PRO: A Profile-based Routing Protocol for Pocket Switched Networks

Murat Ali Bayir
Computer Science & Engineering
University at Buffalo, SUNY

Murat Demirbas
Computer Science & Engineering
University at Buffalo, SUNY

***Abstract*—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 adhoc smartphone networks. We validate the performance of PRO with the "Reality Mining" dataset containing 350K hours of celltower connectivity and Bluetooth connection data, and compare its performance with that of previous approaches.**

***Index Terms*—Pocket Switched Networks, Human Mobility, Opportunistic Routing**

I. Introduction

Cellphone technology has seen an adoption rate faster than any other technology in human history [1]: as of 2009, the number of cellphone subscribers has exceeded 3.3 billion users. 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 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, 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 networks, or opportunistic, store-and-forward networks [2], [3], [4], [5] 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. So, 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], [7], [8], [9] 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 [10], [11], [12], [13]. These work assume different models on human mobility and community-structure, and use them for making routing decisions. Compared to DTN protocols, PSN protocols make use of more information about the network (context awareness), 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. Our previous analysis [14] of MIT's Reality Mining dataset, which is one of the biggest publicly available cellphone connectivity data with 350K hours of celltower connectivity logs [15], shows that significant amount of human mobility (85%) exhibits spatial and temporal regularity where users move between their top-k locations.

Here, 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.
- 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. Due to the space limitations we relegate the details of our theoretical analysis to our technical report [16]. Here we support our theoretical analysis with experimental results in Section IV.
- 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.
- Unlike the synthetic test sets generated by simulators, we validate the performance of our routing protocol with a real dataset. We use the "Reality Mining" dataset [15] which is one of the largest publicly available datasets containing more than 350K hours of celltower and Bluetooth connection data. We choose the Reality Mining dataset for our validation since it is used as an evaluation batch for several works [17], [18] and it is shown to have similar user behavior with several other datasets which implies that the observed phenomenons are not a specific artifact of the data itself [6], [19]. Using the Reality Mining dataset, we compare the performance of our protocol with previous approaches over both cell-based mobility data (**coarse granularity**) and Bluetooth 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 [20] with less than half the communication cost of the epidemic routing. PRO also outperforms the Prophet [5] and Bubble-rap [12] 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).
- PRO routing protocol is completely decentralized and local to the nodes. PRO runs in an adhoc manner and does not depend on any central infrastructure or third party like Telephone Service Providers.
- 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 II we discuss related work on PSNs. In Section III, 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 IV. Finally, we conclude with Section V.

II. 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 IV 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* [20] 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], [22] 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 [23], reference point group mobility model [24], and entity based approaches [25], [26] 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* [5]. 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 [27], [28], [29], [5], [30] 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 [11], [31], [12], on the other hand, use social network structure of humans in routing decisions.

Bubble-rap [12] 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.

III. PRO: Profile Based Routing for Pocket Switched Networks

A. 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. Recent works [19], [34], [35], [36], [37] refined this model and showed that human networks can be modeled as community graphs given in Figure 1. 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 to the same community are called as local neighbors (black edges in Figure 1) and vertices attached to the two sides of edges between different communities are called as remote neighbors (gray edges in Figure 1).

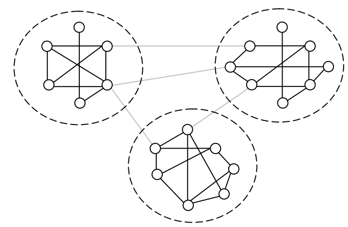


Fig. 1. Community structure in human networks

In a recent work [18], the regularity of inter-contact events in Bluetooth 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 duration and the number of meetings. In our previous work [14], we also 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 will use the similar observation that people in the same community (students in the same class, co-workers) are most likely to meet almost regularly in the same set of locations.

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 [28], [29], [5], [30], [27] 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 that encounter in only in the mornings history based approaches still incorrectly produce very high forwarding probability during afternoons. The same problem also occurs for routing protocols [11], [31], [12] utilizing social network structure; the high popularity of a node in the social network does not guarantee its high popularity at certain 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.

B. 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.

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 [14] which analyzes the Reality Mining dataset. In our design, each cell in the local observation table (Figure 2) 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 $isObserved{=}\ 0$.

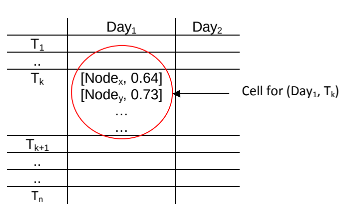


Fig. 2. Structure of observation table

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

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 1.

In PRO, we use a distributed approach based on the concept of Ego networks [38]; 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 of 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 IV-B3. 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. As explained in theoretical analysis section of our technical report [16] and Section IV-B, we use Forwarding_Quota $= 2$. The pseudo code for the forwarding algorithm of PRO is given in Algorithm 1.

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

IV. Experimental Results

We start with an explanation of our dataset and experimental setup in Section IV-A. Section IV-B presents an evaluation of design parameters for PRO. We compare PRO with three well-known DTN protocols in Section IV-C. Finally, in Section IV-E, we present our results on smartphone queries.

A. The Dataset and Experimental Setup

For our experimental evaluation we use the Reality Mining dataset [15] from MIT Media Labs. This dataset was generated by an experiment involving 100 people for the duration of 9 months, where each person is given a Nokia 6600 cellphone. Reality Mining data contains both cellular connectivity and fine granularity peer to peer Bluetooth connection data which makes it very suitable to use as evaluation batch for various routing protocols. We choose the Reality Mining dataset since it is one of the biggest publicly available one set which is already compared with several other datasets in different aspects like cellular connectivity duration [9], Bluetooth connection durations [6], social networks [19] and human mobility [18]. These work showed that the observed phenomenons in the Reality Mining dataset is not a specific artifact of the experiment itself and the dataset is a representative sample of general human mobility and social interaction events.

While the experimental data is collected for the duration of 9 months period, the majority of the users did not participate in the experiments for the whole period. So we selected most crowded 3 months time interval in terms of participant count. We also analyzed the duration of time slices which is used by PRO routing protocol. In order to find suitable time slice length we have used cosine vector similarity and histogram analysis techniques. Our analysis shows that 1 hour time slice duration is the most reasonable time slice length for PRO routing protocol. Since our dataset is good representative of human behavior, these results can be used in different deployments for PSNs. Due to the space limitations here, we refer the reader to our technical report [16] to find detailed discussion about participant and time slice length analysis.

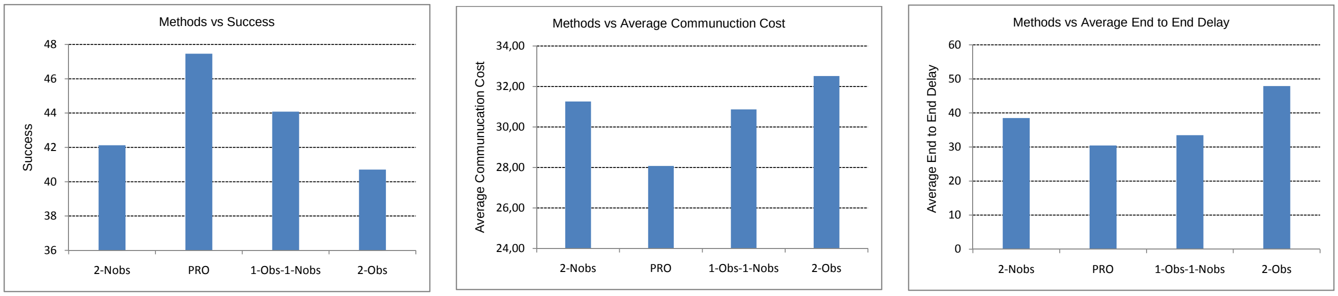
For running routing protocols, we implemented a basic MANET simulator which can be fed with location information of individuals [14] with cell connectivity data as well as Bluetooth connectivity data. Over this simulator, we then implemented routing protocols mentioned in Section 3 as plug-ins. All of the components of the evaluation framework are developed in Java and consist of more than 7K Lines of code.

B. 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.

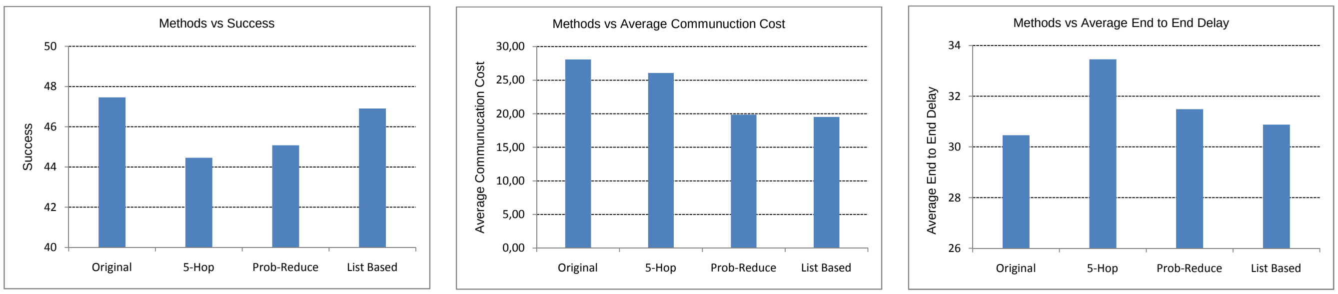
1) Determining The Number of Maximum Forwarding Quota: In this section, we compare the performance of PRO with varying forwarding quotas. Herer, we focus on determining the optimal maximum forwarding quota which corresponds to $Forwarding\_Quota = Quota\_Obs + Quota\_Nobs$ value. Due to the space limitations we only provide experimental results related to success of different versions of PRO protocol (Figure 5). The success is defined as the ratio of messages that arrived to the destination over the number of all generated messages. In Figure 5, for the line labeled with circle data points (Max-Obs), we fix Quota_Nobs to 1 and vary Quota_Obs from 0 to 10 copies. In the same figure, the line with the triangle data points (Max-Nobs) we fix Quota_Obs=1 and vary Quota_Obs from 0 to 10 copies. The Figure 5 shows that there is a significant tipping at point Forwarding_Quota = 2. The similar behavior is shown at point Forwarding_Quota = 2 in the cost and delay analysis which supports our theoretical results given in the technical report [16]. Therefore we decided to use Forwarding_Quota = 2 in the PRO routing protocol.

2) How To Spend the Forwarding Quota: Here, we present experimental results about how to distribute Forwarding_Quota among Quota_Obs and Quota_ Nobs. We investigate the following four combinations:



(a) Success comparison (b) Cost comparison (c) Delay comparison

Fig. 3. Experiments for analyzing quota spending strategies



(a) Success comparison (b) Cost comparison (c) Delay comparison

Fig. 4. Experiments for reducing communication overhead

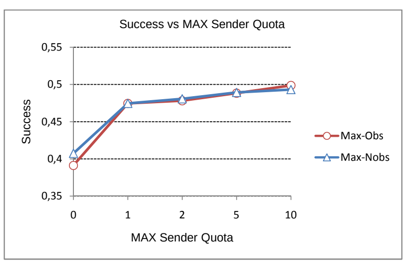


Fig. 5. Analyzing forward quota in terms of success

- 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) combination.
- The second combination corresponds to PRO as described in Section III. 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 3(a)-3(c). 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 in the network where there is no observed nodes (especially in the beginning stages of the routing), and in this case using information dissemination score (nonobserved nodes) contributes significantly for the routing performance. In the remaining of the paper, we use PRO with this second combination as our base protocol.

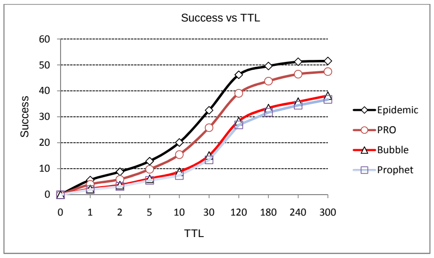
3) Reducing Transmission Overhead: Here, we investigate mechanisms for reducing communication overhead. Our key observation is that the probability of delivery increases with the hop count. Thus, to reduce the communication overhead, we reduce the probability of forwarding to nonobserved nodes (forwarding due to information dissemination scores) as the hop-count increases [1]. We investigate these mechanisms of PRO to this end.

5-Hop: Here, the message transmissions due to information dissemination score are entirely stopped after 5-hops.

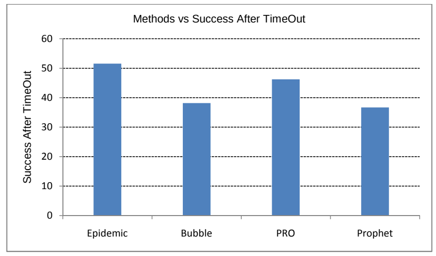
Probabilistic Reduction: In probabilistic reduction scenario, message transmissions due to information dissemination score are decreased with the factor $1/k$ where k is the current hop count $(k > 1)$. In other words, the probability of transmission due to information dissemination score becomes $1/k$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

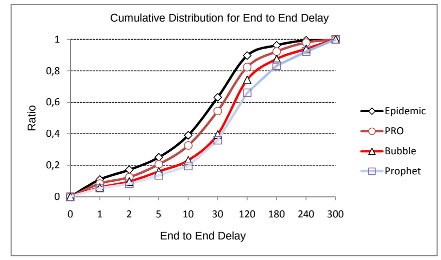
[1]We do not cut back transmissions to observed nodes since their probability delivery is higher.



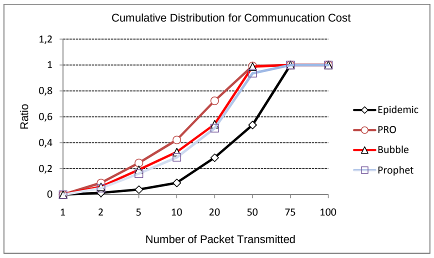
(a) Cumulative success comparison



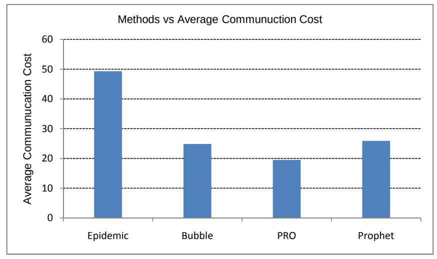
(b) Average success comparison



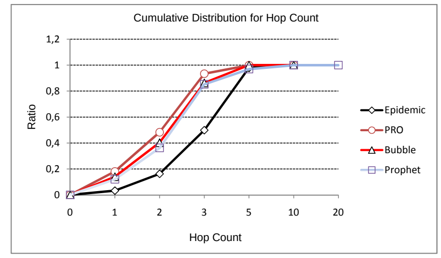
(c) Delay comparison



(d) Cumulative cost comparison

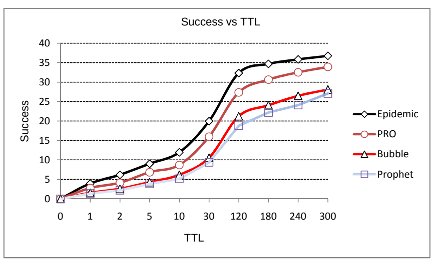


(e) Average cost comparison

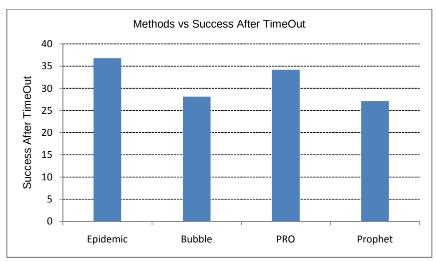


(f) Number of hops comparison

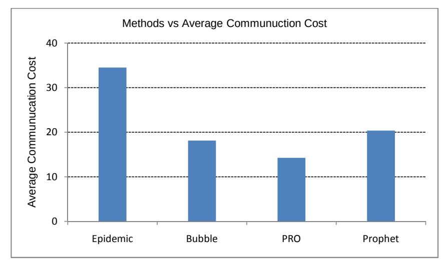
Fig. 6. Comparison with other routing protocols



(a) Cumulative success comparison



(b) Average success comparison



(c) Average cost comparison

Fig. 7. Experiments on Bluetooth connection data

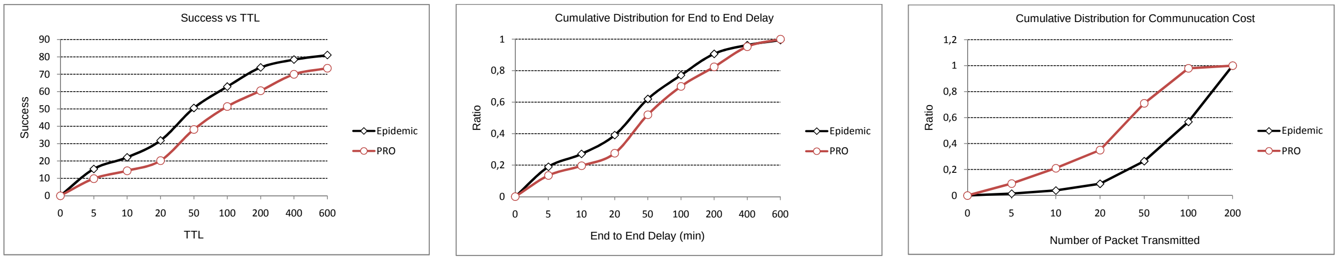
only if the candidate forwarder node has higher information dissemination score than the average of the top $1/k$ portion of the whole list.

We compare these three scenarios with the original PRO with no transmission reduction (Figures 4(a)-4(c)). 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.

C. 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 II. 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 [19], but we want all of the routing algorithms to be self-contained and independent from the dataset. Here time slice length is the only information that we used for PRO. However as we explained in previous sections our dataset is good representative of human behavior, our time slice length still remains considerable value for other deployments.For Prophet [5], we use the delivery prediction function mentioned in Section II. 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 location data (Figures 6(a)-6(f)) and Bluetooth connection data (Figures 7(a)-7(c)). For the success comparison over cell based location data, we provide two figures including cumulative success distribution



(a) Cumulative success comparison

(b) Average success comparison

(c) Delay comparison

Fig. 8. Experiments on smartphone point queries

and average success. Figures 6(a)-6(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 6(c)), we also see that PRO outperforms Bubble-rap and Prophet. The difference is even bigger in intermediate points 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 6(d)-6(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 6(d)-6(e) show that the communication cost of PRO is 20% better than Bubble-rap and Prophet. From Figure 6(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.

D. Experiments on Bluetooth dataset

We provide three different figures for the experiments on the peer to peer Bluetooth connection data (Figures 7(a)-7(c)). Our first observation is that the success performance of all methods are 30%-35% lower compared to celluar data experiments. The reason is that there is less connection opportunity between the pairs due to nature of Bluetooth data. Remember that we accept two nodes are connected if they are in the same cell in the celluar data experiments. However, in this experiment two nodes are only connected if there is a peer to peer short range Bluetooth communucation between them. Unlike the success performance, the cost performances of all methods are improved since there is a trade off between the number of messages floodeed into the network and success rate (parallel to end to end delay).

Our second observation is that relative performance of PRO protocol is similar the ones that we obtained from experiments over cellular connectivity data. The average success of PRO is 20%-25% better than Prophet and Buble-rap while achieving significantly less communucation overhead than Epidemic routing. The reason is that the regularity of human mobility is also inherently contained in the fine granularity Bluetooth data.

E. Experiments on Smartphone Queries

In this section, we present our experimental results related to the smartphone "point queries" we mentioned in the Introduction. Here point queries are pushed to the system by random mobile nodes asking for random locations. In order to update PRO to handle point queries the only modification we make is adding new observation table which stores visited locations (cellular id) instead of observed nodes.

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 8(a)-8(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.

V. 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 [39] 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] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software," *IEEE Pervasive Computing*, vol. 04-2, pp. 28–34, 2005.

[2] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., *Mobile Adhoc Networking*. IEEE Press John Wiley, 2004.

[3] K. R. Fall, "A delay-tolerant network architecture for challenged internets," in *SIGCOMM*, 2003, pp. 27–34.

[4] S. Jain, K. R. Fall, and R. K. Patra, "Routing in a delay tolerant network," in *SIGCOMM*, 2004, pp. 145–158.

[5] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *SAPIR*, 2004, pp. 239–254.

[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.*, vol. 6, no. 6, pp. 606–620, 2007.

[7] E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical routing in delay-tolerant networks," *IEEE Trans. Mob. Comput.*, vol. 6, no. 8, pp. 943–959, 2007.

[8] L. P. et al., "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, pp. 134–141, 2006.

[9] A. P. et al., "Experimenting with real-life opportunistic communications using windows mobile devices," in *CoNEXT*, 2007, p. 65.

[10] C. B. et al., "Impact of social mobility on routing protocols for opportunistic networks," in *WOWMOM*, 2007, pp. 1–6.

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

[12] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *MobiHoc*, 2008, pp. 241–250.

[13] J. Leguay, T. Friedman, and V. Conan, "Evaluating mobility pattern space routing for dtns," in *INFOCOM*, 2006.

[14] M. A. Bayir, M. Demirbas, and N. Eagle, "Discovering spatiotemporal mobility profiles of cell phone users," *WOWMOM*, 2009.

[15] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[16] M. A. Bayir and M. Demirbas, "On the fly learning of mobility profiles for intelligent routing in pocket switched networks," *Technical Report, Deparment of Computer Science and Engineering, University at Buffalo, available at http://www.cse.buffalo.edu/tech-reports/2009-03.pdf*, 2009.

[17] A. Chaintreau, A. Mtibaa, L. Massoulié, and C. Diot, "The diameter of opportunistic mobile networks," in *CoNEXT*, 2007, p. 12.

[18] P. Hui and J. Crowcroft, "Human mobility models and opportunistic communication system design," *Royal Society Philosophical Transactions B*, vol. 366, no. 1872, 2008.

[19] 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.

[20] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks."

[21] Z. J. Haas and T. Small, "A new networking model for biological applications of ad hoc sensor networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 27–40, 2006.

[22] X. Zhang, G. Neglia, J. F. Kurose, and D. F. Towsley, "Performance modeling of epidemic routing," *Computer Networks*, vol. 51, no. 10, pp. 2867–2891, 2007.

[23] 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.

[24] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *ACM/ IEEE MSWIM*, 1999, pp. 53–60.

[25] A. P. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *MOBICOM*, 2003, pp. 217–229.

[26] D. T. et al., "Design and evaluation of an individually simulated mobility model in wireless ad hoc networks," in *CNDSMSC*, 2002.

[27] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," in *ISWC*, 2001, pp. 141–148.

[28] H. Dubois-Ferrière, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *MobiHoc*, 2003, pp. 257–266.

[29] M. Grossglauser and M. Vetterli, "Locating nodes with ease: Mobility diffusion of last encounters in ad hoc networks," in *INFOCOM*, 2003.

[30] K. T. et al., "Shortest path routing in partially connected ad hoc networks," in *GLOBECOM*, vol. 2, 2003, pp. 1038–1042.

[31] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *MobiHoc*, 2007, pp. 32–40.

[32] J. Kleinberg, "Navigation in small world," *Nature*, vol. Nature 406, p. 845, 2000.

[33] J. M. Kleinberg, "The small-world phenomenon: an algorithm perspective," in *STOC*, 2000, pp. 163–170.

[34] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2006.

[35] ——, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. USA*, vol. 103, p. 85778582, 2006.

[36] G. P. et al., "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435(7043), pp. 814 – 818, 2005.

[37] S. Wasserman and K. Faust, Eds., *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[38] P. V. Marsden, "Egocentric and sociocentric measures of network centrality," *Social Networks*, vol. 24, no. 4, pp. 407–422, October 2002. [Online]. Available: http://dx.doi.org/10.1016/S0378-8733(02)00016-3

[39] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, 2004.