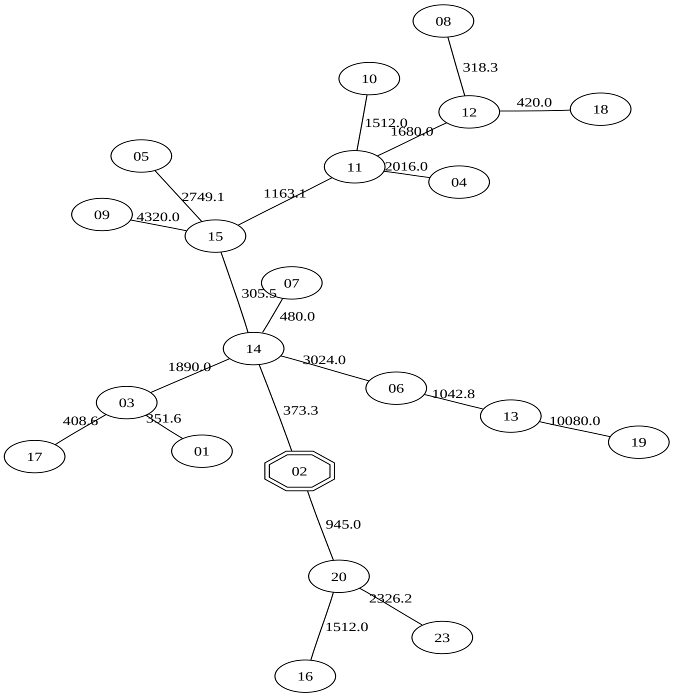


Fig. 4: Resulting min-path spanning tree as calculated by node 02 based on its knowledge of the network state at that moment. Node 02 is indicated by the double-bordered node. Edge weights are expected latencies in minutes.

Our link-state protocol maintains knowledge of all known nodes in the system and diffuses that information in full. For a large set of nodes, the cost of sharing information may become burdensome. There are many options for reducing the size of link-state messages and the link-state graph. Nodes can choose to prune state update messages to specific nodes, depending on expected usefulness of that information. Nodes might take into consideration pruning factors such as the peers' success rate at routing packets, its degree of separation from other nodes, or elapsed time since last contact. Similarly, the link-state graph tables can be reorganized into more hierarchical structures, allowing nodes to determine subsets of data to share, instead of sharing all data.

In a larger or denser device population, a more relaxed sharing model such as a gossipping protocol, where information is shared on a randomized and selective basis can be effective. Because communication opportunities are sparse in our network, we choose to use a strict link-state sharing protocol. We leave the exploration of gossipping protocols in denser networks for future work.

V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the routing algorithms, described in the previous section, in terms of successfully delivered packets and one-way and round-trip latency of packet delivery. As was the case in Section III, we randomly group nodes into sender and receiver pairs. Because certain pairings can result in optimistically good (or pessimistically bad) results, we ran each suite of experiments over five sets of randomized pairings. To obtain comparable results, each suite of experiments used the same pairing and random number seeding. The results presented in this paper combine the five sets of experimental data.

To explore the latency characteristics of our routing algorithms, the packet generation policy must be carefully chosen. The key issue is selecting a generation policy which shows the least and the highest expected latencies. We chose a packet generation policy based on pairwise contact in our data trace since they represent moments of communication opportunity. Whenever a source meets a peer, it creates a packet to its assigned destination (which may not be the peer in contact) immediately before and immediately after the moment of contact. This approach generates the least and the highest expected latencies for every communication opportunity.

To enable us to measure round-trip times, we extended the experiments so that receiver nodes generate an acknowledgement packet (destined for the sender) for every packet they receive. For simplicity, we refer to the original packet sent by a sender as a *ping* and the resulting response packet as an *ack*. Ping and ack packets are handled and routed in the same way, though they may take different paths back. We consider a packet to be successfully delivered if it reaches its destination before the end of the simulation.

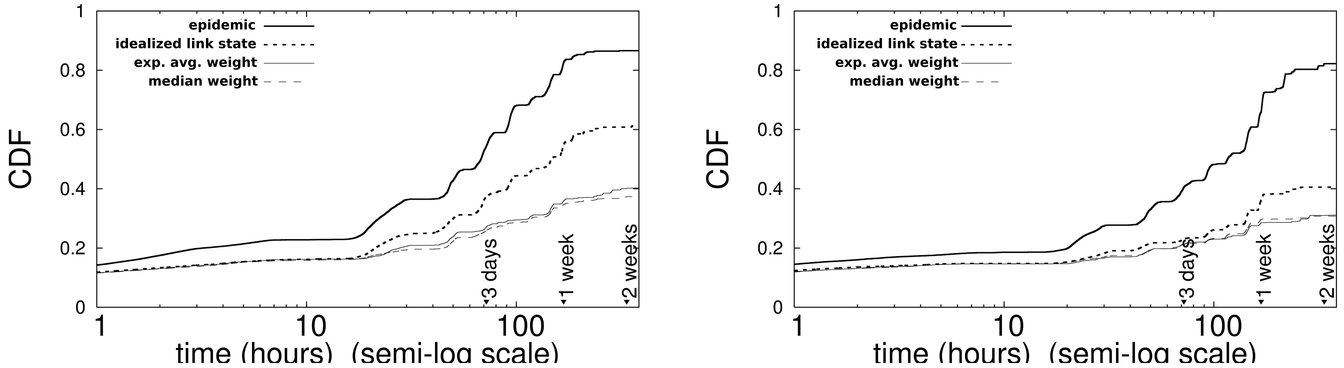
For simplicity, we only present results for our first user study. We found that the results from the two studies show similar characteristics, and the trends found in the first user study are applicable to the second user study.

A. Delivery Rate and Latency

Figure 5 shows a CDF plot of the latencies of first arrival packets for all the routing algorithms. Along the Y-axis is the cumulative proportion of all packets. Along the X-axis is latency, measured in hours on a semi-log scale. The graph shows the comparative latencies of epidemic compared to link state routing using different edge weighting methods. Figure 5(a) shows latencies for ping packets only, and 5(b) shows ping+ack round trips.

The median latency for ping times under epidemic propagation is just under three days and 86% of packets are eventually delivered successfully. The median latency for round trip ping+ack times under epidemic is just under six days, with a successful delivery rate over 82%. Note that at the end of the experiment, many recently generated packets are still "in-flight", counting against the protocol as "undelivered packets". These median ping and ping+ack times (both less than one week) are good, considering the very sparse nature of our network. Recall that our user study subjects are students who have one weekly class in common – outside of class they could be anywhere on or off campus.

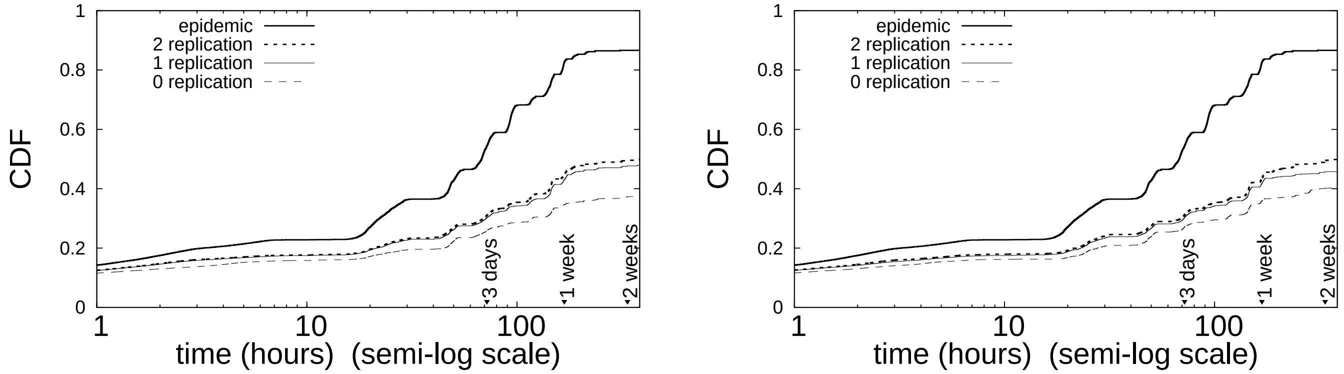
We see in Figure 5(a) that the link-state routing algorithms achieved approximate 40% success rate of packet delivery – almost half of epidemic's success rate. Idealized link state, which did not suffer from a warm-up penalty, performed well



(a) Pings only

(b) Round-trip ping+ack

Fig. 5: Comparative CDF plot of routing algorithms in user study 1



(a) Median Weighted Link State

(b) Average Weighted Link State

Fig. 6: Comparison of 1 and 2 multi-copy impact on median latency weight link state in user study 1

compared to epidemic, achieving over 60% successful packet delivery with a median time of six days. On the upside, the link-state algorithms used, on average, 1/10th of the number of packets compared to epidemic. In the case of idealized link state, this significant reduction in packet replication cost only twice the latency.

B. *Increasing Number of Replicas*

In previous sections, we examined link-state routing strategies using no replication. Now we will study the improvements, if any, in latency reduction that derive from increasing the replication factor. In these experiments, we limit replication to only the source node of the packets; subsequent intermediaries can only forward packets. Without this restriction, replication would eventually result in epidemic-like flooding in the network.

Figure 6 shows the effects of extra replications in the median and average weighting link-state protocols. Idealized link state shows similar characteristics and therefore is not shown. In each of the figures, we show the epidemic protocol, two source replications, one source replication, and no replication for the given routing protocol, indicated by “epidemic”, “2 replication”, “1 replication”, and “0 replication”, respectively.

The figure shows that the first replica has a large impact on improving latency and delivery success for both median and average link-state algorithms. However, additional replication provided limited gains. Allowing replication at intermediary nodes might provide more opportunities for improving latency. However, determining when an intermediate node should replicate is subject of future work.

C. *Increasing Trace Length*

Our link state based algorithms rely on past behavior in order to establish routing decisions. Unfortunately, our two-week long trace provides insufficient time for the algorithms to recoup their warm-up cost. To explore how the algorithms might perform over a longer trace, we concatenated our data trace eight times and repeated the experiments for the routing algorithms. Unlike idealized link state, the median and exponentially weighted link-state graphs can suffer from “bouncing” where conflicting decisions are made due to frequent state changes and transitions.

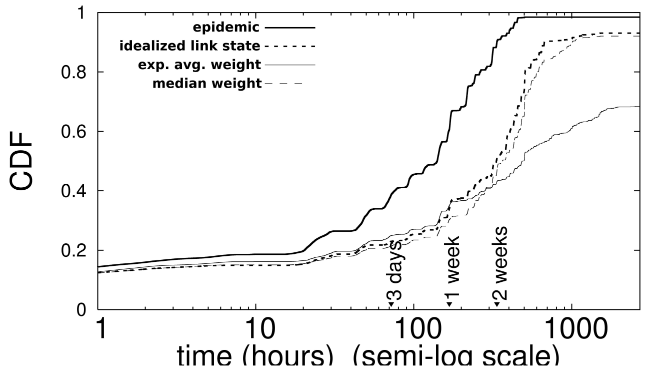


Fig. 7: Comparative CDF plot of round-trip (ping+ack) routing, using repeated trace from user study 1

Our examination of the data trace suggests that many nodes have weekly meeting patterns, even outside of class times. Therefore, for each subsequent concatenation, trace contacts are time-shifted to preserve the day-of-week and time-of-day. We acknowledge that this introduces idealized regularity into the trace. However, our analysis of the data trace shows that some nodes do have regular meeting patterns, even in our sparse trace. The aim of this experiment is to explore what might be possible with more trace data.

Figure 7 shows the ping+ack results for all routing algorithms over the repeated data trace. With the longer data trace, all routing algorithms performed better, with the link state based protocols showing significant improvement.

Under epidemic, the median round trip time in the repeated trace is just over 6 days. Taking advantage of the regularity, the link-state protocols show significant improvement. Using only 1/10th of the packet replication, link-state routing achieves a median latency of just over two weeks.

While these delays are not yet practical, we emphasize the very sparse nature of our network. In most cases packets were successfully delivered to their destination via a multi-hop path faster than waiting for direct contact. We expect latency to decrease with a much denser network.

D. *Discussion*

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.

Despite this idealized assumption our study highlights the inherent trade-off between latency and replication. Replication can improve delivery latency and success but each additional replica represents additional radio usage. As our user study experiment in Section II shows, battery life is a significant limiting factor for mobile devices, and radio use is a significant source of power consumption.

VI. 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. 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, 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 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 more demanding than anticipated.

VII. RELATED WORK

Related work which utilize real mobility of subjects includes ZebraNet [10], [11] and SWIM [12], which used zebras and whales, respectively. However, these works are focused on sensor data collection, and used epidemic propagation [4] for data forwarding. The focus of our work is to determine whether real user mobility can be harnessed for building a delay-tolerant network.

Due to the popularity of wireless networking, many works [13]–[15] have studied traces of wireless access point characteristics, including client movement and packet usage patterns. Chaintreau et al. [16] transformed these WiFi traces to appear as pairwise contact traces for DTN analysis. The transformation assumes that clients which can see an access point can also see each other. However, we believe using WiFi traces is over-optimistic. Due to the large range of WiFi, clients on opposite sides of an access point might not be able to communicate with one another, despite both being able to communicate with the access point.

Chaintreau et al. also provide data traces of pairwise contact collected from a conference. Their analysis shows long-tailed distributions for contact intervals, which suggests that random strangers are not good candidates for efficient forwarding of packets. We hypothesize that physically co-located communities of nodes are better candidates for effective forwarding.

Recent works in forwarding algorithms over DTNs include Spray and Wait [17], where nodes spread a limited number of multiple-copy replicas into the network using heuristics for optimizing distribution. Spray-and-Wait shows promising results under a random walk model [18]. Determining how the algorithm performs under an empirical data trace is a topic of possible future work.

Spropoulos et al. [19] provides a theoretical analysis of several single-copy forwarding strategies under a random walk model. This work introduces a forwarding algorithm based on diffusing contact interval information between nodes as a probabilistic utility function. Our work differs in that we provide an analysis over empirical traces. Furthermore we contend that use of history information, such as our use of a link-state routing algorithm, can provide nodes with more information regarding physical communities. This can subsequently improve routing and route maintenance.

Many related works have examined gossipping protocols for ad hoc routing and resource location [20]–[23]. Li Em AZ. [24] shows that with location information, gossip message forwarding probability can be tailored to an elliptical region, reducing the number of gossip messages by up to 94%. However, these works assume uniform or random placement of immobile ad hoc nodes within a bounded region. Nodes communicate by forming connected ad hoc networks, using radio coverage to transmit from hop to hop. Our work is unique in that we apply a network link state information sharing protocol over data traces of user contact patterns. Our nodes are highly mobile, and connected ad hoc networks rarely form. Thus exploring how a gossipping scheme can be effectively applied to such a mobile network is left for future work.

VIII. CONCLUSIONS

The goal of this work was to determine whether real human mobility patterns can be used to build a delay-tolerant network. To this end, we performed two user studies to collect trace data of pairwise contact between mobile users in a university environment. This data showed that even though our network is sparse, it has good connectivity, and multi-hop forwarding can be used to reduce delivery latencies compared with waiting for nodes to have direct contact.

Nodes in a delay-tolerant network are seldom in contact and do not typically have instantaneous end-to-end connectivity. As a result, traditional routing algorithms for ad hoc networks are not well suited for delay-tolerant network. In addition, routing information in these networks can quickly become stale so replication may be required to improve packet delivery latencies. We study this latency versus replication trade-off by running our traces under epidemic propagation (low-latency, high replication) and link-state routing algorithms (low replication, potentially high latency). Our results show that it is possible to perform single-replica link state based routing in our delay-tolerant network, using only 1/10th of the packets compared to epidemic while incurring only twice the latency (seven vs. three days). While the median latency of the collected traces is measured in days, it is important to take into account the sparse nature of our network. We expect that a deployment with hundreds, or even thousands, of nodes would have much lower latencies.

Finally, we described our experiences in developing and deploying instrumented mobile devices to real users. Our experiences show that power management for consumer devices is still an area with much room for improvement. Currently, mobile devices have either an active or a sleep mode of operation. They would greatly benefit from a third background ambient mode of operation where they sense their network environment.

As future work, we plan to instrument another user study, with improved device battery longevity, and collect longer traces. Further studies will also include bandwidth measurements which will allow a more detailed comparison with other delay-tolerant and Bandwidth data would also allow the evaluation of energy trade-offs in different routing policies. We expect that a more focused user group, for example nurses in a hospital or elder care facility, will provide denser traces of pairwise contacts with improved network connectivity and latencies. Further in the future, we plan to experiment with larger groups of users to determine user communities.

ACKNOWLEDGEMENTS

We would like to extend our thanks to Bell University Labs. Their generous contribution, under grant number 69470, has helped make this research possible. We would also like to thank Stefan Saroiu, Alex Varshavsky, Andres Lagar-Cavilla, and Iqbal Mohomed for their comments in improving this paper. Last but not least, we would like to thank the many anonymous reviewers for their helpful input, and the program committee for their efforts.

REFERENCES

[1] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM Press, 2004, pp. 145–158.

[2] D. B.Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*. Kluwer Academic Publishers, 1996, pp. 153–181, edited by Tom Imielinski and Hank Korth.

[3] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” in *Proceedings of the Conference on Communications Architectures, Protocols and Applications*. New York, New York, USA: ACM Press, 1994, pp. 234–244.

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

[5] J. J. Garcia-Luna-Aceves and M. Spohn, “Transmission-efficient routing in wireless networks using link-state information,” *Mob. Netw. Appl.*, vol. 6, no. 3, pp. 223–238, 2001.

[6] J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara, "User mobility for opportunistic ad hoc networking," in *Sixth IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 2004, pp. 41–50.

[7] "Crossbow technology, inc." http://www.xbow.com. [Online]. Available: http://www.xbow.com/

[8] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.

[9] F. Erina and P. Francesco, "Bluetooth and ieee 802.11 wireless protocols: a survey and a comparison," ISTI - CNR (Institute of the National Research Council), Moruzzi 1, 56124 Pisa, Italy, CMPSCI Tech. Rep. 2003-TR-36, 2003. [Online]. Available: http://dienst.isti.cnr.it

[10] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," in *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, San Jose, California, USA, Oct. 2002.

[11] T. Liu, C. M. Sadler, P. Zhang, and M. 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)*, Boston, Massachusetts, USA, June 2004, pp. 256–269.

[12] T. Small and Z. 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*, Annapolis, Maryland, USA, 2003, pp. 233–244.

[13] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *MobiSYS '03: Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM Press, 2003, pp. 303–316.

[14] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2004, pp. 187–201.

[15] M. McNett and G. Voelker, "Access and mobility of wireless pda users," Computer Science and Engineering, UC San Diego, Tech. Rep., 2004. [Online]. Available: citeseer.ist.psu.edu/662874.html

[16] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding," University of Cambridge Computer Laboratory, CMPSCI Tech. Rep. UCAM-CL-TR-617, 2005. [Online]. Available: http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-617.html

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Multiple-copy routing in intermittently connected mobile networks," Department of Electrical Engineering, University of Southern California, Tech. Rep. CENG-2004-12, 2004.

[18] G. Sharma and R. R. Mazumdar, "Scaling laws for capacity and delay in wireless ad hoc networks with random mobility," in *Proceedings of the IEEE International Conference on Communications*. IEEE, 2004, pp. 3869–3873, Vol.7.

[19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*. IEEE, 2004, pp. 235– 244.

[20] S. Hedetniemi, S. Hedetniemi, and A. Liestman, "A survey of gossiping and broadcasting in communication networks," in *Networks*, 1988.

[21] D. Kempe, J. Kleinberg, and A. Demers, "Spatial gossip and resource location protocols," in *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*. New York, NY, USA: ACM Press, 2001, pp. 163–172.

[22] B. S. Chlebus, L. Gąsieniec, A. Lingas, and A. T. Pagourtzis, "Oblivious gossiping in ad-hoc radio networks," in *DIALM '01: Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications*. New York, NY, USA: ACM Press, 2001, pp. 44–51.

[23] L. Gasieniec and A. Lingas, "On adaptive deterministic gossiping in ad hoc radio networks," in *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002, pp. 689–690.

[24] X.-Y. Li, K. Moaveninejad, and O. Frieder, "Regional gossip routing for wireless ad hoc networks," *Mob. Netw. Appl.*, vol. 10, no. 1-2, pp. 61–77, 2005.