Probability Delegation Forwarding in Delay Tolerant Networks

Xiao Chen
Dept. of Comp. Sci.
Texas State University
San Marcos, TX 78666
xc10@txstate.edu

Jian Shen
Dept. of Math.
Texas State University
San Marcos, TX 78666
js48@txstate.edu

Taylor Groves
Dept. of Comp. Sci.
Texas State University
San Marcos, TX 78666
tg1106@txstate.edu

Jie Wu
Dept. of Comp. Sci. and Eng.
Florida Atlantic University
Boca Raton, FL 33431
jie@cse.fau.edu

***Abstract*—Delay tolerant networks are a type of wireless mobile networks that do not guarantee the existence of a path between a source and a destination at any time. In such a network, one of the critical issues is to reliably deliver data with a low latency. Naive forwarding approaches, such as flooding and its derivatives, make the routing cost (here defined as the number of copies duplicated for a message) very high. Many efforts have been made to reduce the cost while maintaining performance. Recently, an approach called delegation forwarding (DF) caught significant attention in the research community because of its simplicity and good performance. In a network with N nodes, it reduces the cost to $O(\sqrt{N})$ which is better than $O(N)$ in other methods. In this paper, we extend the DF algorithm by putting forward a new scheme called probability delegation forwarding (PDF) that can further reduce the cost to $O(N^{\log_{2+2p}(1+p)}), p \in (0,1)$. Simulation results show that PDF can achieve similar delivery ratio, which is the most important metric in DTNs, as the DF scheme at a lower cost if p is not too small. In addition, we propose the threshold probability delegation forwarding (TPDF) scheme to close the latency gap between the DF and PDF schemes.**

***Index Terms*—delay tolerant networks, forwarding algorithms, routing, traces**

I. INTRODUCTION

Delay tolerant network (DTN) is a type of wireless mobile network that does not guarantee the existence of a path between a source and a destination at any time. When two nodes move within each other's transmission range during a period of time, they *contact* or *meet* each other. When they are out of each other's transmission range, the connection is lost. The message to be delivered needs to be stored in the local buffer. Examples include people carrying mobile devices moving in conferences, university campuses and in social settings. The message delivery in this kind of network is multi-hop and the connection between nodes is *non-predictable*. Furthermore, there is limited knowledge of each node in the network. Besides social settings, this kind of non-predictable DTN has wide applications in environmental monitoring, scientific exploration and military missions.

In such a DTN, the most important metric is the *delivery ratio*, because the network must be able to reliably deliver data. This is the most important metric because of the non-predictable nature of the connections. The second metric is the *delivery latency* [10], which is the time between when a message is sent and when it is received. The third one that attempts to minimize resource consumption such as buffer space or power is the number of *copies* duplicated.

The rudimental routing approach in a non-predictable DTN is flooding [21], which incurs a high cost. Here cost is defined as the number of copies duplicated for a message by a routing protocol. Many algorithms have been put forward to reduce the cost of flooding [2], [5], [6], [11], [15], [18] by forwarding messages to a higher *quality* node that has a better chance to deliver the message to the destination. The quality of a node can be defined by various metrics such as the frequency that a node meets other nodes, the frequency that a node meets the destination, the last contact time of a node with other nodes and the last contact time of a node with the destination, etc.

One approach called *delegation forwarding* (DF) [7] caught significant attention in the research community because of its simple approach and good performance. Its main idea is to assign a quality and a level value to each node. The quality value of a node can be decided using one of the metrics mentioned above. Initially, the level value of each node is equal to its quality value. During the routing process, a message holder compares the quality of the node it meets with its level. It only forwards the message to a node with a higher quality than its level. In addition, the message holder also raises its own level to the quality of the higher quality node. The main difference between DF and the flooding derivatives referenced above is that in the derivatives, a node does not keep a level value and certainly a message holder does not improve its level after it meets a higher quality node and still uses its quality for comparison with other nodes in the rest of the routing while in DF, a node raises its level to the quality of a higher quality node and thus it can only forward the message to a node whose quality is higher than its new level later on. Thus in DF, with the increase of its level, a message holder's forwarding chance is expected to be decreased, which means the number of copies duplicated for a message is expected to be decreased. The authors in [7] show that in an N-node network, delegation forwarding has an expected cost of $O(\sqrt{N})$ while a naive scheme of forwarding to any higher quality node has an expected cost of $O(N)$. In addition, what makes DF appealing is that the performance of

978-1-4244-4581-3/09/$25.00 ©2009 IEEE

DF is as good as other schemes.

In this paper, we show that there is still room to improve DF. We put forward a new scheme called *probability delegation forwarding* (PDF) which can further reduce the cost. Based on DF, our main idea is to insert a probability p into the algorithm. That is, when node u_i meets node u_j with a higher quality than its level, there is a p ($p \in (0,1)$) chance that u_i will forward the message to u_j . The main difference between DF and our algorithm is that under the aforementioned condition, DF will always forward the message while our algorithm may not always forward the message. Analysis shows that using our scheme, the cost will be brought down to $O(N^{\log_{2+2p}(1+p)})$. Simulation results show that PDF can achieve similar delivery ratio as the DF scheme if p is not too small. In addition, we propose another scheme called *threshold-base probability delegation forwarding* (TPDF) to close the latency gap between the DF and PDF schemes.

The rest of the paper is organized as follows: Section II mentions the related work; Section III puts forward the probability delegation forwarding algorithm (PDF); Section IV presents analysis of PDF; Section V shows the simulation results of PDF; Section VI proposes the threshold-based probability delegation forwarding scheme (TPDF); Section VII shows the simulation results of TPDF and the conclusion is drawn in Section VIII.

II. RELATED WORK

Due to the uncertainty and time-varying nature of DTNs, routing poses unique challenges. In the literature, some routing approaches are based on deterministic mobility [8], [9], [12]–[14], [16], [19], [20] while some others are based on non-predictable mobility [2], [5], [6], [11], [15], [18], [21]. Here, we discuss the situation of non-predictable mobility: nodes move dynamically in different directions with different speeds.

If the non-predictable mobility model is used, one rudimental approach for routing is to perform a flooding-based route discovery as in [21] where whenever a host receives a message, it will pass it to all those nodes it can reach directly at that time so that the spread of the message is like the epidemic of a disease. Epidemic routing has the highest performance. However, its cost is too high. Many algorithms have been put forward to reduce the cost [2], [5], [6], [11], [15], [18] by forwarding message only to a higher quality node that is more likely to meet the destination.

Recently, a strategy called delegation forwarding (DF) [7] has been proposed. Its main idea is that each node has an associated quality metric and a level value. A node will forward a message only if it encounters another node whose quality metric is greater than any seen by the message so far. The authors show that despite the simplicity of the strategy, it works surprisingly well. Analysis shows that in an N-node network, delegation forwarding has an expected cost $O(\sqrt{N})$ while the naive scheme of forwarding to any higher quality node has an expected cost $O(N)$. Simulations on real traces show performance as good as other schemes at a much lower cost.

Algorithm DF: Delegation Forwarding

```
Let $u_1, \cdots, u_N$ be nodes
Let $m_1, \cdots, m_M$ be messages
Node $u_i$ has quality $x_{ik}$ and level $\tau_{ik}$ for $m_k$.
INITIALIZE $\forall i, k : \tau_{ik} \leftarrow x_{ik}$
On contact between $u_i$ and node $u_j$:
for $k$ in $1, \cdots, M$ do
  if $m_k$ is currently held by $u_i$ and $\tau_{ik} < x_{jk}$ then
    $\tau_{ik} \leftarrow x_{jk}$
    if $u_j$ does not have $m_k$ then
      forward $m_k$ from $u_i$ to $u_j$
    end if
  end if
end for
```

Algorithm PDF: Probability Delegation Forwarding

```
Let $u_1, \cdots, u_N$ be nodes
Let $m_1, \cdots, m_M$ be messages
Node $u_i$ has quality $x_{ik}$ and level $\tau_{ik}$ for $m_k$.
INITIALIZE $\forall i, k : \tau_{ik} \leftarrow x_{ik}$
On contact between $u_i$ and node $u_j$:
for $k$ in $1, \cdots, M$ do
  if $m_k$ is currently held by $u_i$ and $\tau_{ik} < x_{jk}$ then
    $\tau_{ik} \leftarrow x_{jk}$
    if $u_j$ does not have $m_k$ and $u_i$ is chosen by $p$ then
      forward $m_k$ from $u_i$ to $u_j$
    end if
  end if
end for
```

In this paper, we strive to extend the DF algorithm to further bring down the cost while maintaining similar performance.

III. PROBABILITY DELEGATION FORWARDING (PDF)

In this section, we put forward our algorithm by extending the DF algorithm. In order to see the difference, the details of the DF algorithm and our algorithm are presented close to each other in Algorithms DF and PDF here. In both algorithms, each node is assigned a quality and a level value. Initially, the level of each node is equal to its quality. In the network, multiple messages from multiple sources can be sent to multiple destinations. Without loss of generality, we use one message sent from one source to one destination as an example. The main idea of DF is that when a source has a message to send, it is a message holder. In each hop in DTN, a message holder u_i does not know where the destination is. The only thing it can do is to forward the message to a node u_j with a higher quality than its level hoping u_j can help deliver the message to the destination sooner. If u_i forwards the message to u_j, it also improves its level to the quality of u_j. Now that u_j has the message, it becomes a message holder. In the rest of the routing process, each message holder does the same thing until the destination receives the message. In our algorithm PDF, we believe we can reduce cost even more by simply adding probability p in the DF algorithm. The major difference in our approach is that when a message holder u_i meets a node u_j with a higher quality, there is a p ($p \in (0,1)$) probability that u_i will forward the message to u_j. Since p is a value between $(0,1)$, it is not 100% guaranteed that u_i will forward the message to u_j as in the DF algorithm. This approach does not need global knowledge. Each node decides whether to forward the message or not by itself. It is suitable for a distributed environment such as DTNs.

IV. ANALYSIS

In this section, we compare the costs of the DF and PDF algorithms mathematically. We consider a single message and calculate the number of copies created for each message.

A. Cost of DF

The cost of DF is given in [7]. To make the paper inclusive, we include the idea here. For any node u_i maintaining a quality metric x_i and a level value τ_i, we focus on the gap $g_i = 1-\tau_i$ between the current level and 1. The node that generates the message has an initial level $\tau_i = x_i$. The initial gap $g = 1-x_i$.

Consider a node that updated its gap value n times. The node's current gap is denoted as the random variable G_n. Since nodes meet according to rates that are independent of node quality, the node is equally likely to meet a node with any particular quality value. The next update of the gap occurs when it meets a node with a quality greater than G_n, and all values above this level are equally likely.

Hence, we can write

$$G_{n+1} = G_n \times U, \tag{1}$$

where U is independent of G_n and follows a uniform distribution on $(0,1]$. By induction we then find:

$$E[G_{n+1}|G_n] = \tfrac{G_n}{2}, \text{ hence, } E[G_n] = \tfrac{g}{2^n}.$$

Moreover, from Eq. (1), we see that G_n approximately follows a lognormal distribution (see [3]), with median $\frac{g}{e^n}$. Hence the distribution is highly skewed with most of the probability mass below the mean, and so with large probability we have $G_n \le \frac{g}{2^n}$.

The replication process can be described by a dynamic binary tree T, which contains all the nodes that have a copy of the message. Initially T contains a single node with associated gap g. Each time a node with a copy of the message meets another node having higher quality than any node seen so far, two child nodes are created for the node. Both have an updated gap value. Some branch of the tree will grow faster than others. The total size of the tree represents the upperbound on the number of copies created. We wish to bound the total size of the tree.

We define the set $B = \{i | x_i \ge 1 - \frac{g}{\sqrt{N}}\}$, which we call the *target set*. We will also identify a subtree of the tree T in which children are excluded for nodes having a level above $1 - \frac{g}{\sqrt{N}}$. In other words, all the nodes in the subtree have a gap $< \frac{g}{\sqrt{N}}$. This subtree is called the *target-stopped tree*.

The essential observation is the following: if n is close to $\log_2(\sqrt{N})$, then except with a small probability, a node at generation n in the tree has a gap of at most $\frac{g}{2^n} \le \frac{g}{\sqrt{N}}$. This is because of the highly skewed nature of the distribution of G_n, as described above. Hence, we can safely assume that the target-stopped tree has a depth of at most n. Note that the total number of nodes appearing at generations $0, 1, \cdots, n-1$ is at most $2^n = \sqrt{N}$.

Now we can calculate the total number of copies generated in this process:

$$C_{DF}(n) = 2^n + \tfrac{Ng}{2^n}.$$

In the worst case, g is 1. So,

$$C_{DF}(n) \le C_{WDF}(n) = 2^n + \tfrac{N}{2^n}.$$

The minimum value $\min\_C_{WDF}$ of $2^n + \frac{N}{2^n}$ is obtained by making the two items 2^n and $\frac{N}{2^n}$ equal. That is, $2^n = \frac{N}{2^n}$. Thus, $n = \frac{1}{2}\log_2 N$. So,

$$\min\_C_{WDF} = 2\sqrt{N} = O(\sqrt{N}).$$

B. Cost of PDF

In the PDF algorithm, node i has a p ($p \in (0,1)$) probability to forward the message. For example, if $p = \frac{3}{4}$, then the node has 75% of the chance to forward the message. If the node is not chosen by p, it is equivalent to truncating the subtree from this node in the binary tree. Since the nodes are randomly chosen by the probability p, $E[G_n] = \frac{g}{2^n}$ still holds.

We define the set $B = \{i | x_i \ge 1 - \frac{g}{2^n}\}$ as the target set, and the subtree with all the nodes whose gap $< \frac{g}{2^n}$ as the target-stopped tree.

Now we calculate the total number of copies generated as:

$$C_{PDF}(n) = (1+p)^n + \tfrac{Ng}{2^n}.$$

In the worst case, g is 1. Therefore,

$$C_{PDF}(n) \le C_{WPDF}(n) = (1+p)^n + \frac{N}{2^n}. \tag{2}$$

Now the minimum value $\min\_C_{WPDF}$ of $C_{WPDF}(n)$ can be obtained by making its derivative equal to 0.

$$C'_{WPDF}(n) = (1+p)^n \ln(1+p) - N \cdot 2^{-n} \ln 2 = 0$$

$$\text{So, } (2+2p)^n = \tfrac{N \ln 2}{\ln(1+p)}$$

Then,

$$\begin{aligned} n &= \log_{2+2p} \frac{N \ln 2}{\ln(1+p)} \\ &= \log_{2+2p} N + \log_{2+2p} \ln 2 - \log_{2+2p} \ln(1+p) \end{aligned}$$

So,

$$\begin{gathered} \min\_C_{WPDF} = \\ C_{WPDF}(\log_{2+2p} N + \log_{2+2p} \ln 2 - \log_{2+2p} \ln(1+p)) \\ < C_{WPDF}(\log_{2+2p} N) \end{gathered}$$

If $n = \log_{2+2p} N$, according to Eq. (2),

$$\begin{aligned} C_{WPDF}(n) &= (1+p)^n + \frac{N}{2^n} = 2 \cdot (1+p)^n \\ &= 2 \cdot (1+p)^{\log_{2+2p} N} = 2 \cdot N^{\log_{2+2p}(1+p)} \end{aligned}$$

So,

$$C_{WPDF}(n) = 2 \cdot N^{\log_{2+2p}(1+p)} = O(N^{\log_{2+2p}(1+p)}).$$

Since $p \in (0,1)$, $1 + p < \sqrt{2+2p}$. So $2 \cdot N^{\log_{2+2p}(1+p)} < 2\sqrt{N} = \min\_C_{WDF}$. Therefore, $\min\_C_{WPDF} < \min\_C_{WDF}$. Hence we see that if $p \in (0,1)$, probability delegation forwarding can further reduce the number of copies.

V. SIMULATIONS OF PDF

We conduct simulations to compare DF and PDF. For convenience's sake, in this paper, only a single message is considered in comparison. Actually DF can be treated as a special case of PDF with a probability of 100%. So in the simulations, the results for probability 100% are actually for algorithm DF and the results for probabilities less than 100% are for PDF algorithm with different probabilities.

In our simulations, we use real traces posted on [1]. The data sets consist of contact traces between short-range Bluetooth enabled devices (iMotes [4]) carried by individuals in conference environments, namely Content 2006 and Infocom 2006. In short, we call them Content trace and Info trace. In these traces, the contact of two nodes is independent of the quality of the nodes. That is, each node is equally likely to meet any other node with any quality.

In the simulations, we use three metrics as follows.

- Delivery Ratio: it is the most important network performance metric in DTNs. It is defined as the fraction of generated messages that are correctly delivered to the final destination within a given time period.
- Latency: it is the time between when a message is sent and when it is received. Using the times recorded in traces, we can easily calculate latency. Minimizing latency lowers the time messages spend in the network and reduces contention for resources.
- Copies: it is the number of copies of a message that a protocol generates in routing. It is an approximate measure of the computational resources required, as there is some processing required for each message. It is also an approximate measure of power consumption, and bandwidth and buffer usages as more copies will use more of these resources.

The quality of each node in DF and PDF can be defined differently in different forwarding algorithms as follows:

- Frequency (**Freq**) [6]: Node u_i forwards m_k to node u_j if u_j has more total contacts with all other nodes than does u_i. This algorithm is destination independent.
- Last Contact (**LastContact**) [7]: Node u_i forwards m_k to node u_j if u_j has contacted any node more recently than has u_i. This algorithm is destination independent.
- Destination Frequency (**DestFreq**) [7] : Node u_i forwards m_k to node u_j if u_j has contacted m_k's destination more often than has u_i.
- Destination Last Contact (**DestLastContact**) [5]: Node u_i forwards m_k to node u_j if u_j has contacted m_k's destination more recently than has u_i.

We randomly generate a source and a destination. The probability p should not be too small because in each trace, the observation time is limited. If p is very small, the delivery ratio can be low during the time frame. So we try different probabilities starting from 80% to 100% with an increase step of 5%. If the simulation results can show similar delivery ratio with a much more cost reduction in PDF, there is some merit in the PDF algorithm. For each source and destination pair, under a certain probability, we use all the forwarding algorithms above on both traces. We record delivery ratio, latency and the number of copies used for each set of data. The process is repeated for $10,000$ randomly generated source and destination pairs. The results are averaged and shown in Figs. 1(a), 1(b), 1(c), 1(d), 1(e), and 1(f).

From the results in both traces, we can see that if we use a probability above 80%, the curves in the delivery ratio are almost flat, which means PDF can achieve similar delivery ratio as DF. There is a slight increase in the delivery latency, which means that the latency will increase with the decrease of probability. For the number of copies, we know that DF (probability 100%) uses the most number of copies. Suppose the number of copies used by DF is C_{DF} and the number of copies used by PDF with probability p is C_{PDF}, we calculate ratio $\frac{C_{PDF}}{C_{DF}}$. Since DF is the baseline, its ratio is 100%. As the results in both traces show, more and more copies can be saved with the decrease of probability.

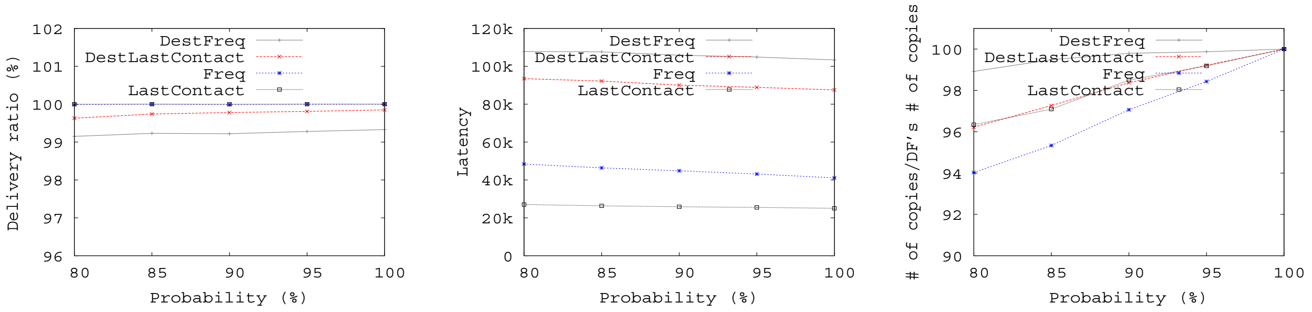
VI. THRESHOLD-BASED PROBABILITY DELEGATION FORWARDING (TPDF)

As we can see from the above simulations, with the decrease of probability, the delivery latency increases. If we use Freq algorithm as an example and look at Fig. 2(b), there is a latency gap between DF and PDF. Our next task is to close the gap between the two. Our main idea is: if node u_i meets node u_j with a much higher quality, that is, if $\frac{x_{jk}-\tau_{ik}}{\tau_{ik}}$ is higher than a certain *threshold* (TH), then without hesitation, node u_i will forward the message to node u_j if u_j does not have the message (see Algorithm TPDF). Otherwise, forward or not will be decided by the probability as in the PDF algorithm. TH is a value which can be set as 0.05 (5%), 0.10 (10%), 0.25 (25%), or 0.50 (50%).

The intuition of this algorithm is that when a node meets a node with a much higher quality, then forwards the message to this node without the decision by the probability, the message will get a higher chance to reach the destination sooner.

VII. SIMULATIONS OF TPDF

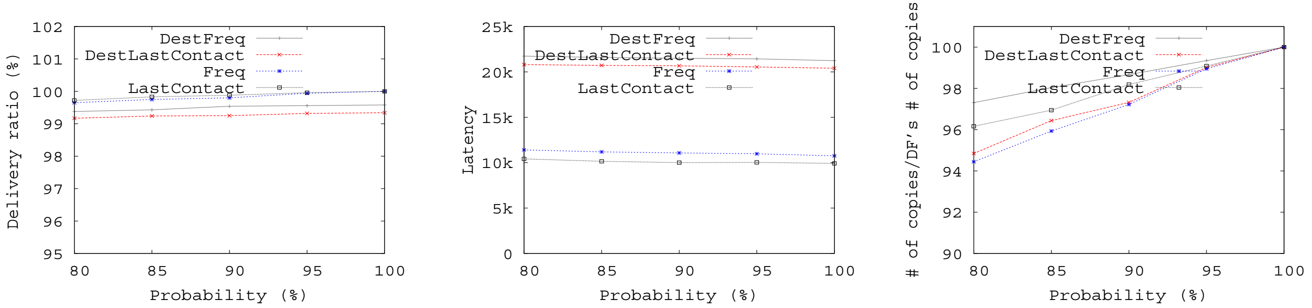
In this section, we conduct simulations to compare TPDF, PDF and DF. DF is PDF with a probability of 100% and PDF is TPDF without the threshold. In our simulations, we set TH



(a) Delivery ratio using Content trace

(b) Delivery latency using Content trace

(c) Ratio of PDF's and DF's copies using Content trace



(d) Delivery ratio using Info trace

(e) Delivery latency using Info trace

(f) Ratio of PDF's and DF's copies using Info trace

Fig. 1. Comparison of DF and PDF using Content and Info traces

Algorithm TPDF: Threshold-based Probability Delegation Forwarding

```
Let $u_1, \cdots, u_N$ be nodes
Let $m_1, \cdots, m_M$ be messages
Node $u_i$ has quality $x_{ik}$ and threshold $\tau_{ik}$ for $m_k$.
INITIALIZE $\forall i, k : \tau_{ik} \leftarrow x_{ik}$
On contact between $u_i$ and node $u_j$:
for $m$ in $1, \cdots, M$ do
  if $m_k$ is currently held by $u_i$ then
    if $\frac{x_{jk} - \tau_{ik}}{\tau_{ik}} > TH$ then
      $\tau_{ik} \leftarrow x_{jk}$
      if $u_j$ does not have $m_k$ then
        forward $m_k$ from $u_i$ to $u_j$
      end if
    else
      if $\tau_{ik} < x_{jk}$ then
        $\tau_{ik} \leftarrow x_{jk}$
        if $u_j$ does not have $m_k$ and $u_i$ is chosen by $p$ then
          forward $m_k$ from $u_i$ to $u_j$
        end if
      end if
    end if else
  end if
end for
```

to be $0.05, 0.1, 0.25,$ and 0.5, and the probability to be 80% for the Content trace and 85% for the Info trace. We still look at the three metrics: delivery ratio, latency and number of copies.

For the delivery ratio, we try Freq, LastContact, DestFreq and DestLastContact algorithms using both traces. The results are shown in Figs. 2(a) and 2(d). From the figures, the delivery ratios of DF, TPDF with TH $= 0.05, 0.1, 0.25,$ and 0.5, and PDF are almost the same, but we can still see that the delivery ratio can get closer to DF's if a threshold is set.

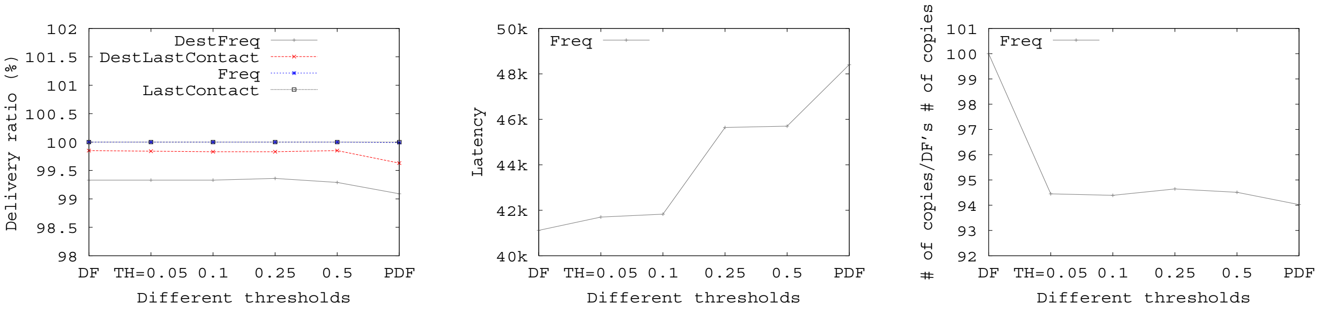
To take a closer look at the delivery latency, we just use Freq and DestLastContact algorithms as examples. In the Freq algorithm, we set the probability to be 80% and use the Content trace while in the DestLastContact algorithm, we set the probability to be 85% and use the Info trace. The results are shown in Figs. 2(b) and 2(e). From the figures, we can see that setting some threshold can bring down latency.

For the number of copies, again we use Freq and DestLastContact algorithms with the same setting. We use DF's copy number C_{DF} as the baseline and calculate ratio $\frac{C_{other\_algorithm}}{C_{DF}}$. The results are in Figs. 2(c) and 2(f). DF has the highest number of copies and PDF has the least. TPDF with some threshold has a copy number between the two.

From the results we know that DF, PDF and TPDF have similar delivery ratio if p is not too small. The selection of a good threshold TH is important to saving more copies at a cost of slight increase in latency. For example, in the Freq algorithm, setting TH=0.10 can decrease the number of copies by 5.9% from PDF at an expense of increasing latency by 1.7% from DF. And in the DestLastContact algorithm, setting TH=0.05 can bring down the number of copies by 3.45% from PDF at a cost of increasing latency by only 0.28% from DF.

VIII. CONCLUSION

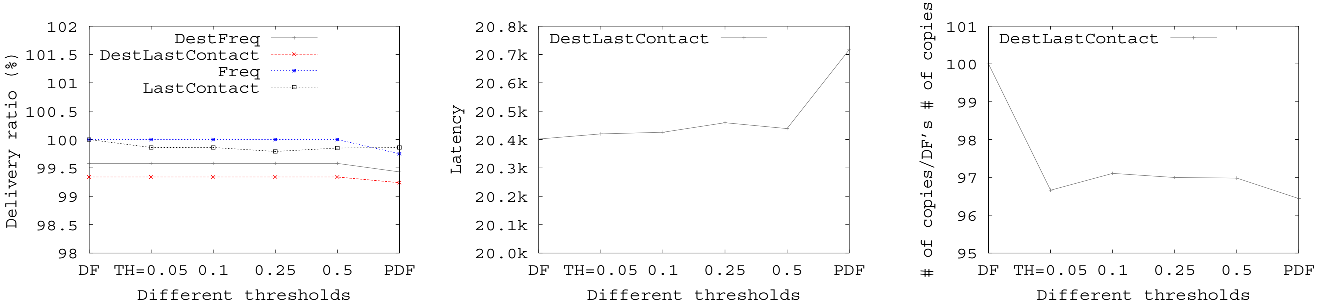
In this paper, we put forward a probability delegation forwarding (PDF) scheme to further reduce the cost in the delegation forwarding (DF) scheme. PDF can achieve similar delivery ratio, which is the most important metric in DTNs,



(a) Delivery ratio using Content trace

(b) Latency of Freq algorithm using Content trace

(c) Ratio of PDF's, TPDF's and DF's copies using Content trace



(d) Delivery ratio using Info trace

(e) Latency of DestLastContact algorithm using Info trace

(f) Ratio of PDF's, TPDF's and DF's copies using Info trace

Fig. 2. Comparison of TPDF, PDF and DF using Content and Info traces

as the DF scheme if p is not too small. The delivery latency in PDF increases a little compared with DF. That can be mutualized by the threshold-based probability delegation forwarding (TPDF) scheme. If a threshold is set properly, TPDF can achieve similar latency as DF at a lower cost.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants CNS 0531410, CNS 0626240, CNS 0835834 and THECB grant ARP 003615-0039-2007.

REFERENCES

[1] CRAWDAD: A community resource for archiving wireless data at dartmoutn. http://crawdad.cs.dartmouth.edu.

[2] X. C. Chen and A. L. Murphy, "Enabling Disconnected Transitive Communication in Mobile Ad Hoc Networks", *Proceedings of the Workshop on Principles of Mobile Computing (POMC)*, August, 2001, pp. 21-27.

[3] A. Broder, A. Kirsh, R. Kumar, M. Mitzenmacher, E. Upfal and S. Vassilvitskii, "The hiring problem and lake wobegon strategies", *Proceedings of ACM-SIAM SODA*, 2008.

[4] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transaction on Mobile Computing 6*, 6 (2007), p. 606-620.

[5] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," *Proceedings of ACM MobiHoc*, 2003.

[6] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, "Diversity of forwarding paths in pocket switched networks", *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, p. 41-50.

[7] V. Erramilli, M. Crovella, A. Chaintreau and C. Diot, "Delegation Forwarding", *Proceedings of ACM MobiHoc*, May 2008, p. 251-259.

[8] J. Ghosh, S. J. Philip, and C. Qiao, "Sociological orbit aware location approximation and routing (SOLAR) in MANET," *Proceedings of ACM MobiHoc*, 2005.

[9] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," *Proceedings of ACM SIGCOMM*, 2004.

[10] E. P. C. Jones and P. A. S. Ward, "Routing Strategies for Delay-Tolerant Networks", *Proceedings of ACM SIGCOMM*, 2004.

[11] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradesoffs and early experiences with zebranet," *Proceedings of ASPLOS-X*, 2002, pp. 96-107.

[12] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," *Proceedings of ACM SIGCOMM Workshop on Delay-Tolerant Networking*, 2005.

[13] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *Lecture Notes in Computer Science*, 3126:239-254, August 2004.

[14] C. Liu and J. Wu, "Routing in a Cyclic MobiSpace," *Proceedings of ACM MobiHoc*, 2008.

[15] C. Liu and J. Wu, "An Optimal Probabilistically Forwarding Protocol in Delay Tolerant Networks," accepted to appear in the *Proceedings of ACM MobiHoc*, 2009.

[16] S. Merugu, M. Ammar, and E. Zegura, "Routing in space and time in network with predictable mobility," *Technical report: GIT-CC-04-07*, College of Computing, Georgia Tech, 2004.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," *Proceedings of IEEE SECON*, 2004, pp. 235-244.

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility", *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW) 2007*, 2007.

[19] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," *Proceedings of ACM MobiHoc*, 2005.

[20] J. Wu, S. Yang, and F. Dai, "Logarithmic store-carry-forward routing in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, 18(6), June 2007.

[21] A. Vahdat, and D. Becker, "Epidemic routing for partially connected ad hoc networks", *Techical Report CS-200006*, Duke University, 2000.