

## Accepted Manuscript

A dynamic multicast tree based routing scheme without replication in delay tolerant networks

Yunsheng Wang, Jie Wu

PII: S0743-7315(11)00223-1  
DOI: 10.1016/j.jpdc.2011.11.007  
Reference: YJPDC 2953

To appear in: *J. Parallel Distrib. Comput.*

Received date: 1 February 2011  
Revised date: 13 September 2011  
Accepted date: 9 November 2011

Please cite this article as: Y. Wang, J. Wu, A dynamic multicast tree based routing scheme without replication in delay tolerant networks, *J. Parallel Distrib. Comput.* (2011), doi:10.1016/j.jpdc.2011.11.007

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



# A Dynamic Multicast Tree based Routing Scheme without Replication in Delay Tolerant Networks

Yunsheng Wang and Jie Wu

*Department of Computer and Information Sciences*

*Temple University*

*Philadelphia, PA 19122*

*Email: {yunsheng.wang, jiewu}@temple.edu*

---

## Abstract

Delay tolerant networks (DTNs) are a special type of wireless mobile networks which may lack continuous network connectivity. Multicast is an important routing function that supports the distribution of data to a group of users: a service needed for many potential DTNs applications. While multicasting in the Internet and in mobile ad hoc networks has been studied extensively, efficient multicasting in DTNs is a considerably different and challenging problem due to the probabilistic nature of contact among nodes. This paper aims to provide a non-replication multicasting scheme in DTNs while keeping the number of forwardings low. The address of each destination is not replicated, but is assigned to a particular node based on its contact rate level and active level. Our scheme 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 proposed in this paper. The *compare* part determines when a new search branch is needed, and the *split* part decides how the destination set should be partitioned. When only one destination is left in the destination set, we use either wait (no further relay) or focus (with further relay) to reach the final destination. The effectiveness of our approach is verified through extensive simulations. Ratio-based-split performs best in the compare-split step, both in synthetic and real traces. Using the wait scheme can reduce the number of forwardings, while using the focus scheme can reduce the latency.

*Keywords:* Contact, delay tolerant networks (DTNs), dynamic multicast tree, efficient protocols, multicast, opportunistic routing.

---

# A Dynamic Multicast Tree based Routing Scheme without Replication in Delay Tolerant Networks

Yunsheng Wang and Jie Wu

*Department of Computer and Information Sciences  
Temple University  
Philadelphia, PA 19122  
Email: {yunsheng.wang, jiewu}@temple.edu*

---

## 1. Introduction

With the advancement in technology, communication devices with wireless interfaces have become more and more universal. Recently, a lot of delay tolerant networks (DTNs) [1] technologies have been proposed to allow nodes in such extreme networking environments to communicate with one another. Delay tolerant networks are wireless networks where, most of the time, an end-to-end path does not exist between some or all of the nodes in the network. The nature of node contact<sup>1</sup> is non-deterministic. These networks have a variety of applications, including crisis environments such as: emergency response and military battlefields, vehicular communication, and deep-space communication.

Several DTNs unicast routing schemes have been proposed [2], [3], [4], [5]. 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 also been some work on multicast routing protocols in DTNs [6], [7], [8], [9], [10]. Existing work focuses on three models: (a) *single node* (also called *ferry model*) ([6] and [7]), in which one single node holds all destinations and delivers to each destination at contacts through movement; (b) *multiple copies*

---

<sup>1</sup>In DTNs, routes are comprised of a cascade of time-dependent *contacts* (communication opportunities) used to move messages from their origins toward their destinations [1].

*model* ([8] and [9]), in which the destination set is replicated at a contact once a certain condition related to the quality of the encountered node is satisfied; (c) *single copy model* [10], where a single copy of each destination is maintained where destinations can be scattered at different nodes. Each destination is forwarded to an encountered node if it has a higher contact frequency of reaching the corresponding destination. This forwarding rule is called *priority-based-split* (PS) in this paper.

Our scheme is based on the single copy model with the objective to reach destinations quickly while minimizing the number of forwardings. We observe that pure priority-based-split may produce an excessive number of forwardings (e.g., for a succession of small improvements). We propose to use the node's *active level* together with the *contact rate level* to determine when and how to split a destination set during a contact. The notion of the active level is based on the observation that an active node has a better chance of contacting a higher priority node later to improve its delivery time. More specifically, we have the following two notions:

- *Contact rate level* with respect to a destination: a priori knowledge or estimation of the number of contacts with the destination in a given period.
- *Active level* of a node: a priori knowledge or estimation of the number of total contacts in a given period.

In this paper, we propose a *compare-split* scheme at each contact during the construction of a dynamic multicast tree. The first step is the *compare* part. When node  $a$ , with a destination subset, has a contact with node  $b$  without any destination subset, we set the condition for splitting as follows: a split occurs when the sum of the contact rate levels for all destinations associated with  $b$  is higher than the one associated with  $a$ . The second step is the *split* part. We propose a *ratio-based-split* ( $RS$ ), which splits the destination subset based on the active levels of two encountered nodes. We then present an *optimal split algorithm*, which partitions the destination subset based on the calculated ratio such that the combined sum of the contact rate levels at nodes  $a$  and  $b$  are maximized.

When there is only one destination in the message holder's destination set, we use two schemes to forward the message to this destination: (1) *wait*: wait until meeting the destination; (2) *focus*: forward the message to a higher contact rate level node until arriving at the destination.

The major contributions of our work are as follows:

- We propose the notions of contact rate level and active level to guide the construction of a multicast tree.
- We present a compare-split rule to balance the need to deliver the message to multicast destinations quickly while keeping the number of forwardings low.
- We develop an optimal split process at each branch of the multicast tree.
- We evaluate the proposed scheme not only in synthetic traces, but also in real mobility traces. The simulation results show the good performance of the compare-split scheme in DTN multicasting.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 is the preliminary work. Section 4 presents an overview of our multicasting scheme. Section 5 provides some other methods. Section 6 analyzes these protocols. Section 7 focuses on the simulation and evaluation. We summarize the work in Section 8.

## 2. Related Work

Many multicast protocols have been proposed to address the challenge of the frequent topology changes in mobile ad hoc networks [11] and [12]. In general, there are two types of multicasting protocols: *tree-based* and *mesh-based*. In tree-based approaches, either the *source-tree-based* (such as multicast extensions to open shortest-path first (MOSPF) [13], protocol independent multicast (PIM) [14], distance vector multicast routing protocol (DVMRP) [15], and multicast on-demand distance vector routing protocol (MAODV) [16]) or *shared-tree-based* (core based tree (CBT) [17]) approaches are used. The former one constructs a multicast tree among all of the member nodes for each source node; usually this is a shortest path tree. This kind of protocol is more efficient for the multicast, but has too much routing information to maintain and has less scalability. The latter one constructs only one multicast tree for a multicast group including several source nodes. The mesh-based method, on-demand multicast routing protocol (ODMRP) [18], and forwarding group multicast protocol (FGMP) [19], are more robust through redundant paths. Almost all protocols are based on building an infrastructure (tree or mesh).

There have been recent works which consider multicasting in DTNs. In the *single node* (also called *ferry*) *model*, one single node holds all destinations and delivers them to each destination at the contacts through movement. In [6], Zhao, Ammar, and Zegura proposed the basic single node model together with new semantics for DTN multicasting, which explicitly specify temporal constraints on group membership and message delivery. Yang and Chuah [7] presented a two-stage single node model, where routes to destinations are first identified through a ferry, followed by the message delivery along the discovered routes. In [20], Wang, Li, and Wu studied a dynamic version of the single node model. Although there is only one single node that holds all destinations, the message holder will only forward the message to a node that has a higher quality. This approach is an extension of the *delegation forwarding* [21] used in DTN multicasting.

In the *multiple copies model*, the destination set is replicated at a contact once a certain condition related to the quality of the encountered node is satisfied. In [20], the message holder (for a particular destination) will replicate a copy to an encountered node that has a higher quality with respect to the destination. The number of copies can be controlled using a ticket-based scheme [22]. In [23], Lee et al. proposed RelayCast, a routing scheme that extends the two-hop relay algorithm in the multicast scenario, which considers the  $k$ -copy replication scheme where a packet is replicated to  $k$  relay nodes. In [8] and [9], the number of tickets ( $L$  initially) is divided into halves for each forwarding. The *single copy model* is similar to the multiple copies model. The difference is that the original node does not maintain a copy. That is, there is only one copy for each destination. In [10], Gao et al. developed a single copy model where the forwarding metric is based on the social network perspective.

In [8] and [9], Spyropoulos et al. also dealt with the situation of when the number of tickets is reduced to one: *spray-and-wait* and *spray-and-focus*. In the spray phase, for every message originating at a source node,  $L$  message copies are initially spread - forwarded by the source and possibly other nodes receiving a copy - to  $L$  distinct relays. In the next phase, *wait* means that the holder will forward the message only to its destination, while *focus* means a message can be forwarded to a different relay according to a given forwarding criterion.

In recent years, biologists have found that *Levy walks* can be commonly used to describe the mobility patterns of foraging animals [24], [25], [26]. Computer scientists also paid attention to this area of research. They stud-

ied *Levy walks* in human mobility [27], [28], [29], which can help us to analyze wireless mobile networks such as DTNs. From [28], in wireless mobile networks, human mobility patterns have features defining *Levy walks*; their *flight*<sup>2</sup> distributions and pause time distributions closely match truncated power-law distributions. The *mean squared displacement* (MSD)<sup>3</sup> also shows significant influence on these mobility patterns.

In this paper, we first propose and apply the destination set splitting in DTN multicasting, which is based on the single copy model. Our methods are all based on the tree structure (but not necessarily shortest in the contact graph) in order to reduce the number of forwardings and latency. When there is only one destination in the destination set, we apply *wait* and *focus* schemes in our solutions.

### 3. Preliminaries

#### 3.1. Objectives

The objective of this paper is to develop an efficient single-copy multicasting scheme in DTNs. Single-copy multicasting reduces the storage requirement of each node. Two performance metrics are used: (1) *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. Efficient multicast means having a fewer number of forwardings and a smaller latency.

#### 3.2. System Models

Assume that there are  $N$  nodes in the whole network. The destination set of a multicast is represented as  $D = \{1, 2, \dots, n\}$ . Each node  $a$  is associated with a contact rate vector  $(f_1^a, f_2^a, \dots, f_n^a)$ , where  $f_i^a$  indicates the frequency that node  $a$  meets destination  $i$  in a given period  $T$ .  $f_i^a$  is also called *contact rate level* for destination  $i$  in period  $T$ . We use  $(c_1^a, c_2^a, \dots, c_n^a)$  to indicate the number of times that node  $a$  meets destination  $i$  in a given period  $T$ . Hence, the contact rate level  $f_i^a$  can be presented as follows:

---

<sup>2</sup>A *flight* is defined to be a longest straight line trip from one location to another that a particle makes without a directional change or pause.

<sup>3</sup>MSD is defined to be the variance of the displacement probability distribution. Here it is a discrete probability distribution.

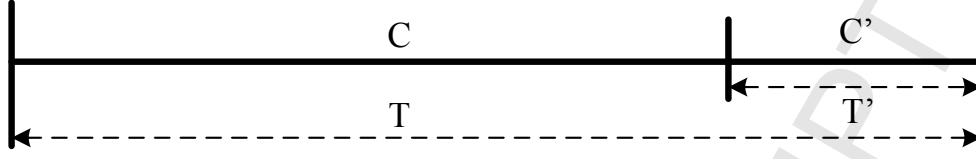


Figure 1: An illustration of the system models.

$$f_i^a = \frac{c_i^a}{T}. \quad (1)$$

In order to involve the effect from recent information, we also define  $T'$ , which is considered as a recent period.  $(c_1^{a'}, c_2^{a'}, \dots, c_n^{a'})$  indicates the frequency that node  $a$  meets destination  $i$  in a given period  $T'$ . Hence, we define the contact rate level  $f_i^a$  as follows:

$$f_i^a = w \cdot \frac{c_i^{a'}}{T'} + (1 - w) \cdot \frac{c_i^a - c_i^{a'}}{T - T'} \quad (2)$$

where  $w$  is the weight of the recent information for the contact rate level.

The *active level* of node  $a$ :  $A_a$  is the number of total contacts per time unit  $T$  that node  $a$  meets with all other nodes in the network.

$$A_a = \sum_{i=1}^N f_i^a. \quad (3)$$

Fig. 1 illustrates the system models considering the recent information.

### 3.3. Challenges and Main Ideas

Contact rate level indicates the contact frequency of reaching a particular destination without further forwarding, while active level indicates the likelihood of contacting other nodes to enhance the contact rate level through forwarding. The challenges lie in the balancing of these two factors when two nodes meet. In our single-copy multicast, the key is to decide when and how a split should occur in constructing a multicast tree.

In this paper, we propose a *compare-split* scheme at each contact during the construction of a dynamic multicast tree. The first step is the *compare* part, which determines when a split should occur. When node  $a$ , with a destination subset, has a contact with node  $b$ , without any destination subset, we set the condition for splitting as follows: a split occurs when the sum of



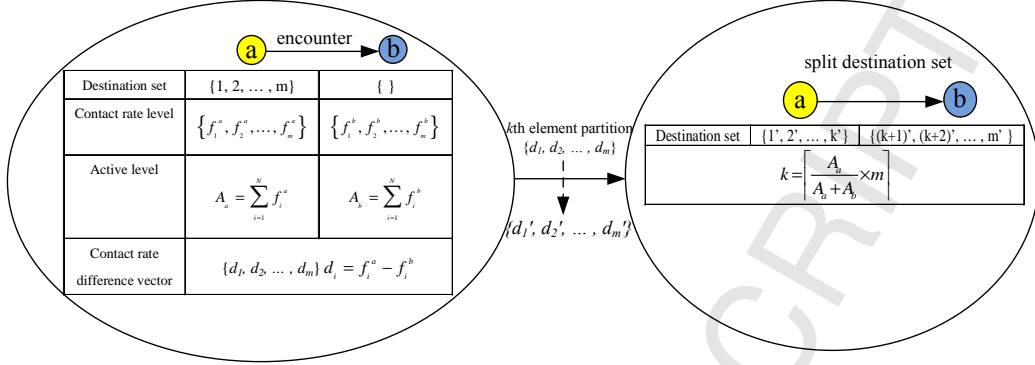


Figure 2: An illustration of ratio-based-split.

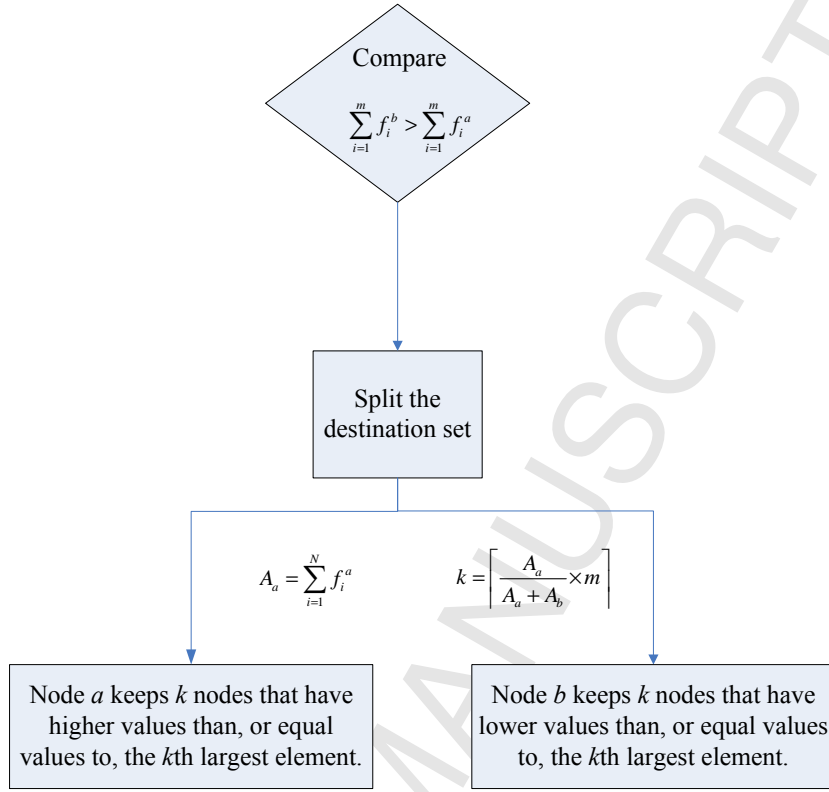
the contact rate levels for all destinations associated with  $b$  is higher than the one associated with  $a$ . The second step is the *split* part, which decides how a split should be done. We propose a *ratio-based-split (RS)*, which splits the destination subset based on active levels of two encountered nodes. We then present an *optimal split algorithm*, which splits the destination subset based on the calculated ratio such that the combined sum of contact rate levels at nodes  $a$  and  $b$  are maximized.

#### 4. Compare-split

In this paper, we propose a *compare-split* scheme at each contact during the construction of a dynamic multicast tree. In this section, we will present the two steps of this method and give an example to explain the whole process. The first step is “compare”, which determines whether a split should occur. The second step is “split”, which decides how a split should be done.

##### 4.1. Compare

The first step for our non-replication multicasting scheme is *compare*. When node  $a$ , with a subset of destinations  $D' \subseteq D$  (as shown in Fig. 2,  $m$  is the size of a subset  $D'$  of destination set  $D$ ,  $n$  is the size of destination set  $D$ ), has a contact with a new node  $b$ , without any destination subset, node  $a$  will first send  $D'$  information to node  $b$  and nodes  $a$ , and  $b$  exchange their contact rate vectors,  $(f_1^a, f_2^a, \dots, f_m^a)$  and  $(f_1^b, f_2^b, \dots, f_m^b)$ , upon their contact.  $m$  is the size of the subset  $D'$  of destination set  $D$ . After comparing these two

Figure 3: An illustration of *compare-split*.

nodes' sum of the contact rate levels for all destinations, if  $\sum_{i=1}^m f_i^b > \sum_{i=1}^m f_i^a$ , then go to the next *split* step.

Note that two rounds of exchanges are used. One round can be saved by exchanging  $(f_1^a, f_2^a, \dots, f_n^a)$  and  $(f_1^b, f_2^b, \dots, f_n^b)$ .  $(f_1^a, f_2^a, \dots, f_m^a)$  and  $(f_1^b, f_2^b, \dots, f_m^b)$  can then be extracted locally.

#### 4.2. Split

The second step is to *split* the destination set. Suppose that  $d_i = f_i^a - f_i^b$  is called the *contact rate difference* between nodes *a* and *b* for destination *i*. The active levels  $A_a$  can be denoted by the number of total contacts that node *a* meets with all other nodes.

$$A_a = \sum_{i=1}^N f_i^a \quad (4)$$

The destination set splitting is based on the ratio of two encountering nodes' active levels. The ratio  $k$  can be denoted as:

$$k = \lceil \frac{A_a}{A_a + A_b} \times m \rceil \quad (5)$$

1. Both  $a$  and  $b$  generate the contact rate difference vector  $(d_1, d_2, \dots, d_m)$ . Find the  $k$ th largest element in  $O(m)$  operations using a general *selection algorithm* [30].
2. Node  $a$  keeps  $k$  nodes that have higher values than, or equal values to, the  $k$ th largest element. In the case of a tie, when two contact rate differences are equal, the node ID is used to break the tie.
3. Node  $b$  keeps  $m - k$  nodes that have lower values than, or equal values to, the  $k$ th largest element.

In step (1), the optimal linear solution is used to find the  $k$ th largest element. The whole split process is shown in Fig. 2.

#### 4.3. Example

Fig. 3 illustrates the whole process of our proposed compare-split method. Next we can use Fig. 4 as an example. Node  $a$ , with a subset of destinations  $D' = \{1, 2, 3, 4, 5\}$ , makes contact with node  $b$ , without any destination subset. First, node  $a$  sends  $D'$  to node  $b$ , and they exchange their contact rate vectors:  $(f_1^a, f_2^a, \dots, f_5^a) = (5, 2, 13, 8, 15)$  and  $(f_1^b, f_2^b, \dots, f_5^b) = (3, 6, 10, 11, 14)$ . After the calculations, we have  $\sum_{i=1}^5 f_i^b = 44$  and  $\sum_{i=1}^5 f_i^a = 43$ . Hence, the sum of the contact rate levels for all destinations associated with  $b$  is higher than the one associated with  $a$ . Then, we go to the second step. The active levels of node  $a$  and  $b$  are 100 and 90, respectively.

We first calculate the *contact rate difference vector*:

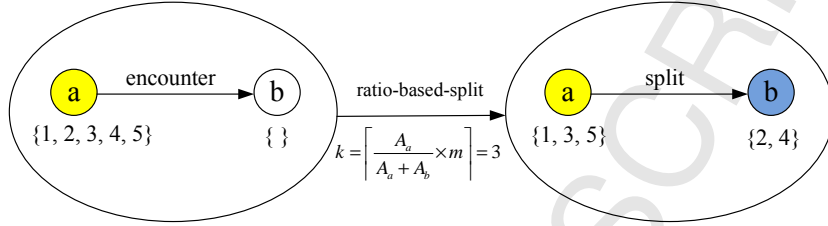
$$(d_1, d_2, \dots, d_5) = (2, -4, 3, -3, 1)$$

and *ratio*:  $k = \lceil \frac{A_a}{A_a + A_b} \times m \rceil = 3$ .

Then, we use the selection algorithm to find the *third* largest number in the contact rate difference vector, which is 1.

After splitting the destination set, node  $a$  keeps 3 destinations:  $\{1, 3, 5\}$ , and destinations 2 and 4 will be assigned to node  $b$ . The *combined contact rate* of node  $a$  and  $b$  is  $f_1^a + f_3^a + f_5^a + f_2^b + f_4^b = 50$ , which is larger than

destination	1	2	3	4	5	Active level
<b>a</b>	5	2	13	8	15	100
<b>b</b>	3	6	10	11	14	90
<b>d</b>	2	-4	3	-3	1	

Figure 4: An example of *ratio-based-split*.

$\sum_{i=1}^5 f_i^a = 43$ . This means that using the compare-split algorithm can increase the contact frequency of meeting with the destinations.

In contrast, the usual greedy way of the splitting process is as follows: (1) possible split 1: node  $a$  will keep the 3 largest contact rate level destinations and assign all other destinations to node  $b$ . In this example, node  $a$  will keep destinations  $\{3, 4, 5\}$  and assign destinations 1 and 2 to node  $b$ . After this process, the combined contact rate of nodes  $a$  and  $b$  is  $f_3^a + f_4^a + f_5^a + f_1^b + f_2^b = 45$ , which is smaller than using the compare-split algorithm; (2) possible split 2: node  $b$  will get the 2 largest contact rate destinations, and node  $a$  keeps the rest. Hence, after splitting, node  $a$  keeps destinations  $\{1, 2, 3\}$ , and node  $b$  keeps destinations 4 and 5. After this process, the combined contact rate of nodes  $a$  and  $b$  is  $f_1^a + f_2^a + f_3^a + f_4^b + f_5^b = 45$ , which is also smaller than the result we get from the compare-split algorithm.

## 5. Implementation & Extensions

There are many other methods that can be implemented in the *compare-split* rule. First, we will explain some conditions in the *compare* phase. Then, we provide three other schemes when splitting the destination set: *random-binary-split (RBS)*, *median-binary-split (MBS)*, and *priority-based-split (PS)*. Finally, we will present two methods: *wait* and *focus* [8], [9], when there is only one destination in the destination subset.

<i>Split</i>
ratio-based-split (RS)
random-binary-split (RBS)
median-binary-split (MBS)
priority-based-split (PS)

Table 1: Different split methods.

### 5.1. Compare

In the previous section, we use the threshold-based condition (when node  $b$  has a higher sum of the contact rate levels for all destinations than node  $a$ , a split will occur) in the *compare* step. Also, we don't have to use any conditions in the first step. We will compare these two methods in our simulation.

Another method is: if node  $b$  already has a destination subset, node  $a$  and node  $b$  will combine their destination sets, then split. It will increase the number of forwardings. We will also compare this method with our scheme in our simulation.

### 5.2. Split

In the split step, we also have many other schemes: *binary-split (BS)* (*random-binary-split* and *median-binary-split*) and *priority-based-split*.

#### 5.2.1. Binary-split (BS)

In binary-split, we will not consider *active level*. The destination split will be *equal partition*. The BS process is shown in Fig. 5: nodes  $\{a, b, c, d, e\}$  are relay nodes, and nodes  $\{1, 2, 3\}$  are destination nodes. When one node meets a destination, it will first assign this destination to it and then use the binary-split.

- *random-binary-split (RBS)*: after meeting with node  $b$ , node  $a$  will give half of the destination subset  $D'$  to  $b$  randomly. This means node  $a$  keeps  $\lceil m/2 \rceil$  nodes, and node  $b$  keeps  $\lfloor m/2 \rfloor$  nodes.
- *median-binary-split (MBS)*: in RBS, message holder  $a$  partitions the destinations randomly. It may assign a destination to a node with a small contact rate level to this particular destination. Hence, the multicast process will have a large latency. We use another equal partition

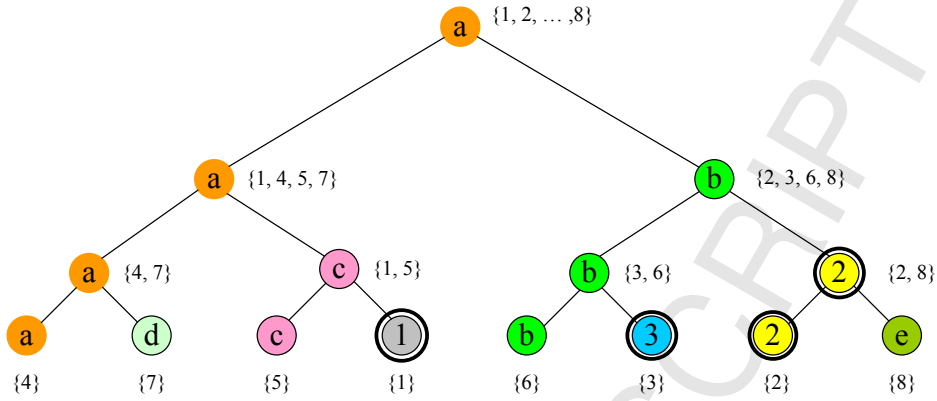


Figure 5: A sample of binary-split.

which is based on contact rate difference. We use the *median of medians algorithm* [30]: a linear solution to find the median of the contact rate difference vector. Then, node  $a$  keeps  $\lceil m/2 \rceil$  nodes that have higher values than, or equal values to, the median, and node  $b$  keeps  $\lfloor m/2 \rfloor$  nodes that have lower values than, or equal values to, the median.

MBS can be viewed as a special case of RS when the active levels of two encounter nodes are approximately the same.

### 5.2.2. Priority-based-split (PS)

Another solution is for node  $a$  to keep the destinations with their contact rate difference values higher than 0 and to assign all other destinations to node  $b$ . This means that only the destinations with higher contact rate levels in node  $b$  than in node  $a$  will be assigned to node  $b$ .

Priority-based-split is shown in Fig. 6. Initially, node  $a$  takes 8 destinations  $\{1, 2, \dots, 8\}$ . In the *split* phase, the copy arrives at destinations  $\{1, 2, 3\}$  and destinations are assigned to nodes  $\{b, c, d, e, f, g\}$ .

### 5.3. Wait and Focus

When there is only one destination that is carried by node  $a$ , we also have two strategies for forwarding decisions, as in [8] and [9]:

- *wait*: Node  $a$  will keep this destination until it meets the particular destination.
- *focus*: Node  $a$  will assign this destination only to a node which has a high contact rate value for this destination.

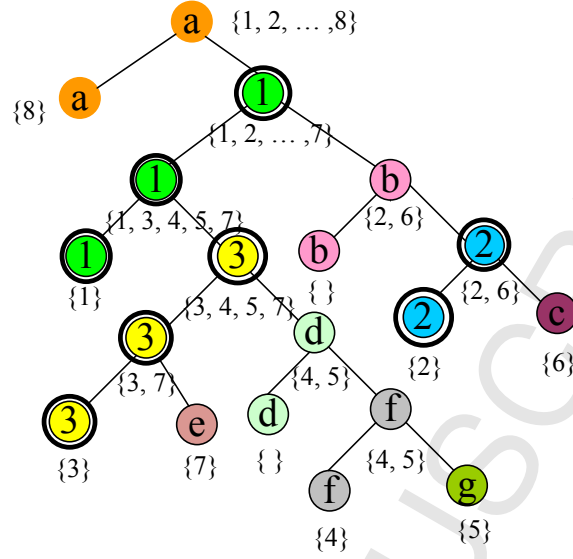


Figure 6: A sample of priority-based-split.

## 6. Analysis

In this section, we will explain the optimal split process at each branch of the multicast tree. Then, we analyze the benefit considering contact rate level and active level at the same time. Finally, we compare the difference among single node, single copy, and multiple copies models.

### 6.1. Optimal Split Algorithm

Our major goal in using the non-replication multicasting scheme in DTNs is to ensure that the delivery of multicast information is done over different paths. Each path has a relatively high contact frequency of reaching the corresponding destination subset quickly. Then, multiple holders for destination nodes can search for destinations in parallel. These solutions can reduce the multicast cost. The number of forwardings is a major metric to measure the cost of the multicasting process. Compare-split can also reduce the latency in DTN multicasting.

Suppose that  $D_a$  is the destination subset kept in node  $a$  and that  $D_b$  is the destination subset assigned to  $b$ , we would like to maximize the *combined contact rate* of  $a$  and  $b$  as follows:

$$\max\left\{\sum_{i \in D_a} a_i + \sum_{j \in D_b} b_j\right\}. \quad (6)$$

**Theorem 1.** *Suppose that  $D_a$  and  $D_b$  are two subsets as a result of  $k$ th element partition.  $d_i = f_i^a - f_i^b$  is called contact rate difference between nodes  $a$  and  $b$  for destination  $i$ . Maximum combined contact rate of visiting any of the destinations within a time period occurs when for each  $i \in D_a$  and  $j \in D_b$ ,  $d_i \geq d_j$ .*

**Proof.** It is clear that any other partition (including the optimal one) can be generated through a sequence of swaps between two elements, one each from  $D_a$  and  $D_b$ . We show that each swap will deteriorate the combined contact rate level. Suppose  $i$  in  $D_a$  and  $j$  in  $D_b$  are swapped. Based on condition  $d_i \geq d_j$ , we have  $f_i^a - f_i^b \geq f_j^a - f_j^b$ , or  $f_i^a + f_j^b \geq f_i^b + f_j^a$ .

Note that  $f_i^a + f_j^b$  is the combined contact rate involving destinations  $i$  and  $j$ , whereas  $f_j^b + f_i^a$  is the combined contact rate after the swap of  $i$  and  $j$ . The theorem follows.  $\square$

This optimal split algorithm can partition the destinations to nodes with higher contact rate levels; hence, it can reduce the number of forwardings and latency in DTN multicasting.

## 6.2. Contact Rate Level and Active Level

Both the contact rate level and active level can be estimated based on past contacts. In fact, each mobile node can start with a predefined default value for both contact rate level and active level. It then iteratively enhances its estimates based on new contacts.

In this part, we analyze the necessity using contact rate level and active level together for compare-split. We will use a multicast with two destinations, *black* and *white* nodes, as an example to illustrate. Initially, node  $a$  holds both destinations. Consider that  $a$  is associated with a tape  $T_a$  of a sequence of numbered slots that hold contacts node  $a$  has with other nodes.

1) Case 1: select  $T_a$  with four randomly selected distinct slots - two for black and two for white. The process is called node's  $T$  assignment. To see the reason of having the same condition (both for contact rate level and active level), it is still better to split both destinations between nodes  $a$  and  $b$  than to let  $a$  keep both. We compare the following two approaches. The completion time for the non-split case is the maximum slot number of the first white node and the first black node in node  $a$ 's tape ( $T_a$ ). The completion time of the split case is the maximum slot number of the first white node's slot number in  $T_a$  and the first black node's slot number in  $b$ 's tape ( $T_b$ ). The latter has a shorter expected delivery time.



2) Case 2: to view the importance of the contact rate level during a split, consider a case where  $T_a$  has three black slots and one white slot, while  $T_b$  has one black slot and three white slots. Both nodes  $a$  and  $b$  have the same activity level, and we can easily extend the argument from Case 1 to the fact that it is better to split. It is obviously better to assign the black destination to node  $a$  and the white destination to node  $b$ . Therefore, the priority-based-split algorithm is important as each node ( $a$  or  $b$ ) will increase its chance to reach the corresponding destination directly, resulting in a smaller latency. A larger contact rate level will also reduce the number of forwardings as its contact rate level is more difficult to be surpassed.

3) Case 3: to view the importance of the active level during a split, consider  $T_a$  with two black slots, two white slots, and four red slots, and  $T_b$  with two black, two white, and no other slots. Although both nodes  $a$  and  $b$  have the same contact rate levels to both destinations, node  $a$  is twice as active as node  $b$ . In this case,  $a$  has contacts with non-destination nodes (red slots) which may have a better contact with destination  $a$  or  $b$ . In other words, destination(s) associated with  $a$  will have a chance to be forwarded to a third node with a better contact rate level to  $a$  and/or  $b$ . Therefore, it is better to assign both destinations to node  $a$ , assuming the benefit from the active status outweighs the benefit from the split (as in Case 1).

### 6.3. Single Node, Single Copy, and Multiple Copies Models

The single node model uses the minimum number of forwardings (in fact, it is the same as the number of destinations). The delivery ratio can be an issue if the holder has a very low contact rate level to a particular destination. Improvement includes creating a delegation when an encountered node that has better contact rate levels to all destinations. Like the single node model, the single copy model also keeps one copy for each destination, but it allows many holders. The number of forwardings is moderate as each copy is forwarded only when there is a better condition (based on the contact rate levels). Latency is an issue; however, it can be easily traded with the delivery ratio as the destination set is quickly partitioned to subsets with only a single node. Each holder can judiciously determine whether and when to terminate a delivery process.

The multiple copies model includes flooding, which copies the destination set at each node encountered. It is the fastest approach, but it incurs a sufficient number of copies per destination. The number of copies can be controlled through delegation (i.e., copy destination set only to ones with

Trace	Number of nodes	Number of destinations
Levy walks model	100	2, 4, 8, 16, 32
Gaussian distribution model	100	2, 4, 8, 16, 32
Intel trace	118	2-8
Cambridge trace	211	2-11

Table 2: Simulation parameters

a better condition). It still has  $\frac{5}{3}\sqrt{N}$  ( $N$  is the total number of nodes in the network) [21] number of forwardings, even for a destination set with one destination. TTL-based or ticket-based approaches can control the number of copies, but it is still a challenge to have a good estimate for TTL and ticket numbers to assure delivery while controlling the number of copies. Excessive copies also consume limited memory space at each node, which can prevent and limit the support of multiple flows.

## 7. Simulation

In this section, we compare the performances of the schemes we mentioned in the previous sections. Each simulation is repeated 1,000 times in MATLAB. In our simulation, the 90 percent confidence interval of each result is within  $\pm 1$  percent. The following metrics are calculated in our simulation.

1. *Average cost*: the average number of forwardings for all destinations to receive the multicast message.
2. *Average latency*: the average latency for all of the delivered destinations to receive the multicast message.
3. *Average latency  $\times$  average cost*: the average latency  $\times$  average cost for all of the delivered destinations to receive the multicast message.

We will compare the multicasting schemes both in synthetic and real traces.

### 7.1. Simulation Methods and Setting

We have used the traces, not only in synthetic mobility models, but also from real traces. We will compare the number of forwardings, latency, and their product in each trace.

In this paper, we consider that the given period  $T$  is the whole period, and  $T'$  is 10% of  $T$ .

Our simulation is based on two situations:

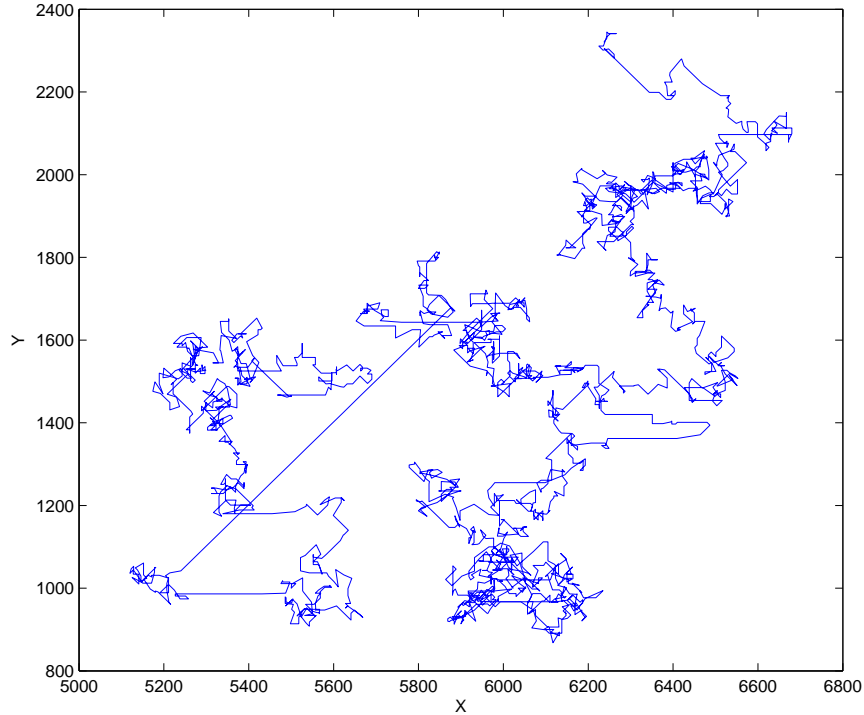


Figure 7: Levy walks model.

- **Without considering recent information (N-RI):** using equations (1) and (3) to calculate the contact rate level and active level, which does not consider the recent period information;
- **Considering recent information (RI):**  $w = 0.5$  in equations (2) and (3), which gives more weight to the recent information.

#### 7.1.1. Synthetic Mobility Models

In the synthetic mobility models, we set up a 100-node environment. We set up two synthetic traces: Levy walks and Gaussian distribution models.

(a) **Levy walks model:** from the simulation results in [28], a Levy distribution with a scale factor  $c$  and exponent  $\alpha$  in terms of a Fourier transformation can be defined as the following:

split	No condition	Threshold-based
median-binary-split (MBS-W)	1	5
random-binary-split (RBS-W)	2	6
ratio-based-split (RS-W)	3	7
priority-based-split (PS-W)	4	8

Table 3: Compare-Split-Wait legend of simulation results

split	No condition	Threshold-based
median-binary-split (MBS-F)	9	13
random-binary-split (RBS-F)	10	14
ratio-based-split (RS-F)	11	15
priority-based-split (PS-F)	12	16

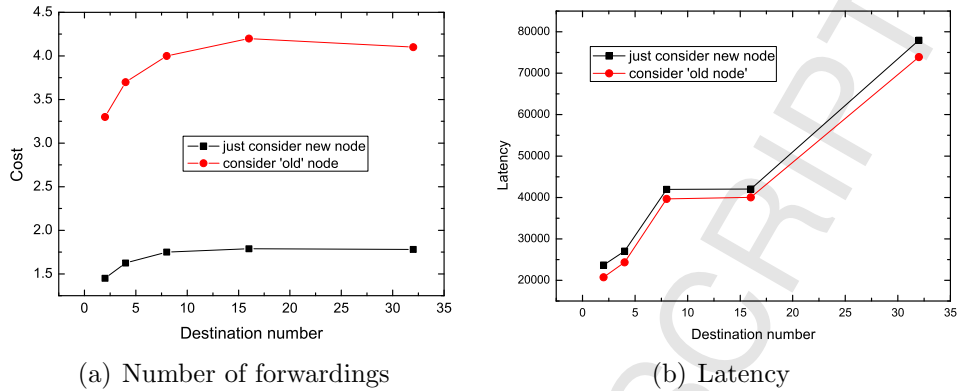
Table 4: Compare-Split-Focus legend of simulation results

$$f_X(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-itx - |ct|^\alpha} dt, \quad (7)$$

where  $\alpha$  is the Levy exponent for flight length distribution, which follows power-law distribution:  $p(l) \sim \frac{1}{l^{1+\alpha}}$ ,  $0 < \alpha \leq 2$ . A power law distribution of pause times is denoted by  $\psi(\Delta t_p) \sim 1/\Delta t_p^{1+\beta}$ , where  $\beta$  is the Levy exponent for pause time distribution,  $0 < \beta \leq 2$ .

Each time two nodes make a contact with each other, we give a contact time and a GPS address. In the Levy walks mobility pattern, we set up the Levy exponent for flight length distribution  $\alpha$ , which is 1, and the Levy exponent for pause time distribution  $\beta$ , which is also 1 in our simulation [31]. Fig. 7 shows the mobility pattern of Levy walks. The active level and contact rate levels can be calculated from the generated trace. Because we plan to examine the performance of equal partitioning, we set the destination numbers as  $2^i, i \in \{1, 2, 3, \dots\}$ .

**(b) Gaussian distribution model:** in this model, we first randomly select a node's active level based on a Gaussian distribution model with  $\mu = 5,000$  and  $\sigma = 3,000$ . Once the active level of a node is selected, the active level is partitioned into contact rate levels to all nodes. Suppose node  $a$ 's contact rate level to node  $b$  is  $k$ , then in  $a$ 's  $T$ ,  $k$  slots are randomly selected. The destination number setting and measuring parameters are the same as the Levy walks model.

Figure 8: Comparison of two *compare* methods.

### 7.1.2. Real Traces

We use Intel and Cambridge traces [32] in our simulation. These data sets consist of contact traces between short-range Bluetooth enabled devices carried by individuals.

**(a) 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. Their contacts are random and the nodes' active levels and contact rate levels are also random. In our simulation, we randomly set one of these 9 nodes as the source, and we choose other different nodes as the destinations. The number of destinations is from 2 to 8. We will make comparisons of the number of forwardings and latency in these different partition models.

**(b) Cambridge trace:** this trace includes Bluetooth sightings by groups of users carrying small devices (iMotes) for six days in the Computer Lab at 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. Their contacts are random and the nodes' active levels and contact rate levels are also random. In our simulation, we set 1 node as the source and choose different nodes as the destinations. The number of destinations is from 2 to 11. We will also compare the number of forwardings and latency, as in the Intel trace.

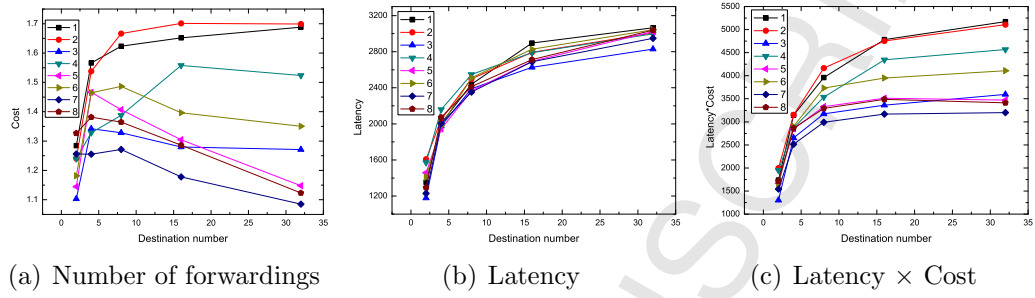


Figure 9: Comparison in Levy walks model: compare-split-wait in N-RI.

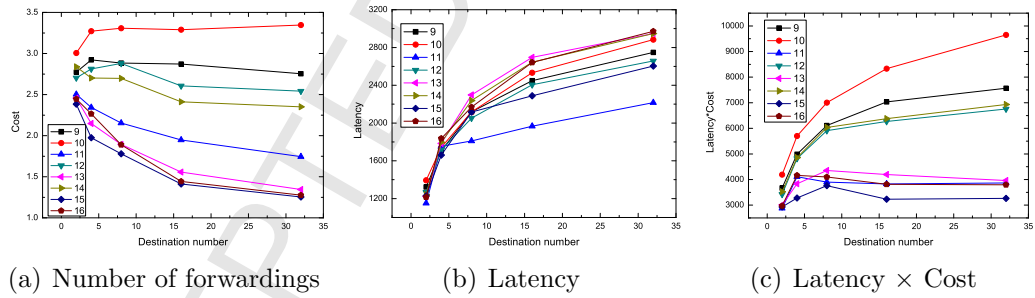


Figure 10: Comparison in Levy walks model: compare-split-focus in N-RI.

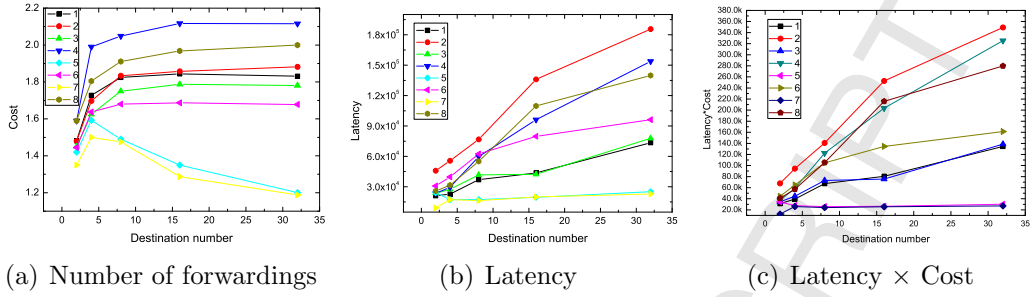


Figure 11: Comparison in Gaussian distribution model: compare-split-wait in N-RI.

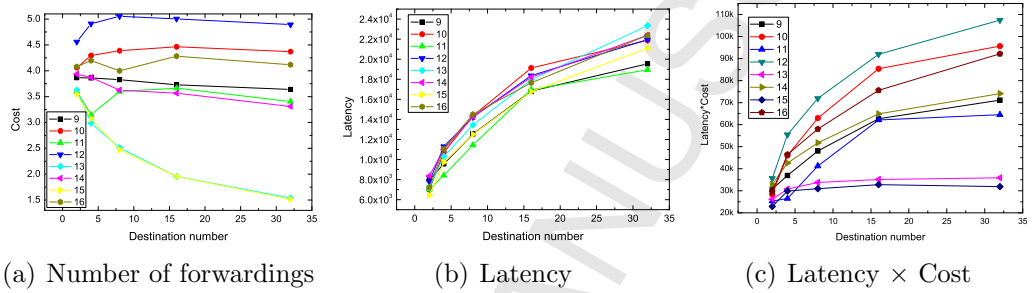


Figure 12: Comparison in Gaussian distribution model: compare-split-focus in N-RI.

## 7.2. Simulation Results

### 7.2.1. Compare

As we mentioned in Section 5, if one node has a contact with a node which already has a destination subset, we propose another method: that these two nodes' destination subsets are combined together and then split. From Fig. 8, we can see that this method increases the number of forwardings compared to our method that just splits the destination subset to a new node; at the same time, it cannot reduce the latency much. Hence, in the rest of this paper, we will not use this method.

We compare the number of forwardings, latency, and their product in 16 multicasting schemes, as shown in Tables 3 and 4.

### 7.2.2. Without considering recent information (N-RI)

In this part, we will compare all schemes without considering recent information based on equations (1) and (3).

#### (a) Results in synthetic mobility models

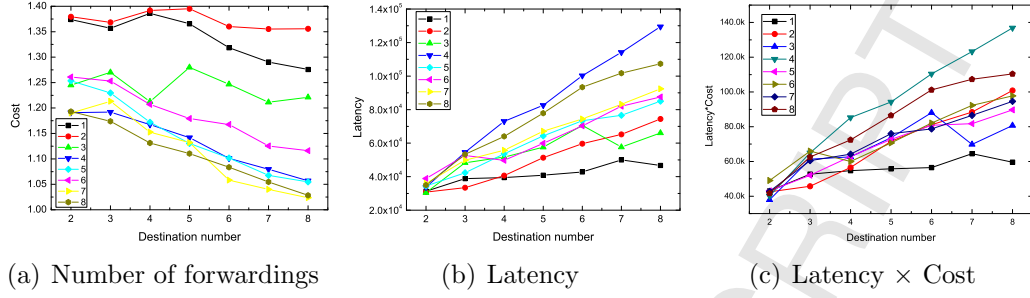


Figure 13: Comparison in Intel trace: compare-split-wait in N-RI.

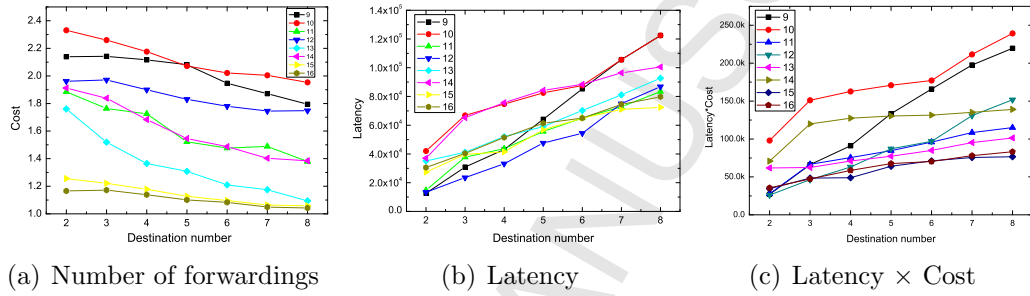


Figure 14: Comparison in Intel trace: compare-split-focus in N-RI.

In the Levy walks model, we compared the number of forwardings, latency, and their product among these 16 solutions, as shown in Figs. 9 and 10. It shows that RS has the fewest number of forwardings and the shortest latency among these four schemes in all conditions (using threshold or not, wait or focus). PS performs better than the other two binary-split schemes, while MBS is better than RBS. We use compare-split-focus with a threshold-based condition in Figs. 10(a), 10(b), and 10(c) to explain. RS-F (Line 15) has about 18% less forwardings than PS-F (16) and 21% less than MBS-F (13) from Fig. 10(a). RS-F reduces the latency by 14% from PS-F and 16% from binary-split in Fig. 10(b). By comparing the product of number of forwardings and latency, we can see from Fig. 10(c) that RS-F performs better than other schemes. Using the threshold-based condition to decide whether to split the destination set can reduce the number of forwardings by about 9.4%. This means using the threshold-based condition can help the message holder to meet higher contact rate nodes. Using the wait scheme can reduce the number of forwardings, while using the focus scheme can reduce the latency.



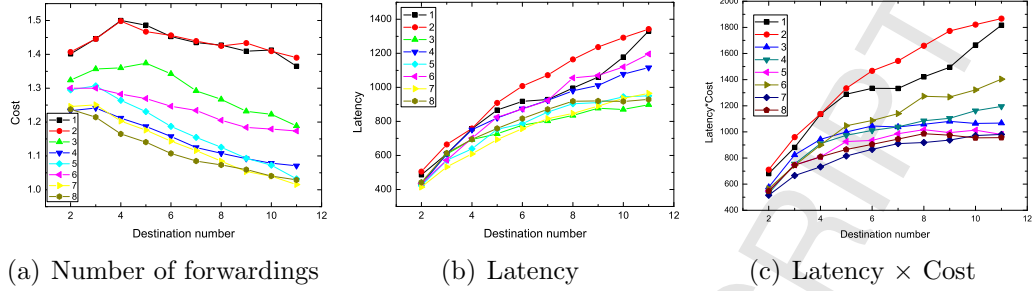


Figure 15: Comparison in Cambridge trace: compare-split-wait in N-RI.

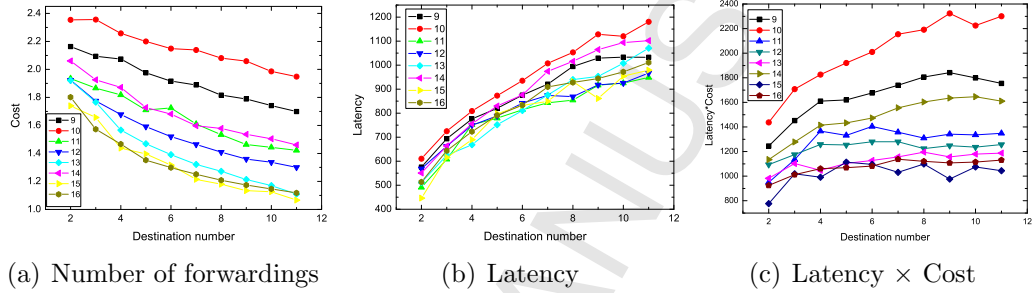


Figure 16: Comparison in Cambridge trace: compare-split-focus in N-RI.

In the Gaussian distribution model, RS and BS perform better than PS, as shown in Figs. 11 and 12. For example, when using compare-split-focus with the threshold-based condition, RS-F(15) has the best performance among these four solutions. Compared with the number of forwardings, it is 2% fewer than MBS-F (13), 16.4% fewer than RBS-F (14), and 33.2% fewer than PS-F (16) from Fig. 12(a). In Fig. 12(b), we know that RS-F has 8% shorter latency than MBS-F, 10% shorter latency than RBS-F, and 12% shorter latency than PS-F in this case. RS and MBS perform better, when comparing the product of the cost and latency, than the other two schemes both in compare-split-focus and compare-split-wait in Figs. 11 and 12. Using the threshold-based condition can reduce the latency by about 2.8% and reduce the number of forwardings by about 6.2% from the no condition in the compare step. Using wait can reduce the number of forwardings by about 60%, while using focus can reduce the latency by about 70% when there is only one destination in the destination subset.

### (b) Results in real traces

In the Intel trace, RS has a similar number of forwardings for each des-

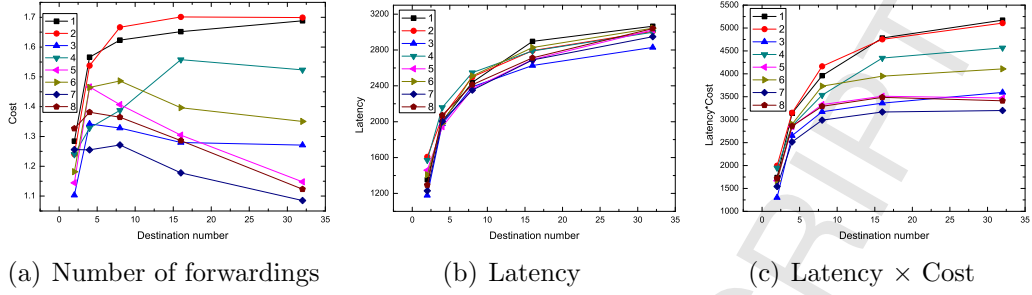


Figure 17: Comparison in Levy walks model: compare-split-wait in RI.

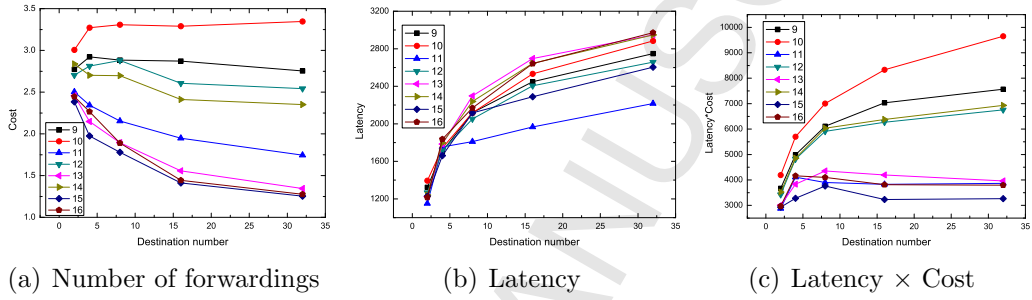


Figure 18: Comparison in Levy walks model: compare-split-focus in RI.

mination as PS, but much shorter latency, about 22% shorter, from Figs. 13 and 14. RS performs better than PS when considering the product of latency and cost, as shown in Figs. 13(c) and 14(c). Using the threshold-based condition in the compare step can reduce the number of forwardings and latency. In the final step, when we want to reduce the number of forwardings, we can choose wait, and if we want to reduce the delay, we can use the focus scheme.

In the Cambridge trace, RS and PS have similar performances in Figs. 15 and 16. RS has shorter latency while PS has a fewer number of forwardings and a smaller product of latency and cost. These two schemes are both better than BS.

### 7.2.3. Considering recent information (RI)

In this part, we will compare all schemes by considering recent information, based on equations (2) and (3) where  $w$  is 0.5.

From Figs. 17 and 18, we can see our design split schemes perform similarly as in N-RI, as shown in Figs. 9 and 10. However, we take more into account recent information than in other schemes, which can provide more

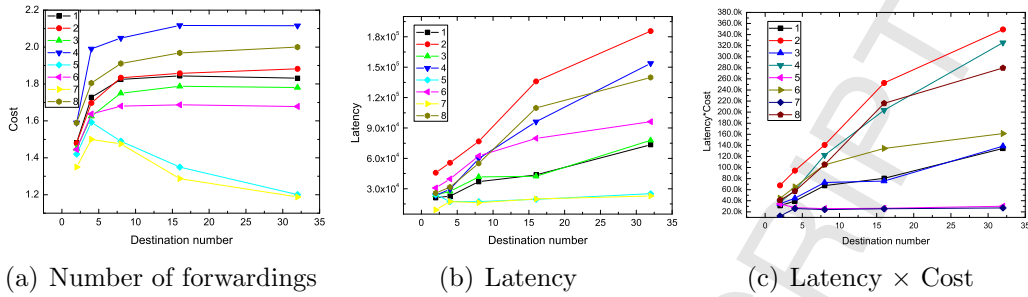


Figure 19: Comparison in Gaussian distribution model: compare-split-wait in RI.

information to the Levy walks model; hence, TI reduces the number of forwardings by about 8% and the latency by about 5% as compared with other schemes. In Figs. 19 and 20, the results do not change a lot from RI, because the recent information has the same contribution to the active level and contact rate level as the long term information in the Gaussian distribution model. From the real traces, as shown in Figs. 21, 22, 23 and 24, we can see our designed schemes perform similarly as in N-RI. At the same time, in RI, the cost, latency, and their product are reduced compared with N-RI. This means that recent information can present the nodes' mobility pattern better than the previously acquired information.

### 7.3. Summary of Simulation Results

We use non-replication multicasting schemes in DTNs. In the Levy walks model, RS is better than BS as the active levels of the nodes vary significantly. Using RS can assign the destinations to high active level nodes, while BS does not consider the active levels. In the Gaussian distribution model, RS is better than PS as active levels of the nodes are more uniform. This phenomenon is pervasive. In two real traces, the active levels vary significantly. It appears that the role of contact rates and active levels are both very important. Hence, using PS and RS is better than using BS. If the compare step with threshold is used before splitting the destination set, the number of forwardings and latency will both decrease. Table 5 shows the best split method in different models. When there is only one destination in the destination set, using the wait scheme can reduce the number of forwardings while using the focus scheme can reduce the latency. From the comparison of N-RI and RI, we can find that the role of recent information is very important.

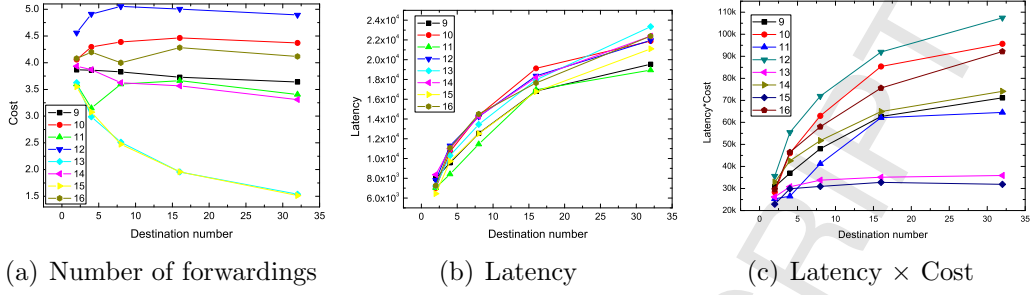


Figure 20: Comparison in Gaussian distribution model: compare-split-focus in RI.

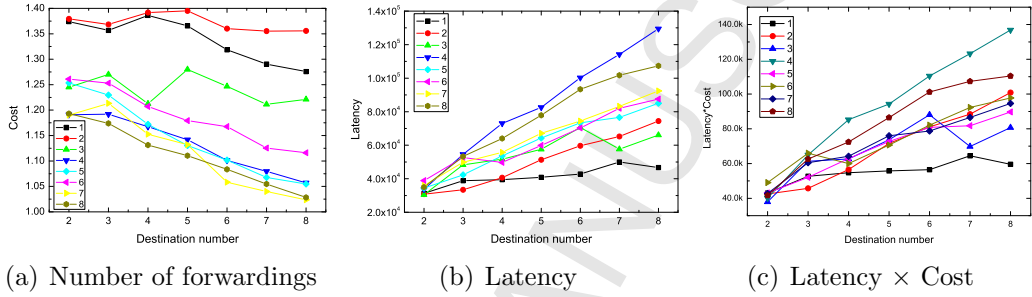


Figure 21: Comparison in Intel trace: compare-split-wait in RI.

## 8. Conclusion

In this paper, we focused on developing a non-replication multicasting scheme in DTNs. Our *compare-split* scheme is based on the single copy model with the objective to reach destinations quickly while minimizing the total number of forwardings. We proposed using the node *active level* together with the *contact rate level* to determine when and how to split a destination set during a contact. The split will occur when the message holder has a contact with a node with the sum of the contact rate levels for all destinations being higher than the message holder. In the split process, we used *ratio-based-split* to split the destination set, then compared it with *random-binary-split*, *median-binary-split*, and *priority-based-split* schemes. When there is only one destination left in the destination set, we used *wait* or *focus* to forward the message to the destination.

We compared the performance of these schemes both in synthetic traces and in real traces. Trace driven simulation results showed that compare-split with ratio-based-split, which considers both the contact rates and active

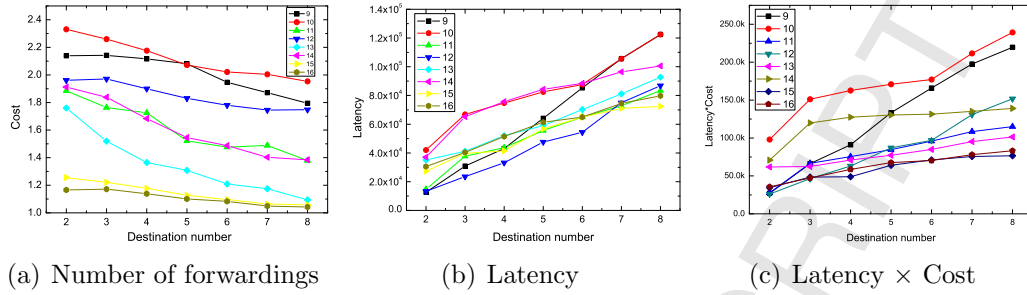


Figure 22: Comparison in Intel trace: compare-split-focus in RI.

Different models	Best split method
Levy walks mode	RS
Gaussian distribution model	RS
Two real traces	RS / PS

Table 5: Conclusion of simulation results

levels, has the best performance. Compare-split-wait has less forwardings while compare-split-focus has shorter latency. We believe that the results obtained from this paper present the first step in exploiting the destination set split rule in single copy DTN multicasting. Future research can benefit from our results by developing specific applications based on the provided schemes in DTNs.

## Acknowledgments

This research was supported in part by NSF grants ECCS 1128209, CNS 1065444, CCF 1028167, CNS 0948184, and CCF 0830289.

- [1] K. Fall, A delay-tolerant network architecture for challenged Internets, in: Proc. of ACM SIGCOMM (2003).
- [2] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, SIGMOBILE Mob. Comput. Commun. Rev. 7 (2003) 19–20.
- [3] C. Liu, J. Wu, Scalable routing in delay tolerant networks, in: Proc. of ACM Mobihoc (2007).
- [4] C. Liu, J. Wu, Routing in a cyclic mobispace, in: Proc. of ACM MobiHoc (2008).
- [5] C. Liu, J. Wu, An optimal probabilistic forwarding protocol in delay tolerant networks, in: Proc. of ACM MobiHoc (2009).

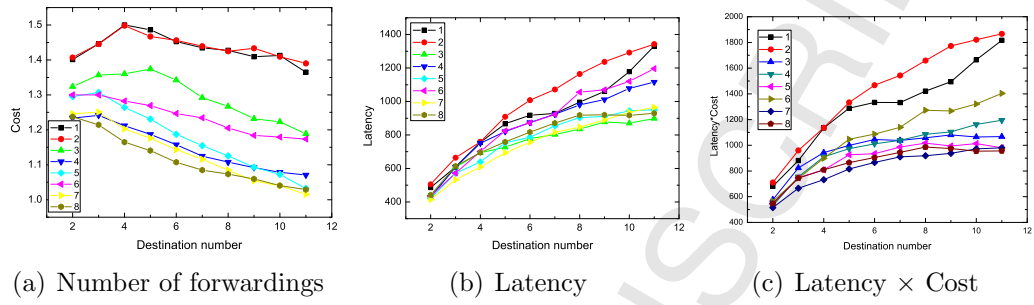


Figure 23: Comparison in Cambridge trace: compare-split-wait in RI.

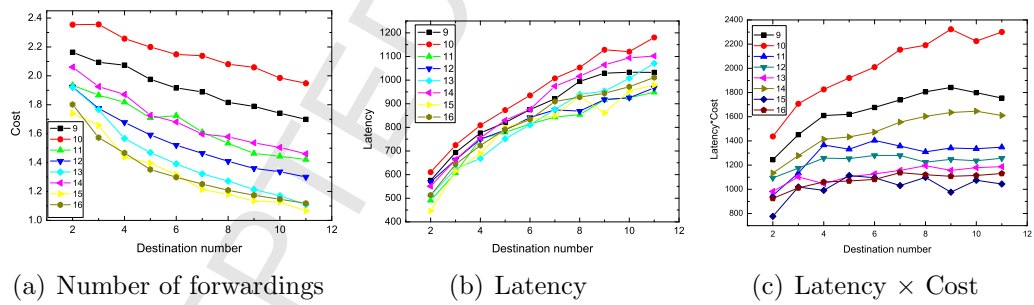


Figure 24: Comparison in Cambridge trace: compare-split-focus in RI.

- [6] W. Zhao, M. Ammar, 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.
- [7] P. Yang, M. Chuah, Efficient interdomain multicast delivery in disruption tolerant networks, in: Proc. of IEEE Mobile Ad-hoc and Sensor Networks (2008).
- [8] T. Spyropoulos, K. Psounis, 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).
- [9] T. Spyropoulos, K. Psounis, C. S. Raghavendra, Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility, in: Proc. of IEEE PERCOMW (2007).
- [10] W. Gao, Q. Li, B. Zhao, G. Cao, Multicasting in delay tolerant networks: a social network perspective, in: Proc. of ACM MobiHoc (2009).
- [11] J. Wu, Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, Auerbach Publications, Boston, MA, USA, 2005.
- [12] S. Yang, J. Wu, New technologies of multicasting in MANET, Design and Analysis of Wireless Networks (2005).
- [13] J. T. Moy, OSPF: Anatomy of an Internet Routing Protocol, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.
- [14] S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, L. Wei, The PIM architecture for wide-area multicast routing, IEEE/ACM Trans. Netw. 4 (1996) 153–162.
- [15] T. Pusateri, Distance vector multicast routing protocol, in draft-ietf-idmr-dvmrp-v3-07 (1998).
- [16] E. M. Royer, C. E. Perkins, Multicast operation of the ad-hoc on-demand distance vector routing protocol, in: Proc. of ACM MobiCom (1999).
- [17] T. Ballardie, P. Francis, J. Crowcroft, Core based trees (CBT), SIGCOMM Comput. Commun. Rev. 23 (1993) 85–95.
- [18] S. J. Lee, W. Su, M. Gerla, On-demand multicast routing protocol in multihop wireless mobile networks, Mob. Netw. Appl. 7 (2002) 441–453.
- [19] C.-C. Chiang, M. Gerla, L. Zhang, Forwarding group multicast protocol (fgmp) for multihop, mobile wireless networks, Cluster Computing 1 (1998) 187–196.
- [20] Y. Wang, X. Li, J. Wu, Multicasting in delay tolerant networks: delegation forwarding, in: Proc. of IEEE Global Communications Conference (GLOBECOM) (2010).
- [21] V. Erramilli, M. Crovella, A. Chaintreau, C. Diot, Delegation forwarding, in: Proc. of ACM MobiHoc (2008).
- [22] S. Chen, K. Nahrstedt, Distributed QoS routing with imprecise state information, in: Proc. of 7th IEEE International Conference on Computer, Communications and Networks (1998), pp. 614–621.
- [23] U. Lee, S. Y. Oh, K.-W. Lee, M. Gerla, Relaycast: Scalable multicast routing in delay tolerant networks, in: Proc. of IEEE International Conference on Network Protocols (ICNP) (2008).
- [24] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, H. E. Stanley, Levy flights search patterns of wandering albatrosses, Nature 381 (1996) 413–415.

- [25] G. Ramos-Fernandez, J. L. Morales, O. Miramontes, G. Cocho, H. Larralde, B. Ayala-Orozco, Levy walk patterns in the foraging movements of spider monkeys (*ateles geoffroyi*), *Behavioural Ecology and Sociobiology* 55 (2004) 223–230.
- [26] R. P. D. Atkinson, C. J. Rhodes, D. W. Macdonald, R. Anderson, Scale-free dynamics in the movement patterns of jackals, *OIKOS* 98 (2002) 134–140.
- [27] M. C. Gonzalez, C. A. Hidalgo, A.-L. Barabasi, Understanding individual human mobility patterns, *Nature* 453 (2008) 779–782.
- [28] I. Rhee, M. Shin, S. Hong, K. Lee, S. Chong, On the levy-walk nature of human mobility, in: *Proc. of IEEE Infocom* (2008).
- [29] M. Kim, D. Kotz, S. Kim, Extracting a mobility model from real user traces, in: *Proc. of IEEE Infocom* (2006).
- [30] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, R. E. Tarjan, Time bounds for selection, *J. Comput. Syst. Sci.* 7 (1973) 448–461.
- [31] NCSU networking research lab: Human mobility models, Downloaded from <http://research.csc.ncsu.edu/netsrv/?q=content/human-mobility-models-download-tlw-slaw>, 2009.
- [32] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, A. Chaintreau, CRAWDAD trace cambridge/haggle/imote/cambridge (v. 2006-01-31), 2006.





Jie Wu is the chair and a professor in the Department of Computer and Information Sciences, Temple University. Prior to joining Temple University, he was a program director at National Science Foundation. His research interests include wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 550 papers in various journals and conference proceedings. He serves in the editorial board of the IEEE Transactions on Computers, IEEE Transactions on Mobile Computing, and Journal of Parallel and Distributed Computing. Dr. Wu is program co-chair for IEEE INFOCOM 2011. He was also general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, ACM WiMD 2009, and IEEE/ACM DCOSS 2009. He also served as panel chair for ACM MobiCom 2009. He has served as an IEEE computer society distinguished visitor. Currently, he is the chair of the IEEE Technical Committee on Distributed Processing (TCDP) and ACM distinguished speaker. Dr. Wu is a Fellow of the IEEE.



Yunsheng Wang received B.Eng. in Electronic and Information Engineering from Dalian University of Technology, Dalian, China, in 2007; the M.Res. in Telecommunication from University College London, London, UK, in 2008. He is currently working toward the Ph.D. degree in the Department of Computer and Information Sciences, Temple University, US. His current research interests include routing in mobile ad hoc networks (MANETs) and delay-tolerant networks (DTNs).

## Research Highlights

- Design contact rate and active levels to guide the construction of a multicast tree.
- Present a compare-split rule to deliver the message to multicast destinations.
- Develop an optimal split process at each branch of the multicast tree.
- Evaluate the proposed scheme both in synthetic traces and real mobility traces.