SMART: A Social and Mobile Aware Routing Strategy for Disruption Tolerant Networks

Konglin Zhu¹, Wenzhong Li¹,², Xiaoming Fu¹

¹ Institute of Computer Science, University of Goettingen, Goettingen, Germany
² State Key Laboratory for Novel Software and Technology, Nanjing University, Nanjing, China

¹{zhu, fu}@cs.uni-goettingen.de, ²lwz@dislab.nju.edu.cn

Abstract—Disruption tolerant networks (DTNs) are sparse mobile ad hoc networks where nodes connect with each other intermittently and end-to-end communication paths do not exist. Data routing in DTNs is challenging and draw much attention from research communities recently. Although many DTN routing strategies have been proposed in the past years, they confront problems such as blind spot and dead end, and also lack of efficient implementation in a decentralized, large-scale, mobile, and dynamic environment. To overcome these difficulties, we introduce a new solution for DTNs which leverages social properties and mobility characteristics of users. Our observation to the mobile trajectories of three data sets collected from real DTNs reveals that user movements appear locality and they tend to form communities correlated to geographic locations. Based on these findings, we propose a Social and Mobile Aware routing sTrategy (SMART). It exploits a distributed community partitioning algorithm to divide the DTN into smaller communities regarding user locations and interaction routines. For intra-community communication, a decayed routing metric convoluting social similarity and social centrality is calculated, which is used to decide forwarding node efficiently while avoiding the blind spot and dead end problems. To enable efficient inter-community communication, we choose the fringe nodes which travel remotely as relays, and propose the node-to-community utilities for routing decision across communities. We present empirical analysis to show that SMART reduces the occurrence of blind spot and dead end to a level below 1%. The efficiency of SMART is evaluated by extensive trace-driven experiments, which illustrate that it outperforms other routing strategies in various real DTN traces.

I. INTRODUCTION

A disruption tolerant network (DTN) is a sparse dynamic wireless network where mobile nodes work on ad hoc mode and forward data opportunistically upon contacts [15]. Since DTNs allow people to communicate without network infrastructure, they are widely used in battlefield, wildlife tracking, and vehicular communication etc. where setting up network infrastructure is hard and costly [2], [6], [24]. A key challenge of DTN communication is on routing, where sending data from the source to the destination via intermittent links in mobile environments is difficult. To achieve communication without setting up end-to-end communication paths, data transmission in DTNs employs the “store-carry-forward” manner, where a node stores and carries data while moving, forwards the data to a relay node on encountering, and propagates the data to further relays until the destination is reached. Routing for DTNs have been widely studied in the past, which fall into two categories: utility-based routing strategies and community-based routing strategies.

Utility-based routing strategies use a utility function for data forwarding decision making and choose the nodes with higher utility values as relays for data delivery. The most widely used utility metric is the frequency of node encounters, where messages are forwarded to a node who is more frequently meeting the destination than the current node [6], [24], [33]. Another important utility metric is the inter-contact time of node pairs, where massages are forwarded to the nodes with smaller inter-contact time to reduce transmission delay [2], [9], [25]. Several strategies consider geographic distance as a metric and try to forward the messages along the shortest geographic distance path [8], [22], [27]. Inspired by the research of social network analysis, the utilities such as social similarity and social centrality are also proposed to enhance data forwarding via social connections [18], [20], [21], [26], [36]. Most existing utility-based routing strategies employ a single or multiple utility metrics to compose their utility functions, such as RAPID [2], SimBet [4], [13], and CAR [27].

However, the utility-based routing strategies have two problems: blind spot and dead end. Consider the scenario that the source node $n_i$ wants to send a message to the destination node $n_d$. In the utility-based routing, $n_i$ needs to forward the message to a node with higher utility. If $n_i$ and all its neighbors have the similar utility near to zero, it will be difficult to decide which node should relay the message. Such problem is called blind spot for the reason that the next hop route is hard to be seen from the current node. The dead end problem occurs when $n_i$ has a higher utility value than all its neighbors, in which case the message is stuck in $n_i$ and not able to be delivered further.

The blind spot and dead end problems are rarely noticed in the previous works, but they commonly exist in utility-based strategies. Fig. 1 shows the proportion of blind spot and dead end when applying five utility-based routing strategies in three DTN datasets (MIT Reality [14], DieselNet [7] and Cabspotting [32]). According to the figure, for most utility-based strategies, there are more than 20% of data transmissions encountering the blind spot and dead end problems in the MIT reality trace. Similar percentages are observed in the DieselNet and Cabspotting traces, varying from 14% to 27%. Such problems will clearly affect the delivery ratio of DTN routing.

The community-based strategies, based on the fact that people tend to group into communities, forward data according
community structure. The nodes within a community have strong connections, while their links across communities are weak ties. The community structure favors intra-community communication where nodes are closely connected, but also encounters the difficulty of inter-community communication via weak links. Existing community-based routing strategies employ naive inter-community mechanism such as flooding [23], or rely on complicated operations to discover direct links [5] or overlapping nodes [18] between communities, which are time-consuming and inefficient. In summary, community-based routing strategies confront two challenges. On the one hand, the existing community partitioning algorithms are complicated and static, which is hard to adapt to DTN environments, thus a distributed community partitioning mechanism is desired. On the other hand, to overcome the blindness of data forwarding among communities via weak ties, it needs to measure the utilities across communities for a better routing decision making.

In this paper, we propose a novel utility-and-community-based routing strategy called Social and Mobile Aware Routing sTragey (SMART) for DTNs. SMART tackles the above problems to achieve the following objectives: (1) significantly alleviating blind spot and dead end problems; (2) distributed community partitioning; and (3) efficient inter-community communication. SMART first introduces a distributed community partitioning method based on the observation that movements of DTN nodes are regular and restricted in local areas where more encounters occur than that in remote areas. With distributed community partitioning, mobile nodes can flexibly adjust their community IDs to assign with the group they most frequently encounter, and the community structure is formed by exchanging only local information, which is easy to be implemented in DTNs. For intra-community communications, the routing utility is calculated by integrating the convolution of social similarity and social centrality with a decay function. It is shown that the transformation and decaying operation on the proposed utility function can reduce the occurrence of blind spot and dead end problems to a low level below 1%. For inter-community communications, nodes frequently traveling across communities are chosen as “fringe nodes”, and the utilities of communicating between fringe nodes and communities are measured for routing decision, which enhances the delivery ratio effectively.

The contribution of this paper is summarized as follows:

- We identify the blind spot and dead end problems in DTNs. Such problems commonly exist in utility-based routing strategies, but they are rarely studied in the past. In this paper, we introduce a transformation mechanism applying convolution and decay operation on utility computation, which can eliminate the cause of blind spot and dead end.
- We observe the locality property of mobile nodes: their movements are mostly restricted in a local area, and only a small proportion of nodes move remotely. Based on the movement locality, we carry out a distributed community partitioning method, which can adapt to the dynamic DTN environments.
- We propose a utility-and-community-based routing strategy called SMART. It divides a DTN into several communities, and exploits different principles for data routing depending on whether the source and the destination are in the same community. Utility functions integrating different routing metrics with convolution and decay function are explored to overcome the difficulties of intra- and inter-community communications.
- We conduct extensive experiments to evaluate the performance of SMART using three real DTN data sets, which show that SMART outperforms other strategies for DTN routing in various scenarios.

II. RELATED WORK

We categorize existing routing schemes for DTNs into two types: utility-based strategies and community-based strategies.

A. Utility-based strategies

Utility-based data routing strategies make forwarding decision according to a certain utility function. A node with higher utility value is preferred to be selected as the relay for data forwarding. The early algorithms use encounter-based utility like encounter frequency or inter-contact time for data routing. PROPHET [24] calculates the delivery predictability based on the assumption that node mobility follows a certain pattern. A node forwards data to its encounter only when this encounter’s delivery predictability is higher. In addition, MaxProp [6] makes forwarding decisions according to path likelihood, which is based on historical information of the nodes’ encounters. RAPID [2] considers DTN as a resource allocation problem. Its goal is to optimize the performance of DTN data forwarding from a specific perspective based on the node contacts in the network. These strategies extract the utility value by evaluating encounter frequency, total or average contact period or average inter-contact time. A group of utility-based strategies route data according to mobility-based utility. MobySpace [22] takes the distance of nodes’ often visiting locations as the utility to conduct data forwarding. It chooses the node with the closest distance as the relay. CAR [27] and MV routing [8] consider the probability of staying at the same location (colocation) as the metric to find the relay. Another group of strategies use utilities in social network context.
The typical strategies contains SDM [18], which selects the multicast relay by evaluating the node’s centrality. The data item is forwarded to the node with higher centrality. SimBet [13] combines social similarity and social centrality as the utility for relay selection. A recent proposed strategy [36], extracts social features of each node and chooses the node more similar with the destination as the relay. Besides, the study [10] considers data forwarding via local social map constructed by top encountering nodes. The data forwarding process needs to calculate the entire path from the source to the destination.

B. Community-based strategies

Community-based strategies make data forwarding decision according to the community structure of the network. By dividing the network into multiple communities, they use different routing strategies to handle the intra-community and inter-community data delivery due to the fact that the connections within a community are rich while the connections between different communities are weak. There are several routing strategies [5], [18], [20], [23] exploiting community structure for data routing in DTNs. Bubble Rap [20] considers the network as a pocket switched network, which consists of many different communities. It forwards data items from outside of the destination’s community according to a node’s global centrality. Within the destination’s community, the forwarding metric is determined by the node’s local centrality. MDM [18] constructs multicasting path by overlapping nodes of communities, and it chooses the node with destination awareness in the community for data delivery. LocalCom [23] uses the community structure to flood the data to dedicated nodes outside of the source community. Within the source community, it uses utility-based routing strategy for intra-community communication. A recent work taking the friendship community for information propagation is proposed in [5]. It considers the friendship community of varied periods of time. The data is forwarded only when the destination is in the same periodical community as the relay, which uses the temporal direct connection between communities to tackle the relay selection issue.

III. Mobile and Social Characteristics of DTN

In this section, we utilize three DTN data sets to explore people mobile and social characteristics. The mobility of DTN users reveals locality and their interactions reveal community structure.

A. DTN data sets

Our study is based on three public available DTN traces: MIT Reality [14], DieselNet [7] and Cabspotting [32]. The MIT Reality data set consists of the location traces of 97 users with Nokia 6600 smart phones at MIT during the 2004-2005 academic year. DieselNet logs mobility traces of 34 buses in Amherst. Each bus is equipped with a computer and a GPS. It records the GPS locations of all the buses during the 20 days from October to November in 2007. Cabspotting is a mobility trace of taxi cabs in San Francisco. Each taxi is outfitted with a GPS tracking device. It contains GPS coordinates of 536 taxis collected over 30 days in San Francisco Bay Area. The statistics of the three data sets are summarized in TABLE I.

<table>
<thead>
<tr>
<th>Traces</th>
<th>MIT Reality</th>
<th>DieselNet</th>
<th>Cabspotting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network type</td>
<td>Bluetooth</td>
<td>802.11b</td>
<td>none</td>
</tr>
<tr>
<td>No. devices</td>
<td>97</td>
<td>34</td>
<td>536</td>
</tr>
<tr>
<td>No. contacts</td>
<td>54,667</td>
<td>2,284</td>
<td>111,153</td>
</tr>
<tr>
<td>Duration (days)</td>
<td>246</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

We define the “encounter” between two mobile devices as the event that they move into each other’s communication range. We use a weighted social graph to formulate encounters among mobile devices. Each device is denoted by a node in the social graph. If two devices encounter, there is an edge between them, which builds a social link between two nodes. We consider two nodes with social link as friends. The number of friends that a node has is called degree of the node. The weight of an edge corresponds to the strength of social links, which can be represented the number of encounters between two nodes.

B. Mobility characteristics

People movements are not random and appear regularities restricting in some geographic areas. For example, people move regularly from home to office and vice versa; buses run along stations according to schedules; taxi drivers tend to pick up guests in some popular areas, etc. We observe the mobility trajectories of two taxis (#108 and #352) chosen from the Cabspotting trace, which are illustrated in Fig. 2. It can be seen that most of time taxi #108 moves in area A, while taxi #352 moves in area B. We call such kind of movement of which the trajectory is mostly restricted in a small area as locality.

To further investigate the locality of movements, we analyze the distribution of human mobility scope. We define geographic mass point as the centroid of a node’s trajectory, which is calculated by averaging the GPS coordinates of its trajectory. We compute the geographic mass points of all nodes in the network and study the geographic distance their trajectory coordinates away from their respective mass points.
Taking Cabspotting data trace as an example, Fig. 3 shows the probability distribution function (PDF) and cumulative distribution function (CDF) of the trajectory coordinates departing from their mass points in the trace. According to the figure, although the farthest coordinates is 30km away from the mass point, most of the movements are nearby their mass points, e.g., about 80% of trajectory coordinates within an area 5km from mass points. The majority of the coordinates concentrates in areas 2km, 3km, and 4km from mass points. This verifies the locality property of nodes movements: most movements of DTN nodes are restricted within a range of small distance, and there are only a small proportion of long distant movements.

![CDF and PDF](image)

**Fig. 3:** The CDF and PDF of Fig. 4: The number of encounters vs. distance.

**C. Distributed Community Partitioning**

Community is defined as a social unit that shares common value. It is a tight and cohesive social entity. Intuitively, communities are formed based on locations or interests [3], [16], [29], [30]. People in the same geographic location or sharing the same interest are likely to be in the same community. In the context of DTNs, we use geographic locations to study community structure and investigate the relation between encounters and geographic distances for the discovery of communities. It is likely that there is correlation between encountering and geographic location: the closer two nodes, the more often they meet each other. We calculate the number of encounters between node pairs as a function of the distance.

Assume there are $m$ communities to be detected:

$$E_i = \{ap_{1}, ap_{2}, \ldots, ap_{m}\},$$

where $ap_{j},$ is the AP that $n_i$ connects to community $C_j$ denoting the number of encounters between $n_i$ and $C_j$. When $n_i$ encounters a node in community $C_j$, it updates its AP value accordingly, and adaptively changes its community affiliation to the community with maximal AP value in the vector. We call our algorithm $m$-partition.

**Algorithm 1:** $m$-partition algorithm

```
input : node $N_i$ and AP vector
output: the community ID of node $N_i$
begin
Assume there are $m$ communities to be detected:
for Encounter with $N_j$ do
    if $N_i.communityID = null$ then
        $N_i.communityID = N_j.communityID$
    else
        $y \leftarrow N_j.communityID$
        $x \leftarrow N_i.communityID$
        if $y = x$ then
            $ap_{x} = ap_{x} + 1$
        else
            $ap_{y} = ap_{y} + 1$
        if $ap_{y} > ap_{x}$ then
            $N_i.communityID = y$
end
```

The algorithm runs dynamically as each encounter occurs in the network in a distributed fashion. Therefore, the community structure may change from time to time and is maintained dynamically. We show that the communication cost of $m$-partition for maintaining community members is low in DTNs. Given two communities $A$ and $B$, there are $m$ nodes in $A$ and $n$ nodes in $B$. Suppose a node $n_i$ in community $A$ needs to switch its community from $A$ to $B$, it first obtains the new community list from the encountered node in community $B$. If we consider the traffic overhead for transmitting one node ID as 1, the communication overhead for obtaining community members will be $n$. It then floods its ID to its new community $B$ to make other nodes in community $B$ be aware of $n_i$. The communication cost will also be $n$. Furthermore, when a node in community $A$ meets a node in community $B$, it checks the community member list to see whether any node changes their community identity. In this case, $n_i$ changes from $A$ to $B$. Suppose there are $k$ encounters between community $A$ and $B$. The cost for transmitting node ID of $n_i$ is $k$. The node in community $A$ floods this information to make the remaining nodes ($m-k$ nodes) in $A$ exclude the membership of $n_i$ from their local community, which needs $m-k$ transmissions of ID.
of \( n_i \). Overall, the communication overhead for maintaining community members caused by one community switch action is \( O(m + 2n) \).

D. Locality of user contacts

According to the above observation, it has high correlation between user movements and their encounters. Thus the locality of user movements is also reflected to their encounters. To investigate such locality, we define two types of encounters after community partitioning: when two nodes move into each other’s communication range, if both nodes are from the same community, we call such encounter a local contact; if the encountering nodes are from different communities, it is called a remote contact. Since a node tends to move in a local area, which results in frequent local contacts. In contrast, only a small proportion of user movements are long-distance, which yields remote contacts forming cross-community communications.

We calculate the proportion of local contacts and remote contacts of the three DTNs after applying our community partitioning algorithm (with \( m = 10 \)). The results are shown in Table II. It presents that local contacts are majority for all three traces and remote contacts are minor. MIT Reality has 19.7% remote contacts; DieselNet has 7.3%; and Cabspotting has 9.9%, which confirms the locality of user contacts. Although remote contacts only take a small fraction, they play an important role for information exchange since they are the “bridges” between communities. The more remote contacts, the more active a community interact with others. It enlightens the idea of using local contacts for intra-community communication and remote contacts for inter-community communication. The detailed approach is presented in the subsequent section.

<table>
<thead>
<tr>
<th>TABLE II: Proportion of local and remote contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traces</strong></td>
</tr>
<tr>
<td>Local contact (%)</td>
</tr>
<tr>
<td>Remote contact (%)</td>
</tr>
</tbody>
</table>

IV. SMART: THE SOCIAL AND MOBILE AWARE ROUTING STRATEGY

In this section, we introduce the social and mobile aware routing strategy (SMART) for DTNs. The basic idea of SMART is to deliver the message in the same community via local contacts, and forward the message to other communities via remote contacts.

A. Preliminaries

Before presenting the detailed design of SMART, we introduce some basic concepts and assumptions.

- Assume a dynamic community partitioning process (i.e. \( m \)-partition) is applied to cluster the social graph into a number of communities \( \mathcal{C} = \{C_1, C_2, \ldots, C_M\} \).
- Each node \( n_i \) is assigned with a community ID \( C_i \) and a set \( \pi(C_i) = \{n_j | \forall n_j \text{ in } C_i\} \) indicating the members in the same community. The community ID and community members are obtained and maintained from the result of community partitioning process.
- Each node records the encounter history for a period of time \( \Delta T \), where \( \Delta T \) is a time window representing one day interval. Each node maintains the history of local contacts that occur within the local community and remote contacts that occur across different communities.

<table>
<thead>
<tr>
<th>TABLE III: Local contact table</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_1 )</td>
</tr>
<tr>
<td>( \zeta_{11} )</td>
</tr>
</tbody>
</table>

- Local contacts are recorded into a local contact table, whose format is shown in Table III. It records the contact frequency of a node with other nodes in the same community. In the table, \( n_j \) (\( j = 1, 2, \ldots, N \)) is the ID of a node in the local community and \( \zeta_{ij} \) (\( j = 1, 2, \ldots, N \)) is the contact frequency between \( n_i \) and \( n_j \) in \( \Delta T \). If \( n_j = n_i \), let \( \zeta_{ij} = -1 \) indicating that the contact with itself is not countable. The contact frequency is the number of encounters over the time period \( \Delta T \). It is calculated by:

\[
\zeta_{ij} = \frac{\sum_{t=0}^{\Delta T} X(t)_{ij}}{\Delta T},
\]

where \( X(t)_{ij} = 1 \) if two nodes contact at time \( t \), otherwise, \( X(t)_{ij} = 0 \).

<table>
<thead>
<tr>
<th>TABLE IV: Remote contact table</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
</tr>
<tr>
<td>( \eta_{11} )</td>
</tr>
</tbody>
</table>

- Remote contacts are recorded into a remote contact table, whose format is shown in Table IV. It records the contact frequency of a node with other communities. In the table, \( C_j \) (\( j = 1, 2, \ldots, M \)) is the ID of a community (there are \( M \) communities in the network) and \( \eta_{ij} \) (\( j = 1, 2, \ldots, M \)) is the sum of encounters that \( n_i \) with nodes in \( C_j \) over \( \Delta T \). Again, \( \eta_{ij} = -1 \) when \( C_j = C_i \) suggesting that the local contacts is not reflected from the remote contact table.

When a source node \( n_s \) sends a message to the destination \( n_d \), \( n_s \) checks the local community members to see whether \( n_d \) is in the same community. If \( n_s \) and \( n_d \) are in the same community, we apply an intra-community communication process. If \( n_s \) and \( n_d \) are in different communities, we apply an inter-community communication process. The two processes are described as follows.

B. Intra-Community Communication

If a source node \( n_s \) and a destination node \( n_d \) are in the same community, it is possible to apply traditional utility-based strategies for data forwarding. However, to avoid the problems of blind spot and dead end as mentioned before, the intra-community routing scheme needs to be carefully designed.
To against blind spot and dead end, we need a social featured routing metric that accumulates encounter effects and decays according to the node’s social status and time elapsed. We consider each encounter has an effect to the utility value, which is positively correlated to the social relation (i.e., social similarity) between two nodes. That is, two nodes with closer social relation leads to higher utility increase in each encounter. Besides, the encounter effect decays depending on its social status and time elapsed. An earlier effect will have less effect remaining due to temporal factor. Meanwhile, a node with high social status will motivate further encounters. To represent this motivation, a node with higher social status in the network should have a slower decaying speed on the encounter effect. Combining temporal and social factor, each encounter effect decays as the social status of a node and the time elapsed. To select a relay, the scheme will evaluate the accumulative effects produced by the encounters and the decaying speed of the effects. We provide the formulation of our scheme as follows.

As the first step, we give the definition of social relation and social status. The social relation denotes the social closeness between two nodes and social status shows the relative importance of nodes in the social network. Although the social relation and social status can be represented in many sophisticated manners [1], [13], we choose two representative expression to illustrate our scheme. Namely, we use social similarity to represent the social relation, and social centrality to represent social status.

**Social similarity:** it is defined as the number of common friends between a pair of nodes, indicating the trustiness and cohesiveness of social ties between them [11], [12]. Formally, social similarity can be calculated by the following equation.

\[
S_{i,j}(\tau) = |F_i(\tau) \cap F_j(\tau)| + 1, \tag{1}
\]

where \(F_i(\tau) \ (F_j(\tau)) \) is the set of friends of node \(n_i \ (n_j)\) at time \(\tau\). The intersection operation is to obtain the common friends between two nodes and plus 1 is to eliminate the effect of 0. When two nodes encounter, they exchange their friend lists to calculate the social similarity. The social similarity between \(n_i\) and the destination \(n_d\) is \(S_{i,d}(\tau)\). Intuitively, if a node has higher \(S_{i,d}(\tau)\) value, it shares more common friends with the destination, thus more likely to meet destination.

**Social centrality:** it is a quantification of the relative importance of nodes in the social network. There are various definitions of centrality. We use the Freeman’s degree [17] to define social centrality in this paper. For a node \(n_i\), its centrality is defined as follows.

\[
C_i(\tau) = \frac{\sum_{k=1}^{N} d_{ik}(\tau)}{N}, \tag{2}
\]

where \(d_{ik}(\tau) = 1\) if a direct link exists between \(n_i\) and \(n_k\) at time \(\tau\) and \(N\) is the number of nodes in the community. Structurally, a central node has stronger connection with other nodes, and it is suitable to serve as a hub for information exchange.

Both social similarity and social centrality only require local information, and they can be calculated locally in DTNs by exchanging information with neighbors.

The encounter effect between two nodes is therefore denoted by the social similarity. To model the decaying effect, we introduce a decay function with respect to social centrality and time as follows.

\[
D_i(t - \tau) = \frac{C_i(\tau)}{t - \tau}, \tag{3}
\]

where \(\tau\) is the time when the encounter occurs. The decay function relying on both social centrality and elapsed time to reduce the accumulative effects of the utility value.

If an encounter occurs in each time unit, the accumulative effects of encounters between two nodes with decay between node \(n_i\) to destination \(n_d\) can be formulated as the convolution of Eq. 1 and Eq. 3.

\[
Y_{i,d}(T) = S_{i,d}(T) \otimes D_i(T),
\]

\[
= \int_{\tau=0}^{T} S_{i,d}(\tau) \ast D_i(T - \tau). \tag{4}
\]

However, the encounter only occurs in several time units. Therefore, the accumulative effects of encounters are represented by a discrete convolution as:

\[
U_{i,d}(T) = \sum_{\tau=0}^{T} X(\tau)_{id} \ast S_{i,d}(\tau) \ast D_i(T - \tau), \tag{5}
\]

where \(X(\tau)_{id} = 1\) when an encounter occurs at time \(\tau\) or when \(\tau = 0\) (to initialize the utility value), otherwise, \(X(\tau)_{id} = 0\). The utility function describes that when each encounter occurs, it yields an encounter effect represented by social similarity. Each effect occurs at different time decays as a decay function composed by social centrality and time, indicating the encounter effects of a node with higher social status decays slower than a node with poor connection to the network and a recent encounter effect decays slower than a older encounter effect.

Based on the above utility function, we propose our intra-community routing principle.

**Intra-community routing principle:** when two nodes \(n_i\) and \(n_j\) in the same community encounter, the routing utility in Eq. 5 is calculated and the node with higher value will be chosen to forward the message.

**C. Inter-Community Communication**

If a destination node \(n_d\) does not belong to the community of source node \(n_s\), we need to choose some relay nodes to forward the message among communities. The idea is using “fringe nodes” to bridge the communication of inter-communities.

A fringe node is a node which is capable to remote contact with other communities. It is measured by the number of links that it connects to other communities. We select nodes with higher links to outside of local community as fringe nodes. Each fringe node is represented by its ID and the remote contact table as mentioned in Table IV to indicate its links to other communities. In our proposed scheme, each community maintains a set of fringe nodes \(\mathcal{F}\). The set \(\mathcal{F}\) is randomly selected initially, and is updated periodically. During
a period $\Delta T$, each node compares its remote contact table with the fringe nodes. If a node $n_i$ finds that it has closer connection with outside communities than a fringe node $n_j$, it will announce itself as the new fringe node. The comparison is described as follows.

Assume $\eta_{i1}, \eta_{i2}, \ldots, \eta_{iM}$ is the remote contact frequency of $n_i$, and $\eta_{j1}, \eta_{j2}, \ldots, \eta_{jM}$ is the remote contact frequency of $n_j$. Define a function $\phi(x, y) = 1$ if $x \geq y$ and $\phi(x, y) = -1$ for the rest. The selection of fringe node is determined by the value

$$\tilde{S}_c = \sum_{k=1}^{M} \phi(\eta_{ik}, \eta_{jk}).$$

If $\tilde{S}_c$ is larger than 1, it means $n_i$ has better connection than $n_j$, and $n_i$ becomes the new fringe node and announces to the other nodes in the local community.

According to the report in [35], a small fraction rewired links are enough to create a small world network. Our analysis to the three traces shows that the fraction of remote contacts varies from 7.3% to 19.7% (as shown in Table II), so we set the number of fringe nodes as 10% of the community size. If the community size is smaller than 10, we set the number of fringe nodes as 1.

Due to dense network connection in the community, the set of fringe nodes $F$ and their remote contact tables can easily spread to nodes in the same community. Since there are more than one fringe nodes which can reach to other communities, it needs to carefully choose the forwarding node for performance consideration.

Assume the source node $n_s$ in community $C$ wants to send a message to the destination $n_d$ in community $C'$ ($C \neq C'$). We first decide whether $C$ and $C'$ are directly connected by looking up the set $F$ in $C$ to check whether there exists a fringe node connecting to $C'$. If there is a connection from fringe node set to the destination community, we say $C$ and $C'$ are directly connected. Otherwise, we say they are indirectly connected.

If $C$ and $C'$ are directly connected, we need to forward the message to a fringe node who can reach $C'$. There might be more than one fringe nodes directly connecting to $C'$, and the candidate set is $C = \{n_j | \forall n_j \in F \text{ and } (n_j \text{ connects to } C')\}$. We need to decide which node in the candidate set as a relay. Our principle is to send the message to a relay having more connections with $C'$. By looking up the remote contact tables of fringe nodes, the algorithm chooses the one with the maximal number of encounters to $C'$ as the relay. If $C$ and $C'$ are indirectly connected, we select the fringe node with maximal number of encounters with outside of $C$ by summing up entries in remote contact tables. The source node forwards the message to the selected fringe node by intra-community routing strategy. After the data is forwarded to the dedicated fringe node, the data transmission between communities becomes a challenge.

To enable efficient inter-community communication, we propose a utility function extended from intra-community utility to forward data from the fringe node to the destination community. Namely, we extend the utility function from node-to-node to node-to-community for inter-community communication. We build the utility function between a fringe node $f$ to the destination community $C'$. To construct such utility, we consider social relation between $f$ and $C'$ as similarity between the node $f$ and a set of nodes in $C'$. However, knowing the friends of all nodes in $C'$ would suffer too much overhead in DTNs. In this case, we provide an estimation that we only count the friends of nodes who have ever encountered with $f$. The similarity is defined as:

$$S_{f,C'}(\tau) = |F_f(\tau) \bigcap F_{C'}(\tau)| + 1,$$

where $F_{C'}(\tau)$ indicates the friends of a set of nodes in $C'$ that ever encountered with $f$ until time $\tau$.

To formulate the social status of node $f$, we extend the concept of centrality from the local community to the entire network. We call it community centrality, denoted by $C_t(\tau)$, which is defined as the proportion of the number of communities it connecting with ($M(f, \tau)$) to the total number of communities ($M(\tau)$) at time $\tau$. It is defined as:

$$C_t(\tau) = \frac{M(f, \tau)}{M(\tau)}.$$

The decay function in the node-to-community utility becomes

$$D_t(t - \tau) = \frac{C_t(\tau)}{T - \tau}.$$

The overall utility function from the node $f$ to community $C'$ thus is defined as:

$$U_{f,C'}(T) = \sum_{\tau=0}^{T} X(\tau)_{f,C'} \ast S_{f,C'}(\tau) \ast D_t(T - \tau),$$

where $X(\tau)_{f,C'} = 1$ when an encounter occurs between node $f$ and community $C'$ at time $\tau$ or when $\tau = 0$ (to initialize $U_{f,C'}(T)$). According the utility function, the fringe node finds the next relay by choosing a node with higher utility value with destination community. The procedure continues until the data reaches destination community.

V. DISCUSSION

In this section, we discuss how SMART tackles blind spot and dead end problems and achieves efficient inter-community communication.

A. Tackling blind spot and dead end problems

The blind spot and dead end problems result from the indecisive utility value of utility-based data routing strategies. The blind spot occurs when utility values of a node and its neighbors are close and nearly to 0. It cannot decide which node should be the next relay. The dead end arises with all neighbors of a node having lower utility value than it. The node cannot conduct the forwarding behavior in the network.

To settle blind spot problem, SMART treats each encounter between different pair of nodes differently. When a node and its neighbors have similar contact rates with the destination node, a node having closer social relation (e.g., larger social similarity) with the destination will have larger encounter effects. Even two nodes having no encounter with the destination, the social centrality controlled decay function can still be used to determine the relay. It hence significantly reduces the occurrence of blind spot. To overcome the dead end, SMART
introduces a decay function combining social centrality and elapsed time to pull down the utility value by letting well connected nodes decay slower but poorly connected nodes decay faster. Specifically, the utility value computed by the decay function decreases with time and social centrality. When there is no encounters for a long time, the utility value of a node will drop to a smaller value. Also, a node with a lower social centrality will decay faster than that with a higher social centrality. Such mechanism can avoid the message stuck in a node for a long time. For instance, assume a piece of data destined to node $D$ is stuck into a dead end node $A$ because $A$ has the highest local utility. With decay function introduced in SMART, the dead end can be avoided if a node has recent encounters with $D$ or higher social centrality value than $A$, which makes the decay speed of its utility slower than that of $A$. Thus the data cannot be stuck into node $A$ forever and it will be forwarded to a new relay with higher utility value eventually, which reduces the occurrence of dead end.

We conduct experiments to show the percentage of blind spot and dead end occurring in SMART. By definition, we consider a node encounters blind spot if it has similar utility value closing to 0 as its neighbors and it cannot find the next relay until the data is expired. A node occurs dead end if its utility value is larger than all its neighbors and the data is stuck into the node until it is expired. We look for the data routing failures caused by blind spot and dead end in three data sets (MIT Reality, DieselNet and Cabspotting). We sum the two types of failures and draw the curve as a function of time as shown in Fig. 5. According to the figure, there is only a tiny percentage of blind spot and dead end appearing in SMART, most of the time lower than 1%. Compared to the experiment results shown in Fig. 1, the percentage drops dramatically. Thus we claim that SMART significantly alleviates the blind spot and dead end problems.

**B. Efficiency of inter-community communication**

The inter-community communication is a difficult task in community-based strategies since it lacks of strong links between different communities. Existing community-based strategies use naive routing mechanism such as flooding [23], or rely on discovering the direct links [5] or overlapping nodes [18] between communities, which are time-consuming and inefficient.

SMART enhances the capability of inter-community communication by selecting fringe node and a node-to-community utility function. The fringe node is selected by the criterion that has rich connection to the remaining network. To bridge the communication between the fringe node and the destination community, SMART introduces a node-to-community utility function, which considers the destination community as an entity. Analogous to intra-community utility, we compose utility function between fringe node and the destination community, and build routing channels among different communities.

We compare the performance of SMART with other community-based strategies, including Bubble Rap [20] and Friendship Based Routing (FBR) [5] to show its efficiency. The comparison of delivery ratio on three data traces (MIT Reality, DieselNet and Cabspotting) is shown in Fig. 6. It is illustrated that for inter-community communication, the delivery ratio of Bubble Rap and FBR is 32% and 33% respectively, while the delivery ratio is improved in SMART greatly, which achieves 47%. For delivery ratio of intra-communication, SMART achieves 85%, which also outperforms the other strategies (with 72% in Bubble Rap and 75% in FBR) on MIT Reality. The performance comparison on DieselNet and Cabspotting also shows the efficient inter-community communication of SMART. Together with our distributed community partitioning algorithm, the proposed SMART strategy shows its effectiveness in dealing with the community-based routing problems.
Fig. 7 shows the performance metrics as a function of community number and time on MIT Reality trace. The delivery ratio of MIT Reality trace is shown in Fig. 7a. According to this figure, when no community partitioning algorithm is applied \((m = 1)\), the delivery ratio is quite low and it increases slowly with time. As the community number is set to an appropriate value (e.g. \(m = 10\)), the delivery ratio increases dramatically, which is almost 2 times as much as that when \(m = 1\). For \(10 \leq m \leq 90\), the delivery ratio becomes stable and has only small fluctuation. When the community number approaches to the size of the data set \((m = 97)\), the performance drops dramatically since the impact of community structure disappears. The average delay is illustrated in Fig. 7b. It is seen that the average delay is almost the same for all community numbers and it only varies with time. The average cost is shown in 7c. Similar to delivery ratio, the average cost is influenced by \(m\) and it increases to a stable value when \(10 \leq m \leq 90\). Similar results are also found in DieselNet and Cabspotting. Due to the limitation of space, we do not show the detailed results for them in this paper. The results suggest that SMART performs better when the community structure is outlined, while the performance of SMART is low when no community structure is indicated in the network. It also reveals that the proper value of \(m\) is within a wide range. In the rest of our experiments, we fix our community number to \(m = 10\).

We show the impact of community partitioning algorithms to the SMART routing scheme in this group of experiments. We evaluate the performance of SMART using different community partitioning algorithms, including m-partition, k-clique percolation algorithm [31] (which considers the adjacent k-clique as communities), and Girvan Newman algorithm [28] (which continues removing edges with the highest betweenness until a certain threshold is reached).

Fig. 8 presents experimental results of MIT Reality trace. As shown in Fig. 8a, the m-partition method outperforms Girvan Newman by 10% and k-clique percolation by 2% in delivery ratio. In terms of average delay, as shown in Fig. 8b, m-partition performs slightly better than the other two algorithms. The three algorithms takes similar average cost as shown in Fig. 8c. Similar results are also observed on DieselNet and cabspotting data traces.

Despite the different algorithms used for community partitioning, the routing performance is quite similar for all three DTN data sets. It indicates that the proposed SMART routing mechanism is not sensitive to community partitioning algorithms. Since Girvan Newman and k-clique percolation need global network topology information which is difficult to obtain in DTNs, the proposed m-partition algorithm is more suitable for distributed implementation in the real world.
D. Performance comparison

We compare SMART with five existing DTN routing strategies: PROPHET [24], SimBet [13], Bubble Rap [20], Friendship Based Routing (FBR) [5], and Epidemic routing [34]. PROPHET is a utility-based strategy according to encounter histories. It forwards data to the nodes with higher delivery rate based on contact history. SimBet is a utility-based strategy according to social features. It considers social properties including similarity and centrality to make data forwarding decision. Bubble Rap is a community-based strategy. It depends on community structure and routes data based on rankings calculated from social centrality. FBR algorithm as another community-based algorithm, it constructs temporal community and use the nodes with direct connection to the destination community for data delivery. Epidemic routing is a flooding strategy. It has high delivery cost, but its delivery ratio and delay approach the theoretical bound. To show the amount of traffic that used for construct routing algorithms, we add control overhead as an additional metric to evaluate different DTN routing strategies.

Fig. 9 shows the performance of various algorithms as a function of time on MIT Reality trace. The delivery ratio is compared in Fig. 9a. It shows that SMART outperforms PROPHET, SimBet, FBR and Bubble Rap. The delivery ratio of SMART is about 10% higher compared to Bubble Rap and FBR, 15% higher than that of SimBet and nearly 20% higher than that of PROPHET. The results also confirm that SMART outperforms utility-based strategies nearly 20% by solving blind spot and dead end problems. The reason PROPHET performs the worst is due to the strong community structure of MIT Reality trace. When source and destination are interconnected by a long path, PROPHET will encounter the blind spot and dead end problems, which degrade its performance.
SimBet exploits social properties to enhance the delivery ratio but it also encounters high proportion of blind spot and dead end problems. Bubble Rap and FBR takes advantages of community structure, so they perform better than PROPHET, but not as well as SMART. Since Epidemic routing represents the theoretical upper bound of delivery ratio, the performance of SMART is below Epidemic routing. Average delay is compared in Fig. 9b. Again, the delay of SMART is lower than the other four strategies (most of the time their performance are very close), but higher than the lower bound (Epidemic routing). Average cost is compared in Fig. 9c. The cost of PROPHET is the highest. The cost of SMART is slightly higher than others due to the decaying effect, which makes SMART take more relays for data delivery. The comparison of control overhead on MIT Reality data trace is shown in Fig. 9d. The overhead of SMART is around 20 KB in the end of the experimental period. PROPHET has over 10% higher overhead than SMART since it needs to exchange transitivity information. FBR takes 4 times of control overhead more than SMART because it requires the encounter information from neighbors.

Fig. 10 presents the performance results of various algorithms as a function of time on DieselNet data set. The delivery ratio is depicted in Fig. 10a. SMART outperforms Bubble Rap by 3%, FBR by 5% and PROPHET by 8%. It has nearly 20% higher of delivery ratio than SimBet. Regarding the average delay and the average cost of each strategy as shown in Fig. 10b and Fig. 10c, SMART has very close average delay with Epidemic, which is less than other strategies. The average cost of SMART is about 50% of that of PROPHET and higher than FBR and SimBet. DieselNet has very similar network structure with MIT Reality and thus has similar trend on delivery ratio with MIT Reality. However, due to the regular and repetition routine of buses in DieselNet, it makes the SimBet meet dead ends quite often and takes more time to wait until destinations. Therefore, it has lower delivery ratio and higher average cost. Since DieselNet has more tight clustering structure, it makes Bubble Rap and FBR perform close to SMART. SMART has similar cost with social-related strategies but much lower cost than PROPHET. The overhead is shown in Fig. 10d. The overhead of SMART is the lowest, which is 4KB in the end of the experimental period. PROPHET has 25% higher overhead than SMART and the overhead FBR is 2.5 times much as that of SMART. Comparison of different algorithms’ performance on Cab-spotting trace is shown in Fig. 11. Fig. 11a depicts the delivery ratio of varied algorithms as a function of time. The SMART has very similar performance as PROPHET. It outperforms FBR by 5%. Bubble Rap algorithm is impacted by weak community structure, which lowers down its delivery ratio around 10% compared to SMART. SimBet has the lowest delivery ratio, which is much lower than other strategies. In terms of average delay as shown in Fig. 11b, SMART costs as low as Epidemic algorithm delay, which is much lower than others. The average costs of various algorithms are similar as shown in Fig. 11c. The overhead is shown in Fig. 11d. The overhead of SMART is around 20 KB in the end of the experimental period, while the overhead of FBR and PROPHET is much higher than that of SMART.

In a summary, the proposed SMART strategy outperforms the utility-based and community-based strategies on various DTN data sets in most of the performance metrics.

VII. CONCLUSIONS

Data routing in DTNs is challenging due to the fact that nodes are constantly moving and the opportunity of communication between node pairs is intermittent. Existing routing strategies encounter the problems of blind spot and dead end, and also lack of efficient implementation in DTNs. In this paper, we first investigate the characteristics of DTNs by analyzing three data sets. We reveal the social and mobile features of DTNs: they have community structure and their movement shows locality. Based on these features, we propose the social and mobile aware routing strategy called SMART. In this strategy, a DTN is divided into a number of communities using an adaptive community partitioning algorithms. Two data routing processes are introduced: intra-community communication and inter-community communication. For intra-community communication, a utility function convoluting social similarity and social centrality with a decay factor is used to choose relay nodes. For inter-community communication, the nodes moving frequently across communities are chosen as relays to carry the data to destination efficiently. It is shown that SMART significantly alleviates the blind spot and dead end problems. It adapts to the community structure by enhancing performance for inter-community communication. We conduct extensive experiments to compare the performance of SMART with other DTN routing strategies. It presents that the proposed routing strategy works well in various DTN traces.

VIII. ACKNOWLEDGEMENTS

This work was partially supported by the the National Natural Science Foundation of China (Grant Nos. 61373128, 61321491), EU FP7 IRSES MobileCloud project (Grant No. 612212), Alexander von Humboldt Foundation and German Academic Exchange Service (DAAD).

REFERENCES


Konglin Zhu received his B.S. degree in Computer Science in 2008 from both North China University of Technology, Beijing, China, and Southern Polytechnic State University, Marietta, United States. He received his M.S. degree in Computer Science from University of California at Los Angeles, United States, in 2009. He is currently a Ph.D student at the Institute of Computer Science of University of Göttingen, Germany. His research interests include design of routing protocols for wireless networks, such as Delay Tolerant Networks, and data analysis in online social networks.

Wenzhong Li received his B.S. and Ph.D degree from Nanjing University, China, both in computer science. He is now an associate professor in the Department of Computer Science, Nanjing University. Dr. Li’s research interests include wireless networks, pervasive computing, and social networks. He has published over 30 research papers at international conferences and journals, which include ICDCS, IWQoS, ICPP, ICC, WCNC, IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, etc. Dr. Li was also the winner of the Best Paper Award of ICC 2009. He is a member of IEEE and ACM.

Xiaoming Fu received his bachelor and master degrees from Northeastern University, China, and his Ph.D. degree from Tsinghua University, Beijing, China. He was a research staff at the Technical University Berlin until joining the University of Göttingen, Germany in 2002, where he has been a professor in computer science and heading the Computer Networks Group since 2007. His research interests include network architectures, protocols, and applications. He is currently an editorial board member of IEEE Communications Magazine, IEEE Transactions on Network and Service Management, Elsevier Computer Networks, and Computer Communications, and has published over 100 papers in journals and international conference proceedings. He is the coordinator of EU FP7 GreenICN and MobileCloud projects, and the recipient of the IEEE LANMAN 2013 Best Paper Award and the 2005 University of Göttingen Foundation Award for Exceptional Publications by Young Scholars.