# Improving Data Forwarding in Mobile Social Networks with Infrastructure Support: A Space-Crossing Community Approach

Zhong Li*, Cheng Wang*†, Siqian Yang*, Changjun Jiang*, Ivan Stojmenovic†‡
* Department of Computer Science, Tongji University, Shanghai, China
† SEECS, University of Ottawa, Canada
‡ School of Software, TNList, Tsinghua University, Beijing 100084, China

*Abstract*—**In this paper, we study two tightly coupled issues: space-crossing community detection and its influence on data forwarding in Mobile Social Networks (MSNs) by taking the hybrid underlying networks with infrastructure support into consideration. The hybrid underlying network is composed of large numbers of mobile users and a small portion of Access Points (APs). Because APs can facilitate the communication among long-distance nodes, the concept of physical proximity community can be extended to be one across the geographical space. In this work, we first investigate a space-crossing community detection method for MSNs. Based on the detection results, we design a novel data forwarding algorithm SAAS (Social Attraction and AP Spreading), and show how to exploit the space-crossing communities to improve the data forwarding efficiency. We evaluate our SAAS algorithm on real-life data from MIT Reality Mining and University of Illinois Movement (UIM). Results show that space-crossing community plays a positive role in data forwarding in MSNs in terms of delivery ratio and delay. Based on this new type of community, SAAS achieves a better performance than existing social community-based data forwarding algorithms in practice, including Bubble Rap and Nguyen's Routing algorithms.**

## I. Introduction

As the development of social networks, more and more Bluetooth/WiFi-based mobile social applications emerge, for example, MobiClique [1], Foursquare [2], E-SmallTalker [3], Sony's Vita [4]. These applications should support real-time communication. Traditional method of deploying large numbers of base stations are costly and the workload of the base stations is heavy. Subsequently, as an alternative to the centralized organization, self-organized ad hoc underlying networks were developed. In ad hoc networks, there are no stable end-to-end delivery paths to guarantee a high efficient data forwarding. This problem was studied extensively in the literature. Recently, based on ad hoc underlying networks, solutions using social community detection results are proposed to design a data forwarding schemes in Mobile Social Networks (MSNs), such as [5]–[9].

However, ad hoc networks mainly act as an extended form for centralized way rather than an individual network architecture in typical real-life applications. In reality, the underlying networks are usually hybrid with infrastructure support. In some areas, due to the lack of internet Access Points (APs) or very weak signals from cellular base stations, users need to use a self-organized way, while in other areas, users can have services via APs. In this paper, we consider the hybrid underlying network with APs support and study how to utilize APs to design a high efficient data forwarding scheme.

We consider a mobile social network in which several APs are deployed. The coverage area of APs are assumed to be a small portion of the entire network, and therefore many nodes do not have direct access to an AP. Traditionally, a community is defined as a group of tight-knit nodes with more internal than external links [10]–[12], such as the common interest community with logical links, or the physical proximity community with geographic aggregation. In hybrid underlying network, APs facilitate the communication among some long-distance nodes. Unlike the traditional concept of a community, we propose a novel concept of *space-crossing community* as a group of nodes with relatively stronger communication capability. Space-crossing communities are formed through spanning physical space to merge physical proximity communities via access points. The common concept of *activity* is defined with respect to the whole network. We define *local activity* of a node with respect to its associated community. The local activity of a node $u$ in its associated space-crossing community $ComSC$ is the ratio of $u$'s encounter probability with other nodes in $ComSC$ to the sum of encounter probability between any two nodes in $ComSC$. Therefore the local activity of a node reflects a statistics of relative encounter probability of a node compared to overall encounter probabilities of all pairs in certain space-crossing community.

To the best of our knowledge, this is the first paper that studies the space-crossing community detection and its influence on data forwarding in MSNs by taking the hybrid underlying network with APs support into consideration. The main contributions of this paper can be summarized as follows:

• A space-crossing community detection method for MSNs is given, including forming physical proximity communities and using APs to obtain final space-crossing communities at each time slot. Static and dynamic combination criteria $\mathcal{S}^a$,

978-1-4799-3360-0/14/$31.00 ©2014 IEEE

$\mathcal{S}^b$ are applied throughout the entire detection method.

• A SAAS (Similarity Attraction and AP Spreading) data forwarding algorithm is proposed, which takes full advantage of the space-crossing communities. SAAS consists of two phases. In the first phase, if a node is in physical proximity communities without APs (or a solitary node), we use social similarity to guide data forwarding. Social similarity is defined by combining node local activity with Pearson correlation coefficient. In the second phase, if a node is in the physical proximity community containing AP, message copies are spread to the proximity nodes and more nodes carry messages to the destination.

• We extensively evaluate SAAS on two hybrid underlying networks: MIT Reality Mining [13] and UIM (University of Illinois Movement) [14] datasets. The results show that SAAS significantly outperforms several existing social community-based algorithms.

The rest of the paper is organized as follows. Section II presents the network model. Section III and Section IV study the space-crossing community detection and its impact on data forwarding in MSNs. We conduct extensive experiments and report our results in Section V. We review related work in Section VI and conclude the paper in Section VII.

## II. Network Model

### A. Dynamic Graph

Mobile social network consists of mobile users and stationary Access Points (APs). We model this *hybrid underlying network with APs support* as a *dynamic graph* which can be defined as a time sequence of network graph, denoted by $\mathcal{G} = \{G_0, G_1, ..., G_t, ...\}$, where $G_t = (V_t, E_t)$ represents a time dependent network snapshot recorded at time $t$; $V_t$ denotes the set of nodes, including the set of mobile users and the set of stationary APs; $E_t = \{(u,v)|u,v \in V_t\}$ denotes the edge set. Both node and edge sets change over time.

### B. Assumption of AP

• An AP can be used as a relay or as a centralized server.

• In our mobile social network, the ratio of the number of APs to the number of mobile users is small. Different from a base station, the coverage area and processing capability of an AP is limited. To be specific, the radius of coverage area is usually about 30-100m.

• As the first work on hybrid underlying networks with APs support, this paper makes a simple assumption that APs are connected along a ring, in order to highlight the main novelty, the impact of infrastructure support on data forwarding in MSNs.

### C. Contact Aggregation for Edges

We treat APs in the same way as the ordinary mobile users. The edge set $E_t$ is formed according to the following methods.

Based on trace analysis of the real-life social datasets which usually contain the contact records of Bluetooth and wifi access points, we add the number of direct contacts between *user pairs* or *user-AP pairs* $u$ and $v$ iteratively in a chosen time period $t_0$ to $t_q$. Denote the number of contacts between node $u$ and $v$ at time $t_i$ as $l_{uv}^{t_i}$; the number of contacts among all nodes in the network at time $t_i$ as $l_*^{t_i}$; the encouter ratio between node $u$ and $v$ at time $t_i$ as $w_{uv}^{t_i}$.

• If the contact traces are sparse, we will implement a weighted growing window mechanism. Let $\sum_{i=0}^{p} l_{uv}^{t_i}$ denote the overall numbers of contacts between node $u$ and $v$ in time period $t_0$ to $t_p$; let $\sum_{i=0}^{p} l_*^{t_i}$ denote the overall numbers of contacts for all users. Thus, we have an encounter ratio value $w_{uv}^{t_p} = \frac{\sum_{i=0}^{p} l_{uv}^{t_i}}{\sum_{i=0}^{p} l_*^{t_i}}$ between node $u$ and $v$ at time $t_p$, where $0 \le p \le q$. An edge between nodes $u$ and $v$ will be created if $w_{uv}^{t_p}$ is larger than the median of $\{w_{u'v'}^{t_p} | u', v' \in V_t$ and $w_{u'v'}^{t_p} \neq 0\}$.

• If the contact traces are dense, we will carry out a weighted sliding window mechanism. The time granularity of the window length $\Delta$ is empirically determined according to different datasets[1]. Let $\sum_{i=p-\Delta}^{p} l_{uv}^{t_i}$ denote the overall numbers of contacts between node $u$ and $v$ during $\Delta$ time window, from the current time $t_p$; let $\sum_{i=p-\Delta}^{p} l_*^{t_i}$ denote the overall numbers of contacts for all users. Thus, we have an encounter ratio $w_{uv}^{t_p} = \frac{\sum_{i=p-\Delta}^{p} l_{ij}^{t_i}}{\sum_{i=p-\Delta}^{p} l_*^{t_i}}$ for nodes $u$ and $v$ at time $t_p$, where $\Delta \le p \le q$. Again, an edge between nodes $u$ and $v$ will be created if $w_{uv}^{t_p}$ is larger than the median of $\{w_{u'v'}^{t_p} | u', v' \in V_t$ and $w_{u'v'}^{t_p} \neq 0\}$.

*Remark 1*: To cope with the invalidation of Bluetooth devices in those traces, we have $w_{uv}^{t_p} = w_{vu}^{t_p}$, by assigning the larger value to the other.

*Remark 2*: Our *weighted* aggregation method avoids the imperfection of the simple growing and sliding time window methods which avoid a social graph loosing heterogeneity and degenerating to a random graph over time [15].

### D. Community Structure

The definition of traditional community depends on the special community detection algorithms or social applications. In our hybrid underlying network, generally, we assume that if people have relatively stronger communication capability with each other, they can form community structure, called *space-crossing community*.

*Space-Crossing Community*: In our hybrid underlying network, due to the help of APs, some geographically distant nodes may have strong communication capability, therefore crossing the physical space. Therefore, the long distance nodes mixed with the close proximity nodes may form space-crossing communities.

First, mobile user pairs or user-AP pairs may form physical proximity communities by encountering each other frequently in their proximity, based on social graph, and groups of tight-knit nodes with more internal links than external links [10]–[12]. Next, we can merge the physical proximity communities through access points and obtain the final space-crossing communities. Here, we allow that space-crossing communities

[1] Usually, the time granularity of the records in the dataset is very small (in seconds). It is too short to reflect the social properties and form social graphs. Thus, $\Delta$ should be larger than the time granularity of the dataset.

can overlap with each other. The detailed description of *Space-Crossing Community Detection* algorithm is provided in Section III.

## III. SPACE-CROSSING COMMUNITY DETECTION

In the dynamic and hybrid underlying networking environment, we execute two steps to detect the space-crossing communities. First, we use AFOCS [8] to obtain physical proximity communities at each time slot. Second, using static combination criterion $\mathcal{S}^a$ and dynamic combination criterion $\mathcal{S}^b$, we get final space-crossing communities at each time slot.

### A. Preliminaries of AFOCS

AFOCS is a community detection algorithm for dynamic and overlapped networks. It is fit for the physical proximity community detection. At initial network snapshot, AFOCS partitions nodes into different groups and combine the overlapped groups to obtain the initial community structure. Subsequently, AFOCS classifies the dynamic changes into several simple actions, including the addition or removal of nodes or edges, and handling them respectively using local information. Comparing with other dynamic detection algorithms [16]–[21], it has the following advantages:

- It does not have the problems of resolution limit and extreme degeneracy experienced in the modularity $Q$-based method [22].
- It does not need the prior knowledge about community structure.
- It has more precise detection and can deal with dynamic and overlapped cases.

Although AFOCS cannot deal with the AP hybrid infrastructure for space-crossing community detection in MSNs, it is a good choice for our basic physical proximity community detection. Note that, from technical prospective, other detection methods also can be used in finding physical proximity community in Section III-B.

### B. Space-Crossing Community Detection Method

First, APs are regarded as common nodes. Based on dynamic graphs in Section II-A, at each time slot, AFOCS is applied to get physical proximity communities. Denote $i-th$ physical proximity community at time $t$ as $ComPP_i^t$. We have a dynamic time sequence of network physical proximity community structure, denoted by $\{\mathcal{PP}_0, \mathcal{PP}_1, ..., \mathcal{PP}_t, ...\}$, where $\mathcal{PP}_t$ represents the set of current physical proximity communities at time slot $t$.

Next, at the initial time slot, for the set of physical proximity communities $\mathcal{PP}_0$, through APs, we use a static combination criterion $\mathcal{S}^a$ to get the initial space-crossing communities. Afterwards, from $\mathcal{PP}_1$ to the timeout, to reduce the complexity of our detection method, a dynamic combination criterion $\mathcal{S}^b$ is used to get space-crossing communities $ComSC_i^t$ at each time slot. The combination criteria $\mathcal{S}^a$ and $\mathcal{S}^b$ are presented in the following Section III-C. Fig.1 gives an intuitive presentation of the space-crossing community detection.

### C. Combination Criterion

***Criterion*** $\mathcal{S}^a$: We combine the neighboring physical proximity communities containing APs in a clockwise or anticlockwise direction between APs (*i.e.*, AP ring) and execute this only once. Finally, we get one space-crossing community for each pair of APs. This method is one of the practical solutions. In the future, we can make an adaptive scheme considering the current network bandwidth and other factors.

***Criterion*** $\mathcal{S}^b$: In dynamic tracking phase, if the associated physical proximity communities of a certain AP change, a *local* combination of physical proximity communities between the changed AP and its neighboring AP to form a new space-crossing community may occur.

*Remark 3*: In Section III, we present a space-crossing community detection method. However, so far, there exist only several benchmarks [10], [23], [24] for evaluating the *goodness of fit* between the traditional community detection results (*e.g.* physical proximity community structure) and the ground truth, not fit for our space-crossing community. From another perspective, our space-crossing community is a new community structure for fully utilizing APs, aimed to improve data forwarding in MSNs. The evaluation of goodness of fit is not the main topic of this paper.

### D. Implementation of Space-Crossing Community Detection

In this paper, APs are used not only to support a hybrid underlying network and the concept of space-crossing community, but also can participate in the community detection and data forwarding. Therefore, we describe two hybrid schemes to implement our space-crossing community detection algorithm.

*Scheme I*: In one-layer AP hybrid infrastructure, APs are deployed in the network area, not covering the entire area. In the physical proximity community containing AP, the centralized algorithm is applied at AP. An AP knows its members and the relationships among them. In physical proximity communities without APs, the distributed algorithm is applied. Each node has accurate knowledge of its neighbors and some local approximate knowledge captured by its neighbors. The required information can be exchanged among neighboring nodes directly. For instance, a powerful node can be assigned to deal with some local changes.

*Scheme II* deals with multi-layer AP hybrid infrastructure. For example, there are two layers and in the top layer an AP is used as a supervisor to control all nodes in the network. Such supervisor is justified because some global preprocessing is executed only once in the algorithm at the initial phase, thus not overloading it.

Both schemes are hybrid. They can balance the network load by sharing the work among nodes and APs.

## IV. SAAS DATA FORWARDING SCHEME

In this section, based on Space-Crossing Community Detection outcome, we design a SAAS (Similarity Attraction and AP Spreading) data forwarding scheme to validate the positive role of the space-crossing community.

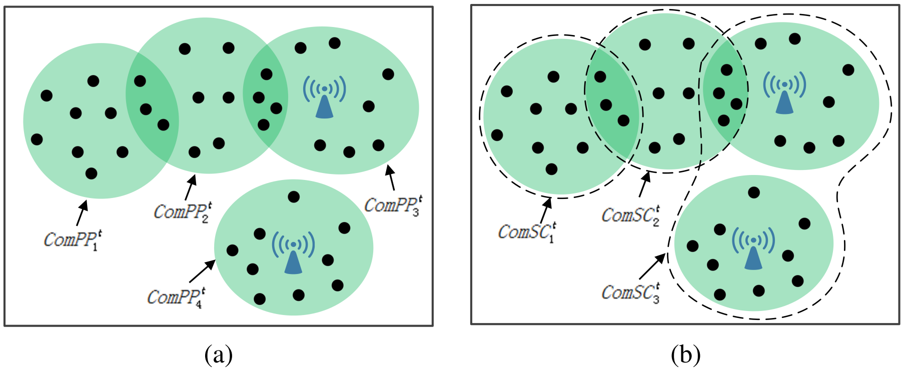



Fig. 1. Subfigure (a) shows Step 1 of space-crossing community detection method. Subfigure (b) shows Step 2. In Subfigure (a), the green dashed areas cover four physical proximity communities; In Subfigure (b), two physical proximity communities contained APs combine into a space-crossing community $ComSC_3^t$. The black dash line circles the final three space-crossing communities.

### A. Pearson Social Similarity

*Definition 1 (Local Activity):* Let $a_{u,i}^t$ denote the local activity of node $u$ in a space-crossing community $ComSC_i^t$ at time $t$. Then,

$$a_{u,i}^t = \frac{\sum_{(u,v)\in ComSC_i^t} w_{uv}^t}{\sum_{(v',v'')\in ComSC_i^t} w_{v'v''}^t}, 1 \le i \le k, v \ne u$$

where $v'$ and $v''$ are any two nodes in $ComSC_i^t$; $w_{uv}^t$ has been defined in Section II-C; $k$ represents the number of space-crossing communities; the numerator represents the sum of the encounter ratios between node $u$ and other nodes in community $ComSC_i^t$ and the denominator represents the sum of the encounter ratios between any two nodes in community $ComSC_i^t$.

Node's local activity can represent the importance of the node in a certain community. A larger local activity means that the node has more interactions with other members in the community. In data forwarding, local activity is important because if the message is given to a node having low local activity, it will cause a low efficiency in terms of delivery ratio.

*Definition 2 (Activity Vector):* We define an activity vector $A_t(u) = (a_{u,1}^t, a_{u,2}^t, ..., a_{u,i}^t, ..., a_{u,k}^t)$ for each node $u$ at time $t$, where $a_{u,i}^t$ denotes the local activity of node $u$ in space-crossing community $ComSC_i^t$ at time $t$. The value of $k$ represents the number of communities after applying the space-crossing community detection method.

There are some social similarity measurements which are often used in previous studies, such as cosine angular distance [25], Hamming feature distance [26], the number of common communities (interests groups) [8]. However, the *distance* based methods often map the social features into a geometric circle or a hypercube, which cannot give a meaningful explanation in real social networks. The *common interests* based method has a drawback that if we choose a node having more common communities with the destination as a relay node, the chosen node may be one with low local activity in its community. In this paper, we introduce Pearson Correlation Coefficient method to define social similarity.

*Definition 3 (Pearson Social Similarity):* Given two activity vectors $A_t(u) = (a_{u,1}^t, a_{u,2}^t, ..., a_{u,i}^t, ..., a_{u,k}^t)$ of node $u$ and $A_t(w) = (a_{w,1}^t, a_{w,2}^t, ..., a_{w,i}^t, ..., a_{w,k}^t)$ of node $w$, we define the social similarity between $u$ and $w$ at time $t$ as $SS_t(u,w)$, having

$$SS_t(u,w) = \frac{E(A_t(u)A_t(w)) - E(A_t(u))E(A_t(w))}{\sqrt{E(A_t(u)^2) - E^2(A_t(u))}\sqrt{E(A_t(w)^2) - E^2(A_t(w))}}.$$

Pearson correlation coefficient reflects the degree of linear dependence between two vectors. Local activity reflects the importance of a node in a certain space-crossing community. *Pearson social similarity* combines calculation of local activity and Pearson correlation coefficient. Assuming that node $w$ is the destination node, there exists a session (path) from node $u$ to node $w$. The candidate relay is node $v$. If the *Pearson social similarity* $SS_t(v,w)$ is larger than $SS_t(u,w)$, then, we can consider that the degree of linear dependence between $v$ and $w$ is larger than that of $u$ and $w$. In each vector component, node $v$ and $w$ are more anastomotic. Reflected in social networks, they are more similar in social aspects, *i.e.*, the interests groups and the local activity of node $v$ are proportional to those of node $w$. Intuitively, if the destination node is in two interests groups $ComSC_1^t$ and $ComSC_2^t$ at time $t$, and the node local activity in $ComSC_1^t$ is larger than in $ComSC_2^t$ then, the node that has the same characteristic with the destination is more appropriate as a relay than one that has not. Thus, a larger *Pearson social similarity* can guarantee the relay node having high chance to forward data successfully.

### B. SAAS Algorithm

In this section, we provide the details of SAAS (Similarity Attraction and AP Spreading) algorithm. It includes two phases: *Similarity Attraction Phase* and *AP Spreading Phase*, as illustrated in Fig. 2.

*1) Similarity Attraction Phase:* In this phase, each node has its activity vector. When the message holder meets another node, they will calculate their *Pearson social similarities* with the destination respectively. The message holder tries to send the message to a node which has larger *Pearson social similarity* than itself and let the node send the message to the destination consecutively.

**Algorithm 1** SAAS: a session from node $u$ to $w$ at time $t$

```
if message holder u is in a physical proximity community containing AP
  if nodes within community do not have the message copies
    node u will transmit the message to AP
    AP spreads copies to all nodes within community
else
  if node u encounters another node v without a message copy
    calculate SS_t(u, w) and SS_t(v, w)
    if SS_t(v, w) > SS_t(u, w)
      node u transmits the message to node v
    else
      node u maintains the message
```

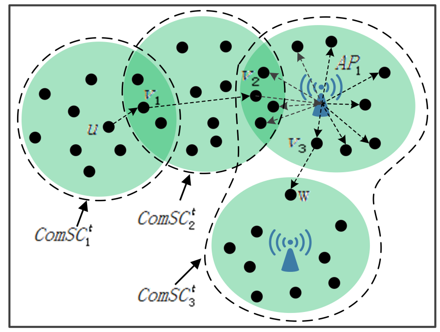



Fig. 2. A session from node $u$ to $w$. The black dotted arrow line means a node moving and forwarding path. Similarity attraction is applied at transmissions from $u$ to $v_1$, $v_1$ to $v_2$ and $v_3$ to $w$. When node $v_2$ receives the message, through Access Point $AP_1$, the message copies are spread in the physical proximity community containing AP.

*2) AP Spreading Phase:* In this phase, if the message holder enters into the physical proximity community containing AP, via the AP, it will give each mobile node within community a message copy and let the nodes jointly send copies to the destination. Note that, with our AP assumption, the ratio of the number of APs to the number of mobile users is small. Thus we limit the number of message copies and the AP spreading is locally carried out. Therefore, the aim of AP spreading is merely for competing with the similar type of forwarding algorithm using limited copies strategy, not for employing large number of copies to increase the delivery ratio. In this phase, the similarity attraction is not applied. That is, in the physical proximity community containing AP, if the encountered node has higher *Pearson social similarity* with the destination than the current message holder, the operation of sending the message will not occur between the holder and the encounter.

We describe SAAS algorithm, as shown in Algorithm 1. We do not distinguish the two phases in sequence. This is because the message exchange in SAAS is compatible with each phase.

*Remark 4*: SAAS is a distributed algorithm, in which each node only requires to calculate its *Pearson social similarity* with the destination and decides whether to send the message to the encountered node or spread message copies in the physical proximity community containing AP.

*Remark 5*: To keep the knowledge of each node synchronous, we update the community list periodically. This time interval is set according to the WiFi scanning time in the dataset.

*Remark 6*: With regard to dissemination among APs, in order to reduce the work load in APs, we assume that an AP does not talk to another AP.

## V. Performance Evaluation

In this section, we evaluate the performance of our SAAS data forwarding scheme. Specially, we evaluate SAAS on two kinds of social datasets. One is dense, the other is sparse. We will test if SAAS is related to the social density of MSNs.

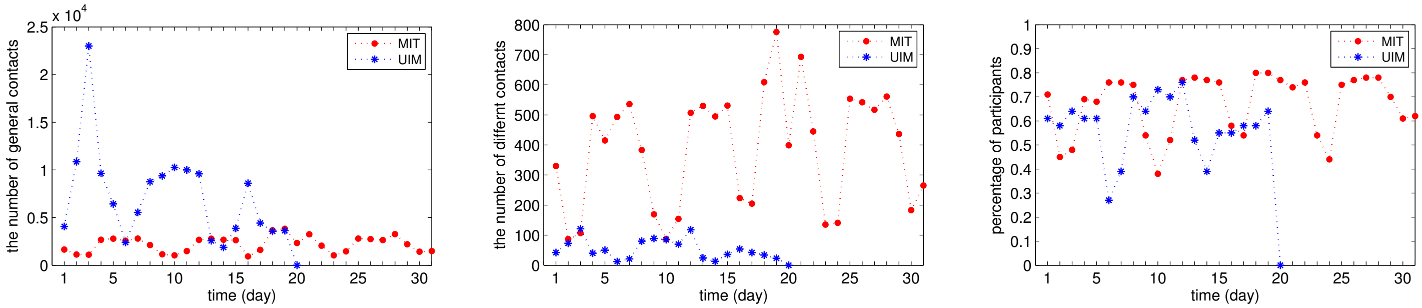
### A. Dataset Selection

The evaluation of SAAS is based on MIT Reality Mining [13] and UIM (University of Illinois Movement) [14]. From the trace analysis as illustrated Fig 3, MIT contains 2188.71 general contacts, 387.23 different contacts[2] and 67.19 percent of participants per day on average. While, UIM contains 6920.9 general contacts, 51.35 different contacts and 55.25 percent of participants per day on average. We can see, the participation degree of MIT and UIM is almost the same, however, the number of general contacts and the number of different contacts for MIT and UIM are opposite starkly. On average, there are 3.46 and 1.57 different contacts *per node* for MIT and UIM respectively. It means that, through contact aggregation (*i.e.*, the weight filtering method in Section II-C), the number of edges attached to a node in MIT is larger than in UIM. Thus, in our simulation, we say that MIT is a dense social network, while UIM is a sparse one. Note that the concept of dense/sparse network is just relative.

In MIT Reality Mining, 97 Nokia 6600 mobile phones were carried by users over the course of 9 months in MIT campus and its surroundings. In the long-term observation, the dataset records the contacts between mobile users and the contacts between users and visible GSM cell towers. In UIM, 28 Google Android phones were carried by users over the course of 3 weeks at the University of Illinois. It is a dataset that contains MACs of Bluetooth and WiFi access points captured by phone plug-in middleware periodically.

In two datasets mentioned above, for cell-phone calling requirements, the number of APs is large and APs cover almost entire network. In order to model our hybrid underlying network, we only select 15 APs and 5 APs at random for MIT Reality Mining and UIM respectively. Therefore, the coverage areas of APs do not cover entire network. Then, according to the method of contact aggregation for edges in Section II, we construct social graphs using the scanning records

[2]The general contact means the sum of all contacts among nodes in a period of time. The different contact means that, if there is more than one contact between two nodes, we will only count the number of contacts once.




(a) (b) (c)

Fig. 3. Subfigure (a), Subfigure (b) and Subfigure (c) show the number of contacts, the number of different contacts and the percentage of participants of MIT and UIM per day respectively. Note that for UIM, in original dataset, records of the 20th day are null.

in those datasets. For MIT Reality Mining, we choose the weighted growing window mechanism. For UIM, we choose the weighted sliding window mechanism.

### *B. Simulation Setup*

We choose the ONE simulator as our experimental tool [27]. It not only provides various mobile models including some complex mobility scenarios in daily life, but also can incorporate real world traces into the simulator. According to the contacts among Bluetooth devices and the contacts between Bluetooth devices and APs, we extract trace files from MIT Reality Mining and UIM datasets. These discrete contact events can be taken as the inputs of the ONE simulator. The datasets include the start time, end time and communication peers. We set the start time as connection up and the end time as connection down. One example of the trace file extracted from MIT Reality Mining is shown as follows:

| | | | | |
|---|---|---|---|---|
| 0 | $CONN$ | 93 | 96 | $up$ |
| 0 | $CONN$ | 93 | 14 | $up$ |
| 128 | $CONN$ | 85 | 17 | $up$ |
| 129 | $CONN$ | 94 | 29 | $up$ |
| | | $\cdots$ | | |
| 1169 | $CONN$ | 28 | 5 | $down$ |
| 1169 | $CONN$ | 28 | 17 | $down$ |
| | | $\vdots$ | | |

For all simulations in this work, each node generates 1000 packets during the simulation time. The packet size is distributed from $50KB$ to $100KB$ uniformly. In the hybrid underlying network, the interface of users (cell-phone carriers) is assigned to two modes: Bluetooth and WIFI, and the interface of APs is assigned to WIFI. Data transmission speed of Bluetooth is $2Mbps$ and transmission range is in $10m$. Data transmission speed of WIFI is $5Mbps$ and transmission range is in $100m$. The scanning interval of Bluetooth is $5min$ and $1min$ for MIT Reality Mining and UIM respectively. The scanning interval of WIFI is $30min$. The buffer size of each node is $5MB$. The source and destination pairs are chosen randomly among all nodes. Each simulation is repeated 20 times with different random seeds. Without losing precision, we set the update interval is 1. For MIT Reality Mining, we set TTL from 30min to 1mon[3]. For UIM, we set TTL from 30min to 3week. In these settings, some come from the MIT and UIM datasets, others are set according to common sense. They are not the determinant parameters in SAAS and the following comparison algorithms.

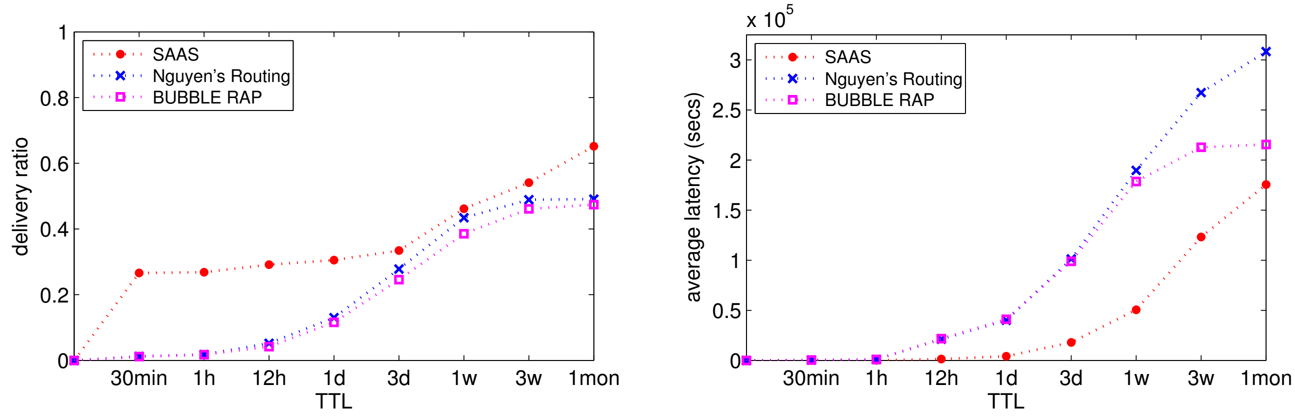
### *C. Experiments on SAAS Data Forwarding Scheme*

*1) Comparisons with Other Forwarding Schemes:* In this section, we compare our SAAS algorithm against BUBBLE RAP [5] and Nguyen's Routing [8] (*i.e.*, two social community-based routing algorithms).

BUBBLE RAP provides a hierarchical forwarding strategy. A node first bubbles the message up the hierarchical ranking tree using the global centrality. When the message reaches the community of the destination node, local centrality is used instead of the global centrality. In Nguyen's Routing, a smart community detection algorithm is proposed and applied to data forwarding in mobile networks. A message is forwarded to an encountered node if the node shares more common communities with the destination than the current one. Note that we select settings or parameters which bring about the best performances for above two alternative algorithms respectively.

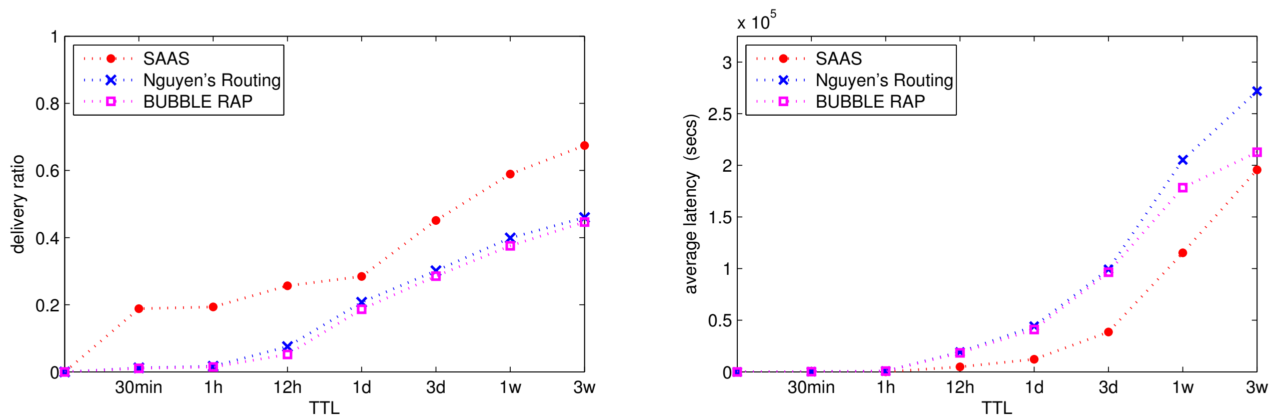
Fig. 4 and Fig. 5 show the delivery ratio and average latency of our SAAS, BUBBLE RAP and Nguyen's Routing algorithms in MIT Reality Mining and UIM datasets respectively. From Fig. 4 and Fig. 5, we can observe, for both MIT Reality Mining (dense social networks) and UIM (sparse social networks), the delivery ratio of SAAS achieves best among those algorithms and the average latency is lowest. This is because SAAS tends to combine the physical proximity communities containing APs to utilize APs for data forwarding. For example, if two physical proximity communities contained APs do not combine (*i.e.*, they belong to two different space-crossing communities), the two activity vectors of two nodes (one is as a message holder in a physical proximity community

[3]MIT Reality Mining Dataset is a long-term observation dataset. Thus, some cumulative social phenomena (local activity, community structure *etc.*) require a period of time to reveal. We do experiments from date $2004-10-01$ to date $2004-10-31$, *i.e.*, a large TTL-1 month, instead of several days. A larger TTL(larger than 1 month) also can be done with more simulation time and the overall trend is similar with 1 month.




(a) (b)

Fig. 4. Simulation Results on MIT Reality Mining Dataset. Subfigure (a) and Subfigure (b) show the delivery ratio and the average latency respectively.




(a) (b)

Fig. 5. Simulation Results on UIM Dataset. Subfigure (a) and Subfigure (b) show the delivery ratio and the average latency respectively.

containing AP, the other is as the encountered node in the neighboring physical proximity community containing AP) will differentiate largely in different vector components. Thus, the *Pearson social similarity* between the two nodes will be small. The result is that the encountered node will be abandoned as a relay. If they combine, the values of the two nodes' local activity vectors will be similar in components. The *Pearson social similarity* between the two nodes will be large which can make APs play roles in data forwarding.

In MIT Reality Mining dataset, Fig.4 (a) shows that SAAS performs best among those algorithms. Its delivery ratio is higher than Nguyen's Routing with 64.82 percent, BUBBLE RAP with 79.25 percent on average. In UIM dataset, Fig.5 (a) shows that SAAS performs best among those algorithms. Its delivery ratio is higher than Nguyen's Routing with 76.22 percent, BUBBLE RAP with 89.93 percent on average. In terms of delivery ratio, at the initial phase, due to the help of APs, it is obvious that SAAS increases quickly in both MIT and UIM datasets. BUBBLE RAP use betweenness as centrality metrics without considering node contact frequency. As long as there exists an edge between two nodes, the edge will be used in betweenness calculation. But in real social networks, the edge may only have a trivial effect in data forwarding. Thus, it has a lower delivery ratio than SAAS. Nguyen's Routing tends to send messages to nodes having many interests with the destination, however, it may deliver them to nodes which have low local activity in their communities (or interests groups). It is the main reason for the low delivery ratio of Nguyen's Routing. Fig. 4 (b) and Fig. 5 (b) show that the delays of those algorithms all increase with TTL increasing. SAAS shows the predominant performance among them. Due to the help of space-crossing communities, some long-distance nodes can communicate through short-path across the geographical space, which lead to the low delay of SAAS.

*2) The Role of Space-Crossing Community:* In SAAS algorithm, when calculating *Pearson social similarity*, we require the community detection outcome of the current social network. Here, especially, we use space-crossing community detection results and AFOCS [8] detection, respectively, to validate the role of space-crossing community.

From Fig. 6 (a) and Fig. 6 (b), we can observe, in both

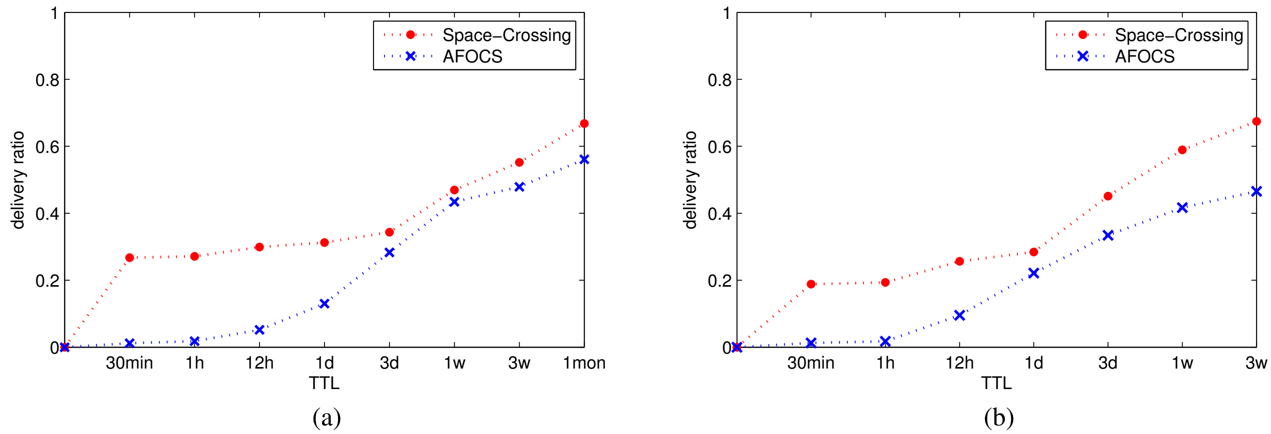



Fig. 6. The role of space-crossing community. Subfigure (a) is based on MIT Reality Mining dataset and Subfigure (b) is based on UIM dataset.

MIT Reality Mining and UIM datasets, that SAAS using the space-crossing community detection shows better performance than the SAAS using AFOCS detection in data forwarding. It validates the important role of space-crossing community in data forwarding for MSNs.

From above results and analysis, SAAS has proven its competitive ability in both dense and sparse social networks. In nature, it benefits from the strong communication community (space-crossing community) created by APs. According to space-crossing communities and node local activity, nodes can choose the appropriate relays to achieve a high efficient data forwarding.

## VI. Related Work

In this paper, we propose a Space-Crossing Community Detection method for the hybrid AP underlying infrastructure and study its impact on data forwarding in MSNs.

For community detection, there exist many classical centralized algorithms that are applied in the area of social networks, biological networks, commercial networks and so on. The recent reviews [12] and [28] may serve as introductory reading in this domain. In the pioneering work, Newman and Girvan [29] constructed communities by removing links iteratively based on the betweenness value. The concept of MODULARITY $Q$ was given to estimate the goodness of a community partition. Then, Newman [30] and Leicht *et al.* [31] extended the work to a weighted and a directed community detection respectively. Based on MODULARITY $Q$, many optimized algorithms were proposed [32]–[35]. Further, as a milestone work, Palla *et al.* [36] proposed a K-CLIQUE method to address the overlapped problem in community detection.

The dynamic community detection problem arises in the development of mobile networks. Proposed algorithms include Particle-And-Density [17] and QCA [18]. However, above centralized community detection methods have computation costs and are difficult to implement in a distributed ad hoc manner for MSNs. Hui's distributed community detection method [37] and AFOCS [8] decentralized method address these issues. Nevertheless, so far, both centralized and distributed methods aim at the non-organized distributed underlying infrastructure. They ignore the hybrid AP organization that exists in practice. From our new perspective, we give the concept and the detection method of the space-crossing community inspired by this hybrid structure.

Some studies have shown that exploiting social relationships can achieve better data forwarding performances. Daly and Haahr [38] proposed SimBet forwarding algorithm in Delay Tolerant Networks (DTNs). It used betweenness centrality and social similarity to increase the probability of a successful data forwarding. Hui *et al.* [5] proposed an algorithm called BUBBLE RAP in DTNs, with making use of node centrality and weighted $k$-clique community structure to enhance delivery performance. Gao *et al.* [6] studied multicast in DTNs from the social network perspective. Fan et al. [7] studied a geo-community-based broadcasting scheme for mobile social networks by exploiting node geo-centrality and geo-community. Nguyen *et al.* [8] proposed a community-based data forwarding algorithm called Nguyen's Routing, by using the number of common interests as forwarding criterion. Wu *et al.* [39] proposed a community home-based multi-copy routing scheme in MSNs. However, none of existing schemes considers the positive role of space-crossing introduced by the hybrid underlying network with APs support in data forwarding.

## VII. Conclusion

In this paper, we study a more realistic underlying infrastructure for MSNs, *i.e.*, the hybrid underlying network with APs support. Due to the help of APs, this hybrid infrastructure motivates a new concept of space-crossing community. We propose a space-crossing community detection method and describe a novel and high efficient data forwarding scheme SAAS based on it. Our future work will focus on the relationship between the density of APs and the performance of MSNs.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments. The research of authors is partially supported by the National Basic Research Program of China (973 Program) under grant No. 2010CB328101, the Integrated Project for Major Research Plan of the National Natural Science Foundation of China under grant No. 91218301, the the National Natural Science Foundation of China (NSFC) under grant No. 61202383, 61003089, the Program for New Century Excellent Talents in University (NCET) under grant No. NCET-12-0414, the Natural Science Foundation of Shanghai under grant No. 12ZR1451200, the National Science Foundation for Postdoctoral Scientists of China under grant No. 2012M510118, the Research Fund for the Doctoral Program of Higher Education of China (RFDP) under grant No. 20120072120075, the Special Foundation of China Postdoctoral Science under grant No. 2013T60463, NSERC Discovery grant, NSERC CRDPJ445199-12; 2013-2015, and the Government of China for the Tsinghua 1000 Plan Distinguished Professor (2012-5) position.

## REFERENCES

[1] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "Mobiclique: middleware for mobile social networking," in *Proc. ACM WOSN 2009*.

[2] "Foursquare." http://foursquare.com.

[3] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," in *Proc. IEEE ICDCS 2010*.

[4] "Sony ps vita-near." http://us.playstation.com/psvita.

[5] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *Proc. ACM MobiHoc 2008*.

[6] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proc. ACM MobiHoc 2009*.

[7] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun, "Geo-community-based broadcasting for data dissemination in mobile social networks," *to appear in IEEE Trans. on Parallel and Distributed Systems*, 2012.

[8] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai, "Overlapping communities in dynamic networks: their detection and moibile applications," in *Proc. ACM MobiCom 2011*.

[9] Z. Li, C. Wang, S. Yang, C. Jiang, and X. Li, "Lass: Local-activity and social-similarity based data forwarding in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, Accepted.

[10] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[11] M. Porter, J. Onnela, and P. Mucha, "Communities in networks," *Notices of the AMS*, vol. 56, no. 9, pp. 1082–1097, 2009.

[12] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[13] K. Nahrstedt and L. Vu, "CRAWDAD data set uiuc/uim (v. 2012-01-24)," Downloaded from http://crawdad.cs.dartmouth.edu/uiuc/uim, Jan. 2012.

[14] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.

[15] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know thy neighbor: Towards optimal mapping of contacts to social graphs for dtn routing," in *Proc. IEEE INFOCOM 2010*.

[16] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 2, p. 8, 2009.

[17] M.-S. Kim and J. Han, "A particle-and-density based evolutionary clustering method for dynamic networks," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 622–633, 2009.

[18] N. P. Nguyen, T. N. Dinh, Y. Xuan, and M. T. Thai, "Adaptive algorithms for detecting community structure in dynamic social networks," in *Proc. IEEE INFOCOM 2011*.

[19] D. Duan, Y. Li, Y. Jin, and Z. Lu, "Community mining on dynamic weighted directed graphs," in *Proc. ACM CNIKM 2009*.

[20] S. Bansal, S. Bhowmick, and P. Paymal, "Fast community detection for dynamic complex networks," in *Complex Networks*. Springer, 2011, pp. 196–207.

[21] R. Cazabet, F. Amblard, and C. Hanachi, "Detection of overlapping communities in dynamical social networks," in *Proc. IEEE SocialCom 2010*.

[22] A. Khadivi, A. Rad, and M. Hasler, "Network community-detection enhancement by proper weighting," *Physical Review E*, vol. 83, no. 4, p. 046104, 2011.

[23] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.

[24] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016118, 2009.

[25] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguñá, and D. Krioukov, "Popularity versus similarity in growing networks," *Nature*, vol. 489, pp. 537–540, 2012.

[26] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *Proc. IEEE INFOCOM 2012*.

[27] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for DTN protocol evaluation," in *Proc. ICST SIMUTools 2009*.

[28] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E*, vol. 80, no. 5, p. 056117, 2009.

[29] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, 2004.

[30] M. Newman, "Analysis of weighted networks," *Physical Review E*, vol. 70, no. 5, p. 056131, 2004.

[31] E. Leicht and M. Newman, "Community structure in directed networks," *Physical Review Letters*, vol. 100, no. 11, p. 118703, 2008.

[32] A. Clauset, M. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, p. 066111, 2004.

[33] R. Guimera and L. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, 2005.

[34] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P10008, 2008.

[35] M. Rosvall and C. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, p. 1118, 2008.

[36] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[37] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proc. ACM MobiArch 2007*.

[38] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. ACM MobiHoc 2007*.

[39] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multi-copy routing in mobile social networks," in *Proc. IEEE INFOCOM 2013*.