This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2010 proceedings.

Socially-Aware Publish/Subscribe System for Human Networks

Yaxiong Zhao and Jie Wu
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122
{yaxiong.zhao, jiewu}@temple.edu

***Abstract*—The development of modern communication technologies fosters human networks (HUNETs)—an information dissemination platform comprised of human-carried wireless-enabled devices. These networks are promising applications of delay tolerant networks (DTNs). However, existing DTN routing protocols do not address unique problems in HUNETs and are unable to utilize the benefits of human mobility. In this paper, we present a publish/subscribe system for HUNETs. In our system, brokers are responsible for collecting subscriptions and dispatching messages, which provide better flexibility and scalability. The core component of our system is a socially-aware broker allocation algorithm called, social election. Social election trades off between the efficiency and overhead, by dynamically controlling the number of brokers in the network. A replication scheme and a buffer management method based on utility are also proposed to facilitate efficient message forwarding. We demonstrate, through simulations, that our system performs well in realistic networks under extensive conditions.**

***Index Terms*—delay tolerant network, publish/subscribe system, content-based addressing.**

I. Introduction

The development of modern communication technologies fosters a new information dissemination platform formed by human-carried wireless-enabled devices. These devices are powered by battery, therefore have only limited communication capability. A good example is the pocket network [1]. These kind of networks are modeled as mobile *delay tolerant networks* (DTNs), and we call them *human networks* (HUNETs). HUNETs are promising in ubiquitous computing. Just like DTNs, HUNETs are occasionally-connected, and suffer from constant network partitioning as the result of high mobility and short radio ranges. Providing efficient communication in HUNETs is a grand challenge. Unicast and multicast routing protocols for DTNs were researched in the past. However, few of them address or utilize the unique problems or properties of HUNETs. We summarize two of the most important questions as follows:

- Distributed content addressing: Communications in HUNETs are closely coupled with human interests. People, therefore, users in the network are communicating information that they share common interests in. Messages in HUNETs express user interests, which cannot be handled efficiently in the traditional routing protocols because they do not have the required constructs. Furthermore, HUNETs do not have end-to-end connectivity, which makes addressing an expensive operation. Users are unable to readily find the IDs of their intended destinations in the first place.
- Social network characteristics: The user mobility in HUNETs is essentially human mobility. One of the most important properties of human mobility is the community structure, which means people move in certain tracks and form closely related groups. This is substantially different from the purely random mobility model assumed in the previous routing protocols. We demonstrate in this paper that utilizing the social structures of HUNETs provides satisfactory performance with less overhead and complexity than the traditional routing protocols.

In this paper, we explore the use of the pub/sub system for efficient communication in HUNETs. In a pub/sub system, users are agnostic of each other. The information generated by the information producers, or *publishers*, is injected into the network, and forwarded by *brokers*, which are nodes responsible for collecting subscriptions and dispatching messages to interested users. The information consumers, or *subscribers*, register their interests to brokers, which are called, subscriptions. Messages are no longer routed based on their record destination. They are dispatched by brokers according to the matching between subscriptions and message content. Because it decouples the communication protocol and network structure, the pub/sub paradigm has inherent anonymity, scalability, and flexibility [2]. Thanks to these superiorities over traditional solutions, pub/sub has been identified as a promising solution to the information dissemination in HUNETs [3].

Pub/sub systems for fixed networks [4] have been researched and developed for many years. These works laid a solid foundation for our work. There are also many pub/sub systems for wireless ad-hoc networks [2], [5], [6], [7]. These works mainly focus on mechanisms to compensate the impact induced by network dynamics. They do not address the non-existence of end-to-end connectivity in HUNETs. Additionally, they do not consider the human mobility of HUNETs either. They are unable to utilize the dynamic community structure in human behaviors to improve performance.

In this paper, we present a new pub/sub system for HUNETs, which explicitly addresses the above questions in its design. The core component of our system is a simple and effective algorithm for allocating brokers in HUNETs called,

978-1-4244-6398-5/10/$26.00 ©2010 IEEE

social election. According to the social structure of HUNETs, we try to find the nodes that have a larger movement area to become brokers, which is measured by the *popularity* of the nodes. Social election uses local contact history to determine each node's role, and to control the number of brokers in the network. Social election takes advantage of the unique social structure of HUNETs. The forwarding structure formed by social election provides efficient message dispatching capability. Additionally, social election is able to control the overhead in message forwarding by adjusting the number of brokers, which is not well addressed in existing protocols.

Our contributions in this paper are as follows:

- We explore the use of the pub/sub system for efficient communication in HUNETs. We analyze its advantages over traditional approaches in flexibility and scalability.
- We propose social election, a dynamic broker allocation algorithm, which can dynamically control the number of brokers in the network. Coupling with utility-based replication and buffer management schemes, our system performs better than the traditional routing protocols.
- We conduct simulations to verify the performance of the proposed system. We use a realistic human mobility model in assessment, which substantiates the advantages of the socially-aware design of our system.

The rest of this paper is organized as follows: section II presents related work. Section III discusses network model. The design of our system is in Section IV. Section V analyzes simulation results. Section VI concludes this paper.

II. Related Work

We refer to [2], [7] for a general survey on pub/sub systems in mobile environments. Generally speaking, the pub/sub system can be centralized or distributed. Many distributed pub/sub systems have been proposed [5], [6], [7]. However, these protocols assume end-to-end connectivity, which are impractical for HUNETs. Resource discovery and content dissemination for mobile ad-hoc networks [4], [5] resemble our problem here. However, these works focus on efficient notification and querying to save precious bandwidth, and they do not consider the connectivity problem and human mobility model in DTNs either.

DTN routing protocols [8], [9], [10] can be used as a building block in HUNET communications. The problem is that the sources and the information holders do not know each other. In the Internet, the Domain Name Service (DNS) provides such an identification service, but in DTNs, users are disconnected from the infrastructure and other users. A ubiquitous naming service is hard to implement, or is too costly for practical use. Current DTN routing protocols focus on unicast, however, communications in HUNETs usually exhibit a many-to-many traffic pattern. This makes traditional routing protocols unfavorable in HUNETs. *Content based addressing*, and the corresponding routing model, are discussed in [11]. Content/resource discovery shares commons with the pub/sub system. A socially-aware routing method is proposed in [3]. Reference [12] studies efficient ranking and searching algorithms for mobile networks. TACO-DTN [13] is a preliminary work on buffer management for the content dissemination of DTNs. Because the infrastructure and artificial user movement are unavailable in HUNETs, it is impractical in HUNETs. DRIP [6] uses geometric separation to select brokers, which is restricted because it requires location information.

III. Network Model and Preliminaries

We discuss the network model and preliminaries on the content based addressing used in this paper.

A. Network models and assumptions

HUNETs consist of dozens to hundreds of wireless devices carried by humans. The reason why HUNETs are considered DTNs is that these devices have short communication ranges, which is a result of their restricted energy supply and hardware capability. The networks are therefore, partitioned, or the end-to-end communication is impractical. Typical scenarios include conferences, exhibitions, and campuses where attendees use smart phones or netbooks.

We assume that all the users are cooperative. Their devices are operated to meet the network-wide goals. Each node, or device carried by a person, has a fixed amount of memory for temporarily holding messages, which is called, a *buffer*. We assume that the buffer sizes of all the nodes in the network are the same. Nodes are able to exchange messages only when located in each other's communication range. The communication ranges of all the nodes are identical.

A pub/sub system has event providers, or *publishers*. Publishers generate messages that some users are interested in. Users who are interest in the messages of certain content are consumers, or *subscribers*. *Brokers* are responsible for collecting messages and subscriptions, and dispatching messages to interested subscribers. A node, equivalently a user, can simultaneously be a publisher, subscriber, and broker, which depends on the network environments.

The methods how messages are delivered from publishers to subscribers can be divided into *PUSH* or *PULL*. Informally speaking, they are different in which part initiates the transmission of the messages. Messages are injected into the network by publishers in PUSH, whereas messages are requested by the subscribers in PULL. We refer to [14] for a more comprehensive discussion. Our system is a PULL system with some adaption for supporting mobility.

B. content-based addressing

In this section, we describe the content-based addressing method used in this paper. We do not present the details about how to implement it, but depict how it works and interacts with other modules. Messages are identified and routed using content-based addressing and routing. The reason is that the messages in HUNETs are delivered according to users' interests, which can be perfectly represented and manipulated in content-based addressing. On the contrary, traditional techniques are impractical in such cases due to the lack of expressiveness.

The most important function is to identify the content of the messages. This function is able to match between messages' contents and subscriptions. Because the same message is of different values to users with varied interests, the degree a subscription and a message match each other is measured by a score; the higher the score, the better the matching. Message delivery has a delay. In order to incorporate it into the system, we associate the matching scores and the delay with a *utility*. The utility is calculated in each forwarding of the messages and is used in making replication decisions. Details will be presented in the next section.

IV. The Design of the System

Traditional pub/sub systems for mobile wireless networks assume end-to-end connectivity. They impose a well-formed structure for delivering messages. The association between users and brokers is rigidly enforced. However, this is impractical in HUNETs because of the lack of end-to-end connectivity and high mobility.

We employ a different approach in our system. A swarm of brokers travel around and collect messages and subscriptions, and no direct association between users and brokers is maintained. Our system has three orthogonal components: *broker allocation*, *message replication*, and *buffer management*. We use a novel process called, *social election* to allocate brokers. Messages are published to brokers without any restrictions. The messages are then propagated among brokers in order to reach the interested subscribers. Replication and buffer management are performed at brokers.

A. The dynamic Broker Allocation Algorithm

The broker allocation algorithm is executed at each node. Its pseudo code is depicted in Algorithm 1. Each node has a threshold of broker numbers for determining the role of the nodes met in the future. The threshold has a *lower bound* and an *upper bound*. For a given node (broker or normal user), when the number of brokers met in the *current time window* drops below the low bound, it designates the normal user it meets next time as a broker until the number of brokers it met satisfies the lower bound.

Algorithm 1 Dynamic broker allocation

```
When meets a new node node_new
N←the number of brokers met in the current time window
if N < lower_bound then
    if node_new is a broker then
        Do nothing
    else
        Designate node_new to a broker
    end if
else
    if node_new is a broker then
        γ ← the popularity difference
        β ← the popularity threshold
        if γ < β then
            Designate node_new as a normal user
        end if
    else
        Do nothing
    end if
end if
```

On the other hand, if the number of brokers met in the current time window surpasses the upper bound, the user tries to reduce the number of brokers by designating the brokers met in the future as normal users. This is achieved by social election. Social election assumes that nodes are likely to behave regularly in HUNETs. That is, a node that travels a wide range of network areas usually works in this way. This is represented as a node's *popularity*, which is measured as the number of different nodes it met in the current time window. The popularity of a set of nodes is the union of the popularity of each node in the set. Social election works as follows: suppose a node met α brokers in the current time window, and $\alpha \geq upper\_bound$, then the next broker will be designated to a normal user if the difference between its popularity and the popularity of the brokers met before in the current time window is less than β. β is called, popularity threshold.

The outcome of this algorithm is a swarm of brokers formed by popular nodes. The number of brokers is controlled, so that we are able to trade-off between efficiency and overhead. Because of the human mobility in HUNETs, those nodes that have a larger activity area, and thus have a higher popularity, will more likely become brokers. As long as brokers are selected, their statuses tend to be stable because of the community structure of HUNETs.

The algorithm has three parameters: time window size T_w, thresholds $\{lower\_bound, upper\_bound\}$, and popularity threshold β. The lower bound and the upper bound reduce the bouncing of the number of brokers, and provide more flexibility. A lower β value means the algorithm is more sensitive to nodes' popularity; whereas a higher β value means the brokers' quantity is more important in the allocation. All decisions are made based on local information. This algorithm allows nodes to set parameters that suit their own needs, therefore, it is flexible and extensible.

1) Switching probability: In our system, each node only needs its contact history in a fixed time window to determine its, or its future neighbors', status. We let each node switch according to a probability to avoid the fluctuation. The switching probability influences the dynamic of the number of brokers in the network. We currently set the value to 0.5. Our simulation results show that this probability has no noticeable impact on performance as long as the number of brokers is the same. A possible extension to Algorithm 1 is to use the ratio of brokers to all nodes met in the time window as a threshold for determining the role of neighbors. However, it lacks scalability. When networks are too sparse, or too dense, too few or too many brokers may be produced, which in turn, hampers the effectiveness, or incurs unnecessary overhead. Non-uniform speed distribution also introduces similar problems.

2) Designation and initiation: There is a subtle issue in determining when and who can switch to a broker or normal user. A normal node may switch to a broker itself when the number of brokers met in the time window drops below the lower bound. Or it can designate one of its encountered normal users as a broker. The later one is a rational choice as it conforms to the intuition that it needs more brokers for its efficient communication. However, consider a situation where a node moves like a ferry between two densely deployed clusters. Because there are enough brokers in each cluster, the ferries may not be selected as brokers, so that the communication between clusters is hindered. But, they meet less brokers when they move into the void area, which signals a sparse node distribution. The ferries should switch to brokers in this case.

B. Replication and buffer management

Because brokers have no knowledge about the publishers and subscribers, replication is driven by the presence of the messages and subscriptions. That is, matched messages and subscriptions are mutually attracted. Replication helps messages reach the source of the matched subscriptions. In order to control the overhead, i.e. the copies among brokers of the same message, a mixed PUSH/PULL scheme is employed. A user publishes messages and registers subscriptions to the brokers. A publisher can only forward one copy of the message to the first broker it meets. Brokers then interchange stored messages and subscriptions. Brokers are allowed to replicate multiple copies of the messages. We present two replication schemes in the following sections, which are based utility.

Subscriptions are periodically registered by the subscribers to the brokers. The time stamp of the subscriptions stored in any broker is refreshed when a fresher subscription is registered, no matter if it is from a normal user or another broker. When brokers switch to normal users, they replicate the messages they stored to other brokers they meet the next time. All subscriptions, other than their own, should be discarded, as normal users do not take part in message replication and forwarding.

1) Utility ranking: We use a utility based method to determine the priority of the messages in replication. The utility of a message is calculated in Eq. 1. It decreases with the *age* of the messages, which means the messages delivered with less delay are preferred. Because the delays are unknown before reaching the destinations, it is obtained by adding up the age of the message, and the age of the subscriptions, in the next-hop forwarder. The rationale of the addition is that messages traverse the reverse path of the subscriptions; the delay is therefore, the lifetime of the subscriptions.

$$U = \frac{matching\ score}{message\ age} \quad (1)$$

$$= \frac{matching\ score}{message\ lifetime + subscription\ lifetime} \quad (2)$$

2) Naïve replication: Naïve replication ranks the messages according to their utilities. The utility of a message in each forwarding is calculated using Eq. 1. As noted above, the message age is obtained by adding the message lifetime and the subscription lifetime, in the next-hop node. Then, highly-ranked messages are forwarded firstly until all the messages are transmitted, or the buffer in the next-hop node is depleted. This operation also applies to the process where messages are forwarded from normal users to brokers.

3) Delegation based replication: Delegation forwarding is a simple strategy in DTN routing [15] in order to reduce messages' copy numbers. We incorporate it into our replication scheme. In *delegation based replication*, the utility of each message in each forwarding is computed as in Eq. 1. In the next forwarding, the utility of a message should be larger than that of the previous one. This strategy reduces the overhead, and maintains the performance. Replication is similar to forwarding replication in DTN routing. Indeed, their disparity is that replication in a pub/sub system is only performed on brokers.

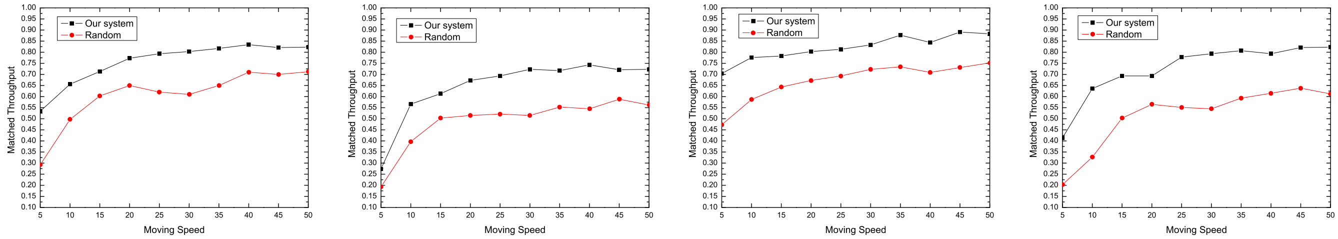
4) Buffer management: Messages are ranked in the buffer using Eq. 1. When the buffer is exhausted, lowest ranked messages are discarded. Matching scores are calculated against all the subscriptions stored in the nodes. And the message age is predicted by adding the message lifetime and the subscriptions lifetime. Some messages may not match any subscriptions. In this case, they are ranked according to lifetime. Fresher messages will be kept, older ones discarded. This is also used in replication when the same situation occurs, where fresher messages will be replicated.

V. Performance evaluation

A. Simulation Settings

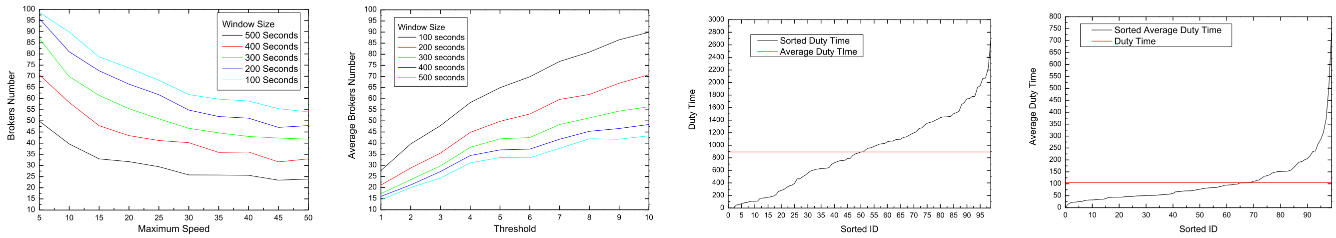
Simulations are run in a discrete event simulator written in C++. We conduct two sets of simulations. One uses the standard random way point (RWP) mobility model. The other uses a specific human mobility model called, SLAW [16]. For the ease of simulation, binary matching is used. That is, each message corresponds to exactly one type of subscription, which gives a matching score 1; otherwise, gives 0. 100 nodes are uniformly placed in a $1{,}000 \times 1{,}000m$ planar area. All nodes have a transmission range of $50m$. The maximum speed varies from $[5 - 50]m/s$. Each simulation runs for 3000 seconds. We randomly generate 5 types of messages, 5 publishers, and 10 subscribers for each type of message. The publishers generate 1 message per second. We assume that 50 messages can be transmitted within 1 second through wireless channels. Transmission errors are not considered. Each node has a buffer that can hold up to 100 messages. The popularity threshold is set as the mean popularity value of all the previous-encountered brokers.

To the best of our knowledge, there is no similar system proposed recently. For comparison, we designed a random allocation algorithm. This algorithm lets each node switch to a broker with a fixed probability. We first obtain the average number of brokers in the network using our system. Then, we set the probability in such a way, that the number of brokers in the networks using random algorithm is the same to that of the social election method. Brokers serve in a cyclic manner. After each cycle, each node will use the above mechanism to



(a) RWP and naïve replication (b) RWP and delegation replication (c) SLAW and naïve replication (d) SLAW and delegation replication

Fig. 1. Matched throughput in networks with different mobility models and replication schemes



(a) vs speed (b) vs threshold (c) duty time (d) average duty time

Fig. 2. The overhead of our system measured in the number of brokers and duty time

determine its role again. We call it, *Random*. The two protocols have similar numeric values in delay. Due to lack of space, we do not present the detailed results.

B. Throughput

An important performance metric of the pub/sub system is the matched throughput, which is measured by the utility of the messages being delivered to the interested users. Because we use binary matching, the results can be expressed as the volume of messages have been delivered. We compare the results of our system and the random algorithm introduced above. The parameters of this set of simulations are a lower bound of 3, an upper bound of 3, and the time window size is 100 seconds. The maximum moving speed is varied to validate the performance under different settings.

Fig. 1(a) and 1(b) give the overall matched throughput of our system and the random algorithm in the standard RWP model. We give the results with moving speeds varying from 5 to 50m/s. The throughput increases with the moving speed because a higher moving speed increases the contact rate. It can be concluded that our system's performance is better than the random algorithm.

Fig. 1(c) and Fig. 1(d) show the performances of our system and the Random using the SLAW model, with the same parameters as above. The performance advantage of our system is larger in the SLAW mobility model. We see similar performance curves as in Fig. 1(a) and 1(b). The performance improvement is larger in less dynamic mobility environments, which results in an increase in availability of contact opportunities in the network. We would like to point out that for other parameter settings, the comparisons of the performance curves of our system and Random are similar.

C. Overhead

In this section, we present the overhead of our system measured in the number of brokers and their duty time. We do not present the results of the random algorithm, for it is designed to exhibit similar properties. Although the number of brokers is not a precise numeric value of the overhead of the system, it represents the overall trends. Because the numeric overhead depends on the traffic pattern and load, it may be misleading in some cases. Fig. 2(a) presents the number of brokers changing with different moving speeds. The parameters are a lower bound of 3 and an upper bound of 3.

Fig. 2(b) gives the number of brokers versus different threshold values. The maximum moving speed for this set of results is $10m/s$. The window sizes vary from 100 to 500 seconds. As depicted in Fig. 2(a), the number of brokers decreases with the increase of the moving speed. This follows the intuition that a higher speed increases the contact rate between nodes, thus, fewer brokers are needed to satisfy the threshold criteria. In Fig. 2(b), larger threshold values generate more brokers. If delegation based replication is used, subscriptions and messages are only spread among a fraction of all brokers; so the actual number of brokers that take part in the serving is even less.

We present the results of duty time for each node in Fig. 2(c) and Fig. 2(d). The parameters are a threshold 3, a time window of 100 seconds, and a maximum moving speed of 10m/s. The values are sorted in ascending order in the Fig. 2(c). The red line is the average duty time for each node. 80 out of 100 nodes serve as brokers for less than 1500 seconds, which is half of the overall simulation time (3000 s). We define duty cycle as the time duration, in which some nodes continuously serve as brokers. We compile the above results to show the average duty cycle with sorted ID in Fig. 2(d). The majority of the

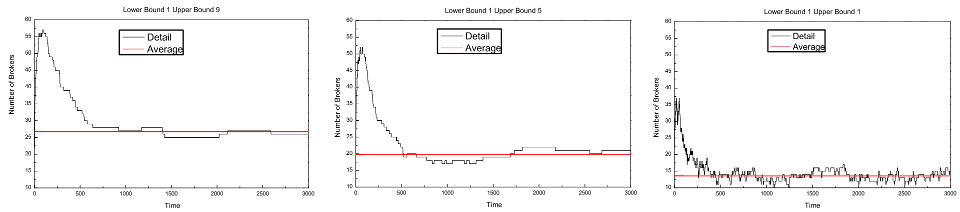


Fig. 3. The changing of the number of brokers in networks with different parameters.

nodes have a relatively smaller duty cycle. This alleviates the overhead imposed on each node, and reduces the probability of exhaustion of the brokers' buffer.

D. Microscopic Behaviors

This subsection presents the microscopic behaviors of the system in terms of the number of brokers over time. The mobility model used is SLAW. This set of results is necessary to assess the stability of the system because, although the overall number of brokers is stable, the frequent changing of nodes' statuses may incur additional operations, and may render unpredictable behavior of the system. Fig. 3 presents the number of brokers over time, with a lower bound of 1, upper bounds from 9 to 1, and a window size of 500 $seconds$. The black line represents the number of brokers in the network in each second. The red line is the average number of brokers after 500 seconds.

We see in these three figures that the trajectory is smoother with a larger bound difference. We conclude that the fluctuation and the average number of brokers depend on the difference between the lower bound and the upper bound. There is a convergence time for the system to become stable. The convergence is fast, and is proportional to the time window size. By adjusting the threshold based on the known history, i.e. proportionally changing the thresholds based on the length of time in which the system has started working, we can reduce the steep changing curve. Note that this optimization has no influence on the protocol performance in the long term because the behavior after the initial convergence time is all the same.

VI. Conclusion and Future Work

In this paper, we considered the efficient communication problem in HUNETs, which is formed by wireless-communication-enabled devices carried by humans. We explored the use of the pub/sub system in such networks. A new pub/sub system is presented. Content-based addressing is used. A socially-aware broker allocation algorithm is proposed to trade-off between efficiency and overhead. Utility-based replication schemes and buffer management are also used to facilitate efficient message forwarding. Simulation results show that our system performs well in terms of throughput and overhead under extensive conditions.

Personal digital devices are becoming more and more prevalent. At the same time, HUNETs are promising to become an efficient information dissemination and content distribution platform. In our future work, we will continue to explore the use of the pub/sub system and socially-aware algorithms in HUNETs. Many interesting applications can be deployed in HUNETs. We will identify them and develop the corresponding protocols, algorithms, and prototype systems.

Acknowledgement

This work is supported in part by NSF grants CNS 0422762, CNS 0626240, CCF 0830289, and CNS 0948184.

References

[1] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, pp. 748–760, Feburary 2005.

[2] Y. Huang and H. Garcia-Molina, "Pub/sub in a mobile environment," in *Proc. of ACM MoBiDE '01*, vol. 5, April 2001, pp. 17–26.

[3] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, pp. 748–760, June 2008.

[4] R. Zhang and Y. Hu, "Hyper: A hybrid approach to efficient content-based publish/subscribe," in *Proc. of IEEE ICDCS '05*, June 2005, pp. 427–436.

[5] T. Pongthawornkamol, K. Nahrstedt, and G. Wang, "The analysis of publish/subscribe systems over mobile wireless ad hoc networks," in *Proc. of ACM MobiQuitous '07*, Aug. 2007, pp. 1–8.

[6] Q. Yuan and J. Wu, "Drip: A dynamic voronoi regions-based pub/sub protocol in mobile computing," in *Proc. of IEEE INFOCOM '08(Mini Conference)*, Aug. 2008, pp. 1–8.

[7] L. Fiege, F. C. Gärtner, O. Kasten, and A. Zeidler, "Supporting mobility in content-based publish/subscribe middleware," in *Proc. of ACM/IFIP/USENIX Middleware '03*. New York, NY, USA: Springer-Verlag New York, Inc., 2003, pp. 103–122.

[8] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. of ACM MobiHoc '04*. New York, NY, USA: ACM, 2004, pp. 187–198.

[9] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *Proc. of ACM SIGCOMM '07*. New York, NY, USA: ACM, 2007, pp. 373–384.

[10] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. of ACM SIGCOMM '04*. New York, NY, USA: ACM, 2004, pp. 145–158.

[11] G. Cugola, A. L. Murphy, and G. P. Picco, "Content-based Publish-subscribe in a Mobile Environment," in *Mobile Middleware*, P. Bellavista and A. Corradi, Eds. Auerbach Publications, 2006, pp. 257–285, invited contribution.

[12] O. Wolfson, B. Xu, H. Yin, and H. Cao, "Search-and-discover in mobile p2p network databases," in *Proc. of IEEE ICDCS '06*. Washington, DC, USA: IEEE Computer Society, 2006, p. 65.

[13] G. Sollazzo, M. Musolesi, and C. Mascolo, "Taco-dtn: a time-aware content-based dissemination system for delay tolerant networks," in *Proc. of ACM MobiOpp '07*. New York, NY, USA: ACM, 2007, pp. 83–90.

[14] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Content-based addressing and routing: A general model and its application," 2000.

[15] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proc. of ACM MobiHoc '08*. ACM, 2008, pp. 251–260.

[16] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A mobility model for human walks," in *Proc. of IEEE INFOCOM '09*. IEEE, April 2009.