Peer Management for iTrust over Wi-Fi Direct

Isaí Michel Lombera, L. E. Moser, P. M. Melliar-Smith, Yung-Ting Chuang
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106 USA
imichel@ece.ucsb.edu,moser@ece.ucsb.edu,pmms@ece.ucsb.edu,ytchuang@ece.ucsb.edu

***Abstract*—This paper describes peer management for the iTrust peer-to-peer information publication, search and retrieval system using Wi-Fi Direct in a mobile ad-hoc network. We present the fundamental concept of the iTrust mobile ad-hoc network, the capabilities and limitations of Wi-Fi Direct on the Android platform, and our novel algorithm for managing peer connections. Specifically, we discuss how to acquire MAC addresses and associate peers with IP addresses in order to discover, connect and transfer data using Wi-Fi Direct on Android mobile devices. An Android device with Wi-Fi Direct hardware capabilities can automatically create and maintain a mobile ad-hoc network with nearby similarly configured mobile devices to enable peers to share information.**

***Keywords*—Android; search; mobile ad-hoc network; peer-to-peer network; Wi-Fi Direct**

I. Introduction

As networked mobile devices continue to enrich our daily lives, the ability to publish, search for, and retrieve information becomes more important. Economic forces have resulted in the fast and efficient centralized search engines of Google, Yahoo!, Bing, etc. that allow networked users to search for information on established and public facing computers on the World Wide Web. Although researchers have begun to explore access to more intimate and private stores of information, there is no clear leader or method to search for information on personal networked devices that do not necessarily have access to the World Wide Web.

Sharing pictures, videos, email messages, and other personal information among people physically close together is not easy or efficient (at least, not without using a centralized server as a *go between*). If two people standing next to each other want to share pictures on their mobile phones, they should *not* have to each first post the picture to Flickr; likewise, if they want to share an email attachment, they should *not* have to forward the original email message through email servers. The picture or file attachment should be wirelessly transmitted directly between the mobile devices in a decentralized peer-to-peer fashion without an intermediate centralized server. Moreover, the effortless single-line search bar made famous by Google to search the centralized Web servers should have an analogous decentralized counterpart for peer-to-peer information search.

To this end, we present iTrust over Wi-Fi Direct. Specifically, in this paper, we discuss our novel peer management techniques that automate the tedious user tasks of discovering, connecting and transmitting data among peers enabled with Wi-Fi Direct. iTrust is a decentralized peer-to-peer publication, search and retrieval system with the aim of disseminating information and preventing censorship; previous implementations of iTrust have focused on HTTP, SMS, and a hybrid HTTP-SMS bridging network architectures [9].

Wi-Fi Direct is a peer-to-peer (P2P) implementation of the traditional Wi-Fi 802.11 client-to-access point scheme that allows devices to create mobile ad-hoc networks [18]. The technology is relatively new to consumer devices and only the Android operating system (version 4.1 and above) [1] supports Wi-Fi Direct; no other mobile platform or device (iOS, BlackBerry, etc.) yet supports P2P functionality with Wi-Fi Direct. Unfortunately, Wi-Fi Direct in Android is still a relatively new technology: reliability is poor, basic functionality is difficult-to-use, and structures for advanced information routing or peer management do not yet exist. In this paper, we use the terms Wi-Fi Direct and Wi-Fi P2P interchangeably, as the latter is the original name of the technology.

In the remainder of this paper, we detail the peer management techniques that we developed for iTrust over Wi-Fi Direct, which enable peers to set up and maintain a mobile ad-hoc network. To provide the context, first we present related work and the fundamental design of the iTrust publication, search and retrieval system. Then, we briefly outline the iTrust over Wi-Fi Direct Android implementation. Next, we present the simple and effective peer management technology, which enables iTrust over Wi-Fi Direct peers to discover and connect to other peers automatically, and to maintain a mobile ad-hoc network in which to publish, search for, and retrieve information. Finally, we present conclusions and future work.

II. Related Work

Other researchers [10], [14], [17] have provided comparisons of distributed search methods for peer-to-peer networks. The structured approach requires the nodes to be organized in an overlay network based on distributed hash tables (DHTs), trees, rings, etc. The unstructured approach uses randomization, and requires the nodes to find each other by exchanging messages. The iTrust system uses the unstructured approach.

Motta and Pasquale [11] recognized the opportunity that Wi-Fi Direct presents, even before it became available on Android. They describe a JXTA middleware architecture for peer-to-peer networks, which exploits the features of mobile devices and optimizes the use of mobile resources. They apply the JXTA middleware to a search infrastructure for structured peer-to-peer networks that uses resource indexing based on a DHT. The iTrust over Wi-Fi Direct system for publication, search and retrieval uses an unstructured approach, which is more appropriate for mobile ad-hoc networks.

Like the iTrust project, the Commotion Wireless project [13] aims to ensure that communication cannot be controlled or cut off by authoritarian regimes. Their device-

as-an-infrastructure distributed communication platform integrates Wi-Fi enabled mobile phones, computers and other personal devices to create a metro-scale communication network that supports local peer-to-peer communication and local-to-Internet communication.

Thomas and Robble [15] have created a mobile ad-hoc network for disaster and emergency relief, using the Wi-Fi chips in Android smartphones, allowing them to connect without using cellular networks. Their Smart Phone Ad-Hoc Networks (SPAN) project reconfigures the onboard Wi-Fi chip of a smartphone to act as a Wi-Fi router to nearby similarly configured smartphones. SPAN intercepts communications at a Global Handset Proxy so that typical applications, such as email, Twitter, etc., still work. In contrast, iTrust for mobile ad-hoc networks uses Wi-Fi Direct, which Android now supports.

The Serval project [3] is developing a wireless ad-hoc mobile phone platform, named Serval BatPhone. The project targets rural and remote populations, disaster and emergency relief, and governments that disable the Internet or the cellular network. The team chose to use Wi-Fi ad-hoc mode in the ISM2400 band and Android mobile phones. At the time, Android phones did not support Wi-Fi ad-hoc mode, so they had to manipulate the Wi-Fi hardware on the Android phones. Our implementation of iTrust for mobile ad-hoc networks uses Wi-Fi Direct, which Android now supports.

Meroni et al. [8] describe an opportunistic platform for Android-based devices using Wi-Fi in a mobile ad-hoc network. The opportunistic platform they propose is intended to address concerns of scalability, flexibility and bandwidth in cellular networks by supporting local peer-to-peer communication between nodes. Their platform enables nodes to query for information and receive responses locally and, thus, to save network bandwidth, if the information is large.

The Distributed Mobile Search Service [6] broadcasts query results locally and forwards them over several hops. It is based on a distributed index that comprises, on each mobile device, a local index cache, containing keywords and corresponding document identifiers for received query results. The iTrust system likewise maintains a distributed index, with metadata and the corresponding node addresses and resource ids stored on the iTrust nodes. However, iTrust distributes metadata and the corresponding node addresses and resource ids first, rather than on receipt of the query results and, thus, has a lower message cost.

The Mobile Agent Peer-To-Peer (MAP2P) system [5] supports mobile devices in a Gnutella [4] file-sharing network using mobile agents. A mobile agent attaches itself to the peer-to-peer network, and acts as a proxy for the mobile device. The iTrust system has a lower message cost than Gnutella and, thus, a lower message cost than the MAP2P system.

The 7DS system [12] supports information sharing among peers in a mobile ad-hoc network. It uses a multi-hop flooding algorithm together with multicasting of queries. In contrast, the iTrust system forwards messages selectively to nodes based on a relay probability that limits the number of nodes to which the metadata and the requests are distributed to about $2\sqrt{n}$ nodes, where n is the number of nodes in the membership [7].

Tiago et al. [16] describe a system for mobile search in social networks based on the Drupal content site management system. Their system is fully distributed, is based on the network of social links formed from the mobile phone's address book, and exploits the independence of nodes. The iTrust over Wi-Fi Direct system is not based on the links provided by the mobile phone's address book but, rather, on the nodes within a node's neighborhood.

Yang et al. [19] describe a search mechanism for unstructured peer-to-peer networks, based on special interest groups formed by nodes with similar interests. The iTrust over Wi-Fi Direct system likewise allows users interested in a particular topic or cause to form a social network, so that they can share information among themselves without fear of censorship.

III. iTrust Fundamental Design

The iTrust over Wi-Fi Direct network consists of peers that form a mobile ad-hoc network. Multiple iTrust over Wi-Fi Direct networks may exist simultaneously, and a peer may join any such network(s) over time. Peers in the same network are said to be in the same *membership*, although they need not *all* necessarily be within range of each other. Figure 1 illustrates how information is published, searched for, and retrieved, in the iTrust network.

Any peer with information to share (which we call a *source* peer) generates metadata describing that information and distributes that metadata to a subset of the membership chosen at random (1). A peer interested in querying or *requesting* information distributes a query to a subset of the membership chosen at random (2). In both the distribution of the metadata and the query, a peer that receives the message may *relay* the message to yet another subset of the membership chosen at random. Prevention of message flooding is incorporated into iTrust but is outside the scope of this paper.

When a peer finds a match between the metadata and the query, we say that an *encounter* or a *match* occurs (3). The peer with the match sends a message to the requesting peer which identifies the source peer holding the desired information (4). The requesting peer then directly fetches the information from the source peer (5).

In prior work [9], we presented iTrust over HTTP and iTrust over SMS. iTrust over HTTP operates over the Internet, and iTrust over SMS operates over the cellular network. iTrust over Wi-Fi Direct is different from both of those iTrust implementations in that it operates over mobile ad-hoc networks and it does not require any infrastructure. We have also evaluated the performance of iTrust, and have established that the probability of a match is high even if the metadata and the queries are distributed to relatively few peers [7]. Moreover, iTrust prevents malicious peers from censoring or subverting the free spread of information [2].

IV. iTrust over Wi-Fi Direct Implementation

The iTrust over Wi-Fi Direct system, shown in Figure 2, is implemented on the Android operating system as three separate parts: the user/app interface, the iTrust over Wi-Fi Direct part, and the underlying Android/Linux platform. The app part (in *yellow*) is where the typical Android *app*, such as the file browser, search app, instant messenger, etc., interfaces with the iTrust over Wi-Fi Direct part for P2P publication, search and retrieval; its use is completely dependent on the needs of the user and is not elaborated here. The iTrust over Wi-Fi Direct part (in various shades of *blue*) is briefly explained below to provide the context in which the peer management

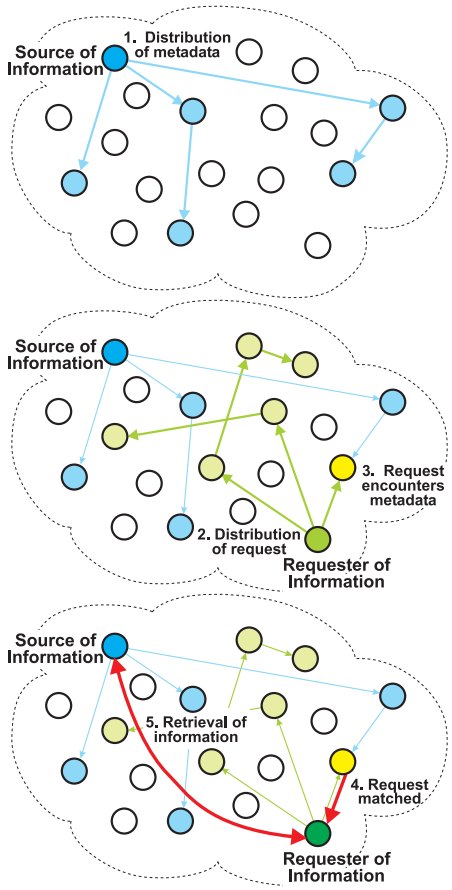

Fig. 1. Information publication, search and retrieval in the iTrust over Wi-Fi Direct network.

algorithm works on the Android operating system. The third part (in *orange*), which relates to Android and Linux, is also outside the scope of this paper.

The three leftmost blocks (in *dark blue*) consist of the primary functions of iTrust that exist regardless of the network transport type, namely: the signal parser, the node core, and the database (DB) adapter. These three blocks are similar to those for the previously implemented iTrust over SMS system [9]. The signal parser processes all received messages and determines what the peer must do next; e.g., if a query message is received, it looks for a metadata match and, if appropriate, instructs the node core to send a match message to the requesting peer. The node core handles all pertinent and bookkeeping functions required for iTrust such as: metadata generation and distribution, match comparisons, query management, message formatting, etc. The DB adapter is a standard SQLite database that stores all iTrust configuration and information tables such as: peer information, metadata or keyword associations, resource or information locations, query tracking, etc.

The remaining four iTrust over Wi-Fi Direct blocks (in *lighter blue*) handle all Wi-Fi Direct functionality and also interface with Android/Linux for data input and output. Briefly, the most basic Wi-Fi Direct functionality in Android *must* be handled by a minimum of two separate objects: an Activity object and a Broadcast Receiver object. The Activity object allows the user or programmer to interact with Wi-Fi Direct

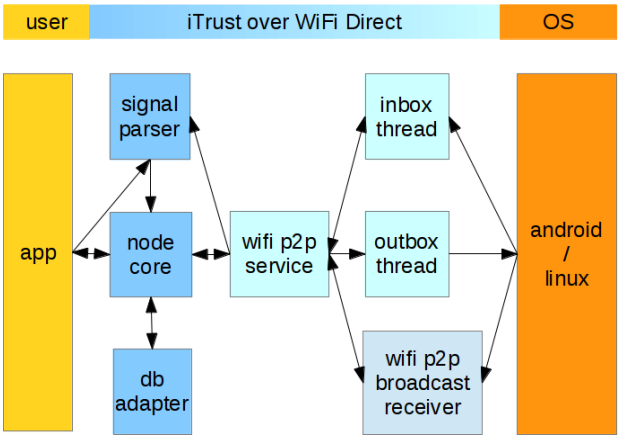

Fig. 2. iTrust over Wi-Fi Direct block diagram.

(the Wi-Fi P2P Service block), whereas the Broadcast Receiver object captures all Android interrupts/callbacks and redirects them to the Activity object for user processing (Wi-Fi P2P Broadcast Receiver).

The Wi-Fi P2P Service block is the main object that controls and handles all Wi-Fi Direct functionality; it is a standard *started* Android Operating System Service. A *started* service in Android parlance is a service that, after being executed once, remains running in the background as long as Android has enough memory, or the service is explicitly killed by the user application (iTrust over Wi-Fi Direct) that started it. This particular block is overloaded with several Android Java Wi-Fi P2P object interfaces to enable it to handle all discovery and connection functions provided by the Wi-Fi P2P Broadcast Receiver callbacks. When a user wishes to send a message, the Wi-Fi P2P Service creates a new Outbox Thread to service the message. When a message is sent to a peer, the Inbox Thread services the message. The Wi-Fi P2P Service handles all important Wi-Fi Direct functions such as: peer management mechanisms, peer discovery protocols, peer connections, and passing incoming and outgoing messages between Android/Linux and the rest of iTrust over Wi-Fi.

The Inbox Thread is created by the Wi-Fi P2P Service, immediately after the Wi-Fi P2P Service finishes instantiation; doing so enables iTrust over Wi-Fi Direct to start listening for incoming messages quickly. This object is not an Android class but, instead, a standard Java thread with a client/server socket; data reading is performed through a normal Java data input stream. Once another peer attempts to connect through the client/server socket, the Inbox Thread reads the data and passes the message along to the Wi-Fi P2P Service. After the current message is serviced, the socket is reset and awaits another incoming message.

The Outbox Thread is created *on demand* to service each outgoing message; after servicing a single message, it dies. Once the Wi-Fi P2P Service receives a message to send, a new Outbox Thread is created and connects to a peer.

The Wi-Fi P2P Broadcast Receiver interfaces directly with the Android/Linux platform. All Android notifications regarding Wi-Fi Direct are captured, and appropriate control is passed along to the Wi-Fi P2P Service object.

V. Peer Management

In this section, we discuss the limitations of Wi-Fi Direct on the Android operating system, our solutions to the peer management problems, and the peer management method. We also present a cursory analysis of the message cost.

A. Limitations of Wi-Fi Direct on Android

Although Wi-Fi Direct is supported on Android operating system (version 4.1 and above), there are serious limitations regarding the peer functionality and data routing techniques.

Android regards Wi-Fi Direct as mostly a Data Link layer function with little support for the Network or Transport layers; namely, there is no direct association between MAC addresses and IP addresses. Regardless of the Wi-Fi Direct specification and whether or not it mandates this association, the lack of a direct MAC address to IP address mapping inhibits many useful network setups.

On activating the Wi-Fi Direct functionality, a peer broadcasts its MAC identifier, its node identifier and its user name, and searches for other peers. When other peers are found, an internal list of MAC addresses is updated. This MAC address list can be queried, and individual peers can be selected and connected to form a P2P network. When a network is created, one peer within the group creates a *soft access point*, establishes itself as the network group owner, and assigns itself an IP address. The IP address of the group owner is transmitted to all peers, and the connection process effectively ends.

However, there are several critical limitations, because only one peer in the network has an effective IP address. This scheme might be adequate for P2P gaming or other simple tasks, such as simple file transfer between two devices; however, it does not scale well to a larger number of peers. Without an effective IP address, three or more peers cannot communicate directly with each other; they are forced to go through the group owner.

Furthermore, there is no reliable way for a peer to determine this information by itself, if it is not the group owner. Android provides no (at least documented) way to query for this information. Standard Java methods do not work either; all Java networking functions return information about the Wi-Fi adapter but *not* the Wi-Fi P2P adapter. The underlying Linux system also provides no help; the Unix Address Resolution Protocol (ARP) table is periodically flushed and not reliable. Additionally, there is no documented way for a node to find its own MAC address for the Wi-Fi P2P adapter.

B. A Method to Manage Peers

To solve this problem, we created the relatively simple method illustrated in Figure 3. Part A illustrates the peer management process; part B illustrates the metadata distribution process; and part C illustrates the query distribution and resource transfer process. Parts B and C are included for completeness of the iTrust over Wi-Fi Direct message discussion, and are described briefly in the next section. Suppose that X and Y are two peers that are in range of each other and are available to connect over Wi-Fi Direct. The following actions occur on each peer independently (each device calls its own API functions).

Once the Wi-Fi P2P Broadcast Receiver object is notified that peer(s) are available, it triggers the Wi-Fi P2P Service object. The Wi-Fi P2P Service object determines the availability

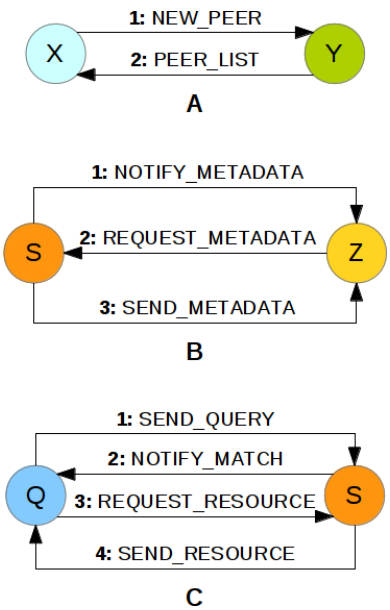

Fig. 3. iTrust message types: (A) Peer management, (B) Metadata distribution and (C) Resource search and retrieval.

of nearby peers and automatically initiates a connection (which is handled automatically by iTrust and, otherwise, would have to be manually initiated by Android). A connection process begins, and the peers negotiate (through callbacks from the peers' Wi-Fi P2P Broadcast Receiver objects to the Wi-Fi P2P Service objects); one peer is randomly chosen as the group owner. At this point, iTrust automatically finalizes the connection without explicit Android confirmation; normally, Android pops up a dialog box on the mobile device and *waits* for the human user to confirm the connection, iTrust connects *automatically*. On finalizing the connection, if a peer determines it is *not* the group peer owner, it sends a *NEW_PEER* message to the group owner.

In Figure 3 part A, X is not the group owner and so it sends a *NEW_PEER* message to Y. The message *NEW_PEER* contains only the MAC address of X; an undocumented but public feature was found in Android and used to extract the peer's Wi-Fi Direct MAC address (effectively a bit-masked value in a parsed Intent object of the Activity/Service class). Recall that only the IP address of the group owner (Y) is known. On accepting the socket data transfer, Y now knows the MAC address and the IP address of X; the MAC address of X is the message payload and, by virtue of creating a socket connection and reading the socket object information, the IP address of X can be determined. The group owner saves this MAC/IP address association in its database, generates a JSON list of all saved MAC/IP address associations, and distributes the JSON list in the *PEER_LIST* message to X. The only data in the *PEER_LIST* message is the JSON list of MAC/IP address associations; peer X decodes the JSON list and saves the associations. Both peers now have *their* up-to-date and identical memberships.

This process, the reporting of MAC addresses and receiving of MAC/IP addresses, can continue for any number of peers in the network (assuming they are within range). If the connection is severed, iTrust automatically reconnects and rebuilds the MAC/IP address pairs.

We tested these peer management mechanisms on two or three physical Nexus 7 tablets with a test harness of the iTrust over Wi-Fi Direct system. Data were transferred easily and automatically, and no user interaction was required apart from making sure that the devices were in range. Furthermore, when the connection was severed due to range or noise, reconnection was automatically established when the problem was corrected. Currently, Wi-Fi Direct does not work on the Android emulator, so additional tests would require more physical devices.

C. Metadata and Query Distribution Messages

Parts B and C are similar to the message protocol used in the iTrust over SMS network [9]. Note that the original iTrust over SMS network required only seven messages, whereas the iTrust over Wi-Fi Direct network requires nine messages (including the two additional peer management messages).

In part B, metadata are distributed as follows: the source peer *S* sends the *NOTIFY_METADATA* message (1), later peer *Z* asks for the metadata using the *REQUEST_METADATA* message (2), immediately after (2) *S* sends the metadata using the *SEND_METADATA* message to *Z* (3).

In part C, a query is distributed and a resource is transferred as follows: the requesting peer *Q* sends the *SEND_QUERY* message (1), the metadata are encountered on *S*, which sends the *NOTIFY_MATCH* message to *Q* (2), later *Q* sends the *REQUEST_RESOURCE* message to *S* (3), immediately after (3) *S* sends the resource using the *SEND_RESOURCE* message to *Q* (4).

D. Message Cost

Given a network of n peers, the number of messages required to synchronize all peer lists is $n + (n-1) + \ldots + 1 = \frac{n(n+1)}{2}$, for $n \geq 2$. This calculation assumes that a new *PEER_LIST* message is sent immediately after every *NEW_PEER* message but before another peer joins (as the process repeats). Immediately sending the *PEER_LIST* message after a *NEW_PEER* message is feasible in small networks; however, it quickly becomes impractical for large networks. In the case of large networks, a simple delay before sending the *PEER_LIST* message allows more time for other peers to *report in*; this delay can be dynamically set by iTrust depending on the membership size.

VI. Conclusion

We have described peer management for iTrust over Wi-Fi Direct on the Android platform that enables peers to construct a mobile ad-hoc network for decentralized publication, search and retrieval. The peer management algorithm for iTrust over Wi-Fi Direct automatically discovers and connects to available peers. We have briefly described the iTrust over Wi-Fi Direct system, discussed limitations of the Android platform with Wi-Fi Direct, and presented our peer management solution to address the current limitations. The iTrust over Wi-Fi Direct peer management algorithm facilitates the creation of mobile ad-hoc networks using mobile devices, and is a step towards making decentralized personal search feasible and more convenient.

In the future, we plan to experiment with the iTrust over Wi-Fi Direct implementation using more devices and to explore the practical costs of the peer management algorithm. In addition, we plan to add an easy-to-use graphical user interface and to release an application that is useable even by computer novices.

Acknowledgment

This research was supported in part by U.S. National Science Foundation grant number NSF CNS 10-16193.

References

[1] Android Developers, Android 4.0 Platform Highlights, http://developer.android.com/sdk/android-4.0-highlights.html

[2] Y. T. Chuang, P. M. Melliar-Smith, L. E. Moser and I. Michel Lombera, "Protecting the iTrust information retrieval network against malicious attacks," *Journal of Computing Science and Engineering* 6(3), 2012, pp. 179–192.

[3] P. Gardner-Stephen, "The Serval project: Practical wireless ad-hoc mobile telecommunications," http://developer.servalproject.org/site/docs/2011/Serval_Introduction.html

[4] Gnutella, http://gnutella.wego.com/

[5] H. Hu, B. Thai and A. Seneviratne, "Supporting mobile devices in the Gnutella file sharing network with mobile agents," *Proceedings of the 8th IEEE Symposium on Computers and Communications*, Kemer-Antalya, Turkey, July 2003, pp. 1035–1040.

[6] C. Lindemann and O. P. Waldhorst, "A distributed search service for peer-to-peer file sharing in mobile applications," *Proceedings of the Second International Conference on Peer-to-Peer Computing*, Linkoping, Sweden, September 2002, pp. 73–80.

[7] P. M. Melliar-Smith, L. E. Moser, I. Michel Lombera and Y. T. Chuang, "iTrust: Trustworthy information publication, search and retrieval," *Proceedings of the 13th International Conference on Distributed Computing and Networking*, Hong Kong, China, January 2012, LNCS 7129, Springer, pp. 351–366.

[8] P. Meroni, E. Pagani, G. P. Rossi and L. Valerio, "An opportunistic platform for Android-based mobile devices," *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, Pisa, Italy, February 2010, pp. 191–193.

[9] I. Michel Lombera, L. E. Moser, P. M. Melliar-Smith and Y. T. Chuang, "Mobile decentralized search and retrieval using SMS and HTTP," *ACM Mobile Networks and Applications Journal* 18(1), 2013, pp. 22–41.

[10] J. Mischke and B. Stiller, "A methodology for the design of distributed search in P2P middleware," *IEEE Network* 18(1), 2004, pp. 30–37.

[11] R. Motta and J. Pasquale, "Wireless P2P: Problem or opportunity," *Proceedings of the Second IARIA Conference on Advances in P2P Systems*, Florence, Italy, October 2010, pp. 32–37.

[12] M. Papadopouli and H. Schulzrinne, "Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices," *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing*, Long Beach, CA, 2001, pp. 117–127.

[13] P2P Foundation, Commotion, http://p2pfoundation.net/Commotion

[14] J. Risson and T. Moors, "Survey of research towards robust peer-to-peer networks: Search methods," Technical Report UNSW-EE-P2P-1-1, University of New South Wales, September 2007, RFC 4981, http://tools.ietf.org/html/rfc4981

[15] J. Thomas and J. Robble, "Off grid communication with Android: Meshing the mobile world," http://www.mitre.org/work/tech_papers/2012/12_2943/12_2943.pdf

[16] P. Tiago, N. Kotiainen, M. Vapa, H. Kokkinen and J. K. Nurminen, "Mobile search – Social network search using mobile devices," *Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, January 2008, pp. 1201–1205.

[17] D. Tsoumakos and N. Roussopoulos, "A comparison of peer-to-peer search methods," *Proceedings of the Sixth International Workshop on the Web and Databases*, San Diego, CA, June 2003, pp. 61–66.

[18] Wi-Fi Alliance, Wi-Fi Direct, http://www.wi-fi.org/discover-and-learn/wi-fi-direct%E2%84%A2

[19] J. Yang, Y. Zhong and S. Zhang, "An efficient interest-group-based search mechanism in unstructured peer-to-peer networks," *Proceedings of the International Conference on Computer Networks and Mobile Computing*, Shanghai, China, October 2003, pp 247–252.