In delay tolerant networks (DTNs), multiple-copy routing is often adopted to improve the probability of successful message delivery but causes more traffic loads. The excessive increase in multiple messages copies often exhausts the network resources and deteriorates its performance significantly. To solve this problem, a mobility similarity-based routing (SBR) algorithm is proposed in this paper. On one hand, destination similarity is used to help the message carrier node find the appropriate relay nodes with the higher opportunity to meet the destination node in order to improve the delivery ratio. On the other hand, carrier similarity is adopted to avoid sending the same copy to the nodes with mobility similarity so as to reduce the number of message copies. Furthermore, inspired by the law of diminishing marginal utility in economics, a buffer management scheme based on the message transmission status is proposed. Experimental results show that the proposed SBR routing algorithm combined with the buffer management scheme can improve the delivery ratio and has the lower overhead ratio compared to other routing algorithms.

1. Introduction

Delay tolerant networks (DTNs) [1] are mobile wireless networks that experience frequent partitions. In DTNs, there may never exist a fully connected path from a source to a destination. The basic assumption on stable end-to-end path of routing protocols is invalid. Many real-world network scenarios fall into this paradigm, such as vehicular ad hoc networks [2], sensor networks for wildlife tracking and habitat monitoring [3], pocket switched networks [4], and mobile social networks [5].

In DTNs, communication is mainly dependent on the node mobility, which plays an important role in overcoming the lack of end-to-end path. The storage-carry-forward (SCF) routing paradigm is used to deliver messages in the challenging environments. Specifically, if a node receives a message from an encountered node, it stores and carries the message until another communication opportunity arises. Depending on the method whether the message carrier node keeps or removes the forwarded message, there are two classes of routing scheme in DTNs: single-copy routing [6] and multiple-copy routing [7].

As the primary purpose of DTN routing is to select some proper relaying nodes to guarantee that the message will be forwarded to the destination, the DTN routing protocols usually use the multiple-copy scheme to provide high delivery ratio. But the multiple-copy schemes cause tremendous network overheads. How to achieve high delivery ratio and maintain low overheads at the same time is the most important research issue for multiple-copy routing.

Moreover, the routing schemes are usually designed only from the viewpoint of single message and do not consider the problem of the usage of the whole network resources. Many proposed routing schemes for DTNs assume that nodes have infinite buffer space and ignore the contention for buffer space in network nodes. In practice, the nodes have limited buffer space in many wireless networks. Even if a node has large buffer space, it may share only a limited small
part with external traffic when acting as a relaying node. Thus, how to manage buffer space significantly affects the performance of multiple-copy routing protocols, especially in the environments where intermittent connectivity and long latency require the data to be stored for a long period throughout the network.

In this paper, we firstly propose a similarity-based routing algorithm in DTNs. Two mobility similarity metrics are adopted, that is, destination similarity and carrier similarity, to choose the next relay node. The destination similarity refers to the mobility similarity between an intermediate node and the destination. The carrier similarity refers to the mobility similarity between the message carrier node and its encountered node. In addition to using the destination similarity to improve message delivery probability, the carrier similarity is also adopted to control the message delivery overheads. On one hand, since the node with high destination similarity has more opportunities to encounter the destination node, the message carrier node will choose these nodes as relaying nodes. On the other hand, the message carrier node will not replicate the message to the nodes with high carrier similarity because these nodes may have the mobility pattern similar to the message carrier node.

Secondly, inspired by the law of diminishing marginal utility in economics, we propose a buffer management scheme for the routing algorithm [8]. This law shows that as a user increases consumption of a product, there is a decline in the marginal utility that user derives from consuming each additional unit of that product. Thus, it is not good to generate excessive redundant copies for a single message because this will occupy very large buffer space of nodes and will restrict the opportunity of other messages to access the buffer space when buffer overflow happens. In contrast, in our proposed scheme, we estimate the status of a message, for example, the total number of its copies in the network and its dissemination speed, and perform buffer replacement and scheduling accordingly. Messages that have a larger estimated number of copies and faster dissemination speed are dropped prior to other messages in the event of buffer overflow or are forwarded after other messages. In this way, the proposed solution provides better fairness among messages and hence improves the overall performance.

The major contributions of this paper are listed as follows.

(i) A novel similarity-based routing (SBR) algorithm is proposed. To the best of our knowledge, no previous studies have exploited the carrier similarity in designing routing protocols in DTNs.

(ii) A buffer management scheme, including message replacement and scheduling, is proposed for the routing algorithm.

(iii) Extensive simulation experiments are conducted on map-based mobility model and random waypoint (RWP) mobility model. The results demonstrate that the proposed routing algorithm combined with the buffer management scheme can achieve the higher delivery ratio and has a relatively lower overhead ratio compared to other routing schemes.

The rest of this paper is organized as follows. Section 2 reviews the previous work on routing and buffer management in DTNs. Section 3 presents a similarity-based routing algorithm. Section 4 proposes a buffer management scheme based on the message transmission status. In Section 5, the routing algorithm with buffer management scheme is evaluated through simulation experiments. Finally, Section 6 concludes the paper.

2. Related Work

This section overviews state-of-the-art DTN routing protocols and buffer management issues.

2.1. Single-Copy Routing. Direct transmission is the basic approach in single-copy routing. The source node sends the message to the destination node directly. Predict and relay (PER) routing [9] used the probability distribution of future contact times to choose the next relay node. Recently, social characters have been explored to design routing protocols in DTNs. MobySpace routing [10] constructed a high-dimensional Euclidean space upon mobility patterns. The nodes that have the mobility pattern similar to the destination are appropriate to act as relay nodes. SimBet [11] routing exploited ego-centric centrality and its social similarity to choose the better carriers for the final destination. In [12], social selfishness was introduced into DTN routing and a social selfishness aware routing (SSAR) algorithm was proposed for the collaborative environment. Whatever the routing metric used, however, single-copy routing faces the reliability problem that the message is not sent to its destination when the single message copy is discarded by the relay node.

2.2. Multiple-Copy Routing. The multiple-copy routing protocols inject multiple copies of a message to provide reliable delivery. As one of the earliest DTN routing protocols, the epidemic routing [13] floods the message throughout the network and thus greatly increases the message delivery ratio. However, the node resources, such as buffer space, bandwidth, and energy, seriously restrain the epidemic routing performance. To address this problem, many approaches have been adopted to reduce the overheads and improve the overall network performance. Based on the method of controlling the overheads, the multiple-copy routing schemes can be further separated into three classes: coding-based [14–17], quota-based [18–20], and utility-based [21–24].

In the coding-based algorithms, a message is divided into a set of code blocks. Destination node can reconstruct the original message when it receives a sufficiently large number of code blocks. In [14], Wang et al. combined erasure coding and replication-based routing schemes to increase the network throughput. Tsapeli and Tsoussidou [15] combined the probabilistic routing with erasure coding to enhance the robustness of the erasure-coding based forwarding in worst-case delays and small-delay scenarios. In [17], Altman et al. exploited linear block-codes and rateless random linear coding to solve the problem of optimal transmission and
scheduling policies with two-hop routing under memory and energy constraints.

In the quota-based routing schemes, the fixed number of message copies is inserted into the network. Thus, the constant overhead is maintained. Spyropoulos et al. [18] proposed two different methods, that is, Source Spray and Wait (SS&W) and Binary Spray and Wait (BS&W), for allocating message copies. In [19], Nelson et al. used the encounter-based metric for optimization of message allocation. Elwhishi et al. [20] proposed a self-adaptive contention aware routing protocol SAURP, which makes forwarding decision based on the utility function value of the encountered node regarding the destination and the number of message copy tokens.

In the utility-based routing schemes, each node maintains a defined utility for destination. The utility can be a function of encounter history between nodes, node resources, and so forth. Lindgren et al. [21] used the past encounters to predict the probability of future encounters. Grasic et al. [22] made a little modification to the routing metric calculations in Prophet which can improve its performance. Ramanathan et al. [23] presented a variant of epidemic routing PREP, which prioritizes messages based on costs to destination, source, and expiry time to decide whether the message is deleted or reserved when it faced the situation of insufficient resources. In [24], Balasubramanian et al. treated DTN routing as a problem of resource allocation and proposed a heuristic approach to maximize the specified performance metrics, but it requires high computation cost.

2.3. Buffer Management. Generally, since the main reason of message loss is the buffer overflow [25–27], various congestion control schemes are proposed [28–30]. In DTN, how to design buffer management schemes becomes very important for the multiple-copy routing performance.

Some buffer management strategies can be performed independent of the underlying routing algorithm. In the Epidemic-IMMUNE routing protocol [31], three buffer management strategies including drop-tail, drop-head, and source-prioritized drop-head were examined. It was shown that appropriate buffer management schemes have great effects on delivery performance. In [32], Krifa et al. proposed a distributed algorithm based on the estimated global information to optimize the delivery ratio and the average latency.

Some other buffer management strategies have been proposed for specific routing algorithms. In [33], Lindgren and Phanse used the delivery predictability metric defined in Prophet routing protocol to decide which message is forwarded first. They evaluated various combinations of queuing policies and scheduling strategies for Prophet protocol. In [34], Erramilli and Crovella designed scheduling and replacement algorithms based on the message priority that is defined in delegation forwarding routing algorithm [35]. In [36], Rashid et al. proposed a buffer management policy, which takes message sizes into account.

Compared to the above solutions, our approach integrates both routing protocol and buffer management strategy. The appropriate relay nodes are chosen by considering the carrier similarity and destination similarity at the same time. Moreover, the messages are replaced and scheduled based on the number of copies and dissemination speed. In this way, our approach achieves higher delivery ratio and lower overheads.

3. Similarity-Based Routing Algorithm

In this section, the system model is presented. The basic definitions and notations that will be used throughout the rest of the paper are also introduced. Then mobility similarity metrics used for routing decision are defined. The design of similarity-based routing algorithm is presented in the end.

3.1. System Model. The network consists of $N$ mobile nodes, each of which has a unique ID and belongs to one community. Each node in the network has the same buffer size and the same transmission range. The probability of moving to other communities from the local community is the same. The global knowledge about nodes’ mobility is unknown to every node in the beginning. Every message generated in the network has a Time-to-Live (TTL) value. The source node and relay nodes will drop message and their copies when their TTL expires. For a given message, there are one source node, one destination node, and $(N - 2)$ intermediate nodes. The source node can deliver message to the destination directly or through intermediate nodes. Two nodes exchange messages when they are within communication range of each other. We assume that the bandwidth is large enough to transmit all messages.

The contact time is defined as the time interval in which two nodes are within their communication range. The inter-contact time is defined as the time interval in which a node pair is not within its communication range. The interdropping time is defined as the time interval in which a single message or its copies are dropped.

Because the transmission time is negligible compared to the intercontact time, the transmission time is often ignored in other studies. Some studies indicated that many popular mobility models, like random waypoint, random walk, and community-based model, have such a property where the intercontact time of mobility model is exponentially distributed or has exponential tail [37,38]. We also consider that the intercontact time follows exponential distributions. The interdropping time of messages is mutually independent and identically distributed random variables, with an exponential distribution.

3.2. Mobility Similarity Metrics. Nodes with similar characteristics of the mobility pattern are discovered in designing routing metrics in DTNs [10,11]. Some previous studies have exploited mobility similarity as routing metric, but they only considered the destination similarity. It is argued that the nodes having the similar mobility pattern with the destination can enhance the delivery ratio and reduce the communication cost. However, from the viewpoint of message carrier node, if the encountered nodes have similar mobility pattern with the message carrier node, then they would not be good relaying nodes. Thus, we propose the carrier similarity to control the overhead of multiple message copies.
Carrier similarity is measured as the proportion of the same history encountered nodes between the message carrier node and the contact node. If a node has high similarity with the message carrier node, it will have similar mobility scope. The node will not be appropriate for diffusing message far away. The message carrier node will hold the messages by itself. In contrast, if a node has low similarity with the message carrier node, it will be capable of disseminating messages far away. Thus, the node is a good potential relaying node. Let $S_i$ and $S_j$ denote the set of encounters experienced by node $i$ and node $j$ within $T$, respectively. When node $i$ contacts node $j$, the carrier similarity, $\text{Sim}^c_i$, calculated by node $i$ is defined as follows:

\[
\text{Sim}^c_i = \frac{|S_i \cap S_j|}{|S_i|}.
\]  

(1)

Destination similarity measures the number of the same nodes encountered by a node and the destination node. If a node has high similarity with the destination, it may have a higher chance to meet the destination. For a given node $i$, the destination similarity is defined below:

\[
\text{Sim}^{d}_i = |S_i \cap S_d|,
\]  

(2)

where $S_i$ and $S_d$ are the set of the encounter nodes experienced by node $i$ and the destination node within $T$, respectively.

3.3. Similarity-Based Routing Algorithm. Here we propose the similarity-based routing algorithm, with the aim of bringing a concise concept of behavior similarity into DTN routing. Similarity-based routing combines carrier and destination similarity to make routing decisions. There are two intuitions behind this algorithm. Carrier similarity could control the number of flooding message copies and avoid missing the good relaying node. Destination similarity could improve forwarding efficiency.

Each node maintains an encounter node vector (nv). An entry in the nv is composed of two items, including node ID and encounter time. The node can obtain contact information in regard to nodes that it has encountered and that it has not yet encountered. When a node encounters other nodes, the node will exchange the nv. If the encounter node already has an entry in the nv, the node will only modify the encounter time. Otherwise, a new entry will be inserted into the nv. The node will refresh its nv in a given time $T$. If the encounter time of a node exceeds the time $T$, the entry of the node will be removed from the nv.

Node contacts are represented as an $n \times n$ symmetric matrix, where $n$ is the number of contacts encountered by a given node. If there is a contact between node $i$ and node $j$, the corresponding element in the matrix is set to 1, otherwise to 0. The count of nonzero equivalent row entries in the matrix represents the number of common neighbors between node $i$ and node $j$. The intermediate nodes may obtain the useful information in regard to the destination node via direct encountered nodes. The destination node may be an indirect node. Then, the similarity with indirect encounter nodes is also needed to be evaluated. Node contacts with direct encountered nodes and indirect encountered nodes are represented as an $n \times k$ matrix, where $n$ is the number of direct encountered nodes and $k$ is the number of indirect encountered nodes. More details of the calculation can be found in [11].

Each node also maintains a summary vector. An entry in summary vector is composed of four items, including message ID, replication number, message TTL value, and initial message TTL value. The summary vector is used to update the forwarding status of a message. It is exchanged when two nodes are encountered. When a node receives a new message, a new entry is added in the summary vector. When the node discards a message, the corresponding entry in the summary vector remains in the node until the TTL value of that message expires.

Because the sizes of the encounter node vector and the summary vector are very small compared to a message, the overheads of the encounter vector and the summary vector are not considered.

The routing algorithm is outlined in Algorithm 1. It represents the communication process when message carrier node $i$ meets the potential relaying node $j$. They exchange some information of their encounter history used for computing routing metrics like Epidemic. Node $i$ calculates the similarity with node $j$.

If the carrier similarity utility between node $i$ and node $j$ is below the predefined carrier similarity threshold $\text{Similarity}^c_{TH}$, it means that node $j$ has the ability to spread the message to more unfamiliar nodes. If the threshold is too small, messages are hard to be sent to other nodes. If the threshold is too large, excessive distribution of messages cannot be controlled. Therefore, $\text{Similarity}^c_{TH}$ is set as 0.5 to
achieve a good trade-off between the overhead and delivery ratio. For all the unknown messages carried by node \(i\), when the destination of messages has higher similarity with node \(j\) than it has with node \(i\), node \(i\) will replicate these messages to node \(j\). If the carrier similarity utility between node \(i\) and node \(j\) is greater than the predefined carrier similarity threshold, it means that node \(i\) and node \(j\) have similar mobility scope. There is no need to increase the number of message copies in the networks. Then node \(i\) will only forward these messages to node \(j\) when the latter has a high destination similarity utility.

4. Buffer Management Scheme

Although multiple-copy routing can improve the probability of delivery rate, it also inevitably brings more traffic into the network. When the node buffer overflows, the network performance decreases sharply. To overcome this problem, we propose a buffer management approach, which exploits messages transmission status in the networks to decide the priority of message replacement and scheduling.

4.1. Message Transmission Analysis. The dissemination of messages in the network is modeled from the perspective of an individual message \(m\). The nodes that hold message \(m\) or its copy in the networks are called infected nodes; other nodes are called uninfected nodes. Let \(x\) and \(s\) denote the number of infected nodes and uninfected nodes, respectively, at a given time. Thus, the total number of the network nodes can be written as

\[
N = s + x. \tag{3}
\]

Let \(\lambda\) represent the encounter rate between the nodes. Each infected node will meet \(\lambda s / N\) uninfected nodes in a unit time. Then, the total number of infected nodes increases by \(\lambda s x / N\) in a unit time. Let \(\mu\) denote the rate of dropping messages. The number of nodes whose status is changed from infected to uninfected is \(\mu x\). Then, the increasing rate of infected nodes is

\[
\frac{dx}{dt} = \frac{\lambda s x}{N} - \mu x. \tag{4}
\]

Messages will be discarded when two encountered nodes exchange message copies and buffer overflow occurs. Under the condition that the encounter rate of nodes is greater than the dropping rate of messages, that is, \(\lambda > \mu\), let

\[
\rho = \frac{\lambda}{\mu}. \tag{5}
\]

Combining (3), (4), and (5) together yields

\[
\frac{dx}{dt} = -\lambda x \left[ \frac{x}{N} - \left(1 - \frac{1}{\rho}\right) \right]. \tag{6}
\]

The number of infected nodes can be expressed as

\[
x = N \left(1 - \frac{1}{\rho}\right) \frac{1}{1 + \left[N \left(1 - \frac{1}{\rho}\right) - 1\right] e^{-\lambda(1-1/\rho)t}}. \tag{7}
\]

We can obtain from (5) and (7) that the increasing rate of \(x\) depends on its initial value. Because the initial value of \(x\) is 1, \(x\) grows faster in the beginning stage of dissemination. The limit value increases as \(\rho\) increases and is given by

\[
x(\infty) = N \left(1 - \frac{1}{\rho}\right). \tag{8}
\]

Let a utility function \(U(x)\) model the delivery probability for each single message. \(U(x)\) is an increasing function of parameter \(x\). In other words, the greater the number of nodes receiving the same message, the higher delivery probability of the message. This implies the following requirement on the derivative of the function \(U(x)\):

\[
\frac{dU(x)}{dx} \geq 0. \tag{9}
\]

According to the law of diminishing marginal utility, we can obtain the equation below:

\[
\lim_{x \to N} \frac{dU(x)}{dx} = 0. \tag{10}
\]

Equation (10) reflects the phenomenon that the improvement of the delivery probability is vanishing when a high delivery probability is reached that is a feasible assumption for general cases.

Note that, in the case of buffer management, the number of infected nodes is not greater than the total number of nodes in the network. Therefore, there is an upper bound to the probability for each single message. For this reason, we get

\[
\lim_{x \to N} U(x) = 1. \tag{11}
\]

It is often supposed that a utility function has some properties of regularity, for example, continuous differentiability, at least piecewise. When these properties are applied to (9), (10), and (11), we can have

\[
\exists c: U(x) < 0, \quad \forall x \geq c. \tag{12}
\]

Formula (12) implies that the concavity of \(U(x)\) at least for \(x\) is greater than a given value. Therefore, when a message has fewer copies in the network, \(U(x)\) has the properties of convex function. From the perspective of the whole network, if such a message is dropped in the network, the decreased utility of this message is larger than the increased utility when a node accommodates a message with a large number of copies. On the contrary, when a message has a large number of copies in the network, \(U(x)\) has the properties of concave function. If such a message is dropped, the decreased utility is less than the increased utility when a node accepts a message with few copies.

From the above analysis, we can obtain that messages with a large number of copies in the network have much more transfer opportunities. These messages will have relatively higher probability to reach their destination nodes. When we control the increasing number of these messages, the lost utility is limited at a low level. For a single message, it
decreases their delivery probability but brings more transfer opportunities for messages that have fewer copies in the network. The increased utility of these messages is greater than the decreased utility of messages that have more copies in the networks. Therefore, it will improve the overall performance.

4.2. Buffer Replacement Scheme. Because of intermittent connectivity in the network, a node could not get the accurate global status about a particular message. It can use statistical learning to estimate the dissemination status of a message when nodes are encountered. We introduce two metrics to measure the priority of a message, including the number of message copies and the dissemination speed of a message. A message that has a smaller replication number is assigned a higher priority. If messages have the same replication number, the message with the lower speed of dissemination is assigned a higher priority.

Let \( R^i_m \) denote the replication number of message \( m \) known by node \( i \). Obviously, it can be seen that message \( m \), which has the greater value of \( R^i_m \), has the strong ability to spread in the network. Meanwhile, more copies of message \( m \) might reside in the network. On the contrary, message \( m \) that has a lower value of \( R^i_m \) might leave fewer copies in the network. Node \( i \) will discard the message that has a higher value of \( R^i_m \) first.

Here, we describe the process of learning the replication number of a single message. The initial value of replication number is set to 1 when a new message is generated in the network. When node \( i \) that carries message \( m \) meets node \( j \) that does not carry message \( m \), the replication number of message \( m \) is processed in the following two cases.

(i) One case is that node \( j \) is selected as a relay node for message \( m \). If node \( j \) does not contain any information about message \( m \), the replication number of message \( m \) is set to \( R^j_m + 1 \) in both nodes. Otherwise, both nodes exchange the summary vector and set the value to \( \max(R^j_m, R^i_m) + 1 \).

(ii) The other case is that node \( j \) is not selected as a relay node for message \( m \). When node \( j \) does not contain any information about message \( m \), the replication number of message \( m \) is set to \( R^j_m \) in both nodes. Otherwise, both nodes exchange the summary vector and set the value to \( \max(R^j_m, R^i_m) \).

When different messages have the same estimated replication number, we use the Rate metric to describe the dissemination speed of a message. According to the path explosion phenomenon, once a message reaches the destination, there are a number of near-optimal paths to the destination. Therefore, more message copies can exist in the networks. Path explosion occurs much faster among the higher contact rate nodes than the lower contact rate nodes. The Rate metric can reflect the nodes contact rate from viewpoint of a message. This metric is defined as follows:

\[
\text{Rate} = \frac{K_m}{\text{TTL}_{\text{init}} - \text{TTL}}, \quad (13)
\]

where \( K_m \) represents the hop count experienced by message \( m \). Our design for the buffer management scheme associates a hop count with each message. The hop count \( K_m \) is estimated according to the message replication number. The original replication number is one. For a given message \( m \) carried by node \( i \), the hop count \( K_m = R^i_m - 1 \). The message that has a higher dissemination speed might have many more copies in the networks.

The node will accept a new message if it has enough free buffer space. Otherwise, the node will compare all the messages in its buffer with the new one according to the priority discussed above. Message with a lower priority will be discarded. The algorithm MTSBR is shown in Algorithm 2.

4.3. Buffer Scheduling Scheme. A set of messages that are determined by routing protocol should be forwarded to a better intermediate node. We call these messages Ready Set (RS). Ideally, message carrier node will transmit all of them to the relay node. Unfortunately, not all the messages could be transmitted due to finite bandwidth or unexpected interruptions. It is important for a node to decide the order in which the messages are transmitted. Meanwhile, routing protocol does not consider whether the relay node has enough buffer space to hold these messages. Obviously, bandwidth and node energy are wasted when transmitted messages are dropped due to buffer overflow. It is also important for a node to decide which messages should be forwarded to relay node. To address these problems, we propose MTSBS scheduling scheme that is outlined in Algorithm 3.

Firstly, MTSBS sorts messages in RS in a descending order according to their priority. In all cases, the message with higher priority will be forwarded. Secondly, MTSBS will choose which messages to forward to relay node. If the lowest priority of messages in RS is greater than the highest priority of messages in the peering node, then the node forwards all the messages. If the highest priority of messages is lower than the lowest priority in the peering node, the node will only forward messages that the peering node could contain in its free buffer. In other cases, node merges the message list in RS and peering node. Then it sorts the merged list and selects the top messages that their buffer occupancy is close to the buffer capacity. These messages residing in the local node are forwarded to relay node.

\[
\text{Algorithm 2: Buffer replacement algorithm: MTSBR.}
\]

\[
\text{Upon receiving message } m \text{ from the encountered node}
\]

\[
\text{while Buffer.freeSize} < m.size \text{ do}
\]

\[
\text{msg} = \min \text{Priority(messages in Buffer Union } m)\]

\[
\text{if msg} == m \text{ then}
\]

\[
\text{deleteMessage(msg)}
\]

\[
\text{else}
\]

\[
\text{deleteMessage(msg)}
\]

\[
\text{Buffer.freeSize} += \text{msg.size}
\]

\[
\text{end if}
\]

\[
\text{end while}
\]
5. Performance Evaluation

We compare the performance of the proposed SBR algorithm against the following three routing algorithms (Epidemic [13], Prophet [21], and ProphetV2 [22]) in DTNs using the ONE [39] simulator.

Epidemic [13]. Messages are flooded to all the encountered nodes. It uses the DO (Drop Oldest: message that has the shortest TTL value is dropped first) and adopts random strategy for message replacement and scheduling, respectively. It is the benchmark that was used for performance analysis and comparison in the previous works.

Prophet [21]. This is a mobility-based approach in DTNs. It calculates the routing metric by using the history of node encounters and transitivity. A message is forwarded to a node that has a higher estimated delivery predictability for a specific destination node than the current message carrier node. It also uses the DO replacement strategy and adopts GRTRMax for message scheduling. GRTRMax forwards messages in descending order of delivery predictabilities.

ProphetV2 [22]. It redefines the transitivity update equation and direct encounter update equation in Prophet.

In this experiment, we also evaluate SBR with different buffer management schemes. SBR-1 denotes SBR routing algorithm with DO replacement and random scheduling scheme. SBR-2 represents SBR algorithm with HBD (History Based Drop) [32] replacement and random scheduling scheme. HBD is a distributed message replacement scheme based on the estimated global information about messages to optimize the specific metric. SBR-3 represents SBR algorithm with our proposed buffer management scheme.

We compare the performance of these algorithms in terms of message delivery ratio, overhead ratio, and average delay.

Algorithm 3: Buffer scheduling algorithm: MTSBS.

5.1. Experimental Settings. Two mobility models, that is, a map-based mobility model and RWP mobility model, are used to evaluate the performance of routing protocols.

Under the map-based model, we use the default map in ONE, which consists of a 4500 m × 3500 m area. The map-based scenario is shown in Figure 1. Each labeled circle in the map represents the node, which belongs to a specific group. In order to investigate the impact of different number of groups, we compare these routing algorithms with 3 and 4 groups, respectively. We set $k$ ($k = 3, 4$) Points-of-Interest (POIs),
which belong to a certain POI group. Each node moves among the POI groups with a specific probability. Nodes move to other POI groups with the probability \( Pr \) (\( Pr = 0.1 \)). Nodes move in the local group with the probability \( 1-(k-1)Pr \). The setting of destination selection probability is similar to the one that was done in [21]. Nodes in each group represent pedestrians. These nodes move with the average speed of 1.34 m/s, which represents a pedestrian average walking speed [40].

Under the RWP model, the simulation area is \( 1 \text{ km} \times 1 \text{ km} \). Nodes are randomly distributed in the field. Nodes have an average moving speed of 1.34 m/s and the pause time of a stop is uniformly distributed in \([0, 120]\) seconds.

For the two simulation models, each node uses an ideal communication module and has a communication range of 10 m. The transmission speed of nodes is 2 Mbps. Simulation time is 4 hours to ensure that the nodes can form the steady mobility pattern and the stable simulation results can be achieved. A new message with TTL is generated every 15 seconds. The size of messages is 1 KB.

### 5.2. Experimental Results

**Varying Buffer Size.** Figures 2 and 3 reveal the impact of buffer size on the performance of routing algorithms in the 3- and 4-group conditions under the map-based mobility model,
respectively. In the two scenarios, pedestrian groups have 40 nodes in each group. Their buffer size varies from 100 to 600 KB. As the results in the 3- and 4-group conditions show a similar trend, we only discuss the results for the 3 groups’ condition.

Figure 2(a) shows that the delivery ratio becomes larger as the buffer size increases. Epidemic floods more copies in the networks, so it has the lowest delivery ratio when the node buffer space is very small. As expected, ProphetV2 significantly outperforms Prophet in the map-based mobility model, because ProphetV2 can deal with the problem that nodes come together and repeatedly exchange their sets of delivery predictabilities. SBR limits the flooding and improves the delivery probability. As for the buffer management schemes, MTSBR could guarantee the transmission efficiency as it incorporates network status to make decision. When the buffer overflows, MTSBR drops messages that have the most copies. Although HBD and DO take the number of message copies into consideration, they do not care about dissemination capacity of messages. Since DO considers only the number of message copies in a local view and does not incorporate network status, SBR-3 has a higher delivery ratio compared to SBR-1 and SBR-2.

It can be seen from Figure 2(b) that three SBR algorithms have the lower overhead ratio than Epidemic, Prophet, and ProphetV2. Epidemic replicates message to any encountered.
nodes. When the node buffer space is very small, more message copies are discarded and retransmitted. So it causes the higher overhead ratio. Prophet and ProphetV2 only replicate messages to the encountered nodes that have a higher delivery probability. SBR algorithms can alleviate traffic to some extent because they are able to control the number of message copies by comparing the carrier similarity between encountered nodes. SBR-3 has the lowest overhead ratio among all the SBR algorithms. The MTSBR replacement scheme could reduce the number of retransmissions. It could partially avoid dropping messages that is in the beginning stage of dissemination. MTSBS will decide which messages to transmit. The scheduler considers the buffer constraint and will not transmit the messages that will be dropped in the next intermediate node. So it has a relatively low overhead ratio.

Figure 2(c) shows that the average delay of all the routing algorithms decreases. When the buffer size increases, more message copies will be saved in the nodes’ buffer. The message copies will have more opportunities to arrive at the destination node. Therefore, the message delay will decrease. SBR algorithms performance in terms of average delay remains acceptable, especially SBR-3 algorithm.

Figure 4 shows the impact of buffer size on the performance of routing algorithms under the RWP mobility model.
Epidemic gains a significant benefit from increased buffer size. It has the lowest average delay when the buffer size exceeds 200 KB in this experiment. SBR algorithms still have the high delivery ratio when buffer space is small, especially SBR-3 algorithm. These results show that our approach can choose appropriate relay nodes with lower overhead. The average delay is also acceptable. The RWP mobility model does not provide predictable mobility patterns that Prophet and ProphetV2 can leverage. So the difference between their performances is not great like that in group conditions.

**Varying Number of Nodes.** Figures 5, 6, and 7 depict the impact of the increasing number of nodes on the performance of different protocols. The number of nodes in each group varies from 20 to 50 under the map-based mobility model. Under the RWP mobility model, the number of nodes varies from 60 to 150.

Figure 5(a) reveals that the delivery ratio does not fluctuate much when the number of nodes increases in each group. For the heavy traffic, buffer contention will become more serious when the number of nodes increases even if the routing protocols adopt the controlled flooding scheme. Since SBR uses message transmission status to manage the buffer space, it has a higher delivery ratio. When the number of network nodes increases, the estimate of the global information in HBD becomes more difficult. It has less impact on routing
than MTSBR. Thus, SBR-3 has higher delivery ratio than SBR-2.

Figure 5(b) shows that as the number of network nodes increases, the transmission opportunities are also increased. As more message copies are sent in the network, the buffer contention becomes more serious, thus resulting in the increasing overhead ratio. The effect of buffer management scheme for SBR becomes obvious. MTSBR limits further spread of messages that might have a large number of copies, and MTSBS could reduce unnecessary transmissions caused by buffer overflow. Therefore, the increase of overhead ratio in SBR-3 is low.

It can be seen from Figure 5(c) that the average delay of all the routing algorithms decreases. Because more nodes participate in the relay activity, more forwarding opportunities arise. It can alleviate the impact of forwarding limitation of SBR algorithms, Prophet, and ProphetV2. It is worth noting that Figure 6 reveals the performance trends similar to those in Figure 5.

Figure 7 shows the results under the RWP mobility model. Communication opportunities arise when the node density increases. We can see from Figure 7(a) that SBR algorithms have relatively higher delivery ratio than other algorithms when the number of nodes increases. Figure 7(b)
shows that the overhead increases when the number of nodes increases. But SBR algorithms can control the number of disseminated messages by carrier similarity to some extent. Our proposed buffer management scheme can further alleviate message retransmission, so SBR-3 has the lowest overhead ratio. In Figure 7(c), we can see that SBR algorithms also have good performance in terms of average delay.

*Similarity Threshold Analysis.* Here, to analyze the impact of similarity threshold $\text{Similarity}_{TH}^s$ on protocol performance, we evaluate SBR-1 under different $\text{Similarity}_{TH}^s$ value with 0.25, 0.5, and 0.75, respectively. The buffer size varies from 100 to 600 KB under the map-based mobility model. As can be seen from Figure 8, the SBR-1 has similar trends for the performance with $\text{Similarity}_{TH}^s$ varying. In the $\text{Similarity}_{TH}^s = 0.25$ case, SBR-1 has the lowest overhead ratio, because messages are difficult to be sent to other nodes. However, SBR-1 has the lowest delivery ratio and the highest average delay. In the $\text{Similarity}_{TH}^s = 0.75$ case, SBR-1 has good performance on delivery ratio and average delay. But it also brings more overhead. For $\text{Similarity}_{TH}^s = 0.5$, SBR-1 has the highest delivery ratio. The overhead ratio and average delay

![Graphs showing delivery ratio, overhead ratio, and average delay vs. number of nodes for different algorithms.](image)
are on the middle level. It achieves a good performance trade-off among delivery ratio, overhead ratio, and average delay.

6. Conclusions

The existing routing protocols in DTNs have considered only the similarity of mobility patterns between the relaying nodes and the destination nodes. In this paper, we take into account the similarity of mobility patterns between the message carrier node and its encountered nodes and propose a similarity-based routing protocol, which uses different similarity as the condition of replication or forwarding. Moreover, we estimate the replication number and spreading speed of messages using encounter history. The message with a smaller replication number and lower speed of dissemination is assigned the higher priority. Furthermore, we propose a buffer replacement scheme MTSBR and a scheduling scheme MTSBS according to the priority. Simulation results show that our routing protocols combined with the buffer management schemes outperform the existing routing protocols in terms of delivery ratio and overhead ratio in guaranteed delay.

Figure 8: Comparison of performance under different similarity threshold.
Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This project is supported by the National Natural Science Foundation of China (Grants nos. 6103204, 61272322, and 61402541), the Humanities and Social Science Youth Foundation of Ministry of Education of China (Grant no. 13YJCZH110), the Construct Program of the Key Discipline in Hunan Province, the Scientific Research Fund of Hunan Provincial Education Department (Grant no. 12C0768), the Major Science and Technology Research Program for Strategic Emerging Industry of Hunan (Grant no. 2012GK4054), the Open Funding of Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory (Grant no. ITDU14010/KX142600017), and Postdoctoral Funding of Central South University and Changsha Bohua Technology Co., Ltd., China.

References

[1] K. Fall, “A delay-tolerant network architecture for challenged internets,” in Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM ’03), pp. 27–34, ACM, Karlsruhe, Germany, August 2003.

[2] K. K. Sevimli and M. Soyturk, “Enabling delay-tolerant communications for partially connected vehicular ad hoc networks,” International Journal of Ad Hoc and Ubiquitous Computing, vol. 11, no. 2-3, pp. 157–168, 2012.

[3] S. Ehsan, K. Bradford, M. Brugger et al., “Design and analysis of delay-tolerant sensor networks for monitoring and tracking free-roaming animals,” IEEE Transactions on Wireless Communications, vol. 11, no. 3, pp. 1220–1227, 2012.

[4] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket switched networks and human mobility in conference environments,” in Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN ’05), pp. 244–251, ACM, Philadelphia, PA, USA, August 2005.

[5] Y. Xie and G. Wang, “Message matching-based greedy behavior detection in delay tolerant networks,” Journal of Computer and System Sciences, vol. 80, no. 5, pp. 903–915, 2014.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efﬁcient routing in intermittently connected mobile networks: the single-copy case,” IEEE/ACM Transactions on Networking, vol. 16, no. 1, pp. 63–76, 2008.

[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efﬁcient routing in intermittently connected mobile networks: the multiple-copy case,” IEEE/ACM Transactions on Networking, vol. 16, no. 1, pp. 77–90, 2008.

[8] Y. Liu, J. Wang, S. Zhang, and H. Zhou, “A buffer management scheme based on message transmission status in delay tolerant networks,” in Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’11), pp. 1–5, IEEE, Houston, Tex, USA, December 2011.

[9] Q. Yuan, I. Cardei, and J. Wu, “An efﬁcient prediction-based routing in disruption-tolerant networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 1, pp. 19–31, 2012.

[10] J. Leguay, T. Friedman, and V. Conan, “DTN routing in a mobility pattern space,” in Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN ’05), pp. 276–283, ACM, Philadelphia, PA, USA, August 2005.

[11] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant MANETs,” in Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’07), pp. 32–40, September 2007.

[12] Q. Li, W. Gao, S. Zhu, and G. Cao, “A routing protocol for socially selfish delay tolerant networks,” Ad Hoc Networks, vol. 10, no. 8, pp. 1619–1632, 2012.

[13] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Tech. Rep. CS-200006, Duke University, 2000.

[14] Y. Wang, S. Jain, M. Martonosi, and K. Fall, “Erasure-coding based routing for opportunistic networks,” in Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN ’05), pp. 229–236, ACM, Philadelphia, PA, USA, August 2005.

[15] F. Tsapeli and V. Tsoussidis, “Routing for opportunistic networks based on probabilistic erasure coding,” in Wired/Wireless Internet Communication: 10th International Conference, WICON 2012, Santorini, Greece, June 6–8, 2012. Proceedings, vol. 7277 of Lecture Notes in Computer Science, pp. 257–268, Springer, Berlin, Germany, 2012.

[16] J. Widmer and J.-Y. Le Boudec, “Network coding for efﬁcient communication in extreme networks,” in Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN ’05), pp. 284–291, August 2005.

[17] E. Altman, L. Sassetelli, and F. D. Pellegrini, “Dynamic control of coding for progressive packet arrivals in DTNs,” IEEE Transactions on Wireless Communications, vol. 12, no. 2, pp. 725–735, 2013.

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efﬁcient routing scheme for intermittently connected mobile networks,” in Proceedings of the ACM Workshop on Delay-Tolerant Networking (SIGCOMM ’05), pp. 252–259, 2005.

[19] S. C. Nelson, M. Bahkt, and R. Kravets, “Encounter-based routing in DTNs,” in Proceedings of the 28th Conference on Computer Communications (INFOCOM ’09), pp. 846–854, IEEE, April 2009.

[20] A. Elwishi, P.-H. Ho, K. S. Naik, and B. Shihada, “Self-adaptive contention aware routing protocol for intermittently connected mobile networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 7, pp. 1422–1435, 2013.

[21] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 7, no. 3, pp. 19–20, 2003.

[22] S. Grasic, E. Davies, A. Lindgren, and A. Doria, “The evolution of a DTN routing protocol—PRoPHETv2,” in Proceedings of the 6th ACM Workshop on Challenged Networks (CHANTS ’11), pp. 27–30, ACM, Las Vegas, Nev, USA, September 2011.

[23] R. Ramanathan, R. Hansen, P. Basu, R. R. Salehs-Hain, and R. Krishnan, “Prioritized epidemic routing for opportunistic networks,” in Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiOpp ’07), pp. 62–66, June 2007.

[24] A. Balasubramanian, B. N. Levine, and A. Venkataramani, “Replication routing in DTNs: a resource allocation approach,” IEEE/ACM Transactions on Networking, vol. 18, no. 2, pp. 596–609, 2010.
J. Huang, J. Wang, and J. Ye, “A buffer management algorithm for improving up/down transmission congestion protocol fairness in IEEE 802.11 wireless local area networks,” *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2228–2240, 2014.

J. Wang, L. Rong, and Y. Liu, “A robust proportional controller for AQM based on optimized second-order system model,” *Computer Communications*, vol. 31, no. 10, pp. 2468–2477, 2008.

J. Wang, L. Rong, and Y. Liu, “Design of a stabilizing AQM controller for large-delay networks based on internal model control,” *Computer Communications*, vol. 31, no. 10, pp. 1911–1918, 2008.

J. Wang, P. Dong, J. Chen, J. Huang, S. Zhang, and W. Wang, “Adaptive explicit congestion control based on bandwidth estimation for high bandwidth-delay product networks,” *Computer Communications*, vol. 36, no. 10-11, pp. 1235–1244, 2013.

T. Zhang, J. Wang, J. Huang, Y. Huang, J. Chen, and Y. Pan, “Adaptive-acceleration data center TCP,” *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1522–1533, 2014.

J. Ye, J. Huang, J. Wang, S. Zhang, and Z. Zhang, “ECN-based congestion probability prediction over hybrid wired-wireless networks,” *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 134620, 11 pages, 2014.

X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance modeling of epidemic routing,” *Computer Networks*, vol. 51, no. 10, pp. 2867–2891, 2007.

A. Krifa, C. Barakat, and T. Spyropoulos, “Optimal buffer management policies for delay tolerant networks,” in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks* (SECON ’08), pp. 260–268, San Francisco, Calif, USA, June 2008.

A. Lindgren and K. S. Phanse, “Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks,” in *Proceedings of the 1st International Conference on Communication System Software and Middleware (COMSWARE ’06)*, pp. 1–10, IEEE, New Delhi, India, January 2006.

V. Erramilli and M. Crovella, “Forwarding in opportunistic networks with resource constraints,” in *Proceedings of the 3rd ACM Workshop on Challenged Networks (CHANTS’08)*, pp. 41–48, ACM, San Francisco, Calif, USA, September 2008.

V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, “Delegation forwarding,” in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’08)*, pp. 251–260, May 2008.

S. Rashid, A. H. Abdullah, M. S. M. Zahid, and Q. Ayub, “Mean drop an efectual buffer management policy for delay tolerant network,” *European Journal of Scientific Research*, vol. 70, no. 3, pp. 396–407, 2012.

T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Performance analysis of mobility-assisted routing,” in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’06)*, pp. 49–60, May 2006.

R. Groenevelt, P. Nain, and G. Koole, “Message delay in MANET,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 412–413, 2005.

A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools ’09)*, pp. 1–10, ACM, Rome, Italy, March 2009.
