Due to the nonexistence of end-to-end path between the sender and the receiver in delay tolerant networks and mobile social networks, consequently successful message transmission faces great challenges. In this paper, an adaptive routing algorithm taking full use of gregariousness characteristics of moving nodes is proposed. We first abstract all social relationships and uniformly represent them using friendship. Then by dynamically dividing nodes into different social groups, we finish flooding message among the target social group where destination node resides. In addition, we propose a social group based flooding model and a message redundancy control model to select fewer but better relay nodes and further reduce message redundancy. Extensive simulations have been conducted based on the synthetic traces generated by working day movement model and the results show that the proposed routing algorithm can get a higher message delivery ratio and a lower overhead ratio compared to Bubble Rap, Epidemic, and ProPHET, thus proving a better routing performance.

1. Introduction

TCP/IP protocol is based on certain assumptions, such as continuous end-to-end connections, low delivery latency, symmetrical bidirectional data transfer rate, and low error rate. In this case, TCP/IP protocol is able to shield heterogeneous networks. So with the TCP/IP protocol stack, traditional Internet has achieved a great success. However in recent years, some emerging challenged networks deployed in extreme environment are unable to meet the above assumptions, for example, vehicular ad hoc networks [1], pocket switched networks [2, 3], underwater sensor networks [4], interplanetary internet, mobile social networks [5, 6], and so forth. These special networks are characterized by intermittent connectivity, sparse node density, limited network resource, node mobility, and so forth. In these networks, there may never be a complete end-to-end path between the sender and the receiver, so traditional TCP/IP protocol is difficult to get efficient achievements. Consequently successful message delivery in such networks faces great challenges.

In 2003, Kevin Fall first put forward the concept of delay tolerant networks (DTNs). Soon afterwards, the DTN architecture [7–9] is proposed, which introduces a bundle layer between the application layer and the transport layer. With the bundle layer, DTNs can shield heterogeneous networks, thus communicating across multiple regions that have different types of network architectures and protocols. In order to cope with frequent network topology partitions and finish end-to-end data communications, DTN routing adopts the store-carry and forward strategy to relay messages hop by hop.

So far, a large number of routing protocols have been proposed to optimize the next hop relay node selection in the case of lacking of global topology knowledge. In order to select appropriate next hop relay nodes to gradually move closer to the final destination node so as to improve message delivery ratio, most of the proposed algorithms forward or replicate messages by taking into account nodes’ physical attributes such as link status and geographic location. However, for specific DTN application scenarios, there usually are some more useful characteristics or information, and making good use of them can further improve routing performance, for instance, the DTN application scenarios in social networks, in which nodes are not completely isolated. On the contrary, a node usually has social relationships or social ties with other nodes. For a university, the students
in the same class have strong relationships that we call classmates. In this case, such students will encounter each other much more frequently than the students outside of the class. Another example is that people having similar interests will regularly get together and eventually form a stable group. In other words, nodes in social networks are characterized by gregariousness. This gives us a heuristic that we can first forward message to the target social group where destination node resides and then flood message among the group in order to quickly and successfully finish message transmission.

In this paper, taking full use of the gregariousness characteristics of nodes in social networks, we proposed dynamic groups based adaptive DTN routing algorithm (DGA). For the reason that nodes in social networks have a variety of social relationships and common interests, we first abstract these social ties and then uniformly represent them using friendship. We do not care about the type of the relationship between two nodes. But we seriously focus on the strength of the relationship, which is usually reflected by the encounter probability between the two nodes. According to nodes’ social circles and similarities, we dynamically add nodes into different social groups and then implement the social group based flooding routing. Our main contributions are summarized as follows.

(i) We abstract all social relationships and uniformly represent them using friendship. And then we use encounter probability to measure the strength of the friendship between two nodes.

(ii) We first propose an ego group model to describe the friend circle of a certain node. And then based on the ego group, we add the node into a certain social group.

(iii) We propose a social group based flooding model to estimate encountered nodes, thus selecting fewer but better relay nodes so as to improve message delivery ratio.

(iv) We also present a message redundancy control model to avoid generating unnecessary redundant message copies, thus further controlling message redundancy.

The rest of this paper is organized as follows. Section 2 discusses some related works. Section 3 gives detailed descriptions of the proposed algorithm. The performance evaluations and comparisons, as compared to Bubble Rap, Epidemic, and ProPHET, are presented in Section 4. Finally, Section 5 summarizes this paper.

2. Related Works

Currently, scholars have proposed a large number of DTN protocols to address the end-to-end communication problem. Firstly, we gave a brief description of the two typical routing strategies Epidemic [10] and ProPHET [11] for the reason that our routing idea is inspired by them. Secondly, we introduced some latest social-based routing protocols.

Epidemic is a typical flooding-based routing algorithm, which tries to replicate messages to all encountered nodes for the purpose of quickly delivering message to destination node. In case of sufficient buffer resource and bandwidth, Epidemic may be able to get the best message delivery ratio. But in most DTN application scenarios, the buffer resource is extremely restricted. In this case, Epidemic is difficult to get satisfying overall routing performance. Despite this, we still think this flooding idea is acceptable, but the key issue is that Epidemic does not evaluate encountered nodes, thus leading to flooding message in the whole network. If we can control the flooding scope to a small group where the destination node resides, then we will get a better routing performance. Our routing idea comes in part from this.

ProPHET is a typical history utility based routing algorithm, which takes full use of the encounter history information and transitivity to predict delivery possibility. There are three delivery predictability metrics in ProPHET. The first metric is defined in (1), where \( P_{(ab)} \) is used to predict and update the delivery predictability that node \( a \) can successfully deliver message to destination node \( b \) and \( P_{\text{init}} \in [0, 1] \) is an initialization constant. The second metric defined in (2) is to update the delivery predictability as time progresses, in which \( \gamma \in (0, 1) \) is the aging constant and \( k \) is the number of time units that have elapsed since the last time the metric was aged. The last metric in (3) uses the transitive property to update the delivery predictability, where \( \beta \in [0, 1) \) is a scaling constant. In social networks, ProPHET is able to describe the case that the nodes with strong relationships will have higher encounter probabilities than strangers due to the higher encounter frequencies between them. This gives us the concept to abstract social relationships and the way to measure the strength of a certain relationship:

\[
P_{(ab)} = P_{(ab)\text{init}} + (1 - P_{(ab)\text{init}}) \times P_{\text{init}}, \quad (1)
\]

\[
P_{(ab)} = P_{(ab)\text{init}} \times \gamma^k, \quad (2)
\]

\[
P_{(ac)} = P_{(ac)\text{init}} + (1 - P_{(ac)\text{init}}) \times P_{(ab)} \times P_{(bc)} \times \beta. \quad (3)
\]

Bubble Rap [12] was the first proposed community based routing protocol, which relies on two social characteristics: community and centrality. Each node in the network belongs to at least one community and has a global rankness and a local rankness, which, respectively, represents the global popularity in the whole network and the local popularity in its local community. The delivery process of a certain message includes two phases. It first bubbles message up based on the global popularity until the message is delivered to the destination community to which the destination node belongs. Then in the local community, it bubbles the message up based on the local popularity until the message is delivered to its destination or its TTL is exhausted. In order to reduce the resource consumption, the original carrier deletes this message once the message has been successfully delivered to the destination community. The simulation results in this paper show that Bubble Rap gets certain advantages in delivery latency and average hop count.

Taking full use of human contact features, Wu and Wang propose a hypercube-based multipath social feature routing algorithm [13], which includes two unique phases: social feature extraction and multipath routing. Specifically,
in social feature extraction process, it first uses Shannon Entropy to capture the $m$ most informative features to build social feature space and then transforms the routing problem into hypercube-based feature matching problem. In the second phase, it proposes two kinds of multipath forwarding strategies. The extensive simulations using the real datasets show that the proposed algorithm achieved a significant improvement in the message delivery ratio and the end-to-end delivery delay.

ComPAS [14] is a typical community-based routing protocol proposed by Xia et al. By exploiting social relationship while replicating message in community, ComPAS can significantly improve routing performance and achieve better efficiency and consistency while keeping the replica relocation cost as low as possible.

Xiao et al. propose a distributed optimal community-aware opportunistic routing (CAOR [15]). By using a home-aware model, they turn mobile social networks into a network that only includes community homes. Besides, they also prove that, in the network of community homes, the minimum expected delivery delay can be computed through a reverse Dijkstra algorithm. Based on the home-aware model, CAOR achieves a satisfying opportunistic routing performance.

HS [16] is a zero-knowledge assisted mobile social network routing algorithm proposed by Wu et al., which spreads a given number of message copies in an optimal strategy by theoretical analysis when assuming that the intermeeting times between any two nodes and between a node and a community follow exponential distribution. In addition, by constructing a Markov chain, HS can also calculate the expected delivery delay and derive an upper bound, thus getting a better routing performance.

3. Routing Framework

Before presenting our DGA routing algorithm, we first define two groups: Ego Group and Social Group in this section, and then introduce our social group based flooding model and message redundancy control model.

3.1. Friendship Definition. There are too many kinds of social relationships in social networks. Besides, nodes’ social behaviors driven by common interests are also complicated. In this case, it may be very difficult to analyze and capture these relationships. In order to reduce routing complexity, we extract their common key features: a pair of nodes with certain social relationship or specific common interest will encounter each other more frequently than strange nodes. Moreover, the strength of their relationship can also be accurately measured by their encounter probabilities. In other words, frequent communications between two nodes indicate higher encounter probability values, which can in turn prove the existence of a strong relationship between two nodes. The key issue we are concerned with is the strength of the relationship, rather than the type of the relationship. Consequently, we abstract all social relationships and uniformly represent them using friendship. Furthermore, we use encounter probability to measure relationship strength.

In this paper, we use (4) to measure the friendship degree and estimate whether two nodes could become friends, where $P_{\text{friend}} \in (0, 1)$ is an initialization constant representing the threshold to set up a friendship and can be set to an appropriate value according to the specific application scenario, and $P_{(a,b)}$ is shown in (1). If $P_{(a,b)} > 0$, then $a$ and $b$ are friend nodes. Finally, a pair of friends is defined as a pair of nodes with a strong social relationship (i.e., a high encounter probability). Here, we only focus on the direct friendship between two nodes. Therefore, we no longer use transitivity to predict the encounter probability. But taking into consideration that the established friendships may be broken as time progresses, we still use the aging (2) to update the encounter probability. With the aging mechanism, two nodes are able to determine whether they are still friends in real time:

$$\text{Friend}_{(a,b)} = P_{(a,b)} - P_{\text{friend}} \quad (4)$$

3.2. Ego Group Definition. In social networks, due to different social behaviors and interests, a node will have different friends. Under normal circumstances, most friends of a node are not each other’s friends or even do not know each other. The only social tie is that they have a common friend for a certain reason. In this case, the friends of a certain node are usually not closely associated. However, these friends’ information are very useful and can be used to determine the social group (defined in next subsection) where the node resides. Here, we define an Ego Group to describe all the friends of a node. As shown in Figure 1, nodes $a$–$e$ become the friends of node $S$, respectively, for different reasons. Then we define the ego group of $S$ as the node set that consists of the node $S$ and all the friends of $S$ (i.e., nodes $a$–$e$). Now for a special example, if there is another node $D$, which has an ego group that also contains the nodes $a$–$e$, then $S$ and $D$ should be assigned to the same social group because they have a similar friend circle.

We assume that each node in social networks has one and only one ego group, in which the node is the unique ego node (i.e., the owner of the ego group) and other nodes are all its friends. The special instance is the ego group only containing the ego node itself. For an ego group, we do not care about the relationships between the friends of the ego node, but we do care about the composition of the ego group. Using (4), a node can easily establish its ego group and maintain it. As time progresses, if a node is no longer the friend of the ego node, then it should be deleted from the ego group.

3.3. Social Group Definition. In social networks, due to complicated relationships and common interests, the behaviors of nodes are characterized by gregariousness. In other words, there is a node set, in which member nodes are closely associated with each other and encounter each other much more frequently than nodes outside of the node set. If we can find such a node set where destination node resides, we can flood message only among the node set in order to improve
message delivery ratio in the case of controlling network overhead. In this paper, we call the above node set the social group. Note that ego group can only reflect that the ego node encounters its friends very frequently but cannot reflect the encounter frequencies between the friends of the ego node.

Now there is a key problem: how to dynamically establish these social groups. Usually, the similar nodes should be added into the same social group. Ego group can reflect the friend circle information of an ego node, which is very useful and can be used as the metric to estimate the similarities between nodes. In this paper, we use the ego group information of a node to determine the social group where the node resides. Concretely, grouping decisions can be made when one of the following cases occurs.

(i) If the friend circles of two ego nodes are almost the same and they are also each other’s friends, then the two nodes are assigned to the same social group.

(ii) If most member nodes of a certain social group are the friends of a node, then the node should also be added to the social group.

The detailed establishment process of social group is presented in Algorithm 1. For the above two cases, we, respectively, define the threshold parameters $T_{eg}$ and $T_{sg}$ to test whether the current node meets these cases. Lines 1-2 and line 8 are testing the first case by using $T_{eg}$. Lines 3–5 and line 10 are testing the second case by using $T_{sg}$. The two parameters can be set to an appropriate value according to the specific application scenario, thus controlling the size of social group. By increasing their values, we can reduce the group size and get a more closely associated social group and vice versa. In this paper, taking into account that we abstract all social relationships, we further assume that each node belongs to and only belongs to one social group. The smallest social group contains only the node itself. In this case, adding a node to a social group is to merge the two social groups. As shown in line 9 and line 11, when adding node $b$ to node $a$’s social group, the update process needs to merge the two social groups $SG_a$ and $SG_b$. But in this moment, only the two nodes can capture and update the group changes. The other member nodes do know nothing about these changes. Fortunately, the member nodes of a social group are closely associated, and they encounter each other very frequently. We can make full use of these frequent encounter opportunities to capture and update those group changes. The update process of lines 6-7 is to finish this operation when encountering another member node of the same social group. Note that the case that a node moves out of its social group may also occur. If and only if a node is no longer a friend of any other social group member node, the node should be removed from the social group. This kind of information should also be captured and updated by the update process (i.e., the update process in lines 6-7) in every encounter chance.

### 3.4. Social Group Based Flooding Model

Based on the above social group information, we can finally implement our flooding routing model. Here we first define the node sets $N_i$ and $SG_{id}$, where $N_i$ denotes the node set that have been encountered by node $i$, and $SG_{id}$ represents the target social group where destination node $d$ resides.

The core strategy in our routing model is that message is flooded to all member nodes once the message is propagated to the target social group where the destination node resides.
which we call social group based flooding. For the purpose of successfully and quickly spreading message into the target social group, we also need to seek help from some intermediate relay nodes. Consequently, the choice of these intermediate relay nodes will greatly affect final routing performance and becomes the emergent thing we need to tackle with. In order to select fewer but better intermediate relay nodes, we attempt to compute the potential likelihood that a node can finish message transmission based on social group flooding. For this purpose, we first need to model group flooding process as a two-virtual-hop routing model. Then we can easily compute the potential likelihood of successful message transmission when the message has been delivered into the target social group through a group member (e.g., node $g_i$ in Figure 2). Figure 2 shows an example of the two-virtual-hop model of the target social group $SG_d$ and describes the possible flooding process starting from node $g_i$. Concretely, a solid arrow in the figure represents a direct end-to-end path from $g_i$ to an intermediate member node and a dashed arrow represents a virtual link from the intermediate member node to destination node $g_d$. Each virtual link consists of all the possible routes from the intermediate member node to $g_d$ that do not pass through $g_i$. In other words, all these possible routes that do not pass through node $g_i$ are abstracted as a virtual link.

In the above abstracted model, there will be $|SG_d| - 1$ different virtual routes from $g_i$ to destination node $g_d$ once message is delivered into the target social group $SG_d$ through the member node $g_i$. For the path $g_i$ $→$ $g_j$ $→$ $g_d$, we use $P_{g_i→g_j}(g_d)$ to represent the delivery possibility that node $g_i$ successfully deliver message to destination $g_d$. Likewise, $P_{g_j→g_d}(g_i)$ is used to represent the total possibility that node $g_j$ can deliver message to $g_d$ along the virtual link $g_j$ $→$ $g_d$ in the case of not passing through $g_i$. Obviously, we have the following equation:

$$P_{g_i→g_j}(g_d) = P_{(g_j,g_d)} \times P_{g_i→g_j}(g_i, g_d).$$  \hspace{1cm} (5)

And for the direct end-to-end path $g_i$ $→$ $g_d$, we also have

$$P_{g_i→g_d}(g_d) = 1.0.$$  \hspace{1cm} (6)

The message may be successfully delivered through any one of the above $|SG_d| - 1$ virtual paths. So taking into account all of them, we can compute the total delivery possibility $P^{SG_d}_{(g_i,g_d)}$ that $g_i$ can successfully deliver a message to destination node $g_d$ by flooding the message among the target social group $SG_d$:

$$P^{SG_d}_{(g_i,g_d)} = 1 - \prod_{g_j \in SG_d \backslash g_i} (1 - P_{g_i→g_j}(g_d))$$

$$= 1 - \prod_{g_j \in SG_d \backslash g_i} (1 - P_{(g_j,g_d)} \times P_{g_i→g_j}(g_i, g_d)).$$ \hspace{1cm} (7)

Here, if $g_i = g_d$, we set

$$P^{SG_d}_{(g_i,g_d)} = P^{SG_d}_{(g_i,g_i)} = 1.0.$$ \hspace{1cm} (8)

Now, in order to get the final value of $P^{SG_d}_{(g_i,g_d)}$, we only need to compute $P^{SG_d}_{(g_i,g_d)}$. Since the virtual link from $g_i$ to $g_d$ does not pass through node $g_i$, the longest path will comprise $|SG_d| - 1$ nodes and the shortest route only comprises 2 nodes. Consequently, the number of all possible paths from $g_i$ to $g_d$ in the virtual link is $C_A$:

$$C_A = A_{|SG_d|−3}^0 + A_{|SG_d|−3}^1 + \cdots + A_{|SG_d|−3}^{|SG_d|−4} + A_{|SG_d|−3}^{|SG_d|−3}$$

$$= \sum_{i=0}^{|SG_d|−3} A_i^{|SG_d|−3}. \hspace{1cm} (9)$$

But due to the strictly limited processing capability in DTNs, it is difficult to compute all delivery possibilities of these routes. Besides, for most DTN applications, the network bandwidth and buffer resource are very precious. Consequently, one optimization goal of DTN routing is to control average hop count for the purpose of further reducing the cost of message transmission. Under this circumstance, we further simplify the virtual link by controlling the maximum hop count. In other words, we only take into account the delivery likelihood within two hops scope when computing $P^{SG_d}_{(g_i,g_d)}$. If the path from $g_i$ to $g_d$ is only one hop, we get the delivery probability

$$P^{1\text{hop}}_{g_i→g_d}(g_i, g_d) = P^{(g_i,g_d)}.$$ \hspace{1cm} (10)
If the path from \( g_i \) to \( g_d \) is only two hops, we get the delivery probability
\[
P_{g_i}^{\text{hop}}(g_i, g_d) = 1 - \prod_{g_j \in \text{SG}_d, g_j \neq g_i, g_j \neq g_d} \left(1 - P(g_i, g_j) \times P(g_j, g_d)\right).
\] (11)

Now within two hops scope, we can finally get
\[
P_{g_i}^*(g_i, g_d) = 1 - \left(1 - P_{g_i}^{\text{hop}}(g_i, g_d)\right) \times \left(1 - P_{g_i}^{\text{hop}}(g_i, g_d)\right)
= 1 - \left(1 - P(g_i, g_d)\right) \prod_{g_j \in \text{SG}_d, g_j \neq g_i, g_j \neq g_d} \left(1 - P(g_i, g_j) \times P(g_j, g_d)\right).
\] (12)

Here, we analyze the cost of computing \( P_{g_i}^*(g_i, g_d) \). When computing \( P_{g_i}^{\text{hop}}(g_i, g_d) \), we can easily get the cost
\[
\text{Cost}_{\text{hop}} = O(1).
\] (13)

To compute \( P_{g_i}^{\text{hop}}(g_i, g_d) \), we need to do \(|\text{SG}_d| - 3\) + \(|\text{SG}_d| - 4\) multiplication operations and \(|\text{SG}_d| - 2\) subtraction operations. So we get
\[
\text{Cost}_{\text{hop}} = (|\text{SG}_d| - 3) + (|\text{SG}_d| - 4) + (|\text{SG}_d| - 2)
= 3|\text{SG}_d| - 9 = O(|\text{SG}_d|).
\] (14)

Then we can get the total cost when computing \( P_{g_i}^*(g_i, g_d) \):
\[
\text{Cost} = \text{Cost}_{\text{hop}} + \text{Cost}_{\text{hop}} + 4 = O(|\text{SG}_d|).
\] (15)

By controlling the cost to \( O(|\text{SG}_d|) \) level, DTN node can easily compute \( P_{g_i}^*(g_i, g_d) \). And then with (7), node can finally compute the delivery likelihood \( P_{g_i}^*(g_i, g_d) \). For this purpose, every social group should maintain a probability matrix \( \text{PM}_{\text{SG}_d} \) as described below to record and update the encounter probability between two group member nodes. And finally we can get the matrix \( \text{PM}_{\text{SG}_d, \text{SG}_d} \), in which the element in ith row and jth column record the \( P_{g_i, g_j}^* \) for nodes \( g_i \) and \( g_j \):

\[
\text{PM}_{\text{SG}_d} = \begin{bmatrix}
P_{g_i, g_d} & \cdots & P_{g_i, g_{|\text{SG}_d|}} \\
\vdots & \ddots & \vdots \\
P_{g_{|\text{SG}_d|}, g_d} & \cdots & P_{g_{|\text{SG}_d|}, g_{|\text{SG}_d|}}
\end{bmatrix}
\]

\[
P_{g_i, g_d} = 1.0, \quad 1 \leq i \leq |\text{SG}_d|.
\] (16)

Now with \( \text{PM}_{\text{SG}_d, \text{SG}_d} \) at hand, we can predict the potential delivery likelihood that an intermediate node (does not belong to the target social group \( \text{SG}_d \)) can successfully finish message transmission in the assistance of the target social group members it encounters. Afterwards, in order to successfully and quickly spread message into the target social group, current node will select an intermediate node as the next hop relay node if the node has a bigger delivery likelihood than current node. Now we assume that current node \( i \) has a message destined for node \( g_d \) and node \( i \) does not belong to the target social group \( \text{SG}_d \). When encountering a member node of \( \text{SG}_d \), node \( i \) will certainly deliver the message to the member node. Then the message will be flooded among the target group \( \text{SG}_d \) until finishing message transmission. But when meeting an intermediate node, which does not belong to the target social group, current node needs to decide whether to deliver the message to the encountered node. For the convenience of describing the detailed process, we define the node set \( \text{NG}_{i,d} \) as follows, which represents the group member nodes of \( \text{SG}_d \) encountered by node \( i \):

\[
\text{NG}_{i,d} = \{ j \mid P(i,j) > 0, \quad j \in \text{SG}_d \}.
\] (17)

Here we can get the matrix \( M_f \), which records the encounter possibilities between node \( i \) and all the nodes in \( \text{NG}_{i,d} \). And then we get the matrix \( M_{\text{SG}_d} \), which records the \( P_{g_j, g_{|\text{SG}_d|}} \) for each \( g_j \in \text{NG}_{i,d} \) and destination node \( g_{|\text{SG}_d|} \). After transforming \( M_f \) to the diagonal matrix \( M_f^{\text{diag}} \), we can compute the final matrix \( M_f \) by the matrix of \( M_f^{\text{diag}} \times M_{\text{SG}_d} \):

\[
M_f = \begin{bmatrix}
P_{(i,g_i)} & \cdots & P_{(i,g_{|\text{SG}_d|})}
\end{bmatrix}
\]

\[
M_{\text{SG}_d} = \begin{bmatrix}
P_{g_{|\text{SG}_d|} \cdot g_{|\text{SG}_d|}}
\vdots
P_{g_j \cdot g_{|\text{SG}_d|}}
\end{bmatrix}
\]

\[
M_f^{\text{diag}} = \text{diag}(M_f)
\]

\[
M_f^{\text{diag}} = \begin{bmatrix}
P_{(i,g_i)} & \cdots & P_{(i,g_{|\text{SG}_d|})}
\end{bmatrix}
\]
Now, each element in $M_p$ denotes the possibility that node $i$ can successfully deliver message to destination $g_d$ through a certain target social group member in $NG_d$. Finally with the matrix $M_p$, we can get the potential possibility $P_{(i,g_d)}$ that intermediate node $i$ (does not belong to the target group) can successfully finish message transmission in the assistance of the target social group members it will encounter in the near future. Obviously, we have

$$M_p = M^\text{diag}_i \times M_{SG,i}$$

(18)

$$P_{(i,g_d)} = P_{(i,g_d)} \times P_{SG,i}$$

$$P_{(i,g_d)} = P_{(i,g_d)} \times P_{SG,i}$$

(19)

3.5. Message Redundancy Control Model. In most DTN application scenarios, the node buffer resource is extremely limited, which determines that the number of message copies should be controlled to get a better routing performance. As a result, improving the utilization efficiency of network resources for each message copy can further increase message delivery ratio. For this purpose, we make attempts to reduce message redundancy when selecting intermediate relay nodes to spread message to target social group. Assuming that node $i$ and node $j$ both carry a same message copy, then for the node encountered by both $i$ and $j$, one of the two message copies is redundant. Concretely, if most nodes that node $i$ may meet will also be encountered by node $j$, then one of the two message copies is considered to be unnecessary. Here based on these considerations, we propose a message redundancy model to avoid creating such redundant message copies when selecting next hop relay nodes.

Here, we assume that the nodes encountered by a node in the past are possible to be encountered again. Then we use the encountered node set to predict the node set that a node will encounter in the near future. For current node $i$ and its neighbor node $j$, we denote the node set $N_i \cap N_j$ as the nodes that will be encountered by both node $i$ and node $j$, which is node set $\{1, 2, 3\}$ as shown in Figure 3:

$$N_i \cap N_j = \{k \mid P_{(i,k)} > 0, P_{(j,k)} > 0\}.$$  

(20)

Now, we describe the detailed message redundancy model as follows. For a node $k$, there are two cases to occur: $k$ will be encountered by both $i$ and $j$ in the near future or not. We represent the two cases using

$$y_k = \begin{cases} 1 & \text{if } i \text{ and } j \text{ will both encounter } k \\ 0 & \text{else.} \end{cases}$$

(21)

According to encounter probabilities between nodes, we can predict the probability that $i$ and $j$ both encounter $k$. Then through a subtraction operation, we can get the probability of the other case. Obviously, we have

$$P(y_k = 1 \mid N_i, N_j) = P_{(i,k)} \times P_{(j,k)}$$

$$P(y_k = 0 \mid N_i, N_j) = 1 - P(y_k = 1 \mid N_i, N_j).$$

(22)

Assuming that node $i$ replicates a message to node $j$, then we can predict whether there will be redundant message copies via node $k$ by comparing the probabilities computed by (22). As defined below, if the probability of $y_k = 1$ is bigger than the probability of $y_k = 0$, then there will be a redundant message copy via node $k$ (i.e., $Y_k = 1$):

$$Y_k = \begin{cases} 1 & \text{if } P(y_k = 1 \mid N_i, N_j) > P(y_k = 0 \mid N_i, N_j) \\ 0 & \text{if } P(y_k = 1 \mid N_i, N_j) \leq P(y_k = 0 \mid N_i, N_j). \end{cases}$$

(23)

Finally for each node $k$ in $N_i \cap N_j$, we get the vector

$$Y = \langle Y_1, \cdots, Y_k, \cdots, Y_{|N_i \cap N_j|} \rangle.$$  

(24)

With the vector $Y$, we use (25) to compute the message redundancy ratio $MR_{(i,j)}$ for node $j$:

$$MR_{(i,j)} = \frac{1}{|N_j|} \sum_{k \in N_i \cap N_j} Y_k.$$  

(25)

It indicates that node $i$ has been able to cover almost all nodes that node $j$ will encounter once the value of $MR_{(i,j)}$ is close to 1. In this case, node $j$ should not be selected as next hop relay node for avoiding message redundancy. On the contrary, node $j$ is considered to be a good choice if the value of $MR_{(i,j)}$ is close to 0 since node $i$ and node $j$ have different social circles in that case. To address this problem, we define a message redundancy variable named as $MR_{\text{threshold}} \in (0, 1)$, which can be flexibly set to an appropriate value according to specific application scenario. When $MR_{(i,j)} < MR_{\text{threshold}}$, node $j$ is considered as a good choice to act as a relay node and vice versa. Aiming to further control network overhead ratio, we propose to set $MR_{\text{threshold}}$ to a small value.
Input:
one-hop neighbors of \( N_i \): neigh_list
messages stored in \( N_i \): msg_list
the value of MR\(_{\text{threshold}}\): MR

output:
(1) For message in msg_list
(2) \( \text{dest} \leftarrow \text{message.destination} \)
(3) \( \text{SG}_d \leftarrow \text{the target social group of dest} \)
(4) For node in neigh_list
(5) \( \text{EG} \leftarrow \text{the egogroup of node} \)
(6) If node = dest OR node \( \in \) SG\(_d\) OR dest \( \in \) EG
replicatemessage to node
(8) Else if \( N_i \in \text{SG}_d\)
(9) If \( P_{L}^{\text{node,dest}}(N_i,\text{dest}) > P_{L}^{\text{SG}_d}(N_i,\text{dest}) \) AND \( \text{MR}(N_i,\text{node}) < \text{MR} \)
replicatemessage to node
(11) End if
(12) Else
(13) If \( P_{L}^{\text{node,dest}}(N_i,\text{dest}) > P_{L}^{\text{EG}}(N_i,\text{dest}) \) AND \( \text{MR}(N_i,\text{node}) < \text{MR} \)
replicatemessage to node
(15) End if
(16) End if
(17) End for
(18) End for

Algorithm 2: DGA routing algorithm on node \( N_i \).

3.6. Detailed DGA Routing Algorithm. Based on the above works, we finally implement our routing algorithm as illustrated in Algorithm 2. There are two cases that current node should deliver message to a node: one is that the node belongs to the social group of destination node, the other is that destination node belongs to the node’s egogroup. Lines 6-7 are to test these cases and finish message transmission. Otherwise, by using the proposed social group based flooding model to compute the potential delivery probability and using the message redundancy control model to reduce redundant messages, lines 8–14 select fewer but better relay nodes to spread message copies in order to quickly deliver message to the target social group.

4. Simulation

By using the ONE [17] simulator, we conduct extensive simulations to evaluate the performance of DGA under various settings. The compared algorithms, simulation settings, evaluation metrics, and results are described as follows.

4.1. Compared Algorithms. For evaluations in this paper, we only focus on the routing algorithms similar to DGA. Firstly, considering that DGA takes advantage of social behaviors among nodes, we add the typical social-based routing algorithm Bubble Rap [12] to the comparisons. This can evaluate the improvements of DGA in terms of using social properties. Secondly, taking into account that DGA abstracts all social relationships and further uses encounter probability to measure relationship strength, we also add ProPHET to the evaluations. The concept of encounter probability presented by ProPHET can be considered the most essential form of social relationships. So in this case, ProPHET can also be regarded as a social tie based routing algorithm. The concept of abstracting social relationships is also from ProPHET. Lastly, the core routing strategy of DGA is flooding based on social group, which comes in parts from Epidemic but different from Epidemic. Introducing Epidemic to the simulations is to find out the improvements of local flooding (DGA) compared to typical global flooding (Epidemic).

4.2. Simulation Settings. To evaluate the routing performance of DGA based on social behaviors, we conduct all simulations on synthetic traces generated by the working day movement (WDM) model [18]. This is because the WDM model brings more reality to the human movement by modeling three major activities typically performed by humans during a working week: sleeping at home, working at the office, and going out with friends. Beyond the activities themselves, the WDM model also includes different transport models. The nodes can move alone or in groups by walking, riding, or driving. The ability to move alone or in groups at different speeds increases the heterogeneity of movement which has impact on the performance of, for example, routing protocols. In addition, WDM introduces communities and social relationships. The communities are composed from nodes which work in the same office, spend time in the same evening activity spots, or live together. Moreover, we can modify the model parameters as needed, so that it can reproduce various empirical mobility properties, which is beneficial to
the routing performance evaluations. This is another reason that we use the traces generated by WDM.

In the trace-driven simulations, there are 50 working offices and 10 meeting spots. Working day length is 4 hours and office wait time is about 10 minutes to 4 hours. The probability of shopping after work is 0.5 and shopping time is about 1-2 h. We totally use 110 nodes, which consist of 10 bus nodes and 100 human nodes. In detail, bus nodes are based on bus movement model and divided into 5 bus groups. Each bus group travels along different traffic routes and consists of two bus nodes. Each bus node moves with speed of 7–10 m/s, wait time of 10–30 s, transmit speed of 10 Mbps, and transmit range of 1000 m. The human nodes are based on WDM model and also divided into 5 groups. Each human group consists of 20 nodes and they have their own meeting spots, homes, and working offices. Each human node moves with speed of 0.8–1.4 m/s, transmit speed of 2 Mbps, and transmit range 20 m.

All other default simulation settings are shown in Table 1.

The simulations are grouped into the three categories: varying buffer size, varying message’s time-to-live, and varying message generation interval. When varying a single setting parameter in simulations, the other setting parameters follow the default settings in Table 1. Firstly, we investigated the routing performance of the four algorithms when varying buffer size from 20 M to 120 M. Secondly, we evaluated their routing performance when varying message’s time-to-live from 4 hours to 24 hours. Lastly, we conducted the simulations when varying the message generation interval from 20 seconds to 100 seconds. All evaluation results are shown in Section 4.4.

### 4.3. Evaluation Metrics

Under the same guideline, we evaluated the above four routing algorithms based on the following metrics.

1. **Delivery ratio**: normally, the ultimate routing goal in DTNs is to successfully deliver message to its destination. This metric is the measure of delivery capability for each algorithm.

2. **Overhead ratio**: it is desirable to have a low overhead ratio, since it reflects the efficiency of message transmission.

3. **Average latency**: end-to-end latency is another important routing goal. A lower average latency means a better routing performance.

4. **Average hop count**: minimizing the number of hops that a message must take in order to reach the destination is a routing goal to reduce transmission cost, such as bandwidth and energy.

### 4.4. Simulation Results

#### 4.4.1. Performance Evaluations by Varying Buffer Size

Regarding the simulation results in Figure 4, DGA achieves the highest delivery ratio and the lowest overhead ratio. The average hop count of DGA is almost as few as Bubble Rap and is fewer than Epidemic and ProPHET. In addition, DGA is also able to get some advantages in delivery latency compared to Epidemic and ProPHET. All these results can verify the great improvements of DGA in routing performance.

As shown in Figures 4(a) and 4(b), DGA can outperform Bubble Rap, Epidemic, and ProPHET in terms of message delivery ratio and overhead ratio. This is because DGA takes full use of the gregariousness characteristic of moving nodes. Based on the proposed ego group model and social group model, DGA can find out the friend nodes that are closely associated with destination node and then only floods message among the target social group. Besides, based on the social group based flooding model, current node is also able to select better nodes as next hop relay nodes to quickly and successfully deliver message to the target social group. In this case, DGA achieves a higher delivery ratio than Bubble Rap. In addition, the proposed message redundancy control model further helps DGA to get a lower overhead ratio. DGA and ProPHET both use the encounter probability to measure the strength of the social tie between two nodes, but ProPHET does not take further optimization strategy to improve routing performance. On the contrary, ProPHET just naively delivers message to the encountered nodes that have bigger probabilities than current node. Therefore the routing performance of ProPHET is worse than DGA. Compared to the global flooding strategy of Epidemic, DGA accurately controls the flooding scope based on social group model, thus greatly avoiding creating too many redundant messages and further controlling network overhead ratio. And DGA takes into consideration nodes’ social relationships for improving routing performance. As a result, DGAs significantly outperform Epidemic.

From Figures 4(c) and 4(d) we can see that the average latency and average hop count of DGA and Bubble Rap are lower than those of Epidemic and ProPHET when buffer size is more than 40 M, which reflects the fact that the routing strategies of DGA and Bubble Rap are more accurate, thus efficiently reducing the cost of message transmission.

Finally from the whole Figure 4, we can make a conclusion that DGA can outperform Epidemic and ProPHET in social networks in terms of delivery ratio, overhead ratio,
average latency, and average hop count. And although the average latency and average hop count of DGA are slightly higher than those of Bubble Rap, the message delivery ratio of DGA is higher than Bubble Rap and the overhead ratio of DGA is only about 50% of that of Bubble Rap. As a result, from the point of view of high delivery ratio and low overhead ratio, DGA is able to outperform Bubble Rap.

4.4.2. Performance Evaluations by Varying Message TTL. The results in Figure 5 show that DGA gets the highest delivery ratio when message time-to-live is more than 8 hours and achieves the lowest overhead ratio when message TTL is less than 16 hours. As same as shown in Figure 4, DGA can still achieve satisfying delivery latency and average hop count. These simulation results prove once again the improvements and routing efficiency of DGA.

From Figure 5(a), we can find that message TTL can significantly affect the message delivery ratios of the four algorithms when buffer size is relatively sufficient. When message TTL is less than 12 hours, the delivery ratios of the four algorithms all keep increasing. This is because that message has a longer time to seek destination node in this case. But when message TTL is more than 12 hours, their delivery ratios keep decreasing. The reason is that long message TTL also leads to the presence of more messages, which exacerbates the consumption of cache resource. However, DGA can keep a higher delivery ratio compared to Bubble Rap, Epidemic, and ProPHET when message TTL is more than 8 hours. To a certain degree, Bubble Rap is able to avoid
generating too many redundant messages by selecting relay nodes based on social behaviors of nodes. But DGA can further reduce redundant messages by using the proposed message redundancy model while selecting relay nodes based on social behaviors. On the contrary, Epidemic and ProPHET do not take efficient strategy to control message redundancy. In this case, they will inevitably create many redundant messages. Therefore from the perspective of the message redundancy, the increases of message TTL help DGA to get a higher delivery ratio. In Figures 5(b), 5(c) and 5(d), DGA and Bubble Rap can also keep their advantages in overhead ratio, delivery latency, and average hop count with the increases of message TTL.

To sum up, considered from message delivery ratio and network overhead ratio, DGA can outperform Epidemic and ProPHET and achieve certain advantages compared to Bubble Rap.

4.4.3. Performance Evaluations by Varying Message Generation Interval. As shown in Figure 6, DGA achieves the highest message delivery ratio and the lowest overhead ratio when message generation interval is more than 20 seconds. In addition, DGA gets significant advantages in delivery latency and average hop count compared to Epidemic and ProPHET. From these results, we can verify the great improvements of DGA.

In Figure 6(a), the delivery ratios of the four routing algorithms all keep increasing with the increase of message generation interval, but DGA can keep the highest delivery.
ratio. The key reason is that the number of new messages generated by network will be reduced when message generation interval keeps increasing, thus weakening the competition for cache resource. In this case, all algorithms can use these relatively sufficient buffer resources to better deploy their routing strategies. Furthermore, DGA can efficiently select fewer but better relay nodes to deliver message to the target social group based on the proposed social group flooding model and find out the nodes that are closely associated with destination node for locally flooding message based on social group model. Thus, DGA can get a higher delivery ratio. Moreover, in Figure 6(b), the overhead ratio of DGA keeps decreasing, while the overhead ratios of the other three algorithms are increasing. This is because DGA can further control message redundancy by using the proposed message redundancy control model, which can verify the efficient improvements of DGA in controlling network overhead ratio. Consequently, DGA gets a lower overhead ratio with the increase of message generation interval.

Similar to Figures 4 and 5, DGA can get the lower delivery latency and fewer average hop count compared to Epidemic and ProPHET, although the delivery latency and average hop count of DGA are still slightly higher than those of Bubble Rap. But taking into account the message delivery ratio and overhead ratio, DGAs still outperform the other three routing algorithms.

5. Conclusion

In this paper, we proposed the dynamic groups based adaptive DTN routing algorithm DGA for social networks. Taking
into consideration a variety of social relations and their different forms, we first abstract these social relationships and then measure them using the encounter probability. In order to take full use of the gregariousness characteristic of moving nodes, we define the ego group model and social group model to dynamically establish different social groups. Based on social group information, we only flood message among the target social group where destination node resides, thus efficiently controlling the flooding scope in the case of improving message delivery ratio. In order to quickly and successfully deliver message to its target social group, we proposed a social group based flooding model to help current node to select better next hop relay nodes. Furthermore, we also present a message redundancy control model to avoid generating unnecessary message copies, thus further controlling message redundancy and reducing network overhead ratio. Extensive simulation results show that the proposed DGA can outperform Bubble Rap, Epidemic, and ProPHET in terms of message delivery ratio and overhead ratio.

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

Acknowledgments
This research is supported in part by Foundation Research Project of Qingdao Science and Technology Plan under Grant no. 12–1–4–2–(14)–jch and Natural Science Foundation of Shandong Province under Grant no. ZR2013FQ022.

References
[1] V. N. G. J. Soares, J. J. P. C. Rodrigues, and F. Farahmand, “GeoSpray: a geographic routing protocol for vehicular delay-tolerant networks,” Information Fusion, vol. 15, pp. 102–113, 2014.
[2] 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 Workshops: Conference on Computer Communications, pp. 244–251, August 2005.
[3] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, “Diversity of forwarding paths in pocket switched networks,” in Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference (IMC ’07), pp. 161–174, October 2007.
[4] J. Partan, J. Kurose, and B. N. Levine, “A survey of practical issues in underwater networks,” SIGMOBILE Mobile Computing Communications Review, vol. 11, no. 4, pp. 23–33, 2007.
[5] V. Nikolaos and Y. Kun, “Mobile social networks: architecture, social properties, and key research challenges,” IEEE Communications Surveys & Tutorials, vol. 15, no. 3, 2013.
[6] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, “Applications, architectures, and protocol design issues for mobile social networks: a survey,” Proceedings of the IEEE, vol. 99, no. 12, pp. 2130–2158, 2011.
[7] X.-M. Fan, Z.-G. Shan, B.-X. Zhang, and H. Chen, “State-of-the-art of the architecture and techniques for delay-tolerant networks,” Acta Electronica Sinica, vol. 36, no. 1, pp. 161–170, 2008.
[8] K. Fall and S. Farrell, “DTN: an architectural retrospective,” IEEE Journal on Selected Areas in Communications, vol. 26, no. 5, pp. 828–836, 2008.
[9] V. Cerf, S. Burleigh, A. Hooke et al., Delay-Tolerant Networking Architecture, RFC4838, IETF, 2007.
[10] A. Vahdat and D. Becker, Epidemic Routing For Partially Connected Ad Hoc Networks, Duke University, Durham, NC, USA, 2000.
[11] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” SIGMOBILE Mobile Computing Communications Review, vol. 7, no. 3, pp. 19–20, 2003.
[12] P. Hui, J. Crowcroft, and E. Yoneki, “BUBBLE Rap: social-based forwarding in delay-tolerant networks,” IEEE Transactions on Mobile Computing, vol. 10, no. 11, pp. 1576–1589, 2011.
[13] J. Wu and Y. Wang, “Hypercube-based multi-path social feature routing in human contact networks,” IEEE Transactions on Computers, vol. 63, no. 2, pp. 383–396, 2014.
[14] F. Xia, A. Ahmed, L. Yang, J. Ma, and J. Rodrigues, “Exploiting social relationship to enable efficient replica allocation in Ad-hoc social networks,” IEEE Transactions on Parallel and Distributed Systems, 2014.
[15] M. J. Xiao, J. Wu, and L. S. Huang, “Community-aware opportunistic routing in mobile social networks,” IEEE Transactions on Computers, 2013.
[16] J. Wu, M. J. Xiao, and L. S. Huang, “Homing spread: community home-based multi-copy routing in mobile social networks,” in Proceedings of IEEE INFOCOM, pp. 2319–2327, 2013.
[17] A. Keranen, J. Ott, and T. Karkkainen, “The ONE simulator for DTN protocol evaluation,” in Proceeding of the Second International Conference on Simulation Tools and Techniques, pp. 1–10, 2009.
[18] F. Ekman, A. Keränen, J. Karvo, and J. Ott, “Working day movement model,” in Proceeding of 1st ACM/SIGMOBILE Workshop on Mobility Models for Networking Research, pp. 33–40, 2008.
