Probabilistic Routing Based on Fine-Grained Contact Characterization in Delay Tolerant Networks

Aysha Al-Hinai and Haibo Zhang
Department of Computer Science, University of Otago, New Zealand
Email: {aysha, haibo}@cs.otago.ac.nz

***Abstract*—Most existing history-based routing protocols for delay tolerant networks resort to coarse-grained encounter information for making message forwarding decisions. However, this coarse-grained information can not give precise expression of the contact patterns between nodes in the network, thereby leading to inaccurate forwarding decisions. In this paper, we present FG-PRoPHET, a probabilistic routing scheme based on fine-grained contact characterization. Each node in FG-PRoPHET uses a slotted sliding window mechanism to maintain the information of historical contacts, by which new contact information can be quickly incorporated and outdated history data can be easily removed. The granularity of collected history data can be controlled by adjusting the size of the sliding window to reflect the contact patterns. Based on the fine-grained contact statistics, a greedy forwarding scheme is designed by combining the advantages of contact duration based forwarding and quota-based routing. The performance of FG-PRoPHET was evaluated through extensive simulations, and results show that, compared with existing schemes, FG-PRoPHET can significantly enhance message delivery rate with very low communication overhead.**

I. Introduction

Delay (or Disruption) Tolerant Networks (DTNs) represent a class of networks that may lack continuous network connectivity. They are designed to provide communications in highly stressed environments that suffer from node sparsity and mobility, high bit-error rate, and large or variable delay. Applications of DTNs include large-scale disaster recovery networks, vehicular networks, sensor networks for wildlife tracking. In a DTN, links might appear or disappear without prior alert, and contemporaneous end-to-end connection is hard to maintain. In addition, DTN devices commonly have constraints in terms of memory, bandwidth and power, which force routing protocols to give special attentions in utilizing these limited resources. These special characteristics have inevitably made routing in DTNs a challenging problem [8].

Due to the lack of end-to-end communications, conventional routing schemes based on single-copy forwarding cannot work in DTNs since a single message copy can be easily lost in such dynamic networks. Flooding-based routing is entirely opposite to the single-copy routing scheme. Epidemic [15] is a representative of flooding-based routing protocols, in which a node tries to send a replica of each message its holds to any node it encounters. In large-scale networks, this can lead to extensive consumption of the limited network resources and cause significant communication overhead. Controlled-flooding protocols [13], [9], [10], [2], [6], [5], [14] prevent nodes from unconditionally exchanging messages with other nodes to avoid excessive message exchanges. However, as will be demonstrated in this work, controlled-flooding protocols can still produce too much communication overhead.

In many DTN applications like vehicular networks, the node movements commonly have some patterns. For example, trams always move on a track with a fixed schedule. Hence routing schemes that leverage nodes' contact history information to predict future contacts are highly desirable to deal with such network dynamics. PRoPHET [13], the only DTN routing protocol that has well-defined Internet Research Task Force (IRTF) Draft, exploits information about contact frequency between each pair of nodes to predict the probability that they will encounter in the future. Most existing history-based routing schemes [13], [7], [11], [4] commonly compress all encounter information between a node pair into a coarse-grained piece of information called delivery predictability, and use it for making message forwarding decisions. However, this approach cannot provide good control of the granularity of the contact history, and outdated contact information can not be easily eliminated from the computation of the delivery predictability. Hence, the delivery predictability computed in this approach can not timely reflect the contact patterns between nodes and can lead to making inaccurate message forwarding decisions.

In this paper we present FG-PRoPHET, a new routing protocol that leverages fine-grained contact characterization to enhance message delivery rate without over-consuming the scarce network resources. Specifically, in this paper:

- We identify the major limitations of existing history-based routing protocols designed for DTNs. We demonstrate that controlled flooding protocols such as PRoPHET still suffer from large communication overhead.
- We propose a slotted sliding window mechanism for maintaining and characterizing the historical contact information. With such a mechanism, new contact information can be quickly incorporated and outdated history data can be easily eliminated during the computation of delivery predictability.
- We present a greedy message forwarding scheme which uses contact duration to calculate the delivery predictability, and takes the advantages of quota-based routing to control the communication overhead.
- We evaluate the performance of FG-PRoPHET through extensive simulations and comparisons with some exist-

978-1-4799-0537-9/13/$31.00 ©2013 IEEE

ing schemes. Simulation results demonstrate that FG-PRoPHET can significantly enhance message delivery rate with very low communication overhead.

The reminder of this paper is organized as follows: Section II summarizes the related work. Sections III gives the motivations behind FG-PRoPHET. Section IV describes the key idea of FG-PRoPHET. Sections V and VI present the detailed design of FG-PRoPHET. Section VII gives the performance evaluation. Finally the paper is concluded in Section VIII.

II. Related Work

Some of the existing routing schemes do not exploit the most relevant history data, which might result in inaccurate routing decisions. HEPRA [3], for instance, uses information about the total number of peers a node has encountered in the past to make forwarding decisions, assuming that a node that had encountered a large number of nodes in the past is a good relay for forwarding messages. However, the total number of encountered nodes is not a reliable metric. For example, if node a, which has encountered 30 nodes in the past, needs to send a message to d. Node a then encounters node b, which has met 60 nodes so far but it has never met node d before. HEPRA will make an imprecise forwarding decision by sending the message to node b.

PRoPHETv1 [13] and PRoPHETv2 [7] both exploit information about contact frequency between each pair of nodes for making forwarding decisions. Sometimes a contact can be very short and may not allow a message to be delivered between the encountered peers. These short contacts may affect the accuracy of the delivery predictability if they are taken into account. PRoPHETv2 limits the impact of short contacts on the calculation of delivery predictability by considering the time between two subsequent contacts. If the time between contacts is less than a threshold (i.e. I-TYP), it is considered as an abnormal contact, and the increase caused by this contact should be proportional to the time between the contacts.

E-PRoPHET [11] incorporates both the contact frequency and contact duration to improve the accuracy of the delivery predictability. 3R [16] is a fine-grained routing protocol which leverages the regularity of fine-grained encounter pattern among mobile nodes to maximize message delivery probability while preserving message delivery deadline. It divides the history data gathered from real-life traces into weekend and working days, and further divides a day into several time-slots. The forwarding decisions are made based on the regularity patterns. However, it only permits a single copy of each message in the whole network and this cannot guarantee good performance in challenged network environments as the single copy can easily be lost in such fragile circumstances. 3R is based on contact frequency rather than contact duration. Nodes in 3R require high storage capabilities because it records all interactions of working days and weekends.

III. Motivations

A. Drawbacks of PRoPHET

PRoPHET uses a metric called "delivery predictability" for message forwarding. The delivery predictability for node a to send a message to node b is denoted by $P(a,b)$ where $P(a,b) \in [0,1]$. Each node should compute and maintain $P(a,b)$ for every node b it encounters. Once node a encounters node b, it updates $P(a,b)$ as follows:

$$P(a,b) = P(a,b)_{old} + (1 - \delta - P(a,b)_{old}) * P_{encounter} \quad (1)$$

where $P(a,b)_{old}$ is the corresponding delivery predictability stored in node a before the current contact occurs. $P_{encounter} \in [0,1]$ is a scaling factor at which the predictability increases on encounters, and δ is a small positive number that effectively sets an upper bound for $P(a,b)$.

As shown in Equation (1), the new predictability is always scaled based on the old predictability. A major problem of this approach is that the new contact pattern may not be quickly reflected and the old contact pattern may take some time to vanish. For example, δ is set to 0 and $P_{encounter}$ is set to 0.05. At time t, $P(a,b) = 0.9$ and $P(c,b) = 0.4$. During the period from time t to time t' where the period length is two hours. Node a contacted node b every one hour and node c contacted node b every ten minutes. Based on Equation (1), at time t', $P(a,b) = 0.90975$ and $P(c,b) = 0.67578$. If node d has a message to send to node b, and encounters both a and c. Node a will be chosen as a better forwarder even though node c encountered b more frequently in the past two hours.

PRoPHET uses the following aging mechanism to update $P(a,b)$ if nodes a and b do not encounter during an interval.

$$P(a,b) = P(a,b)_{old} * \gamma^K \quad (2)$$

where $\gamma < 1$ is an aging constant, and K is the number of time intervals elapsed since the last time $P(a,b)$ was aged. However, this mechanism does not solve the above problem. For example, if the aging interval is set to one hour, $P(a,b)$ will not decrease in the past two hours. One may argue that we need to choose the proper settings for the parameters such as $P_{encounter}$ and γ. However, this is not easy in DTNs as the contact patterns among different nodes are not predictable.

Another drawback of PRoPHET is that the delivery predictability is calculated based on contact frequency. Although contact frequency is a more accurate metric compared to the total number of encounters that HEPRA uses, contact frequency itself can also render inaccurate forwarding decisions. Taking Fig. 1 as an example, with black boxes denoting the contacts. As shown in the figure, node b contacts a very frequently as compared to node d. However, the contacts between a and b are very short in duration, and this can be due to repeated attempts to reconnect with each other where transmission problems occur or when physical obstacles cut a normal logical contact into multiple short contacts [12]. Sometimes contacts which last for very short duration may not allow a message to be delivered. These contacts will, therefore, affect the delivery predictability if each contact is counted

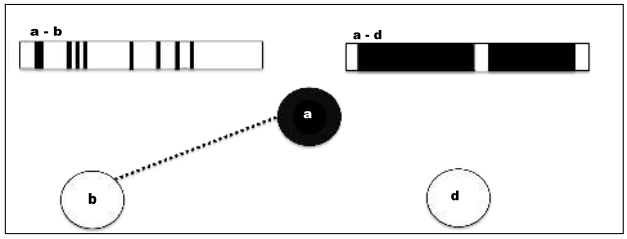

Fig. 1. Using contact frequency to calculate routing metric can sometimes mislead routing

as an individual contact. If a is to choose a next forwarder between b and d in PRoPHETv1, it will choose b as b has a higher contact frequency regardless of the fact that d has had consistently longer and more stable contacts. PRoPHETv2 limits the impact of short contacts on predictability calculation by considering the time between two subsequent contacts. If the time between contacts is less than a threshold (i.e. I-TYP), it is considered as an abnormal contact. However, this does not completely solve the above problem since it is not suitable to choose a global threshold to filter short contacts. In our approach, we will use contact duration instead of contact frequency.

B. Controlled-flooding is Still Not Under Control

Flooding-based protocols such as Epidemic perform blind replication of any message to any encountered node, which may exhaust the network limited resources. PRoPHETv1 and PRoPHETv2 which belong to controlled-flooding history-based schemes control message replication by sending messages only to nodes with better forwarding chances. To investigate the performance of controlled flooding schemes, we run simulations in the ONE simulator [1], which is specially designed for DNTs, to compare Epidemic, PRoPHETv1 and PRoPHETv2. In our simulations, 126 mobile nodes are scattered over the map of Helsinki City (the default map in the ONE simulator), and these nodes are divided into 6 groups with different configurations in terms of moving speed, buffer size and movement model, as shown in Table I. Even though the two "pedestrian" groups use the same movement model, they are configured with different map files. The same for the three "trams" groups. All nodes are configured with Bluetooth interface. For "Trams1", the transmission speed is set to 10M bits/s and the transmission range is set to 1000 meters. For all the other five groups, the transmission speed is set to 500k bits/s and the transmission range is set to 10 meters. We use the default event generator in the ONE simulator to generate messages with intervals between 25s to 35s, and the message size varies from 500k bytes to 1M bytes.

The simulation was run for 12 hours. Fig. 2 compares the three routing schemes in terms of the numbers of (a) *created messages* (excluding replicated messages), (b) *started messages* (equivalent to the total number of message transmissions), (c) *relayed messages*, (d) *aborted messages* (due to connection failure), (e) *dropped messages* (due to buffer overflow), and (f) *delivered messages*. It can be seen that,

Group Type	Buffer	Speed	# nodes	Movement Model
Pedestrian1	20M	0.5-1.5 m/s	40	ShortestPathMap
Pedestrian2	20M	0.5-1.5 m/s	40	ShortestPathMap
Cars	20M	2.7-13.9 m/s	40	ShortestPathMap
Trams1	50M	7-10 m/s	2	MapRoute
Trams2	50M	7-10 m/s	2	MapRoute
Trams3	50M	7-10 m/s	2	MapRoute

TABLE I
NODES SETTINGS IN THE SIMULATIONS

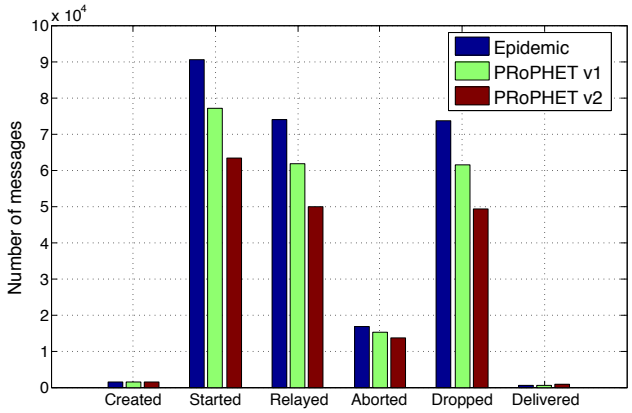

Fig. 2. Comparison between the number of messages that are created, started, relayed, aborted, dropped and delivered

compared with Epidemic, the number of started messages in PRoPHETv1 was reduced by 15.028%, and the number of dropped messages is reduced by 16.559%. Moreover, PRoPHETv2 achieved 30.17% reduction in the number of started messages compared to Epidemic and even 33.08% reduction in the number of dropped messages. However, the number of started, relayed, aborted and dropped messages are still much larger than the number of delivered messages. Even in PRoPHETv2 the number of started messages is more than 85 times of the delivered messages, and more than 98% the started messages are dropped due to either connection failure or buffer overflow. Thus it is essential to design efficient solutions to improve message delivery rate while not introducing much communication overhead and not exhausting the limited network resources.

IV. KEY IDEA BEHIND FG-PROPHET

The key idea of FG-PRoPHET is to build a solid knowledge-based routing scheme that improves message delivery rate with a remarkable balance between gathering fine-grained historical contact information and utilizing the scarce resources of DTN devices. This is achieved by enabling each node to hold the most important historical statistics and building a wise message distribution scheme. FG-PRoPHET has two major functionalities that are responsible for handling message routing and forwarding: (i) history data collection and manipulation, and (ii) delivery likelihood estimation and forwarding decision making. A *time-slotted sliding window* mechanism is used to maintain fine-grained timely statistics of historical contacts. Time in divided into slots, and the

historical contacts are grouped into the corresponding time slots based on the time the contacts occurred. The time-slotted sliding window mechanism automatically incorporates new contact information and remove outdated contact information. By choosing an appropriate size for both the time slots and the siding window based on the network condition, the maintained fine-grained contact information is able to give timely and accurate expressions for the contact patterns between nodes in the network. Based on the fine-grained contact statistics, the contact probability for each encountered peers is computed based on the total duration for all contacts occurred in the sliding window between the encountered peers. Whenever two nodes encounter with each other, they exchange and update the contact probabilities recorded in its local memory. For each message, a fixed number of best forwarders are chosen based on the contact probability, and one message copy is forwarded to an encounter only if it is selected as one of the best forwarders.

V. History Data Collection and Manipulation

The historical contact information is maintained using a slotted sliding window mechanism that consists of two windows: a *short-timing window* (STW) and a *long-timing window* (LTW). The STW records the new contacts occurred in a specified and limited amount of time, and acts as the basic building block for the LTW, whereas the LTW is a large chronological container that contains a set of STWs. The major purpose to introduce the STW is to facilitate the incorporation of new contacts and the removal of outdated contacts. The size of the LTW (i.e., the number of STWs it contains) determines how much history data a node should keep and enforces the flushing of outdated data. The following gives the details about these two windows.

A. Short-Timing Window (STW)

The STW is a data structure stored locally at each node to keep track of the contact statistics between that node and its encountered peers for a short period of time. As shown in Fig. 3, each entry in the STW has the following two fields:

1) **HOST-ID**: which represents the unique identity of the encountered node.
2) **CD**: which represents the total duration for all contacts that occurred with the corresponding HOST-ID within the STW. For instance, node a in Fig. 3 has contacted with b and c for 70 and 159 seconds, respectively.

If two nodes a and b meet for the first time during the STW duration, each of them appends a new entry to its STW, and uses the contact start time $t^{start}_{(a,b)}$ and contact end time $t^{end}_{(a,b)}$ to measure the contact duration. If nodes a and b have previously encountered with each other, they will simply update the corresponding CD field by adding the new contact duration to the old value $CD^{old}_{(a,b)}$, i.e.

$$CD_{(a,b)} = CD^{old}_{(a,b)} + (t^{end}_{(a,b)} - t^{start}_{(a,b)}). \quad (3)$$

The length of the STW is maintained using a timer with the timeout value set to the length of the STW. Once the timer expires, no more updates can be added, and the STW is copied to the LTW. The STW will be cleaned and prepared to collect data for a new STW period.

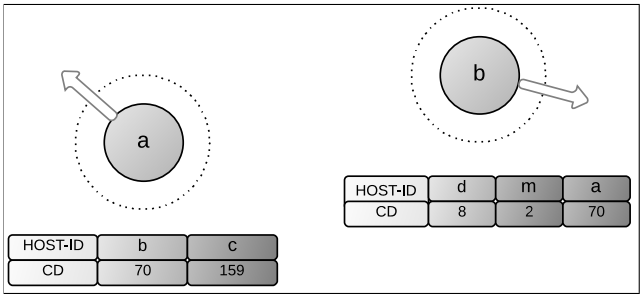

Fig. 3. Use STW to track contacts between nodes for a limited time

B. Long-Timing Window (LTW)

The LTW is basically a container that holds a certain set of STWs. As shown in Fig. 4, each slot in the LTW contains the contact information for one STW duration, which is copied from the STW when the timer is expired. LTW follows the *First-in-first-out* (FIFO) principle to manipulate data entries and dropping during a specified time window. For instance, node a in Fig. 4 maintains a STW with length of 2 hours and a LTW with length of 3 slots. New produced STW will be appended to the rear of the LTW and the oldest STW will be removed from the head of LTW on a FIFO basis.

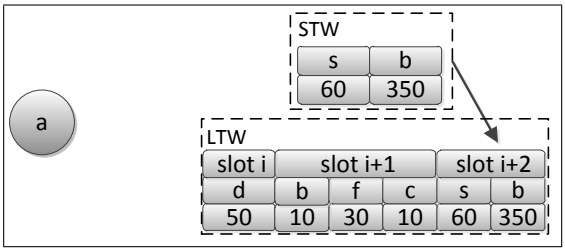

Fig. 4. An example of the STW and LTW maintained at a node

Each node participating in FG-PRoPHET routing stores the history of encountering information in its local memory, and nodes exchange their contact statistics when they meet in order to calculate the message delivery likelihood of the other peers before making any forwarding decision. By controlling the sizes of the LTW and STW, history data collection and manipulation can achieve a good balance between having enough knowledge about network history and utilizing the nodes scarce memory and processing resources.

VI. Delivery Likelihood Estimation and Forwarding Decision Making

Similar to PRoPHET, each node in FG-PRoPHET makes forwarding decisions based on the delivery predicability. The difference is that, instead of using encounter frequency, FG-PRoPHET computes the delivery predicability based on the

fine-grained contact information obtained through the slotted sliding window mechanism.

A. Delivery Predictability Calculation

Before making a forwarding decision, a node needs to compute the delivery predictability to other possibly encountered peers. For example, if node c is carrying a message destined to node b and encounters node a on its way, node c needs to know $P(a, b)$ in order to decide if it should forward the message to node a or not. Node a can compute $P(a, b)$ as follows:

- If the LTW stored in node a is not empty, node a can compute the total duration for all contacts between nodes a and b occurred during the LTW duration, denoted by $CD_{(a,b)}$, by iterating through the LTW, that is,

$$P(a, b) = CD_{(a,b)}/(S_{LTW} * S_{STW}) \quad (4)$$

  where S_{LTW} is the size of the LTW (i.e. number of slots) and S_{STW} is the length of the STW in seconds.
- If the LTW stored in node a is empty (i.e. nodes have just started capturing history information), $P(a, b)$ will be computed using the STW as follows:

$$P(a, b) = CD_{(a,b)}/S_{STW} \quad (5)$$

  Here $CD_{(a,b)}$ is the total contact duration computed based on the STW.

B. Greedy Message Forwarding Scheme

Our greedy forwarding scheme is centered on *network-global knowledge*, which can be obtained by allowing nodes to exchange contact history whenever they meet. However, DTN devices generally have strict constrains on memory and processing power. Thus directly exchanging the collected raw data stored in the LTW is not an efficient and feasible solution. In FG-PRoPHE, all nodes exchange the delivery predictabilities for their encounters instead of the entire LTW. All the delivery predictabilities are recorded in a data structure called *forwarder-list*. Each entry in the forwarder-list contains three pieces of information: the node x that records the actual entry, the node y that x has met and the delivery predictability between both nodes $P(x, y)$. Table II gives an example

p2	t19	t19	t19	t16	t15	t7
t19	c8	p2	p1	c5	w14	c8
0.055	0.0556	0.056	0.022	0.0278	0.056	0.111

TABLE II
AN EXAMPLE FORWARDER-LIST TAKEN FROM THE SIMULATION

forwarder-list stored at node $p2$ in our simulations. The first and second rows represent the nodes that have contacted each other in the past and the third row represents the delivery predictability of the contacts between these nodes. The two columns highlighted in grey color indicate that two nodes $t19$ and $t7$ have encountered node $c8$, and $P(t7, c8)$ is higher that $P(t19, c8)$.

Each node a can update its forwarder-list in the following two cases: (1) If there is any update on the LTW, the forwarder-list also needs to be updated. If there is no existing entry in the forwarder-list, a new one will be appended; otherwise the delivery predictability field of the corresponding entry will be updated to the new value. (2) If node a receives the forwarder-list from another node (e.g. the forwarder-list for $p2$ given in Table II), node a will only update the entries in its forwarder-list that involve $p2$ as a communication peer. This is because that it will take some time for a node to propagate its forwarder list to all the other nodes, and the data in entries that not involve $p2$ might be outdated.

For each message carried by node a, a *best-list* is created by choosing a fixed number of nodes with the highest delivery predictability to the destination of that message. Whenever node a encounters a node b that is in the best-list of the carried message, it will forward a copy of that message to node b. The following gives the steps that two nodes a and b should follow when they come into contact.

1) exchange their forwarder-lists and then update their local forwarders-list accordingly.
2) exchange their Summary Vector (SV) which contains the IDs of the message they carry. Knowing the IDs of messages an encountered node is holding is very essential, because nodes are not supposed to forward messages that they have already carried.
3) If the encountered node b is the destination of that message. Node a transmits that message to node b and removes the message from its memory. The message delivery is successful.
4) If the encountered node b is not the destination of that message, but is in the message's best-list. Node a will produce a copy of that message and forward it to b.

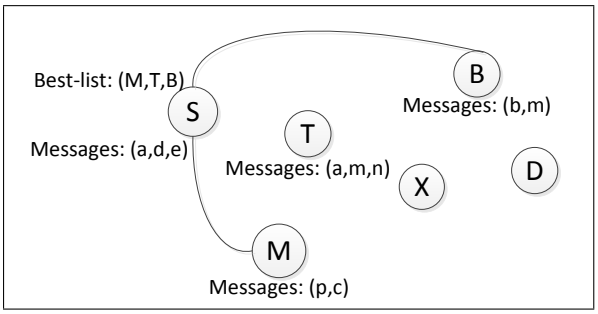

Fig. 5. The Process of Forwarding Decision Making

For example, let's assume that node S in Fig. 5 is carrying a message a destined for D. The best-list for message a contains three forwarders M, T and B. When node S encounters node T, S will exchange the forwarders-list data with T and update its local forwarders-list accordingly. After exchanging the SV, node S knows that T already holds a copy of message a. So S will not forward a message to node T even though it is in its best-list. If node S encounters node X which is not in the message's best-list, S would keep the message

until it encounters a better relay recorded in the best-list. In this example, B and M are in the best list for forwarding a message to node D, and node S will forward a copy of the message to each of them when node b encounters them.

In fact, by incorporating the greedy algorithm into history-based routing, forwarding decisions can be made based on historical knowledge of node interactions, while replications are strictly controlled to safeguard network resources. This provides the advantages of both history based and quota based routing protocols, because it builds forwarding decisions based on the history of nodes interaction and the replications are strictly controlled. The best-list size decides the number of replications per message and thus provides a quota limit to the number of transmissions per message. Only the nodes that historically have strong interactions with a message destination are allowed to handle message forwarding.

VII. Performance Evaluation

In this section we evaluate the performance of FG-PRoPHET through extensive simulations in the ONE simulator. We use the same simulation setup as described in Section III-B, and compare its performance with other existing approaches.

A. Impact of the LTW size

A small LTW may not contain enough history data, whereas a large LTW might contain much outdated history data. Both can lead to inaccurate message forwarding decisions. In this set of simulations, we evaluate the impact of the LTW size on the performance of FG-PRoPHET. The maximum size of each best-list is set to 6, that is, each node can forward a message to at most six nodes. The size of the STW is set to one hour, and the size of the LTW is varied from 2 to 12 hours. Figure 6 shows the average message delivery rate with different LTW sizes. It can be seen that the worst message delivery rate occurred when nodes hold very limited history data. By increasing the LTW size to four hours, the network gives the best performance as nodes can learn better if they are allowed to store more history data and have enough processing power to handle storing and retrieving the history data. However, maintaining a very large LTW can negatively affect the message delivery rate, because it leads to storing out-of-date information that can mislead the forwarding decision making. For example, a node that was a very good forwarder to a destination six hours ago can be a bad forwarder now because it might no longer frequently encounter the destination.

Figure 7 shows the average communication overhead with different LTW sizes. It can be seen that the overhead ratio only slightly varies when changing the length of the LTW. When LTW is set to 4 hours, the network achieves the highest average message delivery rate with the minimum communication overhead. This is because that smaller or larger LTW will lead to inaccurate forwarding decisions, which will produce more message copies as a bad forwarder may further look for other nodes for relaying the message. Thus to achieve a high message delivery rate without wasting the network resources, FG-PRoPHET should be configured with reasonable granularity.

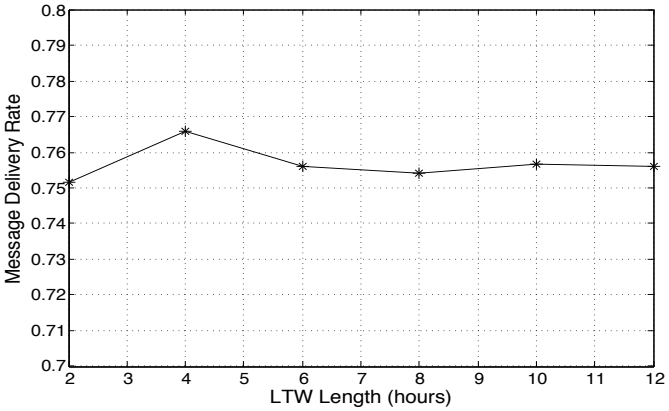

Fig. 6. Impact of LTW size on average message delivery rate

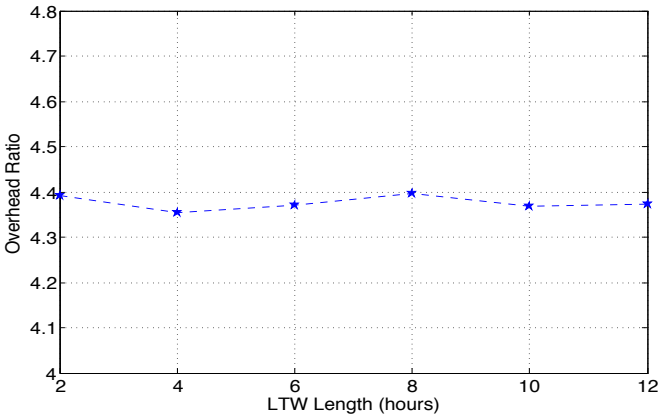

Fig. 7. Impact of LTW size on communication overhead ratio

B. Impact of the best-list size

As described in Section VI-B, the size of a message's best-list dictates the total number of message copies a node can forward. Theoretically the larger the best-list, the higher the message delivery rate. In this set of simulations, we investigate the impact of best-list size on the network performance. The LTW is set to four hours and the size of best-list is varied from 2 to 12. Fig. 8 shows the average message delivery rate with the variation of the size of best-list. It can be seen that involving far less forwarders can result in low message delivery rate. This is because, if only a small number of forwarders are allowed to handle message forwarding, the potential for successful delivery of that message would be low if the forwarders die or the message gets lost. However, increasing the size of best-list does not always enhance message delivery rate. When the size of the best-list is set to 6, the network achieves the highest average message delivery rate. With the further increment of the best-list size, the

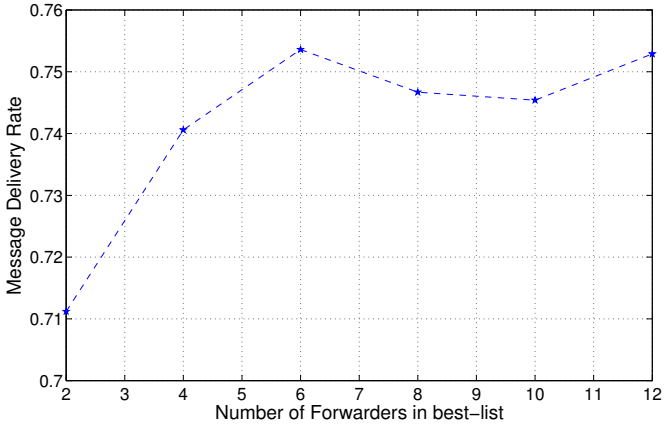

Fig. 8. Impact of best-list size on average message delivery rate

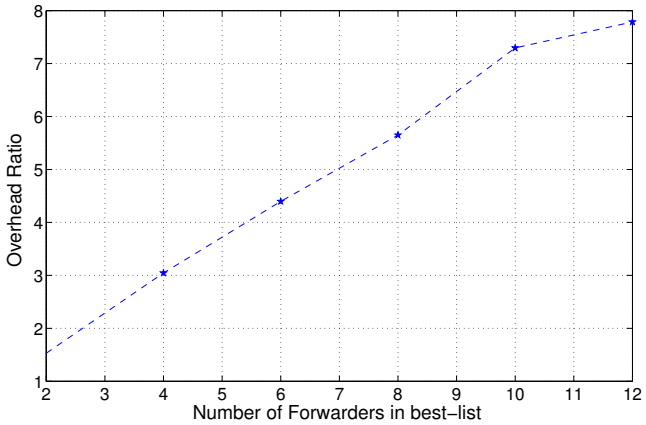

Fig. 10. Impact of best-list size on communication overhead ratio

message delivery rate decreases. This is because that involving more intermediate forwarders increases the number of copies produced per message, resulting in increased buffer usage and increased message dropping rate, as can be seen from Fig. 9. Fig. 10 shows the communication overhead ratio with different sizes of best-list. It can be seen that the overhead ratio always increases with the increase of the size of best-list.

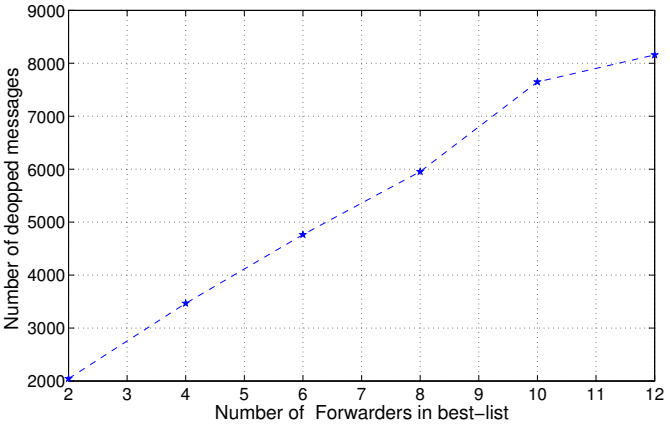

Fig. 9. Impact of best-list size on the number of dropped messages

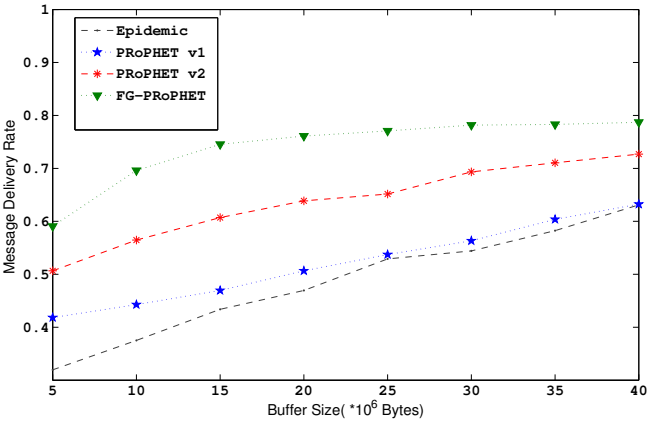

Fig. 11. Buffer size and message delivery rate

C. Comparison with existing solutions

In this set of simulations, the performance of FG-PRoPHET is evaluated in comparison with Epidemic, PRoPHETv1 and PRoPHETv2. We did not compare FG-PRoPHET with the 3R protocol as 3R is not implemented in the ONE simulator, and we leave this as future work. The STW is set to one hour and the LTW is set to four hours. The size of the best-list is set to six.

Fig. 11 shows the average message delivery rate achieved by these four routing schemes under different buffer size. It can be seen that FG-PRoPHET outperforms all the other three schemes in terms of message delivery rate regardless of the buffer size. In average FG-PRoPHET achieves 23.5% improvement compared with PRoPHETv2 and 61.7% improvement compared with PRoPHETv1. This is because FG-PRoPHET makes forwarding decisions based on fine-grained contact duration information. The delivery predictability computed in FG-PRoPHET is much more accurate than that in PRoPHETv1 and PRoPHETv2. Fig. 12 shows the overhead ratio with different setting of buffer size. It is worth noting that FG-PRoPHET has extremely low communication overhead ratio, which is only around 1/16 of that in PRoPHETv2. Moreover, the overhead ratio remains stable regardless of the buffer size. This is because best-list limits the message replication required for a message, unlike Epidemic which speedily drains the nodes buffer because of massive replication. By connecting Fig. 12 with Fig. 11, it is easy to understand why the average message delivery rate in FG-PRoPHET first slightly increases with the increase of buffer size, and then remains roughly stable. With a small buffer size, there will be packets dropped due to buffer overflow. Since the communication overhead ratio in FG-PRoPHET is roughly constant, the average delivery rate remains stable when the buffer is enough to accommodate the messages. However, for the other three protocols, the average message delivery rate increases with the increase of buffer size since the larger the buffer the less packets get

dropped due to the high communication overhead.

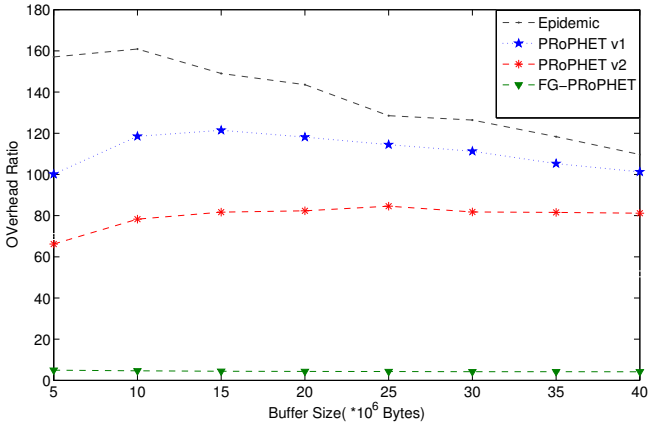

Fig. 12. Buffer size and overhead

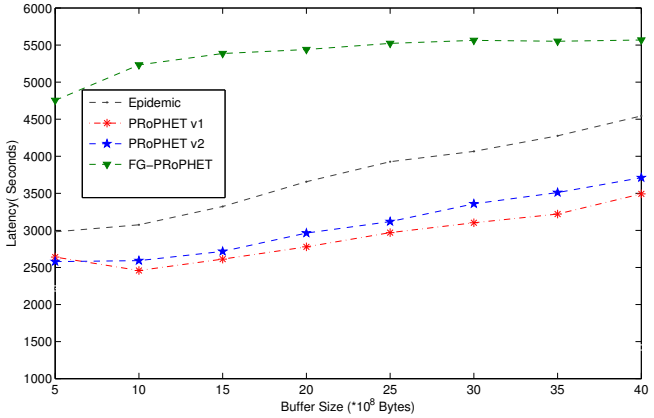

Fig. 13. Buffer size changes latency

Fig. 13 shows the average latency for message delivery with different buffer size. The latency in FG-PRoPHET is larger than the other three protocols as a trade off. This is due to that in FG-PRoPHET each node is allowed to forward a fixed number copies for each message. However, a network that is meant to be delay tolerant should be able to tolerate longer delays.

VIII. Conclusions

This paper presents a novel DTN routing scheme called FG-PRoPHET which uses a sliding window mechanism to maintain fine-grained statistics on historical contacts. FG-PRoPHET has several advantages over the existing investigated routing schemes. Firstly, it collects fine-grained history information and this information gives better accuracy about history compared to coarse-grained information. Secondly, the forwarding decisions in FG-PRoPHET are built on a more robust metric compared to other PRoPHET versions. This metric is contact duration, which is considered to be a good metric because it is robust against frequent disconnections. Thirdly, FG-PRoPHET maintains a reasonable trade-off between high message delivery rate and low message overhead without overwhelming the network's limited resources. This is done through employing the greedy algorithm that sets up a quota limit for the number of transmissions per message.

The performance of FG-PRoPHET is evaluated through extensive simulations and comparisons with other routing schemes. Simulations have shown that FG-PRoPHET considerably outperforms Epidemic and PRoPHET routers by producing more accurate routing decisions, resulting in a significant increase of message delivery rate and better handling of the network buffer resources.

References

[1] The opportunistic network environment simulator. [Online]. Available: http://www.netlab.tkk.fi/tutkimus/dtn/theone/

[2] T. Abdelkader, K. Naik, A. Nayak, N. Goel, and V. Srivastava, "SGBR: A routing protocol for delay tolerant networks using social grouping," 2012.

[3] F. Alnajjar and T. Saadawi, "Performance analysis of routing protocols in delay/disruption tolerant mobile ad hoc networks," in *Proceeding of 10th WSEAS international conference on electronics, hardware, wireless and optical communications*, 2011, pp. 407–417.

[4] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," 2006.

[5] Y. Cao, Z. Sun, N. Ahmad, and H. Cruickshank, "A mobility vector based routing algorithm for delay tolerant networks using history geographic information," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, april 2012, pp. 2757 –2762.

[6] P. C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "GeoDTN+Nav: Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments," *Mob. Netw. Appl.*, vol. 15, pp. 61–82, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11036-009-0181-6

[7] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a dtn routing protocol - prophetv2," in *Proceedings of the 6th ACM workshop on Challenged networks*, 2011, pp. 27–30.

[8] S. Hemal and K. Yogeshwar, "Exploring and exploiting opportunistic network routing in a dtn environment," *International Journal of Future Generation Communication and Networking*, vol. 4, no. 2, pp. 13–22, June 2011.

[9] Y. E. Hui Pan, Crowcroft Jon, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 11 2011.

[10] K. Jahanbakhsh, G. C. Shoja, and V. King, "Social-greedy: a socially-based greedy routing algorithm for delay tolerant networks," in *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, ser. MobiOpp '10. New York, NY, USA: ACM, 2010, pp. 159–162. [Online]. Available: http://doi.acm.org/10.1145/1755743.1755773

[11] Y. Li, X. Li, Q. Liu, and Z. Liu, "E-PROPHET: a novel routing protocol for intermittently connected wireless networks," in *IWCMC'09*, 2009, pp. 452–456.

[12] A. Lindgren, A. Doria, E. Davies, and S. Grasic, *Probabilistic Routing Protocol for Intermittently Connected Networks*, 09th ed., DTN Research Group, April 2011.

[13] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, pp. 19–20, 2003.

[14] Y. Tian and J. Li, "Location-aware routing for delay tolerant networks," in *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on*, aug. 2010, pp. 1 –5.

[15] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Department of Computer Science, Duke University, Tech. Rep., 2000.

[16] L. Vu, Q. Do, and K. Nahrstedt, "3R: Fine-grained encounter-based routing in delay tolerant networks," *A World of Wireless, Mobile and Multimedia Networks, International Symposium on*, pp. 1–6, 2011.