Mobile Information Systems 9 (2013) 331–345
DOI 10.3233/MIS-130166
IOS Press

A P2P query algorithm for opportunistic networks utilizing betweenness centrality forwarding

Jianwei Niu[a,*], Mingzhu Liu[a] and Han-Chieh Chao[b]
[a] *State Key Laboratory of Software Development Environment, Beihang University, Beijing, China*
[b] *Institute of Computer Science and Information, National Ilan University, I-Lan, Taiwan*

Abstract. With the proliferation of high-end mobile devices that feature wireless interfaces, many promising applications are enabled in opportunistic networks. In contrary to traditional networks, opportunistic networks utilize the mobility of nodes to relay messages in a store-carry-forward paradigm. Thus, the relay process in opportunistic networks faces several practical challenges in terms of delay and delivery rate. In this paper, we propose a novel P2P Query algorithm, namely Betweenness Centrality Forwarding (PQBCF), for opportunistic networking. PQBCF adopts a forwarding metric called Betweenness Centrality (BC), which is borrowed from social network, to quantify the active degree of nodes in the networks. In PQBCF, nodes with a higher BC are preferable to serve as relays, leading to higher query success rate and lower query delay. A comparison with the state-of-the-art algorithms reveals that PQBCF can provide better performance on both the query success Ratio and query delay, and approaches the performance of Epidemic Routing (ER) with much less resource consumption.

Keywords: Opportunistic networks, P2P Query, betweenness centrality, social networks, mobile devices, epidemic routing

1. Introduction

Opportunistic networks [33] utilize the "store-carry-forward" paradigm and leverage the mobility of nodes to relay messages from the source to the destination. Thus in this sense, opportunistic network is typically a special case of Delay Tolerant Network (DTN) [9]. Recent years have witnessed a number of proof-of-concept opportunistic network applications such as ZebraNet [16], CarTel [15], Pocket Switch Networks [22], and P2P networks [5]. Information dissemination in opportunistic networks has very different features compared with traditional networks. In traditional wired/wireless networks, there are usually sustained end-to-end paths existing between the source and the destination. As a result, information can be disseminated by maintaining a certain routing topology. However, in opportunistic networks, due to the intermittent links between mobile nodes and the lack of end-to-end routing path, data can only be forwarded hop-by-hop in an opportunistic way. In this situation, the key problem is to find a routing strategy to determine whether to forward or keep the data for the owner when two nodes meet. In this paper, we focus on P2P query in opportunistic networks, more specifically the Social Opportunistic Network (SON) which is composed by mobile devices held by people. In P2P query, a query node

---

[*] Corresponding author: Jianwei Niu, State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China. E-mail: niujianwei@buaa.edu.cn.

first sends an inquiry message across the network to search for another node carrying the corresponding response message. Then, the response message is forwarded to the query node. Before describing our work, we first review some related work.

Routing in delay tolerant networks have been extensively studied during the lasting few years, which focuses on how to select the next-hop data carrier during each encounter. Most work seeks to maintain a balance between message delivery rate and the cost, i.e., the number of in-network message copies. Epidemic Routing (ER) [30] is one of the most famous routing algorithms in opportunistic networks, which utilizes flooding to disseminate information throughout the network. If the resource is not the main concern, ER guarantees the lowest transmission delay and the highest data delivery rate. However, constrains in terms of node energy, bandwidth, and buffer size prevent ER from been widely used in practice. A number of protocols try to reduce the cost of flooding while still maintaining a high message delivery rate. In [12], the authors proposed a 2-hop flooding algorithm, where the source node relays messages to the first L contacts, and then these L nodes forward the messages to the destinations. Consequently, each message reaches its destination in two hops, and there are $(L + 1)$ message copies in the network, which highly reduces the cost incurred by ER flooding. A following proposed algorithm, i.e., Spray and Wait (SW) [24] further improves ER by limiting the number of copies and avoiding performance deterioration caused by channel competition. Recently, the Spray and Focus (SF) [25] algorithm improves the SW algorithm by forwarding messages to nodes with higher probabilities to encounter destinations. Similar routing algorithms are also proposed in [17–19,35].

Several protocols are proposed for disseminating information to nodes with special properties, e.g., within a certain area. In [34], Xu et al. presented a spatial-temporal aware algorithm for data dissemination in which each message carries a time-to-live field and its location information. By this means, the algorithm effectively restricts the data dissemination range and survival time in the network, and thus the network load and bandwidth consumption are reduced. Wischhof et al. [31] proposed to use a periodically active broadcasting algorithm for message propagation and introduced dynamic adjustment of broadcasting cycles to improve message relay efficiency. Based on this idea, Eichler et al. [9] proposed to set the broadcast cycle according to urgency levels of messages. Caliskan et al. [6] introduced location-based fusion techniques into periodically active broadcasting algorithms, which further improved the message distribution efficiency.

Costa et al. [7] proposed a response message return algorithm which for the first time combined deterministic routing and random routing. In this algorithm, inquiry nodes broadcast an Inquiry Message (IM) periodically to nodes within their n hops, and then the IM can spread around the network through node movement. When the message source receives an inquiry message, the response message will be accurately routed to the inquiry node if the source node is close enough to the inquiry node and the source has reserved deterministic routing information. Otherwise, the response message will be randomly relayed by choosing some neighboring nodes until the response message's time to live (TTL) expires. Another inquiry/response propagation method is proposed in [3] which combines the content-based routing with the probability-based routing. The forwarding strategy of response message considers not only the distance between the information carrier and the inquiry node, but also the predicted encountering probabilities between nodes to select the most appropriate node as the next hop, which reduces the delivery delay.

For the same purposes, Pan et al. proposed in [23] a combined info-acquisition algorithm of inquiry message and source message. The algorithm targeted at a specific application scenario – Pocket Switched Networks (PSN) and a content routing-based algorithm is proposed for message inquiry. The algorithm assimilated the overall inquiry process to anti-permeation in physics, in which each inquiry node periodically broadcasts the inquiry messages with TTL, gradually forming concentration gradient of the IM around them. When the source receives an IM, response messages are forwarded to areas with a higher IM concentration like solvent molecules, and eventually back to the inquiry node. On this basis, Eichler et al. [9] proposed the idea of setting broadcasting cycles according to importance level. This algorithm defines several different importance levels for different messages due to limited bandwidth. Also by using the optimization theory, the algorithm preferentially broadcasts the message with the maximum priority through dynamic adjustments of broadcasting priority and frequency, so as to enhance distribution efficiency. Caliskan et al. [6] introduced GPS-based message fusion technology into the message dissemination of periodically initiative broadcasting and adjusted message dissemination rates according to nodes' information pattern, which further improved the dissemination rate.

Ouri et al. [32] proposed a Rank-Based Broadcast (RBB) inquiry algorithm in which inquiry nodes and source nodes periodically broadcast their IMs and source messages, respectively. Intermediate nodes maintain two queues for IMs and source messages, respectively. The IMs are ranked according to their request time, while the source messages are ranked according to their matching degree to IMs. When the communication channel is established, intermediate nodes will first broadcast the IM with the source message with the highest priority. This dissemination method can achieve the optimum use of the broadcast channel, and thus it maximizes the success rate of message dissemination. In [13], it is proposed to use metadata to reduce the additional expenses in RBB. Different from RBB, source nodes only broadcast the metadata instead of an entire source message. When the metadata matches a certain IM, the request message is sent to the message source, triggering it to transmit the source message. Inquiry/response is also studied in VANET (Vehicle Ad hoc NET), where Ilias et al. [20] proposed the concept of vehicle clustering to improve the inquiry process. The message source node periodically broadcasts the source message, collects IMs from neighboring vehicles, and divides inquiry vehicles into clusters according to their moving directions. The message source node selects the cluster with the maximum amount of IMs and forwards the source message to vehicles driving towards the cluster.

So far, we have made a brief survey of routing and inquiry/response algorithms in the literature. In this work, we focus on SON, where nodes are mobile devices carried by people. With the wide spreading of smart phones, we believe inquiry/response is a promising research topic in opportunistic networks. Because the devices are carried by human, the node mobility has some social characteristics, which have rarely been exploited for inquiry/response by the state of the art. We assume that all the nodes in a SON are willing to collaborate to relay messages. We introduce a concept, namely Betweenness Centrality (BC) in SONs. Then, we propose an algorithm for message inquiry/response based on nodes' BCs. Intuitively, nodes with higher BC values indicate that they are more active, and thus have more opportunities to encounter different nodes. Therefore, they should undertake more message forwarding tasks. Cai et al. in [8] found that both the node inter-contact time and nodes' contact frequencies in SON follow the power-law distribution, indicating that a few nodes play a decisive role in network connectivity and message transmission while most of the nodes are not so significant [21,26]. Literature [14,36] revealed that opportunistic networks have a typical "small world" property that on average, most messages can reach their destination through a short path (about 4–5 hops). Therefore, our design is inspired by those interesting findings.

Our major contributions in this paper are summarized as follows:

1) We propose a P2P (peer-to-peer) message inquiry scheme called PQBCF based on nodes' BC value in SON. PQBCF includes the calculation method for the number of copies of IMs and response messages.
2) We carried out extensive evaluating on the performance (message delivery ratio and delivery delay) of the PQBCF algorithm.

The remainder of this paper is organized as follows. Section 2 presents the PQBCF algorithm. In Section 3, the performance of PQBCF is evaluated by simulations. Section 4 provides some concluding remarks.

2. P2P query algorithm based betweenness centrality forwarding

2.1. Problem description and term definitions

The application scenario of our approach is: in SON, when a node (carried by a mobile phone user) intends to request some certain information (e.g., an MP3 file), it first generates an IM with descriptions of the required data. In order to reduce the expected inquiry latency, the generator produces multiple copies of the IM. Intuitively, more copies lead to shorter query delay, however, introducing higher overhead in buffer management for each node. As a result, an inquiry/response strategy should seek to find a tradeoff between query delay and overhead. To achieve this objective, PQBCF calculates the number of copies of the IM in the network according to several factors which include the expected query delay, the mobility, and density of nodes. Then, the IMs are disseminated to the network by using our PQBCF algorithm which is based on a metric called Betweeness Centrality. If the IM is passed to a node with data matching the IM, the node will generate a response message, calculate the number of copies of the response message, and send the response message back to the inquiry node. The key advantage of PQBCF is that it utilizes the betweeness centrality metric inspired from social network analysis, and as a result, PQBCF can significantly reduce the inquiry/response delay. Before describing the details of PQBCF, we first introduce several definitions used throughout this paper.

Definition 1. *Inquiry Message (IM). When a node requests certain information, it generates an Inquiry Message (IM)with format as follows:*

$$IM(\mathit{Type}, ID\_n, ID\_m, TTL, \mathit{Topic}, \mathit{Path}, \mathit{Content})$$

where, *Type* is used to distinguish between an IM or a response message; *ID_n* and *ID_m* are the identity of the IM and the inquiry node, respectively; *TTL* indicates the survival time of this message; *Topic* is the query subject; *Path* keeps track of the nodes passed by during the message delivery process; *Content* stands for the query (e.g., requesting an MP3 file).

Definition 2. *The Response Message (RM). When a node receives an IM and has requested content, the node will generate a RM. Its format is defined as follows:*

$$RM(\mathit{Type}, ID\_n, ID\_m, TTL, ID\_d, \mathit{Path}, \mathit{Content})$$

where *Type, ID_n, ID_m, TTL, Path* are exactly the same with those in the definition of IM; *ID_d* is the identity of the destination node (i.e., the node that posts the inquiry); *Content* is the content requested by the inquiry node (e.g., the requested MP3 file).

Definition 3. *Node Betweenness Centrality. The Betweenness Centrality* $C_B\,(P_i)$ *of node* P_i *is used to quantify the importance of the node* P_i *in message delivery in the whole network. To formally define* $C_B\,(P_i)$*, let* g_{sd} *denote the number of all the messages successfully delivered between a pair of nodes*

Table 1
Network betweenness centrality

	Sum	P_1	$\dots$	P_i	$\dots$	P_n
P_1	g_{1d}	0	$\dots$	$g_{1d}(P_i)$	$\dots$	$g_{1d}(P_n)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
P_s	g_{sd}	$g_{sd}(P_1)$	$\dots$	$g_{sd}(P_i)$	$\dots$	$g_{sd}(P_n)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
P_n	g_{nd}	$g_{nd}(P_1)$	$\dots$	$g_{nd}(P_i)$	$\dots$	0

P_s and P_d. $g_{sd}(P_i)$ represents the number of the messages that pass through P_i during these message forwarding processes. Then, $b_{sd}(P_i)$ is defined as shown in Eq. (1),

$$b_{sd}(P_i) = \frac{g_{sd}(P_i)}{g_{sd}} \tag{1}$$

where $b_{sd}(P_i)$ represents the ratio of the messages successfully delivered via P_i to all messages successful messages between P_s and P_d. $b_{sd}(P_i)$ indicates the importance of P_i in delivering messages for P_s and P_d. Suppose the number of nodes in the network is n, and then the betweeness centrality value of P_i can be calculated by Eq. (2).

$$C_B(P_i) = \frac{2}{(n-1)(n-2)} \sum_{s=1,s\neq i}^{n} \sum_{d=1,d\neq i,s\neq d}^{n} b_{sd}(P_i) \tag{2}$$

Therefore, $C_B(P_i)$ means the average of all $b_{sd}(P_i)$ for any pair of nodes P_s and P_d in the network. In SONs, $C_B(P_i)$ represents the centrality of node P_i. A high $C_B(P_i)$ indicates that node P_i has more opportunities to contact with other nodes. However, $C_B(P_i)$ is only the average for all destination nodes. In order to analyze the messages that node P_i forwards to a specific destination node P_d, we define $C_{Bd}(P_i)$ which can be calculated by Eq. (3). It looks similar with Eq. (2), and however, the destination node is fixed. $C_{Bd}(P_i)$ indicates the importance of P_i in delivering message to d.

$$C_{Bd}(P_i) = \frac{2}{(n-1)(n-2)} \sum_{s=1,s\neq i}^{n} b_{sd}(P_i) \tag{3}$$

To calculate $C_B(P_i)$ and $C_{Bd}(P_i)$, node P_d maintains a table $T(P_d)$ as shown in Table 1. The rows in Table 1 are the source nodes P_s $(s = 1, 2, \cdots, n, s \neq d)$, and the columns are the intermediate nodes P_i $(i = 1, 2, \cdots, n, i \neq d)$. The first column ($Sum$) is the number ($g_{sd}$) of messages successfully delivered from the source node P_s to the destination node P_d. Each of the other columns records the numbers of messages delivered from the source node P_s to the destination node P_d via the intermediate node P_i, respectively.

One should note that Table 1 can be obtained by each node in a distributed fashion. A message is forwarded from its source node to its destination via multiple intermediate nodes. Each intermediate node can thus save its ID into the Path field of the message. When the message reaches its destination node P_d, P_d updates its $T(P_d)$ table according to the *Path* field of this message.

According to the table of Betweenness Centrality, a node is able to calculate a $b_{sd}(P_i)$ value and a $C_{Bd}(P_i)$ value setting itself as the destination node. When two nodes meet, they exchange their reserved $b_{sd}(P_i)$ values and $C_{Bd}(P_i)$ values, till all reserved $b_{sd}(P_i)$ values are synchronized and all $C_{Bd}(P_i)$ values updated. After each encounter, the node consciously updates a $C_B(P_i)$ value according to all reserved $b_{sd}(P_i)$ values. Since Table 1 only maintains the local information corresponding to the node P_d, its buffer consumption is less than $O(n^2)$ even under the worst circumstances, which is desirable for current mobile devices.

2.2. Calculate the number of inquire message copies

In order to reduce the delay of obtaining the corresponding response for an inquiry node, it may disseminate multiple IMs into the network. However, mobile devices usually have very limited buffer size. Thus, caching IMs is the main overhead for an inquiry/response system. Intuitively, more IMs can help to reduce the inquiry delay but incur a higher cost. As a result, the system should carefully organize the number of IM copies to find a tradeoff between inquiry delay and cost. To achieve this goal, we propose to calculate the number of the IM copies (denoted as L) in the network according to the required inquiry propagation delay, node mobility model and network density. An inquiry node can increase L to have a higher probability to receive the response message in the required delay. Therefore, determining the value of L is the key challenge for our algorithm.

In [24], Spyropoulos et al. studied the average message transmission delay in intermittently connected mobile networks. Let S denote the network area. N is the number of mobile nodes. K is the transmission range. Nodes in the network perform independent random walk and *Direct Transmission* (source nodes only forward messages to their destinations). Then, the expected delay of direct transmission is

$$ED_{dt} = 0.5S\left(0.34\log_2 S - \frac{2^{K+1} - K - 2}{2^K - 1}\right) \tag{4}$$

The expected delay of the optimal algorithm is

$$ED_{opt} = \frac{H_{N-1}}{N-1} ED_{dt} \tag{5}$$

It is proved in [24] that ED_{opt} is the low bound of the expected transmission delay of any algorithms, where H_n is the nth harmonic number, i.e,

$$H_n = \sum_{i=1}^{n} \frac{1}{i} \tag{6}$$

The expected IM propagation delay (ED) should be larger than ED_{opt}, i.e., ED meets the following condition where a satisfies $a \geqslant 1$

$$ED = aED_{opt} \tag{7}$$

Then the relationship between *ED*, L and the total number of nodes N in the network can be described as follows.

$$(H_N^3 - 1.2)L^3 + \left(H_N^2 - \frac{\pi^2}{6}\right)L^2 + \left(a + \frac{2N-1}{N(N-1)}\right)L = \frac{N}{N-1} \tag{8}$$

Equation (8) shows the relationship between L, *ED* and N. However, the number of nodes in the network is global information that can only be obtained in a centralized way. Moreover, due to the frequent node movements, it may be dynamically changing over time. As a result, the value of L should be adaptively tuned according to the number of nodes in current areas.

We use a heuristic method to estimate the number of nodes in the network. Assume that the nodes move independently and are identically distributed. It is proved in [24] that the delay (denoted as T_1) of node i in meeting with another node within the area follows an exponential distribution D_{dt} with mean of $D_d/\left(N-1\right)$. Recall that the expected delay of direct transmission is ED_{dt}. Therefore, we have

$$T_i = \frac{ED_{dt}}{N-1} \tag{9}$$

The delay (denoted as T_2) of node i in meeting with two different nodes follows an exponential distribution with mean of $D_{dt}\left(1/\left(N-1\right)+1/\left(N-2\right)\right)$, i.e.,

$$T_2 = ED_{dt}\left(\frac{1}{N-1}+\frac{1}{N-2}\right) \tag{10}$$

We can derive an estimation of the number of nodes N by combining Eqs (9) and (10),

$$N = \frac{2T_2-3T_1}{T_2-2T_1} \tag{11}$$

One should note that the node is able to measure T_1 and T_2 through locally. Thus, the total number of nodes N in the network can be estimated according to Eq. (11) in a distributed way. After obtaining N, we can determine the number L of copies of distributed messages according to Eq. (8).

2.3. *Propagation algorithm for inquiry messages*

An inquiry node needs to disseminate the generated L IM copies in the network. In this paper, we propose the Spread distribution algorithm based on improved binary tree, which chooses nodes with greater values of BC as transmission nodes, making it possible to forward messages as quickly as possible, with specific procedures as follows:

1. Inquire node P_s generates inquiry messages, determines the ratio a of expected transmission delay *ED* and the theoretically optimal transmission delay expected *ED_{opt}* according to the degree of urgency of inquiry contents; calculates the total number of nodes N according to the value of T_1, T_2, confirming the number L of distributed copies of inquire messages, records the node distribution task with mark *token* $\left(P_s\right)=L$, turn to step 2;
2. Node P_m carrying inquiry messages decides whether the TTL value of IMs is 0, if positive, delete the message, turn to step 2; otherwise, to determine whether the label of distributed task token is 1, if positive, turn to step 3; if negative, turn to step 5;
3. Node P_m carrying IMs meets certain node P_i to exchange the value of $T\left(P_m\right)$, $T\left(P_i\right)$ of their BC tables, and calculates the updated value of BC $C_B\left(P_m\right)$, $C_B\left(P_i\right)$. Node P_m asks whether node P_i has been carrying the IM, if positive, repeat steps 3, P_m continues to meet with other nodes; otherwise, turn to step 4;
4. Nodes P_m, P_i determine whether $C_B\left(P_m\right) < C_B\left(P_i\right)$, if established, first of all P_m minus TTL value of the inquire messages by 1, and then forward the message to P_i, meanwhile deleting its own copies of the IM, set the of node P_i as 1, turn to step 2; If negative, directly turn to step 2;
5. Node P_m carrying IMs meets certain node P_i to exchange the value of $T\left(P_m\right)$, $T\left(P_i\right)$ of their BC tables, and calculate the updated value of BC $C_B\left(P_m\right)$, $C_B\left(P_i\right)$. Node P_m asks whether node P_i has been carrying the IM, if positive, turn to step 2; if not, to determine whether $C_B\left(P_m\right) < C_B\left(P_i\right)$; if established, turn to step 6; If negative, turn to step 2;

6. First of all, P_m minus TTL value of the IMs by 1, and then forward the message to P_i, meanwhile allocating half of the distributed task to node P_i, then goes on to the remaining other half of the task, namely $\mathit{token}\left(P_i\right) = \lceil \mathit{token}\left(P_m\right)/2 \rceil$, $\mathit{token}\left(P_i\right) = \lfloor \mathit{token}\left(P_m\right)/2 \rfloor$, turn to step 2.

In this algorithm, two nodes update their $T\left(P_m\right)$ and $T\left(P_i\right)$ value when they encounter each other, and conduct a comparison operation, therefore the algorithm complexity is $O\left(E\right)$, wherein E stands for the number of times that the node carrying IMs meets with other nodes. $E\left(= ax^{-k}, k > 1\right)$ obeys the power law distribution [8], which means that most nodes in the network have low computational complexity, and that the algorithm manifests good scalability.

2.4. Back forwarding algorithm for response messages

When a certain node of the network receives an IM, it first checks the *Topic* field of the IM to determine whether it is able to respond to such inquiries, if positive, then it performs the following Back Forwarding algorithm; if negative, the node will perform as a transmission node to implement Spread distribution algorithm above. The Back Forwarding algorithm is based on the probability of meeting with the target node and selecting forwarding nodes. The specific procedures are as follows.

1. Response node P_r generates a response message, views the source node ID of ID_n field of the inquire message, and fills in the source node ID (the message's destination node ID, without loss of generality, assumed as P_d), turn to step 2;
2. Response node P_r checks the source node's expected inquiry delay *ED* preserved in the *Control* field of the IM, determines to back-forward the ratio of expected delay of the response message and ED_{opt}; measures the values of T_1, T_2 of the response node P_r, calculates the total number of nodes N, determines the number of forwarding copies L of the response message, records the forwarding task mark of response messages $\mathit{token\_r}\left(P_r\right)$, turn to Step 3;
3. Node P_{rm} carrying IMs decides whether the TTL value of IMs is 0, if positive, delete the message, turn to step 3; otherwise, determines whether the forwarding task label $\mathit{token\_r}\left(P_{rm}\right)$ is 1, if positive, turn to step 4; if negative, turn to step 6;
4. Node P_{rm} carrying IMs meets certain node P_i to exchange the value of $T\left(P_{rm}\right)$, $T\left(P_i\right)$ of their BC tables, and calculate the updated value of BC $C_{Bd}\left(P_{rm}\right)$, $C_{Bd}\left(P_i\right)$. Node determines whether node is the destination node of the IM, if positive, forwards the message, 'end of algorithm'; if negative, P_{rm} asks whether P_i carries the response message; if positive, repeat step 4; If negative, turn to step 5;
5. Nodes P_{rm}, P_i determine whether $C_{Bd}\left(P_{rm}\right) < C_{Bd}\left(P_i\right)$, if established, P_{rm} will first minus the TTL value of the response message by 1, and then forwards the message to P_i, meanwhile deleting its own copies of the response message, sets $\mathit{token\_r}\left(P_i\right) = 1$, turn to step 3; If negative, directly turn to Step 3;
6. Node P_{rm} carrying IMs meets certain node P_i to exchange the value of $T\left(P_{rm}\right)$, $T\left(P_i\right)$ of their BC tables, and calculates the updated value of BC $C_{Bd}\left(P_{rm}\right)$, $C_{Bd}\left(P_i\right)$. Node P_{rm} determines whether P_i is the destination node of the IM; if positive, forwards the message, end of calculation; if negative, node P_{rm} asks whether node P_i has been carrying the IM, if positive, turn to step 3, if negative, determines whether $C_{Bd}\left(P_{rm}\right) < C_{Bd}\left(P_i\right)$; if established, turn to step 7; If negative, turn to step 3;
7. Node P_{rm} first minus TTL value of the IMs by 1, and then forward the message to P_i, meanwhile allocating half of the distributed task to node P_i, then goes on to the remaining t other half of the task, namely $\mathit{token\_r}\left(P_i\right) = \lceil \mathit{token\_r}\left(P_m\right)/2 \rceil$, $\mathit{token\_r}\left(P_m\right) = \lfloor \mathit{token\_r}\left(P_m\right)/2 \rfloor$, turn to step 3.

Table 2
Main simulation parameters

Simulation parameter	values
Simulation time (hour)	12 (ONE internal timer)
PQBCF learning time (hour)	3
Simulation of the regional area (width, height: m)	4500, 3400
The number of mobile nodes	300, 500, 1000
Node cache (Mbyte)	5
Message size (byte)	512
Minimum node speed (m / s)	0.5
Node maximum speed (m / s)	3
Transmission distance (m)	10, 30, 50, 75, 100
Transfer rate (kbyte / s)	250
FIFO buffer queue model	FIFO
Message generation cycle (minute)	15
A	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
TTL (hop)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

In this algorithm, two nodes update value $T\left(P_m\right)$, $T\left(P_i\right)$ on meeting each other, and conduct a comparison operation, therefore the algorithm complexity is $O\left(E\right)$, where E stands for the number of times that the node carrying inquire messages meets with other nodes. $E\left(=ax^{-k}, k>1\right)$ obeys the power law distribution [8], which means that most nodes in the network have low computational complexity and that the algorithm manifests good scalability.

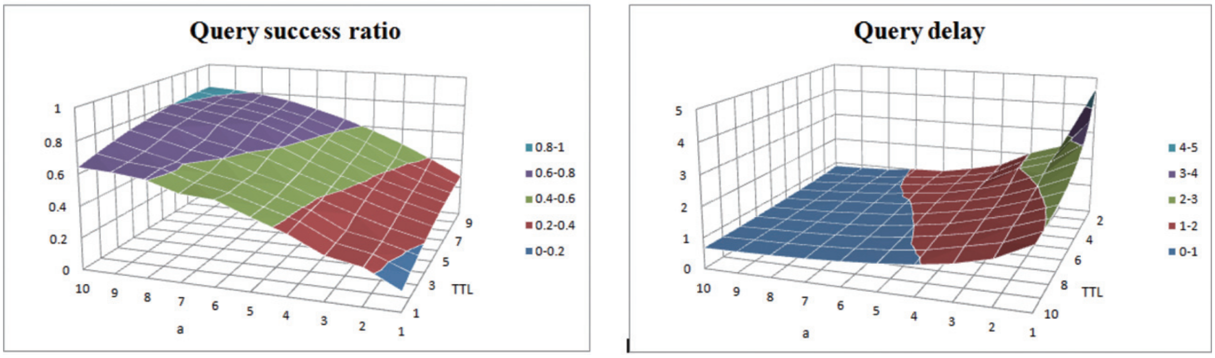
3. Simulation

3.1. Simulation environment

We use the Opportunistic Network Environment (ONE) [29] to evaluate the proposed PQBCF algorithm. The simulation area is a 4500 × 3400 rectangle, where nodes follow the Random Walk (RW) mobility Model. Employing RM model is due to its preferable robustness. The moving speed of each node is within [0.5, 3] m/s. The whole simulation lasts 12 hours. Each node has a buffer size of 5 Mbytes and the size of each message is 512 bytes. The buffer at each node is managed in the FIFO fashion. When the buffer overflows, the first entry would be deleted. We set the sending rate of each node to 250 Kbyte/s. Note that each message has a TTL. When the TTL decreases to 0, the message is automatically discarded. We will evaluate the setting of TTL in our simulation. The detailed simulation settings are shown in Table 2.

At the beginning, nodes are evenly distributed in the network. During the initialization period, each node calculates its BC $C_B\left(P_i\right)$ using local encounter information. In our simulation, we set the length of the initialization period to 3 hours. IMs are generated by the Message Event Generator which comes with ONE. An IM is generated every 15 minutes. The query and the corresponding data are generated in a random fashion across all nodes. The queries from the same node vary from time to time.

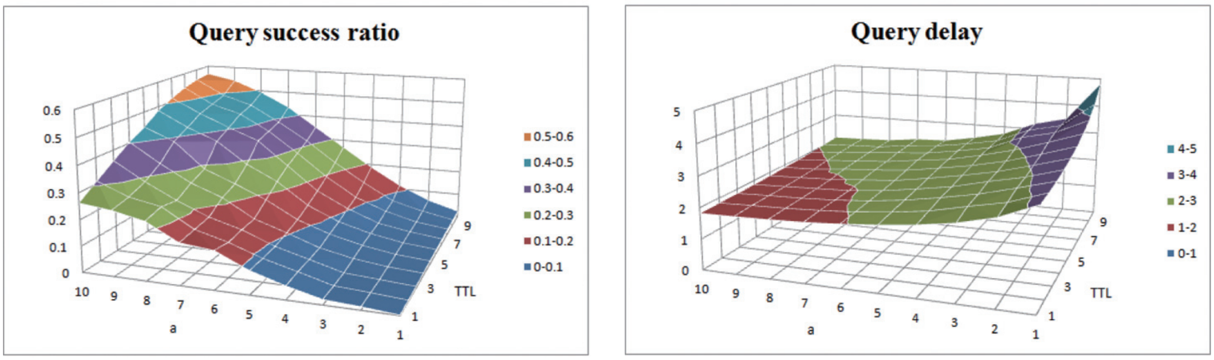
In PQBCF, each node needs to do online estimation for parameters such as N and I. Consequently, each node calculates the values of T_1, T_2 periodically (every 15 minutes) in order to estimate the amount of nodes N_{new} as shown in Eq. (11). The new estimation is then used to obtain a new value of N via a moving average with N_{old} (the old estimation of the number of nodes). This new N is then substituted into Eq. (8) to update the value of L.



(a) Query success ratio

(b) Query delay

Fig. 1. Query success ratio and delay under different settings of a and TTL.



(a) Query success ratio

(b) Query delay

Fig. 2. Impact of a and TTL on PQBCF's performance (range = 30).

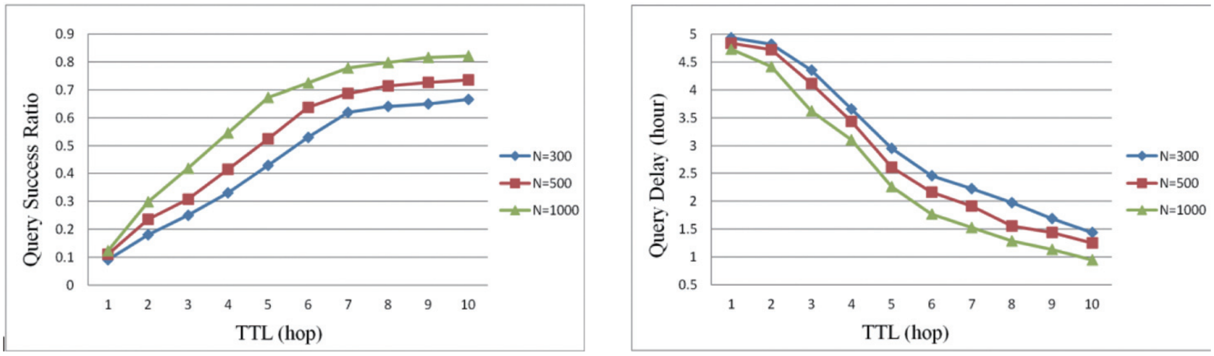
3.2. Simulation results

We evaluate the performance of our PQBCF algorithm with two metrics, i.e., inquiry success rate and inquiry delay. Inquiry success rate is the ratio of IMs which are successfully responded to all generated IMs. Note that each IM carries a TTL field. When the TTL of an IM decreases to 0, the IM is dropped which means that the IM fails. Inquiry delay means the duration between generating an IM and receiving the first correct response message. As for failed inquiries, the inquiry delay is set as 1.5 times of the simulation time, namely 18 hours. The inquiry delay in this experiment is the averaged across all the inquire delays.

3.2.1. The impact of expected delay/TTL/range

As discussed previously, the expected delay ED is set to a times of the theoretic value ED_{opt}, where a is called the expected delay coefficient. In this set of experiments, a increases from 1 to 10. A larger a means the inquiry node expects to receive the response in a longer delay. The TTL value of IMs is selected among 1 to 10 hops. A TTL of 10 means that if an IM fails to meet nodes being able to respond after 10 hops, this IM would be deleted.

Figure 1 shows the query success ratio under different expected delay coefficient a and TTL. The transmission range of each node is set to 100 meters. In Fig. 1, we can see that the query success ratio



(a) Query success ratio (b) Query delay

Fig. 3. Impact of the number of nodes on PQBCF' performance.

grows with the increase of a and also TTL. With $a = 10$ and TTL $= 10$, we can achieve a query success ratio of 82.6%. In practice, the value of a and TTL may vary from one application to another. We can set a small a for emergency scenarios where we only care responses returned in a short duration. This is especially useful when the data has to meet certain timeliness. A long delay will make the received data at the inquiry node out of date. On the other hand, a large a means the query is not in urgency and the inquiry node can wait for a slightly long time to obtain the response. The TTL is used to control the number of IMs in the whole network. A larger TTL makes each IM live longer. As a result, there will be more IMs, which is the main cost of PQBCF. In contrast, if TTL is set too small, IM copies might be deleted soon after generation, resulting in low query success ratio. Figure 2 shows the impact of a and TTL on the performance of inquiry when the transmission range is set to 30 m. In comparison with Fig. 1, we can see a lower query success ratio with higher query delay, which is reasonable since nodes have less chance to communicate with each other. However, they yield the similar trend as that shown in Fig. 1.

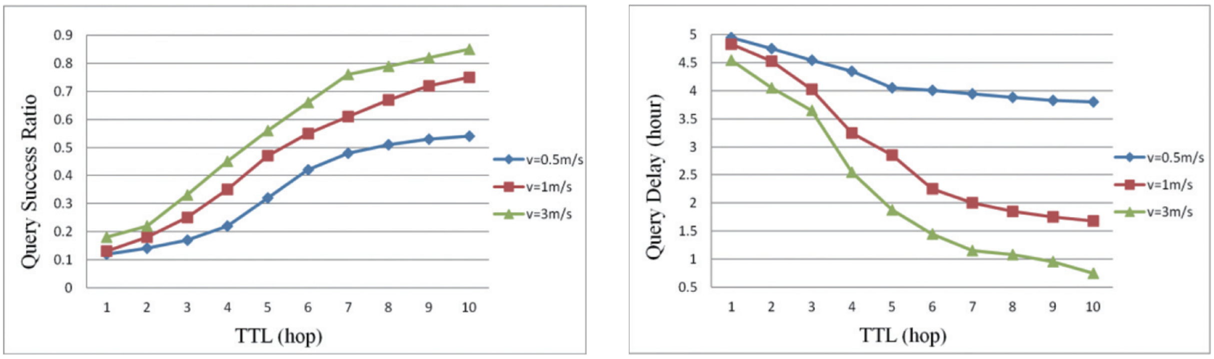
3.2.2. The impact of node density and node's moving speed

Figure 3 shows the impact of the node density in the network on the performance of PQBCF. In this set of simulations, we set $N = 300$, $N = 500$, and, $N = 1000$ respectively. a is set to 6 and the transmission range is set to 100 meters.

Figures 3(a) and (b) show that a higher node density yields a higher query success ratio and a lower query delay, respectively. This is because, with the increase of the total number of nodes, the number of nodes being able to respond to IMs also increases according to the message generation model in our experiment. Another observation in Fig. 3 is that the performance improved by increasing the density of nodes is not significant in comparison with that by increasing the TTL.

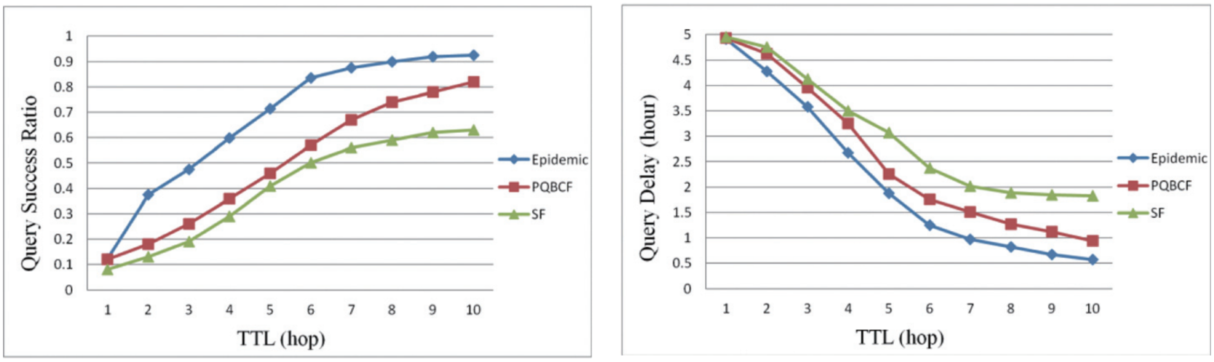
Figure 4 indicates the influence of node's moving speed on the performance of PQBCF algorithm. In this experiment, we examined its performance selectively on $v = 0.5$ m/s, $v = 1$ m/s, $v = 3$ m/s. In this series of experiment, $a = 6$, and the total quantity of nodes is 500 with communication radius of Wi-Fi signal as 100 m.

Figure 4 indicates that with the increase of nodes' moving speed, the inquiry success rate of PQBCF improves accordingly, and vice versa. The higher inquire success rate and decrease in transmission delay result from the following process. With the improvement of nodes' moving speed, the encounter frequency will increase, thus the inquiry message is relayed in higher efficiency based on PQBCF forwarding strategy. Compared to the effectiveness of increasing node number, advancing moving speed



(a) Query success ratio (b) Query delay

Fig. 4. Impact of the nodes' moving speed on PQBCF' performance.



(a) Query success ratio (b) Query delay

Fig. 5. Impact of the TTL on algorithms' performance.

promotes the performance of PQBCF in a larger degree. Since the core strategy of PQBCF is based on message relay utility rather than redundancy in quantity, therefore the increase in speed and further increase in relaying efficiency is more effective for the improvement of performance of PQBCF. Yet similar to increase of node quantity, the increase in nodes' mobility is equally unobvious compared with enhancing TTL.

3.2.3. *Performance comparision*

We compare the performance between PQBCF and several other routing algorithms for opportunistic networks, including Epidemic routing and SF. Epidemic algorithm [4] is a redundant based flooding algorithm where nodes keep copying query messages to all contacts. This algorithm has the highest success rate if no the bandwidth and buffer constrains. However, it consumes too much network resource, resulting in an impractical method in practical scenarios. Spray and Focus algorithm (SF) [28] is a hybrid forwarding algorithm. According to SF, messages are forwarded continuously from nodes with low utility value to higher ones until reaching the destination.

The metric used here are query success ratio and query delay. If nodes in the network have infinite bandwidth and memory size, the Epidemic algorithm absolutely outperforms all others. In our evaluation, Epidemic routing serves as the performance upper bound. SF algorithm first calculates the number of copies L, and then forwards L copies of the same IM to another node that may meet the destination nodes with a higher frequency. Note that the value of the L in PQBCF algorithm and SF algorithm is the same, so the comparison between two algorithms is fair. Our PQBCF algorithm works similar with the SF algorithm except that PQBCF uses the BC of nodes as the utility of forwarding. Figures 5(a) and (b) show the inquiry success rate and inquiry delay for the three different algorithms under different TTL values.

In this set of experiments, $a = 6$, the node transmission range is 100 meters. Figure 5(a) shows that the query success rates of all the three algorithms grow with the increase of TTL. This is because, the greater the TTL is the higher the probability that the IM is forwarded to nodes that can make a successful response to the IM. Because the network resources in our experiment settings are not the bottleneck, Epidemic routing achieves the highest success rate. However, the gap between the Epidemic and our PQBCF is not quite small. The SF algorithm performs the worst among the three. The PQBCF has a significantly higher success ratio than that of the SF algorithm when the TTL is greater than 6.

Figure 5(b) shows the average query delay for the three algorithms with different TTL values. When the TTL is small, the IMs may be discarded before reaching a node with corresponding information leading to a fail query. In Fig. 5(b) we can see our PQBCF outperforms SF and yield a close performance with Epidemic routing.

Figure 6 shows the performance of the three algorithms under different transmission range. In this set of experiments, $a = 6$ and $\text{TTL} = 10$. It can be seen from Fig. 6(a) that, with the increase of transmission range, the inquiry success rate of all three algorithms increased gradually. This is because the increasing transmission range improves the network connectivity, and thus, the IMs can reach the nodes being able to respond with a higher probability. Therefore, the inquiry success rate of the algorithm will be improved accordingly.

In Fig. 6(b), we can see with the increase of range radii of nodes' wireless signal, the inquiry delay of each algorithm has reduced significantly. When the range radiuses approaches 100 meters, the inquiry delay of PQBCF is close to that of the Epidemic algorithm. Compared with the SF algorithm, there is a more significant drop of PQBCF's inquiry delay. This is because, as the range radiuses increases, the network connectivity becomes better, and the nodes forward IMs faster, also forward response messages back to the inquire node in time, therefore, the inquiry delay of the PQBCF algorithm shows a more obvious fall.

4. Conclusion

In this paper, we focus on message acquisition scenarios for opportunistic networks consisting of smart phones. We proposed an efficient P2P query algorithm based on Betweenness Centrality Forwarding (PQBCF) for opportunistic networking. BC is introduced from social networks to indicate the forwarding roles of nodes in data transmission. The PQBCF algorithm selects the nodes with major impacts on data transmission and greater values of BC to forward IMs and response messages. We also design the algorithms to calculate the optimum number of IM and response message copies in the network to meet a certain delay constraint. The simulation experiments show that compared to other algorithms such as the SF algorithm, PQBCF effectively increases inquiry success rate and reduces inquiry delay.

Next, we will implement a prototype system of the PQBCF algorithm and deploy a pilot application on our campus so that it can be evaluated in a real environment.

Acknowledgments

This work was supported by the Research Fund of the State Key Laboratory of Software Development

Environment under Grant No. BUAA SKLSDE-2012ZX-17, the National Natural Science Foundation of China under Grant No. 61170296 and 61190120, the Program for New Century Excellent Talents in University under Grant No. NECT-09-0028.

References

[1] A Community Resource for archiving wireless data, http://crawdad.cs.dartmouth.edu/.

[2] L.A. Adamic and A. Eytan, Friends and Neighbors on the Web, *Social Networks* **25**(3) (2003), 211–230.

[3] R. Baldoni, R. Beraldi and M. Migliavacca, Content-based routing in highly dynamic mobile ad hoc networks, *International Journal of Pervasive Computing and Communications* **1**(4) (2005), 277–288.

[4] V.D. Becker, Epidemic routing for partially connected ad hoc networks, Technique Report, CS-2000-06, Department of Computer Science, Duke University, Durham, NC, 2000.

[5] C.-F. Lai, Y.-M. Huang and H.-C. Chao, DLNA-based Multimedia Sharing System over OSGI Framework with Extension to P2P Network, *IEEE Systems Journal* **4**(2) (June 2010), 262–270.

[6] M. Caliskan and D. Graupner, Decentralized discovery of free parking places, Proc. of the 3rd international workshop on Vehicular ad hoc networks, Los Angeles, ACM Press, 2006, pp. 30–39.

[7] P. Costa and G.P. Picco, Semi-Probabilistic Content-Based Publish-Subscribe in Distributed Computing Systems, *Proc. of the 25th IEEE International Conference ICDCS 2005*, Washington, IEEE Press, 2005, pp. 575–585.

[8] Q.S. Cai and J.W. Niu, Properties of message delivery path in opportunistic networks, *Proc. of the 5th International Conference on Future Information Technology*, Busan, IEEE Press, 2010, pp. 1–6.

[9] S. Eichler, C. Schroth, T. Kosch and M. Strassberger, Strategies for context-adaptive message dissemination in vehicular ad hoc networks, *Proc. of third Annual International Conference on Mobile and Ubiquitous Systems*, California, IEEE Press, 2006, pp. 1–9.

[10] K. Fall, A delay-tolerant network architecture for challenged Internets, *Proc. of 2003 Conf. on Application, Technologies, Architectures, and Protocols for Computer Communications*, Karlsruhe, ACM Press, 2003, pp. 27–34.

[11] L.C. Freeman, Centrality in social networks: Conceptual clarification, *Social Networks* **1**(3) (1978), 215–239, Gossglause.

[12] M. Gossglauser and D. Tse, Mobility increases the capacity of ad hoc wireless networks, *IEEE/ACM Trans. on Networking* **10**(4) (2002), 477–486.

[13] F. Guidec and Y. Maheo, Opportunistic Content-Based Dissemination in Disconnected Mobile Ad Hoc Networks, *Proc. of UBICOMM '07 International Conference*, Papeete, French, 2007, pp. 49–54.

[14] J. Ghosh, S. Philip and C. Qiao, Sociological orbit aware location approximation and routing (solar) in DTN, Technique Report, CSE-2005-12, Department of Computer Science and Engineering, State University of New York, Buffalo, NY, 2005.

[15] B. Hull, V. Bychkovsky and Y. Zhang, CarTel: A distributed mobile sensor computing system, *Proc. of the 4th Int'l Conf. on Embedded Networked Sensor Systems*, Boulder, ACM Press, 2006, pp. 125–138.

[16] P. Juang, H. Oki and Y. Wang, Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet, *Proc. of the 10th ASPLOS*, New York, ACM Press, 2002, pp. 96–107.

[17] A. Lindgren, A. Doria and O. Schelén, Probabilistic routing in intermittently connected networks, *ACM SIGMOBILE Mobile Computing and Communications Review* **7**(3) (2003), 19–20.

[18] C. Liu and J. Wu, Routing in a cyclic mobispace, *Proc. of ACM MobiHoc* New York, ACM Press, 2008, pp. 351–360.

[19] L. Zhou, H.-C. Chao and A.V. Vasilakos, Joint Forensics-Scheduling Strategy for Delay-Sensitive Multimedia Applications over Heterogeneous Networks, *IEEE JSAC* **29**(7) (August 2011), 1358–1367.

[20] I. Leontiadis and C. Mascolo, Opportunistic spatio-temporal dissemination system for vehicular networks, *Proc. of MobiOpp'07*, San Juan, ACM Press, 2007, pp. 39–46.

[21] C. Marta, A. César and B. Albert, Understanding individual human mobility patterns, *Nature* **453** (2008), 779–782.

[22] H. Pan, A. Chaintreau and J. Scott, Pocket switched networks and human mobility in conference environments, *Proc. of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, Philadelphia, ACM Press, 2005, pp. 244–251.

[23] H. Pan, J. Leguay and J. Crowcroft, Osmosis in Pocket Switched Networks, *Proc. of the First International Conference on Communications and Networking in China (CHINACOM 2006)*, Beijing, IEEE Press, 2006, pp. 1–6.

[24] T. Spyropoulos, K. Psounis and C.S. Raghavendra, Spray and wait: An Efficient routing scheme for intermittently connected mobile networks, *Proc. of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN)*, Philadelphia, ACM Press, 2005, pp. 252–259.

[25] T. Spyropoulos, K. Psounis and C.S. Raghavendra, Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility, *Proc. of the IEEE PerCom Workshop on Intermittently Connected Mobile Ad Hoc Networks*, New York, IEEE press, 2007, pp. 79–85.

[26] C.M. Song, Z.H. Qu, N. Blumm and A. Barabási, Limits of Predictability in Human Mobility, *Science Magazine* **327** (2010), 1018–1021.

[27] L.D. Sailer, Structural Equivalence: Meaning and definition, computation and applications, *Social Network* **1**(1) (1978), 73–90.

[28] T. Spyropoulos, K. Psounic and C.S. Raghavendra, Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility, *Proc. Of the IEEE PerCom Workshop on Intermittently Connected Mobile Ad hoc Network*, 2007.

[29] The Opportunistic Network Environment simulator, http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[30] A. Vahdat and D. Becker, *Epidemic Routing for Partially-Connected Ad Hoc Networks*, Technique Report, CS-2000-06, Department of Computer Science, Duke University, Durham, NC, 2000.

[31] L. Wischhof, A. Ebner and H. Rohling, Information dissemination in self-organizing intervehicle networks, *IEEE Transactions on Intelligent Transportation Systems* **6**(1) (2005), 90–101.

[32] O. Wolfson, B. Xu and H.B. Yin, Search-and-Discover in Mobile P2P Network Databases, *Proc. of ICDCS 26th IEEE International Conference*, Lisboa, IEEE Press, 2006, pp. 65–67.

[33] Y.P. Xiong, L.M. Sun, J.W. Niu and Y. Liu, Opportunistic networks, *Journal of Software* **20**(1) (2009), 124–137.

[34] B. Xu, A. Ouksel and O. Wolfson, Opportunistic resource exchange in inter-vehicle ad-hoc networks, *Proc. of Mobile Data Management IEEE International Conference*, California, IEEE Press, 2004, pp. 4–12.

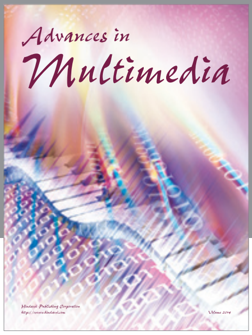
[35] Q. Yuan, L. Cardei and J. Wu, Predict and relay: an efficient routing in disruption tolerant networks, *Proc. of ACM MobiHoc'09*, Louisiana, ACM Press, 2009, pp. 95–104.

[36] J. Yoon, B. Noble, M. Liu et al., Building realistic mobility models from coarse-grained traces, *Proc. of ACM MobiSys 2006*, Uppsala, ACM Press, 2006, pp. 177–193.

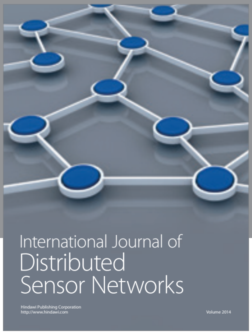
Jianwei Niu received his Ph.D. degrees in 2002 in computer science from Beijing University of Aeronautics and Astronautics (BUAA). He was a visiting scholar at School of Computer Science, Carnegie Mellon University, USA from Jan. 2010 to Feb. 2011. He is a professor in the School of Computer Science and Engineering, BUAA. He has published more than 90 referred papers on such as INFOCOM, TECS, JPDC, and etc., and filed more than 30 patents in mobile and pervasive computing. He served as the Program Chair of IEEE SEC 2008, TPC members of ICC, WCNC, Globecom, IWCMC, and etc. He has served as associate EiC of Int. J. of Ad Hoc and Ubiquitous Computing, associate EiC of Journal of Internet Technology, editors of Journal of Network and Computer Applications (Elsevier). He received New Century Excellent Researcher Award from Ministry of Education of China 2009, Innovation Award from Nokia Research Center, and won the best paper award in CWSN 2012 and the 2010 IEEE International Conference on Green Computing and Communications (GreenCom 2010). His current research interests include mobile and pervasive computing, mobile video analysis.

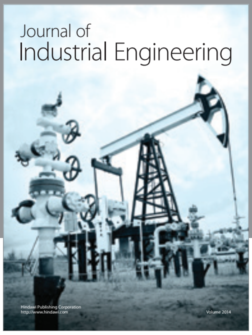
Mingzhu Liu received his B.S. at Beijing Institute of Petrochemical Technology, Beijing in 2007. Now he is a master graduate at Beihang University. His current research interests are DTN and opportunistic networking.

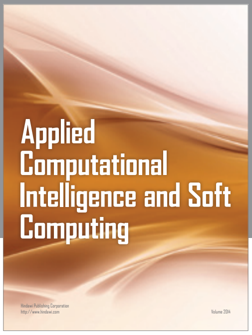
Han-chieh Chao received his MS and PhD at School of Electrical Engineering, Purdue University in 1989 and 1993, respectively. He is a Professor of Department of Electrical Engineering, National Dong Hwa University and Joint Appointed Professor, Department of Electronic Engineering, Institute of Computer Science & Information Engineering, National Ilan University. He is Fellow of IET (IEE) and Fellow & Chartered IT Professional of British Computer Society (BCS). Prof. Chao's research interest include: wireless and mobile computing, high speed networks, IPv6 and digital arts technology.

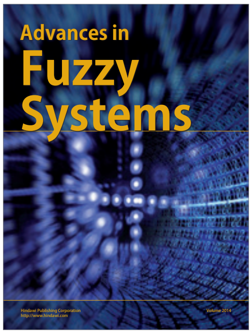

Advances in
Multimedia

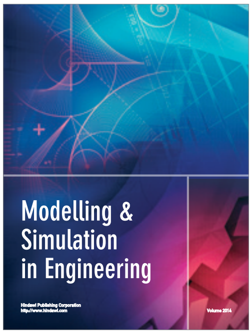

The Scientific World Journal

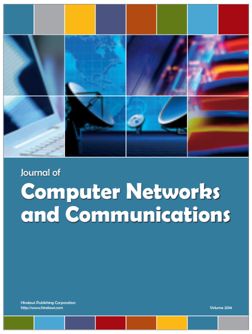

International Journal of
Distributed Sensor Networks

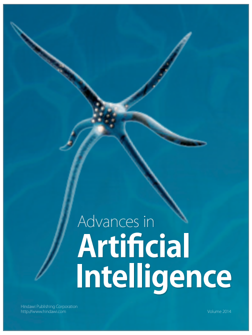

Journal of
Industrial Engineering

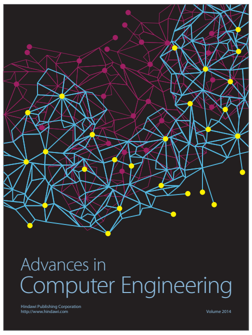

Applied Computational Intelligence and Soft Computing


Advances in
Fuzzy Systems

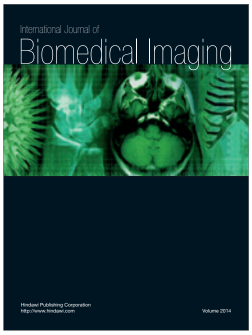

Modelling & Simulation in Engineering

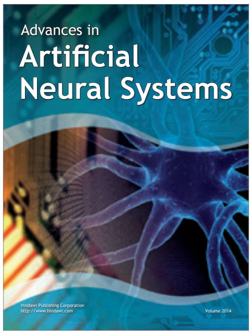

Journal of
Computer Networks and Communications

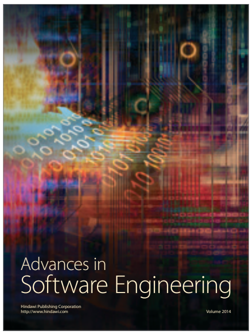

Advances in
Artificial Intelligence


Advances in
Computer Engineering

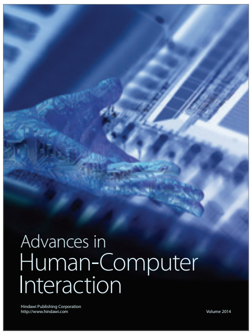

International Journal of
Computer Games Technology

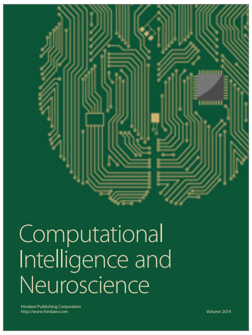

International Journal of
Biomedical Imaging


Advances in
Artificial Neural Systems


Advances in
Software Engineering


Journal of
Robotics


Advances in
Human-Computer Interaction


Computational Intelligence and Neuroscience


International Journal of
Reconfigurable Computing


Journal of
Electrical and Computer Engineering