

RESEARCH ARTICLE

Enhanced Delegation Forwarding in Delay Tolerant Networks

Xiao Chen^{a*} and Jian Shen^b and Jie Wu^c

^aDepartment of Computer Science, Texas State University, San Marcos, TX 78666;

^bDepartment of Mathematics, Texas State University, San Marcos, TX 78666;

^cDepartment of Computer and Information Sciences, Temple University, Philadelphia, PA 19122

Delay tolerant network (DTN) is a sparse wireless mobile network that does not guarantee a path between a source and a destination at any time. In DTN, one critical issue is to reliably deliver data with low latency. In an N -node network, naive forwarding approaches such as flooding and its derivatives make the routing cost very high ($O(N)$). Recently, an approach called delegation forwarding (DF) caught significant attention in the research community because it reduces the cost to $O(\sqrt{N})$ while maintaining good performance. In this paper, we enhance the DF algorithm by proposing a probability delegation forwarding (PDF) that further reduces the cost to $O(N^{1+\log_2(1+p)})$, $p \in (0, 1)$ and a threshold-based probability delegation forwarding (TPDF) that closes the latency gap between DF and PDF. Simulation results show that both PDF and TPDF can achieve similar delivery ratio as DF at a lower cost if p is not too small.

Keywords: delay tolerant networks, forwarding algorithms, routing, traces

1. Introduction

Delay tolerant network (DTN) is a sparse wireless mobile network that does not guarantee the existence of a path between a source and a destination at any time. Messages are routed in a store-carry-forward routing paradigm. 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. The node that holds the message is called a *message holder* or a *custodian* of the message. Examples of DTNs 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 performance is measured by three metrics. The first metric is the *delivery ratio*, which is defined as the fraction of generated messages that are correctly delivered to the final destination within a given time period. The second metric is the *delivery latency* [10], which is the time between when a message is sent and when it is received. Delivery ratio is more important than delivery latency in DTN because even under the non-predictable nature of the connections, the

*Corresponding author. Email: xc10@txstate.edu

network still needs to reliably deliver data but longer latency can be tolerated due to the nature of DTN. The third one is the *cost*, which we define as the number of *copies* duplicated for a message. In this paper, our main focus is to seek methods to reduce the cost of routing without affecting too much the delivery ratio and latency.

The rudimental routing approach in a non-predictable DTN is flooding [21], which incurs a high cost. 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] brought significant attention in the research community because it reduces the cost of routing using a simple approach without sacrificing good performance. In this approach, each node has an associated quality metric and a level value. A node u_i will forward a message only if it encounters another node u_j whose quality metric is greater than its level. After the forwarding, node u_i will improve its level to the quality value of node u_j . The authors show that despite the simplicity of the strategy, it works surprisingly well. Analysis finds out that in an N -node network, delegation forwarding has an expected cost of $O(\sqrt{N})$ while the naive scheme of forwarding to any higher quality node has an expected cost of $O(N)$. Simulations on real traces demonstrate that its performance is as good as other schemes at a much lower cost.

In this paper, we believe there is still room to reduce the routing cost. We propose two schemes called *probability delegation forwarding* (PDF) and *threshold-based probability delegation forwarding* (TPDF) to enhance DF. The PDF can further reduce the cost of DF by inserting a probability p into the algorithm. In other words, 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 TPDF combines DF and PDF by introducing a *threshold* TH ($TH \in (0, 1)$). If node u_i meets node u_j with a higher quality than its level, and if that quality is higher than its level by TH , it will use DF; else if the quality is not that high, it will still use PDF. Analysis of the three algorithms shows that both PDF and TPDF can reduce the cost. PDF reduces the cost to $O(N^{\log_{2+2p}(1+p)})$, which is less than $O(\sqrt{N})$ while the cost of TPDF is in the middle of PDF and DF. Simulation results using real traces demonstrate that both PDF and TPDF can achieve similar delivery ratio as the DF scheme if p is not too small. The delivery latency of PDF is higher than that of DF. But the latency gap of the two can be minimized by TPDF.

The rest of the paper is organized as follows: Section 2 mentions the related work, Section 3 puts forward enhanced delegation forwarding algorithms PDF and TPDF, Section 4 presents cost analysis of DF, PDF and TPDF, Section 5 shows the simulation results comparing the three, and the conclusion is drawn in Section 6.

2. 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, it has non-neglectable drawbacks [17]: it consumes a high amount of bandwidth and energy; may result in poor performance because of high contention for shared resources. As the average node degree increases, it is not scalable in terms of memory size needed and number of transmissions performed. 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 [7] has been proposed. It assigns 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 its level with the quality of the node it meets. It only forwards the message to a node with a higher quality than its level. After forwarding, the message holder raises its own level to the quality of the higher quality node. Comparing DF with the flooding derivatives referenced above, we can see that in the derivatives, a node only has a quality value but no level value. And certainly a message holder can not improve its level after it meets a higher quality node. During the whole routing process, it only uses its quality for comparison with other node's quality. However, 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. Thereby 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 its performance is as good as other schemes.

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

3. Enhanced Probability Delegation Forwarding Algorithms

In this section, we put forward two enhanced delegation forwarding algorithms, namely, probability delegation forwarding (PDF) and thresh-hold based probability delegation forwarding (TPDF) algorithms. PDF enhances DF in reducing message copies by inserting a probability in deciding whether to forward or not whereas TPDF combines DF and PDF by introducing a threshold TH .

3.1 Probability Delegation Forwarding (PDF)

Our first enhancement to DF is to insert a probability p in the DF algorithm. That is, if the probability is set as p ($p \in (0, 1)$), and if a node u_i meets a node u_j with a higher quality than its level, u_i will forward the message to u_j with a probability of p (see Algorithm PDF). In other words, it is not 100% forwarding as in the DF algorithm.

This approach does not need global knowledge. Each node decides whether to forward the message or not by itself.

Algorithm DF: Delegation Forwarding

```

1: Let  $u_1, \dots, u_N$  be nodes
2: Let  $m_1, \dots, m_M$  be messages
3: Node  $u_i$  has quality  $x_{ik}$  and level  $\tau_{ik}$  for  $m_k$ .
4: INITIALIZE  $\forall i, k : \tau_{ik} \leftarrow x_{ik}$ 
5: On contact between  $u_i$  and node  $u_j$ :
6: for  $k$  in  $1, \dots, M$  do
7:   if  $m_k$  is currently held by  $u_i$  and  $\tau_{ik} < x_{jk}$  then
8:      $\tau_{ik} \leftarrow x_{jk}$ 
9:     if  $u_j$  does not have  $m_k$  then
10:       forward  $m_k$  from  $u_i$  to  $u_j$ 
11:     end if
12:   end if
13: end for

```

Algorithm PDF: Probability Delegation Forwarding

```

1: Let  $u_1, \dots, u_N$  be nodes
2: Let  $m_1, \dots, m_M$  be messages
3: Node  $u_i$  has quality  $x_{ik}$  and level  $\tau_{ik}$  for  $m_k$ .
4: INITIALIZE  $\forall i, k : \tau_{ik} \leftarrow x_{ik}$ 
5: On contact between  $u_i$  and node  $u_j$ :
6: for  $k$  in  $1, \dots, M$  do
7:   if  $m_k$  is currently held by  $u_i$  and  $\tau_{ik} < x_{jk}$  then
8:      $\tau_{ik} \leftarrow x_{jk}$ 
9:     if  $u_j$  does not have  $m_k$  and  $u_i$  is chosen by  $p$  then
10:       forward  $m_k$  from  $u_i$  to  $u_j$ 
11:     end if
12:   end if
13: end for

```

3.2 Threshold-based Probability Delegation Forwarding (TPDF)

Our second enhancement is to combine DF and PDF by introducing a threshold TH ($TH \in (0, 1)$) (see Algorithm TPDF). Our main idea is: if node u_i with level τ_{ik} meets node u_j with a much higher quality x_{jk} , 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 like that in DF. Otherwise, if $\frac{x_{jk} - \tau_{ik}}{\tau_{ik}} \leq TH$, but $\tau_{ik} < x_{jk}$, forward or not will still 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 if a node meets a node with a much higher quality, then forwarding message to this node without the decision by the probability will help the message to reach the destination sooner.

3.3 Relationships of DF, PDF and TPDF

Algorithms DF, PDF and TPDF are not totally independent. Among the three, the one that can cover others is TPDF. If TH is 0%, TPDF is PDF; if TH is 100%, TPDF is DF. And PDF can cover DF. In PDF, if the probability $p = 100\%$, PDF is DF.

Algorithm TPDF: Threshold-based Probability Delegation Forwarding

```

1: Let  $u_1, \dots, u_N$  be nodes
2: Let  $m_1, \dots, m_M$  be messages
3: Node  $u_i$  has quality  $x_{ik}$  and threshold  $\tau_{ik}$  for  $m_k$ .
4: INITIALIZE  $\forall i, k : \tau_{ik} \leftarrow x_{ik}$ 
5: On contact between  $u_i$  and node  $u_j$ :
6: for  $m$  in  $1, \dots, M$  do
7:   if  $m_k$  is currently held by  $u_i$  then
8:     if  $\frac{x_{jk} - \tau_{ik}}{\tau_{ik}} > TH$  then
9:        $\tau_{ik} \leftarrow x_{jk}$ 
10:      if  $u_j$  does not have  $m_k$  then
11:        forward  $m_k$  from  $u_i$  to  $u_j$ 
12:      end if
13:    else
14:      if  $\tau_{ik} < x_{jk}$  then
15:         $\tau_{ik} \leftarrow x_{jk}$ 
16:        if  $u_j$  does not have  $m_k$  and  $u_i$  is chosen by  $p$  then
17:          forward  $m_k$  from  $u_i$  to  $u_j$ 
18:        end if
19:      end if
20:    end if else
21:  end if
22: end for

```

4. Analysis

In this section, we compare the costs of the DF, PDF and TPDF algorithms mathematically. We consider a single message and calculate the number of copies created for each message in the routing process. The comparisons of the delivery ratio and latency of the three algorithms cannot be easily done mathematically, so we conduct simulations to compare them in Section 5.

4.1 Cost of DF

The cost of DF is given in [7]. To make the paper self 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 updates 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

$$G_{n+1} = G_n \times U, \quad (1)$$

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

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

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 $G_n \leq \frac{g}{2^n}$.

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 $B = \{i|x_i \geq 1 - \frac{g}{\sqrt{N}}\}$, 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 $\frac{g}{2^n} \leq \frac{g}{\sqrt{N}}$. 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 $0, 1, \dots, n-1$ is at most $2^n = \sqrt{N}$.

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

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

In the worst case, g is 1. So,

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

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,

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

4.2 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 $B = \{i|x_i \geq 1 - \frac{g}{2^n}\}$ 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:

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

In the worst case, g is 1. Therefore,

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

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

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

So,

$$(2 + 2p)^n = \frac{N \ln 2}{\ln(1 + p)}$$

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,

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

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,

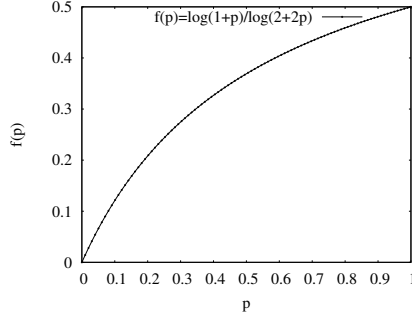
$$\min C_{WPDF} < C_{WPDF}(n) = 2 \cdot N^{\log_{2+2p}(1+p)} = O(N^{\log_{2+2p}(1+p)}).$$

Since $p \in (0, 1)$, $\log_{2+2p}(1 + p) < \frac{1}{2}$ (see Fig. 1).

So,

$$2 \cdot N^{\log_{2+2p}(1+p)} < 2\sqrt{N} = \min C_{WDF}$$

Therefore,

Figure 1. Curve of $f(p)$

$$\min C_{WPDF} < \min C_{WDF}$$

Hence we see that if $p \in (0, 1)$, probability delegation forwarding can further reduce the number of copies.

4.3 Cost of TPDF

In the TPDF algorithm, if a node meets a higher quality node which makes $\frac{x_{jk} - \tau_{ik}}{\tau_{ik}} > TH$, then it will forward immediately if the higher quality node does not have the message, which is equivalent to the DF algorithm; otherwise, if the quality is not that high but still higher than the node's level, forward or not will depend on the PDF algorithm. Therefore, there are two cases here. We assume the probability of picking the DF algorithm is t ($0 \leq t \leq 1$) according to the threshold. So the probability of picking the PDF algorithm is $(1 - t)$. 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, $E[G_n] = \frac{g}{2^n}$ still holds.

We define the set $B = \{i | x_i \geq 1 - \frac{g}{2^n}\}$ as the target set, and the subtree with all the nodes whose gap $< \frac{g}{2^n}$ as the target-stopped tree. Therefore, the total number of copies generated can be calculated as:

$$C_{TPDF}(n) = [t \cdot 2 + (1 - t) \cdot (1 + p)]^n + \frac{Ng}{2^n}$$

In the worst case, $g = 1$. So,

$$C_{TPDF}(n) \leq C_{WTPDF} = [t \cdot 2 + (1 - t) \cdot (1 + p)]^n + \frac{N}{2^n} = [1 + t + p - tp]^n + \frac{N}{2^n} \quad (3)$$

In Eq. (3), if $t = 0$, $C_{WTPDF}(n) = (1 + p)^n + \frac{N}{2^n} = C_{WPDF}$, which is the worst case of PDF. However, if $t = 1$, $C_{WTPDF}(n) = 2^n + \frac{N}{2^n} = C_{WDF}$, which is the worst case of DF. The minimum value $\min C_{TPDF}$ of $C_{TPDF}(n)$ can be obtained by making its derivative equal to 0.

$$C'_{WTPDF}(n) = [1 + t + p - tp]^n \ln[1 + t + p - tp] - N \cdot 2^{-n} \ln 2 = 0$$

So,

$$n = \log_{[2+2t+2p-2tp]} \frac{N \ln 2}{\ln[1+t+p-tp]}$$

Therefore,

$$\begin{aligned} \min \mathcal{C}_{WTPDF} &= [1+t+p-tp]^{\log_{[2+2t+2p-2tp]} N \ln 2 / \ln[1+t+p-tp]} + \frac{N}{2^{\log_{[2+2t+2p-2tp]} N \ln 2 / \ln[1+t+p-tp]}} \\ &= \frac{N \ln 2}{\ln[1+t+p-tp]} \log_{[2+2t+2p-2tp]} [1+t+p-tp] + \frac{N}{\frac{N \ln 2}{\ln[1+t+p-tp]} \log_{[2+2t+2p-2tp]} 2} \end{aligned}$$

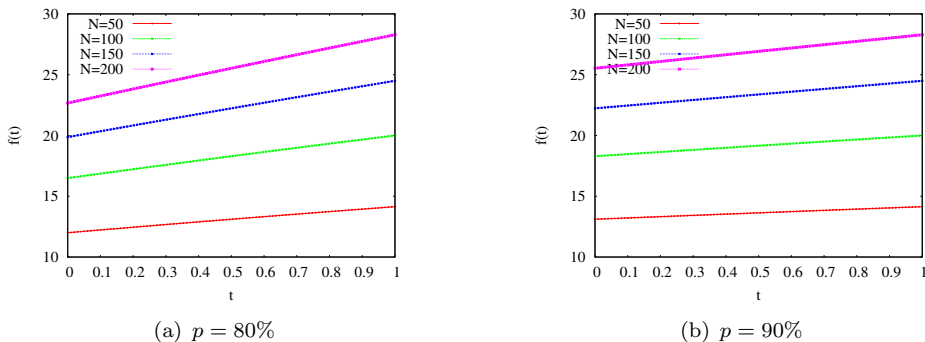


Figure 2. The curves of $f(t) = \min \mathcal{C}_{WTPDF}$ with different N and p

We can see that for a particular probability p , $\min \mathcal{C}_{WTPDF}$ is a function of t . So we denote $\min \mathcal{C}_{WTPDF} = f(t)$ and draw the curve of the function. If we set $p = 80\%$ and 90% and $N = 50, 100, 150$ and 200 respectively, as shown in Figures 2(a) and 2(b), $f(t)$ is a monotonically increasing function. When $t = 0$, it is the case of PDF and when $t = 1$, it is DF . Therefore, the number of copies of TPDF satisfies:

$$\min \mathcal{C}_{WPDF} \leq \min \mathcal{C}_{WTPDF} \leq \min \mathcal{C}_{WDF}$$

5. Simulations

In this section, we conduct simulations to compare the performance of DF, PDF and TPDF. 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 the simulations, we use three metrics as follows.

- **Delivery Ratio:** 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 generated and when it is received. Minimizing latency lowers the time messages spend in the network, reducing contention for resources. So lowering latency indirectly improves delivery ratio.

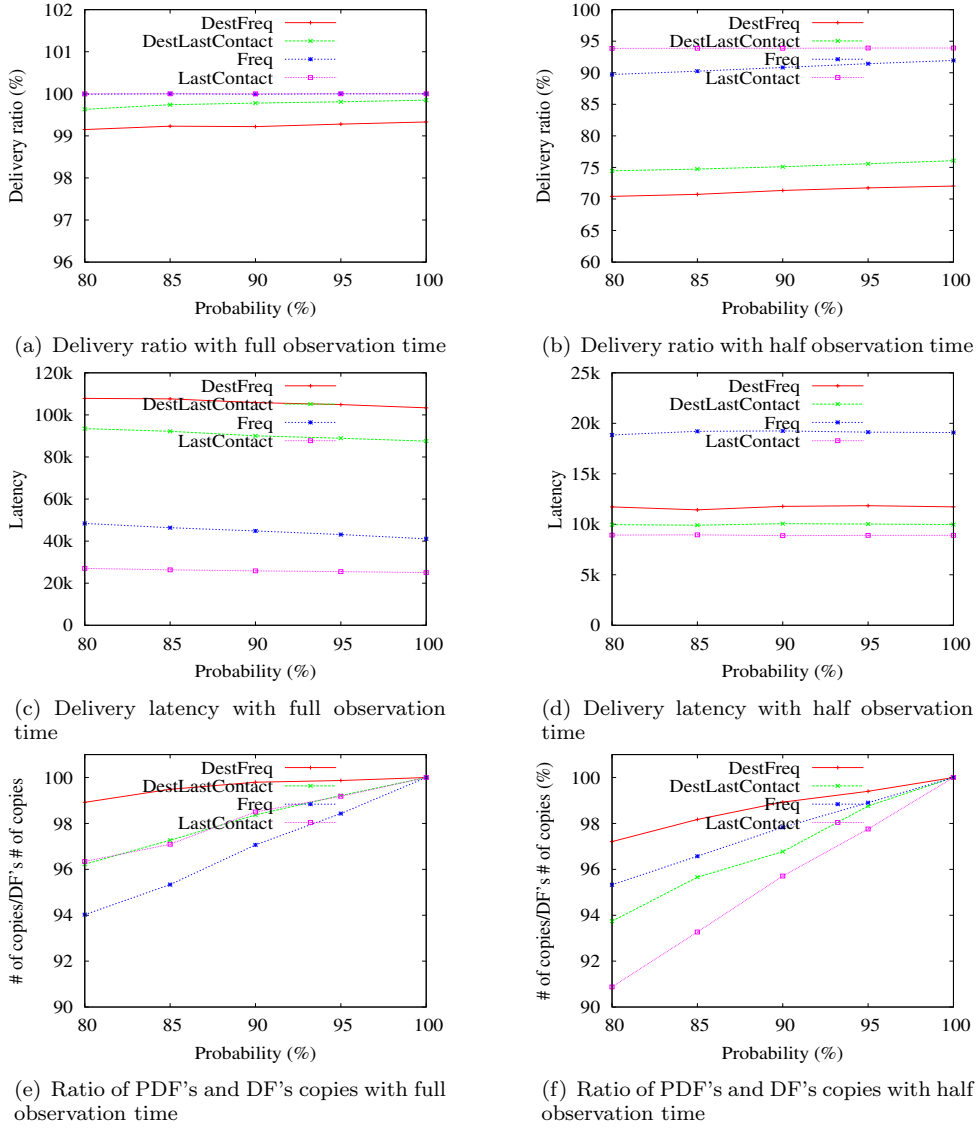


Figure 3. Comparison of DF and PDF using Content trace

- **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. Also it is also an approximate measure of power consumption, bandwidth and buffer usages as more copies will use more of these resources. This is what we call cost in the paper.

The quality of each node in DF and PDF can be decided using 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 .

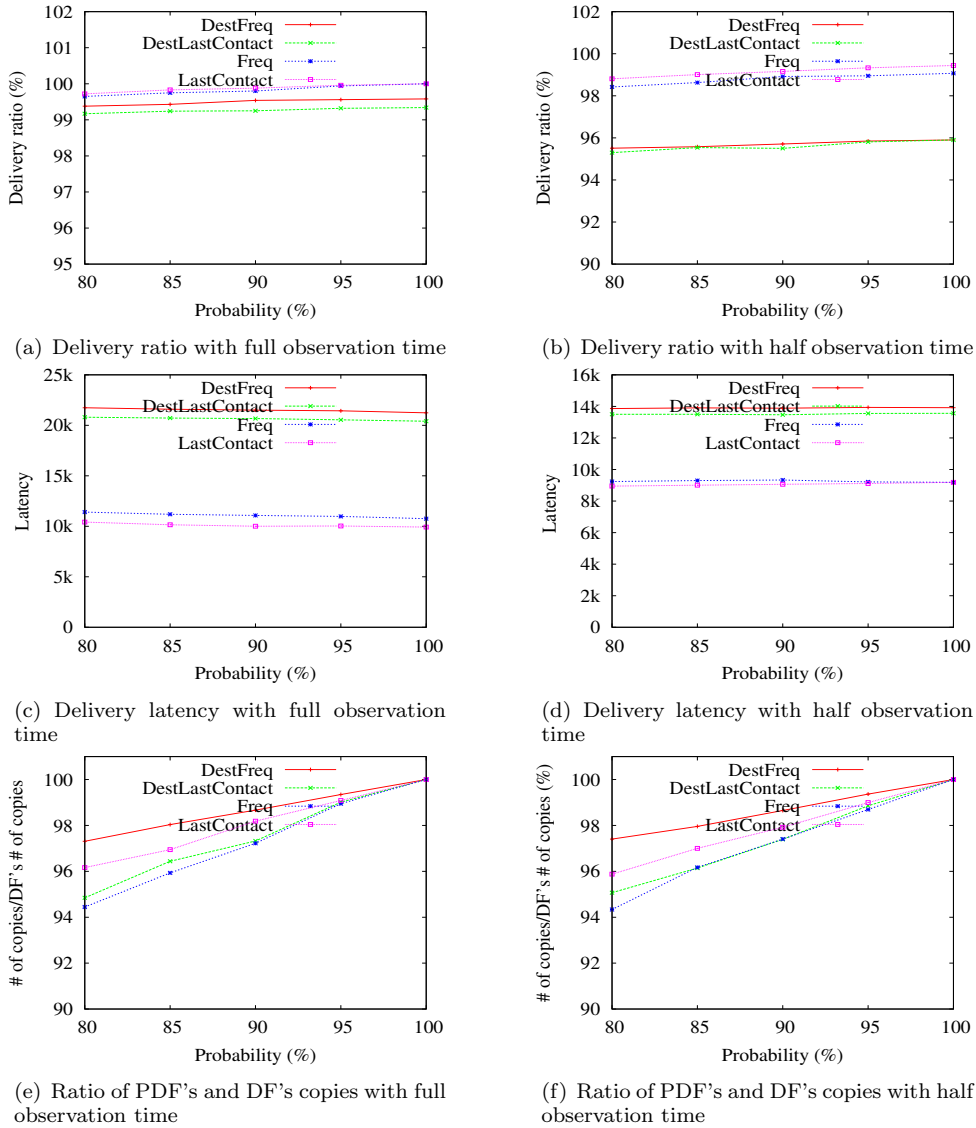


Figure 4. Comparison of DF and PDF using Info trace

5.1 Comparison of DF and PDF

We first compare DF and PDF. 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.

For each of the two traces in the data sets, it has a start time and an end time. If we consider the whole lifespan of each trace, it is called the observation with *full time*. If we just observe a trace in half time, it is the observation with *half time*. The purpose to vary the length of the observation time is to see the performance of the algorithms in different time duration. We randomly generate a source and a destination. We try different probabilities starting from 80% to 100% with an increase step of 5%. We start probability from 80% because we want to save cost but at the same time, we do not want to affect delivery ratio too much. Obviously, if the probability is very small, the delivery ratio will be low during the observation time. 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

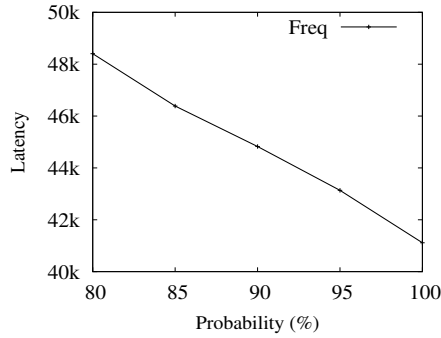


Figure 5. Latency of the Frequency algorithm using Content trace with full observation time

10,000 pairs of randomly generated source and destination pairs. The results are averaged and shown in figures.

Fig. 3 and 4 show the comparison of DF and PDF using the Content trace and Info trace respectively. In each trace, the three subfigures on the left column show the delivery ratio, latency and ratio of PDF's and DF's copies with full observation time, while the three subfigures on the right column present delivery ratio, latency and ratio of PDF's and DF's copies with half observation time. From the results in both traces shown in Fig. 3(a), 3(b), 4(a) and 4(b), we can see that if we use a probability above 80%, the curves in the delivery ratio are almost flat. That means, PDF has similar delivery ratio as that in DF as long as p is not too small. However, there is a slight increase in delivery latency. If we look at the Frequency algorithm using Content trace with full observation time specifically (in Fig. 3(c)) and enlarge its latency graph as in Fig. 5, we can see that DF (with probability 100% in the figure) has the lowest latency, and with the decrease of probability, the latency increases. In other words, the latency can be increased with the decrease of probability.

For the number of copies, we know that DF uses the most number of copies from the analysis. 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 of both traces in Fig. 3(e), 3(f), 4(e) and 4(f) show, more and more copies can be saved with the decrease of probability. This matches our analysis in the previous section.

5.2 Comparison of DF, PDF and TPDF

Next we 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 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 use full observation time for the comparison of the three. 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 Fig. 6(a) and 6(b). From the figures, the delivery ratios of DF and TPDF with $TH = 0.05, 0.1, 0.25$ and 0.5 are almost the same, and PDF's is close to them.

For the delivery latency, to show clearly, 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 Fig. 6(c)

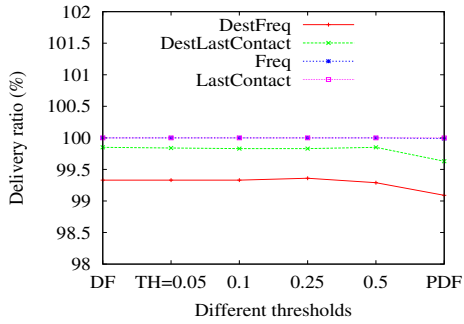
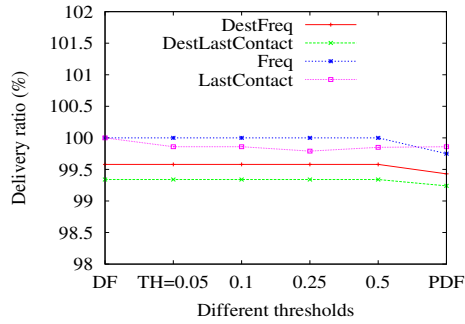
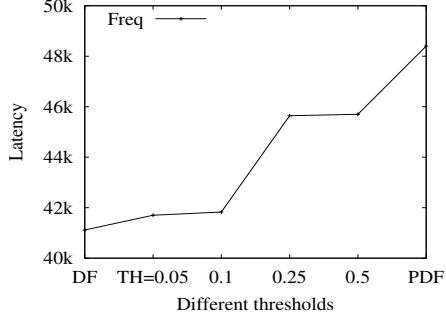
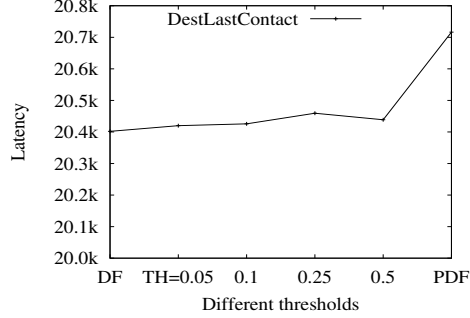
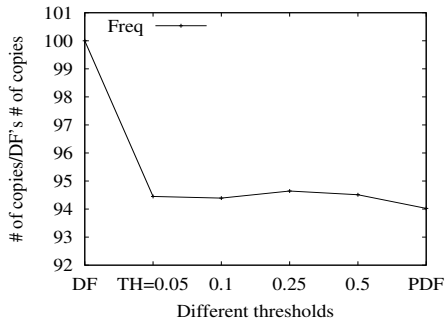
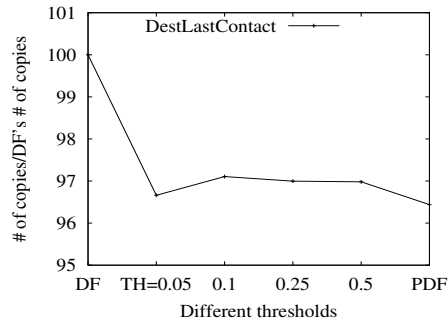
(a) Delivery ratio using Content trace with full observation time ($p = 80\%$)(b) Delivery ratio using Info trace with full observation time ($p = 85\%$)(c) Latency of Freq algorithm using Content trace with full observation time ($p = 80\%$)(d) Latency of DestLastContact algorithm using Info trace with full observation time ($p = 85\%$)(e) Ratio of Freq's and DF's copies using Content trace with full observation time ($p = 80\%$)(f) Ratio of DestLastContact's and DF's copies using Info trace with full observation time ($p = 85\%$)

Figure 6. Comparison of TPDF, PDF and DF

and 6(d). In both figures, PDF has the highest latency and DF has the lowest. TPDF with $TH = 0.05, 0.1, 0.25$ and 0.5 can close the latency gap between PDF and DF.

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 in Fig. 6(e) and 6(f) show that DF has the most number of copies and PDF has the least. TPDF with some threshold has a copy number between the two. This also confirms our analysis result in the previous section.

From the results we know that DF, PDF and TPDF have similar delivery ratio. And 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 as shown in Fig. 6(c) and 6(e), 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 as shown in Fig. 6(d) and 6(f), 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.

6. Conclusion

In this paper, we enhanced the delegation forwarding (DF) scheme by putting forward a probability delegation forwarding (PDF) scheme and a threshold-based probability delegation forwarding (TPDF) scheme. PDF improves DF in reducing message copies by inserting a probability to decide whether to forward or not whereas TPDF combines DF and PDF by introducing a threshold TH . Analysis showed that the cost of PDF is less than that of DF and the cost of TPDF is between the two. Simulations demonstrated that both PDF and TPDF can achieve similar delivery ratio as the DF scheme if p is not too small. The delivery latency in PDF increases a little compared with DF. But that increase can be reduced by the 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 0835834, CNS 0531410 and CNS 0626240.

References

- [1] CRAWDAD: A community resource for archiving wireless data at dartmouth. Available at <http://crawdad.cs.dartmouth.edu>.
- [2] X. C. Chen and A. L. Murphy, *Enabling Disconnected Transitive Communication in Mobile Ad Hoc Networks*, Proc. 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*, Proc. 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*, Proc. of ACM MobiHoc, 2003.
- [6] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, *Diversity of forwarding paths in pocket switched networks*, Proc. of ACM/SIGCOMM IMC, 2007, p. 41-50.
- [7] V. Erramilli, M. Crovella, A. Chaintreau and C. Diot, *Delegation Forwarding*, Proc. of 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*, Proc. of ACM MobiHoc, 2005.
- [9] S. Jain, K. Fall, and R. Patra, *Routing in a delay tolerant network*, Proc. of ACM SIGCOMM, 2004.
- [10] E. P. C. Jones and P. A. S. Ward, *Routing Strategies for Delay-Tolerant Networks*, Proc. 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 tradeoffs and early experiences with zebnet*, Proc. of ASPLOS-X, 2002, pp. 96-107.
- [12] J. Leguay, T. Friedman, and V. Conan, *DTN routing in a mobility pattern space*, Proc. 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*, Proc. of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2008.
- [15] C. Liu and J. Wu, *An Optimal Probabilistically Forwarding Protocol in Delay Tolerant Networks*, Proc. of 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (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*, Proc. of IEEE Secon, 2004.

- [18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, *Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility*, Proc. 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*, Proc. 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*, Technical Report CS-200006, Duke University, 2000.