Decentralized Search and Retrieval for Mobile Networks using SMS

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 the iTrust over SMS decentralized search and retrieval system for mobile networks. Any mobile device in the iTrust network can communicate with any other mobile device in the iTrust network to distribute, search for, and retrieve information. Third-party developers can use the iTrust over SMS API on the Android platform to add this search and retrieval functionality to existing applications quickly and easily. Developers of applications for other mobile device platforms can use the iTrust over SMS protocol to create compatible applications that can communicate using iTrust over SMS. In addition to the iTrust over SMS components and protocol, this paper presents a performance evaluation of the iTrust over SMS system, which shows that the probability of information retrieval is high, even if some of the mobile devices are not available. It also shows that the average search latency is consistently less if all of the participating nodes use the same mobile service provider, and is consistently more if the nodes use different mobile service providers.**

***Keywords*-search; retrieval; SMS; mobile network; peer-to-peer network; social network**

I. Introduction

Many personal social interactions are mediated by mobile devices, which have transformed the human social experience. Anyone with access to a mobile phone, laptop, tablet, or other networked device can communicate with a friend or acquaintance simply and almost instantaneously. We envision a future where any mobile device can distribute, search for, and retrieve information in a decentralized peer-to-peer fashion from another such mobile device.

To that end, we developed the iTrust over SMS system described in this paper. iTrust over SMS is a decentralized search and retrieval system that enables any two mobile devices to share information using the Short Message Service (SMS) that is available on many mobile phones. Modeled on our iTrust over HTTP system [4], [11], [12] for search and retrieval over the Internet, iTrust over SMS brings information sharing to any mobile device with instant text messaging capability. The iTrust over SMS system retains features of the iTrust over HTTP system that protect information against censorship, filtering, and subversion of information.

SMS works on low-end mobile phones, as well as smart phones, and is available worldwide. Global SMS traffic is expected to reach 8.7 trillion messages by 2015, up from 5 trillion messages in 2010 [16]. To quote Giselle Tsirulnik, senior editor at Mobile Commerce Daily, "SMS is cheap, it is reliable, it is universal, and it has unrivaled utility as a bearer for communications, information and services." In developing countries, SMS is the most ubiquitous service for information exchange after human voice.

In this paper, we describe the iTrust over SMS decentralized peer-to-peer search and retrieval system, that enables mobile users to share information over SMS. First, we discuss the fundamental concept of the iTrust system. Next, we describe the iTrust over SMS components as implemented on the Android platform. Then, we briefly describe the iTrust over SMS protocol. After presenting the iTrust over SMS components and protocol, we provide a performance evaluation of iTrust over SMS using several important performance metrics. Finally, we present related work, conclusions, and future work.

II. Fundamental Concept of the iTrust System

The iTrust search and retrieval system involves no centralized mechanisms and no centralized control. We refer to the nodes that participate in an iTrust network as the *participating nodes* or the *membership*. Multiple iTrust networks may exist at any point in time, and a node may participate in several different iTrust networks at the same point in time. The design of iTrust is illustrated in Figure 1, and is explained below with reference to the figure.

Some nodes (the *source nodes*) produce information, and make that information available to other participating nodes. The source nodes produce metadata that describes their information, and distribute that metadata to a subset of the participating nodes chosen at random (1). The metadata includes a list of keywords for the information and the source address (mobile phone number) and resource identifier (id) of the information.

Other nodes (the *requesting nodes*) request and retrieve information. The requesting nodes generate query requests that refer to the metadata, and distribute the requests to a subset of the participating nodes chosen at random (2). The metadata and the requests may have an expiration date / time.

The participating nodes compare the metadata in the requests they receive with the metadata they hold. If a node finds a match (which might involve synonyms and only some of the keywords), which we refer to as an *encounter* (3), the

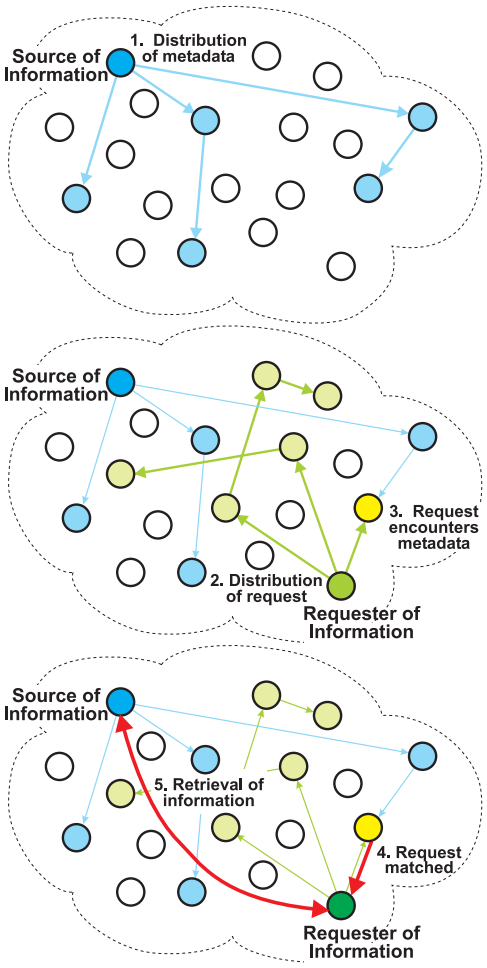

Figure 1. Search and retrieval in the iTrust network.

matching node returns the source address and resource id of the information to the requesting node (4). The requesting node then uses the source address and resource id to retrieve the information from the source node (5).

Distribution of the metadata and the requests to relatively few nodes suffices to achieve a high probability of a match. Moreover, even if some of the randomly chosen nodes are not available, the probability of a match is high, as shown in Section V. Furthermore, it is not easy for a small group of nodes to subvert the iTrust system to control which information is delivered and which is suppressed.

III. The iTrust over SMS System

The iTrust over SMS system requires no cooperation or installation on the part of the mobile network service providers; it is a completely independent system that resides on top of the existing mobile network. For the purposes of this paper, the mobile network is represented by the Short Message Service Center (SMSC), shown in Figure 2. The SMSC of the service provider is a store-and-forward message center for the sending and receiving of SMS text messages. The service provider manages the SMSC so that individual mobile users do not need to know about the SMSC; a text message is simply sent and received at the destination mobile phone. We extend this concept by saying that an SMS message sent to the SMSC eventually reaches a mobile phone node; likewise, an SMS message received by the SMSC was originated by some mobile phone node. This conceptualization allows mobile phones using the iTrust over SMS API to be *directly* connected to each other in a peer-to-peer fashion, regardless of the service provider or technology used by the physical device.

The user in Figure 2 is a placeholder for the applications that developers write to use the iTrust over SMS API, components, and protocol. Examples of such applications include custom GUIs for human applications that are iTrust specific, automated services that connect to the iTrust network, and multi-protocol messaging applications that use iTrust as yet another way to connect heterogeneous networks (*e.g.*, AOL IM / Yahoo Messenger / MSN / XMPP chat clients).

The iTrust over SMS system is implemented as five main components on the Android platform. These components are designed to be used in conjunction with any suitable graphical user interface or application, and are described without reference to a particular graphical user interface or application. Figure 2 illustrates these five components, as well as external interactions with the user application and the SMSC of the mobile network.

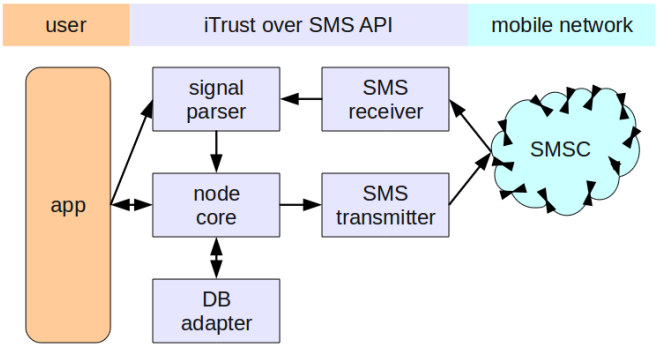

Figure 2. The iTrust over SMS API and components.

Three components make up the bulk of the iTrust over SMS system, and the other two components serve the required tasks of sending and receiving SMS messages. We discuss these components below, in clockwise order in Figure 2, beginning with the SMS receiver.

A. SMS Receiver

The SMS receiver performs the relatively mundane task of servicing incoming SMS messages and passing them on to the signal parser. It uses the SmsManager class found in the Android telephony library to decode single- or multi-part SMS messages before transferring control to the signal

parser. To wait for message servicing, the SMS receiver uses the Android BroadcastReceiver class and registers itself in the Android manifest file as a broadcast receiver Intent with the Intent Filter SMS_RECEIVED (in Android parlance).

On installation, Android routes all received SMS messages to the SMS receiver (like an event handler calling a service routine). Once a message has been passed to the signal parser, the SMS receiver can either *pass through* the message to the default Android application (such as the default IM application) or exit (abandon the message). For debugging purposes, it is useful to pass through the message so that the developer can print the message to the mobile phone, but in day-to-day usage a mobile phone user may elect to abandon the message so as not to receive alerts about all received iTrust over SMS messages.

Because the SMS receiver is registered as a broadcast receiver (a daemon or service), no other iTrust over SMS components are instantiated when an SMS message is received and ready to be parsed. Thus, the SMS receiver must instantiate the other iTrust over SMS components to service the SMS message, wait for the other components to finish processing, and then exit.

B. SMS Transmitter

The SMS transmitter is the simplest of the five components; its sole function is to transmit SMS messages as specified by the node core. The node core provides the SMS transmitter with a text string and mobile phone number, which the SMS transmitter then uses to send an SMS text message. The SMS transmitter is a simple wrapper for the Android SmsManager and SmsMessage classes, and does not require registration as an Android Intent. Whereas SMS transmission is restricted *within* the iTrust over SMS API (only the node core can send messages), any user code *outside* the API may access the Android SmsManager class. This feature allows the iTrust over SMS API to be installed alongside any other IM application on the mobile phone without resource conflicts. It also enables any application that uses the iTrust over SMS API to send SMS messages outside the scope of the iTrust network (*i.e.*, an existing IM application can add iTrust over SMS capabilities to existing services without affecting those services).

C. DB Adapter

The DB adapter controls database access. To insert data into the database or extract data from the database, the node core may use only the publicly available methods of the DB adapter. In addition, the DB adapter handles creation of the database (on installation of the application), and updates to the database (when application updates or iTrust over SMS changes occur).

The database is a standard SQLite3 database provided by Android. Because the iTrust over HTTP system uses SQLite2, the database schema required little modification for the iTrust over SMS system. The only major change was the addition of a resource-to-node primary key to the foreign key linking table to normalize the database schema (mostly a result of moving from iTrust over HTTP's XML metadata format to iTrust over SMS's JSON metadata format).

The DB adapter handles the bookkeeping functions of iTrust over SMS, such as keeping track of nodes, resources, keywords (for describing resources), queries, results (matches), and metadata lists.

D. Node Core

The node core is the most complex of the iTrust over SMS components, and contains the bulk of the code within it. User code in the application primarily interacts with the node core. The primary functions of the node core are: insertion of node / resource / keyword data, encounter matching, query relaying, resource retrieval, database access, and metadata creation and distribution. Data for the local node (mobile phone) is inserted into the database directly by the node core; user code in the application interfaces with public methods in the node core to insert the appropriate information. The user may insert node information (most importantly the node's mobile phone number) or resource information (resource data or text) or keyword text, using the methods provided by the node core. For example, user code may use the Apache Lucene or Tika libraries to generate metadata from resources and then use the node core methods to associate the metadata with the resource.

When a query is received, the query text is compared to the available keywords. If there is an encounter (a match), a response is sent to the querying node in an SMS message and the query is saved for future reference. Whether or not an encounter occurs, the message can be forwarded or relayed to a random selection of other nodes in the iTrust network by simply retransmitting the message. Flooding is avoided using a relaying probability chosen such that the metadata and the requests are distributed to about $2\sqrt{n}$ nodes in a network of n nodes. Moreover, a node never processes or relays metadata or requests that it has previously received (duplicate messages are detected by storing the message parameter *query_id*, which serves as a unique identifier, as described in Section IV) .

To check for encountered data and for query duplication, the node core makes extensive use of the DB adapter. A resource retrieval message triggers the node core to extract the resource data from the DB adapter and package it for transmission by the SMS transmitter to the interested node. If a remote node asks for a resource that does not exist on the local node, the request is ignored and processing of the message is halted.

The node core also handles the sending and receiving of metadata during metadata distribution. To send metadata, the node core on the source node utilizes the Android JSON classes, JSONArray and JSONObject. First, the node core

extracts all local resource row ids from the DB adapter; for each resource, a JSONObject is created. Second, for each resource, all keywords are extracted and converted into a JSONArray; this JSONArray is then attached to the JSONObject describing the resource. Third, the JSONObject resource keyword pairs are strung together into a top-level JSONArray. Note that this final JSONArray pairs the keywords to only the *row id*s of the resource and not the resource data itself; the resource data is not sent with the metadata and, consequently, resources are not replicated across the network.

To receive metadata, the node core utilizes the same Android JSON classes. The process of receiving and parsing the metadata is, for the most part, the reverse order of creating the metadata. First, the top-level JSONArray is parsed for JSONObject resource keyword pairs. Second, each JSONObject resource keyword pair is divided between the JSONObject resource row id and the JSONArray keywords. Third, the resource row id is inserted into the DB adapter, and the JSONArray keywords are transformed back into a string array and then inserted into the DB adapter. The fourth step involves marking the newly inserted resource row id as *non-local*, *i.e.*, the newly inserted keywords are paired to a resource *not* stored on the node receiving the metadata.

As the result of an encounter, the querying node receives a response that includes the source node address and source node resource row id. In the case of a local resource, the source node address is that of the responding node, and the resource row id is from the DB adapter resource row id on that responding node. In the case of a non-local resource, the source node address is the node address of the node that originally distributed the metadata, and the resource row id is *also* from that node.

E. Signal Parser

The signal parser handles incoming SMS messages and appropriately responds according to the iTrust over SMS protocol. Because of the close relationship between the signal parser and the protocol, only a high-level description of the signal parser is required here. For now, it suffices to say that the signal parser can be conceptualized as the "brain" of the iTrust over SMS system. It reads incoming messages, parses the messages, and triggers the appropriate responses in the node core. For this reason, the user application has some control over the signal parser, but only in triggering the execution of certain functions, which propagate down to the node core. The signal parser cannot send information back to the application; the application must interact with the signal parser by sending signals or triggering functions and reading the state of the node core.

IV. The iTrust over SMS Protocol

The iTrust over SMS protocol allows any SMS-capable device to communicate over the iTrust over SMS network regardless of physical or software platform. We provide below a high-level overview of the iTrust over SMS protocol. First, we briefly describe the iTrust over SMS message types. Then, we present two message flow diagrams, with messages labeled by number and message type, that exemplify the metadata distribution and querying tasks of iTrust over SMS.

A. The iTrust over SMS Message Types

Figure 3 shows the seven types of SMS messages that the iTrust over SMS protocol uses to enable information search and retrieval over the iTrust network. The first four messages are used to send queries, send a notification of an encounter or match, request a specified resource on the source node, and send the resource data. The last three messages are used to start metadata distribution, send metadata updates, request metadata updates, and send JSON formatted metadata. All message types are named intuitively; the parameters are used to tailor individual messages to specific queries or resources.

Message	Identifier	Parameter 1	Parameter 2	Parameter 3
SEND_QUERY	itq	*<caller_number>*	*<query_id>*	*<query>*
NOTIFY_MATCH	itr	*<source_number>*	*<query_id>*	*<resource_id>*
REQUEST_RESOURCE	itq	now	*<query_id>*	*<resource_id>*
SEND_RESOURCE	itr	data	*<query_id>*	*<data>*
NOTIFY_METADATA	itm	*<source_number>*	*<expiry_date>*	
REQUEST_METADATA	itm	pull	*unused*	
SEND_METADATA	itm	push	*<data>*	

Figure 3. The iTrust over SMS message types.

B. An Example of Metadata Distribution

Figure 4 shows the typical flow of messages during metadata distribution in the iTrust over SMS network. Node S notifies node Z of a new metadata update (message 1), Z requests the metadata (message 2), and S sends the metadata (message 3). Each numbered message string is the actual text sent in the SMS message; in this example, the parameters have been filled in to represent realistic data sent for a typical distribution task. Message types and parameters are delineated with the @ symbol (no relation to Twitter).

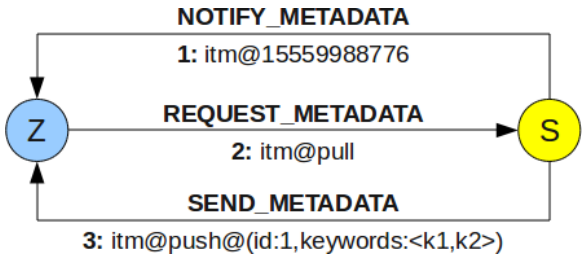

Figure 4. Metadata distribution message flow example.

C. An Example of Search and Retrieval

Figure 5 shows the typical flow of messages during search and retrieval in the iTrust over SMS network. Node Q queries node S (message 1), S sends back a notification that an encounter or match has occurred (message 2), Q requests the resource (message 3), and S sends the resource

to Z (message 4). As in Figure 4, the text strings of the SMS message are shown.

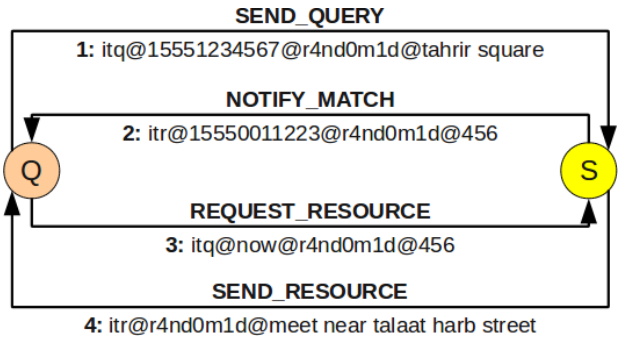

Figure 5. Search and retrieval message flow example.

V. Performance Evaluation

For this performance evaluation, we assume that all of the participating nodes in the iTrust over SMS network have the same membership set. Moreover, we assume that the underlying mobile phone network delivers messages reliably and that the nodes have enough memory to store the source files and the metadata that the nodes generate and receive. Furthermore, we assume that the metadata and the requests are sent directly to the nodes without relaying. As yet, we do not have any data on how much metadata and how many requests might be generated in a real iTrust network.

The primary parameters determining the performance of the iTrust system are:

- n: The number of participating nodes (*i.e.*, the size of the membership set)
- x: The proportion of the n participating nodes that are available
- m: The number of participating nodes to which the metadata are distributed
- r: The number of participating nodes to which the requests are distributed
- k: The number of participating nodes that report matches to a requesting node.

Our performance evaluation of iTrust is based on the hypergeometric distribution [5], which describes the number of successes in a sequence of random draws from a finite population *without* replacement. Thus, in iTrust, the probability of k matches is:

$$P(k) = \frac{\binom{mx}{k}\binom{n-mx}{r-k}}{\binom{n}{r}} \tag{1}$$

for $mx + r \leq n$ and $k \leq \min\{mx, r\}$.

Expanding the binomial coefficients in Eq. (1), we obtain:

$$P(k) = \frac{\left(\frac{mx}{k}\frac{mx-1}{k-1}\cdots\frac{mx-k+1}{1}\right)\left(\frac{n-mx}{r-k}\frac{n-mx-1}{r-k-1}\cdots\frac{n-mx-r+k+1}{1}\right)}{\left(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1}\right)} \tag{2}$$

for $mx + r \leq n$ and $k \leq \min\{mx, r\}$.

In particular, the probability of $k = 0$ matches is:

$$P(0) = \frac{\left(\frac{n-mx}{r}\frac{n-mx-1}{r-1}\cdots\frac{n-mx-r+1}{1}\right)}{\left(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1}\right)} \tag{3}$$

for $mx + r \leq n$.

Consequently, the probability of *one or more* matches is:

$$P(k \geq 1) = 1 - \frac{\left(\frac{n-mx}{r}\frac{n-mx-1}{r-1}\cdots\frac{n-mx-r+1}{1}\right)}{\left(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1}\right)} \tag{4}$$

for $mx + r \leq n$.

A. Analysis of an Average Membership

The membership size used in our analysis must correspond to real-world characteristics. Notably, from [14], we observe that the average Twitter follower size (roughly the same as the iTrust membership size) is slightly more than 100 nodes. Microblogging is a good application for iTrust over SMS, as microblogging is ideally suited to the small text-based messages of SMS. For our analysis, we chose a membership size of $n = 144$ nodes, with $m = r = 24$ metadata / request messages, for reasons that will become clear in Section C.

For an iTrust network with $n = 144$ nodes and $x = 1.0, 0.7, 0.4, 0.2$ available nodes, Figure 6 shows the probabilities of one or more matches, obtained from Equation (4), as the number of nodes to which the metadata and the requests are distributed increases.

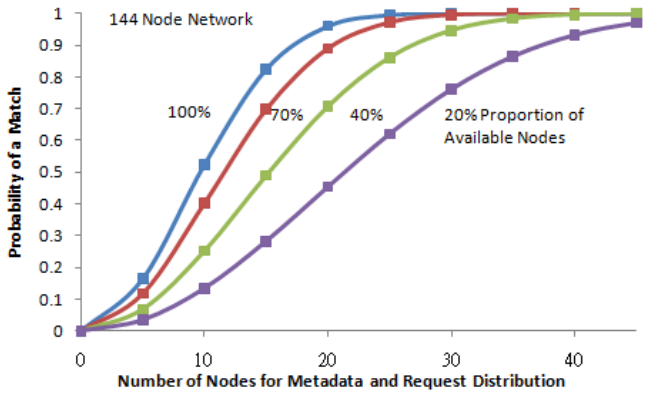

Figure 6. The probabilities $P(k \geq 1)$ of one or more matches as the number $m = r$ of nodes to which the metadata and the requests are distributed increases, for different proportions x of available nodes.

B. Emulation of a Small Membership

The iTrust over SMS system was developed and deployed on a small number of Android mobile phones, and was tested for fitness and robustness on those mobile phones. Unfortunately, it is not economically feasible to purchase enough physical mobile phones and accompanying data / service plans to enable a real-world test of 144 nodes. The next best choice is the deployment of iTrust over SMS on an emulated system of networked physical devices. We ran

multiple instances of the Android operating system on an emulated ARM to x86 environment using UNIX sockets for SMS communication. Multiple instances of the standard Android emulator with Android 2.3 (Gingerbread) images were run up to the limit of allowable ports; specifically, the maximum number of Android emulators is locked at 16 (32 unidirectional ports with each emulator requiring two ports for sending and receiving SMS messages). Furthermore, the standard Android operating system has a built-in SMS sending rate maximum, which limits the number of messages that may be sent in a given time period; those limits were disabled for testing purposes.

The following emulation test was run on an AMD Phenom II 3.4GHz quad core hyper-threaded testbed; each trial (involving 16 nodes) took 16GB of RAM and required 65 seconds to complete. The experiments involved an iTrust network with $n = 16$ nodes, $x = 1.0$ proportion of available nodes, and $m = r = 8$ metadata / request messages.

Figure 7 shows the observed results for 1000 trial runs. Each trial run consists of generating random resource and keyword pairs, relaying the resulting metadata, searching for at least one of the keywords, and finally counting the number of encounters (matches). In the figure, we see that the observed data from the Android application closely follows the analytical data obtained from Equation (2) for the probability $P(k)$ of k matches.

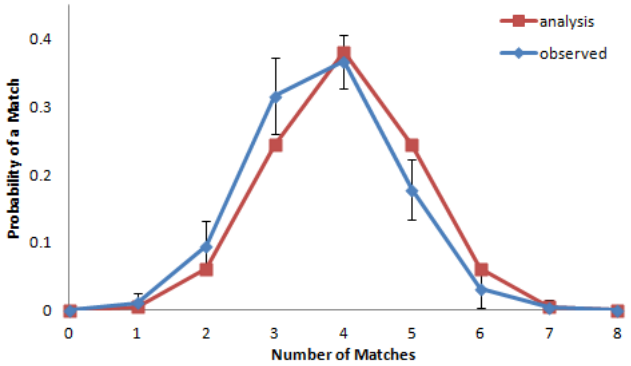

Figure 7. The number k of encounters (matches) vs. the mean probabilities $P_{observed}(k)$ with error bars, for a small Android emulator testbed. The probabilities $P_{analyis}(k)$ are also shown.

Table I lists the analysis and the observed probabilities for the various numbers of matches. For example, for $k = 6$ matches $P_{analysis}(6) = 0.060926$ and $P_{observed}(6) = 0.031020$. The right-most column lists the cumulative observed probabilities for the various numbers of matches. For example, the cumulative observed probability for 1 to 6 matches is $P_{cumulative}(1 \leq k \leq 6) = 0.996000$.

C. The Importance of $2 * \sqrt{n}$

In the above experiments, we used $n = 16$ nodes with $m = r = 8 = 2 * \sqrt{16}$ nodes to which the metadata and the requests are distributed. This choice was deliberate.

Matches	Analysis	Observed	Cumulative
0	0.000155	0.000000	0.000000
1	0.004974	0.011000	0.011000
2	0.060926	0.094143	0.105143
3	0.243705	0.315714	0.420857
4	0.380790	0.366796	0.787653
5	0.243705	0.177327	0.964980
6	0.060926	0.031020	0.996000
7	0.004974	0.004000	1.000000
8	0.000000	0.000000	1.000000

Table I
THE ANALYSIS, OBSERVED, AND CUMULATIVE PROBABILITIES.

In [12], we showed that, if the iTrust membership set contains n participating nodes of which a proportion x are operational, the metadata are delivered to m participating nodes, a request is delivered to r participating nodes, then the probability of one or more matches satisfies:

$$P(k \geq 1) \quad > \quad 1 - e^{-\frac{mrx}{n}} \tag{5}$$

In particular, if $m = r = \lceil 2 * \sqrt{n} \rceil$ and $x = 1.0$, then

$$\begin{aligned} P(k \geq 1) &> 1 - e^{-\frac{\lceil 2*\sqrt{n} \rceil \lceil 2*\sqrt{n} \rceil}{n}} \\ &> 1 - e^{-\frac{2*\sqrt{n}\ 2*\sqrt{n}}{n}} \\ &= 1 - e^{-4} \\ &> 0.9817 \end{aligned} \tag{6}$$

Thus, to obtain a high probability of one or more matches, we choose $m = r = \lceil 2 * \sqrt{n} \rceil$ nodes to which to distribute the metadata and the requests.

D. Average Search Latency

For the mobile phones on which iTrust over SMS was deployed, first we make a basic observation about the *average search latency*, *i.e.*, the duration in time from sending a query to receiving the first match response at the querying node. The average search latency is *twice* the SMS *delivery latency*, *i.e.*, the latency for sending the query and the latency for sending back the response. We ignore the insignificant time required to process an encounter on the matching node (in practice, this time is less than 100 milliseconds on a modern mobile handset). We also ignore any time required to retrieve the resource information because, in practice, the information varies greatly in size (from a short text snippet to a video file that uses multi-part messages) and because the user may choose to retrieve the information for only a few of the matches or even none at all.

Because a large-scale quantitative study of SMS latency has already been done multiple times throughout the almost 20 year history of SMS (see a relatively recent study in [10]), we focus here on a smaller qualitative study related specifically to iTrust over SMS.

First, the delivery latency (and consequently the average search latency) is mostly constant among the service

providers of each mobile phone, so long as the nodes use the same major service provider. Specifically, mobile devices within the same major service provider communicate relatively quickly (usually less than 10 seconds, but sometimes as little as 4 seconds). In particular, two T-Mobile phones had an average search latency of under 10 seconds, two Sprint Nextel phones had similar results, as did two Verizon phones and two AT & T phones, which comprise the four major national service providers in the United States.

Second, the average search latency between devices that use two *different* U.S. service providers is considerably *worse* than devices that use the same U.S. service provider. For example, one T-Mobile phone communicating with a non-T-Mobile phone had an average search latency of seven minutes (about three and one-half minutes per SMS delivery latency); however, after repeated tests between the same two nodes, the average search latency reduced to just under three minutes. Presumably, the SMSC communication between service providers is not optimized or prioritized to handle out-of-network messages (although a more extensive study is required to determine the exact cause). Likewise, the out-of-network SMS messages seem to be delivered faster after the SMSCs establish some form of adaptive or smart routing.

Third, brand licensees (*i.e.*, secondary companies that partner with the four major U.S. service providers) had the *worst* average search latency for out-of-network communication. For example, Virgin Mobile (the brand licensee) uses the Sprint Nextel network (one of the four U.S. networks) and consistently had the worst latency when a Virgin Mobile device was used with any non-Virgin Mobile device, at more than 15 minutes per search to receive a response to a match. The exact reason for this behavior is unknown.

In conclusion, the average search latency varied widely across different service providers. However, the average search latency was consistently less if all of the participating nodes use the same service provider and consistently more if the nodes use different service providers.

VI. Related Work

Existing systems for mobile search, including AOL Mobile [1], Google SMS [6], Windows Live Mobile [17] and Yahoo! OneSearch [19], are based on conventional centralized search engines on the Internet. Those systems use a limited set of pre-defined topics, and either special keywords within the search query (*e.g.*, "directions" to obtain directions) or a specialized parser to determine the intended topic (*e.g.*, "INTC" for a stock quote). The centralized search engines are subject to censorship, filtering, and subversion, which the iTrust over SMS system aims to defeat.

Other mobile search systems, based on centralized search engines, have been developed. The SMSFind system [2], [3] utilizes existing centralized Web search engines. It does not use pre-defined topics but, rather, allows the user to enter an explicit contextual hint about the search topic. SMSFind uses information retrieval techniques to extract an appropriate condensed 140-byte snippet as the final SMS search response, which currently the iTrust over SMS system does not do.

The 7DS system [13] supports information sharing among peers that are not necessarily connected to the Internet. It uses a multi-hop flooding algorithm together with multicasting of queries, which is less trustworthy. In contrast to the above systems, iTrust does not use a centralized search engine; moreover, it does *not* use indiscriminate flooding. Rather, it 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. Moreover, it does not relay metadata or requests that a node has seen previously.

The Distributed Mobile Search Service [9] is a peer-to-peer system that broadcasts query results locally and forwards them over several hops. It is based on a passive distributed index *i.e.*, a local index cache on each mobile device, that contains keywords and corresponding document identifiers, for received query results. iTrust over SMS also maintains a distributed index, with metadata keywords and corresponding node addresses and resource ids. However, iTrust over SMS distributes the metadata and corresponding node addresses and resource ids first, rather than on receipt of the query results.

The Mobile Agent Peer-To-Peer (MAP2P) system [7] supports mobile devices in a Gnutella file-sharing network using mobile agents. The mobile agent (rather than the mobile device) attaches itself to the peer-to-peer network, and acts as a proxy for the mobile device. The iTrust over SMS system has a lower message cost than Gnutella and, thus, a lower message cost than the MAP2P system.

Systems for social networks exploit the trust that members have in each other, and route information and requests based on their relationships. Tiago *et al.* [15] describe a system for mobile search in social networks based on the Drupal content site management system. Their system is based on the network of social links formed from the mobile phone's address book, similar to the iTrust over SMS approach.

Yang *et al.* [20] propose a search mechanism for peer-to-peer networks formed by nodes that share similar interests. Likewise, iTrust over SMS allows users interested in a particular topic or cause to form a social network, so that they can share information. Currently, we are investigating whether such interest groups can be protected against manipulation by subversive participants.

Several information sharing systems for social networks are concerned with privacy and anonymity. OneSwarm [8] is a peer-to-peer system that allows information to be shared by users either publicly or anonymously, using a combination of trusted and untrusted peers. OneSwarm aims to protect the users' privacy, which iTrust over SMS does not aim to do. Rather, iTrust over SMS aims to avoid the censorship

and filtering inherent in centralized search and retrieval, and to ensure the spread of information, which runs counter to the idea of keeping secrets (*i.e.*, privacy).

Quasar [18] is an information sharing system for social networks which, like iTrust over SMS, is probabilistic. Quasar aims to protect the users' sensitive information, which differs from the trust objective of iTrust over SMS.

VII. Conclusions and Future Work

We have described the iTrust over SMS system, a decentralized search and retrieval system that uses SMS messages over mobile networks. A developer for any mobile phone based on the Android platform can use the iTrust over SMS API to add this search and retrieval functionality to an existing application or even create a new application based on iTrust over SMS. Our performance evaluation of iTrust over SMS shows, by analysis for a relatively large network and by emulation for a smaller network, that the probability of information retrieval is high, even if some of the mobile phones are not available. It also shows that the average latency is consistently less when the participating nodes use the same mobile service provider, and consistently more when they use different mobile service providers.

In the future, we plan to offer iTrust over SMS to public users and test the feasibility of average-size social networks in real-life scenarios using physical mobile devices. The iTrust over SMS API allows any type of data to be transmitted; however, currently, the user application must do its own non-plain-text data conversion (*i.e.*, bit packing). We plan to add methods to the iTrust over SMS API that make binary data transfer as easy as that for plain text using existing functions. Moreover, we plan to extend the iTrust over SMS protocol to Wi-Fi Direct and/or Bluetooth to support search and retrieval over mobile ad-hoc networks.

Acknowledgment

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

References

[1] AOL Mobile, http://www.aolmobile.com

[2] J. Chen, B. Linn and L. Subramanian, "SMS-based contextual Web search," *Proceedings of the 2009 ACM SIGCOMM MobiHeld Workshop*, Barcelona, Spain, August 2009, pp. 19–24.

[3] J. Chen, L. Subramanian and E. Brewer, "SMS-based Web search for low-end mobile devices," *Proceedings of the 16th ACM International Conference on Mobile Computing and Networking*, Chicago, IL, September 2010, pp. 125–136.

[4] Y. T. Chuang, I. Michel Lombera, L. E. Moser and P. M. Melliar-Smith, "Trustworthy distributed search and retrieval over the Internet," *Proceedings of the 2011 Interntational Conference on Internet Computing*, Las Vegas, NV, July 2011, pp. 169–175.

[5] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, John Wiley & Sons, New York, NY, 1968.

[6] Google SMS, http://www.google.com/sms

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

[8] T. Isdal, M. Piatek, A. Krishnamurthy and T. Anderson, "Privacy preserving P2P data sharing with OneSwarm," *Proceedings of the 2010 ACM Special Interest Group on Data Communications Conference*, New Delhi, India, September 2010, pp. 111–122.

[9] 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.

[10] X. Meng, P. Zerfos, V. Samanta, S. H. Y. Wong and S. Lu "Analysis of the reliability of a nationwide short message service," *Proceedings of the 26th Annual IEEE Conference on Computer Communications*, Anchorage, AK, May 2007.

[11] I. Michel Lombera, Y. T. Chuang, P. M. Melliar-Smith and L. E. Moser, "Trustworthy distribution and retrieval of information over HTTP and the Internet," *Proceedings of the International Conference on the Evolving Internet*, Luxembourg City, Luxembourg, June 2011, pp. 7–13.

[12] 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, Lecture Notes in Computer Science 7129, Springer, pp. 351–366.

[13] 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.

[14] D. R. Sandler and D. S. Wallach, "Birds of a FETHR: Open, decentralized micropublishing," *Proceedings of the 8th International Workshop on Peer-to-Peer Systems*, Boston, MA, April 2009.

[15] 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.

[16] G. Tsirulnik, "Global SMS traffic to reach 8.7 trillion by 2015: Study," *Mobile Commerce Daily*, February 3, 2011, http://www.mobilecommercedaily.com/2011/02/03/global-sms-traffic-to-reach-8-7-trillion-by-2015

[17] Windows Live Mobile, http://home.mobile.live.com

[18] B. Wong and S. Guha, "Quasar: A probabilistic publish-subscribe system for social networks," *Proceedings of the 7th International Workshop on Peer-to-Peer Systems*, Tampa Bay, FL, February 2008.

[19] Yahoo! OneSearch, http://mobile.yahoo.com/onesearch

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