

## Research Article

### Middleware Architecture for Ad-hoc Social Network

Nagender Aneja and Sapna Gambhir

YMCA University of Science and Technology, Faridabad, India

**Abstract:** The aim of this study is to propose architecture for middleware for Ad-hoc Social Networks. Many social relationships in a limited geographical area are ad-hoc, e.g., in conference or tourist locations where people meet each other for a specific purpose and may not have access to Internet via Wi-Fi or mobile connection. The Ad-hoc Social Network (ASN) helps to meet temporary location-based social needs by creating a social network where like-minded users participate in social communication connected by an infrastructure-less ad-hoc network. This study presents middleware architecture for ASNs that provides software developers a platform for developing mobile apps that enable social connections. The proposed architecture is a software suite that includes an application layer, transport layer, ad-hoc social layer and ad-hoc communication layer. This study also presents different functions and features required for each layer in ASN.

**Keywords:** Architecture, ad-hoc social network, ad-hoc network, middleware, peer-to-peer network, social network

#### INTRODUCTION

Conventional social networking applications allow users to post text, images and videos among other pre-connected users. Many social networking websites, such as Facebook, LinkedIn and Twitter, enable members to connect and share their information electronically with other users with whom they have a pre-established connection, or to share publicly. However, conventional social networking applications have limitations when a user wants to have a shared experience with other individuals with whom the user does not have a pre-established connection, or when users are interested only in a temporary social network. Most of the time, social networks that users are interested in are based on a specific purpose at a certain location; thus, the ASN can be very beneficial (Aneja and Gambhir, 2015; Mangla and Gambhir, 2015). The ASN is a social network established between similar users connected by an ad-hoc communication mode. The ASN can also be useful to home consumers with door-to-door salespersons before answering their call and in some cases to check the salesperson's authenticity. Users may need a temporary social network at their business, at academic meetings and conferences, sports events, or disaster events, e.g., road accident.

Users may also share or connect with other users having similar interests by forming ad-hoc networks even when wireless coverage is poor or not available.

Further, in many parts of the world, Internet via smart phones or other portable devices is not available or affordable. According to McKinsey's (2014) report, more than 60% of the global population is still offline. Therefore, there is a need to provide social network applications that do not rely on the Internet. Social networks over an ad-hoc network can be easily formed using wireless standards like IEEE 802.11b, IEEE 802.11g, IEEE 802.11a, Bluetooth (IEEE 802.15.1), Ultrawideband (IEEE 802.15.3), IEEE 802.15.4 (Zigbee), HiperLAN2, IrDA, HomeRF, IEEE 802.16 (Broadband wireless) and IEEE 802.16a, IEEE 802.16e (Hoebeke *et al.*, 2004).

Various techniques have been proposed to create ASNs (Aneja and Gambhir, 2012; Gambhir and Aneja, 2013); however, these techniques either still exploit online social networks, e.g., Facebook or LinkedIn, or use ad-hoc networks for temporary communication, or have not provided a full system architecture to create an ASN independent of online social networks. Further, some users of online social networks may have thousands of connections; thus, synchronizing the full list of all connections of an online social network in mobile devices with limited memory also poses a challenge.

In many applications of ASNs, the number of users within even a limited range could be very large, e.g., in a concert or conference. The high density of nodes pose further challenges to create an ASN using mobile devices with limited memory and power. There is a

**Corresponding Author:** Nagender Aneja, YMCA University of Science and Technology, Faridabad, India, Tel.: +91-9810501817, +673-8189906

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

need to configure parameter of similarity metric in the case of high node density so as to reduce the number of users that are less similar. In high-density scenarios, there may be lots of replies in response to the sending profile of a user. Therefore, there is a need for a mobile device mechanism to handle these cases.

This study presents end-to-end software architecture for Ad-hoc Social Network Middleware. The presented architecture is a layered architecture that combines similar functions in one layer. Core functions e.g. profile store, privacy modules and matching engine required to establish ad-hoc social network have been combined in Ad-hoc social layer.

## LITERATURE REVIEW

Bottazzi *et al.* (2007) proposed a middleware (Socially Aware and Mobile Architecture, SAMOA) that creates anytime, anywhere social networks among users in physical proximity. SAMOA comprises a place profile, user profile, discovery profile and filtered user profile. The place profile includes a unique identifier, name, description of the place and social activities that characterize the place. The user profile includes their name, age, gender, education and interested user activities. The protocol assumes that places where users move and operate will influence their activities and interactions with other users. SAMOA managers have a discovery profile associated with each place they manage. The discovery profile defines the preferences clients must match to join the manager's social network. Therefore, SAMOA assumes that different places have different characteristics and users in those places have relationship with their current places. However, it is quite possible that different users in the same location may have different profiles or place attributes from the manager's. The proposed work in this study assumes that browsing history at a given location may give a reasonably accurate estimate of current interests and should be included in the current local profile to eliminate the need for a place profile.

Podobnik and Lovrek (2011) proposed an ad-hoc social network service (ahSNS) to create and manage membership to an ad-hoc social network (ahSN). It also manages the transfer of user profiles and other user information from a social network service (SNS) and ahSN. Therefore, even though the authors proposed a mechanism for an ASN, it is only an extension of traditional social networks as profiles are transferred from a traditional social network to an ASN. This study assumes that location-based interests, along with user attributes, are more advantageous for an ASN than interests or user attributes mentioned in the profiles for traditional social networks. Thus, there is a need for architecture as proposed in this study to provide a mechanism to manage both local and global profiles. The local profile is automatically built from the search and browsing history at the users' different locations.

The global profile is a union of all local profiles and is superior to static profiles currently in use by most of social networking services.

Pietiläinen *et al.* (2009) proposed MobiClique, a middleware for mobile social networking, to allow users to maintain and extend their online social networks without using any central server or infrastructure connectivity. The MobiClique allows the user to create a local profile by synchronizing with online social networks, e.g., Facebook and extending it through the ad-hoc network. However, MobiClique uses a store-carry-forward technique for communication and is, thus, more advantageous for disconnected delay-tolerant networks than for location-based ASNs. Further, MobiClique also extends the traditional social network by using the same profile.

Al-oqily *et al.* (2013) proposed that ASNs are more advantageous at places where Internet connectivity is not available, e.g., buses and trains, disaster areas, conferences and university campuses. The authors simulated an ASN with the help of GloMoSim and users' static profiles, including name, age, gender, address, interests and activities. The authors proposed a number of components required for an ASN; however, no mechanism was proposed to derive the local interests of the user. Further, the processes and components proposed by Al-oqily *et al.* (2013) are not easily configured by users.

Dodson *et al.* (2011) proposed an application-level communication protocol and library for writing mobile applications for ad-hoc groups without centralized application servers. The authors proposed that applications be developed using a generic switchboard (or chatroom) service for communication, where a session initiator invites participants to join. This protocol works like a chatroom, where anyone can chat with anyone. The disadvantage of this protocol is that a user may find their profile does not match (Dodson *et al.*, 2011). The protocol also does not provide any method to control user privacy.

Antoniadis and Courcoubetis (2006) presented a number of issues that should be considered when developing mobile social software in a multi-hop peer-to-peer environment. The authors also suggested that there is a need for a mechanism that solves the technical challenges of ASNs. Trieu and Pham (2012) proposed a decentralized approach to build interest-based ASNs using peer-to-peer architecture that brings ownership of user data back to the owner. The system includes three components: user application, keychain storage and verification agents. The decentralized user application runs on a mobile device. Keychain storage is a secure place in the device to store certificates and keys. Verification agents qualify users to join and communicate with other users. The authors proposed the system to handle privacy issues; however, the

system is not an end-to-end architecture that is easily customized.

Several authors (Liaqat *et al.*, 2015; Xia *et al.*, 2014; Liaqat *et al.*, 2014) emphasized and proposed secure routing for ASNs. However, current architecture is very specific and lacks the functionality to configure ASNs according to users' requirements. Further, these requirements are continually changing in this dynamic world; therefore, there is a need for general middleware that provides access to users for further customization and experimentation.

Rana *et al.* (2015) considered user context and social relations in creating ASNs. The users' context is determined by the frequency and duration of communications with others through phone calls, emails and social networks. However, user context as proposed in this study is an extension of the user's online social network. Further, there is limited scope to configure the user's context based on personal choices.

Zhang *et al.* (2014) proposed a hierarchical system architecture involving device, network, social and application layers. However, the system assumes a centralized database of users and new users can be added from phone numbers in the contact list. Therefore, the system is more suitable to share text, images and videos with other registered users. However, the system lacks the management of social profiles and connection to users of similar interests.

The literature reviewed here has outlined existing architecture and methods for ASNs. While prior efforts have been made to propose ASN architecture, current research lacks a general middleware architecture that can be used to create ASNs with different features and provide flexibility to set certain parameters and, thus, a greater user experience. The proposed middleware architecture expands the paradigm and potential for ASNs.

## MATERIALS AND METHODS

To design middleware architecture as it stems from solutions proposed in related work, several challenges need to be addressed. This study does not solve any specific management issues; rather, it is aimed at identifying and discussing the key components of general middleware architecture for ASNs. Middleware architecture is proposed in this study to create an ASN for mobile devices. The architecture provides functions to create and manage ASN members, exchange users' profiles and provide interaction among ASN users. The proposed architecture includes overlay modules for sharing information and managing the ASN, thereby providing a complete platform-independent solution for application developers to create ASNs and other applications.

## RESULTS AND DISCUSSION

The proposed software architecture for ASNs comprises four layers:

- Application layer
- Transport layer
- Ad-hoc social layer
- Ad-hoc communication layer

The ad-hoc communication layer addresses issues related to the management of the network among nodes. This layer also provides basic communication services among interested nodes. The ad-hoc social layer manages user profiles and helps to create a social network on top of the ad-hoc network. The application layer helps users use the ASN by accepting social connections and also to configure the parameters of other layers or modules. This section explains the proposed architecture using an example that includes users  $u_1, u_2, \dots, u_n$ .

**Application layer:** The application layer provides an interface for managing the user profile and searches other users with similar interests. This allows the user to create an ASN for sharing text, images, video, chatting and locating similar users using GPS. The user interface also includes settings to configure the parameters of the ad-hoc social layer and ad-hoc communication layer. User  $u_1$  may use the interface to manage their profile, e.g., the algorithm automatically creates and sets the profile based on location; however, user  $u_1$  may use  $P_1$  to override the default option of the local profile and select a global profile, or add keywords into their profile to receive connection requests from better similar users. User  $u_1$  may also use the interface as displayed in Fig. 1 by clicking a button labeled "Click for ASN" to initiate the ASN to search users and then as an interface to chat or connect with other similar users (Fig. 1). The profiles of similar users are received by data packets via communication path  $R_9$ .

The application layer also allows the user to set the Time-To-Live (TTL) field by processing step  $P_2$ . The TTL is used by lower layers to control visibility of profile w.r.t. number of hops. The network will not distribute the profiles of users at distances more than a "hop" as defined by the TTL. For example, if a user  $u_1$  is not interested in setting up an ASN with users more than three hops away, then the user may use  $P_2$  to set the TTL for three hops and the protocol will ensure that a user receives suggestions of similar users only from within a 3-hop radius. The user may also configure similarity as medium, high, or very high in the settings tab.

**Transport layer:** The transport layer provides end-to-end communication for the ASN and ensures maximum throughput per connection. The transport layer protocol

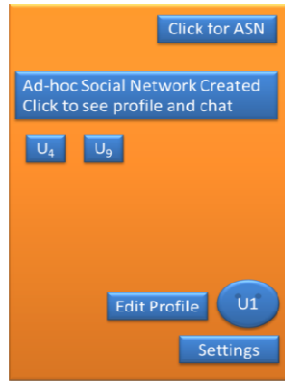


Fig. 1: User interface for ad-hoc social network

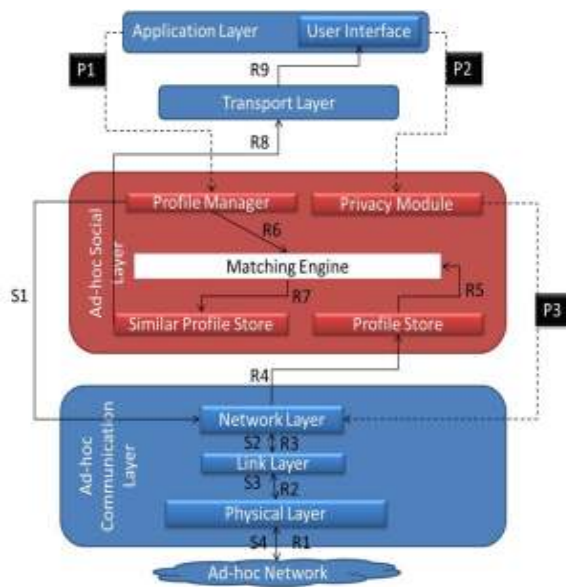


Fig. 2: Middleware architecture displaying end-to-end communication in an ad-hoc social network, wherein  $R_n$  and  $S_n$  represent the flow of receiving and sending profile data packets from one node to another, respectively.  $P_n$  represents processing steps, where  $n$  is a step number

has a mechanism to control congestion and flow in the network. The protocol also adapts dynamically to changing topology and available bandwidth. The transport layer receives similar profiles as mentioned below.

**Ad-hoc social layer:** The ad-hoc social layer is an important layer from a research perspective since not much work has been proposed in terms of components or services that should be provided by this layer. This study proposes an ad-hoc social layer that includes the following components:

**Profile manager:** The profile manager stores and manages the user profile, which covers a variety of details, such as: name, date and/or year of birth, gender,

education, employment, friends, internet search log, browsing history (Lee and Hong, 2011), GPS of locations visited, key attributes of profiles of persons with previous chat history in the ASN, etc. The profile can be a static profile based on user information, a local profile based on current GPS location, or a global profile, which is a combination of all profiles depending on the configuration provided by  $u_1$  (Aneja and Gambhir, 2014). Profiles of other users with whom  $u_1$  has a chat history are also maintained in the profile as it indicates that  $u_1$  is interested in contacting persons of these profiles. The profile may be automatically configured based on GPS location as proposed in Aneja and Gambhir (2014, 2015). The profile manager also sends an appropriate profile of the user to the matching engine in step  $R6$  and initiates sending the profile in step  $S1$  (Fig. 2).

**Profile store:** The profile store stores the profiles of all users received by  $R4$  and currently part of the user's ASN. The received profiles are transferred to matching components. The profile store is updated dynamically based on the information received from neighbors as new members join or leave the ad-hoc network. Therefore, a user  $u_1$  may have profiles of a subset of users from the set  $u_2, u_3, \dots, u_n$  in case some decide to create an ASN.

**Matching engine:** The matching engine matches the user's profile with profiles received from the profile store. Similar profiles that are similar with more than the threshold limit are transferred to "Similar Profile Store" as displayed by  $R7$  in Fig. 2. The matching process can be syntax-based or semantics-based (Raad *et al.*, 2010). Syntax-based similarity lexicographically matches two values; that is, it computes the distance between two values by approximate lexicographical matching. Semantics-based similarity measures the similarity of two values even though the values are lexicographically different. The matching engine will match the profiles of  $u_1$  with all stored profiles based on the similarity options configured by  $u_1$ . For simplicity, let us assume that the matching engine matches profiles of  $u_1$  with  $u_4$  and  $u_9$  while rejecting other profiles.

**Similar profile store:** The similar profile store maintains those profiles that meet a minimum threshold of similarity from all received profiles. The user is prompted to create an ASN with users of stored profiles as displayed in step  $R8$  in Fig. 2. Therefore, the similar profile store will store profiles and contact details of  $u_4$  and  $u_9$  due to the fact that their profiles corresponded to the user's matching parameters. Let us assume that  $u_1$  and  $u_9$  participated in a discussion through the contact option. The profile of  $u_9$  will be stored to indicate that  $u_1$  is most interested in contacting similar users of profile  $u_9$ .

**Privacy module:** The privacy module allows users to control settings (*P2* in Fig. 2) about who can receive their profile in terms of hops, contact them via the ad-hoc network and then add them to their ASN. This module controls the user's visibility in terms of hops. Time-To-Live (TTL) is proposed as the number of hops a packet is permitted to travel before being discarded by a router. Each node will decrease the profile TTL field by one and the packet is discarded when TTL field reaches zero at step *P3* of Fig. 2 (Antoniadis and Courcoubetis, 2006). In the case that users set privacy in terms of hop distance, this module will make sure the TTL field is decreased by one as the profile packet travels by one hop. Additionally, mechanisms to verify and authenticate users and can also be incorporated in this component (Akbari *et al.*, 2012).

**Ad-hoc Communication Layer:** The ad-hoc communication layer is proposed to be flexible in determining the optimal data link, routing and application protocols. This layer comprises a network layer, link layer and physical layer. The ad-hoc communication layer helps connect devices to flow messages in the network without relying on a base station. Each node of an ASN participates in routing data packets of other nodes. The ad-hoc communication layer includes those components required to establish an ASN.

**Network layer:** The network layer comprises routing algorithms used to transmit packets in the ad-hoc network (Perkins and Royer, 1997; Perkins and Bhagwat, 1994). A routing algorithm can be automatically selected depending on whether the ad-hoc network is dense or sparse. For example, flooding may be used if the network is sparse. The network is categorized as sparse or dense based on link density (edge density), which is defined as the ratio of existing links (*l*) to the total number of possible links, e.g., for a network of *N* nodes, the network link density is:

$$D = \frac{m}{0.5 * N * (N - 1)}$$

The maximum link density *D* of a completely connected network is 1. This component is actually responsible to hold the network depending on the configured protocol.

**Data link layer:** The data link layer handles data link connections between nodes. The data link component maintains lists of neighboring nodes and handles the connections. It helps transmit and receive routing data in accordance to the routing table. Data are sent directly to destination if devices are at a distance of one hop; otherwise, to an intermediate device if source and destination are at more than one-hop distance. It may also broadcast packets in accordance with the routing

protocol, routing information, or topology. This component manages communication between nodes  $u_1, u_2, u_3, \dots, u_n$ . The communication includes sending hello packets and profile vectors, managing the network and controlling errors in the packets.

**Physical layer:** The physical layer includes a transmitter and receiver to transfer and receive data packets. Consideration is required with respect to signal reception, interference and noise and preamble length.

**End-to-end communication:** Figure 2 displays all components in different layers as mentioned above. The communication components marked as *Rn*, *Sn* and *Pn* represent, respectively, receiving, sending and processing steps, where *n* represents the step number. A user or node receives packets from the network as it proceeds through steps *R1, R2, R3, R4, R5* along *R6, R7, R8* and *R9* from receiving profiles at physical layer to display similar profiles to facilitate communication at the user interface. Similarly, a user sends packets or profiles via steps *S1, S2, S3* and *S4*. Processing step *P1* provides user flexibility to choose automatically their local or global profile. *P2* configures the TTL value, which sets the number of hops and *P3* sets this TTL value in the transport layer.

## CONCLUSION

This study proposes middleware architecture to create an ad-hoc social network. The middleware architecture has been designed to incorporate those services that are not included in mobile operating systems. Future work will focus on developing middleware architecture for an Android Operating System for mobile devices.

## REFERENCES

- Akbari, R., T. Korkmaz and G. Raju, 2012. Mobile Ad-hoc Networks Security. In: Qian, Z., L. Cao, W. Su, T. Wang and H. Yang (Eds.), Recent Advances in Computer Science and Information Engineering. Lecture Notes in Electrical Engineering, Springer, Berlin, Heidelberg, 127: 659-666.
- Al-oqily, I., M. Al-Shamaileh and S. Oqeili, 2013. An architecture for social ad-hoc networks. Proceeding of the IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT, 2013), pp: 1-5.
- Aneja, N. and S. Gambhir, 2012. Various issues in ad-hoc social networks. Proceeding of National Conference on Recent Trends in Computer Science and Information Technology (RTCSIT), pp: 6-9.
- Aneja, N. and S. Gambhir, 2014. Geo-social profile matching algorithm for dynamic interests in ad-hoc social network. Soc. Networking, 3(5): 240-247.

- Aneja, N. and S. Gambhir, 2015. Geo-social semantic profile matching algorithm for dynamic interests in ad-hoc social network. Proceeding of the IEEE International Conference on Computational Intelligence and Communication Technology (CICT, 2015), pp: 354-358.
- Antoniadis, P. and C. Courcoubetis, 2006. The case of multi-hop peer-to-peer implementation of mobile social applications. Proceeding of the International Conference on Systems and Networks Communications (ICSNC '06), pp: 5-5.
- Bottazzi, D., R. Montanari and A. Toninelli, 2007. Context-aware middleware for anytime, anywhere social networks. *IEEE Intell. Syst.*, 22(5): 23-32.
- Dodson, B., A. Cannon, T.Y. Huang and M.S. Lam, 2011. The Junction Protocol for Ad Hoc Peer-to-peer Mobile Applications. Retrieved from: <https://mobisocial.stanford.edu/papers/junction.pdf>.
- Gambhir, S. and N. Aneja, 2013. Ad-hoc social network: A comprehensive survey. *Int. J. Sci. Eng. Res.*, 4(8): 156-160.
- Hoebeke, J., I. Moerman, B. Dhoedt and P. Demeester, 2004. An overview of mobile ad hoc networks: Applications and challenges. *J. Commun. Netw-UK*, 3(3): 60-66.
- Lee, J. and C.S. Hong, 2011. A mechanism for building ad-hoc social network based on user's interest. Proceeding of the 13th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp: 1-4.
- Liaqat, H.B., F. Xia, J. Ma, L.T. Yang, A.M. Ahmed and N.Y. Asabere, 2015. Social-similarity-aware TCP with collision avoidance in ad hoc social networks. *IEEE Syst. J.*, 9(4): 1273-1284.
- Liaqat, H B., F. Xia, Q. Yang, L. Liu, Z. Chen and T. Qiu, 2014. Poster: Reliable TCP for popular data in socially-aware ad-hoc networks. Proceeding of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '14), pp: 429-430.
- Mangla, S. and S. Gambhir, 2015. Research challenges in Ad-hoc Social Network (ASN). *Int. J. Adv. Res. Sci. Eng.*, 4(1): 124-128.
- Mckinsey, 2014. Offline and Falling Behind: Barriers to Internet Adoption. Retrieved form: <http://www.mckinsey.com/industries/high-tech/our-insights/offline-and-falling-behind-barriers-to-internet-adoption>.
- Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. Proceeding of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94), pp: 234-244.
- Perkins, C.E. and E.M. Royer, 1997. Ad-hoc on-demand distance vector routing. Proceeding of 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp: 90-100.
- Pietiläinen, A.K., E. Oliver, J. LeBrun, G. Varghese and C. Diot, 2009. MobiClique: Middleware for mobile social networking. Proceeding of the 2nd ACM Workshop on Online Social Networks (WOSN '09), pp: 49-54.
- Podobnik, V. and I. Lovrek, 2011. An Agent-based Platform for Ad-hoc Social Networking. In: O'Shea, J. *et al.* (Eds.), *Agent and Multi-agent Systems: Technologies and Applications*. Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 6682: 74-83.
- Raad, E., R. Chbeir and A. Dipanda, 2010. User profile matching in social networks. Proceeding of 13th International Conference on Network Based Information Systems (NBIS), pp: 297-304.
- Rana, J., K. Synnes and J. Kristiansson, 2015. Ad-hoc social network formation for tailored collaboration in shared spaces. *Int. J. Soc. Comput. Cyber-Phys. Syst.*, pp: 33.
- Trieu, Q.L. and T.V. Pham, 2012. STARS: Ad-hoc Peer-to-peer Online Social Network. In: Nguyen, N.T. *et al.* (Eds.), *Computational Collective Intelligence. Technologies and Applications*. Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 7653: 385-394.
- Xia, F., A.M. Ahmed, L.T. Yang, J. Ma and J.J.P.C. Rodrigues, 2014. Exploiting social relationship to enable efficient replica allocation in ad-hoc social networks. *IEEE T. Parall. Distr.*, 25(12): 3167-3176.
- Zhang, D., D. Zhang, H. Xiong, C.H. Hsu and A.V. Vasilakos, 2014. BASA: Building mobile ad-hoc social networks on top of android. *IEEE Network*, 28(1): 4-9.