# Multihop-Delivery-Quality-Based Routing in DTNs

Mingjiong Liang, Zhiguo Zhang, Cong Liu, and Li Chen, *Senior Member, IEEE*

***Abstract*—In *delay-tolerant networks* (DTNs), stable end-to-end connections do not always exist. Messages are forwarded, assisted by the mobility of nodes, in a store–carry–forward paradigm. The mobility of nodes in most DTNs has a certain statistical regularity; thus, using historical information in DTNs to compute the delivery quality of nodes can help to select good forwarding nodes. This paper aims to establish a routing scheme based on multihop delivery quality, which is designed to reduce the energy consumption of message forwarding while maintaining a high delivery rate. We characterized the multihop delivery quality of each node with an expected delay and an expected probability, parameterized by the remaining hop count. Based on these two quality metrics, we developed two algorithms, namely, the *delay-inferred forwarding* (DIF) algorithm and the *probability-inferred forwarding* (PIF) algorithm. The basic idea of DIF and PIF is to find the optimal forwarding path by minimizing the expected delay and by maximizing the expected probability, respectively, in the hop graph that is defined in this paper. We performed extensive trace-driven simulations to compare our algorithm to other representative routing algorithms using several real traces. We observed the following: 1) Compared with the delegation algorithm, which uses one-hop delivery quality, both DIF and PIF significantly improve the message delivery rate, and they yield more improvements as the mobility of nodes becomes more regular; and 2) compared with the state-of-the-art *optimal opportunistic forwarding* (OOF) algorithm, which also uses a multihop delivery quality, DIF and PIF have significantly smaller forwarding overhead (with the maximum reduction in the number of forwarding being over 40%), whereas they are quite close to OOF in terms of both delivery rate and average delay.**

***Index Terms*—Delay-tolerant networks (DTNs), expected delay, expected probability, multihop delivery quality, routing algorithm.**

## I. Introduction

DELAY-TOLERANT networks (DTNs) [1], [2] are sparse mobile networks, where stable end-to-end connections do not always exist. Messages are forwarded, assisted by the mobility of nodes, in a store–carry–forward paradigm. Due to the uncertainty in node contacts, DTNs use opportunistic routing, which forwards multiple copies of a message only to nodes with high delivery quality to improve the delivery rate. The delivery quality of each node is usually calculated assuming long-term regularity in the historical contact information of a DTN. Compared with short-term information (e.g., the time that has elapsed since the last encounter [3]–[6]), long-term regularity is relatively stable over time and able to avoid frequent updates.

To represent the delivery quality, most opportunistic routing schemes use the one-hop quality (e.g., the encounter frequency of a pair of nodes [7]), which has some drawbacks. The most obvious one is that they cannot find multihop good forwarders for a particular destination. Liu and Wu's recent *optimal opportunistic forwarding* (OOF) [8] employs a multihop delivery quality parameterized by the remaining hop count and achieves a significant improvement in performance in terms of delivery rate, with the number of message copies being limited. Currently, battery power duration is the key bottleneck in most mobile devices, and limiting the number of message copies can effectively reduce the power consumption.

In this paper, to further reduce the number of message copies, we introduce two new algorithms to compute the multihop delivery quality based on a hop graph. Here, an expected delay and an expected probability parameterized by the remaining hop count are used to characterize the multihop delivery quality of each node. The expected delay (or expected probability) denotes the time (or probability) for a message to be delivered to destination within a particular remaining hop count. Based on them, two algorithms, called the *delay-inferred forwarding* (DIF) algorithm and the *probability-inferred forwarding* (PIF) algorithm are then developed. The basic idea is to find the optimal forwarding paths by minimizing the expected delay or maximizing the expected probability in a hop graph. The main contributions of this paper can be summarized as follows.

- The time complexity of our algorithms is only $O(HN)$, having a substantial reduction compared with the $O(HN \log N)$ of the OOF scheme (here, $H$ is the maximum hop count of each message, and $N$ is the number of nodes).
- Trace-driven simulations using several real traces are performed, which show that our algorithms significantly reduce the number of message copies, whereas they are quite close to the OOF scheme in terms of the both delivery rate and average delay.

Our proposed algorithms make two assumptions. First, the mobility of nodes has a certain statistical regularity (e.g., some nodes may encounter each other more frequently than others may). Such an assumption is in line with most natural or human-related mobile networks. Second, the storage capacity of each node is large enough to store the mean intermeeting times of all nodes. Since the storage capacity on mobile devices is increasing rapidly, this is a reasonable assumption. Suppose that the number of nodes in the network is 1000 and that the mean intermeeting time of each pair of nodes occupies a storage space of 4 B. Then, our algorithms only require 4 MB

Manuscript received September 28, 2012; revised April 5, 2013, June 13, 2013, and October 23, 2013; accepted December 13, 2013. Date of publication January 9, 2014; date of current version March 10, 2015. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61003241, Grant 61472459, and Grant 61372079, in part by the The Pearl River Tech. Star under Grant 2012009, and in part by the NSF under Grant ECCS 1231461, Grant ECCS 1128209, Grant CNS 1138963, Grant CNS 1065444, and Grant CCF 1028167. The review of this paper was coordinated by Prof. H. Hassanein.

The authors are with the School of Information Science and Technology, Sun Yat-Sen University, Higher Education Mega Centre, Guangzhou 510006, China (e-mail: liangmj@mail2.sysu.edu.cn; lnszzg@mail.sysu.edu.cn; liucong3@mail.sysu.edu.cn; chenli55@mail.sysu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2014.2298407

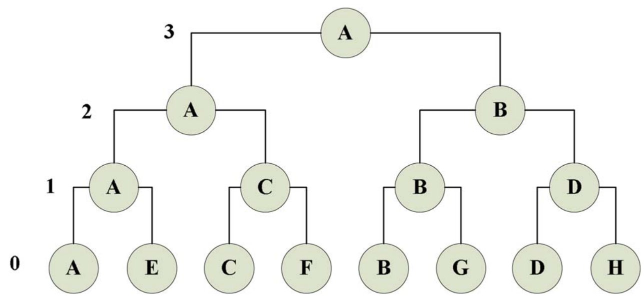



Fig. 1. Hop-count-limited forwarding scheme.

($4 \times 1000 \times 1000$ B) of storage to save the information of all pairs of nodes, which can be easily stored in a typical Secure Digital card with 2 GB of storage.

The remainder of this paper is organized as follows. In Section II, we introduce necessary preliminaries and give an overview of our algorithms (DIF and PIF). Hop graphs are defined and presented in Section III. DIF and PIF algorithms are then presented based on the hop graphs in Sections IV and V, respectively. Simulations and discussions are presented in Section VI, and finally, conclusions are drawn in Section VII.

## II. Preliminaries and Overview

Here, we will first introduce a hop-count-limited forwarding scheme and an expected delay parameterized by the remaining hop count, which are used in the design of our algorithms. Then, we will give a brief overview of our algorithms.

### A. Hop-Count-Limited Forwarding Scheme

In the hop-count-limited forwarding scheme [8], each message holds a value, called the *remaining hop count*. It represents the maximum number of hops that the message can still be forwarded. When a message with a remaining hop count $k$ is forwarded from one node to another, the remaining hop count of both copies in the two nodes becomes $k-1$. When $k=0$, the message cannot be forwarded to any node, except the destination. Therefore, the forwarding history of a message is similar to a full binary tree. An advantage of this forwarding scheme is that it has a constant forwarding cost. Specifically, for a message with initial hop count $H$, the maximum number of copies of the message is $2^H$, which is shown in Fig. 1 for $H=3$. Here, $A, B, C, \ldots, H$ are nodes in the network, and each level shows the nodes holding a message copy and the current remaining hop count after forwarding.

### B. Expected Delay

Most of the routing algorithms only compute one-hop delivery quality, which cannot find multihop good forwarders. For example, although $j$ seldom meets with the destination $d$, $j$ may encounter $m$ in the near future, which has high probability of meeting with $d$. Thus, $j$ is a good forwarder, but it cannot be identified using one-hop delivery quality.

To rectify the aforementioned drawback, an expected delay $D_{i,d,k}$ [8], [9] parameterized by the remaining hop count $k$ is defined in the OOF algorithm. $D_{i,d,k}$ denotes the expected time that it takes to deliver a message from $i$ to destination $d$ with remaining hop count $k$. A smaller expected delay means a higher delivery quality.

The expected delay $D_{i,d,k}$ in OOF is calculated using the method of backward induction. First, the equation for the expected delay is derived as follows:

$$D_{i,d,k} = W_{i,N} \times \left(1 + \sum_{j \in N \backslash \{d\}} \frac{2}{I_{i,j} \times \left(\frac{1}{D_{i,d,k-1}} + \frac{1}{D_{j,d,k-1}}\right)}\right) \tag{1}$$

where $N$ is a set of forwarding nodes of the current message, $I_{i,j}$ is the mean intermeeting time between nodes $i$ and $j$, and $W_{i,N}$ is the average waiting time for $i$ to encounter the first node in $N$. Second, the expected delay $D_{j,d,k-1}, D_{k,d,k-1}, \ldots$ of nodes $j, k, \ldots$ are then sorted in ascending order. Third, $N$ is initialized as an empty set, and then each node from the sorted queue is added into $N$ until $D_{i,d,k}$ reaches its minimum value.

The given analysis shows that the calculation of $D_{i,d,k}$ requires sorting, which has time complexity $O(N \log N)$. Assuming that the maximum value of the remaining hop count is $H$, the OOF algorithm has time complexity of $O(HN \log N)$ to calculate all of its expected delays from 0 to $H$.

### C. Motivation and Overview

In this paper, we aim to establish a routing scheme, which can reduce the number of message copies while maintaining a high delivery rate. To achieve this goal, the key is to pick out the good forwarders and only forward messages to them. Such objective needs delivery quality to be defined. At the same time, the selection of good forwarders and establishment of the forwarding rule are also necessary.

First, we use an expected delay and an expected probability that are parameterized by the remaining hop count to describe the delivery quality of nodes, respectively. These representations can accurately reflect the delivery quality of nodes of a particular hop. In other words, we can find multihop good forwarders using the expected delay or the expected probability.

Second, we establish the forwarding rule by comparing delivery quality (expected delay or expected probability) between different nodes and the destination. Assuming a message of hop $k$, the remaining hop counts of both copies in the two nodes become $k-1$ after forwarding. By comparing the delivery qualities of hops $k$ and $k-1$, we can decide whether to forward a message.

Third, we define a hop graph to calculate the delivery quality. Computing the expected delay or the expected probability is equivalent to finding the shortest path in a hop graph. Our algorithms seek to find the optimal forwarding path by minimizing the expected delay or maximizing the expected probability.

In summary, the main idea is to describe the delivery quality of each node accurately and then to select good forwarders for each message. The expected delay and the expected probability are two alternative ways to represent the delivery quality, and we developed two routing algorithms (DIF and PIF) based on these metrics. The two algorithms have similar steps, and simulations in Section VI will show that they perform well.

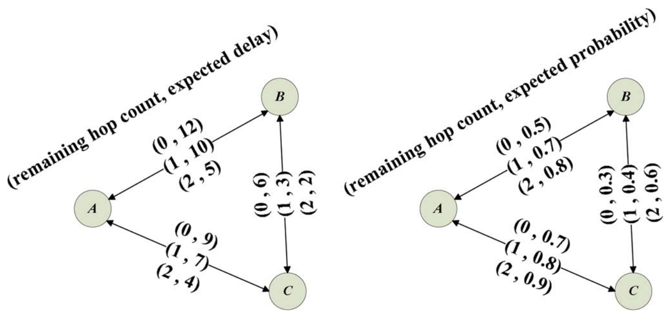



Fig. 2. Delay hop graph and probability hop graph.

## III. Graph Models

We model a DTN as a hop graph $G=(V, E)$, where $V$ is the set of nodes, and $E$ is the set of edges between the nodes. Based on how we define an edge, we can create two different hop graphs: the delay hop graph and the probability hop graph. In a delay hop graph, an edge is a nonempty set of (remaining hop count, expected delay) pairs between a pair of nodes, whereas in a probability hop graph, an edge denotes a nonempty set of (remaining hop count, expected probability) pairs between two nodes.

Fig. 2 shows an example of such graphs, where the graph on the left is a delay hop graph, and the one on the right is a probability hop graph. There are three nodes in each graph, and the maximum remaining hop count is three. The edge between any two nodes is a set containing three elements, each of which has a particular remaining hop count and the expected delay or expected probability between those two nodes associated with that remaining hop count. Each value of expected probability in a probability hop graph ranges between zero and one. For example, when the remaining hop count is zero, the expected delay between node $A$ and node $B$ is 12, and when the remaining hop count is one, the expected probability between them is 0.7.

Based on a delay hop graph, we can calculate the expected delay from one node to another along a particular path. For example, if a message is forwarded along the path from $A$ to $B$ and then to $C$, the expected delay is

$$D\{A \to B \to C\} = d_{A,B} + d_{B,C} = 12 + 6 = 18 \quad (2)$$

where $d_{A,B}$ denotes the value of the delay for remaining hop count 0 for the edge between $A$ and $B$ in the delay hop graph.

Using the multiplication rule of probability, we can calculate the expected probability along a particular path in the probability hop graph. Assuming again that a message is forwarded along the path from $A$ to $B$ and then to $C$, the expected probability is

$$P\{A \to B \to C\} = p_{A,B} \times p_{B,C} = 0.5 \times 0.3 = 0.15 \quad (3)$$

where $p_{A,B}$ denotes the probability value for remaining hop count 0 for the edge between $A$ and $B$ in the probability hop graph.

We need to calculate each expected delay or expected probability value associated with a particular remaining hop count in the graph. We decide whether to forward a message by comparing the values of different edges, which we will discuss in the following.

## IV. Delay-Inferred Forwarding

Here the DIF algorithm is presented, which uses the expected delay $D_{i,d,k}$ to characterize the multihop delivery quality of each node. The proposed algorithm uses the hop-count-limited forwarding scheme to restrict the maximum number of hops of each message, and it seeks to find the optimal message forwarding path by minimizing the expected delay in the delay hop graph.

### *A. Forwarding Rule*

When a message with the remaining hop count $k$ is forwarded from one node to another, the two nodes will both have a message copy, in which their remaining hop counts become $k-1$. If one of the two copies successfully reaches the destination node, the message is considered delivered. Let the expected delay after forwarding be $D$. Then, $D = \min\{D_{i,d,k-1}, D_{j,d,k-1}\}$. Therefore, a message should be forwarded if $D < D_{i,d,k}$, which implies the expected delay will be smaller after forwarding. In fact, when forwarding, we have $D = D_{j,d,k-1}$, which is stated more formally in Theorem 1.

*Theorem 1:* When a message with remaining hop count $k$ is forwarded from node $i$ to node $j$, the expected delay $D$ is exactly equal to $D_{j,d,k-1}$ after forwarding.

*Proof:* For any source $i$ and destination $d$, the expected delays with different remaining hop counts satisfy

$$D_{i,d,k} \le D_{i,d,k-1} \le D_{i,d,k-2} \le \cdots \le D_{i,d,0}. \quad (4)$$

This is because with a higher remaining hop count, there is a higher chance of finding a better forwarding path, and the expected delay will be smaller. If $D = D_{i,d,k-1}$, we have $D = D_{i,d,k-1} < D_{i,d,k}$ according to the given analysis, which contradicts (4). Since $D = \min\{D_{i,d,k-1}, D_{j,d,k-1}\}$, $D = D_{j,d,k-1}$. ■

The steps of forwarding in the DIF algorithm can be listed as follows.

- Let the remaining hop count be $k$ and the expected delay between the node $i$ and the destination node $d$ be $D_{i,d,k}$.
- Suppose that for the remaining hop count $k-1$, the expected delay between the node $j$ (that $i$ meets with) and the destination node $d$ is $D_{j,d,k-1}$.
- If $D_{j,d,k-1} < D_{i,d,k}$, forwarding the message to $j$ will decrease the expected delay $D$; therefore, $i$ will forward the message.
- If $D_{j,d,k-1} \ge D_{i,d,k}$, forwarding the message to $j$ will either increase the expected delay $D$ or keep it the same; therefore, $i$ will not forward the message.

### *B. Calculation of the Expected Delay*

The expected delay $D_{i,d,k}$ of the current hop $k$ can be derived from the last hop. The following analysis first presents the calculation of the expected delay for remaining hop count 0. Then, the expected delay for hop 1 is induced from that for

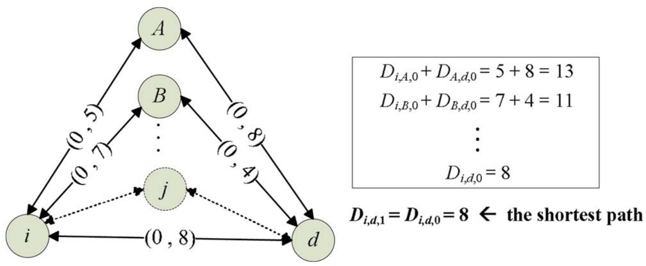



Fig. 3. Calculation of $D_{i,\,d,\,1}$ in the delay hop graph.

hop 0. Finally, the expected delay for any hop $(k > 1)$ is derived using induction.

1) $D_{i,\,d,\,k}$ with $k = 0$

$k = 0$ means that the messages are delivered directly to the destination node, without passing through any relay nodes. Therefore, the expected delay of hop 0 $(D_{i,\,d,\,0})$ is only the expected delay directly between $i$ and $d$. Suppose that the waiting time for the link connecting two nodes follows a uniform distribution i.e., $T \sim U[0,\ I_{i,\,d}]$, where $I_{i,\,d}$ is the mean intermeeting time between nodes $i$ and $d$. Then, according to the definition, we will have

$$D_{i,\,d,\,0} = E(T) = (0 + I_{i,\,d})/2 = I_{i,\,d}/2. \quad (5)$$

2) $D_{i,\,d,\,k}$ with $k = 1$

If $k = 1$, there is one relay node available, which means messages can be first forwarded to the relay node and then be delivered to the destination node. Let the relay node be $j$; then, $D_{i,\,d,\,1} = D_{i,\,j,\,0} + D_{j,\,d,\,0}$. Since $D_{i,\,j,\,0} = I_{i,\,j}/2$, we can rewrite the equation for hop 1 as

$$D_{i,\,d,\,1} = I_{i,\,j}/2 + D_{j,\,d,\,0} \quad (6)$$

which shows a clear calculation of the one-hop delay from the zero-hop delay.

The delay hop graph of hop 1 is shown in Fig. 3, from which we omit some edges for better readability. Node $j$ in the graph represents any relay node, except node $i$ or node $d$, and all expected delays of hop 0 are known. To obtain the minimum value of $D_{i,\,d,\,1}$, we can traverse all the relay nodes, calculate different values using (6), and select the smallest value for $D_{i,\,d,\,1}$. Specifically, $D_{i,\,d,\,1} = \min\{I_{i,\,j}/2 + D_{j,\,d,\,0}\}$.

However, recall (4): $D_{i,\,d,\,1} \le D_{i,\,d,\,0}$. If $\min\{I_{i,\,j}/2 + D_{j,\,d,\,0}\} > D_{i,\,d,\,0}$, the expected delay will become larger if the message is forwarded through any relay node (i.e., the path from $i$ to $j$ and then to $d$ is longer than the one directly from $i$ to $d$ in the graph). In this situation, the optimal solution will be using the direct path and not forwarding the message to any relay node.

According to the given analysis, the calculation process of $D_{i,\,d,\,1}$ is as follows: 1) Initialize $D_{i,\,j,\,1} = D_{i,\,j,\,0}$; 2) for all the relay nodes, compute $I_{i,\,j}/2 + D_{j,\,d,\,0}$; and 3) if $\min\{I_{i,\,j}/2 + D_{j,\,d,\,0}\} < D_{i,\,d,\,1}$, update the value of $D_{i,\,d,\,1}$ using (6).

3) $D_{i,\,d,\,k}$ with $k > 1$

The analysis of $k > 1$ is similar to that of $k = 1$. We use Fig. 4 as an example for discussing the calculation method. Suppose $i$ needs to send a message to $d$, and the expected delay is $D_{i,\,d,\,k}$. If the next hop of $i$ is $j$, we have

$$D_{i,\,d,\,k} = D_{i,\,j,\,0} + D_{j,\,d,\,k-1} = I_{i,\,j}/2 + D_{j,\,d,\,k-1}. \quad (7)$$

$D_{j,\,d,\,k-1}$ is the expected delay of the message delivered from $i$ to $d$ with the remaining hop count $k - 1$. The path between $j$ and $d$ may be multihop, i.e., there may be other nodes between them. However, (7) is a recursive definition, suggesting that there is no need to know the next hop after $j$ or a specific path from $j$ to $d$. We only need to compute the sum of $I_{i,\,j}$ and $D_{j,\,d,\,k-1}$, and then select the optimal node $j$ that minimizes $D_{i,\,d,\,k}$.

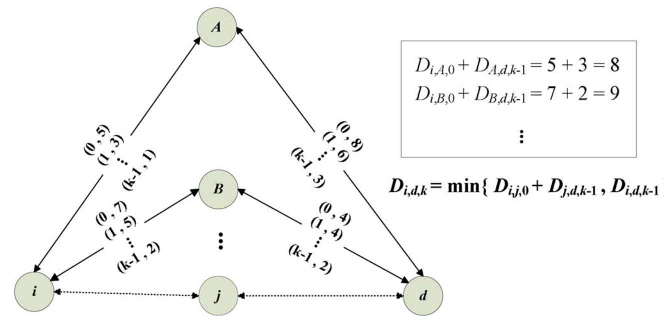



Fig. 4. Calculation of $D_{i,\,d,\,k}$ in the delay hop graph.

Combining the analyses for $k = 1$ and $k > 1$, the specific steps for computing $D_{i,\,d,\,k}$ $(k \ge 1)$ are as follows: 1) Initialize $D_{i,\,j,\,k} = D_{i,\,j,k-1}$; 2) for all the relay nodes, compute $I_{i,\,j}/2 + D_{j,\,d,\,k-1}$; and 3) if $\min\{I_{i,\,j}/2 + D_{j,\,d,\,k-1}\} < D_{i,\,d,\,k}$, update the value of $D_{i,\,d,\,k}$ $(D_{i,\,d,\,k} = \min\{I_{i,\,j}/2 + D_{j,\,d,\,k-1}\})$.

**Algorithm 1** Calculation of $D_{i,\,d,\,k}$ in DIF

```
N ← the number of nodes
I_{i,j} ← the mean intermeeting time of node i and j
Initialize D_min = D_{i,d,k-1}
for j in 1, ..., N do
    if j ≠ i and j ≠ d then
        if I_{i,j}/2 + D_{j,d,k-1} < D_min then
            D_min = I_{i,j}/2 + D_{j,d,k-1}
        end if
    end if
end for
D_{i,d,k} = D_min
```

DIF first initializes $D_{i,\,d,\,k}$ to $D_{i,\,d,\,k-1}$ and then finds a smaller value of $I_{i,\,j}/2 + D_{j,\,d,\,k-1}$ in the node set if there exists one. Therefore, the time complexity for calculating $D_{i,\,d,\,k}$ for a particular hop $k$ is $O(N)$. Supposing that the maximum value of the remaining hop count is $H$, the DIF algorithm has time complexity of $O(HN)$ for calculating expected delays from 0 to $H$. In other words, the DIF algorithm need to establish a routing table for each node that contains the expected delays from hop 0 to $H$. Since the computational complexity for each hop is $O(N)$, the establishment of the whole routing table requires a time complexity of $O(HN)$. Compared with the $O(HN \log N)$ for OOF, the DIF algorithm has lower computational complexity.

### C. Cost of DIF

According to (4), for any source node $i$ and destination node $d$, the expected delays of different hops satisfy

$$D_k \leq D_{k-1} \leq D_{k-2} \leq \cdots \leq D_0 \tag{8}$$

where only the hop count parameter is reserved for better readability.

Normalizing by $D_0$, we have $D_k \in (0, 1]$ for any $k$ according to (8). Thus, we can further rewrite (8) with the upper bound being 1 as follows:

$$D_k \leq D_{k-1} \leq D_{k-2} \leq \cdots \leq 1. \tag{9}$$

Let the gap between $D_0$ and $D_1$ be $G_0$; therefore, $G_0 \in [0, 1)$. Consider a node that has updated its gap $n$ times. The node's current gap is represented by the random variable $G_n$. Since the gap is updated at random, we can write

$$G_{n+1} = G_n \times U \tag{10}$$

where $U$ is independent of $G_n$ and follows a uniform distribution in the interval [0, 1). We have

$$E[G_{n+1}|G_n] = G_n/2. \tag{11}$$

Therefore, by induction, $E[G_n] = G_0/2^n$.

Equation (10) implies that $G_n$ approximately follows a lognormal distribution. Thus, the distribution of $G_n$ is highly skewed with most of the probability mass below the mean, and we can have $G_n \leq E[G_n] = G_0/2^n$ with large probability [10].

According to the DIF forwarding rule, we define the target set $B = \{i|1 - D_i \leq G_0/\sqrt{N}\}$, where $N$ denotes the number of nodes in the network. This set contains all the message copies generated from hops $n$, $n+1, \ldots$, and so on [10]. In addition, the generation of message copies from hop 0 to hop $n-1$ can be regarded as the creation process of a dynamic tree, in which the nodes all have a gap above the threshold value $G_0/\sqrt{N}$. Due to the highly skewed feature of the distribution of $G_n$ described earlier, the gap of the node at generation $n$ in the tree is at most $G_0/2^n \leq G_0/\sqrt{N}$ with large probability when $n$ approximates to $\log \sqrt{N}$. In other words, the maximum depth of the tree is $n$, which satisfies $2^n \approx \sqrt{N}$ [10].

Therefore, the number of all of the message copies generated from all hops can be calculated as

$$C_{\text{DIF}}(G_0) = 2^n + |B| \approx \sqrt{N} + |B| \tag{12}$$

where $C_{\text{DIF}}$ denotes the total number of message copies in the DIF.

Now, we need to bound the size of the target set $B$. Let the variable $g$ denote the gap between 1 and $D_1$ (i.e., $g = 1 - D_1$) for any node, and we define a set parameterized by a threshold $t$ $(t \in [0, 1])$ as $B'(t) = \{g|g \leq t\}$, where the set contains all the nodes with $g$ less than the threshold $t$. Since $0 < D_1 \leq 1$, for any node, the value of $g$ randomly ranges from 0 to 1. Hence

$$E\left[|B'(t)|\right] = t \times N. \tag{13}$$

For example, if $t$ is set to 1, $E[|B'(1)|] = N$, and if $t$ is set to 0.5, $E[|B'(0.5)|] = N/2$.

The definition of the target set $B$ and (9) imply that any element in $B$ has to satisfy

$$1 - D_1 \leq \frac{G_0}{\sqrt{N}} \Longrightarrow g \leq \frac{G_0}{\sqrt{N}}. \tag{14}$$

This is, by definition, the set $B'(G_0/\sqrt{N})$. That is to say, the elements in the set $B$ are all in the set $B'(G_0/\sqrt{N})$. Therefore

$$|B| \leq E\left[\left|B'\left(\frac{G_0}{\sqrt{N}}\right)\right|\right] = \frac{G_0}{\sqrt{N}} \times N = G_0 \times \sqrt{N}. \tag{15}$$

From (12) and (15), we have

$$C_{\text{DIF}}(G_0) \approx \sqrt{N} + |B| \leq (1 + G_0) \times \sqrt{N}. \tag{16}$$

Since $G_0 \in [0, 1)$, $C_{\text{DIF}}$ can be further bounded to

$$C_{\text{DIF}} < 2 \times \sqrt{N}. \tag{17}$$

Recall that the maximum number of message copies is $2^H$ in our hop-count-limited forwarding scheme. Then, we have

$$C_{\text{DIF}} \leq 2^H \tag{18}$$

where $H$ is usually set to 3 (i.e., $H = 3$).

From (17) and (18), $C_{\text{DIF}}$ can be derived as

$$C_{\text{DIF}} \leq \min\{2 \times \sqrt{N}, 2^H\}. \tag{19}$$

## V. Probability-Inferred Forwarding

Here, we will define the expected probability $P_{i,d,k}$ to describe the multihop delivery quality of each node and present the PIF algorithm using this expected probability. The steps of the PIF algorithm are similar to that of DIF. Instead of minimizing the expected delay as in DIF, PIF finds the optimal forwarding path by maximizing the expected probability in the probability hop graph.

**Algorithm 2** Calculation of $P_{i,d,k}$ in PIF

```
N ← the number of nodes
I_{i,j} ← the mean intermeeting time of node i and j
M_{i,j} ← the meeting probability of node i and j
T ← the time slot width
Initialize P_max = P_{i,d,k-1}
for j in 1, ..., N do
    if j ≠ i and j ≠ d then
        M_{i,j} = 1 − exp(−T/I_{i,j})
        if M_{i,j} × P_{j,d,k-1} > P_max then
            P_max = M_{i,j} × P_{j,d,k-1}
        end if
    end if
end for
P_{i,d,k} = P_max
```

### A. Forwarding Rule

We first define the expected probability $P_{i,\,d,\,k}$, which also has the parameter of the remaining hop count to characterize the multihop delivery quality of each node. $P_{i,\,d,\,k}$ denotes the expected probability for a message delivered from $i$ to $d$ with the remaining hop count $k$. When a message with the remaining hop count $k$ is forwarded from one node to another, the two nodes will both have a copy, in which the remaining hop count becomes $k-1$. If one of the two copies successfully reaches the destination node, the message is considered delivered. Let the expected probability after forwarding be $P$; we can have $P = \max\{P_{i,\,d,\,k-1}, P_{j,\,d,\,k-1}\}$. Therefore, a message should be forwarded if $P > P_{i,\,d,\,k}$, which means that the expected probability will be larger after forwarding. In fact, $P = P_{j,\,d,\,k-1}$, the proof of which is similar to the proof in DIF and, thus, is not presented here.

The steps of forwarding in the PIF algorithm and thus can be listed as follows.

- Let the remaining hop count be $k$ and the expected probability between the node $i$ and the destination node $d$ be $P_{i,\,d,\,k}$.
- Suppose that with the remaining hop count $k-1$, the expected probability between the node $j$ (that $i$ meets with) and the destination node $d$ is $P_{j,\,d,\,k-1}$.
- If $P_{j,\,d,\,k-1} > P_{i,\,d,\,k}$, forwarding the message to $j$ will increase the expected probability; therefore, $i$ will forward the message.
- If $P_{j,\,d,\,k-1} \le P_{i,\,d,\,k}$, forwarding the message to $j$ will decrease the expected probability or keep it the same; therefore, $i$ will not forward the message.

### B. Calculation of Expected Probability

First, we define a variable $M_{i,\,j}$, which denotes the encounter probability of nodes $i$ and $j$. Assuming an exponential intermeeting time, we can estimate the $M_{i,\,j}$ [4] by

$$M_{i,\,j} = 1 - \exp(-T/I_{i,\,j}) \tag{20}$$

where $T$ is the residual time to live (TTL) of the message, and $I_{i,\,j}$ is the mean intermeeting time between nodes $i$ and $j$. Since we are only concerned with making relative comparisons between different pairs of nodes, we can set $T = 1$ for simplicity.

The calculation of the expected probability $P_{i,\,d,\,k}$ with the current hop count $k$ also uses the induction method. The following analysis first presents the calculation of the expected probability with the remaining hop count 0 and then generalizes to the calculation of the expected probability of a particular hop $k$ $(k \ge 1)$.

1) $P_{i,\,d,\,k}$ with $k = 0$

When $k=0$, the expected probability is $P_{i,d,0}$. If $i$ encounters $d$, messages can be forwarded. Thus, $P_{i,d,0}$ is just the encounter probability of $i$ and $d$. Using (20), we have

$$P_{i,\,d,\,0} = M_{i,\,d} = 1 - \exp(-T/I_{i,\,d}). \tag{21}$$

2) $P_{i,\,d,\,k}$ **with** $k \ge 1$

We calculate $P_{i,\,d,\,k}$ $(k \ge 1)$ using induction, as we did for $D_{i,\,d,\,k}$ in DIF.

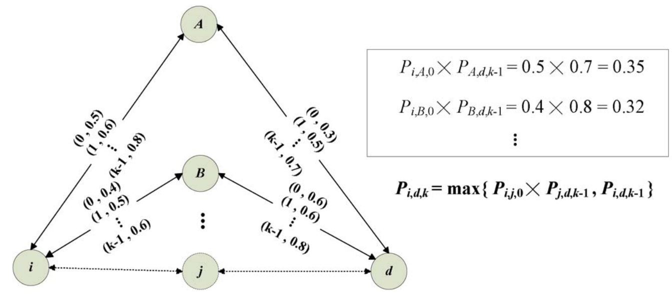



Fig. 5. Calculation of $P_{i,\,d,\,k}$ in the probability hop graph.

TABLE I
SIMULATION SETTINGS

| Parameter name | Value |
|---|---|
| Tickets in Spray-and-Wait ($L$) | 8 |
| Initial hop count ($k$) | 3 |
| Number of messages | 40,000 |

Suppose $i$ needs to send a message to $d$ and the expected probability is $P_{i,\,d,\,k}$. If the next hop is $j$, we can have

$$\begin{aligned} P_{i,\,d,\,k} &= P\{i \to j \cap j \to d\} \\ &= P\{i \to j\} \times P\{j \to d | i \to j\} \\ &= M_{i,\,j} \times P_{j,\,d,\,k-1} \end{aligned} \tag{22}$$

where $P\{i \to j\}$ represents the expected probability of messages delivered from $i$ to $j$.

As shown in Fig. 5, the path between $j$ and $d$ may be multihop, i.e., there may be other nodes between them. However, (22) is a recursive definition, suggesting that there is no need to know the next hop after $j$ or a specific path from $j$ to $d$. We only need to compute the product of $M_{i,\,j}$ and $P_{j,\,d,\,k-1}$ and then select the optimal node $j$ that maximizes $P_{i,\,d,\,k}$.

The computational steps of $P_{i,\,d,\,k}$ $(k \ge 1)$ are as follows: 1) Initialize $P_{i,\,j,\,k} = P_{i,\,j,\,k-1}$; 2) for all the relay nodes, compute $\max\{M_{i,\,j} \times P_{j,\,d,\,k-1}\}$ [$M_{i,\,j}$ can be calculated using (20)]; and 3) if $\max\{M_{i,\,j} \times P_{j,\,d,\,k-1}\} > P_{i,\,d,\,k}$, update the value of $P_{i,\,d,\,k}$ as $P_{i,\,d,\,k} = \max\{M_{i,\,j} \times P_{j,\,d,\,k-1}\}$.

Similar to DIF, the PIF algorithm also has a time complexity of $O(HN)$ for calculating all of the expected probabilities from 0 to $H$.

## VI. Simulations and Discussions

Here, we will investigate the performance of our algorithms (DIF and PIF) and compare them to other representative routing algorithms using four Cambridge Haggle traces [11] and the UMassDieselNet trace [12]. These four Cambridge Haggle traces are the Cambridge trace, the Content trace, the Infocom trace, and the Infocom2006 trace, respectively. We do not include the Intel trace because the trace contains only nine nodes, where all algorithms can almost flood the network, and there would be no clear distinction among them.

Each simulation result was averaged over 40 000 randomly generated messages, and the detailed simulation settings are shown in Table I. We assumed an infinite forwarding bandwidth

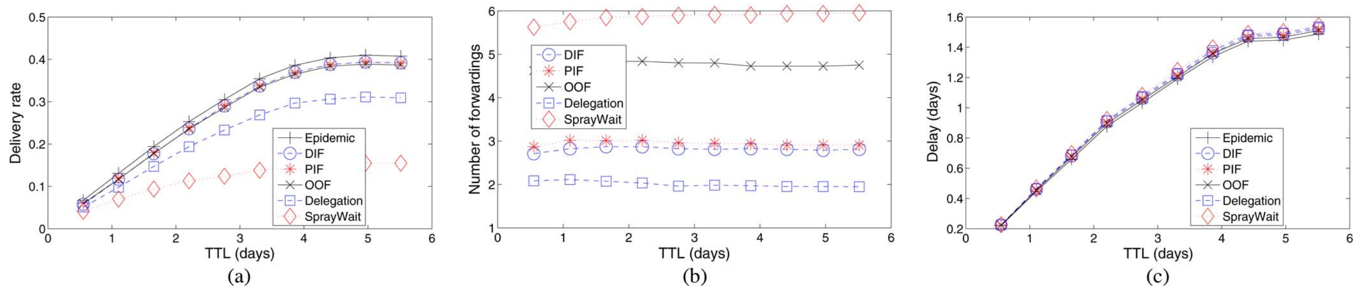



Fig. 6. Simulation results in Cambridge trace. (a) Delivery rate in Cambridge trace. (b) Average number of copies in Cambridge. (c) Delay in Cambridge trace.

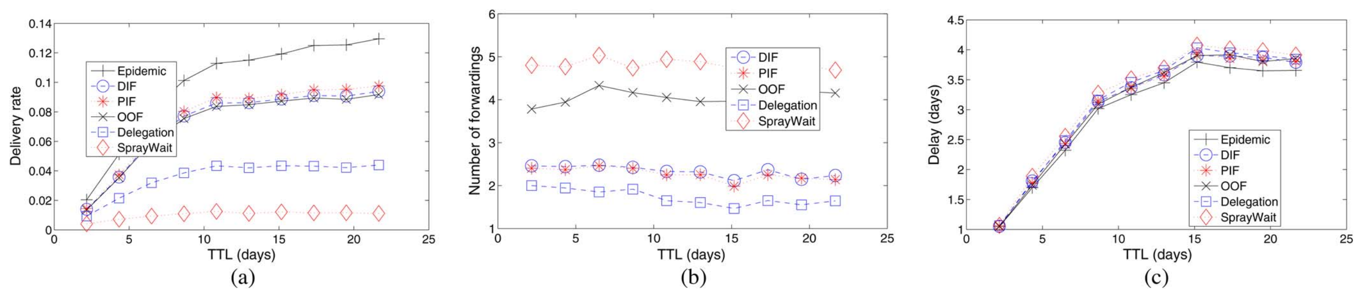



Fig. 7. Simulation results in Content trace. (a) Delivery rate in Content trace. (b) Average number of copies in Content trace. (c) Delay in Content trace.

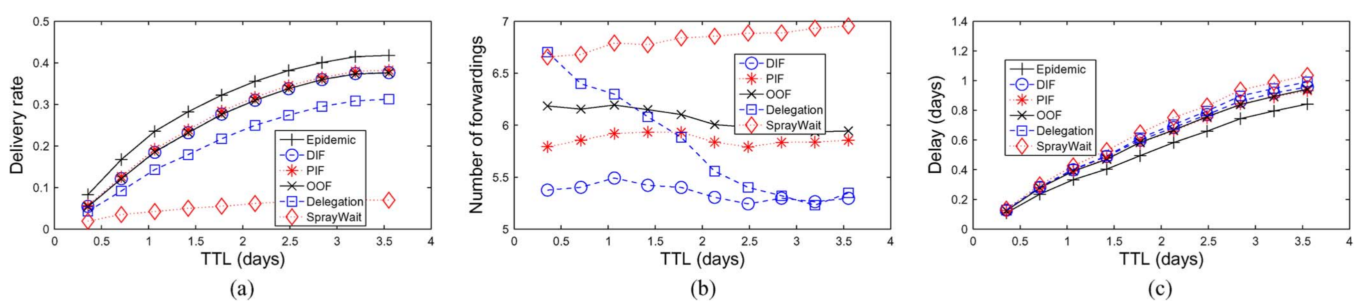



Fig. 8. Simulation results in Infocom2006 trace. (a) Delivery rate in Infocom2006 trace. (b) Average number of copies in Infocom2006 trace. (c) Delay in Infocom2006 trace.

in simulations because we focused on how to select good forwarders and, thus, discussed a high-level forwarding strategy in this paper. However, we have added simulation results in the following to show that our algorithms are also suitable for limited bandwidth.

### A. Comparison of Algorithms

We investigate the proposed algorithms against four representative algorithms, including epidemic routing [13], spray-and-wait [14], delegation forwarding [10], and OOF. The reasons for choosing these algorithms are the following. First, with unlimited bandwidth, the theoretical delivery rate of the epidemic routing algorithm is the highest; therefore, it can be used as an upper bound on the delivery rate. Second, spray-and-wait is a representative algorithm based on the epidemic routing algorithm, which uses logical tickets to keep the total number of copies of each message smaller than $L$. Third, delegation forwarding is a representative algorithm based on the one-hop delivery quality. Its delivery rate is quite high, and the number of message copies is also bounded to a very low level (because of the increasing threshold value of forwarding). Finally, the OOF algorithm also employs expected delay to describe the multihop delivery quality. Our algorithms have lower complexity and forwarding overhead than OOF, as shown in the simulation results.

### B. Results and Discussions

The simulation results using these traces are shown in Figs. 6–10. The routing algorithms are compared in terms of their delivery rate, average number of copies, and average delay, which are shown in subfigures (a)–(c) in Figs. 6–10, respectively. Since the TTL of a message has an important impact on routing performance, our simulations compare different algorithms by changing the TTL. Therefore, the horizontal axis of each figure is the TTL, and it is given in days. However, the vertical axis of each subfigure has different meanings. First, in the subfigure showing the delivery rate, the vertical axis ranges from 0 to 1, which means the proportion of messages successfully reaching the destination. Second, in the one showing the average number of copies, the vertical axis represents the total

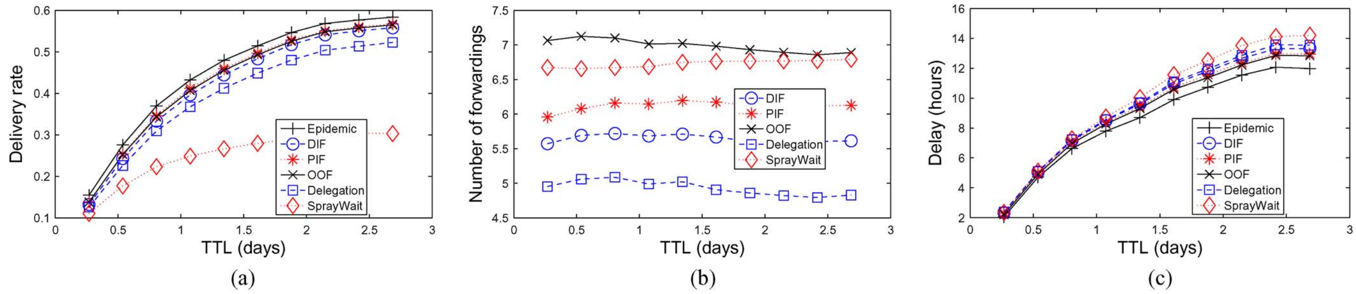



Fig. 9. Simulation results in Infocom trace. (a) Delivery rate in Infocom trace. (b) Average number of copies in Infocom trace. (c) Delay in Infocom trace.

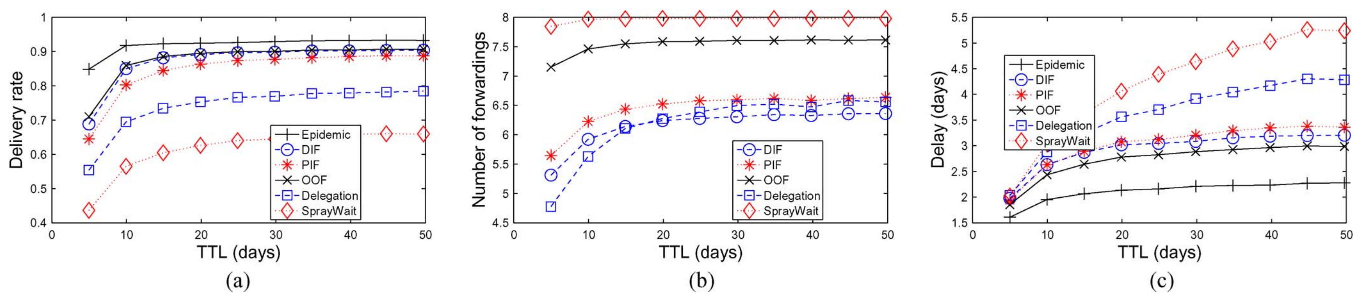



Fig. 10. Simulation results in UMassDieselNet trace (bus-route-based). (a) Delivery rate in UMassDieselNet trace. (b) Average number of copies in UMassDieselNet trace. (c) Delay in UMassDieselNet trace.

number of copies of a message, which reflects the energy cost of forwarding (i.e., fewer copies means lower energy cost). The results of epidemic routing are not shown in the figures because they are much larger than those for the other algorithms in terms of the average number of copies. Third, in the subfigure showing the average delay, the vertical axis represents the average time (in days) for a message from source to destination using a particular algorithm.

The delivery rates of the algorithms are compared in Figs. 6(a)–10(a). We make several observations. First, in some traces [i.e., in Figs. 6(a), 9(a), and 10(a)], the delivery rates of DIF and PIF are close to the epidemic routing algorithm, which is theoretically optimal in terms of the delivery rate. Second, DIF and PIF algorithms perform slightly better than OOF, but the improvement is marginal. Therefore, we can consider the three algorithms to have the same performance in terms of delivery rate. Third, compared with the delegation forwarding algorithm, DIF and PIF algorithms significantly improve the delivery rate, and it increases even more as the mobility of nodes becomes more regular. For example, in Fig. 7(a), DIF and PIF roughly double the delivery rate when compared with delegation forwarding.

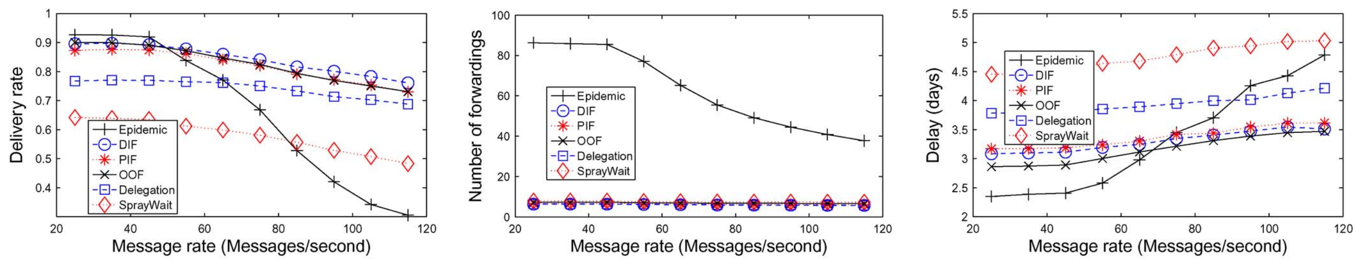
The average number of copies of the algorithms are compared in Figs. 6(b)–10(b). First, compared with OOF, which also uses multihop delivery quality, both DIF and PIF algorithms have better performance in all traces (the number of message copies is fewer than that in OOF). The detailed percent reduction of DIF and PIF over OOF is shown in Table II, where $C_{\mathrm{OOF}}$, $C_{\mathrm{DIF}}$, and $C_{\mathrm{PIF}}$ denote the average number of copies per message in OOF, DIF, and PIF, respectively. The percent reduction of DIF is computed as $(C_{\mathrm{OOF}} - C_{\mathrm{DIF}})/C_{\mathrm{OOF}} \times 100\%$, and the percent reduction of PIF is computed as $(C_{\mathrm{OOF}} - C_{\mathrm{PIF}})/C_{\mathrm{OOF}} \times 100\%$. We observe that when the

TABLE II
PERCENT REDUCTION IN MESSAGE COPIES IN DIF AND PIF OVER OOF FOR EACH TRACE

| Trace | $C_{DIF}$ | $C_{PIF}$ |
|---|---|---|
| Cambridge | 40.9% | 38.2% |
| Infocom | 19.1% | 13.5% |
| Infocom2006 | 11.0% | 4.3% |
| Content | 42.5% | 44.3% |
| DieselNet | 17.8% | 16.2% |

DTN environments are more regular (i.e., as in the Cambridge or the Content trace), the improvement DIF and PIF offer is more obvious. For example, DIF reduces message copies by 42.5% in the Content trace, and PIF reduces them by 44.3%. Second, compared with the delegation forwarding algorithm which uses one-hop delivery quality, our DIF and PIF algorithms require a slightly higher number of message copies in most traces. However, they can become conservative in message forwarding, when adapting to large TTLs. In the UMass DieselNet trace, for example, when the TTL is larger than 20 days, the number of copies in the DIF algorithm becomes fewer than that in the delegation forwarding algorithm. The reason is that DIF and PIF algorithms use the multihop delivery quality, which allows them to know when they have enough time to find the optimal forwarding path.

Figs. 6(c)–10(c) show the average delay of the algorithms in each trace. The epidemic routing algorithm has the lowest average delay, and the highest average delay is displayed by the spray-and-wait algorithm. The OOF algorithm is the second best in terms of the average delay, and DIF and PIF are both close to OOF. To summarize, algorithms using the multihop delivery quality (i.e., OOF, DIF, and PIF) have the lowest delay among all of the algorithms, except the Epidemic Routing algorithm.




(a) (b) (c)

Fig. 11. Simulation results versus message rate in UMassDieselNet trace (bus route based). (a) Delivery rate versus message rate in UMassDieselNet trace. (b) Average number of copies versus message rate in UMassDieselNet trace. (c) Delay versus message rate in UMassDieselNet trace.

### C. *Effect of Bandwidth*

To investigate the effect of bandwidth on our algorithms, we also performed simulations where we varied the message rate. The simulations were conducted as follows. First, we limited the number of messages that two nodes in contact can exchange, which is proportional to the length of the contact given by the trace data. Specifically, we restricted the maximum number of messages to 50 messages per second. Second, we set a message generation rate such that the number of messages each algorithm needs to forward in each contact opportunity under this message generation rate will exceed this bandwidth limitation. We varied the message generation rate from 25 to 115 messages per second (with the step size being 10) in each set of simulations to evaluate and compare the performance of the algorithms under different levels of limitation in bandwidth.

Due to space limitations, we only show the bandwidth-related simulation results for the UMassDieselNet trace in Fig. 11(a)–(c), respectively. The horizontal axis in the figures represents the message rate. We observe that the epidemic routing algorithm degrades most significantly when the message rate exceeds the bandwidth (50 messages per second), but the effect on other algorithms with hop count limitation is minor. Thus, the performance of our algorithms are quite robust to bandwidth limitation.

## VII. Conclusion

In this paper, we have discussed a routing scheme, which can make better use of historical information in the network and improve the routing performance. Specifically, an expected delay and an expected probability parameterized by the remaining hop count were used to characterize the multihop delivery quality of each node, and DIF and PIF algorithms were then developed based on the delay-weighted graph and the probability-weighted graph, respectively. The proposed DIF and PIF algorithms find the optimal forwarding path by minimizing the expected delay and by maximizing the expected probability, respectively. For comparison, we implemented DIF and PIF, as well as other representative algorithms to perform extensive trace-driven simulations on several real traces.

Analysis and simulation results yielded the following observations. First, in some traces, the delivery rates of DIF and PIF algorithms were close to that of the epidemic routing algorithm, which is theoretically optimal. However, epidemic routing has much more forwarding overhead than DIF and PIF algorithms due to its flooding-based forwarding. Second, although the DIF and PIF algorithms need slightly more message copies compared with the delegation routing algorithm (which uses a one-hop delivery quality), they significantly improve the message delivery rate, and the delivery rate increase even more as the mobility of nodes becomes more regular. Third, compared with the OOF algorithm that also employs the multihop delivery quality, DIF and PIF algorithms have two major advantages. First, they significantly reduce the number of message copies (the maximum reduction is over 40%), whereas they are quite close to the OOF algorithm in terms of the delivery rate and the average delay. Second, the time complexity of DIF and PIF algorithms is only $O(HN)$, having a substantial reduction compared with the $O(HN \log N)$ of the OOF algorithm.

## References

[1] K. Fall, "A delay-tolerant network architecture for challenged Internets," in *Proc. ACM SIGCOMM*, 2003, pp. 27–34.

[2] V. Cerf *et al.*, Delay Tolerant Networking Architecture, 2006, Internet draft: draft-irrf-dtnrg-arch.txt, DTN Res. Group.

[3] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proc. IEEE PerCom*, 2007, pp. 79–85.

[4] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proc. ACM SIGCOMM*, 2007, pp. 373–384.

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

[6] M. Grossglauser and M. Vetterli, "Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion," in *Proc. IEEE INFOCOM*, 2003, pp. 1954–1964.

[7] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Service Assurance with Partial and Intermittent Resources*. Berlin, Germany: Springer-Verlag, Aug. 2004, pp. 239–254.

[8] C. Liu and J. Wu, "On multi-copy opportunistic forwarding protocols in nondeterministic delay tolerant networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 6, pp. 1121–1128, Jun. 2012.

[9] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *Proc. ACM MobiHoc*, 2008, pp. 351–360.

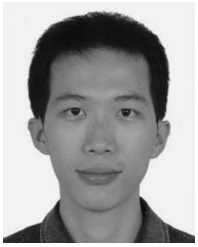
[10] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proc. ACM MobiHoc*, 2008, pp. 251–260.

[11] J. Scott *et al.*, CRAWDAD data set cambridge/haggle (v. 2006-09-15), Sep. 2006. [Online]. Available: http://crawdad.cs.dartmouth.edu/cambridge/haggle

[12] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption tolerant network: Mobility modeling and impact on routing," in *Proc. ACM MobiCom*, 2007, pp. 195–206.

[13] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep., Apr. 2002.

[14] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM WDTN*, 2005, pp. 252–259.



**Mingjiong Liang** received the B.S. degree in software engineering and the M.S. degree in computer software and theory from Sun Yat-Sen University, Guangzhou, China, in 2009 and 2012, respectively.

His main research interests include delay-tolerant networks and routing protocols in wireless networks.

**Zhiguo Zhang** received the B.Eng. degree in control engineering from Xian Jiaotong University, Xi'an, China, in 1982; the M.Eng. degrees in computer science from Shanghai University of Technology, Shanghai, China, in 1989 and from Yamagata University, Yamagata, Japan, in 1993; and the Ph.D. degree in computer science from Sun Yat-Sen University, Guangzhou, China, in 2006.

From 1982 to 1995, he served as an Assistant, a Lecturer, and an Associate Professor with the Department of Computer Science, Northeast Institute of Electrical Engineering. In 1984, he was a Visiting Engineer with Digital Equipment Corporation. Since 1995, he has been with Sun Yat-Sen University, where he is currently an Associate Professor of computer science. His research interests include computation theory, program and software theories, parallel and distributed computing, and biomolecular computing.

**Cong Liu** received the B.S. degree in microelectronics from South China University of Technology, Guangzhou, China, in 2002; the M.S. degree in computer software and theory from Sun Yat-Sen University, Guangzhou, in 2005; and the Ph.D. degree in computer science from Florida Atlantic University, Boca Raton, FL, USA.

He is currently an Assistant Professor with the Department of Computer Science, Sun Yat-Sen University. His main research interests include delay-tolerant networks and deep packet inspection.

**Li Chen** (SM'08) received the B.Sc. degree in applied physics from Jinan University, Guangzhou, China, in 2003 and the M.Sc. degree in communications and signal processing and the Ph.D. degree in mobile communications from Newcastle University, Newcastle upon Tyne, U.K., in 2004 and 2008, respectively.

From 2007 to 2010, he was a Research Associate with Newcastle University. Since 2010, he has been with the School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, where he is currently an Associate Professor and the Associate Head of the Department of Electronic and Communication Engineering. Between 2010 and 2012, he visited the Institute of Network Coding, the Chinese University of Hong Kong, Shatin, Hong Kong, several times. He is currently a principle investigator of two National Natural Science Foundation of China projects and a coinvestigator of the National Basic Research Program (973 program) project. His primary research interests include information theory, channel coding, and wireless communications.

Dr. Chen is a committee member of the Chinese Institute of Electronics on Information Theory. He is a technical program committee member for various international conferences, including the 2013 International Workshop on High Mobility Wireless Communications and the 2014 IEEE International Conference on Communications. He received a British Overseas Research Scholarship.