Multicasting in Delay Tolerant Networks: Delegation Forwarding

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 kind of wireless mobile network 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 designed by us. The effectiveness of our approach is verified through extensive simulation.**

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

I. Introduction

With the advancement in technology, the communication devices with wireless interfaces become more and more universal. Recently, delay tolerant networks [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, and non-interactive Internet access in rural areas.

Several DTNs unicast routing schemes have been proposed [2], [3]. 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. There has been recent work which considers heterogeneous conditions [4], where the authors show the maximum flow that can be achieved by static routing if global information about the nodes' schedules is known. Our scheme is different as we do not assume global information, and forwarding decisions are made in an online manner when nodes are met.

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 to 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.

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' number of forwardings and latency. We use these three methods as forwarding algorithms in real trace simulations.

3) The analysis and simulation results show that our three multicast forwarding algorithms in DTNs all can reduce the cost 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. 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) [5] 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 metric is greater than any node met by the message so far.

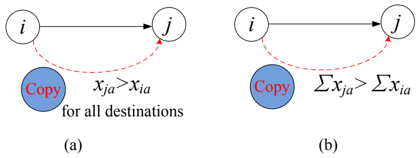


Fig. 1. Single copy multicast in DTNs.

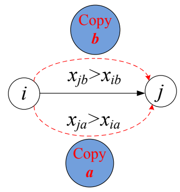


Fig. 2. Multiple copy multicast in DTNs.

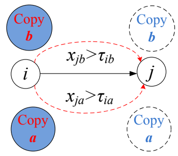


Fig. 3. Delegation forwarding multicast in DTNs.

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 [5] analysis, 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 DTNs. 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, $(a \in (1, D))$. 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. 1(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. 1(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 think 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 this 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. 2.

C. Delegation forwarding multicast

The delegation forwarding multicast algorithm's main idea is to assign a quality value and a level value 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. 3. The copy will be replicated and after the forwarding process, initial message holder i and its relay node j will both have the copy, 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 will consider a single message and calculate the number of forwardings before reaching all destinations.

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 $g_a = 1 - x_{ia}$.

Suppose a node updates its gap value n times. We denote the node's current gap 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 quality greater 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 [5], we then find:

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

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, and so with a large probability we have $G_n \leq (\sum_{a=1}^{D} g_a)/2^n$.

Let us describe the replication process via 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_a. Each time a node with a copy of the message meets another node having a higher quality than any node seen so far, we create two children of the node. The children represent each of the two nodes, and both have associated the updated gap value. We wish to bound the total size of this tree.

We define the set $B_a = \left\{ i|x_{ia} \geq 1 - \frac{g_a}{\sqrt{N}} \right\}$, $a \in (1, D)$, which we call the target set. We identify a subtree of the tree T in which children are excluded for nodes having a threshold above $1 - \frac{g_a}{\sqrt{N}}$. We call this subtree the target-stopped tree.

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 $g_a/2^n \leq g_a/\sqrt{N}$. 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 $0, 1, \ldots, n-1$ is at most $2^n = \sqrt{N}$.

Hence, 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 obtained the total number of forwardings:

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

hence,

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

In contrast, in the normal single copy 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,

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

In the single copy (sum) model, it is the same situation of delegation forwarding. Hence,

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

In the multiple copy model,

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

hence,

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

In contrast, the usual style of a forwarding algorithm, such as flooding, makes no threshold adaptation. Its number of forwardings are:

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

hence,

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

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. 4, using the equations we obtained from analyzing the number of forwardings for these three algorithms, we find that using our models produces a significantly decreased number of forwardings compared with flooding. The normal

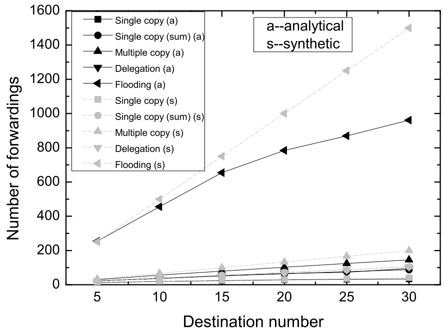
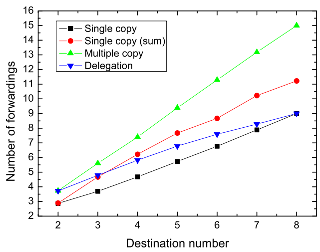
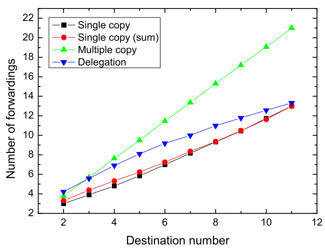


Fig. 4. Comparison of the analytical results and the synthetic model results.



(a) Number of forwardings in Intel



(b) Number of forwardings in Cambridge

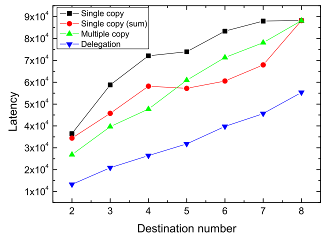
Fig. 5. Comparison of the number of forwardings.

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.

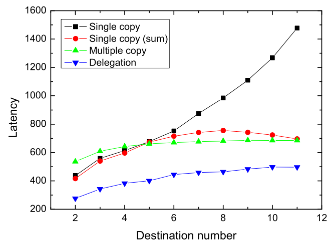
V. Simulation

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 [6] in our simulation. These data sets consist of contact traces between short-range Bluetooth enabled devices carried by individuals.

The number of forwardings and latency will be calculated in our simulation. Each simulation is repeated 1000 times.



(a) Latency in Intel



(b) Latency in Cambridge

Fig. 6. Comparison of latency.

A. *Simulation methods and setting*

1) 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.

2) 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.

B. *Results*

First, we compared the number of forwardings among these three forwarding algorithms, as shown in Fig. 5. We can see that the single copy model using the strong strategy has the fewest number 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, 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. These results are the same as what we analyzed in Section 4.

The results of the latency comparison are shown in Fig. 6. Delegation forwarding has the least amount of latency, which has a 48% time reduction over the single copy model. 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 node, 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 [7]. Many well-known multicast routing protocols have been developed, including Multicast extensions to open shortest-path first (MOSPF) [8] and Core Based Tree (CBT) [9]. 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 [10], Zhao, Ammar, and Zegura propose new semantic models for DTN multicast, in which one single node holds all destinations and delivery to each destination at contacts through movement [11] 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. In [12], 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.

To control the number of forwardings, [13] employs some nodes with desirable patterns as message ferries, and opportunistic forwarding algorithms that analyze the performance of mobility-assisted schemes, theoretically. The delegation forwarding algorithm [5] provides a unified approach to mobility-based metrics, which selects forwarding nodes based on the delivery likelihood. Delegation forwarding assumes no regularity of movement patterns, therefore its approach is naturally more probabilistic in nature. [14] 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.

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 models in DTNs multicast. Then, we analyzed these three models mathematically. We then turn to studying the performance of these three forwarding algorithms 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. 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 CNS 0948184, CCF 0830289, and CNS 0626240.

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 protocolin delay tolerant networks,” in *Proc. of ACM MobiHoc*, 2009, pp. 105–114.
[4] M. Garetto, P. Giaccone, and E. Leonardi, “Capacity scaling in delay tolerant networks with heterogeneous mobile nodes,” in *Proc. of ACM MobiHoc*, 2007, pp. 41–50.
[5] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, “Delegation forwarding,” in *Proc. of ACM MobiHoc*, 2008, pp. 251–260.
[6] 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.
[7] J. Wu, *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*. Auerbach Publications, 2005.
[8] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.
[9] T. Ballardie, P. Francis, and J. Crowcroft, “Core based trees (CBT),” *SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, pp. 85–95, 1993.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] 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.