

Data Dissemination Using Interest-Tree in Socially Aware Networking

Feng Xia^{a,*}, Qiuyuan Yang^a, Jie Li^a, Jiannong Cao^b, Li Liu^a, Ahmedin Mohammed Ahmed^a

^a*School of Software, Dalian University of Technology, Dalian 116620, China*

^b*Department of Computing, Hong Kong Polytechnic University, China*

Abstract

Socially aware networking (SAN) exploits social characteristics of mobile users to streamline data dissemination protocols in opportunistic environments. Existing protocols in this area utilized various social features such as user interests, social similarity, and community structure to improve the performance of data dissemination. However, the interrelationship between user interests and its impact on the efficiency of data dissemination has not been explored sufficiently. In this paper, we analyze various kinds of relationships between user interests and model them using a layer-based structure in order to form social communities in SAN paradigm. We propose Int-Tree, an Interest-Tree based scheme which uses the relationship between user interests to improve the performance of data dissemination. The core of Int-Tree is the interest-tree, a tree-based community structure that combines two social features, i.e. density of a community and social tie, to support data dissemination. The simulation results show that Int-Tree achieves higher delivery ratio, lower overhead, in comparison to two benchmark protocols, PROPHET and Epidemic routing. In addition, Int-Tree can perform with 1.36 hop counts in average, and tolerable latency in terms of buffer size, time to live (TTL) and simulation duration. Finally, Int-Tree keeps stable performance with various parameters.

Keywords: data dissemination, social awareness, community structure, user interest, tree structure.

1. Introduction

The popularity of mobile devices such as smart phones has increased contact opportunities between mobile users in pervasive environments. In this setting, mobile carriers (i.e. human beings) communicate via Bluetooth and Wi-Fi technologies in order to share different kinds of information (such as photos, commercial trades, etc.) between interested users. However, opposed to conventional mobile ad hoc networks, an end-to-end connectivity between users might not exist and they have to carry messages until a new contact is established [1]. Delay Tolerant Networks (DTNs) [2, 3] are special kinds of networks which use *store-carry-and-forward* scheme to forward data between disconnected users. In this paradigm, mobile devices mirror movement patterns and attributes of their owners (i.e., users), hence social characteristics and features of users can be exploited to improve the performance of forwarding protocols. This drives the emergence of socially aware networking (SAN) [4, 5], which aims at exploring social relationships and properties of network users to streamline routing and data dissemination protocols [6, 7, 8].

Social attributes, relationships, and behaviors of mobile users are relatively stable through a long period and they have long-term characteristics. Hence, they have been extensively used to improve the performance of data forwarding algorithms. Contemporary researches in this area mainly use social network analysis [9] techniques to extract different social properties of users. The commonly used social properties include social sim-

ilarity, tie strength, community, node centrality, mobility pattern, etc. Among these, the community structure has been widely used in recent routing protocols such as in [10, 11, 12]. In this strategy, socially similar individuals form a community where the similarity could be friendship, common visiting locations, or similar interests. Generally, individuals inside a community contact one another frequently and this can be beneficial for choosing a proper forwarder within a community.

An additional important social factor that is used extensively to improve the efficiency of data dissemination protocols in SAN paradigm is interest. This is because people with similar interests meet each other frequently and share more data with each other [13]. Hidi [14] provided theoretical proofs on what interest is and how it drives human beings in acquiring knowledge. Studying human behaviors [15, 16, 17] also showed that daily activities of users in social networks such as browsing, cooperation in online societies and playing on-line games are mainly driven by their interests. Furthermore, existing routing protocols [18, 19, 20] validated the value of user interests for data dissemination. Specifically, interests can be assigned equal values of importance as keywords [21], or be extracted from various vectors according to similarity for routing algorithm design [18]. In addition, user interests can constitute interest lists [19, 20] to help to predict movement patterns or build a multi-cast tree. However, all these interest-based approaches have not explored thoroughly on the inherent relationships between user interests and their effect on data dissemination.

In this paper, we get the inspiration from the above-mentioned two social features, i.e., community structure and user interests, in order to answer the following research ques-

*Corresponding author; Email: f.xia@ieee.org

tions: 1) which kinds of relationships can be available between user interests, 2) how to model the interrelationship between user interests, and 3) how the relationships between user interests affect the performance of a data dissemination protocol. To achieve these goals, we analyze relations between user interests and devise an **Interest-Tree** based scheme (Int-Tree) for data dissemination. First, we build the *interest-tree*, a tree-based community structure according to interests of users which is updated dynamically. Then, density of a community and social tie are calculated by social awareness. When a source node contacts an intermediate node, they update their density and social tie information. After that, Int-Tree decides whether the intermediate node is suitable to be a forwarder in accordance with different criteria in forwarding strategy module.

Hereby, we intend to clarify three points: 1) Int-Tree is the name of our scheme whereas interest-tree is the structure we constructed for presentation of community structure; 2) We divide communities based on user interests (i.e. one interest representing one community); and 3) We focus on the effects of relations between user interests and simplify the problem as one-to-one (i.e. one source and one destination) dissemination, which is the basis for one-to-many dissemination.

Int-Tree is based on our previous work BEEINFO [22]. The major difference between this work and the prior one lies on the exploration of relationships between user interests and how they affect the performance of data dissemination in a community-based SAN paradigm. Our major contributions to support this idea are summarized as follows:

- We study the relationships between user interests in SAN paradigm and propose a layered model to map user interests into different levels. Our model is able to present various elements of relationships between user interests including interest inclusion, cross-layer interests and interest intersection.
- We build an interest-tree to illustrate interest inclusion which is a special kind of relationship between user interests. The structure can support to combine the social features density of a community and social tie for data dissemination.
- We conduct extensive simulations that demonstrate the performance and effectiveness of Int-Tree in comparison to Epidemic [23] and PROPHET [24] protocols in terms of delivery ratio, overhead, average latency and hop count. Simulation duration, buffer size and Time-to-Live (TTL) are the most important parameters which are adopted in our simulation.
- We also carry out further simulations to explore how Int-Tree performs under different values of γ , and evaporation factors (α and β).
- Considering the situation of multiple interests, we provided discussions on the challenges and solutions.

The rest of the paper is organized as follows. An overview on interest-based data dissemination protocols, as well as

community-based forwarding algorithms is presented in Section 2. Section 3 describes our interest-based data dissemination problem and Section 4 presents a layered model to analyze the relationships between user interests. Section 5 describes Int-Tree as well as the components. Section 6 presents the simulation results, compares the performance of Int-Tree, PROPHET and Epidemic protocols, and explores the influence of changing parameters on Int-Tree. Section 7 discusses some problems raised by multi-interest situation, alongside the solutions. Finally, Section 8 concludes this paper.

2. Related Work

Several well-designed data forwarding protocols have been proposed for DTNs that were inspired from Epidemic [23] and PROPHET [24] routing algorithms. These algorithms were mainly proposed for intermittently connected networks and did not use the social characteristics of users. In the Epidemic routing, messages are flooded to encounter nodes with unlimited replication policy which results in a data congestion problem in the network. To cope with this problem, several routing protocols have been proposed aiming at limiting the number of message replicas and leveraging a tradeoff between resource usage and message delivery. The PROPHET is a controlled flooding algorithm which makes use of delivery predictability metric to estimate the probability of next relay nodes to deliver messages to destination nodes. These protocols are the foundation of our work since we use the contact history of network nodes to predict the future contacts between the nodes.

According to a definition of interest given by Hidi [14], interest has positive effects, such as contributing to increasing comprehension, and motivating thoughts and actions of people. The theory led researchers to study interest-based human behaviors. Zhou *et al.* [15, 16, 17] proved that some human activities such as rating movies, web browsing, and mobile phone text-messages are interest-driven. They also proposed an interest-driven model and explained the observed relationship between activities and power-law exponents. However, the proposed model did not consider which factors affect the user interests. To address this shortcoming, Yan *et al.* [25] studied the posting behavior of micro-blog users in mobile internet and concluded that social concern affect users' interest. Additionally, they proposed a model with social concern to slow down the decay of interest. Carofiglio *et al.* [26] presented Interest Control Protocol (ICP) to achieve fully efficient and fair flow control for content-centric networking. In spite of the fact that the mentioned methods have explored the user interest from social network perspective and proved that user interests affect human behaviors, they still failed in describing how interest drives human behaviors.

As for socially aware data dissemination, Costa *et al.* [19] proposed SocialCast and exploited predictions based on metrics of social interaction (e.g. receivers' interests, social ties) to identify the best information carriers. In SocialCast, each node broadcasts a control message to its neighbors containing the list of interests and the corresponding list of utility values.

This information is key to making message forwarding decisions. Similarly, Zhu *et al.* [20] applied interest list to their publish/subscribe service, named Ripple, in order to update request dissemination in the cloud.

In [21], the authors introduced Cooperative User Centric Information Dissemination (CUCID) Scheme. The scheme uses keyword space to describe user interests and each user has a vector of probabilities describing how interested the user is in data items described by the keywords. Chen *et al.* [18] proposed SPOON that used an interest extraction algorithm to derive a node's interests, file vector, group vector and node vector. Moreover, it groups common-interest nodes into communities by calculating their similarities. Bjurefors *et al.* [27] examined the data-centric architecture of Haggie and explored how interests and data were presented and distributed. Specifically, nodes express their interests in the form of attributes. A node description is a data object that consists of the node's interests and metadata. The authors in [28] considered contact patterns and interests of users to ensure effective data relaying.

The discussed algorithms in the last two paragraphs explored the impact of user interests on data dissemination. Ripple and Haggie studied the overlap of interests, while SPOON used similarity of vectors to reflect an indirect relation between user interests. In SocialCast, CUCID, and [28], the relation between user interests have not been exploited.

Dissimilar to the above efforts, Wu and Wang [29] recently extracted nodes' social features to help accomplish the multi-path routing. They believe that *people contact with each other more frequently if they have more social features in common*, which is nearly the same as what we stated in this paper. However, our work differs from theirs by focusing on the relationships of user interests and the consequent effects.

Beyond interest and social ties, community is also a widely used concept for data forwarding in SAN paradigm. Some community-based protocols have been proposed. In LABEL [30], nodes deliver messages only to the members in destination community. BUBBLE RAP [31] uses community and central nodes to construct socio-aware overlay for effective routing. The main drawback of these methods is high cost of constructing and maintaining the network overlay. LocalCom [32] and Gently [33] take inter-community and intra-community routings into consideration. LocalCom adopts similarity feature for further detection of community, while Gently uses LABEL and CAR (Context Adaptive-aware Adaptive) [34] for different routing phases. Home spread (HS) [12] considers the locations as *community homes* or *homes* where nodes visit frequently. The concept of *community homes* is also adopted in Community-Aware Opportunistic Routing (CAOR) [11]. Another example is BEEINFO [22] in which the authors introduced the concept of community density and divided communities according to users interests. In this method, optimal relay nodes are chosen based on community density and social tie.

In this paper, we go one step further by exploring the relationship between user interests and its impact on data dissemination in SAN paradigm. To this end, a tree-based structure is presented in order to model user interests and their relationships. Based on the tree structure, the network is divided into

different communities, combining community density and social tie to support data dissemination. Our method differs from the existing protocols by considering the fact that the interrelation between user interests are used to form a community structure which improves the performance of data dissemination in SAN paradigm.

3. Problem Statement

In most of the interest-based data dissemination protocols, it is commonly assumed that users have a list of interests and they give little attention to the relations between user interests. To address this issue, we present a real life scenario to highlight the problem to be solved as follows:

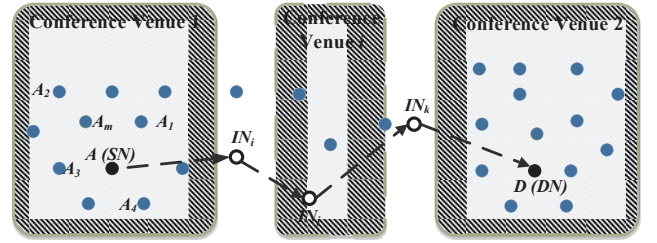


Figure 1: A real life scenario where Alice (A) has some items to forward to David (D). The empty circles and the dashed arrows constitute a possible route from A to D.

Suppose that Alice (A) is attending a presentation during an academic conference at Conference Venue 1, along with other attendees ($\{A_1, A_2, \dots, A_m\}$), as shown in Fig. 1. There are other venues in the meantime and Conference Venue i is a random venue represented by dashed round rectangle. Inspired by the presentation, she finds interesting items, such as a paper related to the topic or a video clip she just recorded. She wants to share these items to a friend of hers, David (D). However, A's movement is restricted for a while because of the presentation and D is out of A's range being in another venue. If the backbone network can not provide service, the direct contact between A and D will not be possible. Thus, to finish the dissemination, A has to rely upon other attendees who can contact D with higher possibility. Fig. 1 also shows a possible route from A to D in this scenario. In the figure, the black dots represent A and D. The empty circles are the potential forwarders. The route from A to D consists of dashed arrows, meaning that there might be more forwarders on the route. The other dots are random nodes in the network.

Here comes the problem: how to choose proper forwarders among the potentials to achieve efficient dissemination? Our solution to this problem is based on interests since people with the same interest meet with one another in higher probability. Additionally, user interests correlate to each other. Therefore, we examine the relations between user interests to help A choose forwarders to the target destination.

4. Relationship between User Interests

In social networks, people usually have a few interests which are stable over a time. Homophily [13] describes that similar characteristics between individuals generally result in a bond between them. As the saying “birds of a feather flock together” suggests, people with similar interest, backgrounds or beliefs tend to form stronger relationships than those with dissimilar ones. This statement can also be supported by a social network theory which is proposed in [35]. Based on this theory, the file interests have clear structured categories, and for the majority of users, 80% of the shared files fall into 20% of the file categories [36]. This is obvious on a campus or a conference venue. For example, the attendees in the same conference room share the same interest with the speech or the keynote, whereas those in different venues are likely to be interested in dissimilar topics.

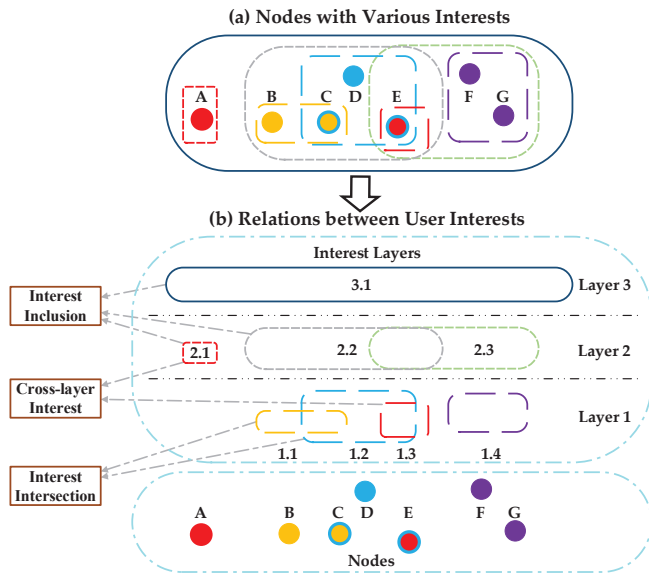


Figure 2: Transition of user interests in part (a) to a layered structure in part (b). The figure illustrates the relations between user interests, including interest inclusion, cross-layer interests, and interest intersection.

Fig. 2 illustrates the transition from nodes (i.e. users) with multiple interests (Fig. 2(a)) to a layered structure (Fig. 2(b)). In Fig. 2(a), there are seven nodes (A, B, C, D, E, F, G) with different interests. Nodes with the same interest have the same color, such as Nodes F and G. If a node (e.g. Node C) is featured with a different color in its border, it has at least two interests. The nodes with similar interests are grouped into dashed or solid rounded rectangles. For example, Nodes B, C, D and E are grouped in the gray dashed rounded rectangle while all the seven nodes are in the solid rounded rectangle. In Fig. 2(b), there are 3 layers of interests. All the interests in layers 1-3 are numbered for simplicity. The figure helps to introduce the relations between user interests, including inclusion, cross-layer and intersection.

In the rest of this section, we borrow some concepts from the

ACM Computing Classification System (CCS) ¹ to explain the various kinds of relations between user interests shown in the left part of Fig. 2(b). However, we are aware that ACM CCS is not enough to describe the complexity about user interests and their relations. For example, users are featured with multiple interests and the interests are related to each other. Hence, ACM CCS provides its value here as an instance. The detailed discussion on multiple interests is presented in Section 7.

4.1. Interest Inclusion

Interest inclusion is a special kind of relation between user interests in which individuals have different interests with diverse scopes and some interests contain others entirely. The rounded rectangles in layers 1-3 in Fig. 2(b) shows the relation of interest inclusion. For example, Interest 3.1 contains all the other interests, and Interest 2.2 contains Interests 1.1-1.3. Additionally, this can be supported by ACM CCS. The keywords in ACM CCS refer to different interests, such that, broader-scope interest like *architectures* covers more specific ones like *serial architectures* and *parallel architectures*, and then the much specific ones (e.g. *multiple instruction*, *multiple data*).

4.2. Cross-layer Interests

Cross-layer interests means an interest appears in at least two layers of an interest-tree as shown in Fig. 2(b). Interests 2.1 and 1.3 in Layers 2 and 3 are the same interest but appear in two layers. This is also reflected by ACM CCS in the case that *sensor networks* is listed in both *Computer systems organization/Embedded and cyber-physical systems* and *Networks/Network types/Cyber-physical networks* which are different layers.

4.3. Interest Intersection

Interest intersection can be understood as overlap. For instance, a node’s interests cover *mobile networks* and *ad hoc networks*. According to CCS Concept, these areas are relevant to each other, meaning they share some features in common. Consequently, this is a situation of interest intersection, as what is shown by Interests 1.1 and 1.2 in Layer 3 of Fig. 2(b).

5. Design of Int-Tree

5.1. Overview

Int-Tree makes use of user interests to construct social communities. Nodes constitute a community when they share the same interests. The components of Int-Tree are depicted in Fig. 3. From this figure, Int-Tree includes *community structure*, *social awareness*, *forwarding strategy*, *message scheduling*, and *buffer management* modules. The community structure is for constructing interest-tree. The structure contains interest information of communities which is of crucial importance in presenting the density of communities, and social tie information of social awareness module. Social awareness is responsible

¹<http://www.acm.org/about/class/2012>.

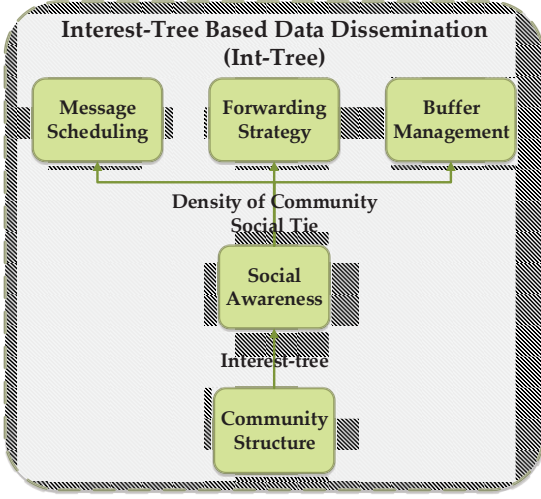


Figure 3: Components of Int-Tree.

for perceiving information of density and social tie. The available social information of nodes make it possible for Int-Tree to choose the appropriate relay nodes and handle efficient message scheduling as well as buffering mechanism. The forwarding strategy takes different strategies to select the next forwarders inside and between the communities. We note that Int-Tree directly applies message scheduling and buffer management modules from BEEINFO [22].

During the forwarding process, some social information of network such as the community, density of the community and social tie are maintained by nodes. The available feature makes our forwarding scheme flexible with the absence of infrastructure in SAN which is adaptable to more dynamic environment. In the following subsections, detail description for each feature is presented.

5.2. Social Awareness in Int-Tree

5.2.1. Community Structure

In Int-Tree, we consider user interests as their social features to form the community structure. In other words, nodes with the same interest form a community. Because of node mobility, nodes in the same geographical area may belong to different communities. Hence, Int-Tree detects communities based on interests, instead of space which is more robust when the parameters of network environment change quickly.

In Section 4, we analyzed the different kinds of relationships between user interests. All these relations contribute to construct a community structure. Intrigued by ACM CCS and researches in [20] and [37] we adopt tree to build interest-tree as the community structure. Because each node has at least one interest and nodes with the same interest form a community, a node can belong to different communities. We regard this as multi-interest situation. It will raise more questions and make our problem more complicated (See Section 7 for detailed discussions). To simplify the problem, we adopt a concept, *major interest*, to refer to a users major research interest. This reflects

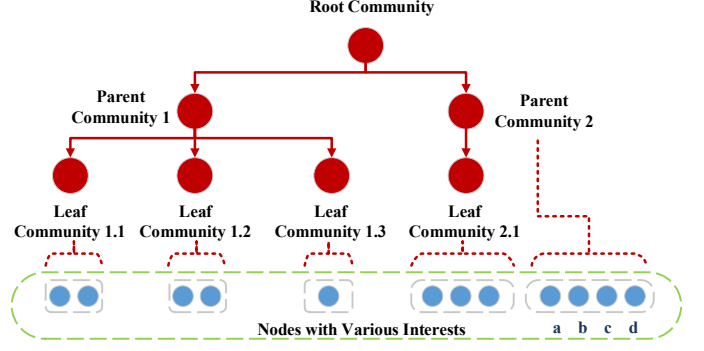


Figure 4: An example of the tree-based community structure (interest-tree).

the reality since each academic scholar devotes most of his/her time and energy on a major research area. Therefore, if there is no specified statement, the problem remains a single interest problem. We treat interest and community as one concept in this paper for simplicity. Thus in the interest-tree, each node corresponds to a community, as shown in Fig. 4. Generally, there are several layers in an interest-tree. A root community with the largest scale is on the top, then parent communities with medium scales in the middle layers and leaf communities with smallest scales. A node can belong to either a parent community or a leaf community, as a result of his/her preference on describing research interests. Considering the limit of space, only one layer of parent communities is chosen for clearance.

In an interest-tree, each node i belongs to a community C_i . For two nodes i and j , there is a common parent community $C_{i \& j}$. We define $ComSeq(C_i \rightarrow C_j)$ as a community sequence from community C_i to $C_{i \& j}$ ($ComSeq(C_i \rightarrow C_{i \& j})$) at first and then from $C_{i \& j}$ to community C_j ($ComSeq(C_{i \& j} \rightarrow C_j)$), along the interest-tree. Take Fig. 4 as an example, the community sequence between Leaf Community 1.1 and Leaf Community 2.1, $ComSeq(1.1 \rightarrow 2.1)$ is $\{1.1, 1, Root\ Community, 2, 2.1\}$ (some words omitted due to space restrain). Similarly, $ComSeq(1.1 \rightarrow 1.2) = \{1.1, 1, 1.2\}$. Compared to plenty of message information, interest information is small enough and costs little storage space, so the constrained resources are saved.

Table 1: Definition of symbols

Symbol	Description
i, j	Nodes i, j
C	Community
C_i	Community of node i
$C_{i \& j}$	Common parent community of nodes i and j
$ComSeq(C_i \rightarrow C_j)$	Community sequence
M	Message
SN	Source node
IN	Intermediate node
DN	Destination node
I_{sn}	Interest of source node
I_{in}	Interest of intermediate node
I_{dn}	Interest of destination node

We would like to make some points clear about interest-tree after formal construction. At first, the information of user interest is obtained before the construction of interest-tree, and the dissemination process. Each node locally stores the interest-tree, the interests of its own and the contacted nodes' interests. Besides, each node also locally conducts the calculation about social tie and density as they can meet others with new interests.

Secondly, we intend to integrate the scheme into a conference managing platform, in which the tree is constructed in a centralized way when the server collects the information uploaded by users (attendees). Users do not need to acquire the tree structure in real time. Instead, it is the server's responsibility to push the latest version to clients regularly. This is determined by the conference scenario, in which we just have to guarantee the structure obtained by all the terminals are the same at a certain time point. Hence, there is no need to notify users neither the change of the interest-tree nor the dissemination process, as long as users have some memory space to store a copy of the tree and pay for the traffic consumption of information uploading, interest-tree acquisition and subsequent updates.

Following these points, we prefer to apply a fixed rule to construct the tree and focus more on the advantages of such tree structure on data dissemination. We summarize notations/symbols definition in Table 1 and state the necessary assumptions used through out this paper:

- Each node is responsible for maintaining the same tree structure of interest;
- Nodes follow their regular mobility patterns;
- A forwarding message has the same interest tag with the destination node.

5.2.2. Density of Community

We define density of a community as the number of passing nodes with different interests (or communities). That is, the bigger the density, the more nodes can be met. Providing that individuals have regular mobility patterns, the density information can be used to select forwarders. Specifically, when the nodes move around, they calculate the density of different communities by counting the pass-by nodes. To the same community, different nodes may have different densities about it. For a certain node, the larger density of a community it has, the more nodes it encounters. In inter-community process, density can help in selecting forwarders to deliver the messages.

In social networks, the degree of sociability is widely utilized to predict the number of encounter nodes for a specific node [38]. We redefine the degree as the density of a community in equation (1):

$$Dst_{i,C_m}(t) = n \quad (1)$$

where $Dst_{i,C_m}(t)$ is the number of nodes (n) belonging to C_m node i has contacted over a time period t . When t is multiple of time window T , we apply the exponentially weighted moving average [38] to calculate the community density for the future time slot Δt as shown in equation (2):

$$Dst_{i,C_m}(t + \Delta t) = (1 - \alpha) \times Dst_{i,C_m}(t - \Delta t) \times \gamma^k + \alpha \times Dst_{i,C_m}(t) \quad (2)$$

where α is community density prediction factor. The evaporation factor γ , and the time interval towards the last update k are used in last update in order to weaken the influence of the old density $Dst_{i,C_m}(t - \Delta t)$.

5.2.3. Social Tie

The social tie indicates the strength of social relationships between two nodes. Nodes with more social similarities or contact duration times have strong social ties with one another [39]. If there is a strong social tie between node A and B, it means that the contact probability between them is high. In this situation, when B is the destination of a message, A is an optimal candidate to relay the message to B. In Int-Tree, we use the social tie feature to choose the best forwarder nodes during intra-community forwarding process.

When a node is transmitting a message, the social tie between nodes determines the efficiency. The reason is that social tie is constructed during the contact process, and it indicates the direct contact probability of two nodes which can be used to find the destination node directly. We use $SoTie_{i,j}(t)$ to measure the social tie of nodes i and j during a time period t ,

$$SoTie_{i,j}(t) = CT_{i,j} \quad (3)$$

where $CT_{i,j}$ is the contact times of i and j over time period t .

Similar to the calculation of density of community, when t is multiple of a time window T , we conduct an evaporation process when combining the past and present values of social tie in order to predict the future social tie, as equation (4):

$$SoTie_{i,j}(t + \Delta t) = (1 - \beta) \times SoTie_{i,j}(t - \Delta t) \times \gamma^k + \beta \times SoTie_{i,j}(t) \quad (4)$$

where β is the prediction factor of social tie.

When i and j contact each other, they update the density or social tie parameters of each other. The update has no relation to messages, and only involves i and j . Algorithm 1 presents the pseudocode of update procedures of community density and social tie. We note that this procedure is described from i 's perspective.

5.3. Forwarding Strategy

The forwarding module is the core of Int-Tree. It combines interest-tree, density of community, and social tie to choose the best forwarder nodes. The destination information, such as ID and interest, is included in the message which make it easy to obtain the corresponding information of the destination node. According to the interests of DN , SN and IN , Int-Tree classifies the environment context into inter-community and intra-community parts, and takes some criteria in different conditions to measure them.

Algorithm 1 Pseudocode of updating community density and social tie

```
1: for all  $j$  connected to  $i$  do
2:   //In time period  $t$ :
3:   if  $I_i == I_j$  then
4:     //Update contact information
5:     if  $i$  has no social tie record of  $j$  then
6:       //Initiate  $i$ 's social tie to  $j$ ;
7:        $CT_{i,j} \leftarrow 0$ ;
8:     end if
9:     //Count the contact time of  $j$ ;
10:     $CT_{i,j} \leftarrow CT_{i,j} + 1$ ;
11:  end if
12:  for all  $C \in ComSeq(C_j \rightarrow C_{root})$  do
13:    //Update inter-community contact information
14:    if  $i$  has no density record to  $C$  then
15:      //Initiate  $i$ 's density to  $C$ ;
16:       $Dst_{i,C} \leftarrow 0$ ;
17:    end if
18:    //Count the number of  $SN$ 's contacted nodes in  $C$ ;
19:     $Dst_{i,C} \leftarrow Dst_{i,C} + 1$ ;
20:  end for
21:  //When  $t$  is multiple of  $T$ 
22:  //Compute density and social tie information:
23:  Compute  $Dst(t + \Delta t)$  using Equation (2)
24:  Compute  $SoTie(t + \Delta t)$  using Equation (4)
25: end for
```

- $I_{SN} == I_{DN}$ and $I_{IN} == I_{DN}$. Under this condition, DN , SN , and IN share the same interest, so it is intra-community process. Thus, social tie is utilized to decide the better forwarder. The node with a higher social tie to DN will be selected as the forwarder among SN , and IN . Otherwise SN will stop the process.
- $I_{SN} == I_{DN}$ and $I_{IN} != I_{DN}$. DN and SN still share the same interest, while IN has a different one. Thus, DN and SN are in the same community with IN as an outsider. This suggests we need a node in the same community with DN to perform intra-community forwarding, so IN is not suitable.
- $I_{SN} != I_{DN}$ and $I_{IN} == I_{DN}$. IN and DN have the same interest, while SN is distinct. Thus IN and DN are in the same community, while SN is not. Therefore, the inter-community forwarding suits the situation better. The forwarding strategy will select IN as a forwarder and the message will be forwarded from SN to IN .
- $I_{SN} != I_{DN}$ and $I_{SN} == I_{IN}$. In this situation, SN and IN belong to the same community but not the destination community. It is inter-community forwarding. Int-Tree will utilize the density information to make decision. To be exact, SN and IN compare their densities to all the C s belonging to the community sequence from C_{DN} to $C_{SN\&DN}$ (i.e. $ComSeq(C_{DN} \rightarrow C_{SN\&DN})$). If IN has the larger density to a certain C in the sequence, SN will choose IN to

Algorithm 2 Pseudocode of forwarding strategy

```
1: Given a message  $M$  in the buffer of a node;
2: for all  $IN$  contacting  $SN$  do
3:   if  $IN$  is  $DN$  then
4:      $SN$  delivers  $M$  to  $IN$ ;
5:   else
6:     if  $I_{IN} == I_{DN}$  then
7:       if  $I_{SN} == I_{DN}$  then
8:         if  $SoTie(SN, DN) < SoTie(IN, DN)$  then
9:            $SN$  delivers  $M$  to  $IN$ ;
10:        end if
11:      else
12:         $SN$  delivers  $M$  to  $IN$ ;
13:      end if
14:    else
15:      if  $I_{SN} != I_{DN}$  then
16:        for all  $C \in ComSeq(C_{DN} \rightarrow C_{SN\&DN})$  do
17:          if  $Dst(SN, C) < Dst(IN, C)$  then
18:             $SN$  delivers  $M$  to  $IN$ ;
19:          break
20:        end if
21:      end for
22:    end if
23:  end if
24: end if
25: end for
```

forward; Otherwise, SN will keep the message.

- $I_{SN} != I_{DN}$, $I_{IN} != I_{DN}$ and $I_{SN} != I_{IN}$. SN , IN and DN share no common interests at all, so they are in different communities (inter-community). Thus, Int-Tree performs the same procedure as the last condition to choose a forwarder.

When the destination node receives a message, it broadcasts a response message to all the nodes that have the message in order to help them to drop the message. Algorithm 2 presents the pseudocode of the forwarding strategy. During the forwarding process, the concept of SN and IN are not related to specific nodes. That is, the source node, say IN , can be a relay node, say SN , in the future.

5.4. Message Scheduling and Buffer Management

Message scheduling decides the order of transmitting the messages between nodes. It ensures that the messages with higher forwarding opportunities have higher priority for forwarding. On the other hand, buffer management module decides which messages should be discarded when the buffer is overloaded. The message scheduling and buffer replacement share similar principles, as they both require excluding or discarding the expired or successfully delivered messages without influencing those messages which are transmitting. We suggest readers who are interested to refer to [22] for more details about these components.

6. Performance Evaluation

6.1. Dataset

The data set adopted in our work was collected during ACM SIGCOMM 2009 [40], available on the website of CRAWDAD (<http://www.crawdad.org/thlab/sigcomm2009/>). It includes a list of the participants' original interests. The dataset contains the activity periods of each participant and device, as well as the Bluetooth device discovery logs of each user. Therefore, we can get the record of all the meetings between all the participants for simulation experiments, instead of applying any synthetic movement model.

6.2. Interest-Tree Construction

The collected data set during ACM SIGCOMM 2009 includes the tested attendees' interest information and contact information. To be specific, there are 76 attendees (nodes) and 711 interests. We numbered the nodes from No. 0 to No. 75 and the interests from No. 1 to No. 711. However, we found out that some interests involve the same nodes. For example, both Interests No. 79 and No. 82 involve Nodes No. 8 and No. 62. To simplify the data set process, we merge the interests which involve the same nodes into one interest, deriving 82 interests, numbered from 1 to 82.

Because we have decided to use nodes' *major interests* to conduct research, it is necessary to find a node's *major interest*. Hereby, we denote node i 's *major interest* as the interest with the largest number in i 's interest list. In this sense, it is appropriate to regard this as a procedure seeking a node's most distinguished interest. After the process, all the nodes's major interests cover 49 of the 82 interests, meaning that some nodes share the same one. Then, we find parent interest for each interest from Interest No. 82 in descending order. If the involved nodes of Interest No. m contain entirely those of Interest No. n ($m \in \{n-1, n-2, \dots, 1\}$), No. m is the parent interest of No. n . For Interests No. 1 (involving all the nodes except for Node No. 44) and No. 60 (involving only Node No. 44), we create Interest No. 0 as their parent interest. Finally, an unbalanced 7-layer multi-tree is generated, with 1 root interest, 35 parent interests, and 47 leaf interests. Table 2 shows the statistics in regard of the tree.

While constructing interest-tree, we merged the interests involving the same nodes, and chose I_i manually. These actions do not fully reflect the reality and may induce biased experiment results. However, these are inevitable due to the missing information of users' real interests in the data set. The incompleteness of the data set is probably the result of concerns about information security or privacy issues. Therefore, the experiment result here is compromised regarding the above issues.

6.3. Simulation Setup

Our experiments are carried out using the Opportunistic Network Environment (ONE) simulator [41]. We chose the Epidemic routing and PROPHET algorithm as the benchmark DTN routing protocols (please see Section 2 for detailed reasons). We run our simulation for the three protocols with different simulation time (21600s~129600s), buffer sizes (5MB~30MB),

and message TTL (60min~360min). Default value of parameters are 21600s, 5MB and 360min, respectively. The wireless transmission applies Bluetooth interface with a 10-meter communication range and 2-Mbps transmission speed. The reasons we use Bluetooth interface include: i) it consumes the least energy compared to Wi-Fi or 3G technology; and ii) the transmission range of Bluetooth is enough for our conference scenario where the possible forwarders move in a small area [42]. Events are generated each 25~35 seconds. The simulation parameters are summarized in Table 3.

In each experiment, we compare the performance of protocols based on the following metrics:

Message Delivery Ratio: the ratio of successfully delivered messages to the total number of unique messages created (excluding redundant messages) in a given period.

Overhead: the ratio of relayed messages (delivered messages excluded) and delivered messages.

Average Latency: the average time between the time a message is generated and the time it is delivered successfully.

Average Hop Count: the average hop-counts when messages are received successfully.

6.4. Preliminary Results

Fig. 5 compares the performance of the three algorithms over different buffer sizes. When the size of buffer is risen, both PROPHET and Epidemic show clearly increasing trend for delivery ratio and average latency, and decreasing trend for overhead and hop-count. However, Int-Tree vibrates little for all the parameters. Comparatively, it can be seen that Int-Tree has the best performance in this situation.

Specifically, when the buffer size is set to 20MB (see Fig. 5(a)), Int-Tree forwards 59.32% messages while the delivery ratios of PROPHET and Epidemic are 10.04% and 12.63%, respectively. As shown in Fig. 5(b), the overhead of Int-Tree remains lower than 60, which is much smaller than the other two protocols where their overhead is higher than 500. Comparing the hop count of the protocols in Fig. 5(c), the average value of Int-Tree is about 1.3423, slightly lower than that of PROPHET (1.3449), while the average hop count of Epidemic vibrates dramatically between 3.29 to 5.19. Fig. 5(d) shows a trend that Int-Tree spends more time than PROPHET when buffer size is smaller than 20 MB, but after the 20-MB point, Int-Tree consumes the least time. On the contrary, Epidemic performs the worst for using more than 1500s to disseminate messages. This is because Epidemic generates many copies of message, causing more buffer operation.

For the stable performance of Int-Tree, we believe it is a result of the small amount of information it deals with. In the dataset, both mobile nodes and user interests are in small scale. Therefore, it requires little buffer for both interest-tree and computation. This inspires us to do more experiments with smaller buffer sizes in the future.

Fig. 6 compares the performance of the protocols with different values of TTL. According to the figure, it can be seen that performance of almost all the protocols are stable when TTL value of messages changes, except that the Epidemic is more

Table 2: Statistics in Regard of the Interest-Tree

Layer No.	Involved Interest No.	Sum (Interest)	Involved Node No.	Sum (Node)
1	0	1	No nodes involved.	0
2	1 60	2	2 3 4 5 6 7 9 14 15 16 18 20 25 27 28 29 31 32 36 40 41 42 44 52 57 67 68 71	28
3	2 3 4 5 6 8 9 10 13 28 30 31 34 37 39 40 59 78 79	19	0 21 43 51 55 61 69 70	8
4	7 11 12 14 15 16 17 18 19 22 23 26 27 35 36 38 41 43 45 46 49 52 53 54 55 56 58 61 62 68 70 71 74 76 77 80	36	1 10 11 13 17 23 30 33 34 35 37 39 45 46 54 58 59 63 65 66 72 74	22
5	20 21 24 25 29 32 47 48 51 63 64 65 66 67 75 81 82	17	19 22 26 47 48 49 50 53 64 73 75	11
6	33 42 44 50 69 72 73	7	8 12 24 56 60 62	6
7	57	1	38	1

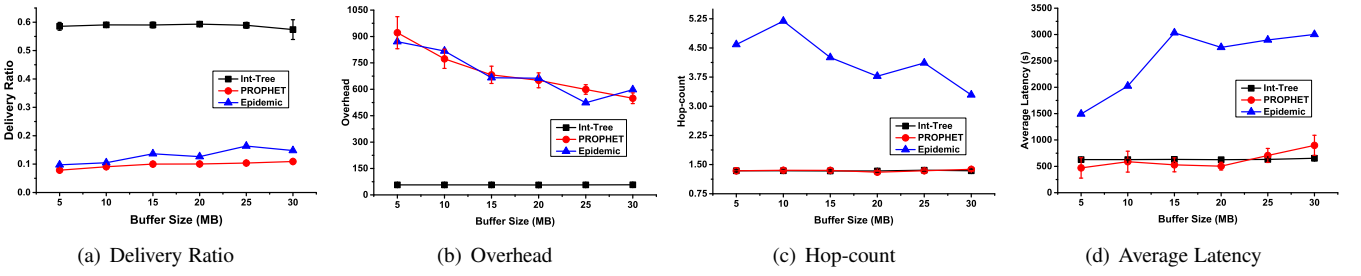


Figure 5: Performance with respect to buffer size.

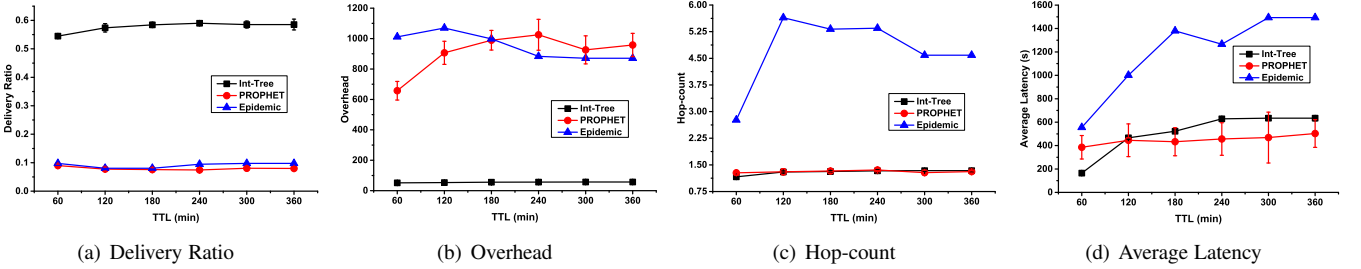


Figure 6: Performance with respect to TTL.

Table 3: Simulation parameters

Simulation Parameters	Values	Default Value
Simulation time	21600~129600 s	216000 s
Number of runs	10	-
Warm up time	1000s	-
Time window T	30s	-
Network Area	$80 \times 80 \text{ m}^2$	-
Wait time at destination	0~120 s	Random
Event interval	25~35 s	Random
Message size	500~1024 KB	Random
Message TTL	60~360 min	360 min
Node buffer size	5~30 MB	5 MB
α	0.7	-
β	0.1	-
γ	0.9	-

sensitive with respect to hop-count and average latency. In addition, Int-Tree has the best performance with a big advantage in terms of delivery ratio and overhead. For hop-count, Int-Tree remains slightly better than PROPHET. As for average latency (see Fig. 6(d)), Int-Tree spends the least time (164.38s) with 60-min TTL and then rises to higher values than PROPHET. However, the highest value of Int-Tree is 634.31s, 26.1% higher than PROPHET, 57.5% lower than Epidemic. Furthermore, the figure suggests a good trend that Int-Tree is getting stable and the gap between Int-Tree and PROPHET is narrowing.

Fig. 7 describes the simulation results under different simulation durations. All the evaluated protocols have experienced similar trend with Fig. 6, except that: i) both PROPHET and Epidemic keep rising on overhead; ii) it takes less hop-count to disseminate messages for Epidemic when simulation duration rises; and iii) it costs Int-Tree much more time than the others when simulation duration is larger than or equal to 64800s.

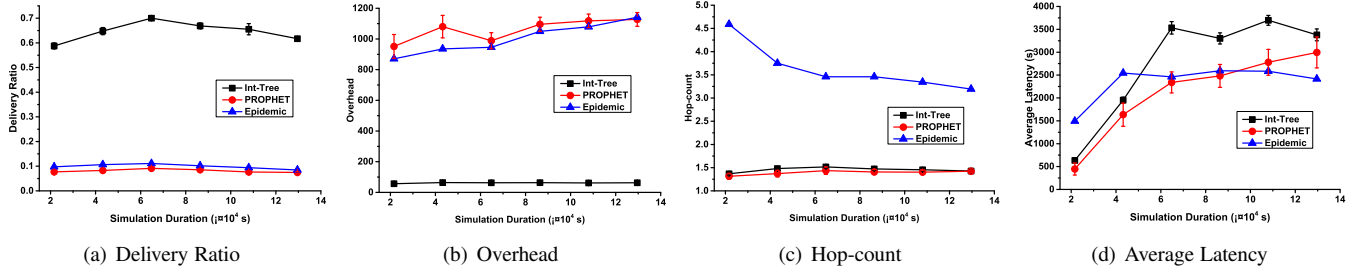


Figure 7: Performance with respect to simulation duration.

The results of our simulations with different values of evaluation parameters show that Int-Tree disseminates data with the highest delivery ratio, the lowest overhead and the minimum hop count. This attributes to that Int-Tree adopts interest-tree, social tie and community density to help make decision on choosing forwarders. Int-Tree deduces unnecessary message copies and transmissions in network which results in low overhead and small hop counts. It also chooses a best forwarder node effectively which increases the delivery ratio. As for the relatively larger latency values, we should notice that it is caused by consuming time to find a proper forwarder. However, we believe this deficiency is tolerable considering that the hop count in our algorithm is less than 1.5 in average.

6.5. Further Experiments

We conduct two groups of simulations on exploring how α , β and γ influence the performance of Int-Tree. α and β are both prediction factors, but for different information. Therefore, we design 25 groups of (α, β) pairs to examine their influence. The bigger they are, the more important the present information is. γ reduces the influence of the history record. The smaller it is, the weaker the history information is. We run the simulations 10 times and other parameters are set default except for simulation duration (21600s). The evaluation criteria remain the same.

Initially, we expected that α and β should have similar impact on Int-Tree. But the result proved a different picture. α does affect the performance of Int-Tree, but no matter how β changes, Int-Tree performs the same. *This might be the result of the scarcity of dataset. Nodes belonging to the same community are in small number, making the social tie values between nodes small and constant during the experiment.* Fig. 8 illustrates the result regarding changing α . The black straight lines represent the corresponding average values. From the figure we can see that the values for the metrics are stable and change in small ranges. This suggests that α will not affect Int-Tree significantly. Moreover, we find that $\alpha = 0.9$ is the best for Int-Tree because it achieves the second highest delivery ratio, the lowest overhead, the third least hop-count, and the second smallest latency. In other words, a large proportion of present information will help Int-Tree to perform.

Fig. 9 shows the performance of Int-Tree over γ . With γ raising, all the evaluated parameters change dramatically, although the ranges are actually very small. Nevertheless, we can still acquire some knowledge from the figure. Generally, delivery ratio experiences the opposite trend from overhead and

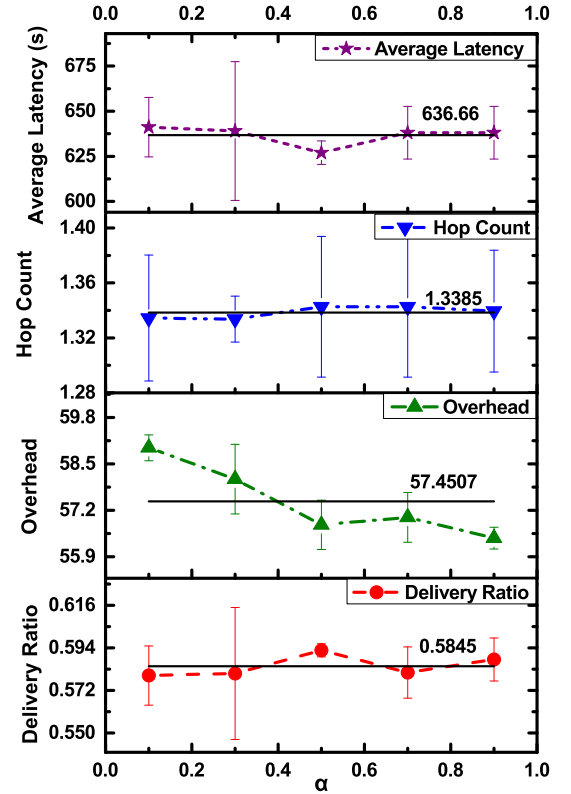


Figure 8: Performance with respect to α .

average latency. This is reasonable because we expect high delivery ratio but low overhead and latency. Moreover, overhead and average latency share the feature of being stable or converging. Hop count shows totally different status with unstable values. Therefore, we need a trade-off to find the best value of γ for Int-Tree and this leads us to 0.3. With this value, Int-Tree achieves the highest delivery ratio, the lowest latency, and average performance on overhead and hop-count. In the meantime, it supports the conclusion reached from Fig. 8 that we need to weaken the influence of the history record.

7. Discussions on User Interests

As a matter of fact, we considered multi-interest as a relationship between user interests at the beginning of our research.

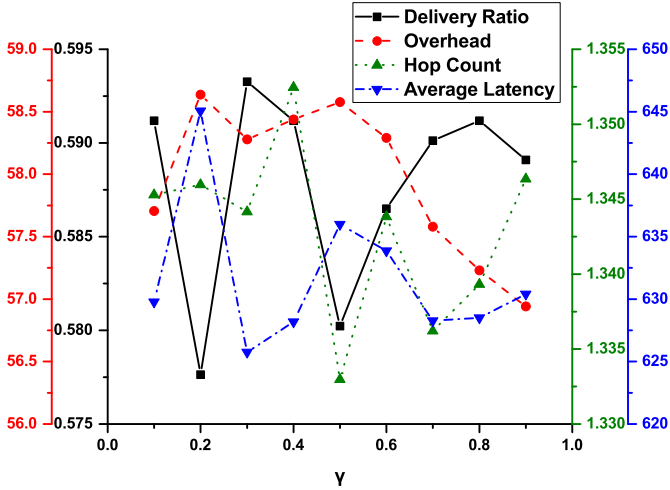


Figure 9: Performance with respect to γ .

However, we found that it is more of a property of mobile users than of a relationship between interests. This property will make the problem more complicated and the scheme needs improvement in many aspects. Nonetheless, we would like to present some potential issues in regarding of multiple interests and put forward some solutions.

- *Will the interest-tree change?* Interest-tree is the base to study the relationships between user interests. Introducing multi-interest will not affect the construction rule, the depth and the value of interest-tree. It will only expand the width. The reason behind this is that the emerged interests include the same nodes. To be specific, interest-tree remains 7 layers after emergence, but the width decreases after emergence, proved by the declined number of interests involved in Layers 2-7 shown in Table 4. Additionally, we ran the experiment with full interest information (i.e., without emergence), and found that the result do not change much, compared to the that with emerged interests. We do not provide the result because Int-Tree needs modification to fully apply to multi-interest situation, and it needs further exploitation on the change of its performance.
- *How to collect users' interest?* As stated in Section 6.2, the ACM SIGCOMM 2009 dataset did not provide the users' real interest information. While in real life, we should pay attention to the questions concerning interest information collection. For example, will the users provide their interest information, how to provide, or even where exactly to put a specific interest in the tree? In fact, we plan to plant the scheme into a conference platform, in which, there is a module for collecting user information at attendee's will. Users can choose their interests from some drop-down menus. The developers will select an construction algorithm to build the tree, ensuring that the structure is stable, flexible and accessible. The algorithm should be able to distinguish the interests in context, to categorize them and

Table 4: The number of interests involved each layer of interest-tree before and after emergence

Layer No.	1	2	3	4	5	6	7
Before emergence	1	4	30	407	116	142	12
After emergence	1	2	19	36	17	7	1

to construct the tree. Moreover, We need to expand ACM CCS structure to a broader scope than computer science. Therefore, techniques in context recognition, or machine learning, or even some inter-discipline ones are required to address this problem.

- *Will Int-Tree change?* Int-Tree should be changed in both updating social information and choosing forwarders. A preliminary solution is to allow nodes i to check its interest list and update the information for each of its interest. When choosing forwarders, SN needs to check both its shared interests with IN and the distinct interests from IN . Furthermore, a special case draws our attention. Node A is interested in a parent interest i , but node B is in (all of) i 's leaf interest(s). Hence, how to deal with this case? Should it be treated as A and B share the same interest? From the perspective of interest-tree, it seems their interests are the same. But from the nodes' perspective, it might be different because of their definitions on the research areas.

8. Conclusion

In this paper, we explored the impact of user interests and the relations between them to improve the efficiency of data dissemination in socially aware networking. Specifically, we focused on interest inclusion, one of common relation types between user interests. Inspired by ACM Computing Classification System, we proposed interest-tree to construct our scheme, named Int-Tree. Using this strategy, Int-Tree can identify communities according to interests. Furthermore, Int-Tree takes advantage of density of community and social tie for effective information dissemination and strong adaptability to the dynamic networking environment. The simulation with real dataset proved that Int-Tree outperforms the benchmark protocols of PROPHET and Epidemic with higher delivery ratio, less overhead and less hop counts. The major drawback of Int-Tree lies in that its performance on average latency is not the best over simulation duration or on the circumstances where $Buffer\ Size \leq 20MB$ or $TTL > 120min$. Moreover, further simulations on the performance of Int-Tree with changing parameters suggested that weakening the effect of history information can cause a good performance, although α , β and γ do not make dramatic difference.

As a first step on studying relations between user interests, we did not provide experimental evidence to study how multi-interest can affect the dissemination, making the results unsubstantial. However, we put forward some potential challenges and solutions. Meanwhile, in another academic paper, a research has been conducted considering the situation where

nodes have multiple interests, accompanied with the intersection of interests. Nevertheless, there are still some problems remained, such as the effect of cross-layer interests and other possible relations.

Acknowledgment

This work is partially supported by the Fundamental Research Funds for the Central Universities (DUT15YQ112) and National Natural Science Foundation of China (61572106).

References

- [1] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks, *IEEE Communications Magazine* 44 (2006) 134–141.
- [2] F. Warthman, Delay-tolerant Networks (DTNs) - A Tutorial, Technical Report, Warthman Associates, 2008.
- [3] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, A. Vasilakos, Routing for disruption tolerant networks: Taxonomy and design, *Wireless Networks* 16 (2010) 2349–2370.
- [4] F. Xia, L. Liu, J. Li, J. Ma, A. V. Vasilakos, Socially aware networking: A survey, *IEEE Systems Journal* 9 (2015) 904–921.
- [5] N. Vastardis, K. Yang, Multi-phase socially-aware routing in distributed mobile social networks, in: Proceedings of 2013 9th International Wireless Communications and Mobile Computing Conference (IEEE IWCMC 2013), Cagliari, Sardinia Italy, 2013, pp. 1353–1358.
- [6] Y. Zhu, B. Xu, X. Shi, Y. Wang, A survey of social-based routing in delay tolerant networks: Positive and negative social effects, *IEEE Communications and Surveys Tutorials* 15 (2013) 387–401.
- [7] F. Xia, A. M. Ahmed, L. T. Yang, J. Ma, J. Rodrigues, Exploiting social relationship to enable efficient replica allocation in ad-hoc social networks, *IEEE Transactions on Parallel and Distributed Systems* 25 (2014) 3167–3176.
- [8] F. Xia, A. M. Ahmed, L. T. Yang, Z. Luo, Community-based event dissemination with optimal load balancing, *IEEE Transactions on Computers* 64 (2015) 1857–1869.
- [9] S. Wasserman, K. Faust, Social network analysis: methods and applications, structural analysis in the social sciences series, Cambridge Univ. Press, 1994.
- [10] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, Y. Sun, Geocommunity-based broadcasting for data dissemination in mobile social networks, *IEEE Transactions on Parallel and Distributed Systems* 24 (2013) 734–743.
- [11] M. Xiao, J. Wu, L. Huang, Community-aware opportunistic routing in mobile social networks, *IEEE Transactions on Computers* 63 (2014) 1682–1695.
- [12] J. Wu, M. Xiao, L. Huang, Homing spread: Community home-based multi-copy routing in mobile social networks, in: Proceedings of the 32nd IEEE International Conference on Computer Communications (IEEE INFOCOM 2013), 2013, pp. 2319–2327.
- [13] M. McPherson, L. Smith-Lovin, J. M. Cook, Birds of a feather: Homophily in social networks, *Annual Review of Sociology* 27 (2001) 415–444.
- [14] S. Hidi, Interest: A unique motivational variable, *Educational Research Review* 1 (2006) 69–82.
- [15] T. Zhou, H. A. T. Kiet, B. J. Kim, B.-H. Wang, P. Holme, Role of activity in human dynamics, *EPL (Europhysics Letters)* 82 (2008) 28002.
- [16] X.-P. Han, T. Zhou, B.-H. Wang, Modeling human dynamics with adaptive interest, *New Journal of Physics* 10 (2008) 073010.
- [17] M.-S. Shang, G.-X. Chen, S.-X. Dai, B.-H. Wang, T. Zhou, Interest-driven model for human dynamics, *Chinese Physics Letters* 27 (2006) 048701.
- [18] K. Chen, H. Shen, H. Zhang, Leveraging social networks for p2p content-based file sharing in disconnected manets, *IEEE Transactions on Mobile Computing* 13 (2014) 235–249.
- [19] P. Costa, C. Mascolo, M. Musolesi, G. P. Picco, Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks, *IEEE Journal on Selected Areas in Communications Special issue on Delay-Tolerant Networks* 26 (2008) 748–760.
- [20] Y. Zhu, J. Wang, C. Wang, Ripple: A publish/subscribe service for multidata item updates propagation in the cloud, *Journal of Network and Computer Applications* 34 (2011) 1054–1067.
- [21] M. Chuah, P. Yang, P. Hui, Cooperative user centric information dissemination in human content-based networks, in: Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems (IEEE IC-PADS 2010), Nagoya, Japan, 2010, pp. 794–799.
- [22] F. Xia, L. Liu, J. Li, A. M. Ahmed, L. T. Yang, J. Ma, Beeinfo: An interest-based forwarding scheme using artificial bee colony for socially-aware networking, *IEEE Transactions on Vehicular Technology* 64 (2015) 1188–1200.
- [23] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Technical Report, Technical Report CS-200006, Duke University, 2000.
- [24] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, in: P. Dini, P. Lorenz, J. N. de Souza (Eds.), *Service Assurance with Partial and Intermittent Resources*, volume 3126 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2004, pp. 239–254.
- [25] Q. Yan, L. Wu, L. Yi, Modeling of posting behavior in mobile internet based on human dynamics, in: Proceedings of 2012 Second International Conference Cloud and Green Computing (IEEE CGC 2012), Hunan, China, 2012, pp. 744–747.
- [26] G. Carofiglio, M. Gallo, L. Muscariello, Icp: Design and evaluation of an interest control protocol for content-centric networking, in: Proceedings of the 31st Annual IEEE International Conference on Computer Communications Workshops (IEEE INFOCOM 2012), Orlando, Florida USA, 2012, pp. 304–309.
- [27] F. Bjurefors, P. Gunningberg, E. Nordström, C. Rohner, Interest dissemination in a searchable data-centric opportunistic network, in: Proceedings of 2010 European Wireless Conference, Lucca, Tuscany Italy, 2010, pp. 889–895.
- [28] W. Gao, G. Cao, User-centric data dissemination in disruption tolerant networks, in: Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011), Shanghai, China, 2011, pp. 3119–3127.
- [29] J. Wu, Y. Wang, Hypercube-based multipath social feature routing in human contact networks, *IEEE Transactions on Computers* 63 (2014) 383–396.
- [30] P. Hui, J. Crowcroft, How small labels create big improvements, in: Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (IEEE PerCom 2007), New York, USA, 2007, pp. 65–70.
- [31] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: Social -based forwarding in delay tolerant networks, in: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc 2008), Hong Kong, 2009, pp. 1–9.
- [32] F. Li, J. Wu, Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks, in: Proceedings of IEEE International Conference on Sensor, Mesh and Ad Hoc Communications and Networks (IEEE SECON 2009), Rome, Italy, 2009, pp. 1–9.
- [33] M. Musolesi, P. Hui, C. Mascolo, J. Crowcroft, Writing on the clean slate: Implementing a socially-aware protocol in haggles, in: Proceedings of IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (IEEE WoWMoM 2008), Newport Beach, USA, 2008, pp. 1–6.
- [34] M. Musolesi, S. Hailes, C. Mascolo, Adaptive routing for intermittently connected mobile ad hoc networks, in: Proceedings of IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (IEEE WoWMoM 2005), Taormina - Giardini Naxos, Italy, 2005, pp. 183–189.
- [35] D. Katsaros, N. Dimokas, L. Tassioulas, Social network analysis concepts in the design of wireless ad hoc network protocols, *IEEE Network* 24 (2010) 23–29.
- [36] A. Fast, D. Jensen, B. N. Levine, Creating social networks to improve peer-to-peer networking, in: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (ACM KDD 2005), Chicago, USA, 2005, pp. 568–573.
- [37] P. Ruiz, B. Dorronsoro, P. Bouvry, L. Tardón, Information dissemination in vanets based upon a tree topology, *Ad Hoc Networks* 10 (2012) 111–127.

- [38] R. Jain, K. K. Ramakrishnan, D. M. Chiu, GrAnt: Inferring Best Forwarders from Complex Networks' Dynamics through a Greedy Ant Colony Optimization, Technical Report RR-7694, INRIA, 2011.
- [39] S. Ioannidis, A. Chaintreau, On the strength of weak ties in mobile social networks, in: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems (ACM SNS 2009), Nuremberg, Germany, 2009, pp. 19–25.
- [40] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, C. Diot, Mobilique: Middleware for mobile social networking, in: Proceedings of the 2nd ACM Workshop on Online Social Networks (ACM WOSN 2009), Barcelona, Spain, 2009, pp. 49–54.
- [41] A. Keränen, T. Kärkkäinen, J. Ott, Simulating mobility and dns with the one (invited paper), Journal of Communications 5 (2010) 92–105.
- [42] K.-T. Chuang, Y.-J. Lin, C.-C. Chen, Bluetooth-based mobile p2p framework for preference-aware data dissemination on social networks, in: Proceedings of 2013 IEEE 14th International Conference on Mobile Data Management (IEEE MDM 2013), volume 2, Milan, Italy, 2013, pp. 110–115.