Noname manuscript No.
(will be inserted by the editor)

Mobile Decentralized Search and Retrieval Using SMS and HTTP

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

Received: date / Accepted: date

Abstract Events over the past year have demonstrated the utility of mobile devices for coordinating mass gatherings and organizing protests in support of social change. However, governments have countered by censoring or disabling centralized search services and social networking sites. This paper describes a decentralized search and retrieval system, named iTrust, that provides resistance against the vulnerabilities of centralized services. It describes the iTrust with SMS interface and the iTrust SMS-HTTP bridge, which enable any SMS-capable mobile phone to communicate with and obtain information from HTTP nodes in the iTrust network. It also describes the iTrust over SMS protocol, which enables mobile phones to communicate directly with each other and share information in a peer-to-peer fashion bypassing the Internet. The paper also describes an Android user interface that builds on the basic SMS capabilities of mobile phones and that offers a user-friendly way of accessing the iTrust with SMS or iTrust over SMS implementation. Finally, the paper presents a performance evaluation of the iTrust search and retrieval system.

Keywords decentralized search and retrieval; HTTP; iTrust; mobile search and retrieval; peer-to-peer network; SMS

1 Introduction

Mobile phones have become pervasive in daily life; mobile applications, in addition to providing basic communication and entertainment services, have become enablers of societal transformation. 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. However,

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

centralized systems, whether controlled by a government or a business, rely on 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. For example, in some countries, the Facebook group meeting service was used to help organize the places and times of protest meetings. In several cases, governments disabled local access to the Internet to hinder the organization of such meetings.

A decentralized search and retrieval system where multiple nodes, or peers, in the network share documents, metadada, and queries 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 retrieve documents 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 first implementation of iTrust, named iTrust over HTTP [1], is based on the HyperText Transfer Protocol (HTTP), and is most appropriate for desktop or laptop computers on the Internet. However, modern day users expect mobile phones to have many of the same capabilities that more traditional computers have. The modern user wants a computer that fits in his (her) pocket (purse) and that is network enabled. In many countries, mobile phones are the only computing platform generally available; thus, it is appropriate to provide the iTrust system on mobile phones.

We have extended the iTrust search and retrieval system based on HTTP, so that it does not rely only on the Internet but can utilize the cellular telephony network. First, we extended the iTrust over HTTP system to allow users of mobile phones to connect to iTrust over HTTP via the Short Message Service (SMS), so that they can benefit from the decentralized search and retrieval service that iTrust provides. We name this system iTrust with SMS. Our objective is not to supplant HTTP but instead to have SMS work along side it, to increase accessibility during dynamic situations where mobile phones are used. Second, we completely re-implemented the iTrust over HTTP system to work only over SMS, thus creating the iTrust over SMS system. Whereas iTrust with SMS allows mobile phones to send text messages to the iTrust over HTTP network, iTrust over SMS allows mobile phones to form self-contained peer-to-peer networks that are not necessarily connected to the Internet. iTrust over SMS allows mobile phones to search for and retrieve documents entirely within the cellular telephony network; an Internet connection is not required. Figure 1 illustrates the three different kinds of iTrust networks.

The remainder of this paper is organized as follows. Section 2 briefly discusses mobile search and SMS, and Section 3 presents the design of the iTrust search and retrieval system. Next, Section 4 describes 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. Section 5 presents the iTrust with SMS user interface that allows users to make queries and receive query

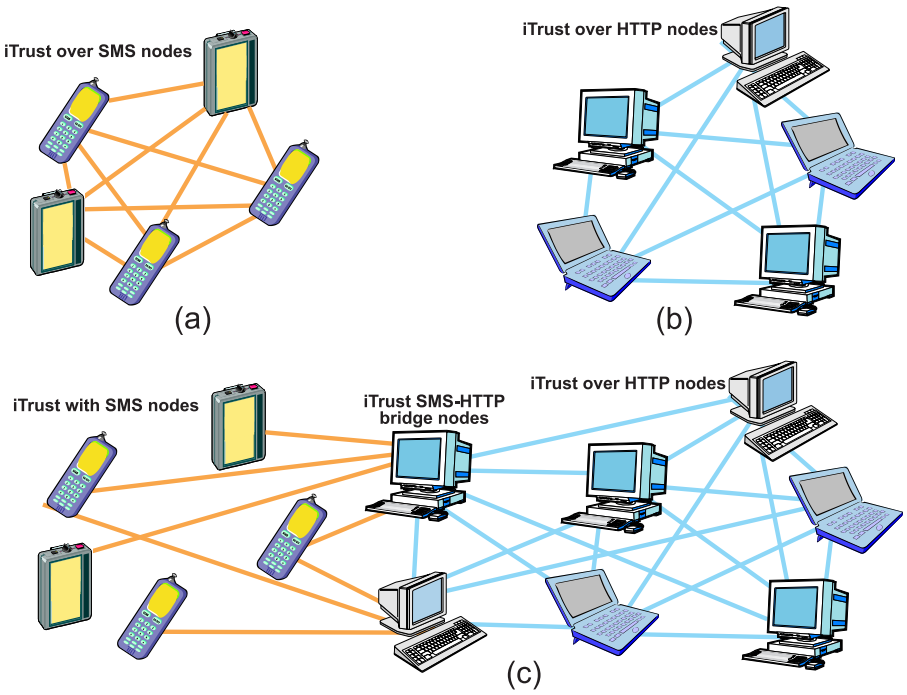


Fig. 1 The three different kinds of iTrust networks, showing: (a) iTrust over SMS nodes, (b) iTrust over HTTP nodes, and (c) iTrust with SMS nodes communicating with iTrust SMS-HTTP bridge nodes communicating with iTrust over HTTP nodes.

results. Next, Section 6 describes the iTrust over SMS protocol and implementation that allows mobile phones in the iTrust network to communicate directly in a peer-to-peer fashion over the cellular telephony network. Section 7 presents the iTrust over SMS user interface that allows users to make queries and receive query results. Following these descriptions, Section 8 presents a performance evaluation of iTrust, and Section 9 presents related work. Finally, Section 10 summarizes our current work, and discusses future work to create an even more robust iTrust search and retrieval network.

2 Mobile Search and SMS

Due to the form factor, the limited bandwidth, and the battery life of the mobile device, mobile search is fundamentally different from desktop search, as Sohn *et al.* [2] have observed. In desktop search, users can use a simple search interface to enter keyword queries. The accuracy 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, 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.* [3] 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 [4], they found that the diversity of search topics for low-end mobile phone searches is much less than that for desktop searches. Several other researchers [5,6] have also focused on the needs of the users, rather than on the mechanisms involved.

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 [7]. 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. Significant work has been undertaken to improve mobile search using SMS text messages [8]. In iTrust, an SMS request (query) consists of a list of keywords, which are typically less than 140 bytes. An SMS response simply returns the requested information if it is small (less than 140 bytes). If the requested information or document is larger than 140 bytes, it is fragmented into multi-part SMS messages. Alternatively, the SMS response can return a URL, which is typically less than 140 bytes.

Finally, the short message size and transmission frequency of SMS messages accustoms users to utilizing the service for almost real-time momentary or fleeting communication. After an hour, or even several minutes, most SMS messages are no longer important to the user; in many cases even important messages are meaningless without surrounding context such as time, circumstances, or information not recorded directly by the mobile device. For this reason, the temporal integrity of an SMS query result is relevant only if a search hit is returned relatively quickly (within minutes) otherwise the information is meaningless without context provided by the user.

3 Design of the iTrust Search and Retrieval System

The iTrust search and retrieval system involves no centralized mechanisms and no centralized control. We call the nodes that participate in an iTrust network 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 shown in Figure 2(a), some nodes, the *source nodes*, produce information, and make that information available to other participating nodes. The source nodes produce metadata that describes their informa-

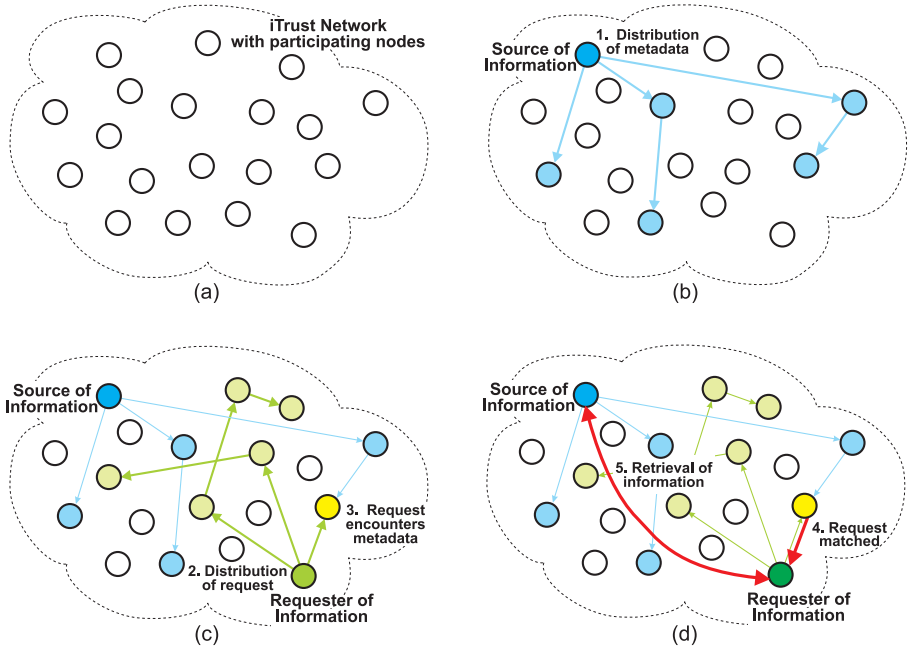


Fig. 2 (a) An iTrust network with participating nodes. (b) A source node distributes metadata to randomly selected nodes. (c) A requesting node distributes its request to randomly selected nodes. (d) A node matches the metadata and the request, and reports the match to the requesting node, which then retrieves the information from the source node.

tion, and distribute that metadata to a subset of participating nodes that are chosen at random, as shown in Figure 2(b). 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 their requests to a subset of the participating nodes that are chosen at random, as shown in Figure 2(c).

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 2(d).

Distribution of the metadata and the requests to relatively few nodes suffices to achieve a high probability of a match. 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 8. Moreover, it is not easy for a small group of nodes to subvert the iTrust mechanisms to censor, filter, or subvert information.

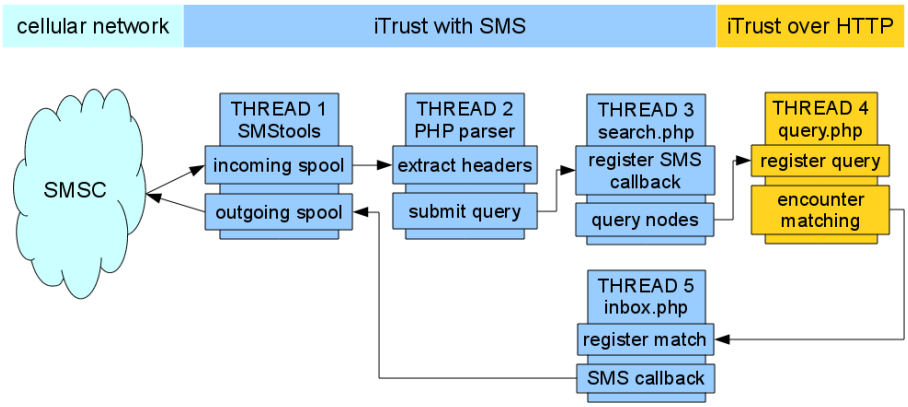


Fig. 3 iTrust with SMS, showing the cellular network, the iTrust with SMS APIs, and the iTrust over HTTP APIs.

4 Implementation of iTrust with SMS

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.) Additionally, a single node may have any number of such inexpensive cellular modems connected thus creating multiple points of communication between the SMS-enabled querying device and the iTrust with SMS node (we later describe the open-source software utilized to service the modems). The result is that an existing iTrust network can remain unchanged; only the iTrust SMS-HTTP bridge node must be software updated.

Figure 3 provides a system block diagram that shows the communication path taken by SMS request and response messages. Specifically, it shows the three main parts of iTrust with SMS, showing the cellular network, the iTrust with SMS interface, and the iTrust over HTTP interface. 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.

4.1 Cellular Network

For the purposes of this discussion, the cellular network is modeled simply by the Short Message Service Center (SMSC), which the mobile phone service providers use to relay SMS messages. In Section 5, 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 and multiple SMSCs that handle SMS relaying. iTrust does not require any service provider agreements or integration with existing mobile networks; it simply uses a mobile phone number like any mobile device seen by the SMSC.

4.2 iTrust with SMS

First and foremost, the iTrust with SMS implementation is an extension of the iTrust over HTTP implementation; SMS capabilities are added to the API and the iTrust over 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 node allows an SMS-enabled mobile phone 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 node, 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 node 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 SMStools to spool both incoming and outgoing SMS messages.

In iTrust with SMS, 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 query-

ing 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 mobile 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 the functionality of pure iTrust over HTTP nodes. Along with query text and timestamp information, the sender's callback phone number is registered to enable results sent to the iTrust 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 mobile phone number itself is not included in the query; only the iTrust SMS-HTTP bridge node is aware of the mobile 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 duplicate relayed queries are ignored and, then, an encounter (match), if any, causes a response message 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 routine in the iTrust with SMS part. It has the dual function of pulling the source information and packaging that information appropriately before passing on 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 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 iTrust 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 the query has been seen previously, processing stops as repeating an old query is not useful. If the query has not been seen previously, the query text is compared against a database consisting of metadata and URLs of the corresponding resources, using the *encounter matching* routine in THREAD 4. 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.

4.4.1 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 mobile phone number and text query. The mobile 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).

4.4.2 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 an SMS message, sent from the iTrust SMS-HTTP bridge node.

4.5 API Function Call Swapping and Race Conditions

In Figure 3, 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 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 phone number before querying another iTrust node. Otherwise, an iTrust node that is not SMS-capable might respond to a query before the callback phone number is registered. In this case, the particular response is not relayed to the mobile phone; however, future responses, that arrive after the SMS callback phone number has been registered, will be relayed.

Simply swapping the order to that shown in Figure 3 prevents a race condition from occurring.

5 iTrust with SMS User Interface

The addition of iTrust with SMS to iTrust over HTTP requires 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. Whereas 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 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, we consider a 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 in which a user searches for meeting information around *Tahrir Square* in Cairo, Egypt.

5.1 iTrust with SMS Using the Generic Instant Messaging 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 simply SMS messages that are sent to the mobile 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 an SMS-capable mobile phone; the SMS message may be sent to a

Fig. 4 iTrust with SMS, using the generic Instant Messaging interface.

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 interface.

Figure 4 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 in the Android platform (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 mobile 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 user without reference to the URL, document file name, or address of the source node of the document. This presentation is consistent with the iTrust with SMS functionality, which requires that the iTrust SMS-HTTP bridge node itself must fetch the document, package it in an SMS-compatible format, and send back the 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 an 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. Figure 5 shows 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 mobile phone numbers to iTrust SMS-HTTP bridge nodes, and a framework for handling non-textual result data.

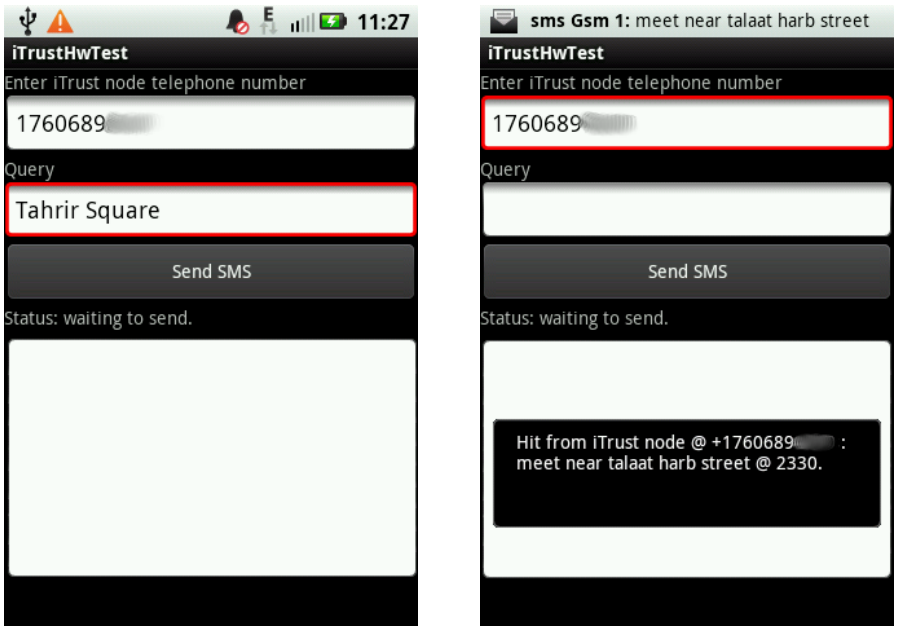
Figure 5(a) shows the entry of a query into a text editing area that is similar to that in the iTrust over HTTP search interface. Above the query is the pre-entered mobile phone number of the iTrust SMS-HTTP bridge node. Although this interface is a minimal enhancement to the generic SMS interface, the 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 enter another search query.

Figure 5(b) shows the result data returned from the iTrust SMS-HTTP bridge node; the result is the same as the result for the generic SMS interface. 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 passed on to the Android platform, presumably to be opened by a pdf reader application available on the mobile phone. In this case, the user would need a separate reader application appropriate to the file type. The iTrust system searches and retrieves all files, regardless of format (as long as the metadata are properly generated); however, the user is responsible for appropriate decoding.

6 iTrust over SMS Protocol and Implementation

The iTrust over SMS protocol allows any SMS-capable mobile device to communicate with any other such device in the iTrust over SMS network, regardless of hardware or software platform. The iTrust over SMS protocol is described below in terms of: (1) message formats, (2) metadata distribution message types and examples, and (3) search and retrieval message types and examples.

Although the iTrust over SMS protocol is platform agnostic, the first implementation of the protocol was developed in conjunction with a Java-based



(a) (b)

Fig. 5 iTrust with SMS with the custom Android interface: (a) Searching for information and (b) Viewing a hit.

iTrust over SMS implementation for the Android platform. Therefore, for completeness, first we briefly describe the iTrust over SMS implementation on Android, and then we describe how that implementation follows the iTrust over SMS protocol, before we describe the protocol itself.

6.1 iTrust over SMS Implementation on Android

Figure 6 shows the Android-based implementation of iTrust over SMS, which comprises the user interface, the iTrust over SMS API, and the mobile (cellular) network. The user interface might be an Instant Messaging application, document sharing application, or (in our case) a test application for experimentation with the iTrust over SMS protocol. The mobile network in Figure 6 is identical to that in Figure 3; it can be regarded as interconnected SMSC entities that transport messages between mobile peers.

The iTrust over SMS API consists of five components, two components handle the basic input and output of messages and three components handle the primary iTrust over SMS functionalities. The SMS receiver component handles input, and the SMS transmitter component handles output. Previously, we described the individual functions of iTrust with SMS, by tracing how a

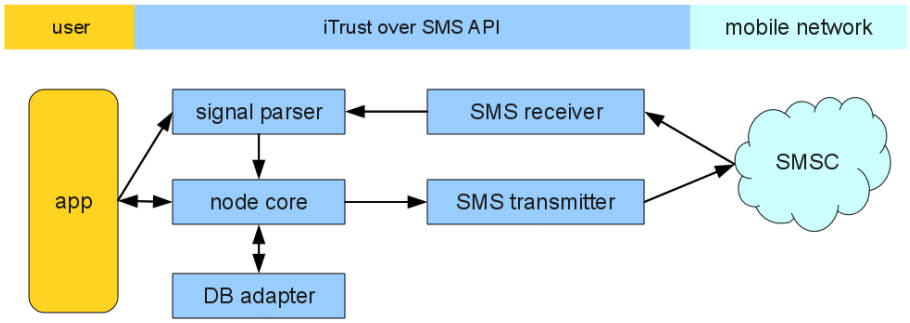


Fig. 6 The iTrust over SMS API and components, along with the user interface and the mobile network.

message flows between threads. Here, we describe how an incoming message is acted on, and an outgoing message is generated, in iTrust over SMS.

A message is received by the SMS receiver component (based on an Android *BroadcastReceiver* intent), which passes the message on to the signal parser component. Correspondingly, a message is created by the node core component, which passes the message on to the SMS transmitter component. The signal parser component decodes text messages, and the node core component acts on or responds to messages (making encounters, adding nodes to the iTrust membership, relaying queries, distributing metadata, *etc*). If a message exceeds the SMS limit of 140 octets, an Android utility splits the message into chunks and reassembles multi-part SMS messages on arrival (through the use of the SMS user data header). Thus, we say that the signal parser *reads* the iTrust over SMS protocol, and the node core (which might have to send a response message) *writes* the protocol. The database adapter component handles the bookkeeping tasks required of the node core component by the use of various SQLite database tables.

An important Android-related concern deals with the transmission of SMS text messages. Unfortunately, there is a long-standing bug in Android, which prevents the proper transmission of certain characters in SMS text messages. Specifically, the characters *[ ]* { } cannot be correctly sent in an SMS text message, because the GSM alphabet table is incorrectly set by Android. According to the 3G TS 23.038 version 3.3.0 technical specification, the SMS packing scheme (specifically, the packing of 7-bit characters) allows an extended 7-bit alphabet to be used. The GSM 7-bit alphabet extension table includes the characters *[ ]* { }; however, because the table cannot be correctly enabled by Android, the characters are effectively unavailable. We produced a work-around that forces the characters *[ ]* { } to be transformed into the characters *( )* <>; the latter characters are not in the extension table, and so the default table can be used instead. Because Android supports the default table, the message is sent correctly. Thus, the JSON string representation of the metadata is transformed from valid JSON to quasi-JSON and placed in

the outgoing message string ready for the SMS transmitter component to send the data. On arrival of the message, the receiving node must perform the reverse transformation (*i.e.*, replace all *( )* <> characters by *[ ]* { } characters), before processing the JSON data.

With this explanation of the Android implementation and its connection with the iTrust over SMS protocol, and the quasi-JSON work-around for Android, we now delve into the protocol itself.

6.2 Message Formats and Types of the iTrust over SMS Protocol

Figure 7 illustrates the two basic message formats of the iTrust over SMS protocol; the primary distinction is that one format accommodates three parameters and the other format accommodates two parameters. The figure depicts the body of a text message contained in an SMS text message, in this case, a single text identifier followed by several text parameters, each separated by the @ character.

Three parameter SMS message

Identifier	Delimiter	Parameter 1	Delimiter	Parameter 2	Delimiter	Parameter 3
	@		@		@	

Two parameter SMS message

Identifier	Delimiter	Parameter 1	Delimiter	Parameter 2
	@		@	

Fig. 7 The message formats of the iTrust over SMS protocol.

Figure 8 shows the seven different message types of the iTrust over SMS protocol; the first four are used for search and retrieval and the remaining three are used for metadata distribution. The leftmost column lists the seven message types; lower-case plain text represents string literals, and italicized angle-bracketed text represents placeholders for variable text. Cells labeled *unused* are reserved for future use. The search and retrieval functions are separated into two message types: messages with identifier *itq* search or query for information, and messages with identifier *itr* return or retrieve information. The remaining three metadata distribution messages have the identifier *itm*, and use only two parameters.

An important design consideration for the iTrust over SMS protocol is that a message should be relatively easy for humans to understand, even if doing so increases the complexity of the signal parser. With the three message identifiers

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>*	

Fig. 8 The message types of the iTrust over SMS protocol.

itq, *itr*, and *itm*, a human can easily understand whether a particular iTrust over SMS protocol message is a query, reply, or metadata message.

The signal parser processes each incoming message; messages with identifiers *itq* or *itr* are parsed for three parameters, and messages with identifier *itm* are parsed for two parameters. Programmatically, each message is considered an enumeration, and processing is switched on a case-by-case basis. Parameters are parsed left to right, which creates string tokens delimited by the character @; parsing ceases after the final delimiter is found. Therefore, messages with three parameters are fully processed after the first three leftmost occurrences of @, and messages with two parameters are fully processed after the first two leftmost occurrences of @. Doing so allows the final parameter to use the character @ any number of times without breaking the parsing rules (*i.e.*, it is not necessary to escape @ in the final parameter). For example, the messages *itq@aaa@bbb@ccc* and *itq@aaa@bbb@ccc@ddd* are both valid messages; the former has final parameter *ccc*, and the latter has final parameter *ccc@ddd*. Two parameter messages also follow the same pattern. For example, the messages *itm@111@222* and *itm@333@444@555* are both valid messages; the former has final parameter *222*, and the latter has final parameter *444@555*. The rationale behind this particular way of parsing will become evident when the seven message types are described.

6.3 Metadata Distribution for iTrust over SMS

Although searching for and retrieving information in the iTrust network constitutes the bulk of the time spent by users or applications, first the metadata must be distributed before encounters for searched information can occur. Below, we describe the three messages involved in distributing metadata, graphically show how the messages are sent between iTrust nodes, and finally present an example with actual SMS text messages.

6.3.1 Metadata Distribution Message Types

NOTIFY_METADATA. The NOTIFY_METADATA message notifies a node in the iTrust network that metadata are ready to be read. The message can be sent in two different ways: by the source node of the metadata or relayed

through another node. If the message is sent by the source node of the metadata, the source node creates the two parameter message with the source phone number and expiration date. The source phone number is the mobile phone number of the source node that stores the resource described by the metadata, and the expiration date is a Unix timestamp (number of seconds, in the Unix epoch) after which reading the metadata is no longer useful. For example, if the source node holds data for meeting information, the meeting location and time are not useful the day after the meeting. If the message has been relayed, a node that receives the NOTIFY_METADATA message saves the source phone number and expiration date and then relays the message to some other node in the iTrust membership. There is no need for the relaying node to modify the message; all necessary information has already been placed in the message by the source node. When retrieving information, a node may decide to prioritize retrieval based on source phone number or expiration date; the reasons for prioritization depend on the user application. In particular, the user application might decide to ignore the expiration date and retrieve metadata whenever it is convenient; it is neither expected nor required that a receiving node immediately act on a NOTIFY_METADATA message.

REQUEST_METADATA. When a node wants to receive metadata, it sends a REQUEST_METADATA message to the source node that holds the desired resource information. The first parameter is filled with the string literal *pull*; the second parameter is unused. On receiving a REQUEST_METADATA message, the source node immediately creates and sends back a SEND_METADATA message.

SEND_METADATA. When a node receives a request for metadata, it immediately creates and populates the SEND_METADATA message. The first parameter is filled with the string literal *push*, and the second parameter is filled with the quasi-JSON encoded metadata resource/keyword pairs. The quasi-JSON string should have any characters not in the GSM default alphabet table removed to avoid conflicts (to work around the Android limitation previously discussed). As explained earlier, parsing the two parameter message type stops after the first two leftmost @ symbols are found; therefore, the metadata itself does not need to be re-evaluated for @ string escapes. On receiving a SEND_METADATA message, a node decodes the second parameter and inserts the resource/keyword pairs into its database.

If there is a third node (or other previous node in the relay chain), the participation of the relay node ends immediately after the NOTIFY_METADATA message is relayed.

6.3.2 Examples of Metadata Distribution

Figure 9 shows example diagrams that depict the flow of messages between nodes during the distribution of metadata in the iTrust over SMS network.

Parts A and B represent independent interactions; messages in the two parts do *not* chronologically precede or follow one another. Node S is the source node that has resources locally stored; nodes Z and Y are other nodes that receive metadata. The lines and arrows show the directions and the destinations of the messages; each line is labeled with the message sent. The numbers preceding the messages signify the order in which the messages are sent; the numbers are *not* part of the messages sent.

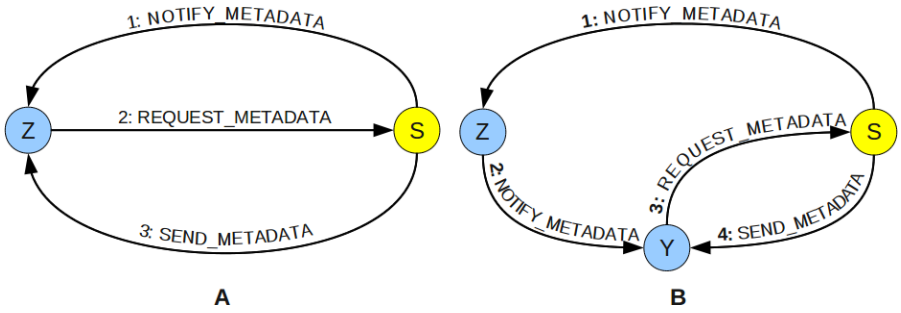


Fig. 9 Metadata distribution message flow for iTrust over SMS.

Part A: Metadata Distribution between Two Nodes. Node S sends a NOTIFY_METADATA message to node Z informing Z that metadata are ready for Z to read at Z's convenience. Z sends a REQUEST_METADATA message to S requesting metadata to be sent immediately. S creates the metadata and sends it to Z in the SEND_METADATA message.

Part B: Metadata Distribution between Three Nodes. Node S sends a NOTIFY_METADATA message to node Z as indicated by message *1*. Z relays the message to node Y as indicated by message *2*. At Y's convenience, Y sends a REQUEST_METADATA message to S requesting metadata to be sent immediately. S creates the metadata, and sends it to Y in the SEND_METADATA message.

6.3.3 Example of SMS Text Messages for Metadata Distribution

Figure 10 shows an example of how SMS text is transmitted between nodes in the iTrust membership during metadata distribution. The nodes and other descriptions are the same as those in Figure 9; messages are numbered in the order in which the messages are sent.

Node S sends message *1* to node Z; the first parameter is filled with S's mobile phone number. Z sends message *2* to S; the first parameter is filled with the string literal *pull*. S sends message *3* to Z; the first parameter is filled with *push*, and the second parameter is filled with quasi-JSON metadata.

Note that, if message *1* had been relayed by an intermediate node, *Z* would know to call *S* at *15559988776*, because that phone number was included in message *1*.

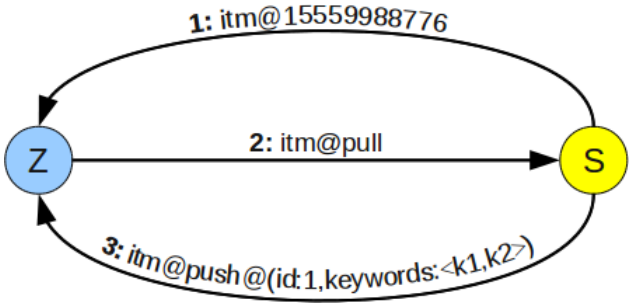


Fig. 10 SMS text message example of metadata distribution for iTrust over SMS.

6.4 Search and Retrieval for iTrust over SMS

Search and retrieval of resources involves four types of iTrust over SMS messages: two query messages and two response/retrieval messages. Below, we describe these four types of messages, give an example of message passing between nodes, and finally present an example of actual SMS text messages sent for search and retrieval of resources.

6.4.1 Search and Retrieval Message Types

SEND_QUERY. The SEND_QUERY message is used to query a node in the iTrust network for resources. The message contains three parameters: call number, query id, and query text. The call number is the mobile phone number of the node that is issuing the query. The query id is any text string that nodes in the iTrust membership use to track the query. The same query id is used for all four iTrust over SMS messages pertaining to the search request; in effect, it is a global identifier. The query text should be checked, by the user application, to ensure that it is within the GSM default alphabet table.

If a node is originating the query, it creates these three parameters and then sends the message. If a node is relaying the query, it relays the message without modification; it can use the query id to prevent relaying the same message more than once (to prevent network flooding). On receiving a SEND_QUERY message, a node immediately checks whether an encounter has occurred by comparing the query text against its available resources. If an encounter has indeed occurred, it sends a NOTIFY_MATCH message; otherwise, it takes no further action.

NOTIFY_MATCH. When a node has an encounter, it responds to the original querying node with a NOTIFY_MATCH message. The message is sent directly

to the call number in the first parameter of the SEND_QUERY message; it is not sent to the node that relayed the query. The NOTIFY_MATCH message contains three parameters: the source phone number, the query id, and the resource id. The query id is the same as that in the SEND_QUERY message; again, it is a global identifier for the query and may be used by the application for various purposes. For example, an application might ignore a query that it did not originate, to protect against rogue nodes that send spurious NOTIFY_MATCH messages.

If the resource is stored locally, the source phone number is the mobile phone number of the node at wich the encounter occurred (*i.e.*, the node that received the SEND_QUERY message and is about to send the NOTIFY_MATCH message), and the resource id is from the local resource table. If the resource is *not* stored locally, the source phone number is the mobile phone number of the node where the resource *is* stored and the resource id is from the resource table of that node.

Receiving the NOTIFY_MATCH message requires relatively little processing. Because the node that sent the NOTIFY_MATCH message did the processing required to find the node on which the resource is located and set the message parameters accordingly, the only required action is to save the values for further processing before discarding the message. The user or application can decide when to retrieve the message at the source phone number; retrieval of the document is not mandatory and is done at the convenience of the user or application, using the REQUEST_RESOURCE message.

REQUEST_RESOURCE. When a node wants to retrieve a particular resource, it directly contacts the node that stores the resource, using the REQUEST_RESOURCE message. The message contains three parameters: the string literal *now*, the query id, and the resource id of the stored resource on the receiving node. Although the query id is not strictly needed in this case (it's possible that the associated SEND_QUERY message was never relayed to the receiving node), it is still sent for possible use by the application. On receiving a REQUEST_RESOURCE message, a node immediately looks up the resource using the resource id and sends it using the SEND_RESOURCE message. If the resource id does not exist in its table, the node ignores the message and stops processing.

SEND_RESOURCE. When a node receives a REQUEST_RESOURCE message, it immediately gets the resource data and packages it for transmission in the SEND_RESOURCE message. The SEND_RESOURCE message has three parameters: the string literal *data*, the query id, and the data itself. Again, the query id is sent only for optional tracking by the user application that interfaces with iTrust over SMS. Transmitted data in the third parameter of the SEND_RESOURCE message can be in any format suitable for the application as long as it fits within the GSM default alphabet table (again due to the Android bug). The iTrust over SMS API provides several convenience functions for inserting (extracting) plain text data into (from) the message, which make

sending (receiving) plain text trivially simple. To send (receive) custom data apart from plain text, the user application simply needs to escape (unescape) that custom data.

Note that only the original querying node is involved with each of the four message types. Intermediate nodes may relay queries or send notifications of a match, but their involvement ends immediately thereafter.

6.4.2 Examples of Search and Retrieval

Figure 11 shows example diagrams that depict the flow of messages between nodes during the search and retrieval of resources in the iTrust over SMS network. Parts A, B, and C represent independent interactions; messages in the three parts do *not* chronologically follow or precede one another. Node S is the source node that has the resources locally stored; node Q is the querying node that sends the original search query; and nodes Z and Y are other nodes in the iTrust network. Again, the lines and arrows show the directions and destinations of the messages; the numbers preceding the messages signify the order in which the messages are sent.

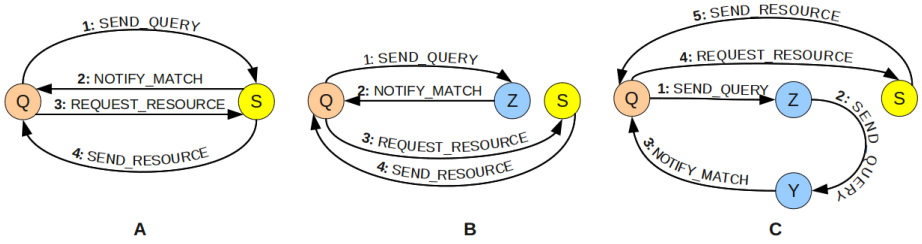


Fig. 11 Search and retrieval message flow for iTrust over SMS.

Part A: Search and Retrieval between Two Nodes. Node Q sends a SEND_QUERY message to node S. S has an encounter, and responds to Q with a NOTIFY_MATCH message. When it is convenient, Q sends a REQUEST_RESOURCE message to S. On receiving the REQUEST_RESOURCE message, S sends the resource to Q in the SEND_RESOURCE message.

Part B: An Intermediate Node Has an Encounter. At some prior time, node S distributed metadata to node Z. Node Q sends a SEND_QUERY message to Z; Z immediately has an encounter as a result of Q's query and the metadata distributed by S. Z sends a NOTIFY_MATCH message to Q. When it decides, Q sends a REQUEST_RESOURCE message to S. S sends the resource to Q in the SEND_RESOURCE message.

Part C: A Search Query Is Relayed. At some prior time, node S distributed metadata to node Y. Node Q sends a SEND_QUERY message to node Z as shown by message *1*. Z does not have a match but relays the SEND_QUERY

message to Y as shown by message *2*. Y immediately has an encounter between Q's query and the metadata distributed by S, and sends a NOTIFY_MATCH message to Q. At its convenience, Q sends a REQUEST_RESOURCE message to S. S sends the resource to Q in the SEND_RESOURCE message.

6.4.3 Example of SMS Text Messages for Search and Retrieval

Figure 12 shows an example of how SMS text is transmitted between nodes in the iTrust membership during a typical search and retrieval interaction. The nodes and other descriptions are the same as those for Figure 11; messages are identified by the order in which they are sent.

Node Q sends message *1* to node S with three parameters: caller phone number *15551234567*, query id *r4nd0m1d*, and query *tahrir square*. Immediately, S has an encounter and knows to call back node Q at *15551234567* with message *2*, which contains the source phone number *15550011223*, query id, and resource id *456*. When convenient, Q responds to S by sending message 3, which contains the query id and resource id *456* of the resource that it wants to retrieve. S immediately responds to Q by sending message 4, which contains the query id and the resource data *meet near talaat harb street*.

Note that, if message *1* had been relayed, any node that had an encounter would know to contact node Q at *15551234567*, because the phone number is included in the message. Furthermore, if the encounter had occurred on metadata held by another node, the source phone number *15550011223* in message *2* would tell Q which node to contact to retrieve the resource.

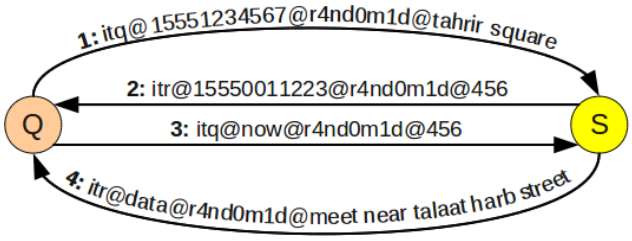
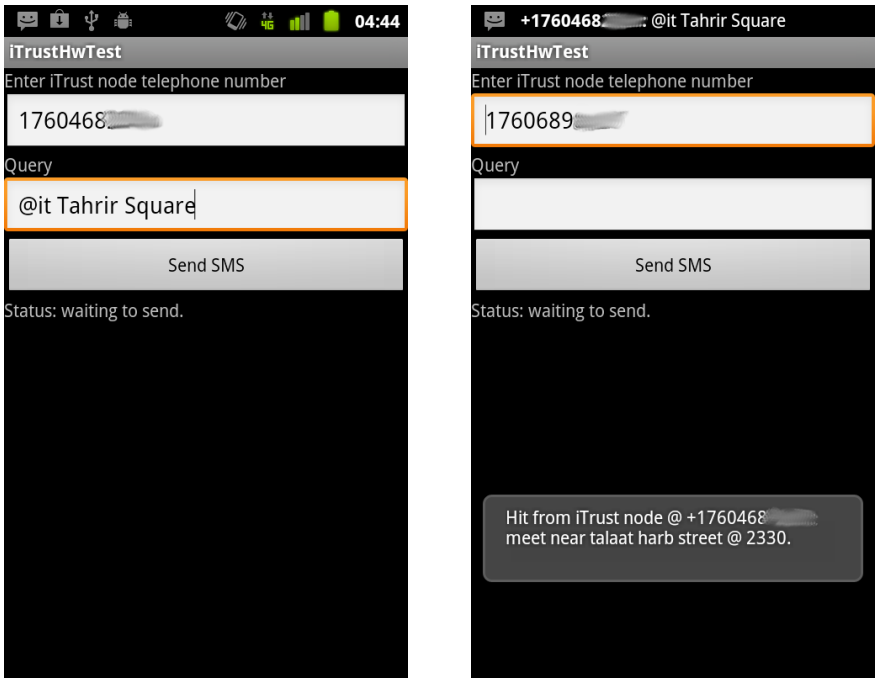


Fig. 12 SMS text message example of search and retrieval for iTrust over SMS.

7 iTrust over SMS User Interface

The custom Android user interface shown in Figure 5 for iTrust with SMS was re-purposed for iTrust over SMS. The custom Android user interface for iTrust over SMS is shown in Figure 13.

From the end user's perspective of searching for and retrieving information, the interfaces are the same. The underlying difference is that instead of communication between a mobile phone and an iTrust SMS-HTTP bridge



(a) (b)

Fig. 13 iTrust over SMS with the custom Android interface: (a) Searching for information and (b) Viewing a hit.

node, communication occurs directly (peer-to-peer) between mobile phones in the iTrust over SMS network. Appropriate defaults were chosen for functions inherent to iTrust over SMS but not configurable in the custom user interface originally developed for iTrust with SMS. For example, metadata distribution is done automatically on application startup for iTrust over SMS (because iTrust with SMS has no need for this function). Additionally, because Android uses the *BroadcastReceiver* intent to act as an event handler for incoming messages, as explained earlier, the custom Android interface does not need to register another event handler. The event handler written for the iTrust over SMS API suffices.

In addition to the above simple user interface for searching and retrieving resources, several use cases were considered when developing an Android app specifically targeted for ease of use for the average non technical user. The complete use case details and usage of the app are found in [34] however we briefly review them here. We identify four main types of users from a sporadic document searcher to avid document searcher and detail, with use case scenarios and app descriptions, how the app accomidates each usage pattern. In particular, the app is designed as simple as possible and resembles the built-in

search functionality present on the Android platform; other functions such as query managing and document managing are included for those users wanting greater control of the document retrieval process. Technical details, such as message identifiers etc., are completely hidden from the smartphone user. Critically, the ease of use for smart phone users does not inhibit iTrust over SMS functionality for low-cost *dumb* phone users who are already accustomed to quickly texting messages using a thumbpad. While the average smart phone user demands the convenience of an app, the average dumb phone user willingly thumb types SMS messages for enhanced mobile device functionality such as P2P document searching.

8 Performance 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 in the iTrust network have the same membership. Moreover, we assume that communication is reliable and timely, and that all of the participating 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 requests are sent directly to the nodes without relaying, and that the nodes do not delay in requesting metadata or reporting matches.

The 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 operational (*i.e.*, $1 - x$ is the proportion of non-operational nodes)
- 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.

8.1 Probability of a Match

First, we consider the probability that, for a given request, a match (encounter) occurs, *i.e.*, that one or more nodes have a match for that request. The results for the probability of a match hold for iTrust over HTTP, iTrust with SMS, and iTrust over SMS.

8.1.1 Analysis

Our performance analysis of iTrust is based on the hypergeometric distribution [9], which describes the number of successes in a sequence of random draws from a finite population *without* replacement. In iTrust, the probability

of *exactly* k matches follows the hypergeometric distribution with parameters n, x, m and r, and is given by:

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

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

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

$$P(0) = \frac{(\frac{n-mx}{r}\frac{n-mx-1}{r-1}\cdots\frac{n-mx-r+1}{1})}{(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1})} \quad (2)$$

for $mx + r \leq n$.

Consequently, the probability of a match (*i.e.*, *one or more* matches) is given by:

$$P(k \geq 1) = 1 - \frac{(\frac{n-mx}{r}\frac{n-mx-1}{r-1}\cdots\frac{n-mx-r+1}{1})}{(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1})} \quad (3)$$

for $mx + r \leq n$. If $mx + r > n$, then $P(k \geq 1) = 1$.

Figures 14, 15, and 16 show the probability of a match obtained from Equation (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 probability of a match increases and approaches 1, as $m = r$ increases.

8.1.2 Simulation

Using our implementation of iTrust, we performed simulation experiments to validate the analytical results for the probability of a match obtained from Equation (3).

Before we ran our simulation program, we deleted all resources and data from the node. Next, the program adds the nodes to the membership. Then, 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 the source 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. Next, 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.

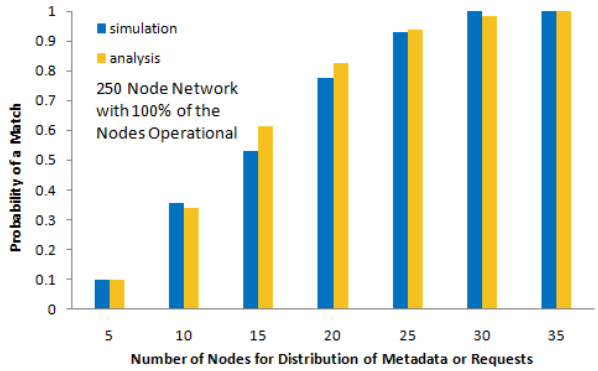


Fig. 14 Match probability vs. number of nodes for distribution of metadata or requests in an iTrust network with 250 nodes where 100% of the nodes are operational.

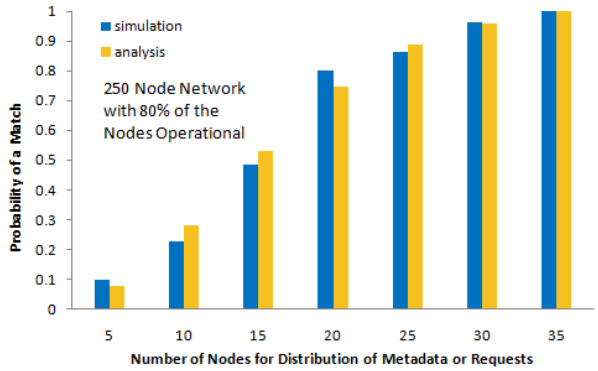


Fig. 15 Match probability vs. number of nodes for distribution of metadata or requests in an iTrust network with 250 nodes where 80% of the nodes are operational.

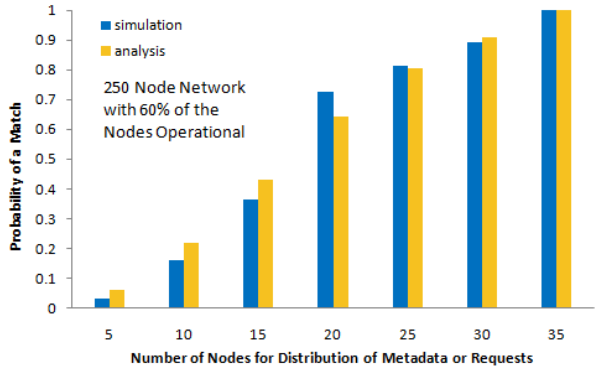


Fig. 16 Match probability vs. number of nodes for distribution of metadata or requests in an iTrust network with 250 nodes where 60% of the nodes are operational.

Figures 14, 15, and 16 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 Equation (3). As these results indicate, iTrust retains significant utility even in the case where a substantial proportion of the nodes are non-operational.

8.2 Number of Messages to Achieve a Match

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

8.2.1 Analysis

For iTrust over HTTP and iTrust over SMS, the mean number Y of messages required to achieve a match is given by:

$$Y = r + \sum_{k=1}^{\min\{mx,r\}} kP(k) \tag{4}$$

The term r on the right side of Equation (4) represents r requests from the requesting node to other participating nodes. The sum represents the number k of matches (response messages), weighted by the probability $P(k)$ of k matches obtained from Equation (1).

For iTrust with SMS, the mean number Y of messages required to achieve a match is given by:

$$Y = 2 + r + \sum_{k=1}^{\min\{mx,r\}} kP(k) \tag{5}$$

The term 2 on the right side of Equation (5) represents: 1 request message from the mobile phone to the iTrust SMS-HTTP bridge node and 1 match response message from the iTrust SMS-HTTP bridge node to the mobile phone. The term r on the right side of the equation represents r requests from the iTrust SMS-HTTP bridge node to iTrust over HTTP nodes. The sum is the same as that in Equation (4) and represents messages sent from the matching iTrust over HTTP nodes to the iTrust SMS-HTTP bridge node.

Figures 17, 18, and 19 show the number of messages obtained from Equations (1) and (5) 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 of a match increases (and as $m = r$ increases), but is bounded by $2+2r$ because, in Equation (5), $\sum_{k=1}^{\min\{mx,r\}} kP(k) \leq \sum_{k=1}^{r} kP(k) \leq r \sum_{k=1}^{r} P(k) \leq r$.

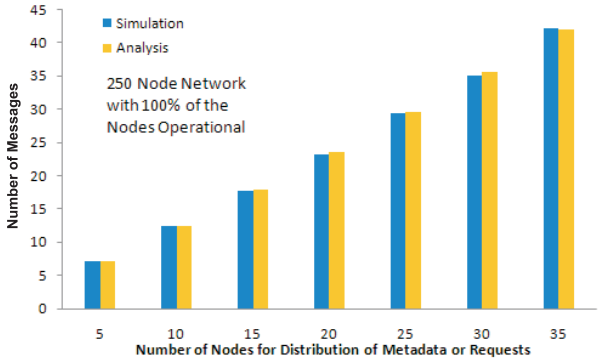


Fig. 17 Number of messages vs. number of nodes for distribution of metadata or requests in an iTrust with SMS network with 250 nodes where 100% of the nodes are operational.

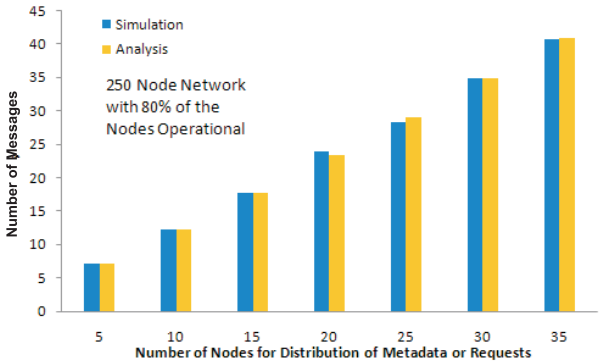


Fig. 18 Number of messages vs. number of nodes for distribution of metadata or requests in an iTrust with SMS network with 250 nodes where 80% of the nodes are operational.

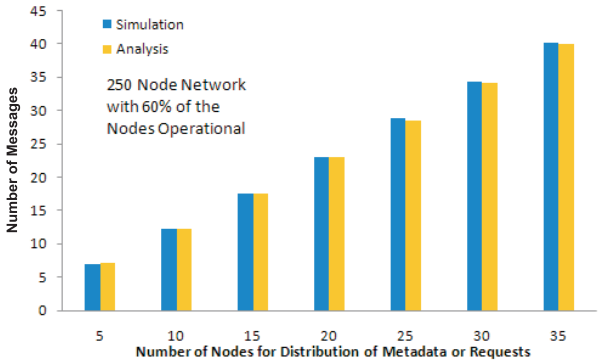


Fig. 19 Number of messages vs. number of nodes for distribution of metadata or requests in an iTrust with SMS network with 250 nodes where 60% of the nodes are operational.

8.2.2 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 (1) and (5). The simulation experiments were performed as described previously in Section 8.1.2.

Figures 17, 18, and 19 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 (1) and (5).

Figures 14, 15, and 16 and Figures 17, 18, and 19 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.

9 Related Work

Existing mobile search services include AOL Mobile [10], Google SMS [11], Windows Live Mobile [12], and Yahoo! OneSearch [13]. Those services provide Web search on mobile devices, and use conventional centralized Web search engines. They provide a limited set of pre-defined topics, and use either special keywords within a search query (*e.g.*, "directions" to obtain directions) or a specialized parser to determine the intended topic (*e.g.*, "INTC" for a stock quote). For queries related to arbitrary topics, the results obtained are sometimes not meaningful or not consistent. Moreover, the centralized search engines on which those systems depend are subject to censorship, filtering, and subversion.

The SMSFind system [14,15] also utilizes a conventional centralized search engine at the back-end. However, 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 iTrust currently does not do but which might be valuable for a future version of iTrust.

Bender *et al.* [16] 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 [17], Risson and Moors [18], and Tsoumakos and Roussopoulos [19] 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 mes-

sages over existing links. The iTrust system uses the unstructured approach, which is less vulnerable to manipulation.

Gnutella [20], one of the first unstructured networks, uses flooding of requests to find information. Extensions of Gnutella involve supernodes [21], which improve efficiency but incur some of the trust risks of centralized strategies, and biased random walks with one-hop data replication [22], which use randomization and replication like iTrust does. Freenet [23] is more sophisticated and efficient than Gnutella, because it learns from previous requests. In Freenet, nodes that successfully respond to requests receive more metadata and more requests. Thus, it is easy for a group of untrustworthy nodes to conspire together to gather most of the searches into their group, making Freenet vulnerable to subversion.

The Mobile Agent Peer-To-Peer (MAP2P) system [24] 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.

The Distributed Mobile Search Service [25] 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 corresponding URLs first, rather than on receipt of the query results and, thus, has a lower message cost.

The 7DS system [26] 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, iTrust does not use multicasting or flooding, which are too expensive in message cost.

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.* [27] investigate the integration of social network search with Web search, and conclude that such integration can lead to more timely and efficient search experiences. Tiago *et al.* [28] describe a system for mobile search in social networks based on the Drupal content site management system, using the network of social links formed from the mobile phone's address book. Yang *et al.* [29] propose a search mechanism for unstructured peer-to-peer networks, based on special 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.

Several peer-to-peer information sharing systems are concerned with trust. Quasar [30] 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. Quasar is

a publish-subscribe system which uses a structured overlay approach to route information between nodes, iTrust is neither a publish-subscribe system and does not use any structured overlay to route messages. Furthermore, Quasar uses highly aggregated routing vectors and directed walks while iTrust does not use routing vectors nor is it geared for anycast-like directed walks. OneSwarm [31] is a peer-to-peer system that allows information to be shared either publicly or anonymously, using a combination of trusted and untrusted peers. Safebook [32] is a social network that preserves anonymity, by communicating information through intermediary nodes. Both OneSwarm and Safebook aim to protect the users' privacy, which iTrust does not aim to do. Rather, the trust objective of iTrust is to support the free flow of information and to prevent censorship, filtering, and subversion of information. However, some of the ideas of those systems might be useful for a future version of iTrust.

10 Conclusions and Future Work

The iTrust with SMS system enables SMS-capable mobile phones to communicate with iTrust SMS-HTTP bridge nodes that act as relays to iTrust over HTTP nodes for information search and retrieval in the iTrust network. Thus, an SMS-capable mobile phone can access information on any number of interconnected iTrust over HTTP nodes, and the iTrust over HTTP nodes can be queried from any SMS-capable mobile phone for search and retrieval of information. An Android mobile phone application provides a custom interface to facilitate search and retrieval over the iTrust network.

The iTrust over SMS system enables SMS-capable mobile phones to communicate directly over the cellular telephony network to distribute metadata and to search for and retrieve information. Information stored locally on any iTrust over SMS mobile device can be sent directly to another such mobile device by Instant Messages; Internet access is not required. In the iTrust over SMS network, mobile devices can send information to mobile devices using different platforms, such as Android, iOS, *etc.*, as long as each platform implements the iTrust over SMS protocol. As described in this paper, the iTrust over SMS protocol is implemented as an Android application, and the custom user interface for iTrust with SMS was re-purposed for iTrust over SMS.

Although the iTrust SMS-HTTP bridge nodes provide search and retrieval access to the iTrust network for SMS-capable mobile phones, an 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, mobile phones, and they are hard to read on the small screens of mobile phones. Of importance to many mobile phone users is the ability to upload and download images, video, and audio files directly from their mobile phones. In the future, we plan to develop an iTrust over SMS application to handle multimedia files in addition to the text files it currently handles.

Currently, we are developing a ranking algorithm for iTrust. If multiple documents are found, with metadata that match the keywords in a request,

the ranking algorithm evaluates each such document, at the source node or at the requesting node, and presents the documents to the requester in an appropriate order.

We also plan to add Wi-Fi capabilities to iTrust to create a mobile ad-hoc network of local peer-to-peer iTrust nodes. In addition, we plan to investigate the use of Bluetooth, instead of Wi-Fi, to create a local mobile ad-hoc network for iTrust. Thus, iTrust mobile users will be immune from government shut-down of cellular towers, and will be fully autonomous to search and retrieve documents from peers in their local network. These additions will strengthen the availability and robustness of information search and retrieval in iTrust.

Acknowledgment

This research was supported in part by U.S. National Science Foundation grant number NSF CNS 10-16193. An earlier and shorter conference version of this paper appeared in [33].

References

1. Michel Lombera, I, Chuang, Y T, Melliar-Smith, P M, Moser, L E (2011) Trustworthy distribution and retrieval of information over HTTP and the Internet. In: 3rd International Conference on the Evolving Internet, Luxembourg City, Luxembourg (June 2011), pp 7-13
2. Sohn, T, Li, K A, Griswold, W G, Hollan, J D (2008) A diary study of mobile information needs. In: 26th ACM SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy (April 2008), pp 433–442
3. Kamvar, M, Baluja, S (2006) 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
4. Kamvar, M, Kellar, M, Patel, R, Xu, Y (2009) 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 (April 2009), pp 801–810
5. Church, K, Smyth, B (2009) Understanding the intent behind mobile information needs, In: 14th International Conference on Intelligent User Interfaces, Sanibel Island, FL (February 2009), pp 247–256
6. Larsen, J, Axhausen, K W, Urry, J (2006) Geographies of social networks: Meetings, travel and communications, Mobilities 1(2):261–283. Routledge, London, UK (July 2006)
7. Tsirulnik, G (2011) 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
8. Schusteritsch, R, Rao, S, Rodden, K (2005) Mobile search with text messages: Designing the user experience for Google SMS, In: Conference on Human Factors in Computing Systems, Portland, OR (April 2005), pp 1777–1780
9. Feller, W (1968) An Introduction to Probability Theory and Its Applications I, John Wiley & Sons, New York, NY
10. AOL Mobile, http://www.aolmobile.com
11. Google SMS, http://www.google.com/sms
12. Windows Live Mobile, http://home.mobile.live.com
13. Yahoo! OneSearch, http://mobile.yahoo.com/onesearch
14. Chen, J, Linn, B, Subramanian, L (2009) SMS-based contextual Web search. In: ACM SIGCOMM MobiHeld Workshop, Barcelona, Spain (August 2009), pp 19–24

15. Chen, J, Subramanian, L, Brewer, E (2010) SMS-based Web search for low-end mobile devices. In: 16th ACM MobiCom International Conference on Mobile Computing and Networking, Chicago, IL (September 2010), pp 125–136
16. Bender, M, Michel, S, Triantafillou, P, Weikum, G, Zimmer, C (2006) P2P content search: Give the Web back to the people. In: 5th International Workshop on Peer-to-Peer Systems, Santa Barbara, CA (February 2006)
17. Mischke, J, Stiller, B (2004) A methodology for the design of distributed search in P2P middleware. IEEE Network 18(1):30–37 (January 2004)
18. Risson, J, Moors, T (2007) 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
19. Tsoumakos, D, Roussopoulos, N (2003) A comparison of peer-to-peer search methods. In: 6th International Workshop on the Web and Databases, San Diego, CA (June 2003), pp 61–66
20. Gnutella, http://gnutella.wego.com/
21. Yang, B, Garcia-Molina, H (2002) Improving search in peer-to-peer networks. In: 22nd IEEE International Conference on Distributed Computing Systems, Vienna, Austria (July 2002), pp 5–14
22. Chawathe, Y, Ratnasamy, S, Breslau, L, Lanham N, Shenker S (2003) Making Gnutella-like P2P systems scalable, In: ACM SIGCOMM Applications Technologies, Architectures and Protocols for Computer Communications Conference, Karlsruhe, Germany (August 2003), pp 407–418
23. Clarke, I, Sandberg, O, Wiley, B, Hong, T (2001) Freenet: A distributed anonymous information storage and retrieval system. In: Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA (July 2001), Lecture Notes in Computer Science 2009, Springer, Heidelberg, pp 46–66
24. Hu, H, Thai, B, Seneviratne, A (2003) Supporting mobile devices in Gnutella file sharing network with mobile agents. In: 8th IEEE Symposium on Computers and Communications, Kemer-Antalya, Turkey (July 2003)
25. Lindemann, C, Waldhorst, O P (2002) A distributed search service for peer-to-peer file sharing in mobile applications. In: 2nd International Conference on Peer-to-Peer Computing, Linkoping, Sweden (September 2002), pp 73–80
26. Papadopouli, M, Schulzrinne, H (2001) 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 (2001), pp 117–127
27. Gummadi, K P, Mislove, A, Druschel, P (2006) Exploiting social networks for Internet search. In: 5th ACM SIGCOMM Workshop on Hot Topics in Networks, Irvine, CA (November 2006), pp 79–84
28. Tiago, P, Kotiainen, N, Vapa, M, Kokkinen, H, Nurminen, J K (2008) Mobile search – Social network search using mobile devices. In: 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV (January 2008), pp 1201–1205
29. Yang, J, Zhong, Y, Zhang, S (2003) An efficient interest-group-based search mechanism in unstructured peer-to-peer networks. In: International Conference on Computer Networks and Mobile Computing, Shanghai, China (October 2003), pp 247–252
30. Wong, B, Guha, S (2008) Quasar: A probabilistic publish-subscribe system for social networks. In: 7th International Workshop on Peer-to-Peer Systems, Tampa Bay, FL (February 2008)
31. Isdal, T, Piatek, M, Krishnamurthy, A, Anderson, T (2010) Privacy preserving P2P data sharing with OneSwarm. In: ACM SIGCOMM Special Interest Group on Data Communications Conference, New Delhi, India (September 2010), pp 111–122
32. Cutillo, L A, Molva, R, Onen, M (2011) Safebook: A distributed privacy preserving online social network, In: IEEE World of Wireless, Mobile, and Multimedia Networks Conference, Lucca, Italy (June 2011), pp 1–6
33. Michel Lombera, I, Chuang, Y T, Moser, L E, Melliar-Smith, P M (2011) Decentralized mobile search and retrieval using SMS and HTTP to support social change, In: 3rd International Conference on Mobile Computing, Applications, and Services, Los Angeles, CA (October 2011)

34. Michel Lombera, I, Moser L E, Melliar-Smith, P M, Chuang, Y T (2012) A Mobile Peer-to-Peer Search and Retrieval Service for Social Networks. In: 1st International Conference on Mobile Services, Honolulu, HI (June 2012)