Decentralized Mobile Search and Retrieval Using SMS and HTTP to Support Social Change

Isaí Michel Lombera, Yung-Ting Chuang, L.E. Moser, and P.M. Melliar-Smith

Department of Electrical and Computer Engineering,
University of California, Santa Barbara,
Santa Barbara, CA 93106 USA
{imichel,ytchuang,moser,pmms}@ece.ucsb.edu

Abstract. Recent events have demonstrated the utility of mobile devices to coordinate mass gatherings and organize protests in support of social change and the cause of democracy. However, a common attack against the social networking abilities of mobile phone wielding protesters has been government action to censor centralized search and social networking sites. This paper describes a decentralized search and retrieval system, named iTrust, that provides greater resistance against the vulnerabilities inherent in centralized services. In particular, it describes the iTrust with SMS interface and the iTrust SMS-HTTP bridge, which enable any SMS-capable mobile phone to communicate with other nodes in the iTrust network. It also describes an Android mobile phone interface that builds on the basic SMS capabilities of a mobile phone and that offers a user-friendly way of accessing the iTrust with SMS implementation. Finally, the paper presents an evaluation of the iTrust search and retrieval system.

Keywords: decentralized search and retrieval, HTTP, iTrust, mobile search and retrieval, SMS.

1 Introduction

As mobile phones have become pervasive in day-to-day life, mobile applications have transcended from basic communication and entertainment services into enablers of societal and political transformation. Recently, social networks such as Twitter and Facebook, as well as search services such as Google and Bing, have been used to help coordinate mass uprisings and revolutions in the world. Unfortunately, centralized systems, whether controlled by a government or a business, are reliant on one or a few nodes that can be easily subverted or censored. If a service provider does not cooperate with such censoring entities, access to the service might be denied entirely. In Egypt and Syria, the Facebook group meeting service was used to help organize protest meeting places and times. In both countries, the government disabled the Internet to hinder the organization of those meetings.

J.Y. Zhang et al. (Eds.): MobiCASE 2011, LNICST 95, pp. 152–171, 2012.

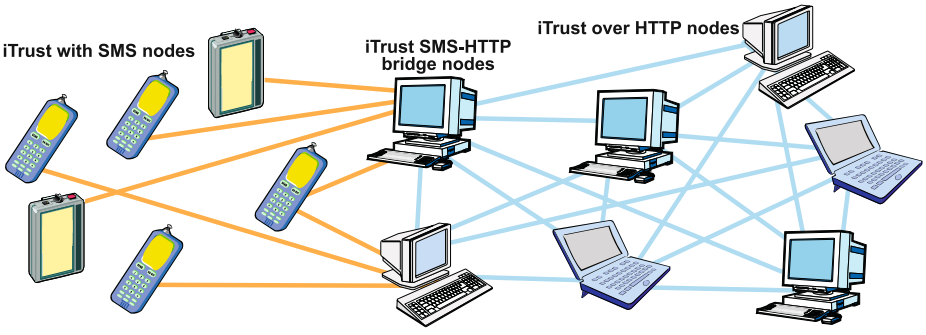

Fig. 1. The iTrust network, showing the iTrust with SMS nodes, the iTrust SMS-HTTP bridge nodes, and the iTrust over HTTP nodes

A decentralized search and retrieval system where multiple nodes, or peers, in the system share queries, metadata, and documents can better withstand temporary or sustained network blocking and shutdowns. Peers can re-route network traffic away from non-operational or non-responsive nodes and can, in some cases, fetch a document from one of several alternative sources.

The iTrust system is a distributed search and retrieval system that does not rely on a centralized search engine, such as Google, Yahoo! or Bing; thus, it is resistant to censorship by central administrators. Our previous implementation of iTrust is based on the HyperText Transfer Protocol (HTTP), and is most appropriate for desktop or laptop computers on the Internet. However, most participants in demonstrations probably use mobile phones to organize their activities. In many countries of the world, mobile phones are the only computing platform generally available; consequently, it is appropriate to provide the iTrust system on mobile phones.

Thus, we have extended the iTrust search and retrieval system based on HTTP, so that it does not rely only on the Internet but can also utilize the cellular telephony network. In particular, we have extended the iTrust system to allow users of mobile phones to connect to iTrust via the Short Message Service (SMS), so that they can benefit from the decentralized search and retrieval service that iTrust provides. Our objective is not to supplant HTTP but instead to have SMS work along side it, to increase accessibility during the dynamic environment of a demonstration or protest. Figure 1 illustrates the extended iTrust network.

In this paper, first we briefly describe the design of the iTrust search and retrieval system that uses HTTP over the Internet. Next, we describe the implementation of iTrust with SMS, focusing on the iTrust SMS-HTTP bridge that allows any hardware-capable iTrust over HTTP node to act as a relay of queries that originate from an SMS-capable mobile phone. We also describe how information is fetched and transmitted over the iTrust SMS-HTTP bridge to the querying mobile phone. This description is followed by a typical use case of iTrust with SMS by a mobile phone and also a description of a custom Android

application that enables users to make queries and receive query results. Next, we present an evaluation of iTrust and, then, we present related work. Finally, we summarize our current work and discuss future work to create an even more robust iTrust network.

2 Mobile Search and SMS

Mobile search is fundamentally different from desktop search, due to the form factor, the limited bandwidth, and the battery life of the mobile device. Sohn *et al.* [1] address human factors in their study of mobile information needs.

In desktop search, users can use a simple search interface to enter keyword queries. The accuracy of the results is generally satisfactory if the desired results are within the first 10 URLs returned. If not, the user can interactively refine his/her queries in subsequent search rounds.

In mobile search, it is expensive and tedious for a user to explore even the two most relevant pages returned by a traditional centralized search engine. Moreover, in mobile search, the information sought tends to focus on narrower topics, and the queries often are shorter, *e.g.*, requests for phone numbers, addresses, times, directions, *etc.*

Kamvar *et al.* [2] have found that most mobile search users have a specific topic in mind, use the search service for a short period of time, and do not engage in exploration. In a subsequent study [3], they found that the diversity of search topics for low-end mobile phone searches is much less than that for desktop searches.

The Short Message Service (SMS) works on low-end mobile 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 [4]. 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 protocol for information exchange after human voice.

In SMS-based search, the query and the response are limited to 140 bytes each. Moreover, the user has to specify a query and obtain a response in one round of search. An iTrust SMS request (query) consists of a list of keywords, which are typically less than 140 bytes. An iTrust SMS response simply returns the requested information if it is small (less than 140 bytes). If the requested information or document is larger, it is fragmented into multi-part SMS messages. Alternatively, the iTrust SMS response can return a URL, which is typically less than 140 bytes.

3 Design of iTrust

The iTrust search and retrieval system uses HTTP over the Internet and 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 time.

In an iTrust network, 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 participating nodes that are chosen at random, as shown in Figure 2. The metadata are distinct from the information that they describe, and include a list of keywords and the URL of the source of the information.

Other nodes, the *requesting nodes*, request and retrieve information. Such nodes generate requests (queries) that refer to the metadata, and distribute the requests to a subset of the participating nodes that are chosen at random, as shown in Figure 3.

The participating nodes compare the metadata in the requests that they receive with the metadata that they hold. If such a node finds a match, which we call an *encounter*, the matching node returns the URL of the associated information to the requesting node. The requesting node then uses the URL to retrieve the information from the source node, as shown in Figure 4.

Distribution of the metadata and the requests to relatively few nodes suffices to achieve a high probability that a match occurs. Moreover, the strategy is robust. Even if some of the randomly chosen nodes are subverted or non-operational, the probability of a match is high, as shown in Section 6. Moreover, it is not easy for a small group of nodes to subvert the iTrust mechanisms to censor, filter or subvert information.

4 Implementation of iTrust

The iTrust with SMS system enables any node (laptop, desktop, server) to act as a bridge between an SMS-capable mobile device and an iTrust over HTTP node. The only requirement for an iTrust with SMS node is having a hardware interface for receiving and transmitting SMS messages; a simple and inexpensive cellular modem suffices. Note that only a single hardware interface is required for sending and receiving SMS messages. (Not all iTrust nodes need to be SMS-capable.) The result is that an existing iTrust network can remain unchanged, only the iTrust SMS-HTTP bridge node must be software updated.

To explain the iTrust with SMS-HTTP bridge, we trace the path taken by an SMS request (query) message sent to the iTrust network and the path taken by an SMS response (result) message sent from the iTrust network.

Figure 5 provides a system block diagram that shows the communication path taken by SMS request and response messages. Specifically, it shows the three main components of the iTrust implementation with the SMS-HTTP bridge: the cellular network, an iTrust with SMS node, and an iTrust over HTTP node. The blocks (numbered threads or spools) show only the APIs relevant to the discussion of iTrust with SMS. Each block actually has many more APIs for the iTrust over HTTP implementation. Additionally, thread blocks are numbered to explain the examples. In a typical iTrust network, multiple threads can be running for each iTrust node.

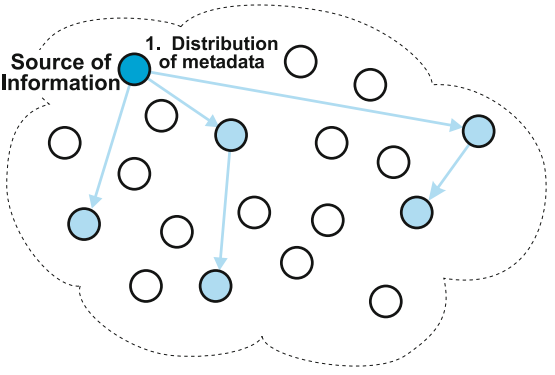

Fig. 2. A source node distributes metadata, describing its information, to randomly selected nodes in the network

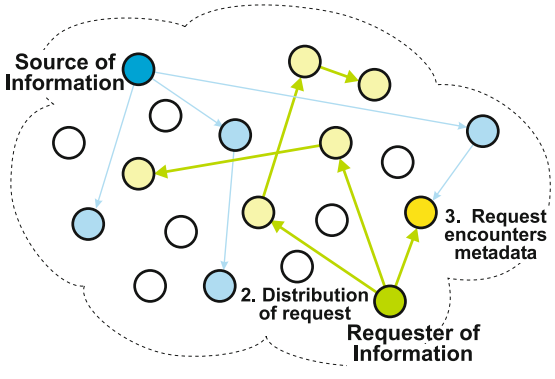

Fig. 3. A requesting node distributes its request to randomly selected nodes in the network. One of the nodes has both the metadata and the request and, thus, an encounter occurs.

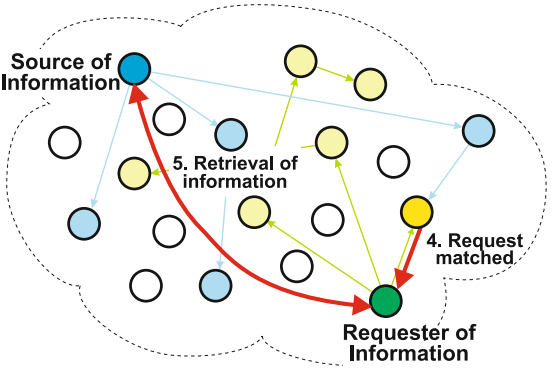

Fig. 4. A node matches the metadata and the request and reports the match to the requester, which then retrieves the information from the source node

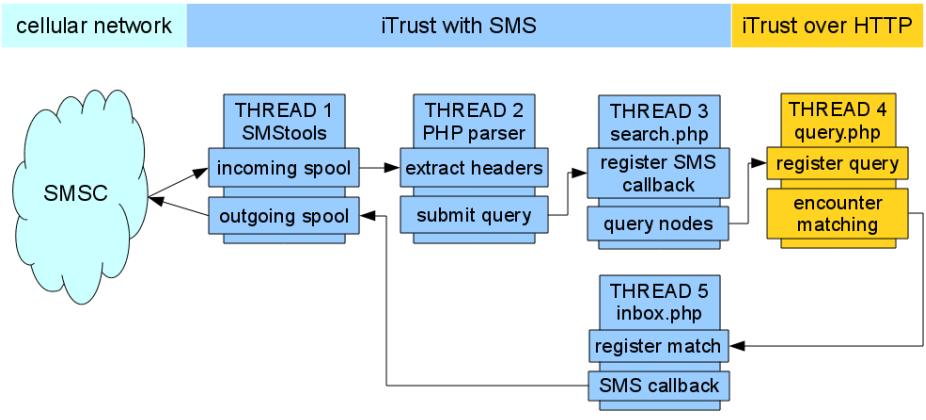

Fig. 5. The iTrust system block diagram showing the cellular network, the iTrust with SMS component, and the iTrust over HTTP component

4.1 Cellular Network

The cellular network, for the purposes of this discussion, is modeled simply by the Short Message Service Center (SMSC), which the mobile phone service providers use to relay SMS messages. In the next section, we expand the SMSC concept slightly to include mobile phones to enable presentation of the user interface for iTrust with SMS.

Briefly, the SMSC is a store-and-forward entity in the network of the mobile phone service provider. When a user sends an SMS message, the message is stored in the SMSC and, when possible, it is forwarded to the intended destination. If the destination is unavailable, the message is spooled for later transmission.

For the iTrust network, there is no distinction between a single SMSC or multiple SMSCs that handle SMS relaying. iTrust does not require any service provider agreements or integration with existing mobile networks; it simply uses a *cell phone number* like any mobile device seen by the SMSC.

4.2 iTrust with SMS

First and foremost, iTrust with SMS is an extension of the iTrust over HTTP implementation; SMS capabilities are added to the API and the iTrust HTTP implementation remains intact and operational. Thus, an iTrust with SMS node can interact with both an Internet node and a cellular network node. The iTrust SMS-HTTP bridge allows SMS-enabled mobile phones in the cellular network to interact with iTrust over HTTP nodes on the Internet.

In addition to the custom code written for the iTrust SMS-HTTP bridge, the open-source SMStools package is used to handle incoming and outgoing spooling of SMS messages. SMStools offers several advanced features that are easily leveraged by iTrust including SMS message formatting, header automation, and message validation.

The iTrust SMS-HTTP bridge requires a single hardware interface for sending and receiving SMS messages. Optionally, SMStools can be configured to handle multiple cellular modems from multiple cellular network providers and can spool the SMS messages accordingly. However, the typical iTrust configuration uses a single cellular modem to act as both the incoming and the outgoing SMS device and to have SMStools spool both incoming and outgoing SMS messages.

Within the iTrust with SMS component, THREAD 1 consists of SMStools which spools both incoming and outgoing SMS messages. Incoming SMS messages are registered with an event handler that triggers a command-line (*not* a Web server) PHP script in THREAD 2. Outgoing SMS messages are sent by writing a properly formatted plain text file and placing it in a specific SMStools monitored directory, so that an SMS response message is created and sent to the querying mobile device. Outgoing SMS messages are further explained below in the THREAD 5 functionality description.

The SMS message parser in THREAD 2 performs simple text processing to extract headers such as the sender's cell phone number and query. The extracted data are then packaged into an HTTP GET statement and submitted as a query to THREAD 3.

Particularly in THREAD 3, iTrust with SMS functionality is tightly integrated with existing iTrust over HTTP functionality; however, it remains distinct from pure iTrust over HTTP nodes. Along with query text and timestamp information, the sender's callback cell phone number is registered to enable results sent to the SMS-HTTP bridge node to be relayed back to the mobile phone. The bridge node then queries the nodes in the iTrust network as if the query originated directly from the bridge node (not as an SMS-relayed query). The cell phone number itself is not included in the query package; only the SMS-HTTP bridge node is aware of this cell phone number. Thus, the bridge node masquerades as an iTrust over HTTP node performing a routine search.

Nodes in the iTrust network execute the routines in THREAD 4 when queried for results. First, the query is registered so that any duplicate relayed queries are ignored and then an encounter (match), if any, causes a response message containing a result to be sent back to the querying node. THREAD 4 exhibits typical iTrust over HTTP behavior, no SMS information or awareness is required from a node running this thread.

The *SMS callback* routine in THREAD 3 is perhaps the most extensive part of the iTrust with SMS component. It has the dual function of pulling the source information and packaging that information appropriately before handing off the message to SMStools for spooling.

In THREAD 5, first, the resource is automatically fetched from the source node and temporarily stored on the bridge node for further processing. Second, the document (if it is less than 140 bytes) is formatted for SMS and the callback cell phone number of the original SMS querying user is added. Third, the message is written to an SMStools monitored directory, which further appends relevant message fields (*i.e.*, SMSC information, text formatting, *etc.*) before spooling

the message for delivery (THREAD 1). Finally, the message is sent to the SMSC for delivery to the user's mobile device.

4.3 iTrust over HTTP

The iTrust over HTTP implementation runs on laptop, desktop or server nodes on the Internet and perhaps also on mobile phones on the Internet. There might be hundreds or thousands of iTrust over HTTP nodes in a typical iTrust network. The primary goal of each iTrust over HTTP node is to match a query it receives with a local resource and to respond with a URL for that resource, if an encounter or hit occurs. Each iTrust over HTTP node relays the query to its own membership list as specified by the local node administrator's preferences and/or load balancing services built into iTrust. The exact method of query relaying and load balancing is outside the scope of this paper. Only a few APIs related to encounters are discussed here.

When a query arrives at a node, the query is registered in THREAD 4 using the *register query* routine. If it has been seen previously, processing stops as repeating an old query is not useful. If the query is indeed new, the query text is compared against a database consisting of metadata and URLs of the corresponding resources in THREAD 4 using the *encounter matching* routine. If the query keywords match locally stored metadata, the node responds to the requesting node with the URL. Note that, in this case, the requesting node is the iTrust SMS-HTTP bridge node. It is *not* the SMS mobile phone node.

4.4 A Typical SMS Request/Response Path

A typical path along which SMS request and response messages travel from the mobile phone and back again is described below.

Sending the Request. A user sends an SMS request (query) message from his/her mobile phone with a simple text query. After being relayed by the SMSC, the SMS message enters the iTrust SMS-HTTP bridge node through a cellular hardware interface (such as a cellular modem) and is held in the incoming spool (THREAD 1). A new message in the incoming spool triggers an event handler (THREAD 2), which then loads a PHP script to process the spool and extract the user's cell phone number and text query. The cell phone number is registered for callback purposes (THREAD 3), and the query enters the iTrust network exactly as if it were originated by an iTrust over HTTP node. The query is relayed through the iTrust network until an encounter occurs (THREAD 4).

Receiving the Response. A response message is sent from an iTrust over HTTP node to the iTrust SMS-HTTP bridge node (THREAD 5). After normal processing by iTrust, the resource is fetched and placed in local storage. The locally stored resource (or a URL for the locally stored resource, if the resource is large) is further processed into an SMS message, placed into the outgoing spool, and relayed to the SMSC (THREAD 1). The user receives a new SMS message, sent from the iTrust SMS-HTTP bridge node.

4.5 API Function Call Swapping and Race Conditions

In Figure 5, in the iTrust with SMS component under THREAD 3, there are two APIs: *register SMS callback* and *query nodes*. The iTrust over HTTP nodes (where a *register SMS callback* is simply a register query callback) have the order of these two calls swapped for performance reasons. In practice, querying a node before registering the query leads to better performance in the Apache prefork model. This model inherently prevents the occurrence of a race condition, because the query is registered long before another node responds with a result. This behavior holds true particularly for threads numbering in the several thousands; however, in practice, even a self-query on a single node does not result in a race condition.

The iTrust SMS-HTTP bridge node has a stricter requirement. An iTrust with SMS node must *always* register the SMS callback cell phone number before querying another iTrust node. Otherwise, an iTrust node that is not SMS-capable might respond to a query before the callback cell phone number is registered. In this case, the particular response is not relayed to the mobile phone; future responses, that arrive after the SMS callback cell phone number has been registered, will be relayed.

Simply swapping the order to that shown in Figure 5 prevents a race condition from occurring.

5 iTrust with SMS User Interface

The addition of iTrust with SMS to iTrust over HTTP required not only an additional bridge mechanism on the iTrust nodes, but also a new interface to allow the mobile phone user to interact with the iTrust network. While iTrust over HTTP requires the use of a Web browser to search and retrieve documents, iTrust with SMS needs a more user-friendly mobile phone interface that conforms to the expectations of the user for a typical Instant Messaging (IM) service. For iTrust with SMS, we compare a generic SMS Instant Messaging interface with a custom-built Android interface for iTrust with SMS.

As an example, consider an ad hoc protest demonstration scheduling service that periodically distributes meeting locations and times to iTrust nodes. For each demonstration, there exists a file that includes basic information such as meeting location and time. A query from one iTrust node begins a search among other participating nodes in the iTrust network, and an encounter returns the demonstration named file that includes the meeting information. In particular, we consider the case that a user searches for demonstration information related to *Tahrir Square* in Cairo, Egypt.

5.1 iTrust with SMS Using the Generic IM Interface

The interface for iTrust with SMS is minimalistic in both function and use, compared to the Web interface for iTrust over HTTP. Requests (queries) are

Fig. 6. iTrust with SMS, using a generic Instant Messaging interface

simply SMS messages that are sent to the cell phone number of the iTrust SMS-HTTP bridge node; similarly, responses are SMS messages containing document data sent back to the user. There is no user hardware requirement apart from having an SMS-capable mobile phone; the SMS message may be sent to a dumb phone or a smart phone, with the user experience remaining consistent. Because the primary focus of a user of iTrust with SMS is simply to make a query, there is no interface for modifying the membership, adding resources, or configuring user parameters, as in the iTrust over HTTP Web interface.

Figure 6 shows an image of a typical iTrust with SMS interaction between a mobile user and an iTrust node. This particular screen shot uses the standard built-in SMS application bundled with Android (specifically, Android version 2.1); however, apart from aesthetics, the interaction is the same for iOS, webOS, Symbian, *etc.* Note that the only information required to interact with an iTrust node, apart from the query, is the cell phone number of the iTrust node (which is partially obscured). This particular Instant Messaging interface presents all SMS messages between the same callers in a single scrolling conversational type format. In this example, the display shows the user query *Tahrir Square* message sent to the iTrust node. A response message is sent back from the iTrust node to the user approximately one minute later (as shown in the last message); this result (or hit) is the data that correspond to the user's search keywords.

Note that the data itself are returned to the SMS user without reference to the URL, document file name, or address of the source node of the document. This

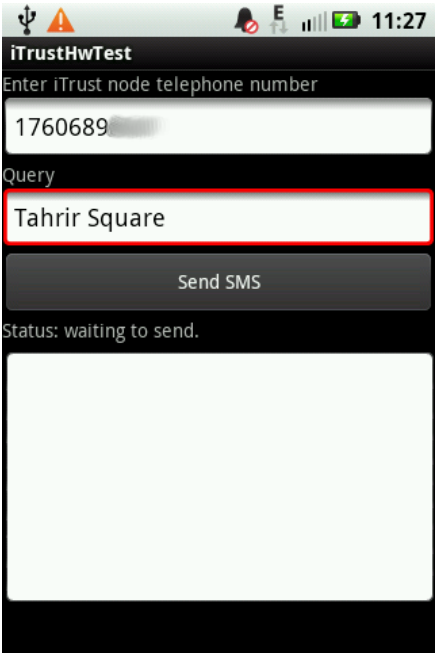

Fig. 7. iTrust with SMS, searching with the custom Android interface

presentation is consistent with the iTrust with SMS functionality, which requires that the SMS-HTTP bridge node itself must fetch the document, package it in an SMS-compatible format, and send back the SMS result. In contrast, the iTrust over HTTP interface simply presents a list of hits and does not fetch the document data automatically.

This simple and direct interaction makes it easy to carry on a conversation of sorts with the iTrust node by simply asking questions (submitting queries) and reading answers (hit data).

5.2 iTrust with SMS Using the Custom Android Interface

The custom Android application for interacting with an iTrust with SMS node is a hybrid of the generic SMS Instant Messaging interface and the iTrust over HTTP interface. Figures 7 and 8 show the submission of a query from the SMS-capable mobile phone and the returned result from the iTrust SMS-HTTP bridge node, respectively. The custom Android interface for iTrust over SMS enhances the generic SMS Instant Messaging interface in that it provides: familiarity for users accustomed to iTrust over HTTP, preset cell phone numbers to iTrust SMS-HTTP bridge nodes, and a framework for handling non-textual result data.

Figure 7 shows the entry of a query into a text editing area that is similar to that in the iTrust over HTTP search interface. Above the query is the pre-entered cell phone number of the iTrust SMS-HTTP bridge node. Although this

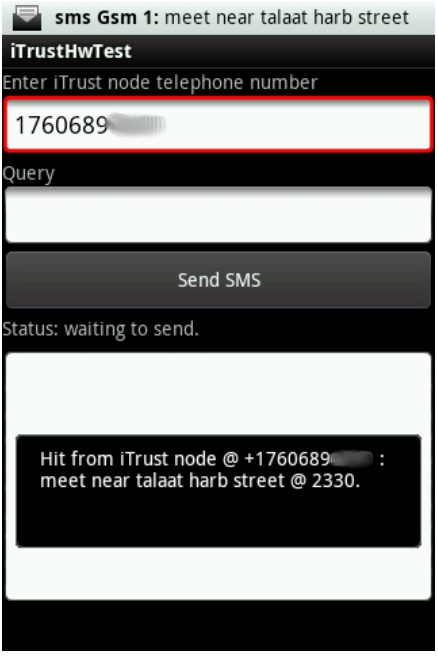

Fig. 8. iTrust with SMS, viewing a hit with the custom Android interface

is a minimal enhancement to the generic SMS interface, the rapid and transient nature of most SMS interactions favors features that reduce extraneous information not related to the SMS message itself. Additionally, once the query is sent, the query text area is cleared, so that the user can easily begin entry of another search query.

Figure 8 shows the result data returned from the iTrust SMS-HTTP bridge node; the result is the same as that for the generic SMS interface result. The resultant data are displayed in text format; however, alternate formats can be handled by the built-in framework. For example, a Portable Document Format (PDF) file sent over SMS would be *offloaded* or *handed off* to Android presumably to be opened by a PDF reader application available on the mobile phone. In this case, the user would be responsible for having access to a separate reader application appropriate to the file type. The iTrust system searches and retrieves all files, regardless of format (so long as the metadata are properly generated); however, the user is responsible for appropriate decoding.

6 Evaluation of iTrust

To evaluate iTrust, we consider the probability of a match, and also the number of messages required to achieve a match, using both analysis and simulation based on our implementation of iTrust. We assume that all of the participating

nodes have the same membership set of participating nodes. In addition, we assume that communication is reliable and that all of the participating nodes have enough memory to store the source files and the metadata.

6.1 Probability of a Match

First, we consider the probability that, for a given request, a match (encounter) occurs, *i.e.*, that there are one or more nodes at which a match occurs for that request.

Analysis. We consider an iTrust network with a membership of n participating nodes, where a proportion x of the n nodes are operational (and, thus, a proportion $1-x$ of the n nodes are not operational). We distribute the metadata to m nodes and the requests to r nodes. The probability p that a node has a match is given by:

$$p = 1 - \left(\frac{n-mx}{n} \frac{n-1-mx}{n-1} \cdots \frac{n-r+1-mx}{n-r+1} \right). \tag{1}$$

Equation (1) holds for $n \geq mx + r$. If $mx + r > n$, then $p = 1$.

Figures 9, 10 and 11 show the probability p of a match obtained from Equation (1) with $n = 250$ nodes where $x = 100\%$, 80% and 60% of the participating nodes are operational, respectively, as a function of $m = r$. As we see from the graphs, the probability p of a match increases, and approaches 1, as $m = r$ increases.

Simulation. Using our implementation of iTrust, we performed simulation experiments to validate the analytical results for the probability of a match obtained from Equation (1).

Before we run our simulation program, we delete all resources and data from the node. Next, the program adds the nodes to the membership. Once the nodes are added to the membership, we supply the number n of nodes for distribution of metadata and requests, and the proportion x of operational nodes, to the simulation program. Next, we call the source nodes to upload files and the program then creates the corresponding metadata. Then, the program randomly selects m nodes for metadata distribution and distributes the metadata to those nodes. Then, the program randomly selects r nodes for request distribution and distributes the requests to those nodes. If one or more nodes returns a response, there is a match and the simulation program returns 1; otherwise, there is no match and the simulation program returns 0.

We repeated the same process 100 times for the source nodes and correspondingly for the requesting nodes, and plot the mean results in our simulation graphs. We collected simulation data for 250 participating nodes when 100%, 80% and 60% of the nodes are operational.

Figures 9, 10 and 11 show the simulation results with 250 nodes where 100%, 80% and 60% of the participating nodes are operational, respectively, as a function of $m = r$. As we see from these graphs, the simulation results are very close

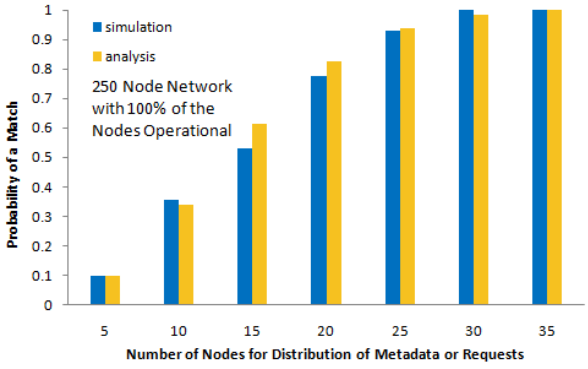

Fig. 9. Match probability vs. number of nodes for distribution of metadata or requests in a network with 250 nodes where 100% of the nodes are operational

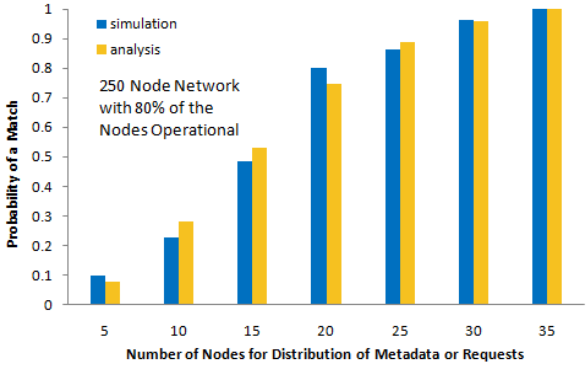

Fig. 10. Match probability vs. number of nodes for distribution of metadata or requests in a network with 250 nodes where 80% of the nodes are operational

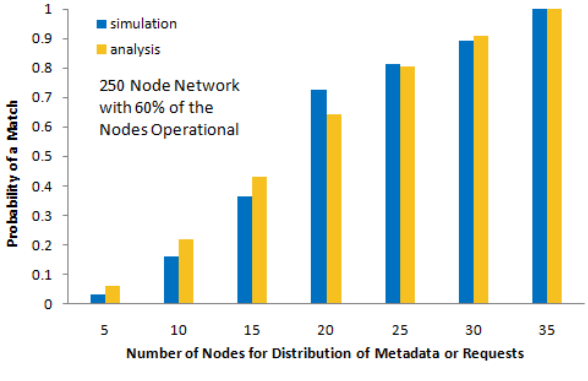

Fig. 11. Match probability vs. number of nodes for distribution of metadata or requests in a network with 250 nodes where 60% of the nodes are operational

to the analytical results calculated from Equation (1). As these results indicate, iTrust retains significant utility even in the case where a substantial proportion of the nodes are non-operational.

6.2 Number of Messages to Achieve a Match

Next, we consider the mean number of messages required to achieve a match for a given request.

Analysis. Again, we consider an iTrust network with a membership of n participating nodes, where the proportion of participating nodes that are operational is x. We distribute the metadata to m nodes and the requests to r nodes. The probability p of exactly k matches is given by:

$$p(k) = \frac{(\frac{mx}{k}\frac{mx-1}{k-1}\cdots\frac{mx-k+1}{1})(\frac{n-mx}{r-k}\frac{n-mx-1}{r-k-1}\cdots\frac{n-mx-r+k+1}{1})}{(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1})}. \quad (2)$$

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

The mean number y of messages required to achieve a match is given by:

$$y = 2 + r + \sum_{k=1}^{\min\{mx,r\}} kp(k). \quad (3)$$

The terms on the right side of Equation (3) represent: 1 request from the mobile phone to an iTrust SMS-HTTP bridge node, r requests from the iTrust SMS-HTTP bridge node to iTrust over HTTP nodes, k responses reporting matches from the iTrust over HTTP nodes to the iTrust SMS-HTTP bridge node, and 1 response from the iTrust SMS-HTTP bridge node to the mobile phone.

Figures 12, 13 and 14 show the number of messages obtained from Equations (2) and (3) with $n = 250$ nodes where $x = 100\%$, 80% and 60% of the participating nodes are operational, respectively, as a function of $m = r$. As we see from the graphs, the number of required messages increases as the probability p of a match increases (and as $m = r$ increases), but is bounded by $2 + 2r$.

Simulation. Using our implementation of iTrust, we performed simulation experiments to validate the analytical results for the mean number of messages to achieve a match obtained from Equations (2) and (3). The simulation experiments were performed as described previously in Section 6.1.

Figures 12, 13, and 14 show the simulation results with 250 nodes where 100%, 80% and 60% of the participating nodes are operational, respectively, as a function of $m = r$. As we see from these graphs, the simulation results are very close to the analytical results calculated from Equations (2) and (3).

Figures 9, 10 and 11 and Figures 12, 13 and 14 show the benefit-cost tradeoffs between the probability of achieving a match and the number of messages required to achieve a match. Note that the number of messages required to achieve a match is much greater than for centralized search engines, but is much less than for flooding strategies.

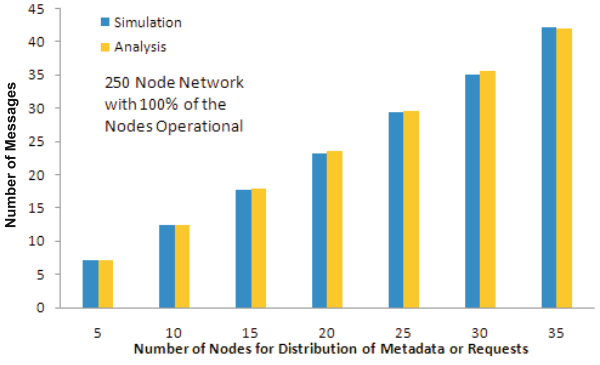

Fig. 12. Number of messages vs. number of nodes for distribution of metadata or requests in a network with 250 nodes where 100% of the nodes are operational

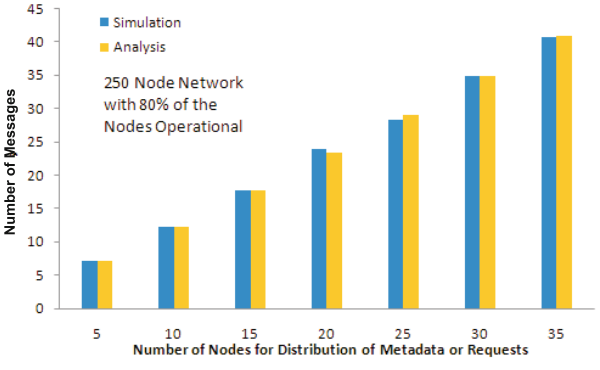

Fig. 13. Number of messages vs. number of nodes for distribution of metadata or requests in a network with 250 nodes where 80% of the nodes are operational

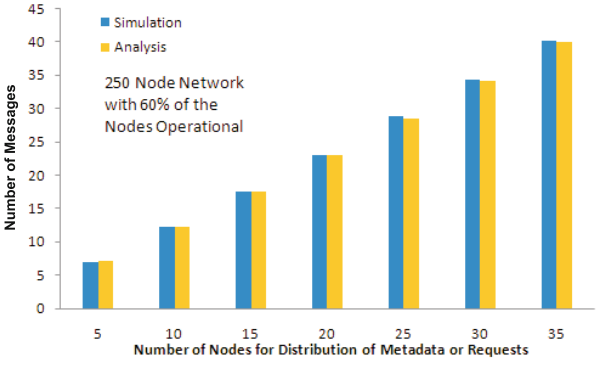

Fig. 14. Number of messages vs. number of nodes for distribution of metadata or requests in a network with 250 nodes where 60% of the nodes are operational

7 Related Work

Existing services for mobile Web search, including AOL Mobile [5], Google SMS [6], Windows Live Mobile [7] and Yahoo! OneSearch [8], are based on conventional centralized search engines. However, the results obtained from those systems are often not meaningful or not consistent for queries related to arbitrary topics. The reason is that they 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). Moreover, the centralized search engines are subject to censorship, filtering and subversion.

Other mobile search systems, based on centralized search engines, have been developed. The SMSFind system [9,10] utilizes existing conventional centralized search engines at the back-end. SMSFind 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 iTrust does not do. The 7DS system [11] supports information sharing among peers that are not necessarily connected to the Internet. The 7DS system uses a multi-hop flooding algorithm together with multicasting of queries, which is not trustworthy. In contrast to these systems, iTrust does not use a centralized search engine and does *not* use flooding, which is too expensive in message cost.

Bender *et al.* [12] recognize the need for decentralized peer-to-peer Web search because "existing Web search is more or less exclusively under the control of centralized search engines." Mischke and Stiller [13], Risson and Moors [14], and Tsoumakos and Roussopoulos [15] provide 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.*, which is efficient but is vulnerable to manipulation by untrustworthy administrators. The unstructured approach uses randomization, and requires the nodes to find each other by exchanging messages over existing links. The iTrust system uses the unstructured approach, which is less vulnerable to manipulation.

The distributed mobile search service of Lindemann and Waldhorst [16] broadcasts query results locally and forwards them over several hops. It is based on a passive distributed index that comprises, on each mobile device, a local index cache, containing keywords and corresponding document identifiers, where all received query results are cached. The iTrust system also maintains a distributed index, with metadata keywords and corresponding URLs stored on the iTrust nodes. However, iTrust distributes the metadata and the corresponding URLs first, rather than on receipt of the query results, which results in iTrust's having a lower message cost than their distributed mobile search service.

The Mobile Agent Peer-To-Peer (MAP2P) system [17] 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. In some respects, the MAP2P mobile agent is

similar to the iTrust SMS-HTTP bridge node, but iTrust has a lower message cost than Gnutella and, thus, MAP2P.

Systems for social networks exploit the trust that members have in each other, and route information and requests based on their relationships. Gummadi *et al.* [18] investigate the integration of social network search with Web search. They conclude that such integration can lead to more timely and efficient search experiences. Tiago *et al.* [19] 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. Yang *et al.* [20] propose a search mechanism for unstructured peer-to-peer networks based on interest groups, formed by nodes that share similar interests. iTrust likewise allows users interested in a particular topic or cause to form a social network, so that they can share information among themselves. Currently, we are investigating whether such interest groups can be protected against manipulation by subversive participants.

Several peer-to-peer information sharing systems are concerned with trust. Quasar [21] is a probabilistic information sharing system for social networks with many social groups. The objective of Quasar is to protect the users' sensitive information, which is different from the trust objective of iTrust. OneSwarm [22] is a peer-to-peer system that allows information to be shared either publicly or anonymously, using a combination of trusted and untrusted peers. OneSwarm aims to protect the users' privacy, which iTrust does not aim to do. Rather, the trust objective of iTrust is to support free flow of information and prevent censorship, filtering and subversion of information. It might be advantageous to integrate ideas from Quasar or OneSwarm into a future version of iTrust.

8 Conclusions and Future Work

The iTrust with SMS system consists of SMS-capable mobile phones that communicate with iTrust SMS-HTTP bridge nodes that act as relays for search and retrieval requests over the iTrust network. An SMS-capable mobile phone can interact with any number of inter-connected iTrust over HTTP nodes. The iTrust network can be queried from any SMS-capable mobile phone for search and retrieval of basic information. In our implementation, an Android mobile phone application provides a custom interface to facilitate quick searches.

While the iTrust SMS-HTTP bridge provides search and retrieval access to the iTrust network for SMS-capable mobile phones, the iTrust with SMS node lacks the full capabilities of an iTrust over HTTP node. Notably, large documents cannot be easily and efficiently uploaded from, or downloaded to, the mobile phone, and they are hard to read on the small screen of the mobile phone. Of importance to many mobile phone users is the ability to upload and download images, video or audio recordings directly from their mobile phones. For these reasons, we plan to develop an iTrust over SMS application that transforms an SMS-enabled mobile phone into an effective and fully functional peer in the iTrust network.

We also plan to add mobile ad-hoc Wi-Fi capabilities to create a mesh network of local peer-to-peer iTrust nodes. Thus, iTrust mobile users will be immune from even government shutdown of cellular towers, and will be fully autonomous to search and retrieve documents from peers in the local Wi-Fi network. These additions to the iTrust network will strengthen the availability of search and the robustness of retrieval to enable movements of social change.

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

References

1. Sohn, T., Li, K.A., Griswold, W.G., Hollan, J.D.: A Diary Study of Mobile Information Needs. In: 26th ACM SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy, pp. 433–442. ACM Press, New York (2008)
2. Kamvar, M., Baluja, S.: A Large Scale Study of Wireless Search Behavior: Google Mobile Search. In: 24th ACM SIGCHI Conference on Human Factors in Computing Systems, Montreal, Quebec, Canada, April 2006, pp. 701–709. ACM Press, New York (2006)
3. Kamvar, M., Kellar, M., Patel, R., Xu, Y.: Computers and iPhones and Mobile Phones, Oh My!: A Log-Based Comparison of Search Users on Different Devices. In: 18th International Conference on the World Wide Web, Madrid, Spain, pp. 801–810. ACM Press, New York (2009)
4. Tsirulnik, G.: Global SMS Traffic to Reach 8.7 Trillion by 2015: Study. In: Mobile Commerce Daily, February 3 (2011), http://www.mobilecommercedaily.com/2011/02/03/global-sms-traffic-to-reach-8-7-trillion-by-2015
5. AOL Mobile, http://www.aolmobile.com
6. Google SMS, http://www.google.com/sms
7. Windows Live Mobile, http://home.mobile.live.com
8. Yahoo! OneSearch, http://mobile.yahoo.com/onesearch
9. Chen, J., Linn, B., Subramanian, L.: SMS-Based Contextual Web Search. In: 2009 ACM SIGCOMM MobiHeld Workshop, pp. 19–24. ACM Press, New York (2009)
10. Chen, J., Subramanian, L., Brewer, E.: SMS-Based Web Search for Low-End Mobile Devices. In: 16th ACM MobiCom International Conference on Mobile Computing and Networking, Chicago, IL, pp. 125–136. ACM Press, New York (2010)
11. Papadopouli, M., Schulzrinne, H.: Effects of Power Conservation, Wireless Coverage and Cooperation on Data Dissemination among Mobile Devices. In: ACM Symposium on Mobile Ad Hoc Networking and Computing, Long Beach, CA, pp. 117–127. ACM Press, New York (2001)
12. Bender, M., Michel, S., Triantafillou, P., Weikum, G., Zimmer, C.: P2P Content Search: Give the Web Back to the People. In: 5th International Workshop on Peer-to-Peer Systems, Santa Barbara, CA (February 2006)
13. Mischke, J., Stiller, B.: A Methodology for the Design of Distributed Search in P2P Middleware. IEEE Network 18(1), 30–37 (2004)
14. Risson, J., Moors, T.: 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. Tsoumakos, D., Roussopoulos, N.: A Comparison of Peer-to-Peer Search Methods. In: Sixth International Workshop on the Web and Databases, San Diego, CA, pp. 61–66 (June 2003)
16. Lindemann, C., Waldhorst, O.P.: A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications. In: 2nd International Conference on Peer-to-Peer Computing, Linkoping, Sweden, pp. 73–80. IEEE CS Press, Los Alamitos (2002)
17. Hu, H., Thai, B., Seneviratne, A.: Supporting Mobile Devices in Gnutella File Sharing Network with Mobile Agents. In: 8th IEEE Symposium on Computers and Communications, Kemer-Antalya, Turkey. IEEE CS Press, Los Alamitos (2003)
18. Gummadi, K.P., Mislove, A., Druschel, P.: Exploiting Social Networks for Internet Search. In: 5th ACM SIGCOMM Workshop on Hot Topics in Networks, Irvine, CA, pp. 79–84. ACM Press, New York (2006)
19. Tiago, P., Kotiainen, N., Vapa, M., Kokkinen, H., Nurminen, J.K.: Mobile Search – Social Network Search Using Mobile Devices. In: 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, pp. 1201–1205. IEEE CS Press, Los Alamitos (2008)
20. Yang, J., Zhong, Y., Zhang, S.: An Efficient Interest-Group-Based Search Mechanism in Unstructured Peer-to-Peer Networks. In: 2003 International Conference on Computer Networks and Mobile Computing, Shanghai, China, pp. 247–252. IEEE CS Press, Los Alamitos (2003)
21. Wong, B., Guha, S.: Quasar: A Probabilistic Publish-Subscribe System for Social Networks. In: 7th International Workshop on Peer-to-Peer Systems, Tampa Bay, FL (February 2008)
22. Isdal, T., Piatek, M., Krishnamurthy, A., Anderson, T.: Privacy Preserving P2P Data Sharing with OneSwarm. In: 2010 ACM SIGCOMM Special Interest Group on Data Communication Conference, New Delhi, India, pp. 111–122. ACM Press, New York (2010)