

P2P-iSN: A Peer-to-Peer Architecture for Heterogeneous Social Networks

Phone Lin and Pai-Chun Chung, National Taiwan University
Yuguang Fang, University of Florida

Abstract

The unprecedented growth and influence of Social Network Sites (SNSs) have opened the opportunity for researchers to explore a large amount of social and behavioral data. The heterogeneity of SNSs further sparks research innovations to develop methods and applications that integrate resources and offer more seamless services across SNSs. Specifically, aiming at the integration of social relationship data, a much less studied subject, we propose a peer-to-peer architecture, namely *P2P-iSN*, to integrate the heterogeneous SNSs. The *P2P-iSN* allows users from heterogeneous SNSs to communicate without involving the SNS they have registered with. Under this architecture, we propose a Global Relationship Model (GRM) to capture the relationship strength between users and then develop a searching mechanism, namely *i-Search*, to find the optimal social path between any two users who are meaningfully connected in heterogeneous SNSs. We evaluate the performance of *P2P-iSN* and show that our *P2P-iSN* can effectively support many future applications such as improved trust/reputation metrics and integrated content-sharing. With the proposed *P2P-iSN*, SNS developers can design more effective user-centric SNS applications.

Social Network Sites (SNSs) such as Facebook™ and LinkedIn™ have transformed today's society by providing easily accessible platforms for users to connect, communicate, and share vast amount of information. With SNSs, people keep in touch with their contacts, reconnect with old acquaintances, and establish new relationships with others based on shared features such as hobbies, interests, and overlapping friendship [1]. The population of SNS users and the number of SNSs have been growing rapidly. For example, Facebook™ is estimated to have over one billion active users [2]. Although it is hard to obtain an accurate estimate, there are thousands of SNSs all over the world that provide different kinds of services.

As a result, a user may register with multiple SNSs for different social network applications, carry multiple SNS accounts, interact with contacts from different SNSs, publish and access different web contents, and share contents within each SNS community. While SNSs offer different services, one key feature shared among SNSs is how they are built around users and users' existing social networks [3, 4]. Yet each SNS is isolated, so users manage their profiles and build relationship separately on different SNSs. The content for the same user in different SNSs may overlap, so it becomes a burden for users to manage contents across different SNSs. This is the landscape of heterogeneous SNSs.

With growing influence and complexity of SNSs, researchers have been developing methods to connect users and aggregating data across SNSs so that each SNS no longer stands alone. For example, the study in [5] summarizes how social networks connect services allow users to leverage their information on multiple SNSs, from using single ID to access multiple SNS

accounts to publish contents simultaneously on multiple SNSs. However, the aggregation of social relationship data has not been well studied, which motivates this article. Here, we propose a system architecture to integrate heterogeneous SNSs and investigate a model to characterize the social relationships among a large number of users across heterogeneous SNSs.

To be more specific, we offer the following example. Imagine that within an SNS, user *b* is in user *a*'s friend list. We say that there is a directional social link denoted by " $a \rightarrow b$ " from user *a* to user *b*. Building with these directional links, users in an SNS form a social graph [6]. When there exists a social path between two users in an SNS, we say that there is a "relationship" between these two users, and more precisely, from the source to the destination. We say that there is a "global relationship" from user *a* and user *b* if there is a social path from *a* to *b* over heterogeneous SNSs. These are basic notations that are used and elaborated in our model. By identifying "global relationship" among users over heterogeneous SNSs, this article opens the possibility for users from different SNSs to interconnect their various networks, resulting in an integrated network of interconnected heterogeneous SNSs, which enable users to communicate with a larger community of users. A comprehensive study of this integrated SNS landscape can help SNS developers design user-centric SNS applications with more features [3].

In this article, we first propose a peer-to-peer (P2P) network, namely *P2P-iSN*, to integrate heterogeneous SNSs and establish global relationships over the integrated SNSs. As shown in Fig. 1, *P2P-iSN* consists of two kinds of nodes: *peer nodes* and *index peer nodes*.

A peer node is installed on an end-device (e.g., PDA or desktop) for a user to access SNSs, and its main functionality is to integrate heterogeneous SNSs. The user of a peer node may register to one or more SNSs on his end-device, and possibly login to one or more SNSs at the same time. To associate these different accounts of the same user from heterogeneous SNSs, a unique user ID may be required. The concept known as OpenID¹ in [7] can serve this purpose although any other uniquely identifiable ID can be used. A unique user ID can be some kind of authenticated information like user's cell phone number or verifiable email address. The index peer node is responsible for maintaining the status (i.e., online or off-line) and the routing information (i.e., IP address) of each peer node. Here is the sketch of the operations over P2P-iSN. When a peer node is turned on, it reports to the index peer node the online status, which consists of its ID and IP address of the peer node. Upon receiving the online status, the index peer node updates the online status for the peer node. If a user *a* of the peer node **na** and a user *b* of the peer node **nb** are on each other's friend list in a SNS, and **na** and **nb** are turned on, these two online peer nodes can communicate with each other by using the corresponding IP addresses queried from the index peer node. The peer nodes can establish social paths among users from different SNSs and build the so-defined "global relationship." We describe the details of P2P-iSN later.

With the peer-to-peer network architecture, P2P-iSN allows users from heterogeneous SNSs to communicate without involving any specific SNS, and the integration is independent of any specific SNS. In other words, the integration does not incur overhead to the SNSs. By applying P2P-iSN, we develop a Global Relationship Model to assess the strength of the global relationship between two users from heterogeneous SNSs. Based on the global relationship model, we propose a searching mechanism, namely *i-Search*, to find the social path between two users from heterogeneous SNSs. We also develop an analytical model to evaluate the performance of the *i-Search* mechanism in terms of the "path found" probability and conduct extensive simulation studies to validate our analytical results.

The rest of the article is organized as follows. We describe the design and implementation of P2P-iSN. We propose the global relationship model and the corresponding searching mechanism, the *i-Search*, to identify the global relationship between two users in the heterogeneous SNSs. We develop an analytical model and conduct simulations to study the performance of P2P-iSN. We then conclude this article.

P2P-iSN

In this section, we propose a peer-to-peer architecture, namely *P2P-iSN*, to integrate heterogeneous SNSs. P2P-iSN consists of two kinds of nodes: *peer nodes* and *index peer nodes*.

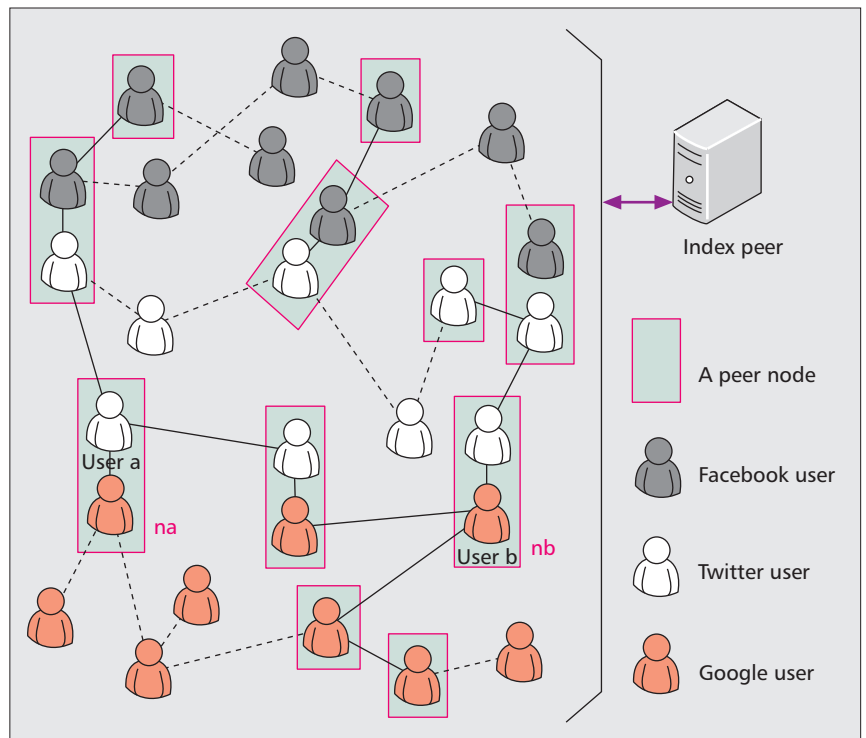


Figure 1. System architecture of the proposed P2P-iSN.

The main functionality of a peer node is to integrate the heterogeneous SNSs through the Friend List maintenance (to be elaborated later). Peer nodes communicate with each other directly and form a peer-to-peer network. An index peer node maintains the status and the IP address of the peer node. The design and implementation of two kinds of nodes are described in the following subsections.

Peer Nodes

A peer node is installed on an end device (e.g., PDA or desktop) used by a user to access an SNS. A user may register to one or more SNSs on his end device, and login to one or more SNSs at the same time. Because a user may use different IDs in different SNSs, to associate these different accounts of a user, a unique user ID is required. This concept is similar to OpenID in [7]. The unique user ID can be a user's cell phone number or email address, which can be used to uniquely identify this user. In this article, we use the cell phone number as an example for the unique ID. The phone book in a peer node (e.g., Jenny's end device) is used as the base to integrate heterogeneous SNSs. Take Fig. 2a as an example. Jenny has a friend John with phone number "0910456."

A database, *friend list*, is maintained to store the information about a user's friends. Figure 2b shows the format of a friend list. The friend list consists of three kinds of information: personal information, social network information, and address information.

Personal information field stores the IDs of the user's friends, including the ID in SNS, phone number, and email address. In different SNSs, users may use different IDs. As shown in Fig. 2b, for example, Jenny's friend, John, uses the ID, "John_f", on Facebook™ and use the ID, "John_t", on Twitter™. The phone number associates the entry in the phone book with the entry in the friend list. An entry in the phone book may be mapped to multiple entries in the friend list.

Social Network Information field consists of four subfields, including SN Type, T Value, Timestamp, and Online. The SN Type indicates which SNS the friend has registered. For exam-

¹ OpenID is a protocol that authenticates a user's digital identity. A user could register on any one of the Identity Providers, which are websites that handle user authentication, such as Facebook™, Google™ and MySpace™, etc. Once the register describes the identity of the user, the user, carrying the same ID, can browse all websites that support OpenID.

Personal information

Social network information

Address information

ID	Phone no.	ID	Phone no.	Email	SN type	T value	Timestamp	Online	IP	Port
John	0910456	John_f	0910456	John@gmail.com	Facebook	0.9	11'1211	On_12'0215_1430	140.112.5.5	12345
Bob	0910123	John_t	0910456	John@gmail.com	Twitter	0.85	12'0214	Off_12'0214_1430	Null	Null
		Bob_f	0910123	Bob@gmail.com	Facebook	0.6	12'0110	On_12'0209_1000	140.112.6.6	11100
⋮		⋮								
⋮		⋮								

(a)

ID	Phone no.	Email	SN type	T value	Timestamp	Online	IP	Port
John_f	0910456	John@gmail.com	Facebook	0.9	11'1211	On_12'0215_1430	140.112.5.5	12345
John_t	0910456	John@gmail.com	Twitter	0.85	12'0214	Off_12'0214_1430	Null	Null
Bob_f	0910123	Bob@gmail.com	Facebook	0.6	12'0110	On_12'0209_1000	140.112.6.6	11100
⋮								
⋮								

(b)

Figure 2. An example of friend list: a) Jenny's phone book; b) Jenny's friend list.

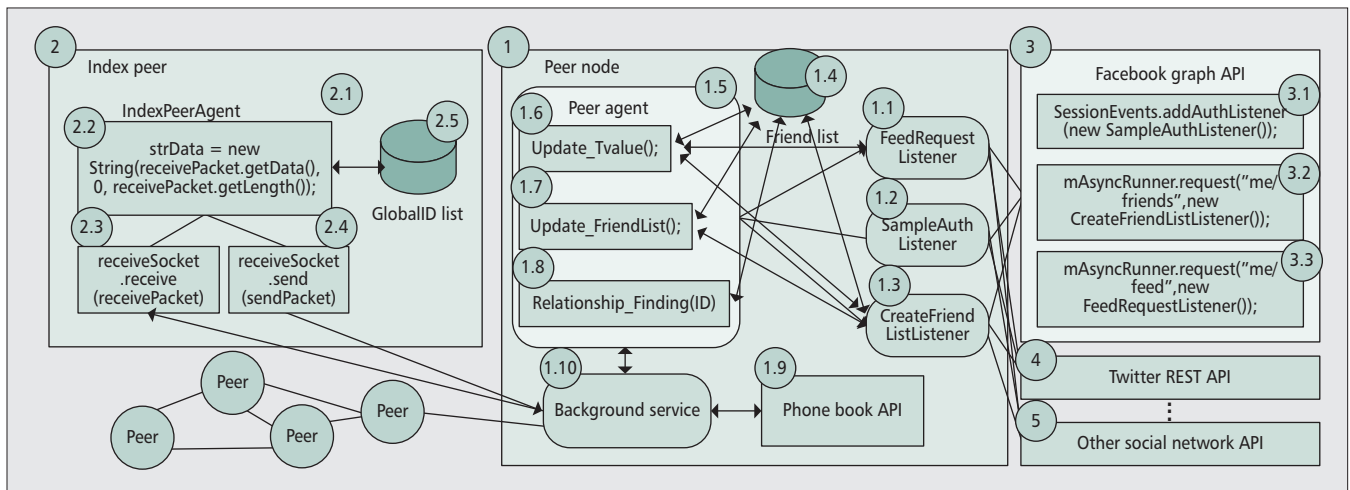


Figure 3. Software architecture of a P2P-iSN.

ple, in Fig. 2b, Jenny's friend, John, registered to Facebook™ using ID "John_f." The T Value stores the result calculated by using Eq. 1 in the global relationship model that will be elaborated later, indicating the frequency that a user performs some kind of social activities on his friend (e.g., Jenny posts a comment, click a "Like", or send a message on John's wall in Facebook™). For example, in (1) in Fig. 2b, the T Value for Jenny ← John on Facebook™ is 0.9. The Timestamp field stores the time when the T Value was calculated. The Online indicates that whether the friend is on the SNS now or not and when John logs in to the Facebook™ last time. If the value of Online is "On" ("Off"), the time is when John logs in (logouts) Facebook™. For example, in Fig. 2b, "On_12'0215_1430" implies that John f logs in Facebook™ at 14:00 on Feb. 15, 2012, and is now on Facebook™.

Address Information field stores the IP address and the port number of the friend's end-device. This information is valid when the peer node of the friend is turned on.

Figure 3 (1) shows the software architecture of a peer node. A peer node consists of five classes and a function, **PeerAgent** (Fig. 3 (1.5)), **FeedRequestListener** (Fig. 3 (1.1)), **SampleAuthListener** (Fig. 3 (1.2)), **CreateFriendListListener** (Fig. 3 (1.3)), **BackgroundService** (Fig. 3 (1.10)), and a **Phone Book API** (Fig. 3 (1.9)). The details for the five classes are given below:

The **FeedRequestListener** class (Fig. 3 (1.1)) is responsible

for getting the status of a user's social activities on SNS by invoking the API, `mAsyncRunner.request("me/feed", new FeedRequestListener())` (Fig. 3 (3.3)), provided by the SNS (e.g., Facebook Graph API [8]).

The **SampleAuthListener** class (Fig. 3 (1.2)) is responsible for authenticating a user when he turns on the peer node and login an SNS. The **SampleAuthListener** class is implemented by using the API `SessionEvents.addAuthListener(new SampleAuthListener())` (Fig. 3 (3.1)) provided by the SNS.

The **CreateFriendListListener** class (Fig. 3 (1.3)) is responsible for obtaining the IDs of a user's friends in an SNS by invoking the API, `mAsyncRunner.request("me/friends", new CreateFriendListListener())` (Fig. 3 (3.2)), and maintaining the user's friend list.

The **BackgroundService** class (Fig. 3 (1.10)) is responsible for the message exchange between two peer nodes and between a peer node and an index peer node. The class provides the communication channel among peer nodes for the i-Search mechanism. To be more specific, a peer node uses this class to request another peer node to execute the i-Search mechanism to be elaborated later. The peer node uses this class to inform his online status to the index peer node.

The **PeerAgent** is the main class (Fig. 3 (1.5)). There are three functions defined in **PeerAgent** including the `Update_TValue()` function (Fig. 3 (1.6)), the

Update_FriendList() function (Fig. 3 (1.7)), and the Relationship_Finding() function (Fig. 3 (1.8)). The Update_Tvalue() function and the Update_FriendList() function are used to respectively update the T Value and Online field in the friend list. The Relationship_Finding() function implements the iSearch mechanism to identify the directional social path between two users.

The **Phone Book API** is used to fetch a user's phone book friends. Several smart phone operating system provides such an API, for example, Android API [9]. It is executed in the login procedure and will be elaborated later. By using the phone number, we can identify two or more accounts of the same user to integrate different SNSs.

Index Peer Node

An index peer node is a database that maintains the GlobalID list with the format as shown in Fig. 4. For each online peer node, an entry is created in the GlobalID List for the peer node. Similar to the Friend List, the GlobalID List consists of three kinds of information: Personal Information, Social Network Information, and Address Information for an online user.

The **Personal information** field stores the IDs of a user, including the ID in SNS used by the user to login an SNS, phone number, and email address. Note that a user may turn on a peer node by logging into one or more SNSs concurrently, there may be one or more SNS IDs for the same user (i.e., multiple entries for the same user exist in the GlobalID List). These multiple entries are linked using the same phone number (or email address) of the user. Note that in our implementation, we may select only one of the IDs for a user to be stored in the index peer node (e.g., the user's phone number) so that the IDs for a user can be kept unknown to the index node.

The **Social Network Information** field stores the SNS Type indicating which SNS the user logs in currently (i.e., online).

The **Address Information** field stores the IP address and the port number of the peer node turned on by the user. This information is valid when the peer node is turned on.

Figure 3 (2) shows the software architecture of the index peer node. There are one main class **IndexPeerAgent** (Fig. 3 (2.1)) and a database GlobalID List (Fig. 3 (2.5)). In the main class **IndexPeerAgent** (Fig. 3 (2.1)), the receiveSocket.receive() function (Fig. 3 (2.3)) is executed to receive the message sent from a peer node. Upon receiving a message, the receivePacket.getData() function (Fig. 3 (2.2)) is invoked to get the information carried in this message. The receiveSocket.send() function (Fig. 3 (2.4)) is responsible for sending the response message to a peer node.

Turning on A Peer Node

This section describes the execution of a peer node. When a user turns on the peer node on his end device, the Login procedure is executed. Figure 5 illustrates the message flow for the Login procedure with the following steps:

Step 1. When a user turns on the peer node, a **SampleAuthListener** class is created, and the SessionEvents.addAuthListener(new SampleAuthListener()) function is exercised to authenticate the user in an SNS.

ID	Phone no.	Email	IP	Port	SN type
John_f	0910123	John@gmail.com	140.112.5.5	12345	Facebook
Bob_f	0910456	Bob@gmail.com	140.112.6.6	11100	Facebook
Bob_t	0910456	Bob@gmail.com	140.112.6.6	11100	Twitter
Jenny_f	0910789	Jenny@gmail.com	140.112.7.7	16161	Facebook
Jenny_t	0910789	Jenny@gmail.com	140.112.7.7	16161	Twitter
•					
•					
•					

Global ID list

Figure 4. An example of GlobalID list.

Step 2. If the authentication is successful, the SNS responds with the user SNS ID in the return of the SessionEvents.addAuthListener() function.

Step 3. The peer node creates a **BackgroundService** class to send a message (i.e., the User_Online_Message message) carrying the user's ID, Phone No., Email, IP address, port number, and SN Type, to the index peer node. The index peer node creates an entry for the user in the global ID list.

Steps 4 and 5. The peer node creates a **CreateFriendListener** class (i.e., the FriendList_Request and FriendList_Response message pair) to get the IDs of the user's friends from the SNSs, and creates an entry for each friend in the Friend List.

Steps 6 and 7. The peer node uses the **BackgroundService** class to send a message (i.e., the Friends_OnlineStatus_Request and Friends_OnlineStatus_Response message pair) to the index peer node to query the online friends of the user.

Steps 8 and 9. The peer node creates a **FeedRequestListener** class to collect the social activity information to calculate the T value from the SNS by exchanging the T Value_Parameter_Request and the T Value_Parameter_Response message pair.

The Global Relationship Model

In this section, based on P2P-iSN, we propose the Global Relationship Model to identify the global relationship between two users across heterogeneous SNSs. We first propose a tool to measure the global relationship strength between any two users across heterogeneous SNSs. Then we propose an i-Search mechanism to find a meaningful directional social path between two peer nodes in P2P-iSN.

The Strength of Global Relationship

Before we actually search for users' global relationships, we need a tool to measure the relationship strength between any two users across heterogeneous SNSs. We modify the decay function defined in classical sociology on network relations [10, 11] to come up with more precise measurements on global relationship strength in heterogeneous SNSs.

A directional social link $a \rightarrow b$ is associated with *frequency*, which is denoted by $f(a, b)$, to capture how often a user a performs some kind of social activities with user b (e.g., a posts a comment on b 's wall, click a "Like", send a message, or makes a call to b). Consider there are C kinds of social activities. For $1 \leq i \leq C$, let λ_i denote the frequency that a user a performs the i th kind of activity with user b . We define $f(a, b)$ by

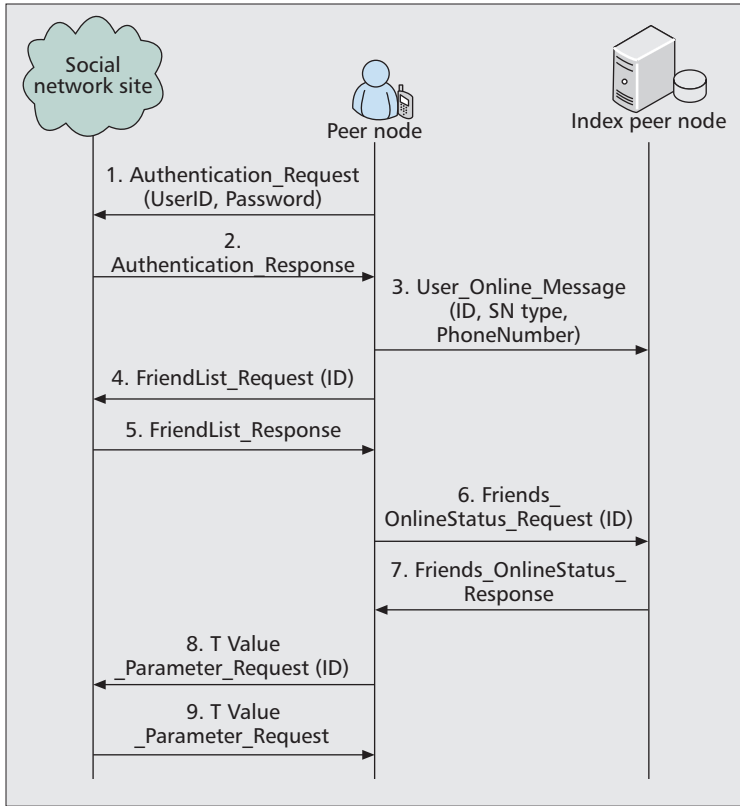


Figure 5. The message flow for the login procedure.

$$f(a,b) = \sum_{i=1}^C w_i \lambda_i$$

where w_i is the weight for the i th kind of activity, $0 \leq w_i \leq 1$ for $1 \leq i \leq C$, and $\sum_{i=1}^C w_i = 1$. Note that in Eq. 1, the weight w_i is a fine-tuning tool to reflect different degrees of interactions in a relationship. For example, while clicking a “Like” often carries a more casual connotation, sending an email message implies stronger intention to communicate with another user, so we can use a larger w_i to the latter kind of social activity. As for λ_i , it can be obtained from the measurement in an SNS for a given time period (e.g., per month or per day). In the directional social link $a \rightarrow b$, a larger $f(a,b)$ value implies that user a pays more attention to user b . For example, assume there is only one kind of social activity (i.e., $w_1 = 1$), comment posting. If a user a posts five comments on average per day on b ’s wall in the SNS (i.e., $\lambda_1 = 5/\text{day}$), then $f(a,b) = w_1 \lambda_1 = 5/\text{day}$. We use a threshold θ to bound $f(a,b)$. In other words, if $f(a,b) \geq \theta$, we say that user a has enough attention on user b .

We say that an *interaction factor* exists between user a and user b if $a \rightarrow b$ and $b \rightarrow a$ exist, and the value $F(a,b)$ for an interaction factor is defined by

$$F(a,b) = \left(\frac{\min\{f(a,b), \theta\}}{\theta} \right) \left(\frac{\min\{f(b,a), \theta\}}{\theta} \right)$$

A larger $F(a,b)$ means more interactions between user a and user b , and from Eq. 2, we have $0 \leq F(a,b) \leq 1$ and $F(a,b) = F(b,a)$.

Consider a social graph formed by heterogeneous SNSs. For example, in Fig. 6, there are two SNSs in a social graph. Suppose that a directional social path exists from user u_1 (in the SNS, S_1) to user u_{L+1} (in the SNS, S_2) through users,

u_2, u_3, \dots, u_L , where at least one of the $L+1$ users is a peer node. Denote the directional social path as a set of links, “ $\mathbf{P} = \{u_1 \rightarrow u_2, u_2 \rightarrow u_3, \dots, u_{L-1} \rightarrow u_L, u_L \rightarrow u_{L+1}\}$.” This directional social path consists of L directional links (i.e., the distance between u_1 and u_{L+1} is $|\mathbf{P}| = L$). To convey this directional social path, we say that a *global relationship* exists between u_1 and u_{L+1} . We propose a function $Z(\mathbf{P})$ to measure the strength of the global relationship between u_1 and u_{L+1} , which is defined by

$$Z(\mathbf{P}) = \begin{cases} 1, & \text{if } L = 0; \\ \prod_{i=1}^L F(u_i, u_{i+1}) & \text{otherwise (i.e., } L \geq 1). \end{cases}$$

From Eq. 2, we have $0 \leq F(u_i, u_{i+1}) \leq 1$ and $F(u_{i+1}, u_i) = F(u_i, u_{i+1})$ for $1 \leq i \leq L$. Then $0 \leq Z(\mathbf{P}) \leq 1$. Furthermore, for the reverse directional path \mathbf{P}' of \mathbf{P} (i.e., $\mathbf{P}' = \{u_{L+1} \rightarrow u_L, \dots, u_3 \rightarrow u_2, u_2 \rightarrow u_1\}$), we have $Z(\mathbf{P}') = Z(\mathbf{P})$. A larger $Z(\mathbf{P})$ implies stronger global relationship. The strength $Z(\mathbf{P})$ is a powerful tool that can provide more precise friend recommendation and trust/reputation metrics, and also serve as a basis for content sharing across SNSs, just to name a few.

Once we can characterize the global relationship strength between any two users, we can then search for a directional social path from user a and user b .

In this article, based on the $Z(\cdot)$ function, we propose the *i-Search* mechanism in the next section for social path searching in the P2P-iSN network, which is also considered as one of our major contributions in this article.

The i-Search Mechanism

In this section, we propose an *i-Search* mechanism to find a directional social path between two peer nodes in P2P-iSN. Although searching in a social graph has been studied in the previous works [12], most of these studies are centralized in the sense that a social graph is well maintained at a central node. Fewer studies have addressed distributed search over a P2P social network, which is the main focus of this article.

The *i-Search* mechanism is similar to the flooding search that has been widely adopted in communications networks (e.g., [13]). The *i-Search* mechanism establishes social paths link by link. When a link is added into a path, global relationship strength is calculated for the new path using the $Z(\cdot)$ function in Eq. 3. If the global relationship strength for the new path is below a threshold Δ , the social path search stops. Note that Δ is used to guarantee that the global relationship strength for the constructed path is strong enough so that users are motivated to use the global social relationship for further SNS applications.

The past social network research findings in the sociology [11] indicated that $F(a,b) = 0.5$. However this may not be true anymore for online social networks. In online social networks, $F(a,b)$ would be much high than 0.5. In this study, for demonstration purpose, we set up Δ based on the research findings in the sociology (i.e., the interaction factor for link $a \rightarrow b$ is $F(a,b) = 0.5$). If we consider a path \mathbf{P} with length $|\mathbf{P}| = 4$, then using the $Z(\cdot)$ function in Eq. 3, the global relationship strength for the path is $Z(\mathbf{P}) = 0.5^4 = 0.0625$, which is considered very weak relationship. Therefore, in the performance study later, we set $\Delta = 0.5^3 = 0.125$.

In other words, it is likely that the social path (searched by the i-Search mechanism) has path length no larger than 3. As mentioned in [14], with path length no larger than 3, the flooding search is considered with low complexity. This is the main reason why we use the flooding search. In fact, as long as the interaction factor for link $a \rightarrow b$ is $F(a, b) \leq \beta < 1$, the global relationship strength will be exponentially decreasing, and hence the flood search should have low complexity.

Details of the i-Search mechanism are given below: The index peer node maintains the online status (including the ID and IP address of the peer node) for the online peer nodes. A friend list is maintained in the peer node, which stores the online information for all friends of the peer node. To simplify our description, we use “the friend b of a peer node a ” to imply that the social link $a \rightarrow b$ exists.

When a peer node is turned on, it reports its online status to the index peer node, and receives the latest online status for his friends from the index peer node. With the latest online information, the peer node can determine whether his friend is online (i.e., a peer node is turned on). A online peer node can communicate with his online friends directly. We run a recursive algorithm, iSearch, in the peer node (Algorithm 1). In this algorithm, the set \mathbf{G} is the friend list of a peer node. The input parameter s stores the ID of the peer node who calls Algorithm 1, and r is the ID of the peer node to be searched. Initially, we set $\mathbf{P} \leftarrow \emptyset$.

Consider the scenario where the peer node a searches for the peer node d . A user a can “request” his friend b to execute the iSearch algorithm (i.e., $b.iSearch()$ in Algorithm 1) through the direct communication if b is online. That is, the directional social path \mathbf{P} is established along the online peer nodes.

Note that the i-Search mechanism may find multiple global social relationships between two peer nodes. For the peer node who triggers the i-Search mechanism, he can use the one with the largest global social relationship strength. Furthermore, we can speed up the execution of the i-Search mechanism by caching the searching results on the peer nodes. To simply our discussion, we do not include the study for the effects of the cache in this article.

Performance Analysis

All peer nodes and the corresponding social links in P2P-iSN form a social graph. A peer node may be turned on or off during the execution of i-Search, and the iSearch request can reach the friends only when the friends are online. In other words, a social link $a \rightarrow b$ does not exist if peer node a or b is turned off (i.e., off-line). Therefore, the physical network topology of P2P-iSN changes dynamically when the i-Search mechanism is being executed.

Let P_f be the “path found” probability that a directional social path exists when a peer node a executes the i-Search mechanism to find a peer node d . The online status of a peer node affects the P_f significantly. In this section, we propose an analytical model to obtain an approximation value for P_f .

To simplify our discussion, we assume that the peer nodes in P2P-iSN are independently and identically distributed (i.i.d) in terms of network behaviors such as online status, interac-

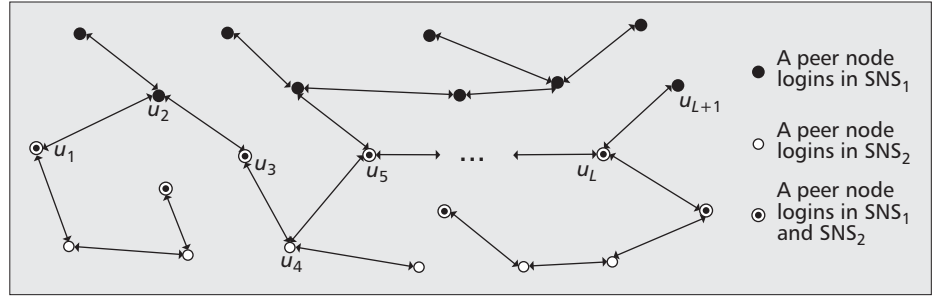


Figure 6. An example for a social graph over heterogeneous SNSs

```

Input:  $s, r, \mathbf{P}, Z(\mathbf{P})$ 
Output:  $\mathbf{P}_{new}, Z(\mathbf{P}_{new})$ 
1 foreach  $v : v \in \mathbf{G} - \mathbf{P}$  do
2   if  $v = r$  then
3      $\mathbf{P}_{new} \leftarrow \mathbf{P} \cup \{s \rightarrow v\};$ 
4      $Z(\mathbf{P}_{new}) \leftarrow Z(\mathbf{P})F(s, v);$ 
5     return;
6   else if  $v$  is online, and  $Z(\mathbf{P})F(s, v) > \Delta$  then
7      $v.iSearch(v, r, \mathbf{P} \cup \{s \rightarrow v\}, Z(\mathbf{P})F(s, v));$ 
8   else if  $v$  is off-line, or  $Z(\mathbf{P})F(s, v) \leq \Delta$  then
9     quit;
10  end
11 end

```

Algorithm 1. iSearch.

tions, etc. As discussed earlier, in this article, we set $\Delta = 0.5^3 = 0.125$ in the i-Search mechanism. In our analytical model, we use the constraint $|\mathbf{P}| \leq 3$ instead of $\Delta \leq 0.125$, i.e., the i-Search mechanism quits when the path length reaches 3 with conclusion that no global social path is found.

Assume that a peer node is turned on (i.e., online) for a time period x (with the density function $f_x(\cdot)$ and mean $1/\mu_x$), and then it is turned off (i.e., off-line) for a time period y (with the density function $f_y(\cdot)$ and mean $1/\mu_y$). The peer node alternates between x and y . Suppose that iSearch request arrivals to a peer node form a Poisson process. Then according to the alternating renewal process [15], the probability p_{on} that an iSearch request arrives when a peer node is online can be obtained by

$$p_{on} = \frac{E[x]}{E[x] + E[y]} = \frac{\mu_y}{\mu_x + \mu_y} \quad (4)$$

We generate the social graph for P2P-iSN using the Watts-Strogatz model [16] with the three parameters α (i.e., the rewiring probability), n (i.e., the total number of peer nodes in P2P-iSN), and m (i.e., the average number of friends of a peer node). With the setup:

$$0 < \alpha < 1 \text{ and } n \gg m \gg \ln n \gg 1 \quad (5)$$

the Watts-Strogatz model has the small-world property, including small average path length and high clustering, which can also be applied to study SNS [3].

Let N_t denote the expected number of the peer nodes that receive the i-Search request message during the execution of the i-Search mechanism. Consider the scenario that the peer node a executes the i-Search mechanism to search for a directional social path to d . If d belongs to one of the N_t peer nodes, then the directional social path from a to d is found. Therefore, we have

$$P_f = \frac{N_t}{n}.$$

We derive N_t as follows. There are two types of nodes including “far-nodes” and “near-nodes” defined in the Watts-Strogatz model. The far-nodes represents the peer nodes that have social links after rewiring with probability α . The near-nodes represents the peer nodes that have social links initially.

In the social graph of the P2P-iSN, let N_f and N_n , respectively, be the expected numbers of far-nodes and near-nodes that receive an i-Search request when the i-Search mechanism is executed. Then we have

$$N_t = N_f + N_n.$$

N_f and N_n are obtained as follows. One round means that the i-Search request is delivered using a directional social link $a \rightarrow b$ when both peer nodes a and b are online. In the i-Search mechanism, there are at most three rounds to construct a directional social path. In each round, a peer node that triggers the round can be either a far-node or near-node:

Case 1 The peer node that triggers the round is a far-node.

In this case, there are on average $m\alpha p_{on}$ far-nodes and $m(1 - \alpha)p_{on}$ near-nodes that can receive the i-Search request.

Case 2 The peer node that triggers the round is a near-node. Because there is high probability that the near-node sends the i-Search request to another near-node that has received this i-Search request previously, we consider that only far-nodes can receive the i-Search request for the approximation. In this case, there are on average $m\alpha p_{on}$ far-nodes that can receive the i-Search request.

We use the following iterative procedure to calculate the N_f and N_n .

Procedure 1.

Input parameters: α, m, μ_x, μ_y .

Output measures: N_f, N_n, N_t .

Step 1. Select initial values, $N_f \leftarrow 1, N_n \leftarrow 0$, and $round \leftarrow 0$;

Step 2.

$$N_f \leftarrow m\alpha \left(\frac{\mu_y}{\mu_x + \mu_y} \right) (N_f + N_n);$$

$$N_n \leftarrow m(1 - \alpha) \left(\frac{\mu_y}{\mu_x + \mu_y} \right) N_f; round ++.$$

Step 3. If ($round \leq 3$), then go to Step 2. Otherwise, go to the next step.

Step 4. $N_t \leftarrow N_f + N_n$; return.

The analytical model is validated by simulation experiments of a discrete event-driven simulation model, which has been widely adopted to simulate the mobile communications network in several studies (e.g., [17]). The simulation model simulates the online/off-line behavior of a peer node and the behavior of the i-Search mechanism.

In the simulation model, we adopt the aforementioned discrete event-driven approach [17] and define five types of events listed below:

- The **QUERY_ARRIVAL** event represents that an online peer node starts the i-Search mechanism to find another peer node.
- The **QUERY_FORWARD** event represents that an online peer node sends a i-Search request to his online friend.
- The **QUERY_RESPONSE** event represents that an online peer node returns the results (i.e., a path is found) for the

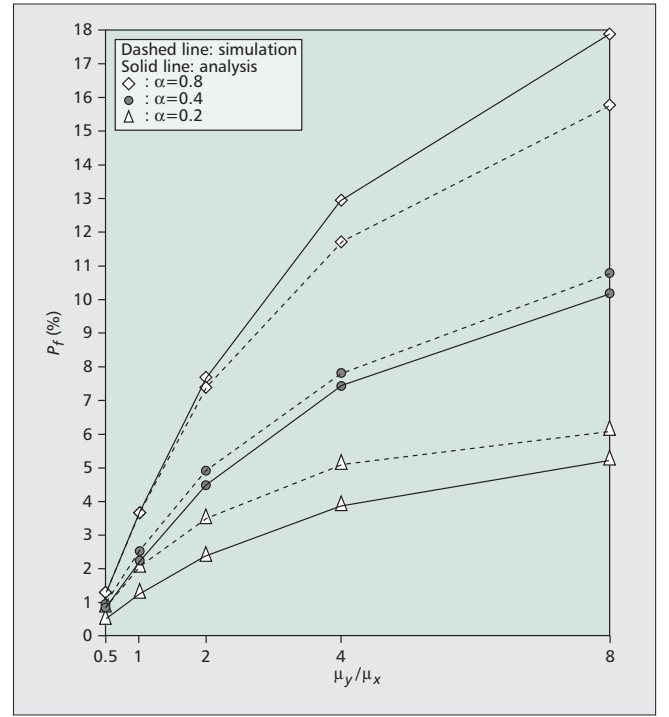


Figure 7. Effects of ($n = 1000$; $m = 6$).

execution of the i-Search algorithm to the peer node who sends the i-Search request.

- The **ONLINE** event represents that a peer node is turned on.
- The **OFFLINE** event represents that a peer node is turned off.

We maintain a timestamp t_s to indicate the time when an event occurs. The events are inserted into an event list and deleted/processed from the list in a non-decreasing timestamp order. During execution of the simulation, a simulation clock t_c is maintained, which indicates the progress of simulation. The following variables are used in the simulation model:

- N_r indicates the number of rounds that have been executed for an i-Search request.
- a is the ID of the peer node who triggers the i-Search mechanism.
- d is the ID of the peer node to be found.
- l indicates whether a social link exists between two peer nodes.

We use the following counters in our simulation model to calculate the output measure:

- The C_f counter counts the total number of finding a path successfully.
- The C_q counter counts the total number of the **QUERY_ARRIVAL** events that have been processed.

We repeat the simulation runs until C_q exceeds 100,000 to ensure the stability of the simulation results. Then we obtain the output measure:

$$P_f = \frac{C_f}{C_q}$$

Figure 7 and Fig. 8 show the comparison between the analytical and simulation results, whose details of the parameter setups are described below. The figures indicate that the simulation results approximate the analytical results well.

In the following, we study the effects of the input parameters on the P_f performance for the i-Search mechanism. In our study, we set the input parameters following the constraints in

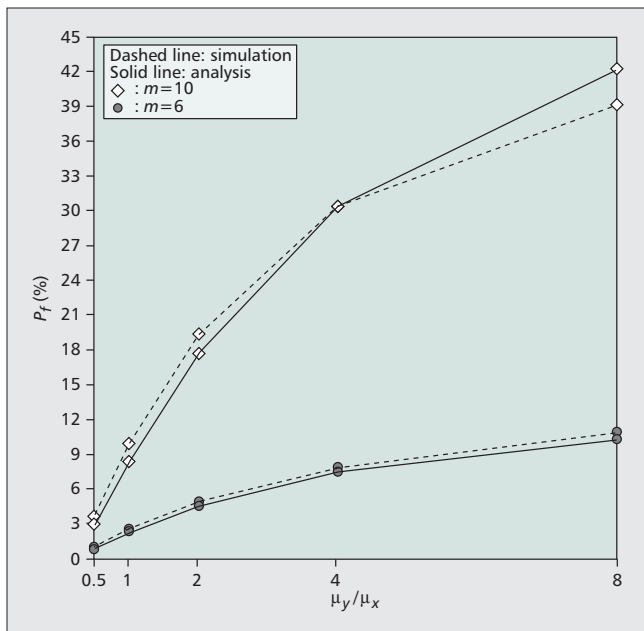


Figure 8. Effects of m ($n = 1000$; $\alpha = 0.4$).

Eq. 5, and we set the total number of Peer nodes $n=1000$. The effects of the input parameters are described as follows.

In Fig. 7 and Fig. 8, we change μ_y/μ_x from 0.5 to 8. A larger μ_y/μ_x implies that the peer node spends more time online. For example, when $\mu_y/\mu_x = 0.5$ and $\mu_y/\mu_x = 8$, from Eq. 4, we have $p_{on} = 1/3$ and $p_{on} = 8/9$, respectively. Both figures show that the path found probability P_f increases as μ_y/μ_x increases. It is worth noticing that we have P_f larger than 15 percent when $\mu_y/\mu_x = 8$ and $\alpha = 0.8$ as shown in Fig. 7 (with $m = 6$), and P_f is around 40 percent when $\mu_y/\mu_x = 8$ and $m = 10$ as shown in Fig. 8.

Observing Fig. 7 where we set $m = 6$, we study the effects of α . A larger α implies that the social graph of P2P-iSN is sparser (i.e., more far-nodes). Figure 7 indicates that P_f increases as α increases, which means that in a sparser social graph, the i-Search mechanism attains better found probability. In Fig. 8, we study the effects of m where we set $\alpha = 0.4$. A larger m implies more friends of a peer node. Figure 8 shows that with more friends, the i-Search mechanism achieves better P_f performance.

In summary, when in a sparser social graph and a peer node has more friends, there is 40 percent probability that the i-Search mechanism could find a global social relationship for the user, i.e. a social path with strong relationship strength.

Conclusions

In this article, we propose P2P-iSN, a peer-to-peer network architecture to integrate multiple SNSs without incurring excessive overhead to the SNSs. With integrated model, we could develop an effective approach, a Global Relationship Model, to evaluate the global relationship strength between two users with more precision. With P2P-iSN and the Global Relationship Model as the foundation, we propose the i-Search mechanism to find the social path with certain level of social relationship strength in a P2P social network. Through both analytical and simulation models, we have justified our approach. Our study indicates that when the social graph is sparser and a peer node has more friends, our proposed approach can find the desired social path with high probability comparing to traditional approach and can effectively establish global social relationship for users from heterogeneous SNSs. This research finding shows that by appropriately inte-

grate users in various SNSs, the proposed integrated framework could enable users to enhance their social connections over cyberspace and create more social and economic opportunities for the users.

Of course, there are many design challenges in designing a viable integrated social network framework like ours. The privacy concerns may be the biggest stumbling block as whenever a unique identifier is used. Users may not want to participate to avoid revealing the connections in their accounts. Thus, appropriate privacy protection and incentive mechanisms must be in place, which is one of our future research tasks.

Acknowledgement

The work of P. Lin was supported in part by the National Science Council of Taiwan under grant NSC 102-2219-E-002-018, by the Ministry of Economic Affairs (MOEA) of Taiwan under grant 102-EC-17-A-03-S1-214, Chunghwa Telecom, and ICL/ITRI in Taiwan.

The work of Y. Fang was partially supported by the US National Science Foundation under grants CNS-1147813 and CNS-0916391.

References

- [1] C. Zhang et al., "Privacy and Security for Online Social Networks: Challenges and Opportunities," *IEEE Network*, vol. 24, no. 4, July/Aug. 2010, pp. 13–18.
- [2] <http://newsroom.fb.com/content/default.aspx?NewsAreaId=22>.
- [3] A. Mislove et al., "Measurement and Analysis of Online Social Networks," *Proc. 7th ACM SIGCOMM Conf. Internet Measurement*, 2007, pp. 29–42.
- [4] N. Ellison and D. Boyd, "Social Network Sites: Definition, History, and Scholarship," *J. Computer-Mediated Communication*, vol. 13, no. 1, Oct. 2007, pp. 210–30.
- [5] M. N. Ko et al., "Social-Networks Connect Services," *Computer*, vol. 43, no. 8, Aug. 2010, pp. 37–43.
- [6] J. Bae, "A Global Social Graph as a Hybrid Hypergraph," *Proc. 5th Int'l. Joint Conf. INC, IMS and IDC*, Aug. 2009, pp. 1025–31.
- [7] <http://openid.net>.
- [8] <https://developers.facebook.com/docs/reference/api/>.
- [9] <http://developer.android.com/reference/packages.html>.
- [10] L. Katz, "A New Status Index Derived From Sociometric Analysis," *Psychometrika*, vol. 18, no. 1, Mar. 1953, pp. 39–43.
- [11] S. B. Ronald, STRUCTURE: A General Purpose Network Analysis Program Providing Sociometric Indices, Cliques, Structural and Role Equivalence, Density Tables, Contagion, Autonomy, Power and Equilibria in Multiple Network Systems (Version 4.2). New York: Columbia University Press, 1991.
- [12] B. Yu and M. P. Singh, "Searching Social Networks," *Proc. 2nd Int'l. Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '03)*, 2003, pp. 65–72.
- [13] N.B. Chang and M. Liu, "Controlled Flooding Search in a Large Network," *IEEE/ACM Trans. Net.*, vol. 15, no. 2, Apr. 2007, pp. 436–49.
- [14] S. Jiang et al., "LightFlood: Minimizing Redundant Messages and Maximizing Scope of Peer-to-Peer Search," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 5, May 2008, pp. 601–14.
- [15] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory*, Springer Verlag, 1995.
- [16] D. J. Watts and S. H. Strogatz, "Collective Dynamics of "Small-World" Networks," *Nature*, vol. 393, no. 6684, 1998, pp. 440–42.
- [17] H.-L. Fu et al., "Energy-Efficient Reporting Mechanisms for Multi-Type Real-time Monitoring in Machine-to-Machine Communications Networks," *Proc. IEEE INFOCOM 2012 Conf.*, Mar. 2012, pp. 136–44.

Biographies

PHONE LIN [SM] (plin@csie.ntu.edu.tw) is Professor in National Taiwan University, holding professorship in the Department of CSIE, Graduate Institute of Networking and Multimedia, Telecommunications Research Center, and Optoelectronic Biomedicine Center. Lin serves on the editorial boards of several journals such as *IEEE Trans. on Vehicular Technology*, *IEEE Wireless Communications Magazine*, *ACM/Springer Wireless Networks*, *Computer Networks*, and *Wireless Communications and Mobile Computing*. He has also been involved in several prestigious conferences, such as holding the TPC Co-Chair of WPMC 2012 and the Workshop Chair of M2M & IoT of WCMC 2013. Lin has received numerous research awards such as Distinguished Electrical Engineering Professor Award of the Chinese Institute of Electrical Engineering in 2012, Junior Researcher Award of Academia Sinica in 2010, Ten Outstanding Young Persons Award of Taiwan in 2009, and Best Young Researcher of IEEE

ComSoc Asia-Pacific Young Researcher Award in 2007. He is an ACM Senior Member. He received his BSCSIE and Ph.D. degrees from National Chiao Tung University, Taiwan in 1996 and 2001, respectively.

PAI-CHUN CHUNG (pai25@pcs.csie.ntu.edu.tw) received his bachelor and master degrees in Computer Science and Information Engineering (CSIE) from National Taipei University and National Taiwan University in 2008 and 2012, respectively. His main research interest is Internet social networks.

YUGUANG "MICHAEL" FANG [F] (fang@ece.ufl.edu) received Ph.D. degrees from both Case Western Reserve University and Boston University in 1994 and 1997, respectively. He joined the Department of Electrical and Com-

puter Engineering at University of Florida in 2000 and has been a professor since 2005. He holds a University of Florida Research Foundation Professorship from 2006 to 2009, a Changjiang Scholar Chair Professorship with Xidian University, China, from 2008 to 2011, and a Guest Chair Professorship with Tsinghua University, China, from 2009 to 2012. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002, and is a recipient of the Best Paper Award in IEEE International Conference on Network Protocols in 2006. He also received a 2010–2011 UF Doctoral Dissertation Advisor/Mentoring Award and the 2009 UF College of Engineering Faculty Mentoring Award. He is currently serving as the Editor-in-Chief of *IEEE Transactions on Vehicular Technology*.