MobiSN: Semantics-based Mobile Ad Hoc Social Network Framework

Juan Li

Department of Computer Science
North Dakota State University
Fargo, ND 58108, USA
j.li@ndsu.edu

Samee Ullah Khan

Department of Electrical and Computer Engineering
North Dakota State University
Fargo, ND 58108, USA
samee.khan@ndsu.edu

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 a concrete, generalized, and novel framework to develop a fully functional mobile ad hoc social network. The proposed framework provides effective and efficient solutions to social network construction, semantics-based user profile matching, and multi-hop semantics-based routing. Moreover, the proposed framework because of its generality is applicable in applications of critical importance, such as disaster-recovery, homeland security, and personnel control. Furthermore, the proposed framework is rigorously benchmarked using an elaborate simulation setup and released as a prototype system that can be run on cellular phones.**

Keywords-social networks; mobile computing; collaborative computing; ad hoc networks; sematics

I. 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. 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. Plazes [21] and Bluepulse [3], on the other hand, 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, and restaurants enables people to communicate and share their experiences without the need to have Internet access and with minimum required infrastructure. To address this need, we propose a mobile social network framework referred to as MobiSN (acronym for Mobile Social Network). MobiSN shifts 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.

Unlike traditional social networks in which social communities are pre-built from real-world relationships, users in MobiSN are automatically grouped by their social behaviors. This raises the challenge of discovering users of similar social patterns. Exiting applications, such as Jambo Networks [9] and Nokia Sensor [19] construct friendship relation between mobile users by mapping keywords embedded in their user profiles. However, the network may consist of various types of mobile devices and users may use different vocabularies and categorizations, 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. To circumvent these problems, MobiSN proposes a semantics-based matchmaking scheme for discovering similar users. The 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 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.

Due to mobility, limited battery power, constrained bandwidth, lack of centralized control, and specialized infrastructure demand, constructing an ad hoc social network poses new design challenges compared to conventional web-based social networks. MobiSN addresses these challenges by proposing a semantics-based multi-hop routing protocol for social network construction and query forwarding that exploits the interactions between the overlay routing and physical routing protocols to optimize the routing performance.

The rest of the paper is organized as follows. Section II describes the architectural and system design of the MobiSN system. In Section III, we evaluate the proposed method and show the effectiveness of MobiSN with a comprehensive set of simulations. Related work and concluding remarks are provided in Sections IV and V, respectively.

II. System Design

A. Overview

The system architecture of the proposed MobiSN is depicted in Fig. 1. MobiSN 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 by virtual links. Each virtual link corresponds to a communication path that may be composed of several physical hops. Nodes connected via virtual links are referred to as *friends*. Once friendship relations are established, friends can perform social operations, such as sharing resources, sending messages, and browsing profiles of each other.

To materialize the functionality of mobile social network, the proposed MobiSN system architecture consists of three main components: (a) profile manager, (b) matchmaker, and (c) routing controller. The profile manager creates and stores ontology-based user profiles and/or resource metadata. The matchmaker intelligently matches user profiles keyed by their similarities, so that users with similar interests can be connected (or made friends) in the social network layer. To effectively and efficiently discover *friends* or resources, we design a novel semantics-based multi-hop query routing controller. These three components (enumerated above) must work in concert to lay the foundations of a general purposed mobile ad hoc social network. Below we detail the three essential components.

B. Ontological Profile

In MobiSN, each user has an ontology-based profile. Utilizing ontology to represent user profile (or resource metadata) makes the representation more expressive and less ambiguous than a keyword-based representation (that is the current state-of-the art) [13]. We separate the ontology definition into two parts: (a) terminological box (T-Box) and (b) assertion box (A-Box). (These boxes are defined in the description logic terminology [1]) In MobiSN, the T-Box ontology defines the common understanding for all of the important concepts and their corresponding relationships. The A-Box is a finite set of assertion axioms, which includes a set of axioms for the descriptions of concrete data and relations. We adopt some concepts from FOAF [5], such as *Person*, *Interest*, *Image*, *Name*, and *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 MobiSN. 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. Due to the lack of space, we are restricted to provide only surface-level information pertaining to ontology. We encourage the readers to find more details about our proposed ontology by referring to [11]. 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.
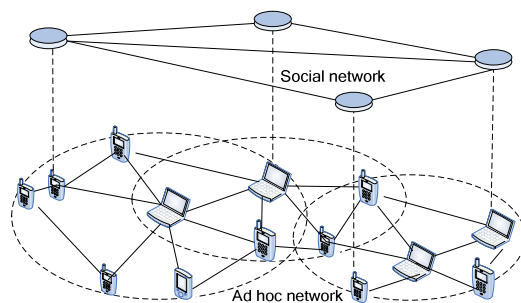

Figure 1. System overview.

C. Profile Matchmaking

The MobiSN matchmaker component can measure the similarity (a) between user profiles and (b) between queries and resources. Given two user profile instances, the MobiSN matchmaker is able to overcome differences in vocabularies and can support inference mechanisms. For instance, a user interested in "mobile networks" (superclass of "ad hoc network") can be matched with another user interested in "ad hoc networks". Conversely, a user interested in "peer-to-peer computing" can also be inferred to be 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.

We must note that because of diversity and the potentially large number of instance elements, profile matching on instance (A-Box) level can be very complex and sensitive to vocabularies. To circumvent these problems, we employ *projection* as a technique to reduce the complexity of matching, i.e., a particular instance is mapped to the shared T-Box ontology. Such projection warrants that ontology distance can be computed with respect to the same T-Box ontology graph. Once a projection is performed, we adopt the idea of distance-based approach [14], [20], [22] to measuring the semantic similarity between ontology concepts. The basic idea behind the distance-based ontology is to identify the shortest path between two concepts in terms of the number of edges and then translate that distance into semantic distance. Our approach improves the accuracy by integrating factors, such as the depth of a node in the hierarchy and the type of links. Below we formally define and explain the similarity measurement.

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

Definition 2 (*Concept Distance*). The semantic distance between two concepts C_a and C_b in an 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 tree, C_{i+1} is C_i's parent, and w_i is the weight of edge presented as a distance factor.
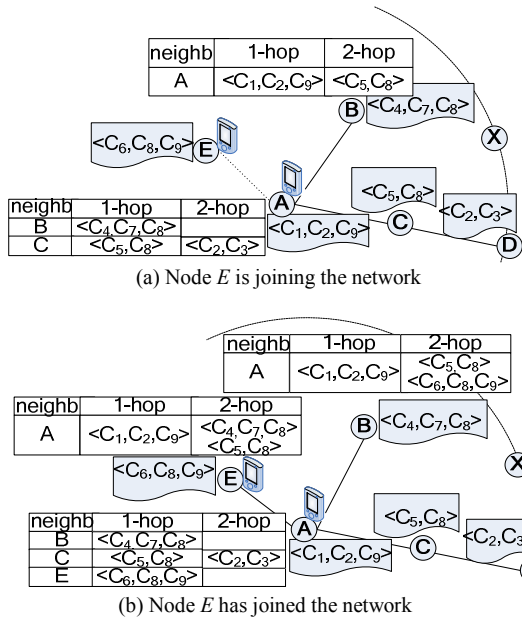
(a) Node E is joining the network

(b) Node E has joined the network

Figure 2.   Maintaining routing indices (radius =2).

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

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

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 ($0<t\leq1$), the profile P_x is said to be semantically related to P_y.

D. Discovery Routing

In social networks, an important issue is to discover other peers with similar interests, nodes with interested contents, and membership servers (for multicast sessions). Due to the absence of a centralized intelligence, discovery in mobile ad hoc networks is a very challenging predicament. To circumvent such problems, we propose a Semantics-based Distance-Vector (SDV) query routing algorithm, in which a node can make routing decisions by knowing only its immediate neighbors and limited resource information. The SDV routing is only used as a hint to find matching nodes. When potential matching nodes have been located, the discovery query is redirected to the candidate nodes for the finial matchmaking. 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.

1) Routing table construction

Each node maintains a routing table, allowing a node to select the best neighbors to transmit query. The routing table contains both local and neighboring ontology information at various distances. Each entry in the routing table depicts resources available from a neighbor. Each entry (row) contains k columns for resources at different distances within $[1, k]$ 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 from that neighbor. For example, in Fig. 2 (a), by its routing table, node A knows it can reach nodes B and C's ontology vector within 1-hop, and a vector $<C_2, C_3>$ within 2- hops through node C.

Fig. 2 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.) 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 C) of this update.

To reduce the overhead of transmitting routing information, a soft-state update mechanism is used, in which routing information is exchanged periodically. Moreover, the storage required by a routing table is modest as we only store T-Box concepts in vectors, which is of a limited size. Furthermore, duplicated vectors are removed before being sent to neighbors.

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., $sim(X,Q)>t$, 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. 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 neighbors closest to the current node in network latency. If there is no match, the querying node searches for a match in the 2-nd column and so on. A query may be transferred several hops until arriving at the matching node or the query TTL expires.

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.

E. Prototype Implementation

The MobiSN 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 a social network. These features include: profile management, friend matchmaking, and routing control. We also implemented the ad hoc multi-hop text messaging and file sharing as part of the service pack for MobiSN. Fig. 3 shows the screenshot of an application running on the SonyEricsson_K750 cellular phone. The prototype can be downloaded from MobiSN website [16].

Figure 3.   MobiSN prototype running on the SonyEricsson_K750.

TABLE I
PARAMETERS USED IN THE SIMULATIONS

parameter	range (default)
network size	100-1100 (500)
environment area	200m x 200m
node moving speed	1-20m/s (1m/s)
node transmission rage	10m
node pause time	0s-80s (20s)
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	0.6, 0.8 (0.8)
no. of queries per simulation session	20K

III.   EXPERIMENTAL SETUP AND DISCUSSION OF RESULTS

To effectively simulate MobiSN, an enclosed ad hoc network environment was considered. The enclosed area that contained different nodes was off an area of 200m x 200m. 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 [2]. In the random waypoint 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. Once a node reaches its destination, the node pauses for a predefined interval of time, and then it repeats this movement pattern. The transmission range of a node was predetermined to be 10 m.

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 I.

We begin to evaluate MobiSN by first recording the effectiveness of our similarity measure for *friend* discovery. We compare our ontology-based similarity measure with semantics-free exact-match. For the exact-match-based search, a query only matches keywords without taking into account the keywords' specific meanings 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.

Fig. 4 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. 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, MobiSN users may have the leverage to relax the criteria for *friend* matching to construct a social network with sparse user population. (An option that is not available in any of the currently existing social networks.) Moreover, because ontological search eliminates the semantic ambiguity problem, such as polysemy and homonymy, results returned are relevant with extremely high precision.

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 [7] , [17]. The flooding-based technique was chosen as a reference approach for its simplicity and prevalence, which in fact, made it a widely used baseline for

many previous research efforts [10]. The p-flooding technique is a controlled flooding that 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.

Fig. 5 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 technique is much higher compared to the other two techniques. In contrast, the proposed SDV technique identifies comparative number of results and incurs much less query overhead compared to the other two techniques.

Fig. 6 illustrates the bandwidth overhead that includes the query and maintenance overheads to implement each of the routing protocols. Fig. 6(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. Fig. 6(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 SDV proactively maintains routing tables for each node so that a route is generally available when needed. In essence, the proposed SDV technique performs best in ad hoc networks that are characterized by heavy query traffic.

Fig. 7 provides 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 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.

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 improves the success rate in forwarding the query messages to right destinations, and at the same time produces less overhead. Moreover, it is resilient to high mobility. Due to the lack of space, we do not provide further benchmark results for MobiSN; however, we encourage the readers to browse [12] for extended discussion and reports on several MobiSN benchmarking results.
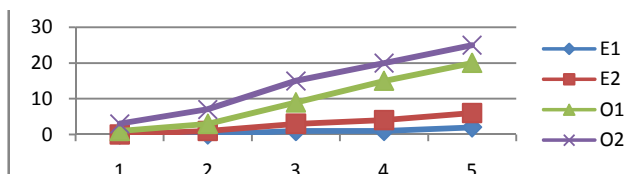

Figure 4. Number of friends discovered vs. routing hops.
E1: Exact instance-level match, E2: Exact class-level match, O1: Ontology similarity measure with t=0.8, O2: Ontology similarity measure with t=0.6.
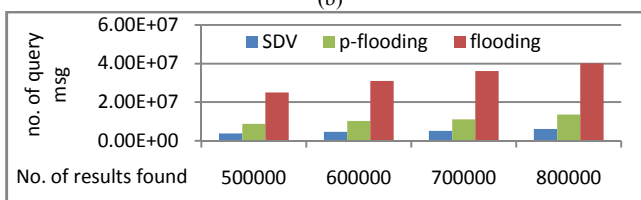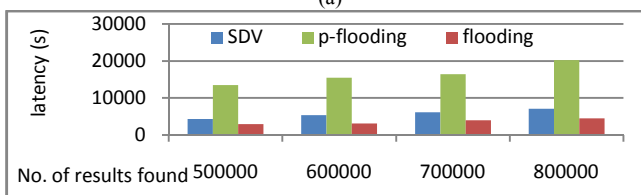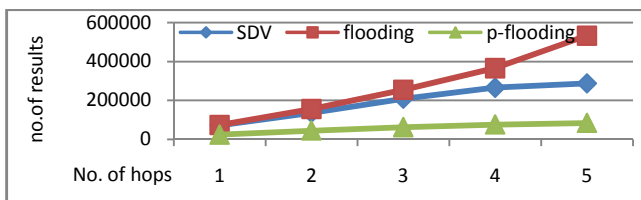

(a)

(b)

(c)

Figure 5. The comparison of query efficiency.
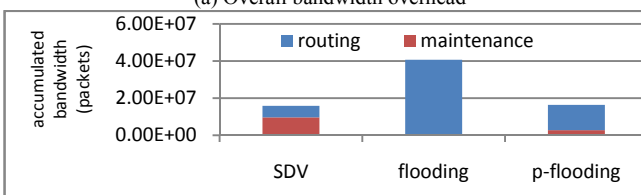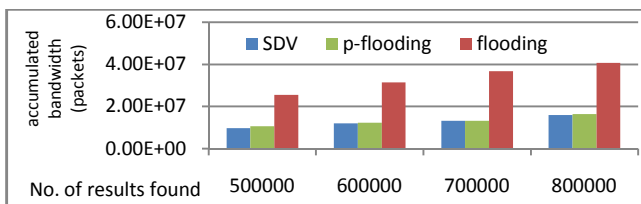

(a) Overall bandwidth overhead

(b) Overhead compostion

Figure 6. The comparison of system overhead.
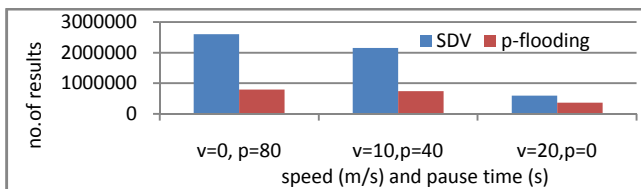

Figure 7. The effect of mobility (v: moving speed, p: pause time).

IV. RELATED WORK

In the recent years, we have observed a surge in social networking applications that use pure ad hoc communications rather than costly Internet access. For example, Jambo Networks [9] 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 [19] and MobiLuck [15] that offer profile-based matchmaking. In Nokia Sensor and MobiLuck applications, users are required to maintain a profile that is exchanged for another user's profile when devices are within communication range. MobiSN may be considered similar to the above mentioned applications because all utilize ad hoc networking. However, MobiSN extends the simple single-hop keyword-based friend matching to a larger-scale multi-hop intelligent semantic-based friend matching, and provides a general architecture that will be germane for different social networking applications.

There have been many service discovery strategies proposed for MANET [4], [8], [18], [23], [24]. In MobiSN, friends and resources can be viewed as a special kind of services – providing friendship or other services. Therefore, the friend/resource discovery can be viewed as a service discovery. Due to the complexity and heterogeneity of user (or resource) profiles in mobile social network, existing resource discovery approaches (based on matching identifiers or attribute-value pairs) cannot be used to discover friends in ad hoc social network. In most cases, a user may not know the appropriate attributes of friends. Our proposed friend-discovery protocol intelligently integrates semantic similarity with state-of-the-art ad hoc routing procedures to achieve maximal performance.

Ontologies [6], such as FOAF [5] have been used in some social network applications, such as Facebook to define user profiles. MobiSN is the first system that integrates ontologies not only in profile definitions but also in matchmaking and routing. Therefore MobiSN is by definition a fully semantics-based social network, and the first one in existence.

V. CONCLUSION

This paper introduced a new network application scenario for social interaction – the mobile ad hoc social network, MobiSN. The MobiSN system will eventually revolutionize the way humans may interact with one 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 MobiSN, social network applications, such as resource sharing, profile browsing, and instant messaging, can efficiently be performed. Moreover, MobiSN is able to support an entire new class of advanced semantics-aware social network applications.

REFERENCES

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, "The Description Logic Handbook: Theory, Implementation, Applications," Cambridge University Press, Cambridge, UK, 2003.

[2] C. Bettstetter and C. Wagner, "The Spatial Node Distribution of the Random Waypoint Mobility Model," in Workshop on Mobile Ad Hoc Networks, 2002, pp. 41-58.

[3] Bluepulse, available at: http://www.bluepulse.com/

[4] C. Campo, M. Munoz, J. C. Perea, A. Marın, C. Garcıa-Rubio, "PDP and GSDL: A New Service Discovery Middleware to Support Spontaneous Interactions in Pervasive Systems," in 3rd International Conference on Pervasive Computing and Communications Workshops, 2005, pp. 178-182.

[5] FOAF, available at: http://www.foaf-project.org/

[6] T. R. Gruber, "Principles for the Design of Ontologies Used for Knowledge Sharing." International Journal of Human-Computer Studies, pp. 907-928, 1995..

[7] Z. Haas, J. Halpern, and L. Li, "Gossip Based Ad Hoc Routing", in 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2002, pp. 1707-1716.

[8] S. Helal, N. Desai, V. Verma, C. Lee, "Konark - A Service Discovery and Delivery Protocol for Ad-hoc Networks", in IEEE Wireless Communications and Networking Conference (WCNC), 2003, pp. 2107-2113.

[9] Jambo Networks, available at: http://www.jambo.net.

[10] V. Lenders, M. May, and B. Plattner, "Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach," in 6th IEEE International Symposium on World of Wireless Mobile and Multimedia Networks (WoWMoM), 2005, pp. 120-130.

[11] J. Li "Building Distributed Index for Semantic Web Data", in International Conference on Advanced Information Networking and Applications (AINA), 2009, pp. 660-667.

[12] J. Li and S. U. Khan, "On How to Construct a Social Network from a Mobile Ad Hoc Network", Department of Computer Science, North Dakota State University, Technical Report, NDSU-CS-TR-09-009, 2009.

[13] J. Li, S. Vuong, "Grid Resource Discovery Based on Semantic P2P Communities", in 21st ACM Syposium on Applied Computing (SAC), 2006, pp.754-759.

[14] J. Li and S. Vuong, "SOON: A Scalable Self-Organized Overlay Network for Distributed Information Retrieval", in 19th IFIP/IEEE International Workshop on Distributed Systems (DSOM), 2008, pp.1-13.

[15] MobiLuck, available at: http://www.mobiluck.com/.

[16] MobiSN, available at: http://cs.ndsu.nodak.edu/~juali/MobiSN.html.

[17] S. Ni, Y. Tseng, Y. Chen , J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),1999, p.151-162

[18] M. Nidd, "Service Discovery in DEAPspace", IEEE Personal Communications, vol. 8, no. 4, pp. 39-45, 2001.

[19] Nokia Sensor, available at: http://www.nokia.com/.

[20] T. Pedersen, S. Patwardhan, J. Michelizzi, "WordNet: Similarity-Measuring the Relatedness of Concepts," in 19th National Conference on Artifical Intelligence (AAAI), 2004.

[21] Plazes website, available at: http://plazes.com/.

[22] R. Rada, H. Mili, E. Bicknell, M. Blettner. "Development and Application of a Metric on Semantic Nets," IEEE Transaction on Systems, Man, and Cybernetics, vol. 19, no. 1, pp. 17-30, 1989.

[23] F. Sailhan, V. Issarny: "Scalable Service Discovery for MANET," in 3rd IEEE International Conference on Pervasive Computing and Communications, 2005, pp. 235-244.

[24] UPnP Forum: Understanding Universal Plug and Play, White Paper, available at: http://www.upnp.org.