

A Semantics-based Approach to Large-Scale Mobile Social Networking

Juan Li · Hui Wang · Samee Ullah Khan

© Springer Science+Business Media, LLC 2011

Abstract Mobile ad hoc social networks are self-configuring social networks that connect users using mobile devices, such as laptops, PDAs, and cellular phones. These social networks facilitate users to form virtual communities of similar interests or commonalities. This paper proposes semantics-based mobile social network (SMSN), a novel framework of a fully functional mobile ad hoc social network that incorporates semantics of users' social data. SMSN provides effective and efficient solutions to social network construction, semantics-based user profile matching, multi-hop semantics-based routing, and privacy management. Moreover, SMSN is rigorously benchmarked using an elaborate simulation setup and released as a prototype system that can be run on cellular phones. Due to its generality, SMSN can be applied to a wide range of critical applications, such as disaster-recovery, homeland security, and personnel control.

Keywords social networking · semantics · mobile ad hoc network · security · privacy

J. Li (✉)
Department of Computer Science, North Dakota State University,
Fargo, ND 58108, USA
e-mail: j.li@ndsu.edu

H. Wang
Department of Computer Science, Stevens Institute of Technology,
Hoboken, NJ 07030, USA
e-mail: hui.wang@stevens.edu

S. U. Khan
Department of Electrical and Computer Engineering,
North Dakota State University,
Fargo, ND 58108, USA
e-mail: samee.khan@ndsu.edu

1 Introduction

Over the past several years we have witnessed an enormous interest in social networks, such as Facebook, Myspace, Flickr, YouTube, LinkedIn, and Yahoo!360. Social network users independent of their physical local can build new or participate in existing online digital communities that share similar interests. Simultaneously to the surge of social networking, mobile devices, such as laptops, PDAs, and cellular (smart) phones, have been widely used. A natural trend is to integrate social networks with mobile devices. For example, MySpace and Facebook have provided limited versions of their services on mobile phones. Other websites, such as Plazes [41], Dodgeball [8], Jaiku [17], and Bluepulse [4], provide social network services to mobile users only. In these systems, users interested in accessing the social networking applications can use their mobile devices while on the go. Most of such social networks only consider using mobile devices as tools to access the pre-existing social networks and rely on the centralized management paradigm.

In practice, instead of pre-built social networks, there exists tremendous need of building social networks spontaneously from mobile devices. The construction of such social networks over mobile devices in the events or at locations, such as conferences, expositions, galleries, stadiums, and restaurants enable people to communicate and share their experiences without the need to have Internet access and with minimum required infrastructure. These networks shift from the existing social networking archetype towards a mobile ad hoc network (MANET) that potentially connects all types of devices that are equipped with short-range communication medium, such as Bluetooth and Wi-Fi.

Unlike traditional social networks in which social communities are pre-built from real-world relationships, users in spontaneous mobile social networks are automatically grouped by their social behaviors. This raises the challenge of discovering users of similar social patterns that acts as the basis of the network infrastructure. There exist applications, for example, Jambo Networks [18], Nokia Sensor [33] and MobiLuck [31] that support construction of spontaneous social network. In these systems, users are required to input keyword-based profiles. The discovery of friendship is based on mapping of keywords in user profiles. However, the network may consist of various types of mobile devices (with each equipped with different applications); as a consequence, data in the network will likely be of diverse formats. Therefore, the keyword-based matching mechanism cannot effectively discover social patterns from heterogeneous user profiles. Moreover, users may use different vocabularies and categorizations. This adds additional complexity of analysis on large-scale, heterogeneous data to discover hidden social communities.

To circumvent the above mentioned problems, we propose a semantics-based mobile social network framework (SMSN). In SMSN, mobile devices store user social profiles including their interests, and hobbies. Some data of the profile can be extracted from the applications being executed on mobile devices. By the analysis of the profiles, communities of users of similar interests are discovered and automatically constructed. A major component of SMSN is the semantics-aware matchmaking and discovery mechanism, by which the similarity measurement of user profiles is equipped with rich *Semantics-based* reasoning. Our proposed semantics-based matchmaking adds machine-accessible semantics to the system. It uses ontology to analyze user profiles and application data that enables the inference of relationship and similarity between users and resources. Therefore, it allows the exchange of heterogeneous social data without loss of meaning between various applications and/or user profiles of diverse types, and greatly improves the interoperability between diverse devices and data.

In many cases, it is necessary to construct large-scale social networks connecting more users. Therefore, devices must be connected with each other forming a multi-hop network. Unfortunately, existing spontaneous mobile social networks [18, 31, 33] are exclusively confined to single hop access that is within the transmission range as provided by Wi-Fi and Bluetooth. In a multi-hop ad hoc network to discover friends, the most straightforward way is broadcasting. However, broadcasting-based transmission typically results in extremely poor performance even for medium-size networks. Moreover, due to mobility, limited battery power, constrained bandwidth, lack of centralized control, and specialized infrastructure demand, an ad hoc social network

poses new design challenges compared to conventional web-based social networks. SMSN addresses these challenges and ensures the scalability and efficiency by proposing a semantics-based multi-hop routing protocol for social network construction and query forwarding. Our proposed scheme avoids redundant flooding and reduces the system overhead by integrating semantics-aware discovery with the underlying ad hoc routing layer. This integration maximally exploits the interactions between the overlay routing and physical routing protocols to optimize the routing performance.

Below we enumerate the specific contributions of SMSN.

- We propose an ontology model to define user profiles that adds semantics to the mobile social network, thus improving the expressiveness and interoperability of the system.
- We propose a comprehensive profile matchmaking scheme that accurately measures the similarities between different users in the semantic level.
- We design a novel resource discovery routing protocol that integrates the logical overlay network routing with the physical ad hoc network routing.
- We propose a privacy preserving mechanism for profile matching to prevent releasing user's sensitive data when user profiles are exchanged for similarity measurement.
- We conduct comprehensive simulation experiments to test different aspects of the designed system and implement a prototype system to demonstrate feasibility.

The rest of the paper is organized as follows. Section 2 describes the architectural and system design of the SMSN system, including the details of profile generator, semantics-based user profile matchmaking, semantics-based multi-hop routing, and privacy-preserving user profile matching. In Section 3, we evaluate the proposed method and show the effectiveness of SMSN with a comprehensive set of simulations. Related work and concluding remarks are provided in Sections 4 and 5, respectively.

2 System design

2.1 Overview

The system architecture of the proposed SMSN is depicted in Fig. 1. SMSN consists of two layers: (a) a physical ad hoc network layer and (b) a virtual social network layer. Devices in the ad hoc network layer are assumed to be equipped with wireless interfaces (e.g., WiFi, Bluetooth) that must support ad hoc communication mode. Nodes in the (virtual) social network layer are connected via virtual links. Each virtual link corresponds to a communication path that may be composed of several physical hops. Nodes

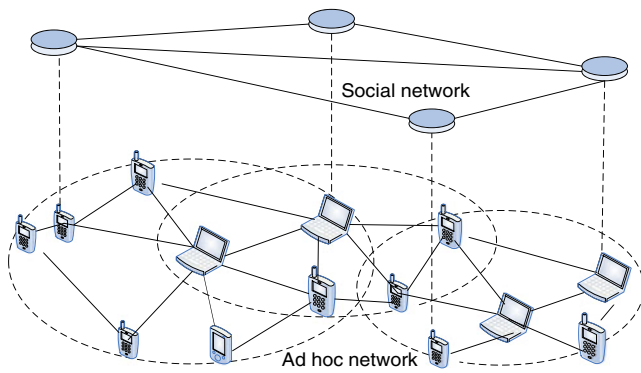


Fig. 1 System overview

connected via virtual links are referred to as *friends*. Once friends are established, they can perform social operations, such as sharing resources, sending messages, and browsing profiles of other SMSN users.

To materialize the functionality of mobile social network, the proposed SMSN system architecture consists of four main components:

- (a) profile generator that defines user and/or resource profiles with their semantics integrated. On basis of built-in semantics, the profiles contain extensible information such as user identifications, roles, and preferences. The profiles may also keep dynamic context information such as location and state information obtained from device sensors.
- (b) matchmaker that intelligently matches profiles keyed by their similarities, so that users with similar interests can be connected as potential friends at the social network layer. The matchmaking is semantics-based, i.e., it matches the meanings of the profiles not the terms. For example, users interested in “mobile wireless network” can be matched with users interested in “ad hoc network”, even though they do not have matching keywords.
- (c) routing controller that effectively and efficiently discovers *friends* and preferred resources in a large-scale network. In particular, we design a novel semantics-based multi-hop query routing controller called SDV (semantic distance vector routing). With small overhead for computing the right routes, SDV is bandwidth- and energy- efficient.
- (d) privacy manager that enables participants to have full control over their preferred anonymity, which allows their personal information to be set on different privacy levels. Further, the privacy manager supports privacy-preserving profile matching so that user profiles are matched without revealing any private information.

The aforementioned components work in concert to lay the foundations of a general purposed mobile ad hoc social network. Below we detail the four essential components.

2.2 Ontological profile generator

User/resource profile is a crucial element of the SMSN infrastructure. Our goal of devising an effective profile scheme is to make it detailed enough, so that users can express various kinds of queries, and structured enough, so that the system is capable of efficiently locating the relevant users and resources. Moreover, profiles must be able to express the machine-executable semantic meaning of user/resource information. In consequence, our design of profiles requires shared representations of knowledge as the basic vocabulary from which profiles can be asserted. An ontology, “a shared and common understanding of a domain” [13], is precisely intended to convey that kind of shared understanding. Therefore, we use ontology to represent user and resource profile. The ontology-based representation is more expressive and less ambiguous than a keyword-based representation [27]. It provides an adequate grounding for the representation of coarse- to fine-grained user interests and is able to deal with the subtleties of user preferences. In addition, the ontology provides formal, machine-executable meaning on the concepts. Moreover, ontology standards support inference mechanisms that can be used to enhance semantic matchmaking.

To cope with the openness and extensibility requirements, we adopt two W3C recommendations: the Resource Description Framework (RDF) [43] and the Web Ontology Language (OWL) [35] as our ontology language. We describe the ontology profile in OWL-DL [35]. In particular, we separate the profile definition into two parts: the terminological box (T-Box) and the assertion box (A-Box) as defined in the description logic terminology [2, 28]. The T-Box is a finite set of terminological axioms, which includes all axioms for concept definition and descriptions of domain structure. The A-Box is a finite set of assertion axioms, which includes a set of axioms for the descriptions of concrete data and relations.

In SMSN, the T-Box ontology defines the common understanding for all the important concepts and their relationships. For the creation of the T-Box ontology we adopted a top-down approach; first we select important general concepts, which were later enriched and specialized. We adopt some concepts from FOAF [14], such as Person, Interest, Image, Name, Gender; however, we do add more dynamic concepts, such as current activity, current terminal, location, motion state and orientation. Addition of such important concepts enables us to effectively capture the mobility aspect in SMSN. Our principle in T-Box ontology design is to create a general yet extendable ontology that will be able to adapt to the needs of every application. Figure 2 depicts a part of the high-level picture of our proposed T-Box profile ontology.

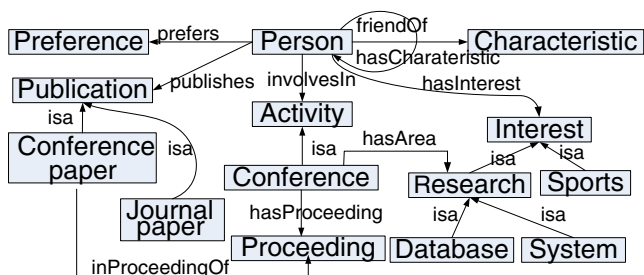


Fig. 2 Part of the T-Box ontology of the user profile

With the defined T-Box ontology, users can define their individual A-Box ontology by instantiating the T-Box. In fact, ontological user profile is an annotated instance of the reference T-Box ontology. Figure 3 shows an example of A-Box instances that are based on the T-Box ontology defined in Fig. 2. The salient feature of the semantic user profiles is that they can be used in a wide variety of completely different applications. Because the user carries his/her profile along with his/her mobile devices, user profiles are independent of any specific social network. We also would like to point out that the proposed ontology-based profiling can be utilized in several other applications, such as homeland security profiling, inmate profiling, and border security profiling.

To simplify the representation, we store the OWL user profiles in the form of $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ expressions, called triples in RDF terminology. The *subject* denotes the resource which has a Universal Resource Identifier (URI). The *predicate* denotes traits or aspects of the resource and expresses a relationship between the *subject* and the *object*. The *object* is the actual value. In real scenarios, some users are not willing to spend much time describing their detailed information to the system. Our profile matchmaking mechanism overcomes this problem by expanding the initial information stored in user profiles through explicit semantic relations with other concepts in the ontology. This shows another advantage of using ontology-based profiles over the keyword-based profile representations. We must note that due to the fact that the present mobile devices (e.g., cell phones,

```

<Person rdf:about="#Jen">
  <name>Jen Li</name>
  <affiliation rdf:resource="#NDSU"/>
  <hasInterest rdf:resource="#Data mining"/>
  <hasInterest rdf:resource="#P2P_Networking"/>
  <involveIn rdf:resource="#VLDB2009"/>
  <friendOf rdf:resource="#Wendy"/>
</Person>

<Confernece rdf:about="#VLDB2009">
  <hasArea rdf:resource="#Core_Database"/>
  <hasArea rdf:resource="#Data mining"/>
</Confernece>
...

```

Fig. 3 Part of the T-Box ontology of the user profile in OWL

PDA's, etc.) intend to be equipped with large memory space, it is feasible to store the ontology on these devices.

To protect their privacy, users can specify parts of their profile data as private, so that others cannot access these parts. But this causes the challenge of matching user profile without revealing the private data. We will elaborate on our proposed solution to this issue in Section 2.5.

2.3 Profile matchmaker

The SMSN matchmaker component matches (a) user profiles and (b) queries and resources by their semantic similarity. Compared with simple keyword-based matching approach, the SMSN matchmaker is able to overcome differences in vocabularies and support inference mechanisms. For instance, users interested in “mobile networks” (superclass of “ad hoc network”) can be matched with users interested in “ad hoc networks”. Conversely, a user interested in “peer-to-peer computing” (subclass of “distributed systems”) can also be inferred to be potentially interested in “distributed systems”. Moreover, a user interested in knowing more about “China” can also be assumed to be interested in knowing more about its capital – Beijing. These inference characteristics are exploited in our profile matching model.

Profile matching is implemented by calculating the similarity between profiles to be matched. Computing the semantic similarity between two objects is very challenging. Technologies, such as natural language processing, information integration, and graph matching, are in need to measure the similarity of the syntactical, structural, and semantic aspects of the ontology data. There has been extensive research [19, 25] focusing on measuring the semantic similarity between two objects in the field of information retrieval and information integration. However, their methods are very comprehensive and computationally intensive, and thus not suitable to mobile wireless devices. In this paper, we take advantage of the profile ontology and propose a simple mechanism to compute the semantic similarity between two profiles. As a result, profile matching can be performed in the mobile ad hoc network in which devices have limited computing capacity and bandwidth.

2.3.1 Instance projection

As mentioned, user profiles are represented as instances (A-Box) of the profile ontology. A naïve way to match user profiles is to match their corresponding A-Boxes. However, due to diversity and the potentially large number of instance elements, profile matching on instance (A-Box) level can be very complex and sensitive to vocabularies. Such challenges can be addressed by mapping the particular instance to the

shared T-Box ontology. This is based on the idea that, on a semantic level, similar instances (A-Box individuals) should behave similarly with respect to the same concepts (T-Box class or properties). The rationale is to compare individuals on the grounds of their corresponding concepts, which correspond to nodes and edges of the T-Box graph. Therefore, the first step of our matching mechanism is to project A-Box on the instance level to the T-Box on the concept description level before comparing them (recurring to the notion of the most specific concept [38] of an individual with respect to the ABox). This projection from A-Box to T-Box requires the entailment of an assertion (instance-checking). In Fig. 4, two user profiles represented as A-Box instances are projected to the reference T-Box ontology. In this way, ontology distance can be computed with respect to the same T-Box ontology graph. This greatly reduces the complexity of computing the semantic distance of user profiles.

2.3.2 Profile similarity measurement

Through projection, the instance level profile matchmaking is converted to a problem of computing semantic similarity between sub-graphs of the T-Box ontology. A number of approaches have been proposed to determine the similarity of two objects. They can be divided into two categories: distance-based approaches and information content-based approaches. The distance-based approach [29, 38, 42] is to find the shortest path between two concepts in terms of number of edges in a ontological graph; the distance of the shortest path corresponds to the semantic distance between these two concepts. Information content-based approaches [19, 48] are inspired by the perception that pairs of words which share many common contexts are semantically related. Thus the basic idea of these methods is to quantify the frequency of the co-occurrences of words within various contexts. Because our similarity computation is based on a common ontology graph, the distance-based approach is more suitable to our framework. However, most distance-based approaches assume that all edges in the graph have a uniform weight, which does not accurately reflect the edges' semantic variability. We propose a weighted-distance measurement approach, which improves the accuracy by integrating factors including the depth of a node in the ontology hierarchy and the type of the links connecting nodes in the format as weight into the measurement metric. On this basis, the semantic relations

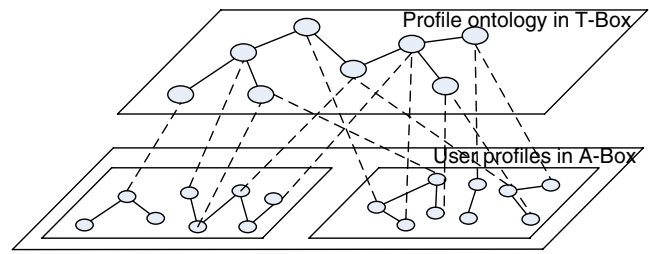


Fig. 4 Example of projecting A-Box profiles to T-Box ontology

between concept nodes are respected and thus the accuracy of the estimation is enhanced.

Specifically, when exploring semantic relations at the T-Box level, we assign different types of links with different semantic distance-factor. For example, we assign the smallest distance-factor to the *equivalent-class* edge that denotes the two concepts at the end of edge are identical. Moreover, we generally consider that the distance-factor of *sub-class (super-class)* relation is shorter than that of other relations. To support flexibility, we allow applications and users to customize the distance-factor to reflect their special need.

Another important factor affecting the semantic distance is the depth of the nodes in the ontology hierarchical graph. In particular, concept nodes sharing ancestors (i.e., common more general concepts) at lower levels should be more similar than those whose common ancestors are at higher levels. For example, the concepts “Database” and “System” (with common ancestor “Research”) in Fig. 2 are more similar than “Research” and “Sports” (with common ancestor “Interest”), even though the shortest paths between “Database” and “System” and that of “Research” and “Sports”, have the same length.

To materialize the aforementioned principle, we formally define the similarity measurement as following.

Definition 1 (Profile Summary) The profile of user u can be represented as a vector of T-Box concepts (classes and properties) $P_u = \{C_1, C_2, \dots, C_n\}$ projected from the user’s instances.

In this definition, a vector of T-Box concepts can be used to summarize the profile in that users share a common global T-Box ontology, thus the concept vector is able to be mapped to a sub-graph in the global ontology graph.

Definition 2 (Concept Distance) The semantic distance between two concepts C_a and C_b in a given ontology is defined as:

$$dis(C_a, C_b) = \frac{1}{2} \left(\frac{\sum_{i \in path(C_a to C_p)} w_i dis(C_i, C_{i+1})}{\sum_{i \in path(C_a to C_{root})} w_i dis(C_i, C_{i+1})} + \frac{\sum_{j \in path(C_b to C_p)} w_j dis(C_j, C_{j+1})}{\sum_{j \in path(C_b to C_{root})} w_j dis(C_j, C_{j+1})} \right),$$

where C_p is the common ancestor of C_a and C_b in the hierarchical ontology graph, C_{root} is the root of the ontology tree, C_{i+1} is C_i 's parent, and w_i is the weight of edge presented as a distance factor (i.e., the closer relationship, the smaller the distance).

The semantic distance above defines the distance as a “relative” distance to the distance between nodes and their common ancestors, thus it integrates the edge weight with the depth and the length of the shortest path.

Definition 3 (Concept Similarity) The concept similarity between two concepts C_a and C_b is defined as:

$$sim(C_a, C_b) = 1 - dis(C_a, C_b).$$

The similarity between two concepts is defined as the complementary of their semantic distance which is a value between 0 and 1; consequently, the similarity is also between 0 and 1.

Definition 4 (Profile Similarity) Given two profiles P_x and P_y , the similarity between the two profiles is defined as:

$$sim(P_x, P_y) = \frac{\sum_1^n \max_{j \in [1, m]} sim(Cx_i, Cy_j)}{n},$$

where n is the number of concepts in profile P_x and m is the number of concepts in P_y . If $sim(P_x, P_y)$ is larger than a user-defined similarity threshold t ($0 < t \leq 1$), the profile P_x is said to be *semantically related* to P_y .

Given two user profiles that are summarized as collections of T-Box concepts, the similarity between these two profiles is the normalized similarity of all related pairs of concepts. The problem can be converted to compute pair-wise distances of a DAG, which is a simple problem for a reasonably large graph. The complexity of this problem can be represented as $O(|P_x| * |P_y|)$ ($\leq O(n^2)$), where P_x and P_y are the set of T-Box concepts of the two profiles being compared, and n is the number of nodes of the ontology graph. Because the comparison is based on the common ontology graph, all the pair-wise distances can be pre-computed so that the results can be obtained quickly for future computation. As illustrated by our experiments at Section 3, this profile similarity measurement is scalable with the increase of the properties (concepts) in the profile. Note that by this definition, similarity is not a symmetric relation, i.e., “how similar is A to B ?” may give a different answer from “how similar is B to A ?” Employing such an asymmetric measurement exactly follows human judgment: sometimes, we say one object is similar to another one, but not vice versa.

The aforementioned matchmaking matches profile summarized at the T-Box level to locate potential friends. This process

normally does not involve privacy issues because the comparison is based on general domain of the user interests. It is possible that there are needs to match users’ interests on specific topics (e.g., democracy). We call it the *exact profile match* at the *fine-level*. Exact profile matching needs users to exchange their specific interests, which may involve private information. Thus we require such measurement be privacy-preserving. The details of how to evaluate the exact match of user profiles on the fine-level with preserved privacy are discussed in Section 2.5.

2.4 Routing controller

In social networks, an important issue is to discover users of similar interests and resources with particular properties. Due to the absence of a centralized intelligence, user/resource discovery in mobile ad hoc networks is a very challenging predicament. To circumvent such problems, we propose a fully decentralized semantics-aware routing algorithm, namely SDV (acronym for Semantic Distance-Vector routing), on top of ad hoc networks to efficiently discover friend and resources.

In SDV routing protocol, nodes proactively distribute semantic summary of user/resource profiles within a certain range to their network neighborhood. Based on the semantic summaries, nodes construct routing tables. Specifically, each node maintains an entry for every neighbor (adjacent node) in the ad hoc network. Distance information, in terms of the number of hops, is included so that the nodes are aware of not only where but also how far the profiles in the neighborhood are located. A node can make routing decisions by knowing only its immediate neighbors and limited resource information. In addition, a leap mechanism is used to expedite the searching process by skipping over the “barren” areas, i.e., the areas consisting of few resources/profiles.

The SDV routing is only used as a hint to find potential matching nodes. When they have been located, the discovery query is redirected to the candidate nodes for the final matchmaking. Real communication of a node is commonly concentrated on a small number of particular candidate nodes. Our proposed scheme avoids redundant flooding and reduces the system overhead by integrating resource discovery with the underlying ad hoc routing layer. This integration maximally exploits the interactions between the overlay routing and physical routing protocols to optimize the routing performance.

2.4.1 Routing table construction

Each node maintains a routing table, allowing a node to select the best neighbor to transmit query. The routing table contains both local and neighbor ontology information at various distances in terms of number of hops. Each entry in

the routing table depicts resources available from a neighbor. Each neighbor entry (row) contains k columns for resources within the range: $[1, k]$ hops. The k -hop limit is termed *radius* and is predetermined. The first column of the neighbor entry contains the profile vector from that neighbor. The i -th column stores a set of profile vectors from all nodes that are i -hops away via that neighbor. For example, in Fig. 5 (a), by its routing table, node A knows it can reach node B and node C 's ontology vector within 1-hop, and a vector $\langle C_2, C_3 \rangle$ within 2- hops through node C .

Figure 5 also illustrates the updates on routing tables when a new node E joins the network. Node E joins the network by connecting to the existing nodes of the ad hoc network within its transmission range (in this case, node A). After the connection is established, node E sends its profile vector to node A . Similarly, node A updates node E with all of the resource and distance information that node A has knowledge of. Specifically, node A merges all neighboring entries (rows) in the routing table into a single entry by aggregating elements of the same distance and removing duplicates. Node A also adds its local vector as the first column to the merged entry. After the adjustments, node A sends the merged entry to node E . The merged entry represents resources accessible from node A and their shortest distances to node A . (Note that the last column (2-hop column) is removed from the merged entry. This is done due to the fact that the distance to node E will be farther than the pre-specified *radius* that is equal to two.)

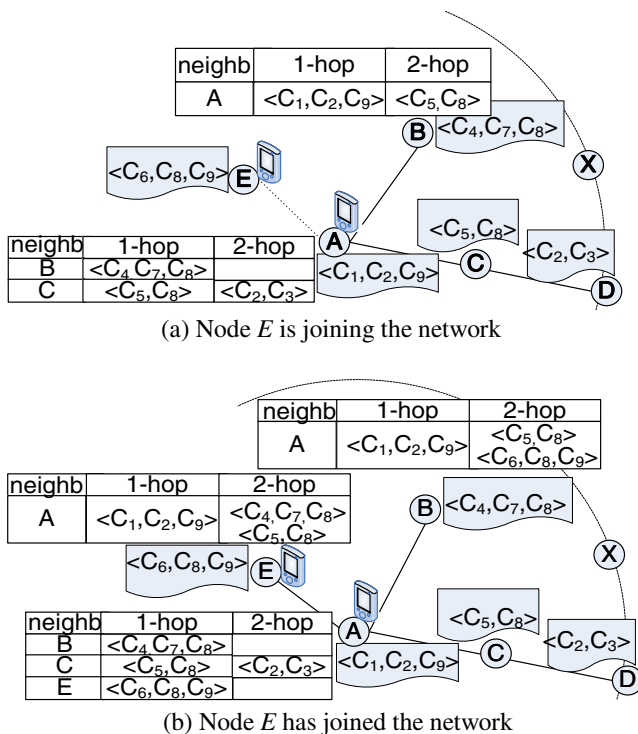


Fig. 5 Maintaining routing indices ($radius=2$). a Node E is joining the network. b Node E has joined the network

After node E receives the merged routing entry from node A , node E adds 1 hop to each column of the entry, and adds an additional row in its routing table (as shown in Fig. 2 (b)). After node A receives node E 's information and updates its routing table, node A informs its neighbors (nodes B and node C) of this update.

To reduce the overhead of transmitting routing information, a soft-state update mechanism is used, in which instead of being exchanged for each update, routing information is exchanged periodically. In particular, at regular time intervals, each node sends updates (if there is any) to and receives updates from its directly connected neighbors. When a node receives routing information from a neighbor, it updates its local table if the neighbor suggests a "better" route than what it is aware at present. Eventually all information within the range determined by *radius* are known. The storage required by a routing table is modest as we only store T-Box concepts in vectors, which is of a limited size. Moreover, duplicated vectors are removed before being sent to neighbors. Furthermore, the storage space per neighbor can be adjusted by increasing or decreasing the *radius*.

2.4.2 Query forwarding

A discovery query is in the form of a vector of ontology concepts. When node N receives a *friend-discovery* query Q that tries to find a friend for a node X , node N computes the semantic similarity between node X and itself. If node N is semantically related to node X , i.e., $sim(Q,N) > t$, node N will send a reply to node X . At the same time, node N will also evaluate the similarity between node X and node N 's *friends* that most probably also are related to node X . If the query's time to live (TTL) has not expired, N examines the 1-st column in its routing table. If it finds the desired profiles, then it is straightforward that those profiles are only one hop away, and the query is forwarded to the matching neighbor closest to the current node. If there is no such match, the querying node searches for a match in the 2-nd column. It repeats the above procedure, until it finds a match or traverses all columns in the routing table. A query may be transferred several hops until arriving at the matching node or the query TTL expires. To avoid processing queries more than once, every query has a unique query ID and every node keeps a list of recently received query IDs. If a query has been received before, it will be discarded.

2.4.3 Leap

By manipulating *radius*, we limit the distance profile information can traverse. This reduces the size of the routing table and consequently the overhead of transmitting

the routing information. However, a node may not find any (node) matches from its routing table to forward a given query. When this happens, a query can *leap* over the “barren” area through a long-distance link. Before reaching the long-distance nodes, the *leap* query is simply forwarded between intermediate nodes without similarity matching. After reaching the long-distance nodes, the *leap* query becomes a normal SDV query. To discover those long-distance links, the system employs an aggressive caching technique, that is, after query evaluation is finished, the query answer travels along the reverse path to the requester. Whenever it is passed through a node, it is cached on that node. Candidate long-distance nodes should be located outside the neighborhood area and preferably those nodes that responded to the previous queries.

2.5 Privacy manager

As mentioned in Section 2.3, after users of similar interests are identified, there may exist the need to further search for exact matched user profiles. As defined in Section 2.3, at fine-level, user profiles are represented as a set of triples $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ that describe users’ properties. We say triple $t_1 \langle \text{subject}_1, \text{predicate}_1, \text{object}_1 \rangle$ equals $t_2 \langle \text{subject}_2, \text{predicate}_2, \text{object}_2 \rangle$ if $\text{subject}_1 = \text{subject}_2$, $\text{predicate}_1 = \text{predicate}_2$, and $\text{object}_1 = \text{object}_2$. Then given two user profiles P_1 and P_2 , $|P_1 \cap P_2|$ is defined as the number of equivalent triples in P_1 and P_2 . We say user u_1 has an *exact-matching* with user u_2 if $\frac{|P_1 \cap P_2|}{|P_1|} \geq \mu$, where P_1 and P_2 are the profiles of user u_1 and u_2 , and μ is a given threshold. Note that matching is not symmetric, i.e., u_1 has a matching profile with user u_2 but u_2 may not match with u_1 .

An important issue in the exact match scenario is privacy. Because the exact matching of user profiles involves users’ private information, for example, profession and hobbies, no one is willing to reveal his/her profile to the others, especially strangers. Then the challenge is how to check whether two users have exactly matched profiles without revealing their profiles to each other.

To address this challenge, we design our privacy controller component to realize privacy-preserving exact matching of user profiles. We assume that although the users do not want to reveal their whole private profiles, they do not care to let the others know the mutual common interests. Based on this assumption, we adapt the private set intersection protocol [10, 24]. Before we explain the details, we introduce the basic concepts of the private set intersection protocol.

The problem of *privacy-preserving set intersection* is to design a protocol for two or more parties, each having a private dataset, to compute the intersection of their sets without revealing to each other any of the remaining elements. For example, suppose that A has set $\{a_1, a_2, a_3, a_4\}$ and B has set $\{a_1, a_2, b_1, b_2\}$. Then both A and B can learn that $\{a_1, a_2\}$

is the intersection set. However, A cannot learn that B has b_1 and b_2 , similarly B cannot learn that A has a_3 and a_4 . Several cryptographic solutions have been proposed recently for the privacy-preserving set intersection problem [10]. Ref. [10] presents a protocol based on the use of homomorphic encryption and balanced hashing. *Homomorphic encryption* is a well-known technique in cryptograph [7, 36]. A cryptosystem with encryption function E is said to be *homomorphic*, if $E(x).E(y) = E(x + y)$, Where x and y are two plaintext message blocks. Then by the protocol in [10], party A computes a degree- m polynomial that has $\{x_1, \dots, x_m\}$ as roots: $P(x_j) = \sum_{i=0}^m \alpha_i x_j^i$. Using a homomorphic encryption scheme, the coefficients α_i of the polynomial are encrypted with the client’s private key and $E(\alpha_i)$ are sent to party B . B then uses the homomorphic properties of the ciphertexts to compute $E(P(y_i)) = E(\sum_{i=0}^m \alpha_i y_j^i)$. Moreover, B multiplies each result by a random number r_j to get an intermediate result, and adds to it an encryption of the value of the input, $E(r_j * P(y_j) + y_j)$. At the end, for each of the elements in the intersection of the two parties’ inputs, the result of this computation is the value of the corresponding element, whereas for all other values the result is random.

To adapt the privacy-preserving set intersection protocol to SMSN, each triple $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ in the user profiles is transferred to a string by concatenating subject, predicate and object together. Then each user profile is equivalent to a set of keyword strings. By applying the private set intersection protocol in on the sets of keyword strings, the intersection between the user profiles that correspond to specific matching interests is returned. The complexity of the protocol is $O(mn)$, where m and n are the number of private triples in two input profiles. It is straightforward that the protocol is secure because no user learns more than the computed intersections of their private profiles.

2.6 Prototype implementation

The SMSN prototype is fully implemented in Java 2 Micro Edition (J2ME) and deployed on a set of twelve SonyEricsson_K750 cellular phones. J2ME’s Bluetooth API (JSR 82) is used for Bluetooth-enabled mobile phones. The prototype implements all of the core features of our SMSN framework. These features include: profile generating, friend matchmaking, routing control, and privacy management. We also implemented the ad hoc multi-hop text messaging and file sharing as part of the service pack for SMSN. Figure 6 shows the screenshot of an application running on the J2ME simulator. We have released the open source SMSN project on Google Code (with name MobiSN). The prototype can be downloaded from SMSN website [46], and the Google Code site [50].

Fig. 6 SMSN prototype running on the J2ME simulator



3 Experiment

In this section, we evaluate the performance of the proposed system in ad hoc networks.

3.1 Setup

To effectively simulate SMSN, an enclosed ad hoc network environment was considered. We created networks containing different numbers of nodes, spreading randomly over an area of 200 m×200 m. The density of the nodes was adjusted throughout the simulations. The mobility of the nodes was similar to that of the “random waypoint” model as reported in [3]. In this model, initially, the nodes are randomly distributed within the enclosed area. Each node has a randomly picked destination, towards which, the node moves at a predetermined speed. The faster their relative speed, the more dynamic the network is. Once a node reaches its destination, the node pauses for a predefined interval of time, then it repeats this movement pattern. The pause time controls how long a node can remain in one place before moving. The longer the pause time, the more stable the network is. The transmission range of a node was predetermined to be 10 m.

In the network, each node has its own set of generated ontological profile. To abridge the ontology data, we consider the ontology data as a pure hierarchical relationship with some *equivalentClass* relationship. Specifically, the ontology data was generated by observing the T-Box ontology schema that includes the definition of the classes and corresponding properties. The classes and properties form a multilevel hierarchy. We limit the number of levels and branching factor to four. A user profile was created by instantiating the classes. Each user profile contained 1–5 interest classes with detailed instantiation. The various simulation parameters and their default values are listed in Table 1.

3.2 Results and discussions

We start the evaluation of SMSN by first recording the effectiveness of our ontology-based similarity measure for *friend* and resource discovery. We compare our ontology-based similarity measure with semantics-free exact-match as two different ways of matching profiles in the process of friend and resource discovery. The exact-match approach matches profiles by comparing their keywords to identify if the profiles have enough common keywords (beyond a threshold). This approach does not take into account the specific meanings of these keywords in the ontology or the relationships between keywords. For our ontological similarity matching, we use two similarity threshold t , 0.6 and 0.8, respectively.

Table 1 Parameters used in the simulations

Parameter	Range (default)
Network size	100-1100 (500)
Environment area	200 m*200 m
Node moving speed	1-20 m/s (1 m/s)
Node transmission range	10 m
Node pause time	0 s-80 s (20 s)
Query rate	100 queries/s
Query message size	1 packet
Partial routing update message size	1 packet
Complete routing table update message size	15 packets
TTL	1-9 (4)
No. of walkers	1-5 (2)
Routing table radius	1-5 (2)
Routing table update frequency	every 2 seconds
No. of ontological classes in node's profile	1-5
Total no. of classes in the common ontology	50
Semantic similarity threshold	0.8
No. of queries per simulation session	20 K

Figure 7 depicts the findings of our first set of simulations. As it can be seen, our ontology-based matching technique identifies multifold more relevant *friends* compared to the exact-matching. This is simply because our similarity function measures the similarity on the semantic level rather than the syntax level. Therefore, even if two profiles are quite different literally, they can be pretty related semantically. We also observe that when the similarity threshold decreases, the number of results increases. This relationship holds true because more relevant relations can be identified. This relationship also is an important property when a given social network may not have many participants. Therefore, SMSN users may have the leverage to relax the criteria for *friend* matching to construct a social network with sparse user population that is an option that is not available in any of the currently existing social networks. Moreover, because ontological search eliminates the problem of semantic ambiguity, such as polysemy and homonymy, results returned are relevant with extremely high precision.

To evaluate the scalability of our profile matching algorithm, we measure the time cost needed to compute the similarity between profiles with increased number of properties. The experiment was performed on the J2ME simulator environment. In the experiment, the global profile ontology is modeled as a hierarchical tree-like structure representing the classification of general interests. The tree's depth is 4 and its branching factor is 5. We vary the number of interests each user possesses and then measure the latency needed to compare the different sized profiles. As shown in Fig. 8, the latency does not change much as the number of properties increases. This proves that our profile matching algorithm is scalable in terms of the profile size.

Simulations also were carried out to validate and characterize the performance of the proposed ad hoc routing algorithm. For comparisons, we implemented node discovery based on flooding and probabilistic-flooding (p-flooding) techniques [16, 32]. The flooding-based technique was chosen as a reference approach for its simplicity and prevalence that in fact, made it a widely used baseline for many previous research efforts [26]. The p-flooding technique is a controlled flooding that

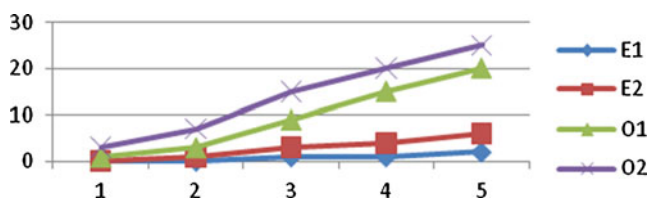


Fig. 7 Number of friends discovered vs. routing hops E1: Exact instance-level match, E2: Exact class-level match, O1: Ontology similarity measure with $t=0.8$, O2: Ontology similarity measure with $t=0.6$.

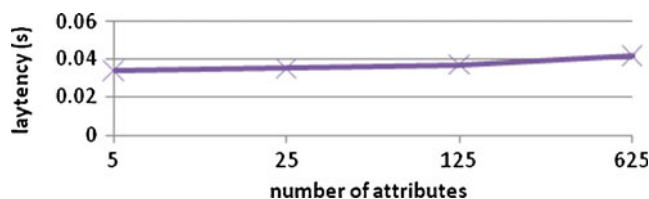


Fig. 8 Profile size vs. computation time

rebroadcasts only with a certain probability p . To simplify our simulation set up, instead of using a given probability p , each node chose a certain predetermined number of neighbors (referred to as walkers) to forward the query. We set the number of walkers as three for both the SDV and p-flooding routing procedures.

Figure 9 compares the efficiency of three different discovery mechanisms, in terms of number of results found with respect to the number of hops used (Fig. 5(a)), the query latency of finding certain number of results (Fig. 5 (b)), and the query overhead caused to locate certain number of results (Fig. 5(c)). It is an undeniable fact that flooding can identify more results and use shorter latency compared to the other two techniques. However, the query overhead of the flooding is much higher compared to the other two approaches. In contrast, the proposed SDV technique identifies comparative number of results and incurs much less query overhead compared to the other two techniques.

Figure 10 illustrates the bandwidth overhead including both the query and maintenance overhead incurred by each

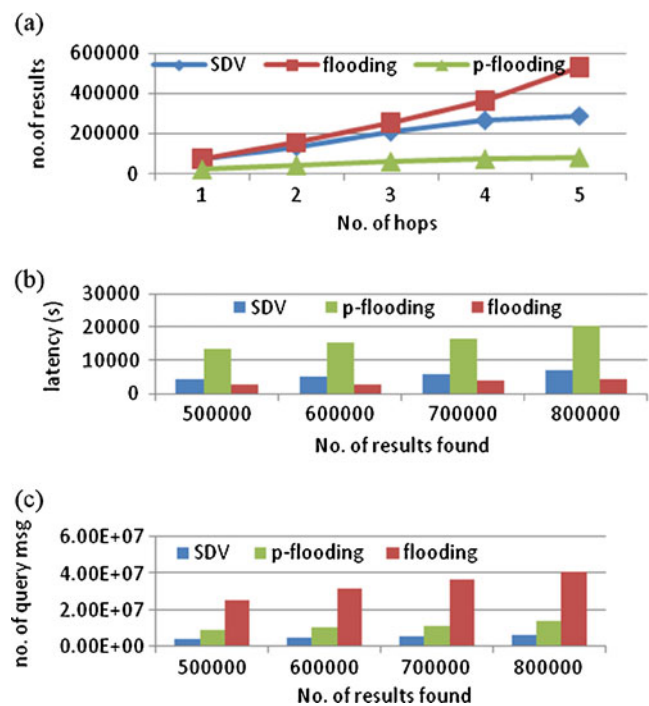


Fig. 9 The comparison of query efficiency

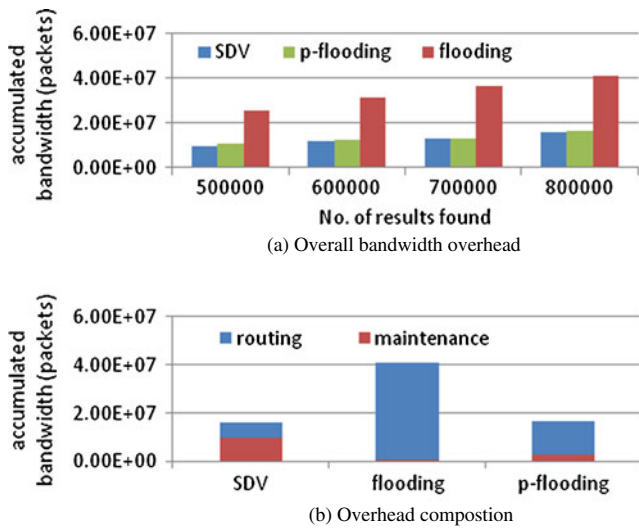


Fig. 10 The comparison of system overhead. a Overall bandwidth overhead. b Overhead composition

of the three routing protocols. Figure 10(a) compares the overall bandwidth overhead. We can see that the SDV and p-flooding techniques use much less bandwidth compared to flooding technique to get the same number of results. Figure 10(b) shows the composition of the overhead, i.e., the ratio of the query overhead and maintenance overhead. Most of the overhead of the flooding and p-flooding techniques is caused by query forwarding, and a minute overhead is caused by maintenance. For the proposed SDV technique, the maintenance overhead accounts for a higher proportion, because it proactively maintains routing tables for each node so that a route is generally available when needed. In essence, out of all three protocols, the proposed SDV technique performs best in ad hoc networks that are characterized by heavy query traffic.

Figure 11 gives an overview of how mobility affects the system performance. Specifically, it shows the number of query results returned with different moving speed and pause time. We can see that the SDV technique is resilient to mobility. It performs well in most mobile situations. In the worst case when mobility level is very high (nodes do

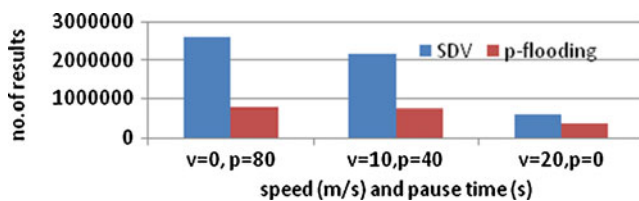


Fig. 11 The effect of mobility (v: moving speed, p: pause time)

not pause and move so fast that cannot happen in a social network reality), the performance of SDV deteriorates to a state similar to p-flooding.

Finally, Fig. 12 illustrates the influence of the radius on SDV's query performance. When radius=0, the SDV algorithm degrades to a p-flooding algorithm. Initially, increasing the radius increases the nodes' knowledge of the network, thus improving the query performance. When the radius grows to 3, nodes have a good knowledge of the network; further increasing the radius does not bring more benefit but incurs much more overhead because of a larger routing table.

As a summary, the simulation results demonstrate the unique properties and superior advantages of the proposed framework. The profile similarity measure experiment illustrates the intelligence of the ontology-based measure in discovering friends and resources compared to traditional key-word-based measurement. In the query routing experiments, we have shown that the SDV routing algorithm, compared to flooding and p-flooding, significantly improve the success rate in routing the query messages to right destinations, and at the same time produces less overhead. Moreover, it is resilient to high mobility.

4 Related work

This section gives an overview of technologies and research related to this paper.

4.1 Mobile social networking

The notion of mobile social network is not new. As mentioned in Section 1, there are many mobile social network applications and commercial products. In recent years, we have seen a few social networking applications that use ad hoc communications rather than costly Internet access. For example, Jambo Networks [18] uses Wi-Fi enabled laptops, cell phones, and PDAs to match users within walking distance that have similar interests and would like to meet face to face. There also are Bluetooth enabled commercial applications, such as Nokia Sensor

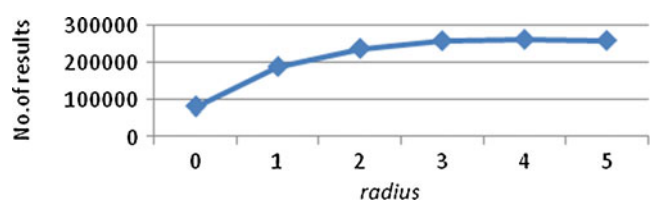


Fig. 12 The effect of radius

[33] and MobiLuck [31] that offer profile-based match-making. SMSN is similar to these applications in the sense that they all utilize ad hoc networking; however, SMSN extends the simple single-hop keyword-based profile matching to a large-scale multi-hop intelligent semantic-based profile matching, and provides a general architecture that will be germane for different social networking applications.

Besides the commercial products, there are also many research works studying the new trend of mobile social networks [51–54]: BlueDating [53] provides dating services for mobile phones with Bluetooth support. Users can specify their personal profiles and the profile of the persons that they are interested. When two persons come into physical proximity, profiles are matched. PodNet [51] uses a delay-tolerant content distribution architecture allowing exchanging multimedia content from mobiles to mobiles in an opportunistic fashion using Wifi. MEN (for Mobile Encounter Network) [54] is a short-range wireless communication scheme. MEN can be used as an information search system on the market environment. It acts as a middleware for bridging consumers and the service/product providers. MEN allows services and products being promoted at the location where they are available.

Among these existing research works, our system is most similar to the SAMOA system [52]: SAMOA also utilizes semantics-based profile modeling and matching algorithms for social-network extraction, and creates social networks among users in physical proximity. Our SMSN system has three major advantages over SAMOA system: (1) SAMOA system distinguishes users as managers and clients. The major functions such as profile matching can only be performed at the managers' site. (2) Although the paper does not explicitly point out, SAMOA system has higher computation (memory/CPU/power) requirements for the manager devices because of the complex semantic operation required by their use of Protégé and Pellet. Therefore, the role of a manager in SAMOA can only be taken by devices equipped with higher power such as desktop, laptop, and cannot be taken by handheld devices such as cell phones and PDAs which only have limited computation power. On the other hand, our SMSN system decomposes the semantic profile to small elements and simplifies the semantic operations to set operations. All peers in our system can be resource-constrained mobile devices such as PDAs and cell phones. (3) The SAMOA system uses broadcasting and IP-based point to point routing to realize communication between users. In SMSN, we propose a cross-layer routing algorithm to forward messages to related nodes only. As shown in our experiments, this routing strategy can dramatically save the network bandwidth.

4.2 Other related technologies

4.2.1 *Ad hoc network routing*

Our proposed multi-hop resource discovery routing protocol integrates semantic index with state-of-the-art ad hoc routing procedures to achieve maximal performance. Existing ad hoc routing protocols may be grouped into three different categories: proactive/table-driven, reactive/on-demand, and hybrid. Proactive protocols maintain routing tables for each node and periodically update the table to track the changes in network topology. They are more suitable for small-scale MANETs with low node mobility and heavy traffic. DSDV [39], WRP [30], OLSR [5], CGSR [6], and TBRPF [34], are examples of this type of protocols. Reactive routing protocols, on the other hand, find and maintain routes only when needed. The obvious advantage with discovering routes on-demand is to avoid incurring the cost of maintaining routes that are not used. Examples of this type of protocols are DSR [22], AODV [40], and TORA [37]. Previous experiments show that hybrid protocols can provide a better compromise between communication overhead and delay as well as better scalability. There have been efforts of combining proactive and reactive approaches, such as ZRP [15], ZHLS [21], HSLs [44] and CBRP [20]. Our proposed SDV procedure may be considered to be similar to hybrid protocol; however, the SDV procedure: (a) integrates semantic resource information with underlying routing procedures, (b) proactively maintains resource routing information of nearby users, and (c) use long-distance links to discover farther resources on-demand.

4.2.2 *Semantic web and ontology*

In our work, we employ techniques from the Semantic Web to define user profile thus making search more expressive and intelligent. The Semantic Web relies on ontologies [12] that structure underlying data for the purpose of comprehensive and transportable machine understanding. FOAF (Friend-of-a-Friend) [14] is an ontology to describe user profiles, friends, affiliations, creations etc. It has been used in some social network applications, such as Facebook to define user profiles. As the mobile ad hoc nature of our system, we cannot use FOAF directly; instead, we adopt partial concepts from FOAF and add more dynamic concepts. Von Hessling et al. [47] propose a model where semantic user profiles are used in a P2P mobile environment. The user profile is represented with simple Description Logics(DL) [2], consisting simply of the union of interests and disinterest. The DL is used to match services and profile description. SMSN is the first system that integrates ontologies not only in profile definitions but also in

matchmaking and routing; therefore SMSN is by definition a fully semantics-based social network, and the first one in existence.

4.2.3 Privacy preserving control

Secure two party computation was first investigated by Yao [49] and was later generalized to multiparty computation. Goldreich proves that there exists a secure solution for any functionality [11]. It shows that the size of the protocol depends on the size of the input. To reduce the computation complexity for large inputs, several algorithms have been proposed for efficient computation on specific functionalities. Examples include secure summation [45] scalar product [1], secure set union [23], and private permutation [9]. Our paper considers private-preserving set intersection operation.

5 Conclusion

This paper introduced a new network application scenario for social interaction – the semantics-based mobile ad hoc social network, SMSN. The SMSN system will eventually revolutionize the ways that humans interact with each another by removing the barriers resulting through unfamiliarity. In particular, we designed an ontology model to define users' profile and proposed a semantics-aware discovery mechanism to locate users with similar interests to create social network. With SMSN, social network applications, such as resource sharing, profile browsing, and instant messaging, can be efficiently performed. Moreover, SMSN is able to support an entire new class of advanced semantics-aware social network applications.

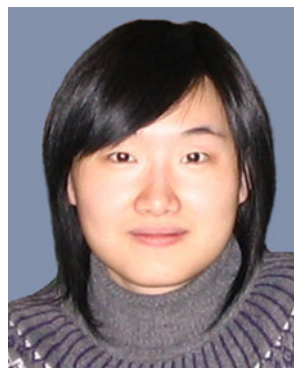
References

- Atallah MJ, Du W (August 8–10 2001) Secure multi-party computational geometry. In Seventh International Workshop on Algorithms and Data Structures (WADS 2001), Providence, Rhode Island, USA.
- Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (2003) The description logic handbook: theory, implementation, applications. Cambridge University Press, Cambridge
- Bettstetter C, Wagner C (2002) The spatial node distribution of the random waypoint mobility model. In *Proc. WMAN*.
- Bluepulse website: <http://www.bluepulse.com/>.
- Clausen T, Jacquet P (October 2003) Optimized Link State Routing Protocol (OLSR). *RFC 3626, IETF Network Working Group*.
- Chiang CC, Wu HK, Liu W, Gerla M (April 1997) Routing in clustered multihop, mobile wireless networks with fading channel. *Proceedings of IEEE Singapore International Conference on Networks (SICON)*, Singapore.
- Damgard I, Jurik M (2001) A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *4th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC '01)*, LNCS 1992, pp 119–136.
- Dodge ball website: <http://www.dodgeball.com/>.
- Du W, Atallah MJ (December 10–14 2001) Privacy-preserving statistical analysis. In *Proceeding of the 17th Annual Computer Security Applications Conference*, New Orleans, Louisiana, USA.
- Freedman M, Nissim K, Pinkas B (May 2004) Efficient private matching and set intersection. In *Advances in Cryptology – Eurocrypt '04*, volume 3027 of LNCS, pp 1–19. Springer-Verlag.
- Goldreich O, Micali S, Wigderson A (1987) How to play any mental game - a completeness theorem for protocols with honest majority. In *19th ACM Symposium on the Theory of Computing*, pp 218–229.
- Gruber TR (1995) Principles for the design of ontologies used for knowledge sharing. *Int J Hum-Comput Stud* 907–928.
- Gruver WA, Boudreaux JC (1993) Intelligent manufacturing: programming environments for CIM. Springer-Verlag, London
- FOAF website: <http://www.foaf-project.org/>.
- Haas Z, Pearlman M, Samar P (2002) Zone Routing Protocol (ZRP). IETF Internet Draft.
- Haas Z, Halpern J, Li L (2002) Gossip based ad hoc routing. In *Proc. Of INFOCOM*.
- Jaiku website: <http://www.jaiku.com/>.
- Jambo Networks: <http://www.jambo.net>.
- Jiang J, Conrath D (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceeding of the Int'l Conf. Computational Linguistics (ROCLING X)*.
- Jiang M, Li J, YC Tay YC (July 1999) Cluster Based Routing Protocol (CBRP) functional specification. IETF Internet Draft, draft-ietf-manet-cbrp-spec-01.txt.
- Joa-Ng M, Lu I (1999) A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE J Sel Area Comm* 17(8):1415–1425
- Johnson D, Maltz D, Hu Y-C (July 2004) The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, draft-ietfmanet-dsr-10.txt.
- Kantarcioglu M, Clifton C (2004) Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Trans Knowl Data Eng* 16(4).
- Kissner L, Song D (August 2005) Private and threshold set-intersection. In *Advances in Cryptology – CRYPTO'05*.
- Lee J, Kim M, Lee Y (1993) Information retrieval based on conceptual distance in IS-A hierarchies. *J Doc* 49:188–207
- Lenders V, May M, Plattner B (2005) Service discovery in mobile ad hoc networks: a field theoretic approach. In *Proc. of the WoWMoM*.
- Li J, Vuong S (2006) Grid resource discovery based on semantic P2P communities. In *Proc. of the 21st ACM SAC*, pp 754–759.
- Li J (2009) Building distributed index for semantic web data. In *Proc. of the 23rd IEEE AINA*.
- Li J, Vuong S (2008) SOON: a scalable self-organized overlay network for distributed information retrieval. In *Proc. of the 19th IFIP/IEEE DSOM*. pp 1–13.
- Murthy S, Garcia-Luna-Aceves JJ (November 1995) A routing protocol for packet radio networks. *Proceedings of the First Annual ACM International Conference on Mobile Computing and Networking*, Berkeley.
- MobiLuck: <http://www.mobiluck.com/>.

32. Ni S, Tseng Y, Chen Y, Sheu J (1999) The broadcast storm problem in a mobile ad hoc network. *Proc. of the 5th ACM/IEEE MobiCom*, pp 151–162.
33. Nokia Sensor: <http://www.nokia.com/>.
34. Ogier R, Templin F, Lewis M (February 2004) Topology dissemination based on reverse-path forwarding (TBRPF). <http://www.ietf.org/rfc/rfc3684.txt>, Internet Draft, IETF.
35. OWL Web Ontology Language: <http://www.w3.org/TR/owl-features/>.
36. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. *Advances in Cryptology – EUROCRYPT 1999*, LNCS 1592:223–238.
37. Park VD, Corson MS (April 2002) Temporally-Ordered Routing Algorithm (TORA) version 1: functional specifications. *Internet draft, draft-ietf-manet-tora-spec-01.txt*.
38. Pedersen T, Patwardhan S, Michelizzi J (2004) WordNet: similarity-measuring the relatedness of concepts. *In Proc of AAAI*.
39. Perkins CE, Watson TJ (1994) Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for mobile computers. *In Proc of ACM SIGCOMM*, pp 234–244.
40. Perkins C, Belding-Royer E, Das S (2003) Ad hoc On-Demand Distance Vector (AODV) routing. *RFC 3561*.
41. Plazes website: <http://plazes.com/>.
42. Rada R, Mili H, Bicknell E, Blettner M (1989) Development and application of a metric on semantic nets. *IEEE Trans Syst Man Cybern*.
43. Resource Description Framework (RDF): <http://www.w3.org/RDF/>.
44. Santivanez CA, Stavrakakis I, Ramanathan R (October, 2001) Making linkstate routing scale for ad hoc networks. *In: Proceedings of MobiHoc 2001, Long Beach*.
45. Schneier B (1995) *Applied cryptography*, 2nd edn. Wiley.
46. SMSN website, <http://www.cs.ndsu.nodak.edu/~juali/MobiSN.html>.
47. von Hessling A, Kleemann T, Sinner A (2–2005) Semantic user profiles and their applications in a mobile environment. *Fachberichte Informatik*.
48. Wu Z, Palmer M (1994) Verb semantic and lexical selection. *In Proc. of the 32nd Annual Meeting of the Association for computational Linguistics*.
49. Yao AC (1986) How to generate and exchange secrets. *In Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pp 162–167. IEEE.
50. SMSN open source Google Code website, <http://code.google.com/p/mobisn/>.
51. May M, Karlsson G, Lenders V (2007) A system architecture for delay-tolerant content distribution. *In WRECOM*.
52. Bottazzi D, Montanari R, Toninelli A (2007) Context-aware middleware for anytime, anywhere social networks. *IEEE Intel Syst* 22(5):23–32
53. Beale R (2005) Supporting social interaction with smart phones. *IEEE Pervasive Computing* 4(2):35–41
54. Korhonen VA, Pyykkönen R (2009) Creating context as you go. *In Proc. of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, pp 37–40.



Juan Li received her B.S. in computer science in 1997 from the Northern Jiaotong University, and her M.S in computer science in 2001 from the Chinese Academy of Sciences, both in Beijing, China. She received her Ph.D. in computer science in 2008 from the University of British Columbia, Vancouver, Canada. She is currently an assistant professor at North Dakota State University, USA. Her research interests include distributed system, mobile wireless network, and semantic web.



Hui Wang received the BS degree in computer science from Wuhan University in 1998, the MS degree in computer science from University of British Columbia in 2002, and the PhD degree in computer science from University of British Columbia in 2007. She has been an assistant professor in the Computer Science Department, Stevens Institute of Technology, since 2008. Her research interests include data management, database security, and data privacy.



Samee U. Khan received the BS degree in computer systems engineering from the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, in 1999 and the PhD degree from the University of Texas, Arlington, Texas, in 2007. He is currently an assistant professor in the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, North Dakota. His research has been disseminated in over 50 publications and supported by research grants from various agencies that total in over 0.7 M. His research interests include designing game theoretical resource allocation algorithms for parallel and distributed computing systems.