Delegation Forwarding in Delay Tolerant Networks Multicasting

Yunsheng Wang, Xiaoguang Li, and Jie Wu
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA
Email: {yunsheng.wang, xiaoguang.li, jiewu}@temple.edu

***Abstract*—Delay tolerant networks (DTNs) are a special type of wireless mobile networks which may lack continuous network connectivity. Multicast supports the distribution of data to a group of users, a service needed for many potential DTNs applications. While multicasting in the Internet and mobile ad hoc networks has been studied extensively, due to the unique characteristic of frequent partitioning in DTNs, multicasting in DTNs is a considerably different and challenging problem. It not only requires new destinations of multicast semantics, but also brings new issues to the design of routing algorithms. In this paper, we propose new forwarding models for DTNs multicast and develop several multicast forwarding algorithms. We use delegation forwarding (DF) in DTNs multicast and compare it with single and multiple copy multicast models, which are also proposed in this paper. From the analytical results, we have the following conclusions: (1) Although the single copy model has the smallest number of forwardings, its latency is much longer than the other two models. (2) Among these three models, the delegation forwarding model has the least delay. The effectiveness of our approach is verified through extensive simulation both in synthetic and real traces.**

***Index Terms*—Delay tolerant networks (DTNs), delegation forwarding (DF), forwarding algorithms, message replication, multicast.**

I. Introduction

With the advancement in technology, the communication devices with wireless interfaces become more and more universal. Recently, delay tolerant networks (DTNs) [1] technologies have been proposed to allow nodes in such extreme networking environments to communicate with one another. There is no end-to-end path between some or all of the nodes in DTNs. These networks have a variety of applications in situations that include crisis environments, such as emergency response and military battlefields, vehicular communication, deep-space communication, and non-interactive Internet access in rural areas.

Several DTNs unicast routing schemes have been proposed [2], [3], [4], [5], [6], [7]. However, having an efficient delivery service for multicast traffic is equally important. We cannot directly apply the multicast approaches proposed for the Internet or well-connected mobile ad hoc networks to DTNs environments because of the sparse connectivity among nodes in DTNs. Our scheme is different from the previous approaches as we do not rely on global information. That is, forwarding decisions are made using local information only when nodes encounter.

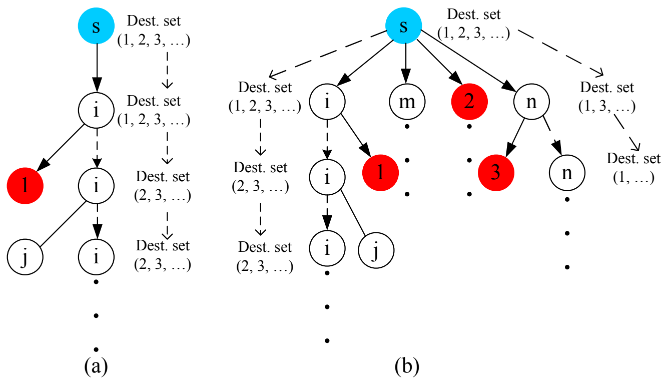
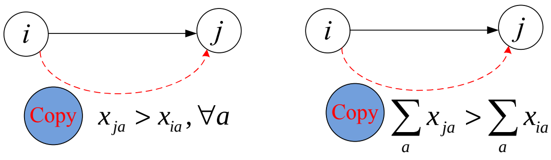


Fig. 1. Multicast tree: (a) single copy; (b) multiple copy and delegation forwarding.

In this paper, we focus on improving the performance of multicast in DTNs by developing three multicast forwarding algorithms: (1) *single copy multicast*: which has only one copy for all destinations. The message holder will only forward the copy to a node whose quality is higher considering all destinations; (2) *multiple copy multicast*: which has one copy for each destination. The message holder for each destination can be different. The message holder for a particular destination will forward the copy to an encountered node which has a higher quality, with respect to the destination; (3) *delegation forwarding multicast*: the message holder for each destination will replicate the copy (for that destination) and forward it to an encountered node that has a higher quality than all previous nodes seen so far, with respect to that particular destination. Our proposed multicast schemes are based on the dynamic multicast trees, as shown in Fig. 1. In Fig. 1(a), we can see that in the single copy model, there is only one tree branch. In multiple copy and delegation forwarding models, there are multiple copies of the message in the network, hence, there are multiple tree branches to seek the destination nodes in Fig. 1(b).

The major contributions of our work are as follows:

1) We present three multicast models in DTNs: single copy, multiple copy, and delegation forwarding.
2) Then, we formally analyze these three models in terms of the number of forwardings and latency. We use these three methods as forwarding algorithms in synthetic and



(a) (b)

Fig. 2. Single copy multicast in DTNs.

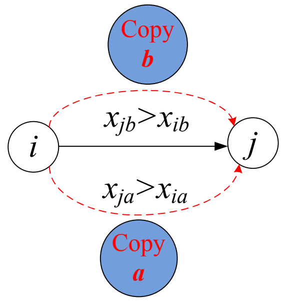


Fig. 3. Multiple copy multicast in DTNs.

real trace simulations.

3) The analytical and simulation results show that our all three multicast forwarding algorithms in DTNs can reduce the number of forwardings compared with flooding. The single copy model has the fewest number of forwardings. Latency comparison indicates that delegation forwarding has the least amount of latency.

The rest of this paper is organized as follows: Section II discusses preliminary work: delegation forwarding algorithm. Section III presents an overview of our algorithms implemented in DTNs multicast. Section IV analyzes these algorithms. Section V focuses on the evaluation. Section VI reviews the related work. We summarize the work in Section VII.

II. Preliminary Work

Recently, an approach called delegation forwarding (DF) [7] caught significant attention in the research community because of its simplicity and impressive performance. Its main idea is to assign a quality and a level value to each node. We will use the frequency of a node meeting the destinations as the quality value of a node in this paper. Initially, the level value of each node is equal to its quality value. During the routing process, a message holder only forwards the message to a node with a higher quality than its own level. In addition, the message holder also raises its own level to the quality of the higher quality node. This means a node will forward a message only if it encounters another node whose quality value is greater than any node met by the message so far.

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 and its total number of forwardings are expected to be decreased. Thus, using DF can reduce the network cost. From analysis in [7], we see that in an N-node network, delegation forwarding has an expected cost of $O(\sqrt{N})$ when compared with a naive scheme of forwarding to any higher quality node having an expected cost of $O(N)$.

Because DF's performance is capable of reducing the cost in DTNs, in this paper we will extend it into DTNs multicast research to analyze two metrics: (1) *the number of forwardings*: the number of forwardings for a whole multicast process. This can be considered as the cost for the multicast process; (2) *latency*: the average duration between a message's generation and the arrival time at the last destination. "High performance" means fewer number of forwardings and smaller latency.

III. Multicast Forwarding Algorithms

In this section, we will introduce three forwarding algorithms designed for DTNs multicast. First, we assume there are N nodes and D destinations in delay tolerant networks. When nodes come into contact, they are capable of exchanging messages.

A. Single copy multicast

The main idea of the single copy multicast model is that the source node will multicast a single copy to D destinations. Quality value x_{ia} denotes the frequency node i, which meets with destination a, $(a \in \{1, 2, ..., D\})$. When node i meets with node j, if for all destinations $x_{ja} > x_{ia}$, then the copy will be forwarded from node i to node j. Otherwise, unless node j is a destination, node i will not forward the message to node j. This means the message holder will just forward the copy to a node which has a higher quality for all destinations. Fig. 2(a) shows the forwarding decision rule for this algorithm.

We also apply a weak strategy in our simulation. We call it *single copy (sum)*. When node i meets node j, they compare the sum of the quality value for all destinations. If $\sum_{a=1}^{D} x_{ja} > \sum_{a=1}^{D} x_{ia}$, node i will forward the copy to node j. When the copy is forwarded to one of the destinations, this destination will be deleted from the destination set. Fig. 2(b) gives the simple forwarding algorithm, as we mentioned above.

B. Multiple copy multicast

Although single copy multicast has a smaller number of forwardings, it has a much longer delay. We believe that another algorithm based on the multiple copy multicast will reduce the latency. Compared with the single copy model, there are D copies (same as the destinations number) in the source node in the multiple copy multicast model. The main idea is, after meeting with node j, which has higher quality x_{ja} for destination a, node i will forward a copy to node j and 'ask' node j to forward this copy to destination a. If node j is a destination, node i will forward a copy to this destination node without hesitation. The destination node can also be a relay for other destinations. This forwarding algorithm is shown in Fig. 3. As shown in Fig. 3, copy a is forwarded from node i to node j, because node j has a higher probability to meet with destination a ($x_{ja} > x_{ia}$).

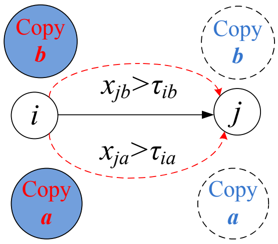


Fig. 4. Delegation forwarding multicast in DTNs.

C. Delegation forwarding multicast

For any node i, the forwarding problem is a simple question: "upon contact with node j, should node i forward the message to node j?" For many algorithms, the answer to the forwarding question is "forward the message if node j's quality is higher than node i." However, the cost of this approach can still be quite high. To reduce the cost, we seek to forward the message only to the highest quality nodes that have previously met. Conceptually, we would like to forward less, and give the message to the nodes which are the best candidates for eventual delivery to the destinations. Thus, the forwarding question becomes "is node j among the very highest quality nodes?"

The delegation forwarding multicast algorithm's main idea is to assign a quality value (which is static) and a level value (which is dynamic) for each node to each destination. Initially, the level value τ_{ia} for destination a of each node is equal to its quality value x_{ia} for destination a. During the routing process, a message holder i compares the quality x_{ja} of the node j it meets with its level value τ_{ia}. It only forwards the message to a node with a higher quality value than its own level value and 'asks' this node to help forward the message to destination a. This approach does not need global knowledge. Each node decides whether it should or should not forward the message by itself. This is suitable for a distributed environment, such as DTNs. In addition, the message holder also raises its own level to the higher quality. If node j is one of the destinations, node i will forward a message to it and also use the strategy to determine whether node j is a good relay to forward the message.

The DF algorithm is shown in Fig. 4 and Algorithm 1. The main difference from the previous two models is, in DF, the message will be replicated and after the forwarding process, initial message holder i and its relay node j will both have the copy of the message, therefore, there will be multiple nodes to seek the destinations. This means DF can reduce the cost and delay dramatically. The analysis and simulation results support our expectations.

IV. Analysis

In this section, we offer mathematical models according to our algorithms proposed in previous sections. We compare the number of forwardings and latency of these three multicast models.

Algorithm 1 Delegation Forwarding

There are N nodes in the network.
There are D destinations need to be multicast.
Node n has quality x_{nd} and level τ_{nd} for destination d.
INITIALIZE $\forall n, d : \tau_{nd} \leftarrow x_{nd}$.
On contact between node i, which is the message holder for destination a and node j:
if $x_{ja} > \tau_{ia}$ **then**
    $\tau_{ia} \leftarrow x_{ja}$
    if node j does not have the message for destination a **then**
        replicate a message to node j.
    end if
else
    if node j is the destination a **then**
        replicate a message to node j.
    end if
end if

A. Number of forwardings

In the following, we will consider a single message and calculate how many times it is forwarded before reaching all destinations. We will first offer the results of the delegation forwarding multicast model. The results of the other two models can be derived from this model.

1) Number of forwardings with the delegation forwarding multicast model: The cost of DF in DTN unicast is given in [7]. To make the paper self-contained, we include some of the ideas and proof methods in [7]. For any node i maintaining a quality metric for destination a: x_{ia} (which lies between $(0, 1]$) and a level value τ_{ia}, we focus on the gap $g_{ia} = 1 - \tau_{ia}$ between the current level and 1. The node that generates the message has a level value initially equal to its quality value, *i.e.*, $\tau_{ia} = x_{ia}$. We denote the initial gap $g_a = 1 - x_{ia}$.

***Theorem** 1:* Given the level value τ_{ia}, the expected number of forwardings in the delegation forwarding multicast model is

$$E[F_{delegation}] \lesssim \frac{1}{2}\sqrt{N} + \frac{1}{3}D \cdot \sqrt{N},$$

where N is the number of nodes, and D is the number of destinations.

Proof: Suppose a node 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 of the nodes then occurs as soon as it meets a node with a higher quality value than G_n, and all values above this level are equally likely.

Hence, we can write

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

where U is independent of G_n and follows a uniform distribution on $(0, 1]$. According to [7], in our multicast scheme, we can find:

$$E\left[G_{n+1}|G_n\right]=\frac{G_n}{2},\text{ hence, } E\left[G_n\right]=\frac{\sum_{a=1}^{D} g_a}{2^n}.$$

Moreover, from Eq. (1), we see that G_n approximately follows a lognormal distribution, with median $(\sum_{a=1}^{D} g_a)/e^n$. Hence, the distribution is highly skewed with most of the probability mass below the mean. So with a large probability, we have $G_n \leq (\sum_{a=1}^{D} g_a)/2^n$.

As in [7], the replication process can be described by a dynamic binary tree T, which contains all the nodes that have a copy of the message. We define the set $B_a = \left\{i|x_{ia} \geq 1-\frac{g_a}{\sqrt{N}}\right\}$, $a \in \{1,2,...,D\}$, 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 threshold above $1-\frac{g_a}{\sqrt{N}}$. All nodes in the subtree have a gap less than $\frac{g_a}{\sqrt{N}}$. This subtree is called the *target-stopped tree*.

According to [7], the essential observation is the following: if n is close to $log_2\left(\sqrt{N}\right)$, then except for a small probability, a node at generation n in the tree has a gap of at most $g_a/2^n \leq g_a/\sqrt{N}$. 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 $0,1,\ldots,n-1$ is at most $2^n=\sqrt{N}$.

In [7], Erramilli et al. offer the number of forwardings in the delegation forwarding unicast model. Hence, in delegation forwarding multicast model of D destinations, the total size of this tree is at most:

$$C_{delegation} \lesssim \sqrt{N}+|\sum_{a=1}^{D} B_a| = (1+\sqrt{\sum_{a=1}^{D} g_a})\cdot\sqrt{N}.$$

Then, we obtain the total number of forwardings:

$$F_{delegation} \lesssim \frac{1}{2}(1+\sqrt{\sum_{a=1}^{D} g_a})\cdot\sqrt{N},$$

Since, we know $\sqrt{\sum_{a=1}^{D} g_a} \leq \sum_{a=1}^{D}\sqrt{g_a}$.

Hence,

$$\int_0^1 \sqrt{\sum_{a=1}^{D} g_a}\,dg_a \leq \sum_{a=1}^{D}\int_0^1 \sqrt{g_a}\,dg_a = \frac{2}{3}D.$$

Therefore,

$$\begin{aligned} E[F_{delegation}] &= \int_0^1 \frac{1}{2}(1+\sqrt{\sum_{a=1}^{D} g_a})\cdot\sqrt{N}dg_a \quad (2)\\ &\lesssim \frac{1}{2}\sqrt{N}+\frac{1}{3}D\cdot\sqrt{N}. \end{aligned}$$

■

2) Number of forwardings with the single copy multicast model: In this part, we analyze the two single copy multicast models: *single copy* and *single copy (sum)*.

***Theorem** 2:* In the single copy model, the expected number of forwardings is:

$$E[F_{single}] \lesssim D\cdot log_D N,$$

Proof: In the single copy (all) model, we need to compare all of the destinations. When all of the quality values are larger than the current one, we will forward the copy. Thus, the probability of forwarding is equal to $\frac{1}{D}$, where D is the number of destinations.

Thus, in the single copy (all) model, the expectation of G_n becomes:

$$E\left[G_n\right]=\frac{\sum_{a=1}^{D} g_a}{D^n},$$

where D is the number of destinations.

Using the same methods, we obtain the number of forwardings:

$$F_{single} \lesssim log_D(N\cdot\sum_{a=1}^{D} g_a),$$

hence,

$$E[F_{single}] \lesssim D\cdot log_D N. \quad (3)$$

■

***Theorem** 3:* In the single copy (sum) model, the expected number of forwarding times is the same as the delegation forwarding multicast model:

$$E[F_{single(sum)}]dg \lesssim \frac{1}{2}\sqrt{N}+\frac{1}{3}D\cdot\sqrt{N}.$$

Proof: In the single copy (sum) model, when node i meets node j, the probability of the sum value on node i, which is larger than node j, is equal to 0.5. Thus, it is the same situation of delegation forwarding. Hence, the probability of the forwarding decisions is also 0.5.

$$E[F_{single(sum)}]=\int_0^1 F_{delegation}dg \lesssim \frac{1}{2}\sqrt{N}+\frac{1}{3}D\cdot\sqrt{N}. \quad (4)$$

■

3) Number of forwardings with the multiple copy multicast model: In this part, we will discuss the multiple copy multicast model.

***Theorem** 4:* In the multiple copy model, the expected number of forwardings is:

$$E[F_{multiple}] \lesssim D\cdot log_2 N.$$

Proof: Since the probability for the node to forward the copy is $\frac{1}{2}$, according to Eq. 3, we have:

$$F_{multiple} \lesssim log_2(N\cdot\sum_{a=1}^{D} g_a),$$

hence,

$$E[F_{multiple}] \lesssim D \cdot log_2 N. \quad (5)$$

■

4) Number of forwardings using flooding: In this part, we will discuss the number of forwardings using flooding.

***Theorem** 5:* When using flooding routing protocol, the expected number of forwardings is:

$$E[F_{flooding}] \approx \frac{D \cdot N}{2}.$$

Proof: From [7], we know that the number of forwardings of non-delegation forwarding in the unicast method is:

$$C_{no-delegation}(g) = gN,$$

where g is the initial gap value and N is the number of nodes. Thus, in the multicast forwarding, we have:

$$F_{flooding} \approx N \cdot \sum_{a=1}^{D} g_a,$$

hence,

$$E[F_{flooding}] \approx \frac{D \cdot N}{2}. \quad (6)$$

■

We find that our methods all have a smaller number of forwardings compared with flooding.

We use the synthetic trace to compare the number of forwardings of these methods. We will also compare these with our analysis results.

In the synthetic mobility model, we set up a 100-node environment. There are 67,226 contacts in 100,000 time slots. From Fig. 5, using the equations we obtained from analyzing the number of forwardings for these three algorithms, we find that our model produces a significantly decreased number of forwardings compared with flooding. The normal single copy model has the fewest number of forwardings, while the multiple copy model has the largest cost in these three models. Delegation forwarding reduces the cost gap between the single copy model and the multiple copy model. Simulation results also meet the analytical results.

B. Latency

We assume the contact time between one node to its next relay is t. The total time for multicast in these three models is:

$$T = \sum_{i=1}^{D} C_i \cdot t, \quad (7)$$

where C_i denotes the contact times between two destinations d_{i-1} and d_i.

1) In the single copy model, the probability of meeting with a higher quality node is $\frac{1}{D^n}$

$$T_{single} = t \cdot \sum_{i=1}^{D} D^i = t \cdot N \cdot \frac{D \cdot (D^D - 1)}{D-1}. \quad (8)$$

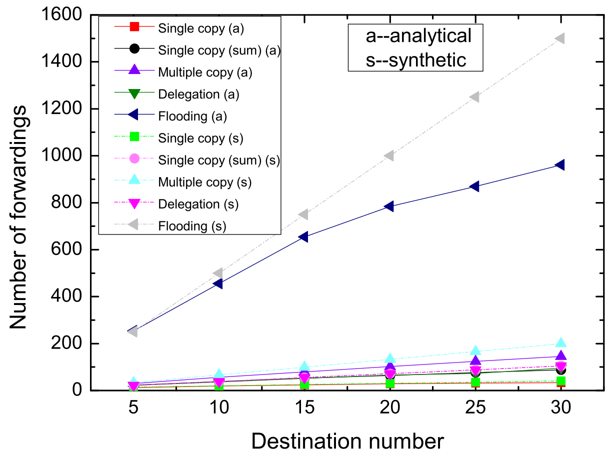


Fig. 5. Comparison of the analytical results and the synthetic model results.

2) In the multiple copy model, the probability of meeting with a higher quality node is $\frac{1}{2^n}$

$$T_{multiple} = t \cdot \sum_{i=1}^{D} 2^i = 2t \cdot N \cdot (2^D - 1). \quad (9)$$

3) In the delegation forwarding model, we need to calculate the maximum height of the target-stopped tree, as mentioned in Section IV(A). There are $n = log_2(N \cdot \sum_{a=1}^{D} g_a)$ generations to finish multicasting the copies to all destinations.

In the worst case, $g_a = 1$, $(a \in \{1, 2, ..., D\}$, then

$$T_{delegation} = t \cdot \sum_{i=1}^{log_2 DN} 2^i = 2t \cdot (DN - 1). \quad (10)$$

We can clearly see that $T_{delegation} < T_{multiple} < T_{single}$. Delegation forwarding has the best performance in DTNs multicast latency.

V. Simulation

In the previous sections, we analyzed the single copy, multiple copy, and delegation forwarding multicast algorithms in DTNs multicast, and have shown that they can dramatically reduce the number of forwardings. In this section, we evaluate the performance of the multicast routing algorithms presented in this paper. We use the Intel and Cambridge traces [8] in our simulation. These data sets consist of contact traces between short-range Bluetooth enabled devices carried by individuals.

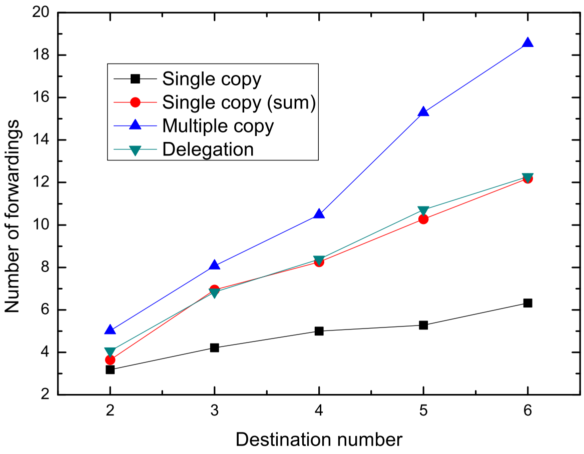
The following metrics are calculated in our simulation. Each simulation is repeated 1,000 times.

1. *Average cost*: the average number of forwardings for all destinations to receive the message.

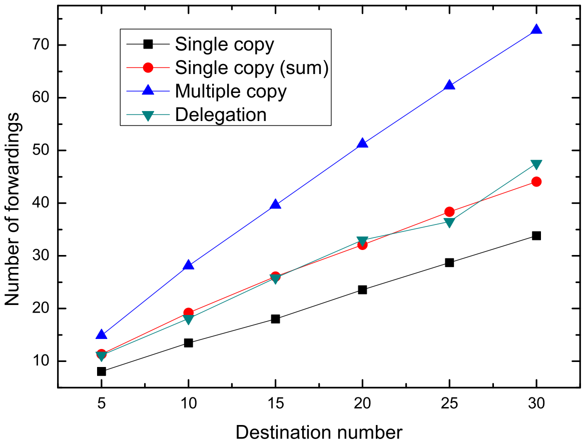
2. *Actual delay*: the average latency for all the delivered destinations to receive the message.

A. Simulation methods and setting

1) Synthetic trace: In synthetic mobility models, we set up 20-node and 100-node environments. There are 9,501 contacts in 10,000 time slots in the 20-node trace, and there are 67,226

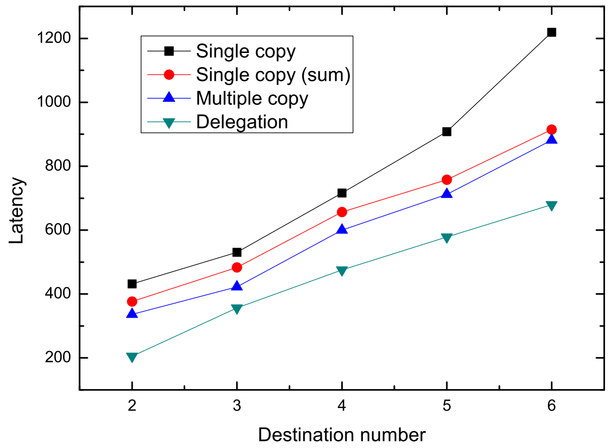


(a) Number of forwardings in 20-node trace

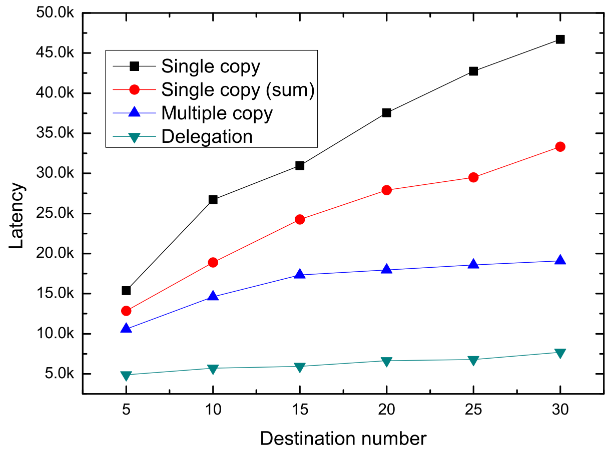


(b) Number of forwardings in 100-node trace

Fig. 6. Comparison of the number of forwardings in synthetic traces.



(a) Latency in 20-node trace



(b) Latency in 100-node trace

Fig. 7. Comparison of latency in synthetic traces.

contacts in 100,000 time slots in the 100-node trace. We will compare these three models in terms of the number of forwardings and latency.

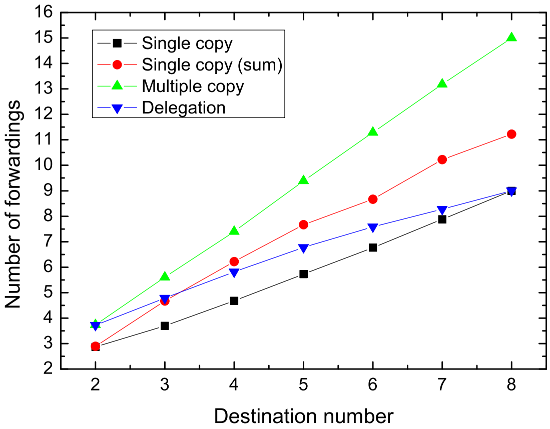
2) Intel trace: This trace includes Bluetooth sightings by groups of users carrying small devices (iMotes) for six days in the Intel Research Cambridge Corporate Laboratory. There is 1 stationary node, 8 nodes which are corresponding to mobile iMotes, and 118 nodes corresponding to external devices. There are $2,766$ contacts between these nodes. In our simulation, we randomly set one of these 9 nodes as the source, and choose other different nodes as the destinations. The destination numbers are from 2 to 8. We will compare these three models in terms of the number of forwardings and latency.

3) Cambridge trace: This trace includes Bluetooth sightings by groups of users carrying small devices (iMotes) for six days in the Computer Lab at the University of Cambridge. 12 nodes are corresponding to iMotes, while 211 nodes correspond to external devices. In total, only 12 iMotes could be used to produce this trace. Others were suffering from hardware resets. There are $6,732$ contacts between these nodes. In our simulation, we set 1 node as the source and choose different nodes as the destinations. The destination numbers are from 2 to 11. We will also compare the number of forwardings and latency as in the Intel trace.

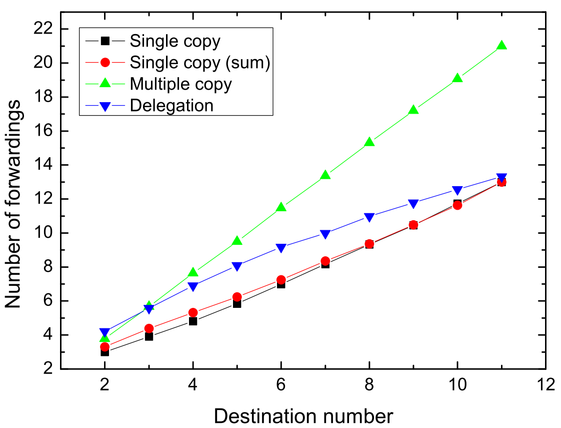
B. Results

First, we compared the performance of these forwarding algorithms in the synthetic traces, as shown in Figs. 6 and 7. In both the 20-node trace and the 100-node trace, we can see that the single copy model with strong strategy has the fewest number of forwardings. The delegation forwarding has a similar number of forwardings as the single copy (sum) model, and both better than the multiple copy model. From Fig. 6(a), we can see that the single copy model has 50% fewer number of forwardings than the delegation forwarding model. Delegation forwarding has about 30% fewer number of forwardings compared with the multiple copy model. In the 100-node trace, the results are similar. At the same time, delegation forwarding has much shorter latency than other models, while the single copy model has the longest latency among these protocols, in Fig. 7.

Then, we compared the number of forwardings among these three forwarding algorithms in real traces, as shown

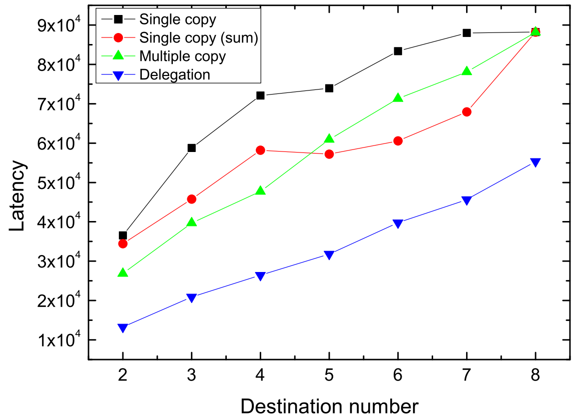


(a) Number of forwardings in Intel

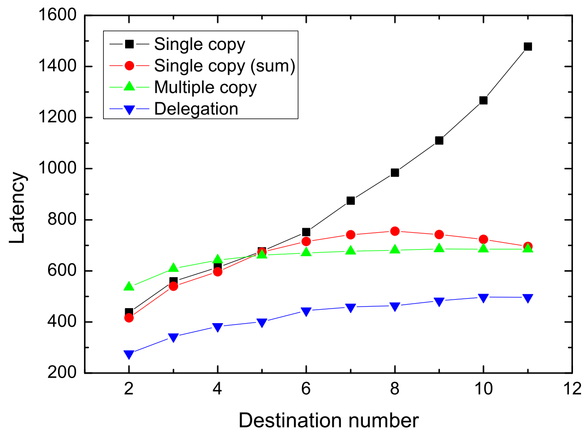


(b) Number of forwardings in Cambridge

Fig. 8. Comparison of the number of forwardings in real traces.



(a) Latency in Intel



(b) Latency in Cambridge

Fig. 9. Comparison of latency in real traces.

in Fig. 8. We can see that the single copy model using the strong strategy has the fewest number of forwardings. The delegation forwarding has a smaller number of forwardings than the multiple copy model in both Intel and Cambridge traces. In the Intel trace in Fig. 8(a), it needs about 1.2 times the number of forwardings to arrive at a destination using the strong strategy single copy model while the weak strategy needs 1.48 times. The multiple copy model and delegation forwarding model need 1.9 and 1.4 times, respectively. In the Cambridge trace, the number of forwardings per destination in the strong strategy and weak strategy single copy model is 1.2 and 1.3, respectively. Also, they are 1.9 and 1.5 times for the multiple copy and delegation forwarding models, respectively, as shown in Fig. 8(b). These results are the same as what we analyzed in Section IV.

The results of the latency comparison are shown in Fig. 9. Delegation forwarding has the least amount of latency, which has a 48% time reduction over the single copy model with the strong strategy. The single copy model has the longest latency among these algorithms. The delegation forwarding model has the least amount of latency, both in the Intel and Cambridge traces.

C. Summary of simulation

We first use these three forwarding methods in DTNs multicast. Simulation results confirmed that they have their own benefit used as the forwarding algorithm in DTNs multicast. We know that the single copy model has the longest latency and fewest number of forwardings, both in the simulation and analytical results. The multiple copy model reduces the latency from the single copy model, because it has more of a chance to meet with other higher priority nodes. Delegation forwarding uses many branches to forward the copies, so it has the shortest latency among these models, which has been proven by analytical results and simulation results. Although the delegation forwarding model has a slightly increased number of forwardings than the single copy model, it reduces the cost from the multiple copy model significantly. These forwarding algorithms are all better than flooding when comparing the number of forwardings.

VI. RELATED WORK

Many multicast protocols have been proposed to address the challenge of the frequent topology changes in mobile ad hoc networks [9]. Many well-known multicast routing protocols have been developed, including multicast extensions to open shortest-path first (MOSPF) [10], protocol independent multicast (PIM) [11], and core based tree (CBT) [12]. The typical source-tree routing algorithm applies the shortest path tree (SPT) algorithm and separates multicast trees needed to be computed, one for each sender. Most of the algorithms are based on the single copy model in MANETs. Our method is shortest-path tree based in both single or multiple copy.

There has been recent work which considers multicast in DTNs. In [13], Zhao, Ammar, and Zegura propose new semantic models for DTN multicast and develop several multicast routing algorithms with different routing strategies that allow users to explicitly specify temporal constraints on group membership and message delivery. [14] studies multicast in DTNs from the social network perspective. Gao et al. develop a single copy model where the forwarding metric is based on the social network perspective. Yang and Chuah [15] present a ferry based interdomain multicast delivery scheme, where a ferry is used to deliver multicast messages across groups that are partitioned. In [16], Lee et al. propose RelayCast, a routing scheme that extends the two-hop relay algorithm in the multicast scenario to improve the throughput bound of wireless multicast in DTNs. In [17], Wu and Wang provide a nonreplication multicasting scheme in DTNs while keeping the number of forwardings low, which is based on a dynamic multicast tree where each leaf node corresponds to a destination. Each tree branch is generated at a contact based on the compare-split rule.

The basic idea of Epidemic routing [18] in DTNs is to select all nodes in the network as relays. To control the number of forwardings, [19] employs some nodes with desirable patterns as message ferries, and opportunistic forwarding algorithms that analyze the performance of mobility-assisted schemes, theoretically. Some works make efforts in improving data forwarding performance by either determining the data delivery likelihood or spraying data [20] to relays waiting for contacts with destinations. The delegation forwarding algorithm [7] provides a unified approach to mobility-based metrics, which selects forwarding nodes based on the delivery likelihood. In contrast to DTNs, where mobility can be predicted or future information is known [21], [22], DF assumes no regularity of movement patterns, therefore its approach is naturally more probabilistic in nature. [23] is an extension of the DF algorithm. Based on DF, Chen et al. insert a probability p into the algorithm, which means it will not always forward the message to a higher quality node. This algorithm can further reduce the cost.

VII. CONCLUSION

In this paper, we studied the problem of multicasting in DTNs. We focused on the multicast forwarding algorithms. We discussed the single copy, multiple copy, and delegation forwarding algorithms in DTNs multicast. Then, we analyzed these three models mathematically. We finally turn to studying the performance of these three forwarding algorithms not only in synthetic traces, but also in real mobility traces. Trace driven simulation results have shown that using delegation forwarding has the smallest latency, while the single copy model has the fewest number of forwardings. In the future, we go forward with the probability delegation forwarding (PDF) [23] and threshold-based probability delegation forwarding (TPDF) schemes to further reduce the number of forwardings. We believe that this paper presents the first step in exploiting forwarding decision rules in DTNs multicast. Future research can benefit from our results by developing specific applications based on the provided multicast forwarding architecture in DTNs.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants CCF 1028167, CNS 0948184, and CCF 0830289.

REFERENCES

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

[2] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[3] C. Liu and J. Wu, “An optimal probabilistic forwarding protocol in delay tolerant networks,” in *Proc. of ACM MobiHoc*, 2009, pp. 105–114.

[4] V. Erramilli and M. Crovella, “Forwarding in opportunistic networks with resource constraints,” in *Proc. of the Third ACM workshop on Challenged networks (CHANTS)*, 2008, pp. 41–48.

[5] M. Y. S. Uddin, B. Godfrey, and T. Abdelzaher, “Relics: In-network realization of incentives to combat selfishness in dtns,” in *Proc. of IEEE International Conference on Network Protocols (ICNP)*, 2010.

[6] J. Reich and A. Chaintreau, “The age of impatience: optimal replication schemes for opportunistic networks,” in *Proc. of ACM the 5th international conference on Emerging networking experiments and technologies (CoNEXT)*, 2009, pp. 85–96.

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

[8] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, “CRAWDAD trace cambridge/haggle/imote/cambridge (v. 2006-01-31),” Jan. 2006.

[9] J. Wu, *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*. Auerbach Publications, 2005.

[10] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[11] S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, “The PIM architecture for wide-area multicast routing,” *IEEE/ACM Trans. Netw.*, vol. 4, no. 2, pp. 153–162, 1996.

[12] T. Ballardie, P. Francis, and J. Crowcroft, “Core based trees (CBT),” *SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, pp. 85–95, 1993.

[13] W. Zhao, M. Ammar, and E. Zegura, “Multicasting in delay tolerant networks: semantic models and routing algorithms,” in *Proc. of ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005, pp. 268–275.

[14] W. Gao, Q. Li, B. Zhao, and G. Cao, “Multicasting in delay tolerant networks: a social network perspective,” in *Proc. of ACM MobiHoc*, 2009, pp. 299–308.

[15] P. Yang and M. Chuah, “Efficient interdomain multicast delivery in disruption tolerant networks,” in *Proc. of IEEE Mobile Ad-hoc and Sensor Networks*, 2008, pp. 81–88.

[16] U. Lee, S. Y. Oh, K.-W. Lee, and M. Gerla, “Scaling properties of delay tolerant networks with correlated motion patterns,” in *Proc. of ACM CHANTS*, 2009, pp. 19–26.

[17] J. Wu and Y. Wang, "A non-replication multicasting scheme in delay tolerant networks," in *Proc. of The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2010.
[18] V. Srinivasan, M. Motani, and W. T. Ooi, "Analysis and implications of student contact patterns derived from campus schedules," in *Proc. of ACM MobiCom*, 2006, pp. 86–97.
[19] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. of ACM MobiHoc*, 2004, pp. 187–198.
[20] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. of ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005, pp. 252–259.
[21] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. of ACM SIGCOMM*, 2004, pp. 145–158.
[22] C. Liu and J. Wu, "Scalable routing in delay tolerant networks," in *Proc. of ACM MobiHoc*, 2007, pp. 51–60.
[23] X. Chen, J. Shen, T. Groves, and J. Wu, "Probability delegation forwarding in delay tolerant networks," in *Proc. of IEEE ICCCN*, 2009, pp. 1–6.