

# BEEINFO: Data Forwarding Based on Interest and Swarm Intelligence for Socially-aware Networking

Jie Li, Li Liu, Feng Xia  
School of Software, Dalian University of Technology  
Dalian 116620, China  
f.xia@acm.org

## ABSTRACT

Socially-aware networking provides a promising paradigm for data forwarding by exploiting the involved nodes' social properties in mobile social networks. However, individuals' learning capability and awareness to the dynamic environment have not been well explored in the literature. In this paper, we give a brief introduction of an interest-based scheme called BEEINFO. Inspired by swarm intelligence, BEEINFO takes advantage of individuals' perceiving and learning capability to gather information of density and social tie during communication. Moreover, it classifies communities based on nodes' interests and distinguishes data forwarding into situations of inter-community and intra-community. Furthermore, BEEINFO performs efficient message scheduling and buffer management to improve performance. The simulation results show that BEEINFO outperforms PROPHET and Epidemic in message delivery ratio, overhead and hop count, except for average latency.

## Categories and Subject Descriptors

C.2.2 [Computer-Communication Networks]: Network Protocols; H.1.2 [Information Systems]: User/Machine Systems—*human factors*

## Keywords

Socially-aware networking; Forwarding; Interest; Swarm intelligence

## 1. INTRODUCTION

Socially-Aware Networking (SAN) is an emerging paradigm to solve problems of networks consisting of mobile nodes with social properties, e.g. social relationship, and mobility patterns. These characteristics can be utilized to design efficient data forwarding protocols in e.g. mobile social networks.

In particular, social relations of mobile nodes have generally long-term features and they are more stable than node

mobility patterns. *Community, interest, social tie*, etc. are the commonly used social concepts in SAN. For instance, people are usually interested in some files or data in specified categories. Moreover, people with the same interest tend to meet frequently, bonding each other with social tie and forming a community. Additionally, individuals' regular mobility pattern is important to predict the future mobility. Many experiments [1, 7, 8] proved the high efficiency of socially-aware methods, involving the above properties. However, SAN is faced with some challenges, including how to conduct community construction and detection, predict future encounters, and calculate social relationship efficiently. These are correlated with some facts, such as the changing network topology, the constrained resources, the threat of potential congestion caused by information exchange.

These challenges are due to the lack of adaptability to dynamic environments. It is essential to find an appropriate solution. Fortunately, swarm intelligence has provided some clues to detect and adapt to the changes of environment context in a timely fashion. For example, Artificial Bee Colony (ABC) algorithm [2] imitates the bees' behaviors to search for nectar, as well as their ability to detect the changing density of nectar sources.

In our work, BEEINFO is designed as an artificial BEE colony inspired Interest-based FOrwarding scheme under the framework of SAN. BEEINFO takes advantage of mobile users' social properties, mobility regularities and learning capability to detect the dynamic environment, including density and social tie. Additionally, it classifies the community based on personal interests, eliminating the cost by community detection and construction. Furthermore, the interest information is small enough to predict density and social tie, saving resources, e.g. buffer and energy. All these features contribute to the novelty of BEEINFO.

## 2. INSPIRATION FROM ARTIFICIAL BEE COLONY

ABC algorithm imitates the natural bees' behavior of searching for nectar sources. In the process, the scout bees search in a random direction and gather information simultaneously. Once the best sites are spotted, some of the scout bees are assigned as follower bees to work on the sites. With the back and forth, the bees evaluate and maintain the density of left nectar source for the future detection.

The process reveals bees' awareness and learning capability, which is of crucial use to improve individuals' adaptability to dynamic environments. This further leads to the following inspirations when we design BEEINFO. 1) We de-

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).

MobiCom '13, September 30–October 4, 2013, Miami, FL, USA.

ACM 978-1-4503-1999-7/13/09.

<http://dx.doi.org/10.1145/2500423.2504569>.

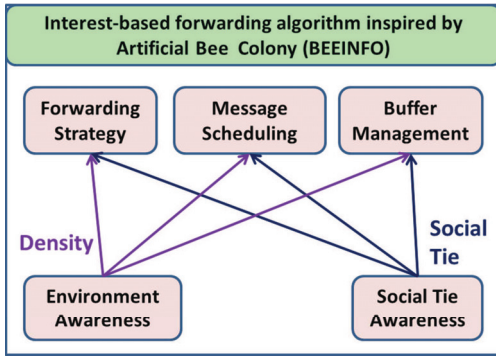


Figure 1: Components of BEEINFO.

fine *density* to identify the number of passing nodes with different interests. The bigger the density, the more nodes can be met. 2) *Social tie* should be maintained by nodes and it denotes the strength of social relationship between nodes with the same interest. 3) Nodes update the information of density and social tie when being in contact.

### 3. DESIGN OF BEEINFO

#### 3.1 Overview

Fig. 1 depicts the five components of BEEINFO: *environment awareness*, *social tie awareness*, *forwarding strategy*, *message scheduling*, and *buffer management*. Environment awareness and social tie awareness provide density and social tie information separately, which are essential to make decisions in forwarding, scheduling message, and managing buffer. This structure stands on the following assumptions: 1) each node maintains its one and only interest, 2) nodes with the same interest belong to the same community, and 3) nodes follow their regular mobility patterns.

#### 3.2 Environment and Social Tie Awareness

In BEEINFO, environment awareness and social tie awareness gather density and social tie information, respectively. To be specific, density is the number of nodes a mobile user encounters over a time window  $T$ . In addition, considering the influence of the past and present information, BEEINFO predicts density based on the exponentially weighted moving average, similar to [3]. After that, an evaporation process is performed to weaken the influence of previous density.

$$Density_i(t) = \sum_{t=0}^T n \quad (1)$$

$$Density_i(t + \Delta t) = \alpha \times Density_i(t - \Delta t) + (1 - \alpha) \times Density_i(t) \quad (2)$$

$$Density_{new} = Density_{old} \times \gamma^k \quad (3)$$

Eqs. (1), (2) and (3) describe the computational formulation of density, where  $\alpha$  is community density prediction factor,  $\gamma$  is evaporation factor and  $k$  is the time interval towards last update.

We use similar method to measure, predict and evaporate social tie, as shown in (4), (5) and (6). In (4),  $\lambda_{i,j}$  is the contact frequency and  $d_{i,j}$  is the contact duration of  $i$  and

$j$  in time window  $T$ .  $\beta$  in (5) is social tie prediction factor.  $\gamma$  and  $k$  keep the same.

$$SoTie_{i,j}(t) = (\lambda_{i,j} \times d_{i,j}(t))/T \quad (4)$$

$$SoTie_{i,j}(t + \Delta t) = \beta \times SoTie_{i,j}(t - \Delta t) + (1 - \beta) \times SoTie_{i,j}(t) \quad (5)$$

$$SoTie_{new} = SoTie_{old} \times \gamma^k \quad (6)$$

#### 3.3 Forwarding Strategy

Forwarding strategy is the core of BEEINFO. The information of destination node (DN), including ID and interest, is contained in the message. When a source node (SN) meets with an intermediate node (IN), firstly, they compare their interests and see if they belong to the same community, and then compare density or social tie information to decide whether IN is a good forwarder. Specifically, density is for inter-community forwarding and social tie is for intra-community process. Note that when making a decision, BEEINFO follows the rule that intra-community forwarding is prior to inter-community process. After comparison, our scheme takes different actions to deal with distinct situations, as described below.

(1) If DN, SN and IN belong to the same community, it is intra-community forwarding. The node with the higher social tie will be selected as the forwarder among SN and IN.

(2) DN and SN are in the same community, but IN is an outside node. In this situation, IN is not a suitable forwarder and BEEINFO needs a node in the same community with DN for intra-community forwarding.

(3) When IN and DN belong to the same community but SN is an outside node, IN will be chosen as a forwarder and the message will be transmitted from SN to IN.

(4) SN and IN belong to the same community but not the destination community where DN belongs. It is inter-community forwarding. The node with the higher density to DN will be chosen as forwarder among SN and IN.

(5) If SN, IN and DN are in three communities, it satisfies the inter-community forwarding scenario. BEEINFO will perform the same action as in (4).

#### 3.4 Message Scheduling and Management

Message scheduling and buffer management share the same rule with Forwarding Strategy. Besides, they both exclude or discard the expired or successfully delivered messages without influencing those messages under transmission. The difference between them lies in the priority of impact factors.

In message scheduling, intra-community messages are processed first. Then those with higher social tie (or density for inter-community) are transmitted, followed by the newer ones. Buffer replacement has the reverse order, meaning that the inter-community messages are replaced first, and then those with lower density (or social tie for intra-community) and finally it is the older messages being replaced.

### 4. EVALUATION AND DISCUSSION

We have conducted preliminary simulations on BEEINFO, PROPHET [5], and Epidemic [6]. The simulations were conducted with the Opportunistic Network Environment Simulator [4] in Points of Interest (POIs) movement

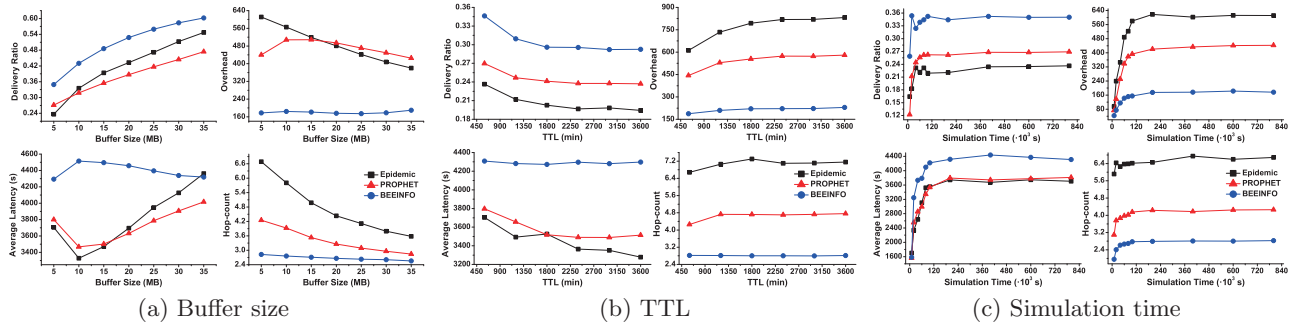


Figure 2: Performance with respect to buffer size, TTL and simulation time.

model. Basically, there are four POI groups and each group consists of 40 nodes with same interest, indicating one community. In each POI scenario, there is a high probability for a node to meet others in the same community and a low possibility to move to other POIs. Additionally, we set a POI5 group to represent the place that all the nodes may reach with the same probability.

The protocols' performance is evaluated over various buffer sizes (5~35 MB), TTLs (600~3600 min) and simulation times (10000~800000 s). Nodes move with speeds from 0.5 m/s to 1.5 m/s in the area of  $8000 \times 7800 m^2$ . The message size varies in the range of 500~1024 kB. To calculate density and social tie,  $\alpha$  and  $\beta$  are both assigned 0.3, and  $\gamma$  is 0.98. Other parameters are set with default values and not listed here given the limited space.

The metrics to evaluate the protocols are message delivery ratio, overhead, average latency and hop count. Overhead is the ratio of relayed messages (delivered messages excluded) and delivered messages.

We ran the simulation 10 times and calculated the average values. The results in Fig. 2 display that generally BEEINFO keeps the same trends with PROPHET and Epidemic, in terms of delivery ratio, overhead and hop count. Specifically, BEEINFO achieves the highest delivery ratio, with the changing parameters. The overhead of BEEINFO is far lower than those of PROPHET and Epidemic, and keeps less than 160. Besides, it takes less than 3 hops to successfully deliver messages for BEEINFO, which is also the best among the protocols. The data also show that BEEINFO is not as sensitive as PROPHET and Epidemic to buffer size, TTL, and simulation time concerning overhead and hop count. The results of BEEINFO are stable, but those of the other protocols fluctuate in some occasions, such as the overhead of PROPHET with buffer size and the hop count of Epidemic with TTL.

Nonetheless, BEEINFO experiences the highest average latency. This is caused by the fact that it does not transmit messages until it chooses a proper forwarder and it costs time. However, Fig. 2(a) shows an encouraging phenomenon, which is BEEINFO's latency declines after reaching the peak about 4496s, when buffer size increases. The reason is that BEEINFO controls message redundancy more effectively, with message scheduling and buffer management. Besides, the latency of BEEINFO is also more stable than PROPHET and Epidemic, especially when TTL and simulation time are considered.

The conclusions we reach from the experiments are: 1) BEEINFO outperforms PROPHET and Epidemic in deliv-

ery ratio, overhead and hop count, except for average latency; 2) TTL and simulation time have the least effect on BEEINFO.

Our research on BEEINFO provides a foundation for upper layer application design, such as automatic information sharing in an academic conference. However, we need to explore more on what causes the high latency and how to reduce it. Besides, there are more fields we need to concentrate on as well, including 1) how message copies affect the performance, 2) if interest and the relation of interests can make a difference or not, and 3) how the protocols perform under the circumstances of real traces/datasets.

## 5. REFERENCES

- [1] K. Chen, H. Shen, and H. Zhang. Leveraging social networks for p2p content-based file sharing in disconnected manets. *IEEE Trans. Mob. Comput.*, PrePrint, December 2012.
- [2] S. K. Dhurandher, S. Misra, P. Pruthi, S. Singhal, S. Aggarwal, and I. Woungang. Using bee algorithm for peer-to-peer file searching in mobile ad hoc networks. *J. Netw. Comput. Appl.*, 34(5):1498–1508, September 2011.
- [3] R. Jain, K. K. Ramakrishnan, and D. M. Chiu. Grant: Inferring best forwarders from complex networks' dynamics through a greedy ant colony optimization. Technical report, INRIA, RR-7694, 2011.
- [4] A. Keränen, T. Kärkkäinen, and J. Ott. Simulating mobility and dtns with the one (invited paper). *J. of Commun.*, 5(2):92–105, February 2010.
- [5] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Comput. Commun. Rev.*, 7(3), 2003.
- [6] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.
- [7] J. Wu, M. Xiao, and L. Huang. Homing spread: community home-based multi-copy routing in mobile social networks. In *Proc. IEEE INFOCOM*, 2013.
- [8] M. Xiao, J. Wu, and L. Huang. Community-aware opportunistic routing in mobile social networks. *IEEE Trans. Comput.*, to be published, 2013.