On The Fly Learning of Mobility Profiles for Intelligent Routing in Pocket Switched Networks

Murat Ali Bayir, Murat Demirbas
Department of Computer Science and Engineering
University at Buffalo, SUNY
14260, Buffalo, NY, USA
{mbayir,demirbas}@cse.buffalo.edu

In this paper, we propose a novel routing protocol, PRO, for profile-based routing in pocket switched networks. Differing from previous routing protocols, PRO treats node encounters as periodic patterns and uses them to predict the times of future encounters. Exploiting the regularity of human mobility profiles, PRO achieves fast (low-delivery-latency) and efficient (low-message-overhead) routing in intermittently connected pocket switched networks. PRO is self-learning, completely decentralized, and local to the nodes. Despite being simple, PRO forms a general framework, that can be easily instantiated to solve searching and querying problems in smartphone networks. We validate the performance of PRO with the "Reality Mining" dataset containing 350K hours of celltower connectivity data, and compare its performance with that of previous approaches.

1. Introduction

Cellphone technology has seen an adoption rate faster than any other technology in human history [11]: as of 2009, the number of cellphone subscribers has exceeded 3.3 billion users [1]. The rate of innovation in this field has also been head-spinning. Nokia, Google, Microsoft, and Apple have all introduced cellphone operating systems (Symbian, Android, Windows Mobile, iPhoneOS) and provided APIs for enabling open application development on the cellphones. These modern cellphones, which are dubbed as *smartphones*, enable location-aware services as well as empowering the users to generate and access to multimedia content. As such, smartphones open new opportunities for searching and information retrieval applications. Consider the following scenario.

Scenario: Mike is about to go to lunch with a colleague. He is trying to decide between an on-campus or off-campus lunch location. He finds the Student Union cafes much more convenient than off-campus locations unless there is a student event in the Union that makes conversation impossible. So he uses his smartphone to query the noise level of the Student Union. His query is forwarded hop-by-hop over the smartphones of students on campus, and reaches a smartphone in the Student Union, which answers the query by taking audio-level samples and re-routes the reply back to Mike's phone.

Delay Tolerant Networks (DTNs), which are also known as intermittently connected

[1] www.wirelessintelligence.com

networks, or opportunistic, store-and-forward networks [1, 19, 28, 35, 41] investigate routing techniques that would be of use in the above scenario. In DTNs, nodes are free to move and no centralized network infrastructure exists to provide communication among these mobile nodes. Instead, DTN routing protocols exploit the capability of nodes to perform a peer-to-peer data exchange with other nodes they encounter and strive to achieve data transfer even when the connectivity in the network is intermittent. However, the above described smartphone networks also introduce new challenge for DTN routing protocols. *The nature of human mobility and the structure of social networks* emerge as important factors in smartphone networks, while DTN routing algorithms have been oblivious to them.

Recently Pocket Switched Networks (PSNs) [6, 23, 31, 16, 40] have been formulated as a subfield of DTNs where each node represents a person with a communication device. Several PSN routing protocols have been proposed recently [4, 5, 25, 34]. These work assume some model on human mobility and community-structure, and use this model for making routing decisions. Compared to DTN protocols, PSN protocols make use of more information about the network, and in return aim to find faster paths to the destination with low message overhead (by involving a small number of selected nodes for message forwarding).

In this work, we are motivated by the observation that using smartphones it is possible to maintain more detailed contextual information about the nodes in the network, and hence design faster and more lightweight routing protocols than the existing work on PSNs. More specifically, we propose to employ smartphones to learn the regularity of human mobility profiles. An analysis [2] of MIT's Reality Mining dataset, which is one of the biggest publicly available cellphone connectivity data with 350K hours of celltower connectivity logs [12], shows that significant amount of human mobility (85%) exhibits spatial and temporal regularity where users move between their top-k locations. This behavior implies that the intercontact events between individuals also show spatial and temporal regularity and are predictable. For example, students meet the same specific set of students at particular locations and times for lectures periodically.

Our contributions. In this work, we propose a fast (low-delivery-latency) and efficient (low-message-overhead) routing protocol for PSNs, based on the regularity of human mobility profiles and of intercontact events. Our protocol, namely PRO (profile-based routing protocol), is simple yet general enough to be easily instantiated to solve the smartphone search application scenario we introduced above. In particular the contributions of our paper are as follows.

- In a break from previous routing protocols, our protocol treats node encounters as periodic patterns and exploit them to predict the times of future intercontacts. Our profile-based estimation of intercontacts yields an accurate ranking of the potential forwarding nodes as to their ability to deliver the message earlier to the destination. Our PRO routing protocol uses self-learning nodes, and does not require pre-tuning. The protocol is completely decentralized and local to the nodes.

- We provide an analysis of the effect of forwarding quota at each node and show that forwarding the message to 2 other nodes is the most efficient strategy in terms of communication overhead and delay trade-off. Selecting 2 as the quota improves the latency asymptotically compared to using 1 as the forwarding quota, whereas incrementing the quota to more than 2 leads to diminishing returns. We support our analysis with experimental results in Section 5.

- We give a simple algorithm for making routing decisions. A node selects the highest ranked 2 nodes in its immediate neighborhood and forwards the message to these nodes. Nodes that predict an intercontact with the destination node in the near future (*observed nodes*) have priority over nodes that are unlikely to see the destination node (*non-observed nodes*). Among the observed nodes, nodes that are likely to meet the destination node sooner have more priority. If the current node is unable to fill its forwarding quota with eligible observed nodes, it uses the available quota on non-observed nodes. Among the non-observed nodes, nodes whose profiles differ most from the profile of the current node have more priority. The rationale for this selection is to spread the message to as diverse communities as possible to improve the probability of encountering observed nodes in those communities.

- We validate the performance of our protocol with the "Reality Mining" dataset [12] which is one of the largest publicly available data set containing more than 350K hours of celltower connection and blue tooth connection data. We choose the Reality Mining data set for our validation since it is used as an evaluation batch for several works and it is shown to have similar user behavior with several other data sets which implies the observed phenomenons are not the specific artifact of the data itself [6, 7, 26, 24]. Using the Reality Mining dataset, we compare the performance of our protocol with previous approaches over both cell based location data (coarse granularity) and blue tooth connection data (fine granularity). Our results show that PRO achieves similar success rate and latency (10% less success and 10% more delay time) as the epidemic routing [44] with less than half the communication cost of the epidemic routing. PRO also outperforms the Prophet [35] and Bubble-rap [25] routing protocols (at least 20% less delay time and 25% more success) with less communication cost (at least 25% less communication than these two protocols).

- Using the Reality Mining dataset, we also evaluate the impact of the availability of Internet connectivity on the routing performance. Internet connectivity, when available, acts as a logical link/shortcut and reduces the end-to-end delay significantly.

- Finally, we measure the performance of PRO on smartphone queries described above and show that PRO achieves similar query performance with Epidemic routing (in terms of delay and success) while using significantly less communication cost.

Outline of the paper. In Section 2 we discuss related work on PSNs. In Section 3, we present analytical results for finding the optimum number of forwarding quota. In Section 4, we present our PRO algorithm for profile-based forwarding of messages. Using the Reality Mining dataset, we evaluate the performance of PRO and compare it with previous work on routing in PSNs in Section 5.

2. Related Work

In this section, we categorize and present PSN routing protocols in three broad categories. In each category, we pick a representative popular protocol and discuss it in more detail. Later, in Section 5 we use those three representative protocols to compare and contrast with our protocol.

Flooding-based protocols. In DTNs, replication of the original message is an effective way to increase the probability of successful delivery to the destination. *Epidemic routing* [44] is a representative example of these type of flooding-based routing protocols. In epidemic routing, the messages in the network diffuse like viruses by pairwise contacts between nodes: when two nodes encounter they exchange all of their messages. A node is infected if it accepts a message from another node for forwarding.

The advantage of the epidemic routing is that it has low latency, and it determines a lower limit for the latency of message delivery. On the other hand, too many copies of the initial message increase the overhead drastically in terms of traffic congestion and energy. Several versions of the epidemic routing protocol [21, 47] have been proposed in order to limit the message overhead by imposing constraints such as time limit, maximum hop count, forwarding probability or applying different back-infection techniques to inform nodes about the successful delivery of the message.

Probabilistic model-based protocols. A second category of DTN routing protocols is based on proactive assumptions about node mobility. Random way-point model [30], reference point group mobility model [22], and entity based approaches [29, 14] are examples of this category. These protocols assume/impose a mobility model a priori instead of constructing a model after studying real data.

A representative protocol in this category is *Prophet routing* [35]. The idea behind Prophet is that the probability of message delivery can be calculated by using transitive delivery probabilities. When node i meets node j, the delivery probability of node i for j is updated as $P_{i,j}(k+1) = (1 - P_{i,j}(k)) * P_0 + P_{i,j}(k)$. Here, $P_0 = 0.75$ is the initial probability given as an input to the system. When node i and j do not meet for m periods, the delivery probability is decreased exponentially using an aging factor: $P_{i,j}(k+m) = \alpha^m * P_{i,j}(k)$. Prophet uses the transitive delivery probability when making forwarding decisions. When node i and j meet, i computes the delivery probability to z through j by using the formula: $P_{i,z}(k+1) = (1 - P_{i,z}(k)) * P_{i,j}(k) * P_{j,z}(k) * \beta + P_{i,z}(k)$. Here $\beta = 0.25$ is a parameter denoting the impact of transitivity. i forwards a message for destination z to j, if j has higher delivery probability than i, which holds when $P_{i,z} < P_{j,z}$.

History and social network based protocols. This last category is the one most suited for routing in PSNs. History based approaches [9, 10, 20, 35, 43] depend on the previous observation data in order to predict future interactions. The idea is that if a mobile node has observed another mobile node frequently, the probability of observing the same node is also high in the future. Social network based approaches [5, 8, 25], on the other hand, use social network structure of humans in routing decisions.

Bubble-rap [25] is a representative protocol in this category, as it considers the importance of individuals in social networks for making forwarding decision. Bubble-rap is based on two popularity ranking metrics, called global and local ranking. Global ranking stands for the popularity of the individual in the whole social network calculated as the average number of people the individual observed in recent time slices (e.g., the last six hour time slice). Local ranking is the ranking of each individual in its local community proportional to the average number of people observed in the same community. Forwarding decisions in Bubble-rap are taken by considering these two popularity metrics:

- When two nodes meet, if the sender node is in the same community with the destination of the packet, Buble-rap checks for whether the encountered node is also in the same community, if so the local rankings of sender and potential forwarder are compared; if the encountered node wins, the packet is forwarded.
- If the sender is not in the same community with the destination of the packet, Buble-rap forwards the packet to the encountered node if the encountered node is in the same community with the destination of the packet or if the the global ranking of the encountered node is bigger.

Our PRO routing protocol also falls in this social network based protocols category. Our approach differs from earlier work in this category because it predicts future contact times between nodes using regularity of human behavior and makes forwarding decisions based on this information. In our experiments section, we compare and contrast our protocol with Epidemic routing, Prophet, and Bubble-rap quantitatively in more detail.

3. Analyzing the Impact of Forwarding Quota

3.1. Preliminaries

In this section, we explain basic mathematical models of information dissemination in Mobile Networks. We discuss the derivation of the important parameters and give the fundamental functions for analyzing information dissemination in Mobile Networks. After that, we will use these base functions for analyzing the impact of sender quota on routing performance in Section 3.2. The more detailed discussion about mathematical model of information dissemination by using other constraints except sender quota can be found in [47].

We start with the discussion of dissemination strategies with most expensive case in terms of message overhead which is used in the traditional Epidemic Routing [44]. In this schema every node sends message to the encountered node providing that encountered node didn't received the current message before. Here we provide analysis for Epidemic Routing and its probabilistic version since we use the same idea to analyze the impact of sender quota.

Let N be the size of population and $I(t)$ be the number of mobile nodes (analogy to infected nodes in the epidemic spread) carrying specific message. Let be the pairwise meeting rating of two nodes in the system, which is proportional to speed and transmission range of the nodes over limited area under random mobility model which is exponentially distributed [3, 17]. Under these assumptions the rate for the number of infected nodes can be written as:

$$I^{'}(t) = \beta * I(t) * (N - I(t)) \tag{1}$$

$$I(0) = 1 \tag{2}$$

This equation tells that the infection rate is proportional to the infection condition when one infected node from set of infected nodes (among $I(t)$ elements at time t) encountered with nodes from the set of susceptible nodes (among $(N - I(t))$ elements at time t). Solving ordinary differential equation (1) with initial condition (2) yields:

$$I(t) = \frac{N}{1 + (N-1)e^{-\beta N t}} \tag{3}$$

In [42], a cumulative distribution function is proposed as $P(t)$=Probability($T_{tripTime}$ $< t$) for analyzing average packet delivery time $E[T_{tripTime}]$. It is also stated that for the population size N, the change in the cumulative distribution function $P(t)$ is proportional to:

$$P^{'}(t) = \beta I(t)[1 - p(t)] \tag{4}$$

With initial condition $P(0)$=0, solving ordinary differential equation (4) yields to

$$P(t) = 1 - \frac{N}{N - 1 + e^{\beta N t}} \tag{5}$$

From (5) the expected average packet delivery time in normalized form can be calculated as:

$$E[T_{TripTime}] = \int_{0}^{\infty} (1 - p(t))dt = \frac{\ln N}{\beta(N-1)} \tag{6}$$

By using these equations (3) and (4), the expected number of packets delivered $E[T_{PacketCount}]$, at the time of delivery to destination (excluding packet to destination) is calculated (7) in [47].

$$E[T_{PacketCount}] = \int_{0}^{\infty} I(t)P'(t)dt - 1 = \frac{N-1}{2} \tag{7}$$

In the more general case called Probabilistic forwarding [47], the message forwarding is conditional even the receiver doesn't have the current message. For the simplicity, the approach in [47] assumed that messages are forwarded with respect to constant probability $\rho \in [0, 1]$. In this scenario the probability factor affects the change in the number of infected node as follows:

$$I^{'}(t) = \beta * \rho * I(t) * (N - I(t)) \tag{8}$$

Under this modeling the following equations are obtained for probabilistic routing:

Table 1
Equations for Probabilistic Model

$I(t) = \frac{N}{1+(N-1)e^{-\rho\beta N t}}$
$P(t) = 1 - \left(\frac{N}{N-1+e^{\rho\beta N t}}\right)^{\frac{1}{\rho}}$
$E[T_{TripTime}] = \left[\frac{\ln N}{\beta(N-1)}, \frac{\ln N}{\beta\rho(N-1)}\right]$
$E[T_{PacketCount}] = \frac{\rho(N-1)}{\rho+1}$

3.2. The Impact of Sender Quota on Routing Performance

Previous work on analyzing DTN routing protocols [3, 17, 42, 47] do not focus on analysis of forwarding quota for the performance. In this section, we analyze the forwarding quota, which we define as the maximum number of copies a node can forward to other nodes for any message. In the following discussion, we denote Forward-K as a routing strategy where each message can be forwarded at most K times.

We start with the analysis of the lower-bound for forwarding, namely Forward-1 strategy. Using this strategy, at any time t there exists only a single node in the system that can deliver the message towards another node. In this case, the number of infected nodes and infection rate becomes proportional to the pairwise meeting rate β. For constant population size N, we derive the following expressions for infection rate $I(t)$, and cumulative distribution function $P(t)$:

$$I(t) = \beta t \textbf{ and } I^{'}(t) = \beta \tag{9}$$

$$P(t) = \frac{\beta t}{N} \textbf{ and } P(t)^{'} = \frac{\beta}{N} \tag{10}$$

$P(t)$ stands for the probability of a message to arrive to the destination node before a given time t: $T_{TripTime} < t$. From (2) the expected average packet delivery time in normalized form can be calculated as:

$$E[T_{TripTime}] = \frac{1}{(N/\beta)} \int_0^{\frac{N}{\beta}} (1 - p(t))dt = \frac{1}{2} \tag{11}$$

Since each node can forward at most one packet, the total number of hops traveled by the packet gives the number of nodes that has the message at the time of delivery. Assuming that each condition has equal probability, the $E[T_{PacketCount}]$ becomes:

$$E[T_{PacketCount}] = \frac{1}{N}[1 + 2... + N - 1] = \frac{N-1}{2} \tag{12}$$

Before increasing the forwarding quota to $K \geq 2$, we first give the following definition.

Definition (Saturated Node): At time t, a node is called saturated with respect to message M if it has already forwarded K copies of the current message M.

Lemma 1: For $K \geq 2$ in Forward-K strategies and with infinite population size N, the ratio of saturated nodes in the infected set is always smaller than the ratio of unsaturated nodes in the infected set.

Proof of Lemma 1: The proof is by contradiction. Let $I = A + B$ be the number of infected nodes where A is the number of saturated nodes and B is the number of unsaturated nodes. We assume that $A > B$. We can model the infection process as a directed graph (Figure 1). We know that each infected node has exactly one incoming edge since a node cannot accept copy of same message second time.

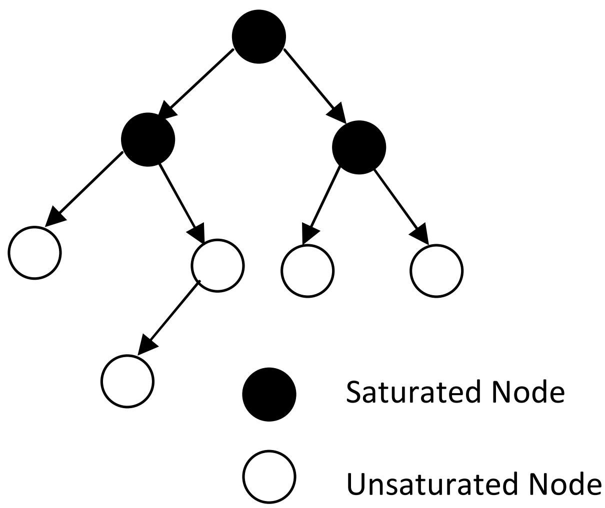


Figure 1. Graph model of infection with K=2

Clearly there exists $I-1$ directed edge in this graph since the number of infected node is $I-1$ excluding the initial node. The maximum value for A corresponds to the condition that all infection edges should come from the edges in the set A. In this case the following equality should be satisfied:

$$I = AK + 1 \tag{13}$$

From I=$A + B$, we get B as the following:

$$B = A(K - 1) + 1 \tag{14}$$
$$B \geq A + 1 \tag{15}$$

Since $K \geq 2$, we get equation (7) which contradicts with the assumption $B < A$.

Next we give a stronger theorem which analyzes the impact of sender quota on delivery time in terms of asymptotical functions.

Theorem 1 (*): For $K \geq 2$, The expected delivery time for Forward-K strategy is $\Theta\left(\frac{\ln N}{\beta(N-1)}\right)$, and is asymptotically $\Theta(N/\ln(N))$ times less than Forward-1 strategy.

Proof of Theorem 1: Since we already showed that the normalized expected trip time of Forward-1 strategy is $E[T_{TripTime}] = \frac{1}{2}$, the remaining part of the proof focuses on showing the expected delivery time for Forward-K strategies for $K \geq 2$. The expected trip time for the probabilistic routing in normalized form [47] is given as:

$$E[T_{TripTime}] \in \left[\frac{\ln N}{\beta(N-1)}, \frac{\ln N}{\beta\rho(N-1)}\right] \tag{16}$$

Lemma 1 shows that for $K \geq 2$, the number of unsaturated nodes in the system is always greater than the number of saturated nodes. If we select a node randomly among infected node set, the probability of selecting unsaturated node is proportional to $\frac{B}{I}$ which is also proportional to message forwarding probability.

From (5) and (6) $\frac{B}{I}$ is equal to:

$$\frac{A(K-1)+1}{AK+1} \tag{17}$$

For $K \geq 2$, the following inequality is always satisfied:

$$\rho = \frac{A(K-1)+1}{AK+1} > \frac{1}{2} \tag{18}$$

If we replace the ρ in equation (8), we get the following range for expected trip time in Forward-K schedule.

$$E[T_{TripTime}] \in \left[\frac{\ln N}{\beta(N-1)}, \frac{2\ln N}{\beta(N-1)}\right] \tag{19}$$

From equation (11), clearly the expected delivery time for Forward-K strategy becomes asymptotically on the order of $\Theta\left(\frac{\ln N}{\beta(N-1)}\right)$ for $K \geq 2$. Obviously this order is asymptotically $\Theta(N/\ln(N))$ times smaller than complexity value for expected trip time of Forward-1.

Theorem 1 shows that selecting forwarding quota K=2 is better than selecting forwarding quota K=1 since it improves the latency asymptotically. Theorem 1 also states that incrementing the quota to more than 2 does not improve the latency asymptotically which leads to diminishing returns. In fact, our experimental results in section 5.3 also supports the results of Theorem 1.

4. PRO: Profile Based Routing for Pocket Switched Networks

4.1. Design Issues

We begin with a discussion of social networks to identify dynamics of human behavior. Small world property [32, 33] is the most fundamental feature of the social networks where the average distances between any two vertices of the network is proportional to the logarithmic scale of the number of vertices N. Recent works [26, 39, 38, 15, 45] showed that the structure of human networks is more complex than small world graphs. These work show that human networks can be modeled as community graphs given in Figure 2. In the community model, a network contains densely connected group of vertices with only sparsely connected vertices between the groups. The neighbor vertices that belong

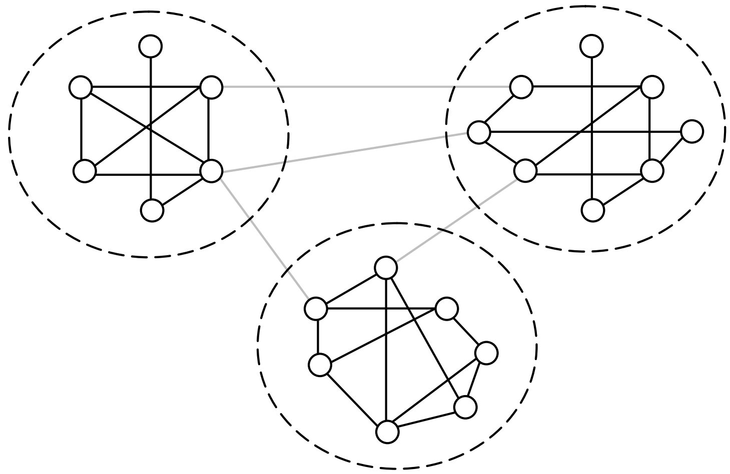


Figure 2. Community structure in human networks

to the same community are called as local neighbors (black edges in Figure 2) and vertices attached to the two sides of edges between different communities are called as remote neighbors (gray edges in Figure 2).

In a very recent work [24] related to Human Mobility Model and community structure, the regularity of inter-contact events in blue tooth level is analyzed. This works showed that inter-contact events between people that knows each other (friend or in the same community) shows regularity in terms of meeting time length and the number of meeting. As it is also analyzed in our previous work [2], we have discovered that the mobility profiles of cell phone users including the spatio temporal mobility patterns shows regularity in days of week and 6 hour length time slices domain. Here, we are going to use the idea that people in the same community (students in the same class, co-workers, same team members) most probably meet regularly in the same set of locations. During these regular meetings possible peer to peer opportunity occurs between smart devices carried by these people. Here we don't focus on regularity of exact peer to peer to communication, to illustrate; two friends may go to fitness centers on Wednesdays between 6.00 pm - 7.00 pm, In these scenario their blue tooth enabled smart phones may connect 6.15 pm in one day and 6.23 pm in another day. Here the important point is that the possible inter-contact time interval (1 hour between 6.00 pm and 7.00 pm on Wednesdays) is regular. This implies that each inter-contact event will possibly occur between 6.00 and 7.00 pm on Wednesdays periodically.

Our Novel Algorithm PRO is distinguished by the way it employs the regularity of intercontact events between nodes in the same community. Although this phenomenon is one of the most important properties of human behavior, it has not been explored fully by previous approaches. History based approaches [10, 20, 35, 43, 9] consider frequent encounters in the near past to predict encounters in the near future. However, the time interval between regular intercontacts does not need to be short, there may be a regularity repeated with longer time intervals. As an example, for two people, history based approaches still produce very high forwarding probability during afternoons even if they encounter only in the mornings and not in the afternoons. The same problem still occurs for routing protocols [5, 8, 25] utilizing social network structure; the high popularity of a node in the social network does not guarantee its high popularity at common time periods such as "mornings in the weekdays".

PRO also employs community structure of Social Networks for fast and light weight routing. To this end, PRO selects the carrier nodes with the maximum information dissemination gain when the current carrier node does not have any local information about destination. The idea here is to cover maximum number of communities when there is no available lead to the destination. But when there are some neighboring nodes that are likely to be in the same community as the destination, PRO gives priority to those nodes.

4.2. PRO Protocol

In this section we present PRO in two parts. In the first part, we explain internal data structures stored in each node. In the second part we present the forwarding algorithm.

4.2.1. Internal Data Structures

In PRO, each mobile node uses internal data structures to keep track of periodic intercontact events with other nodes. Each node reflects intercontact events as updates to observation scores that are stored in the *local observation table*.

Local Observation Table: Each cell in the local observation table corresponds to a periodic time slice in the "week" domain. The justification of this structure follows from [2] which analyzes the Reality Mining dataset. In our design, each cell in the local observation table (Figure 3) stores observation rankings for other nodes which were previously encountered at the time interval corresponding to that cell. Inside each cell, we store a hash table which keeps observation rankings for encountered nodes. Observation ranking is a metric that denotes the probability of observing a node periodically at that time interval. The important point here is that the observation ranking is highly dynamic, the effect of the most recent observations are higher than the effect of the previous observations. For each encountered node X, we use the following iterative functions for updating observation ranking in the corresponding cell.

- $Rank(x)_n = (1-\alpha) * Rank(x)_{n-1} + \alpha * isObserved$, where $\alpha \in (0,1)$, isObserved $\in \{0,1\}$

The observation score k step prior is reflected in the current score with the factor $(1-\alpha)^k$ which goes to zero when k is large, as $\alpha \in (0,1)$. When a node is encountered, the value kept in the hash-table of the corresponding cell is updated with respect to ranking function by using *isObserved*$= 1$. At the end of each day (or the time interval corresponding to each column), the nonobserved nodes for the current column (the ones that already exist in the hash-table inside the cells) is updated with *isObserved*$= 0$.

4.2.2. Forwarding Algorithm

Forwarding algorithm is designed by using two important metrics: observation score and information dissemination score.

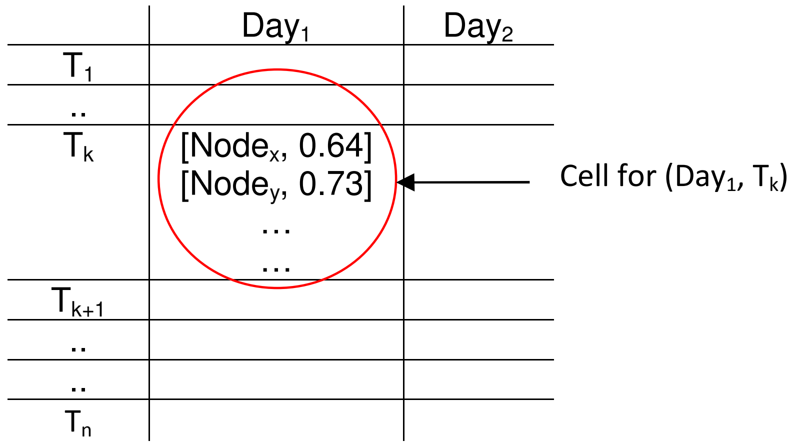


Figure 3. Structure of observation table

Observation Score: Observation score is the metric which is correlated with the probability of observing the destination node in the near future. For a given node A, the observation score of another node B is calculated as follows: If the current slice is X and the slice that corresponds to maximum delay tolerance is X+K, then the observation score of node A with respect to destination node B becomes:

- $OS(B, d) = [1/1]Rank(B)_x + [1/2]Rank(B)_{x+1} + \ldots + [1/(K+1)]Rank(B)_{x+k}$

Clearly the closest time slice X has more effect on the observation score which increases the probability of selecting nodes with earliest delivery times to the destination.

Information Dissemination Score: Information dissemination score measures whether the encountered node is a good candidate for distributing the packet to other nodes. This metric contributes significantly when no information about destination is available (neither current nor encountered nodes have high observation scores). In this case, PRO tries to forward the packet to other communities by using gray links (inter community links) in Figure 2.

In PRO, we use a distributed approach based on the concept of Ego networks [37]; only local topological information of nodes are used for calculating information dissemination score. The idea behind the information dissemination score is that if the potential receiver node observes different set of nodes than the node set that the current node, then that receiver node has higher probability of observing nodes in different communities in the near future. We calculate the information dissemination score between current node A and receiver node B as follows:

- $IDS(A, B) = [1/1]Diff_x + [1/2]Diff_{x+1} + \ldots +$
  $[1/(1+k)]Diff_{x+k}$

In this expression, we use $Diff_x$ as the number of nodes that the receiver node observes differently from the current node in the current time interval x (which is the size of the set $|B \setminus A|$ for time slice x).

Forwarding: For the forwarding process, observation and information dissemination scores are calculated for all of the nodes in the communication range. During the forwarding process, PRO gives priority to the observation score since the nodes that observe the destination regularly are more suitable candidates for forwarding directly to destination. PRO requires the following observation score criteria to hold for forwarding: the receiver node should have higher observation score than the current node for destination of current packet. If there is no candidate receiver node with enough observation score, PRO checks for the information dissemination score of other nodes in the communication range. If the current node encounters a candidate node with information dissemination score greater than the internal threshold stored in the current node, then the packet is forwarded to that candidate node. The threshold for the information dissemination score, Nobs_Thr, is calculated by using a list of information dissemination scores of previously encountered nodes as discussed in Section 5.3.3. If there are no suitable nodes in the communication range, the message is kept until a new node with suitable conditions is encountered or time out.

In addition to these two criteria PRO restricts the number of copies that can be forwarded for each message. Forwarding_Quota represents the maximum number of copies that can be forwarded for a message by single node. Quota_Obs and Quota_Nobs are for restricting the number of copies that can be forwarded using observation and information dissemination scores correspondingly. We provide more detailed analysis about forwarding quota in experimental results section. As explained in Section 5.3.1, we use Forwarding_Quota = 2.

The pseudo code for the forwarding algorithm of PRO is given below:

5. Experimental Results

We start with an explanation of our dataset in Section 5.1, and discuss our experimental setup in Section 5.2. Section 5.3 presents an experimental analysis of PRO. We compare PRO with three well-known DTN protocols in Section 5.4. In Section 5.5, we measure the impact of the availability of Internet connection on routing performance. Finally, in Section 5.6, we present our experimental results on smartphone queries.

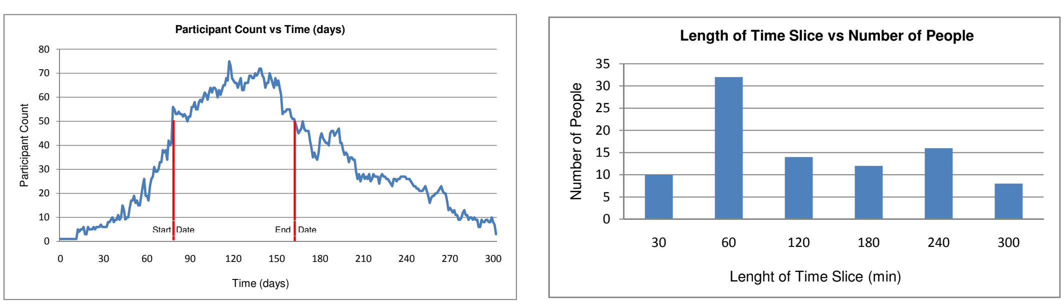
5.1. The Dataset

We use the Reality Mining dataset [12] from MIT Media Labs. The dataset was generated by an experimental study involving 100 people for the duration of 9 months, where each person is given a Nokia 6600 cellphone with software that continuously logs celltower connectivity data. Due to the lack of GPS in the Nokia 6600, the location is not recorded in terms of an exact longitude-latitude pair, but rather in terms of the celltower currently connected. However, in order to render the celltower ids meaningful, the cellphone software prompts the user to provide a tag when it encounters a celltower id for the first time. This way, the celltower locations are able to be tagged semantically with a specific meaning for that user. Thus, we have building level granularity in most common locations by using location tags.

While the experimental data is collected for the duration of 9 months, majority of the users did not participate in the experiments for the whole period. We analyzed the number of participant for each day in the 9 month period and selected the most crowded periods.

Algorithm 1 Forwarding Algorithm of PRO

// Direct Delivery To Destination
ForEach encountered $node_i$ **do**
  If $node_i = p.dest$ **and** $p.finalized = false$ **Then**
    If $p \notin node_i$ **Then**
      Forward p to $node_i$
    $p.finalized = true$
End For
// Give Priority to Observed Nodes
ForEach encountered $node_i$ **do**
  If $(p.obs + p.nobs) <$ Forwarding_Quota **Then**
    $tScore =$ calcObsScore($p.destination$, $node_i$)
    If $tScore > p.Score$ **and** $p.obs <$ Quota_Obs **and** $p \notin node_i$ **Then**
      Forward p to $node_i$
      $p.obs++$
End For
// NonObserved Carrier Nodes
ForEach encountered $node_i$ **do**
  If $(p.obs + p.nobs) <$ Forwarding_Quota **Then**
    $disScore =$ calcDisScore($this$, $node_i$)
    If $disScore >$ Nobs_Thr **and** $p.nobs <$ Quota_Nobs **and** $p \notin node_i$ **Then**
      Forward p to $node_i$
      $p.nobs++$
End For

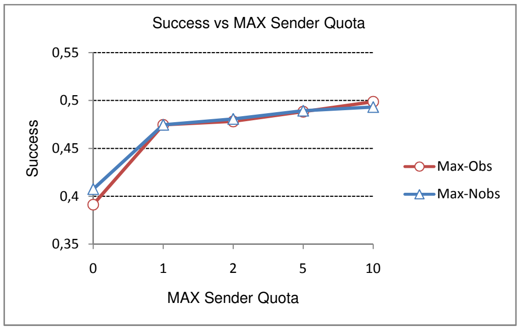


(a) Participant analysis

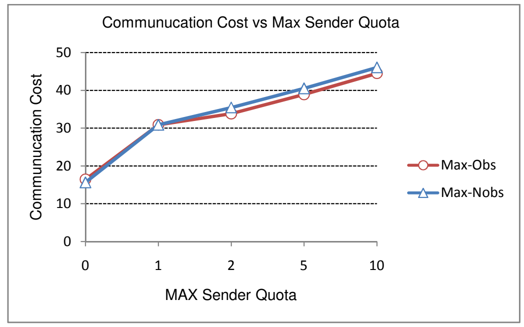
(b) Time slice length analysis

Figure 4. Experiments for analyzing the data set

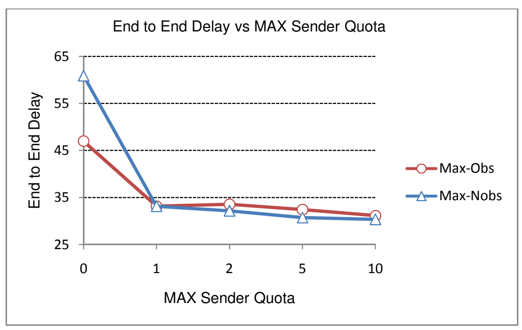
The graph for participant analysis is given in Figure 4(a). We separate the most crowded time intervals with two bars in the figure. The length of selected period is approximately



(a) Success comparison



(b) Cost comparison



(c) Experiments for analyzing the forwarding quota

3 months with at least 50 people joining the experiments on each day. Within this time period, we select the first one month period for our experiments on optimizing the PRO protocol in Section 5.3. The next two months period is used for comparison with other routing protocols, evaluation of Internet availability and evaluation of smartphone queries.

An important thing to note from this participation statistics is that, even in the most crowded period the participation is only 65%, hence this imposes an upperbound of 65% on the success of routing to any node in this dataset. The participant counts in Figure 4(a) are daily counts. The daily count includes the number of people that open their cellphones on that day regardless of the length of time period that the cellphone is open.

5.2. Experimental Setup

For performing our simulations, we implemented a basic MANET simulator which is fed with location information of individuals with connection time information (location name, connection and disconnection time). We then implemented routing protocols mentioned in Section 3 as plug-ins to the MANET simulator. All of the components of the evaluation system and routing protocols are developed in Java and consist of more than 7K Lines of code.

We next explain how we determine the duration of time slices to be used in the local observation tables of the PRO protocol. Recall that the idea of time slice length is that each consecutive time slice should be long enough to capture change in the surrounding set of nodes (nodes in communication range). The idea is that if the "observed person set" in two consecutive time interval are very similar, then there is not enough change in the set of neighboring nodes and we can combine these two time slices in to one.

In order to capture this change concept, we use vector similarity over the set of nodes observed in two consecutive time intervals. Let A be the set of nodes observed in time interval T_k and B be the set of nodes observed in the consecutive time interval T_{k+1}. First, we find $C = A \cup B$. Then, for each element in C, we generate observation vectors with length $|C|$ for both A and B. While generating observation vectors, $\forall$ $node_i \in C$ and $node_i \in$ corresponding set (A or B), the i-th component of observation vector for corresponding set becomes 1. If $node_i \notin$ corresponding set (A or B) then i-th component of observation vector becomes 0. Finally, we calculate the cosine similarity between these two observation vectors as a similarity metric between two consecutive time slices.

In our experiments, we tried 6 different time slice lengths between 30 minutes and 300 minutes. We left out time slice lengths less than 30 minutes and more than 5 hours. When using time slices that are less than 30 minutes, the cosine similarity between two consecutive time interval becomes zero due to the insufficient length of time interval (and not due to changes in social network dynamics). Also more than 300 minutes is too big because we cannot fit two such slices between 9.00 am and 7.00 pm when people are most active.

The results of these experiments are given in Figure 4(b). For each person, we calculate the average consecutive time slice similarity for 6 different time slice lengths, and then we choose the time slices with minimum similarity for the corresponding person. To illustrate; 60 minute is the time slice which has minimum similarity value for more than 30 people (which is nearly 1/3 of the whole population). After this analysis we have fixed the value of the time slice length to 60 minutes for all mobile nodes in the experiment.

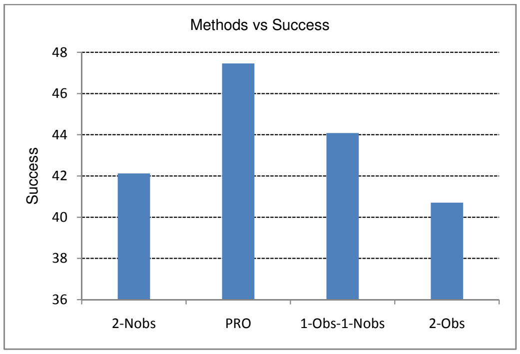
It may seem that increasing the time slice length should also increase the dissimilarity between two consecutive time slices since any node included within K minutes interval should also be included in L, $K < L$, minutes time intervals. However, while calculating similarity of two sets A and B, the relative magnitude of $A \cap B$ over $A \cup B$ is also important, since similar dimensions can dominate dissimilar dimensions during the computation. To illustrate: the angle between two vectors A=$(1, 1)$ and B=$(1, 0)$ is larger than the angle between A=$(1, 1, 1, 0, 0)$ and B=$(1, 1, 1, 1, 1)$ although the number of different dimensions in the latter case is bigger than that of the former.

5.3. Experiments on PRO

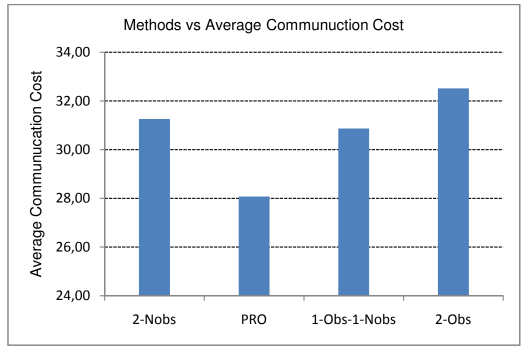
We present our experimental analysis of PRO in three subsections: analysis of maximum forwarding quota, analysis of routing strategies for spending forwarding quota, and finally reducing the communication overhead.

5.3.1. Determining The Number of Maximum Forwarding Quota

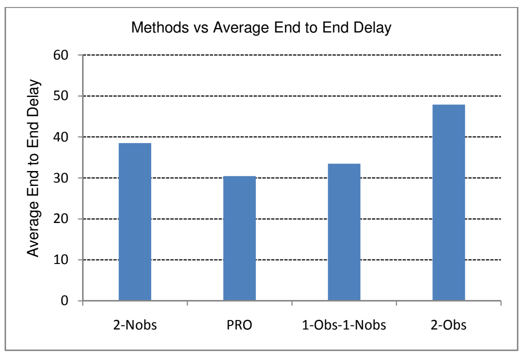
Here, we compare the performance of PRO with varying forwarding quotas. In our experiments we focus on determining the optimal maximum forwarding quota which corresponds to $Forwarding\_Quota = Quota\_Obs + Quota\_Nobs$ value. The results of these experiments are given in Figures 5(a)-5(b). In these Figures, for the line labeled with circle data points (Max-Obs), we fix Quota_Nobs to 1 and vary Quota_Obs from 0 to 10



(d) Success comparison



(e) Cost comparison



(f) Delay comparison

Figure 5. Experiments for analyzing quota spending strategies

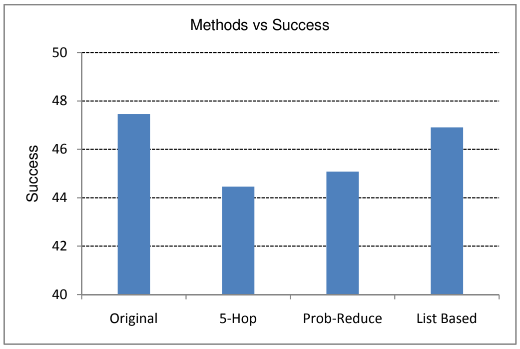
copies. In the same Figures, the line with the triangle data points (Max-Nobs) we fix Quota_Obs=1 and vary Quota_Obs from 0 to 10 copies.

Figures 5(a)-5(b) support our theoretical results in Section 3, since there is a significant tipping point in terms of success, communication cost, and end to end delay when the Forwarding_Quota is set to 2. Here success is defined as the ratio of messages that arrived to the destination over the number of all generated messages. Analyzing the trend in the graphs, we can see that increasing total send quota after 2 does not contribute to the performance significantly. Therefore, we choose Forwarding_Quota = 2 for PRO algorithm. Our results can also be generalized for different PSNs which depend on social interactions in human networks since the size of our data set is large and it exhibits properties of human social networks [13, 26, 46].

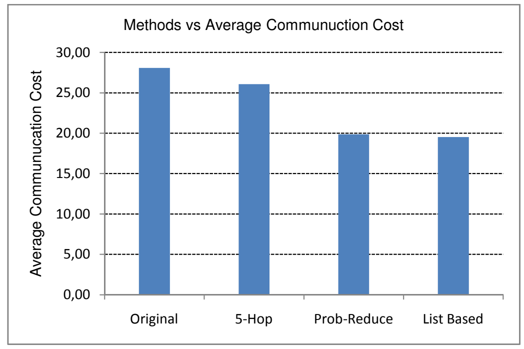
5.3.2. How To Spend the Forwarding Quota

Here, we present experimental results about how to allocate Forwarding_Quota among Quota_Obs and Quota_ Nobs. We investigate the following four combinations:

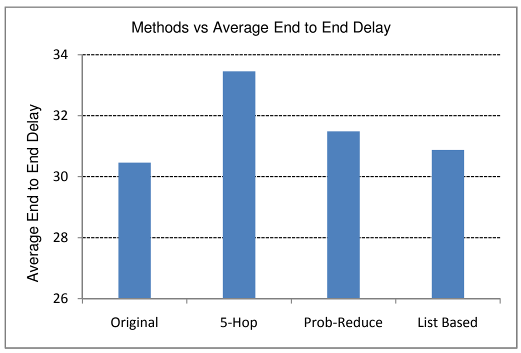
- In the first combination (2-Nobs), we require PRO to use the entire forwarding quota on Non-observed nodes. In other words we use (0, 2) for (Quota_Obs, Quota_Nobs)



(a) Success comparison



(b) Cost comparison



(c) Delay comparison

Figure 6. Experiments for reducing communication overhead

combination.

- The second combination corresponds to PRO as described in Section 4. This is a flexible approach that gives priority to Observed nodes (when available) over Non-observed nodes.
- In the third combination (1-Obs-1-Nobs), we use strictly $(1, 1)$ for (Quota_Obs, Quota_Nobs).
- The fourth combination (2-Obs) is the dual of the first combination, we require the algorithm to spend the entire forwarding quota on Observed nodes.

The results of these experiments are given in Figures 5(d)-5(f). We observe that the second combination outperforms the others in terms of success, overhead, and end to end delay. The important result here is that there may be some states during execution where none of the candidate forwarder nodes include any Observed nodes (especially in the beginning stages of the routing), and in this case using information dissemination score (Non-observed nodes) contributes significantly for the routing performance. In the remaining of the paper, we use PRO with this second combination as our base protocol.

5.3.3. Reducing Transmission Overhead

In this section, we investigate mechanisms for reducing communication overhead. The key idea for reducing communication overhead is that the probability of delivery increases with the hop count. Thus, to reduce the communication overhead in our protocol, we reduce the probability of forwarding due to information dissemination scores (i.e., forwarding to Non-observed nodes) as the hop-count increases. We do not use any reduction for transmissions due to observation score since the probability of delivery to the destination is high for observed nodes. We investigate the following versions of PRO for reducing the communication overhead:

5-Hop: In 5-Hop scenario, message transmissions due to information dissemination score are entirely stopped after 5-Hops.

Probabilistic Reduction: In probabilistic reduction scenario, message transmissions due to information dissemination score are decreased with the factor $1/k$ where k is the current hop count ($k > 1$). In other words, the probability of transmission due to information dissemination score becomes $1/k$ at the $k-th$ hop.

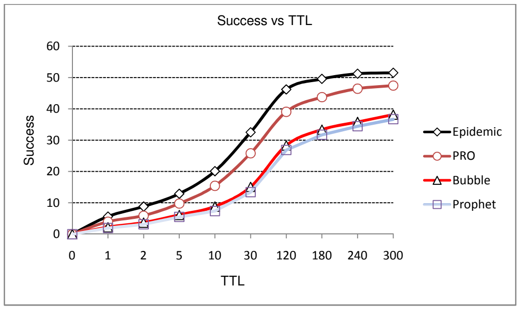
List Based Reduction: In this case, each mobile node keeps a sorted list of information dissemination scores of previously encountered nodes. Each score is updated with the most recent observation. At hop k, a message is transmitted only if the candidate forwarder node has higher information dissemination score than the average of the top $1/k$ portion of the whole list.

We compare these three scenarios with the original PRO with no transmission reduction. The results are given in Figures 6(a)-6(c) in terms of average communication cost, message success and average end to end delay. It should be noted that list based approach and probabilistic reduction decreases communication overhead significantly (nearly 30%). Among the three cases, list based approach gives the best results in terms of both end to end delay and overhead with similar success rates as the original version. Therefore we use list based version of PRO as our base protocol and compare it with other protocols in the next section.

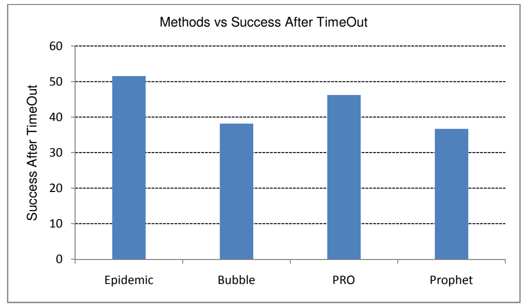
5.4. Comparison with Other Routing Methods

In this section, we compare PRO with three popular MANET protocols: Epidemic Routing, Bubble Rap and Prophet routing. The details of the routing protocols are discussed in Section 2. For the Bubble-rap, we use a single community case, because using optimal k-community with distributed community detection requires testing and pre-knowledge of k [26], but we want all of the routing algorithms to be self-contained and independent from the dataset. Moreover, since all of the people in the experimental data set are affiliated with MIT, they usually visit the same buildings at MIT frequently [2]. For Prophet [35], we use the delivery prediction function mentioned in Section 2. Each of these protocols has passive back-infection for the successfully delivered messages. That is if a forwarder node encounters another node which contains the status of current message as delivered, then the forwarder node also changes the status of the current message as delivered. Then, this message is not forwarded to any other node and is deleted. We also use a timeout of 5 hours: when this timeout value is elapsed, the corresponding message is deleted from the current node.

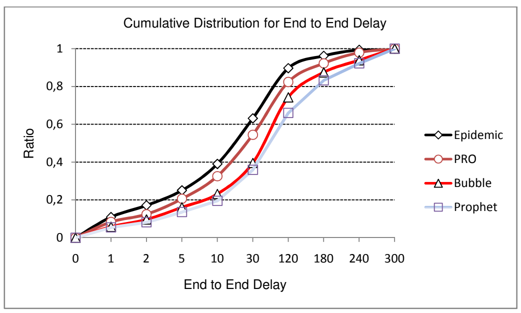
The results of our comparison experiments are given in two separate sets, on cell based



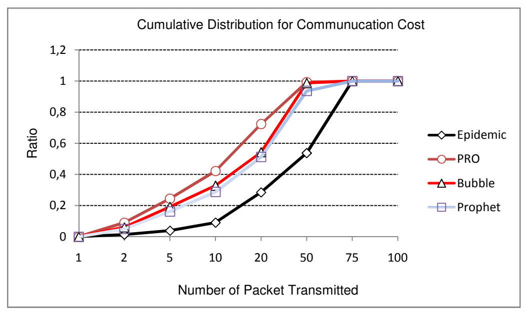
(a) Cumulative success comparison



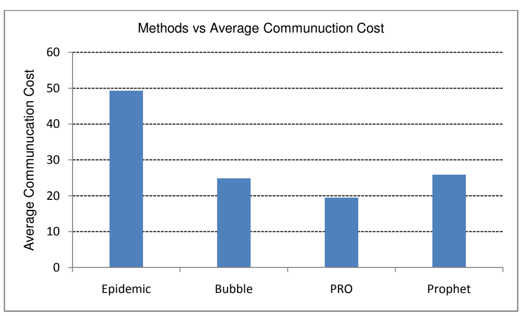
(b) Average success comparison



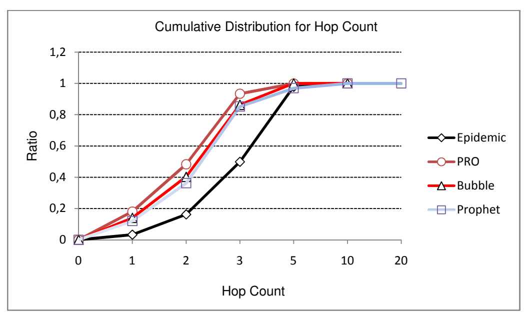
(c) Delay comparison



(d) Cumulative cost comparison



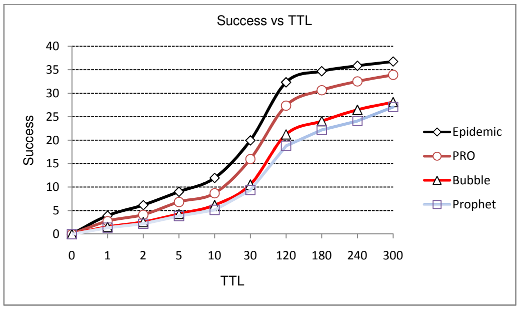
(e) Average cost comparison



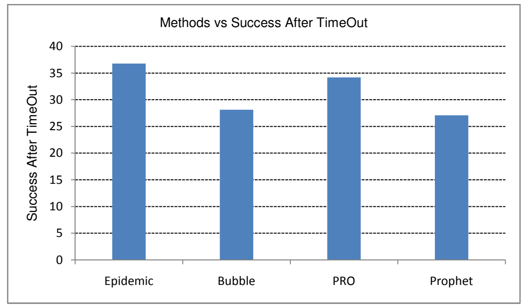
(f) Number of hops comparison

Figure 7. Comparison with other routing protocols

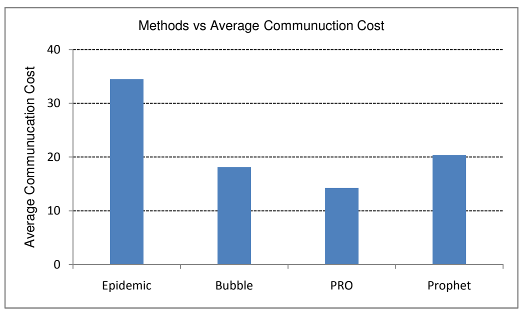
location data (Figures 7(a)-7(f)) and blue tooth connection data (Figures 8(a)-8(c)). For the success comparison over cell based location data, we provide two different graphical views which are cumulative success distribution and average success. Figures 7(a)-7(b) show that the success of PRO is closer to epidemic routing than other methods. When the average success is examined, the average success of PRO is found to be 25% better than that of Bubble-rap and Prophet. The success of PRO is around 47% whereas that of Bubble-rap and Prophet are under 38%. When we analyze the cumulative distribution of arrived messages with respect to arrival time (Figure 7(c)), we also see that PRO outperforms Bubble-rap and Prophet. The difference is even bigger in intermediate points



(a) Cumulative success comparison



(b) Average success comparison



(c) Average Cost Comparison

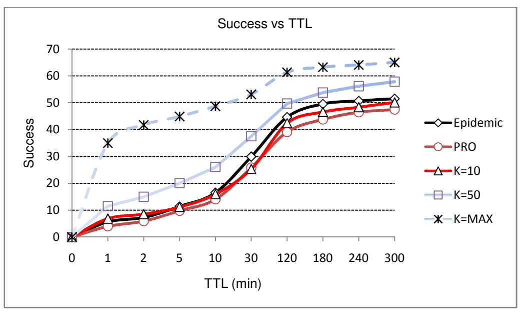
Figure 8. Experiments on Bluetooth Connection Data

such as 30 min where PRO is relatively 30%-35% better than Bubble-rap and Prophet.

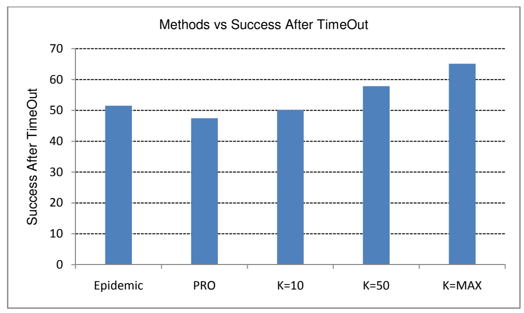
We also measure the communication costs and the average number of hops needed for packages. Similar to the analysis above we provide cumulative distribution and average views for these results (Figures 7(d)-7(f)). For the communication cost, we count the transmitted copy of the initial message until all copies are deleted. Each copy carries the initial timestamp of the original message and if the timeout value is elapsed with respect to initial timestamp then the corresponding messages are deleted. Mobile nodes also keep the status of each message in terms of delivery success and use back-infection concept as discussed above.

Figures 7(d)-7(e) show that the communication cost of PRO is 20% better than Bubble-rap and Prophet. From Figure 7(e), the average communication cost of PRO is around 20 messages whereas Buble-rap and Prophet has communication cost around 25 messages. That is, PRO outperforms Buble-rap and Prophet in terms of delay performance, success, and communication overhead. Moreover, the delay and success performance of PRO is very close to Epidemic routing while its communication overhead is at least 2 times better than Epidemic routing.

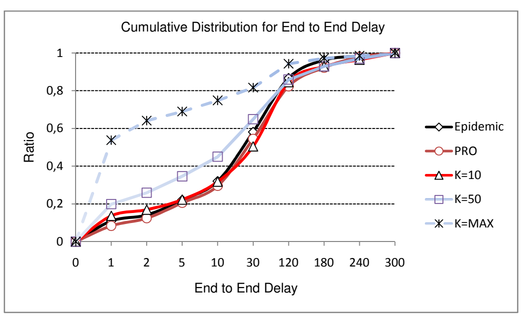
We provide three graphical views for the experiments over the peer to peer bluetooth connection data (Figures 8(a)-8(c)). The relative results we obtain from the performance experiments are similar to the ones obtained from the experiments on cell based location data. However, the average success of the all methods decreases 30% in the Blue tooth experiments (Figures 8(a)-8(b)) since there is less intercontact opportunities. Remember that we accept two peer as connected to each to other when they are in the same cell in the experiments over the cell based location data. However two peer are connected to each other only when they have short range blue tooth connection in the second case which decreases the length of intercontact times and number of intercontact events. Unlike the decrease in the success performance, the cost performance are improved in the bluetooth connnection case since there is less intercontact events and less messages forwarded to thorough the network (Figures 8(c)).



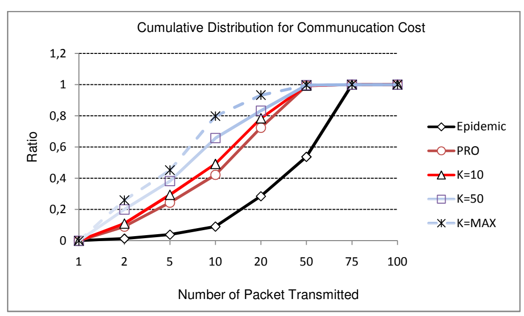
(a) Cumulative success comparison



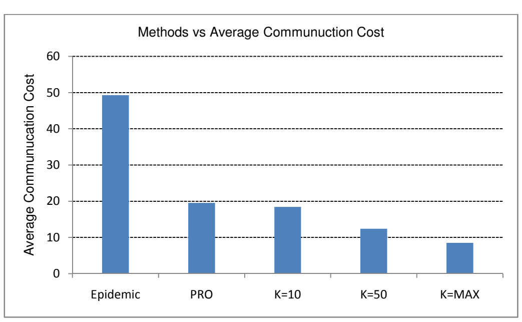
(b) Average success comparison



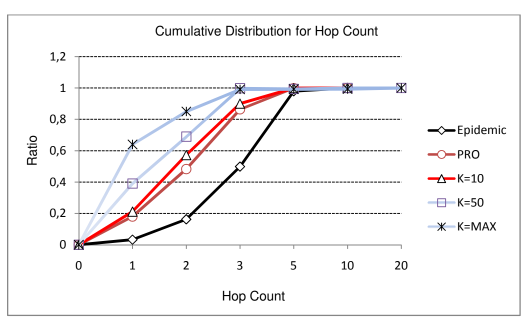
(c) Delay comparison



(d) Cumulative cost comparison



(e) Average cost comparison



(f) Number of hops comparison

Figure 9. Impact of Internet connection on routing performance

5.5. The Impact of Internet Connection on Routing Performance

The new generation smartphones are equipped with 802.11 connection capability which enables them to connect to Internet without using data plans from telephone service providers. The idea of uploading data from sensors or smartphones to the Internet by opportunistic connection is a popular one [18, 27]. Here, we use the same facility to enhance our system by adding 802.11 capacity to mobile nodes and Internet access points to particular locations. We assumed so far that two nodes can communicate with each other when they are in the same location. In the Internet enabled scenario of our experiments, we locate access points at random locations and enable mobile nodes to communicate with each other via Internet. This way, we provide a logical connection between two nodes even they are in different locations, provided that both locations have access points.

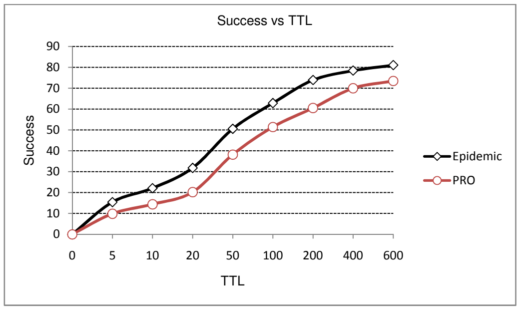
We measure the impact of Internet connection availability on routing performance by trying different densities of Internet access points. To this end, we first found the dominating celltower locations where the significant amount of simulation time (more than 99%) is elapsed. As a result of our dominating set analysis, we found nearly 240 (K=MAX) such locations. We then placed access points to random subset of these celltowers in the simulation. In our simulations we try 3 setups. In the first one we select $K = 10$ random locations, in the second one we select $K = 50$ random locations, and in the last one we enable all of the locations with Internet connection. When we analyze the success of the Internet enabled version in Figures 9(a)-9(b), we see that the success of message delivery goes to maximum 65% (30% improvement with respect to original version) which is also the theoretical upper bound for any protocol. The remaining 35% gap is due to the fact that the destination node cannot be reachable at any time as given in participant analysis graph (Figure 4(a)). If we look at the cumulative distribution graph for end to end delay, we can see that more than half of the messages are transmitted in less than one minute by K=MAX scenario (Figure 9(c)). The average delay for K=MAX scenario is around 7 minutes which is much better than the original version of PRO protocol (more than 30 minutes).

The improvement in Internet enabled scenario is also observed in the communication cost. As seen from Figures 9(d)-9(e), the communication cost of K=MAX is less than half of the original version (without Internet connection). We also observe significant improvements in the hop count. On average a single packet arrives to destination in 1.5 hops in K=MAX scenario, whereas each packet needs 2.5 hops in the original version of PRO(Figure 9(f)).

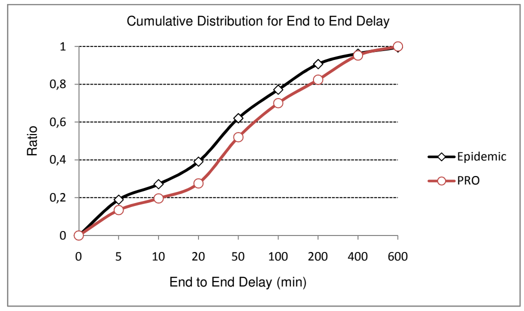
Our results in Figures 9(a)-9(f) also show that even for the relatively small K=50 case (that is, with Internet connectivity at 20% of locations) significant improvements are observed over the performance of the original version of PRO.

5.6. Experiments on Smartphone Queries

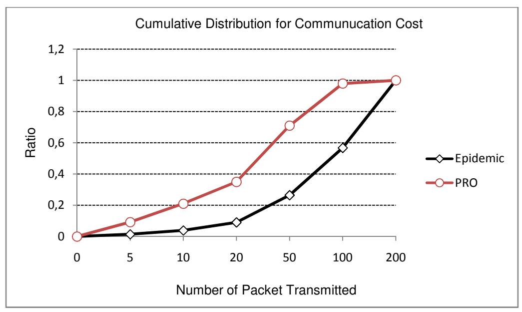
In this section, we present our experimental results related to the smartphone "point queries" we mentioned in the Introduction. In our experiments these queries are sent by random nodes asking for random locations. In order to update PRO to handle point queries the only modification we make is to give locations an id number and store observation scores at each node for the locations (similar to the way observation scores are kept for each encountered node id).



(a) Cumulative success comparison



(b) Average success comparison



(c) Delay comparison

Figure 10. Experiments on smartphone point queries

The query forwarding phase for a point query is carried out in the same manner as routing to a node id; the only difference is in this case the node id is the id of the location the point query asks to sample. The observation score and information dissemination scores with respect to the location id are used without any changes. When a node receives a query packet which asks for an information related to its current location or near future location, the node replies to the query immediately if it is already on query location, or later when it enters the query location. The reply is rerouted back to the id of the node that initiated the query using PRO.

For this section, we only compare with epidemic routing. Figures 10(a)-10(c) show that the success and delay performance of PRO is considerably close to epidemic routing (10% more delay on the average, 8% less success). Yet, the communication overhead of PRO is at least 2.5 time better than Epidemic routing. In fact, the average communication cost per query is around 40 messages for PRO whereas this value is more than 100 messages for epidemic routing.

6. Concluding Remarks

In this paper, we presented a novel routing protocol, PRO, for profile-based routing in PSNs. Differing from previous routing protocols, PRO treats node encounters as periodic patterns and uses them to predict the times of future encounters. Exploiting the regularity of human mobility profiles, PRO achieves fast (low-delivery-latency) and efficient (low-message-overhead) routing in intermittently connected PSNs. Our experiment results using the Reality Mining dataset show that PRO achieves similar success rate and latency (10% less success and 10% more delay time) as the epidemic routing with less than half the communication cost of the epidemic routing. PRO also outperforms the Prophet and Bubble-rap routing protocols (at least 20% less delay time and 25% more success) with less communication cost (at least 25% less communication than these two protocols).

Despite being simple, PRO constitutes a general framework, that can be easily instantiated to solve searching and querying problems in smartphone networks. In this paper we instantiated PRO to solve the smartphone point queries, and presented performance results for that scenario. Another interesting scenario for smartphone querying is what we call as I-spy queries, inspired by the "I spy ..." children game. This scenario is on image search. In contrast to the first scenario, in this scenario the location is not well-defined. Rather the user asks for a picture of an object that fits his description in this vicinity, such as a red signpost or a big oak tree. To instantiate PRO to query for this description, the description is first hashed using SIFT descriptors [36] and an id is produced. PRO is then employed to route a message to this id. Of course, this is not an exact match search, so approximate matching techniques should be investigated. I-spy querying also requires that nodes exchange the SIFT descriptors of the images they store when they meet. So, another open research question for I-spy querying is on performing this advertising and querying in a scalable and peer-to-peer manner. The privacy and security aspects of point queries and I-spy queries also need to be investigated.

REFERENCES

1. S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, editors. *Mobile Adhoc Networking*. IEEE Press John Wiley, 2004.
2. M. A. Bayir, M. Demirbas, and N. Eagle. Discovering spatiotemporal mobility profiles of cell phone users. *WOWMOM*, 2009.
3. C. Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. *Mobile Computing and Communications Review*, 5(3):55–66, 2001.
4. C. Boldrini, M. Conti, and A. Passarella. Impact of social mobility on routing protocols for opportunistic networks. In *WOWMOM*, pages 1–6, 2007.
5. C. Boldrini, M. Conti, and A. Passarella. Exploiting users' social relations to forward data in opportunistic networks: The hibop solution. *Pervasive and Mobile Computing*, 4(5):633–657, 2008.
6. A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. Mob. Comput.*, 6(6):606–620, 2007.
7. A. Chaintreau, A. Mtibaa, L. Massoulié, and C. Diot. The diameter of opportunistic mobile networks. In *CoNEXT*, page 12, 2007.
8. E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *MobiHoc*, pages 32–40, 2007.
9. J. A. Davis, A. H. Fagg, and B. N. Levine. Wearable computers as packet transport

mechanisms in highly-partitioned ad-hoc networks. In *ISWC*, pages 141–148, 2001.
10. H. Dubois-Ferrière, M. Grossglauser, and M. Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *MobiHoc*, pages 257–266, 2003.
11. N. Eagle and A. Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 04-2:28–34, 2005.
12. N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.
13. N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. 2009.
14. D. T. et al. Design and evaluation of an individually simulated mobility model in wireless ad hoc networks. In *CNDSMSC*, 2002.
15. G. P. et al. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814 – 818, 2005.
16. L. P. et al. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44:134–141, 2006.
17. R. G. et al. The message delay in mobile ad hoc networks. *Perform. Eval.*, 62(1-4):210–228, 2005.
18. S. B. E. et al. The bikenet mobile sensing system for cyclist experience mapping. In *SenSys*, pages 87–101, 2007.
19. K. R. Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM*, pages 27–34, 2003.
20. M. Grossglauser and M. Vetterli. Locating nodes with ease: Mobility diffusion of last encounters in ad hoc networks. In *INFOCOM*, 2003.
21. Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. *IEEE/ACM Trans. Netw.*, 14(1):27–40, 2006.
22. X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *ACM/ IEEE MSWIM*, pages 53–60, 1999.
23. P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *WDTN*, 2005.
24. P. Hui and J. Crowcroft. Human mobility models and opportunistic communication system design. *Royal Society Philosophical Transactions B*, 366(1872), 2008.
25. P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *MobiHoc*, pages 241–250, 2008.
26. P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In *2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, 2007.
27. B. Hull and et al. Cartel: a distributed mobile sensor computing system. In *SenSys*, pages 125–138, 2006.
28. S. Jain, K. R. Fall, and R. K. Patra. Routing in a delay tolerant network. In *SIGCOMM*, pages 145–158, 2004.
29. A. P. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *MOBICOM*, pages 217–229, 2003.
30. D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

31. E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward. Practical routing in delay-tolerant networks. *IEEE Trans. Mob. Comput.*, 6(8):943–959, 2007.
32. J. Kleinberg. Navigation in small world. *Nature*, Nature 406:845, 2000.
33. J. M. Kleinberg. The small-world phenomenon: an algorithm perspective. In *STOC*, pages 163–170, 2000.
34. J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for dtns. In *INFOCOM*, 2006.
35. A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. In *SAPIR*, pages 239–254, 2004.
36. D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 2004.
37. P. V. Marsden. Egocentric and sociocentric measures of network centrality. *Social Networks*, 24(4):407–422, October 2002.
38. M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103:85778582, 2006.
39. M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2006.
40. A. K. Pietilainen and C. Diot. Experimenting with real-life opportunistic communications using windows mobile devices. In *CoNEXT*, page 65, 2007.
41. J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A networking architecture designed around mobile users. In *IFIP WONS*, 2006.
42. 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 *MobiHoc*, pages 233–244, 2003.
43. K. Tan, Q. Zhang, and W. Zhu. Shortest path routing in partially connected ad hoc networks. In *GLOBECOM*, volume 2, pages 1038–1042, 2003.
44. A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks.
45. S. Wasserman and K. Faust, editors. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
46. E. Yoneki, P. Hui, and J. Crowcroft. Visualizing community detection in opportunistic networks. In *Challenged Networks*, pages 93–96, 2007.
47. X. Zhang, G. Neglia, J. F. Kurose, and D. F. Towsley. Performance modeling of epidemic routing. *Computer Networks*, 51(10):2867–2891, 2007.