iTrust: Trustworthy Information Publication, Search and Retrieval

P. M. Melliar-Smith, L. E. Moser, I. Michel Lombera, Y. T. Chuang

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93016
{pmms, moser, imichel, ytchuang}@ece.ucsb.edu [*]

Abstract. The iTrust system is a decentralized and distributed information publication, search and retrieval system, whose objective is to prevent censorship and filtering of information accessed over the Internet. In iTrust, metadata describing information are randomly distributed to multiple participating nodes. Similarly, requests containing keywords are randomly distributed to multiple participating nodes. If a participating node receives a request and the keywords in the request match the metadata it holds, the participating node sends the URL for the information to the requesting node. The requesting node then can retrieve the information from the source node. In this paper, we present the iTrust messaging and membership protocols. We establish lower bounds for the probabilities of a match if all of the participating nodes are operational and if a proportion of the participating nodes are non-operational or subverted. We provide probabilistic results for n participating nodes, where the metadata and the requests are distributed to a multiple of the square root of n nodes. These results show that distribution of the metadata and the requests to relatively few nodes suffices to achieve a high probability of a match, even if some of the nodes are non-operational or subverted.

Keywords: Internet; censorship; publication; search; retrieval; iTrust

1 Introduction

The free flow of information is one of the basic tenets of liberty and democracy. The Internet – distributed, uncontrolled, unbiased and dispassionate – greatly facilitates the free flow of information. Our trust in the accessibility of information over the Internet and the Web (hereafter referred to as the Internet) depends on benign and unbiased administration of, and access to, centralized search engines and centralized search indexes. Unfortunately, the experience of history, and even of today, shows that we cannot depend on such administrators to remain benign and unbiased in the future.

To ensure the free flow of information, an alternative to centralized search – an effective decentralized, distributed search – must be provided. It is important to provide a decentralized, distributed search infrastructure before it

[*] This research is supported in part by NSF Grant CNS-1016193.

is needed, and to ensure that it is available when it is needed, even though a user might normally use a conventional centralized search engine. A trustworthy decentralized, distributed search infrastructure can provide great assurance to the users of the Internet that a small number of administrators cannot prevent them from distributing their ideas and information and from retrieving the ideas and information of others.

The main objective of this research is to ensure the free flow of information over the Internet. The iTrust system, described in this paper, provides trustworthy publication, distribution and retrieval of information over the Internet, with no centralized mechanisms and no centralized control. The desired trust that we aim to achieve is that participants in an iTrust network can be confident that no small group of nodes or people can prevent the publication, distribution and retrieval of information.

In [14], we presented an overview of the iTrust strategy, described the HTTP implementation and user interface of iTrust, and presented an initial performance evaluation. In [4], we presented the basic idea of iTrust, described the architecture of iTrust, and presented performance results as the number of nodes increases and as the proportion of operational nodes decreases.

In this paper, we present the iTrust messaging and membership protocols. We establish lower bounds for the probabilities of a match if all of the nodes are operational and if a proportion of the nodes are non-operational or subverted. We provide probabilistic results for an iTrust network with n participating nodes, where the metadata and the requests are distributed to a multiple of the square root of n nodes. These results show that distribution of the metadata and the requests to relatively few nodes suffices to achieve a high probability of a match. Moreover, even if some of the nodes are non-operational or subverted, the probability of a match is high. Consequently, it is not easy for a small group of nodes to control which information is delivered and which is not.

2 The iTrust Strategy

The nodes that participate in an iTrust network are referred to as the *participating nodes* or the *membership*.

Some of the participating nodes, the *source nodes*, produce information, and make that information available to other participating nodes. The source nodes also produce metadata that describes their information. The source nodes distribute the metadata, along with the URL of the information, to a subset of the participating nodes chosen at random.

Other participating nodes, the *requesting nodes*, request and retrieve information. Such nodes generate requests that contain keywords, and distribute the requests to a subset of the participating nodes chosen at random. Nodes that receive a request compare the keywords in the request with the metadata they hold. If 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. A *match* between the keywords in a request received by a node and the metadata held by a node might be an exact match or a partial match, or might correspond to synonyms.

Initially, we assume that the metadata, generated by the source nodes, are small, much smaller than the information itself. Thus, the metadata can be communicated to particpating nodes that have no interest in the information. The information is potentially large, such as a video file, and is communicated only to the nodes that need it. Each participating node generates only a small proportion of the information available, and retrieves only a small proportion of that information. Nodes produce new information at unpredictable intervals, and new information is communicated quickly to the nodes that need it.

It is possible, indeed quite likely, that a single request might result in multiple responses with the same URL for a given set of metadata. In that case, the duplicates are suppressed by iTrust at the requesting node. It is also possible that a request might result in multiple responses with different URLs. Currently, we are investigating several ranking algorithms for iTrust at the requesting node, perhaps personalized to the requesting node.

In iTrust, we do not aim for secret or anonymous communication of the metadata or information. Metadata and requests are "public," because nodes must be able to match the keywords in the requests against the metadata. Rather, we aim for information publication, distribution and retrieval that cannot be easily censored, filtered or subverted. In iTrust, we use existing public key/private key encryption mechanisms to protect the communication of metadata and information against inspection and censorship by Internet routers.

In iTrust, we aim for as high a probability of a match as feasible, given the available resources (communication, processing, storage). We recognize that iTrust is more costly, particularly in communication, than a centralized search engine; however, history indicates that people are willing to accept that extra cost if they suspect censorship of a topic that they regard as important. We aim to minimize the extra cost for communication, processing and storage, but are not restricted by that cost.

3 The iTrust Messaging Protocol

At one extreme, all of the *metadata* can be flooded to all of the nodes in the network. At the other extreme, all of the *requests* for information can be flooded to all of the nodes in the network. Neither of those strategies is sufficiently efficient to be practical.

Thus, for iTrust, we use a different messaging protocol for information publication, distribution and retrieval. The steps involved in the iTrust messaging protocol are given below and are illustrated in Figures 1, 2, 3 and 4.

1. Nodes with information (the *source nodes*) distribute their metadata randomly to a set of participating nodes in the network. Some of those nodes might forward the metadata they receive to other nodes in the network.
2. Nodes that need information (the *requesting nodes*) distribute their requests randomly to a set of participating nodes in the network. Again, some of those nodes might forward the requests they receive to other nodes in the network.

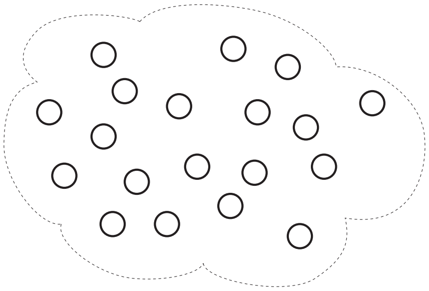

Fig. 1. A network with participating nodes.

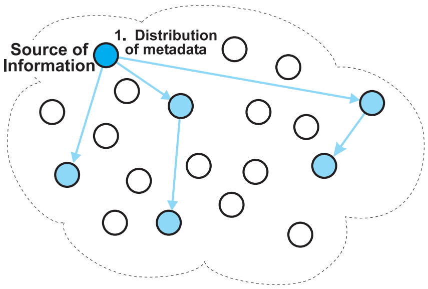

Fig. 2. A source node distributes metadata, describing its information, to randomly selected nodes in the network.

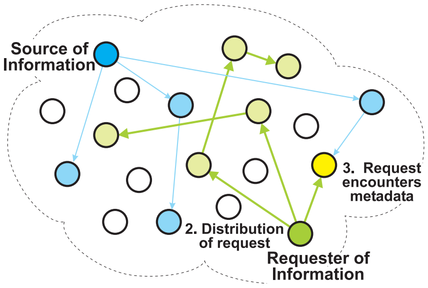

Fig. 3. A requesting node distributes its request to randomly selected nodes in the network. One of the nodes has both the metadata and the request and, thus, an encounter occurs.

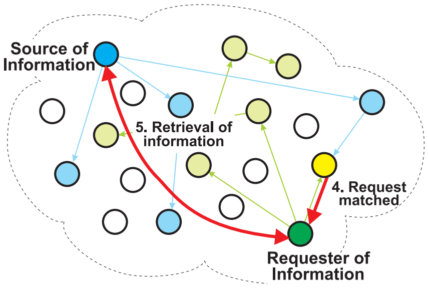

Fig. 4. A node matches the metadata and the request and reports the match to the requesting node. The requesting node then retrieves the information from the source node.

3. If a node receives both the metadata and a request, the node determines whether the metadata and the keywords in the request match.
4. If a node finds that its metadata matches the keywords in the request, the matching node provides, to the requesting node, the URL where the requesting node can retrieve the information. If a node finds that its metadata does not match the keywords in the request, it does nothing.
5. The requesting node then retrieves the information from the source node using the URL provided by the matching node.

For appropriately chosen parameters, it is probable that at least one node receives both the metadata and a request with corresponding keywords, *i.e.*, that the request encounters the metadata and a match occurs.

4 The iTrust Membership Protocol

For iTrust to work, the nodes need to know the nodes to which the metadata and the requests are distributed, *i.e.*, the *participating nodes* or the *membership*. We use the iTrust messaging protocol itself to publish, distribute and retrieve membership information. Each node maintains a membership table that contains, for each member, its Internet address and its public key.

An extensive literature on membership exists (see, *e.g.*, [2,3]), but most of that work is not relevant to iTrust. Prior work has focused on an agreed accurate membership, despite asynchrony, unreliable processors, unreliable communication, and even malice. It is impossible to achieve an agreed accurate membership [2], but good approximations are possible. Our requirements for membership present a much easier and less costly problem.

In iTrust, the nodes selected at random for distribution of the metadata and the requests constitute only a small proportion of the participating nodes. If the membership includes nodes that are no longer participating, those nodes are equivalent to non-operational nodes. Similarly, if the membership is not yet updated to include recently joined nodes, the metadata and the requests are not distributed to those nodes. The iTrust strategy still works if a substantial proportion of the nodes are non-operational, as shown in Section 6.

4.1 Joining the Membership

The protocol for joining the membership exploits the iTrust messaging protocol for publication, distribution and retrieval. The steps involved in joining the membership are given below, and are illustrated in Figures 5 and 6.

1. A node wishing to join the membership contacts any current member to obtain the current membership. It does so using mechanisms that are outside the iTrust network, perhaps email, conventional Web search, twitter, Facebook or even printed publications.
2. The node then publishes its own joining the membership, using the iTrust messaging protocol for publication, distribution and retrieval.
3. The participating nodes periodically request and retrieve information about new nodes that have joined the membership.

Periodically, a participating node can compare its membership with the membership of another node chosen at random. The node can then augment its membership with the nodes known to the other node and vice versa.

Bootstrapping involves a single node or a small group of nodes that form the initial iTrust membership.

4.2 Leaving the Membership

The protocol for leaving the membership also exploits the iTrust messaging protocol for publication, distribution and retrieval. The steps involved in leaving the membership are given below and are illustrated in Figures 7 and 8.

1. A node that wishes to leave the membership publishes its departure and then leaves.
2. Other nodes periodically request membership change information.

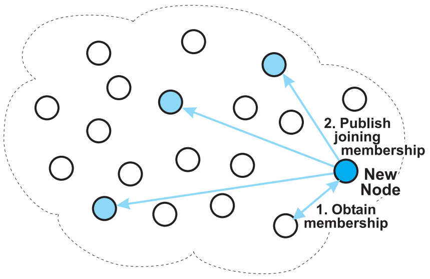

Fig. 5. A node joins the membership by first obtaining the current membership from a member and then publishing its joining the membership.

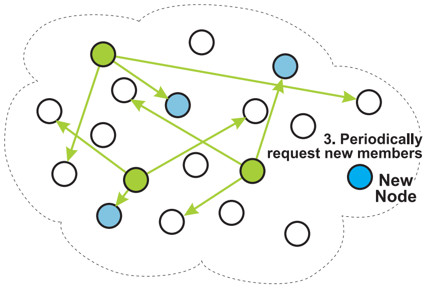

Fig. 6. Other nodes periodically request information about new nodes joining the membership.

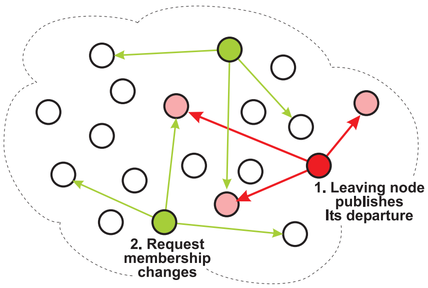
1. Leaving node publishes Its departure
2. Request membership changes

Fig. 7. A node leaves the membership by first publishing its departure and then leaving. Other nodes periodically request information about membership changes.

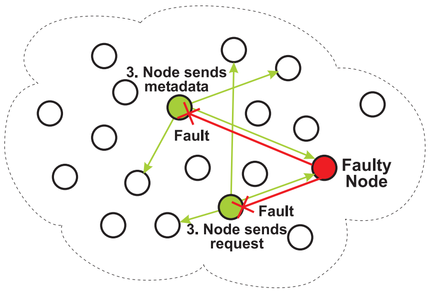

Fig. 8. A faulty node does not acknowledge metadata or request messages, which alerts other nodes of its failure. Other nodes can then remove the faulty node from the membership.

3. A node might leave the membership without publishing its intention, in particular if it becomes faulty. Such an event is detected when another node sends metadata or a request to the faulty node and does not receive an acknowledgment. The node then removes the faulty node from its membership and sends the metadata or the request to another node.

It is not appropriate to allow a node to publish the departure of another node, because doing so might enable a malicious node to remove many nodes from the membership. Rather, over time, each node individually discovers the departure of a node. When a node publishes its own departure, a digital signature (based on asymmetric encryption) is used to authenticate that publication.

5 The iTrust Implementation

The current iTrust implementation is based on HTTP and, thus, TCP/IP. As such, iTrust establishes a direct connection between any two nodes that need to communicate. Both the metadata and the requests (queries) are replicated. At each node, iTrust maintains a local index (SQLite database) of metadata and corresponding URLs for the information (resources).

The iTrust implementation on a node consists of the Web server foundation, the application infrastructure, and the public interface. These three components interact with each other to distribute the metadata and the requests to the nodes, and to retrieve resources from the nodes. Figure 9 shows these three components.

5.1 Web Server Foundation

The Apache Web server, shown in Figure 9 (a), provides the basis of the current implementation of iTrust. The Web server foundation contains no custom code; all software is used as is, which enables rapid deployment.

The iTrust implementation utilizes several standard modules, including the session and logging modules. The session module allows tracking of users on each node, so that multiple users can interact with the same node at the same time in a convenient manner. The logging module is enabled only for debugging and simulation, and can be disabled at any time by the node administrator.

The iTrust implementation also utilizes several compiled-in modules, including cURL, SQLite, and the PHP Extension Community Library (PECL) for HTTP. The cURL functions are used primarily for inter-node communication and resource-specific actions. SQLite is used for administrative information such as node, metadata and resource information. PECL HTTP is used for inter-node search and metadata queries.

5.2 Application Infrastructure

The application infrastructure, shown in Figure 9(b), is divided into three parts: metadata functions, node- and resource-related functions, and Java jar files. All three parts interact with the Web server foundation, whereas only some functions are exposed to the public interface component. Resources can be automatically scanned for metadata, or they can be manually tagged with metadata by the user, depending on the user's preference.

The metadata functions handle the creation and distribution of metadata, both internal and between nodes. To generate metadata automatically from existing resources, the metadata XML engine scans all resources and creates an XML list that relates the metadata and the resources. Other metadata functions deal with the distribution of the XML list to other nodes, or with the receipt of XML lists from other nodes. The metadata functions scan the received XML lists, and insert the metadata into the receiving node's SQLite database.

The helper functions handle bookkeeping tasks related to nodes and resources. They insert nodes into the membership, insert metadata into the database, and upload or fetch resources. The helper functions also deal with node querying and query relaying via PECL HTTP.

Java jar files are used to generate metadata quickly and easily, and to provide the user with other conveniences. The Apache Tika and Lucene packages are used to generate metadata from resources automatically and efficiently, if the user chooses not to generate the metadata manually. The WordNet dictionary provides spell checking and synonym suggestions.

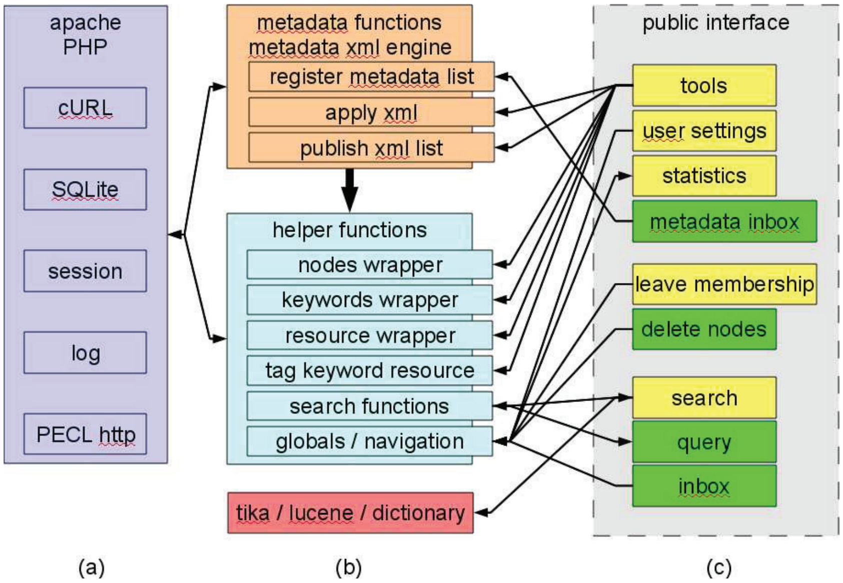

Fig. 9. The iTrust system, which comprises (a) the Web server foundation, (b) the application infrastructure, and (c) the public interface.

5.3 Public Interface

The public interface, shown in Figure 9(c), allow the user and the system administrator to interact with iTrust. The human interfaces (light boxes in the figure) consist of PHP driven HTML Web pages. The computer interfaces (dark boxes in the figure) handle all inter-node communication, including queries, resource distribution, and metadata list distribution.

Administration is performed through the tools and other Web pages. Tools allow a node administrator to add nodes or metdata keywords using simple HTML form text boxes. Adding a resource involves uploading a file (form file input) or providing a URL (form text box input). User settings and statistics Web pages provide feedback about the membership size, resource count, *etc.* The administrator may generate and distribute metadata lists or update a node's metadata lists. The administrator may also remove a node from the membership.

Searching is performed using a Web page, where the user enters a search query to request a resource. The query is sent from the current node to participating nodes using computer interfaces in a simple inbox fashion. A participating node reads its inbox for query requests, and sends back a response if there is a match.

6 Evaluation of iTrust

For this evaluation of iTrust, we assume that all of the participating nodes have the same membership set S. We assume that the metadata and the requests are distributed uniformly at random to the participating nodes, without forwarding or relaying of messages. We assume that a match is an exact match between the keywords in a request and the metadata describing the information. The keywords in a request might match the metadata for two different resources with different URLs; in such a case, the matches associated with the two resources are considered separately. Initially, we assume that all of the participating nodes in the membership set S are operational; later, we relax that assumption.

The primary parameters determining the performance of iTrust are the number n of participating nodes (*i.e.*, the size of the membership set S), the number m of participating nodes to which the metadata are distributed, and the number r of participating nodes to which the requests are distributed.

In iTrust, all of the requests are distributed and processed concurrently; however, in the proofs below, we consider the requests as successive trials.

Theorem 1. If the iTrust membership set contains n participating nodes, the metadata are delivered to m participating nodes, a request is delivered to r participating nodes, $m + r > n$, and p is the probability of a match, then $p = 1$.

Proof. Let M be the subset of nodes to which the metadata are delivered, and R be the subset of nodes to which the request is delivered. Because $m + r > n$, M and R intersect in at least one node and, thus, $p = 1$.

From Theorem 1, it follows that, if $m = r = \lceil \frac{n+1}{2} \rceil$ nodes, *i.e.*, the metadata and the requests are delivered to a majority of the nodes, then a match occurs. However, choosing $m = r = \lceil \frac{n+1}{2} \rceil$ nodes, does not scale as the number n of participating nodes increases. Therefore, for larger values of n, we consider distributing the metadata and the requests to fewer participating nodes, specifically $\sqrt{n}$, $\sqrt{2n}$ and $2\sqrt{n}$ nodes, and investigate the probabilities of a match in these cases. Note that, for $n \geq 12$, $\lceil \sqrt{n} \rceil < \lceil \sqrt{2n} \rceil < \lceil 2\sqrt{n} \rceil \leq \lceil \frac{n+1}{2} \rceil$.

Theorem 2. If the iTrust membership set contains n participating nodes, the metadata are delivered to m participating nodes, a request is delivered to r participating nodes, $n \geq m + r$, and p is the probability of a match, then

$$p = 1 - \frac{n-m}{n}\frac{n-m-1}{n-1}\cdots\frac{n-r+1-m}{n-r+1}$$

Proof. First, we find the probability q of no match on any of the r trials at the r nodes to which the requests are delivered. The probability of a match on the first trial is $\frac{m}{n}$ and, thus, the probability of no match on the first trial is $1 - \frac{m}{n} = \frac{n-m}{n}$. Likewise, the probability of no match on the second trial is $\frac{n-1-m}{n-1}$, and so on. Finally, the probability of no match on the rth trial is $\frac{n-r+1-m}{n-r+1}$.

Thus, the probability q of no match on any of the r trials is:

$$q = \frac{n-m}{n}\frac{n-1-m}{n-1}\cdots\frac{n-r+1-m}{n-r+1}$$

n	$\lceil\sqrt{n}\rceil$	p
10	4	0.9286
100	10	0.6695
1000	32	0.6527
10000	100	0.6358
100000	317	0.6351
1000000	1000	0.6325
Lower Bound		0.6321

Fig. 10. Probability p of a match when the metadata and the requests are distributed to $\lceil\sqrt{n}\rceil$ nodes.

n	$\lceil\sqrt{2n}\rceil$	p
10	5	0.9960
100	15	0.9290
1000	45	0.8800
10000	142	0.8707
100000	448	0.8668
1000000	1415	0.8653
Lower Bound		0.8647

Fig. 11. Probability p of a match when the metadata and the requests are distributed to $\lceil\sqrt{2n}\rceil$ nodes.

n	$\lceil 2\sqrt{n}\rceil$	p
10	7	1.0000
100	20	0.9934
1000	64	0.9874
10000	200	0.9831
100000	633	0.9823
1000000	2000	0.9818
Lower Bound		0.9817

Fig. 12. Probability p of a match when the metadata and the requests are distributed to $\lceil 2\sqrt{n}\rceil$ nodes.

Consequently, the probability p of a match on one or more of the r trials is $p = 1 - q$, and the result follows.

Theorem 3. If the iTrust membership set contains n participating nodes, the metadata are delivered to m participating nodes, a request is delivered to r participating nodes, and p is the probability of a match, then $p > 1 - e^{-\frac{mr}{n}}$.

Proof. As in the proof of Theorem 2, the probability q of no match on any of the r trials is:

$$\begin{aligned} q &= \frac{n-m}{n}\frac{n-m-1}{n-1}\ldots\frac{n-r+1-m}{n-r+1} \\ &< \frac{n-m}{n}\frac{n-m}{n}\ldots\frac{n-m}{n} \\ &= (\frac{n-m}{n})^r = (1-\frac{m}{n})^r \end{aligned}$$

Using Maclaurin's series, $e^x = 1 + x + \frac{x^2}{2!} + \ldots$ for all x and, thus, $1 + x < e^x$. Letting $x = -\frac{m}{n}$, we have $1 - \frac{m}{n} < e^{-\frac{m}{n}}$ and, thus, $(1 - \frac{m}{n})^r < e^{-\frac{mr}{n}}$. Consequently, $p = 1 - q > 1 - (1 - \frac{m}{n})^r > 1 - e^{-\frac{mr}{n}}$.

Figures 10, 11 and 12 show, for an iTrust membership with n participating nodes, the probability p of a match when the metadata and the requests are distributed to $\lceil\sqrt{n}\rceil$, $\lceil\sqrt{2n}\rceil$ and $\lceil 2\sqrt{n}\rceil$ nodes, respectively. For a given value of n, the number of nodes to which the metadata and the requests are delivered increases in each case, and the probability of a match increases correspondingly. These results are obtained from the formula given in Theorem 2.

Figures 10, 11 and 12 also show lower bounds for the probability p of a match when both the metadata and the requests are distributed to $\lceil\sqrt{n}\rceil$, $\lceil\sqrt{2n}\rceil$ and $\lceil 2\sqrt{n}\rceil$ nodes, respectively. These lower bounds are obtained from the inequality given in Theorem 3.

In the above evaluation, we have chosen specific values of m and r, such that $m = r$, *i.e.*, the number of nodes to which the metadata are distributed is the same as the number of nodes to which the requests are distributed. However, m and r need not be the same. Currently, we are investigating the use of different values of m and r.

Now, we relax the assumption that all of the nodes are operational. Thus, we assume that a proportion x of the n participating nodes are operational (and, thus, a proportion $1 - x$ of the n participating nodes are non-operational). Furthermore, we assume independence of the nodes that are non-operational.

Theorem 4. If the iTrust membership set contains n participating nodes of which a proportion x are operational, the metadata are delivered to m participating nodes, a request is delivered to r participating nodes, $mx + r > n$, and p is the probability of a match, then $p = 1$.

Proof. The proof is similar to that of Theorem 1.

Theorem 5. If the iTrust membership set contains n participating nodes of which a proportion x are operational, the metadata are delivered to m participating nodes, a request is delivered to r participating nodes, $n \geq mx + r$, and p is the probability of a match, then

$$p = 1 - \frac{n - mx}{n} \frac{n - 1 - mx}{n - 1} \cdots \frac{n - r + 1 - mx}{n - r + 1}$$

Proof. First, we find the probability q of no match on any of the r trials at the r nodes to which the requests are delivered. Consider the first trial. The probability that the node that receives the request has the metadata is $\frac{m}{n}$, and the probability that the node has the metadata and is operational is $\frac{mx}{n}$. Thus, the probability of no match on the first trial because the node does not hold the metadata or is not operational is $1 - \frac{mx}{n} = \frac{n-mx}{n}$. Likewise, the probability of no match on the second trial because the second of the r nodes does not hold the metadata or is not operational is $\frac{n-1-mx}{n-1}$, and so on. Finally, the probability of no match on the rth trial is $\frac{n-r+1-mx}{n-r+1}$.

Thus, the probability q of no match on any of the r trials is:

$$q = \frac{n - mx}{n} \frac{n - 1 - mx}{n - 1} \cdots \frac{n - r + 1 - mx}{n - r + 1}$$

Consequently, the probability p that one or more of the r nodes that receives the request has a match and is operational is $p = 1 - q$, and the result follows.

Theorem 6. If the iTrust membership set contains n participating nodes of which a proportion x are operational, the metadata are delivered to m participating nodes, a request is delivered to r participating nodes and p is the probability of a match, then $p > 1 - e^{-\frac{mrx}{n}}$.

Proof. The proof is similar to that of Theorem 3.

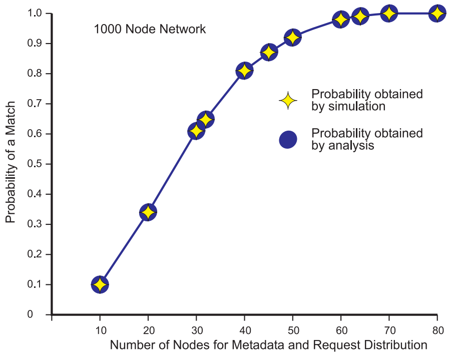

Fig. 13. Probability of a match, obtained by simulation and by analysis, as the number of nodes to which the metadata and the requests are distributed increases.

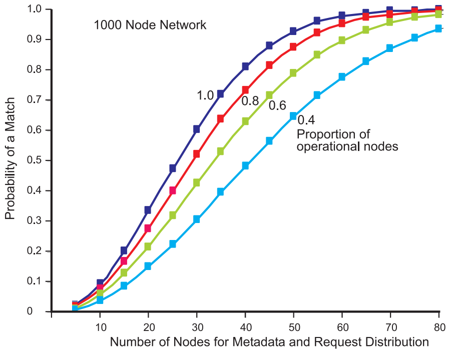

Fig. 14. Probability of a match as the number of nodes to which the metadata and the requests are distributed increases, for various proportions of operational nodes.

Figure 13 compares the probabilities of a match for an iTrust membership with $n = 1000$ nodes, obtained from the analytical formula given in Theorem 2 and from our simulation. For the simulation, each set of metadata was distributed once, and each of the search requests was performed 10,000 times and the results were averaged. The figure shows the probability of a match when the metadata and the requests are distributed to $m = r = 10, 20, 30, 40, 50, 60, 70, 80$ nodes and also to $m = r = \lceil\sqrt{1000}\rceil = 32$, $m = r = \lceil\sqrt{2000}\rceil = 45$, and $m = r = \lceil 2\sqrt{1000}\rceil = 64$ nodes. As the figure shows, the results obtained from the analytical formula and from the simulation are really close.

Figure 14 shows the probabilities of a match for an iTrust membership with $n = 1000$ participating nodes, obtained from Theorem 5, when a proportion of the nodes are non-operational. The figure shows the probability of a match as the number of nodes to which the metadata and the requests are distributed increases when a proportion $x = 1.0, 0.8, 0.6, 0.4$ of the participating nodes are operational. As the figure shows, iTrust retains significant utility in circumstances in which a substantial proportion of the nodes are non-operational, which might be the circumstances in which the information is most needed.

7 Related Work

The centralized search engine strategy, such as that of Google, stores metadata for information in a centralized index, and matches keywords in the requests against the metadata stored at the central site. The centralized search engine strategy is used commercially for Internet search because it is efficient, but it is vulnerable to manipulation, filtering and censorship. The centralized publish/subscribe approach [8] also uses a centralized index, against which the queries are matched, raising the same issues of trust as centralized search engines.

Bender *et al.* [1] 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." Eugster *et al.* [8], Mischke and Stiller [15], and Risson and Moors [16] provide comparisons of distributed search methods. The structured approach requires the nodes to be organized in an overlay network, based on distributed hash tables, trees, rings, *etc.* The unstructured approach uses randomization, and requires the nodes to distribute and find information by exchanging messages. iTrust uses the unstructured approach.

Cohen and Shenker [6] have studied how replication can be used to improve search in unstructured peer-to-peer networks. They show that square root replication is theoretically optimal in terms of minimizing the overall search traffic. They replicate objects based on access frequencies (popularities), whereas iTrust uses uniform random replication of objects, so that popular nodes are not more vulnerable to attacks.

Gnutella [10], one of the first unstructured networks, uses flooding of requests to find information. Extensions of Gnutella involve supernodes [20], which improve efficiency but incur some of the trust risks of centralized strategies. Freenet [5] 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.

Other peer-to-peer systems, such as that of Lv *et al.* [13], use random walks to improve on the flooding of Gnutella. They start with uniform random replication of data, but then adaptively adjust the replication degree based on the query rate, and use square root replication to improve performance. They also consider creation and deletion of the replicas of the data (or metadata). BubbleStorm [17] replicates both queries and data, and combines random walks with flooding to perform exhaustive search. It also considers churn, leaves and crashes, like the iTrust membership protocol does.

Zhong and Shen [22] use random walks for requests, where the number of nodes visited by a request is proportional to the square root of the request popularity, as in [6]. Ferreira *et al.* [9] use random walks to replicate both queries and data to the square root of the number of nodes in the network. Unlike [6], in their system, replication of metadata and requests is independent of access frequency (popularity), as in iTrust. Like these other researchers, we also exploit the square root function in iTrust.

PlanetP [7] maintains a local index that contains metadata for documents published locally by a peer, and a global index that describes all peers and their metadata. It replicates the global index throughout the network using gossiping. Galanx [18] uses a local peer index to direct user queries to relevant nodes in the network. It is based on the Apache Web server and on the BerkeleyDB data store. iTrust likewise utilizes the Apache Web server, and maintains a local index of metadata and corresponding URLs for the data. None of the above unstructured systems is particularly concerned with trust, as iTrust is.

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.* [11] investigate the integration of social network search with Web search. They conclude that such integration can lead to more timely and efficient search experiences. Yang *et al.* [21] propose a search mechanism for unstructured peer-to-peer networks based on interest groups, formed by nodes with similar interests. iTrust likewise allows users interested in a particular topic or cause to form a social network, so that they can share information.

Two other systems that, like iTrust, are concerned with trust are Quasar and OneSwarm. Quasar [19] is a probabilistic publish/subscribe system for social networks. Quasar aims to protect the users' sensitive information, which is different from the trust objective of iTrust. OneSwarm [12] is a peer-to-peer system that allows information to be shared either publicly or anonymously, using a combination of trusted and untrusted nodes. OneSwarm aims to protect the users' privacy, whereas iTrust aims to support free flow of information and to prevent censorship and filtering of information.

8 Conclusions and Future Work

The iTrust information publication, search and retrieval system, addressed in this paper, is particularly valuable for individuals who fear that the conventional Internet search mechanisms might be censored or subverted. The very existence of iTrust can help to deter such censorship or subversion attempts.

We are currently investigating a range of possible attacks on iTrust and countermeasures to such attacks. Our objective for iTrust is a network in which individual nodes can detect a potential attack, and can adapt to an attack to maintain trustworthy information publication, search and retrieval even when under attack.

We are also implementing an SMS/MMS version of iTrust that can operate over the cellular network. In the future, we plan to create a Wi-Fi and/or Bluetooth version of iTrust for mobile ad-hoc networks. In such a network, iTrust nodes can share resources by forwarding metadata, requests and resources through intermediate nodes without the need for a wireless access point, a cellular network connection, or even an Internet connection.

References

1. 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 (2006).
2. Chandra, T.D., Hadzilacos, V., Toueg, S., Charron-Bost, B: On the Impossibility of Group Membership. In: 15th ACM Symposium on Principles Distributed Computing, pp. 225–267 (1996).
3. Chockler, G.V., Keidar, I., Vitenberg, R.: Group Communication Specifications: A Comprehensive Study. ACM Computing Surveys 33:4, 427–469 (2001).
4. Chuang, Y.T., Michel Lombera, I., Moser, L.E., Melliar-Smith, P.M.: Trustworthy Distributed Search and Retrieval over the Internet. In: 2011 International Conference on Internet Computing (2011).

5. Clarke, I., Sandberg, O., Wiley, B., Hong, T.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009, Springer, Heidelberg, pp. 46–66 (2000).
6. Cohen, E., Shenker S.: Replication Strategies in Unstructured Peer-to-Peer Networks. In: 2002 ACM SIGCOMM Conference, pp. 177–190 (2002).
7. Cuenca-Acuna, F.M., Peery, C., Martin, R.P., Nguyen, T.D.: PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. In: 12th IEEE International Sympoisum on High Performance Distribured Computing, pp. 236–246 (2003).
8. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The Many Faces of Publish/Subscribe. ACM Computing Surveys 35:2, 114–131 (2003).
9. Ferreira, R.A., Ramanathan, M.K., Awan, A., Grama, A., Jagannathan, S.: Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks. In: Fifth IEEE International Conference on Peer-to-Peer Computing, pp. 165–172 (2005).
10. Gnutella, http://gnutella.wego.com/
11. Gummadi, K.P., Mislove A., Druschel, P.: Exploiting Social Networks for Internet Search. In: 5th Workshop on Hot Topics in Networks, pp. 79–84 (2006).
12. Isdal, T., Piatek, M., Krishnamurthy, A., Anderson, T.: Privacy Preserving P2P Data Sharing with OneSwarm. In: 2010 ACM SIGCOMM Conference, pp. 111–122 (2010).
13. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and Replication in Unstructured Peer-to-Peer Networks. In: 16th ACM International Conference on Supercomputing, pp. 84–95 (2002).
14. Michel Lombera, I., Chuang, Y.T., Melliar-Smith, P.M., Moser, L.E.: Trustworthy Distribution and Retrieval of Information over HTTP and the Internet. In: Third International Conference on the Evolving Internet (2011).
15. Mischke, J., Stiller, B.: A Methodology for the Design of Distributed Search in P2P Middleware. IEEE Network 18:1, pp. 30–37 (2004).
16. 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, RFC 4981, http://tools.ietf.org/html/rfc4981 (2007).
17. Terpstra, W.W., Kangasharju, J., Leng, C., Buchman, A.P.: BubbleStorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search. In: 2007 ACM SIGCOMM Conference, pp. 49–60 (2007).
18. Wang, Y., Galanis, L., DeWitt, D.J.: Galanx: An Efficient Peer-to-Peer Search Engine. Technical Report, University of Wisconsin, Madison (2003).
19. Wong, B., Guha, S.: Quasar: A Probabilistic Publish-Subscribe System for Social Networks. In: 7th International Workshop on Peer-to-Peer Systems (2008).
20. Yang, B., Garcia-Molina, H.: Improving Search in Peer-to-Peer Networks. In: 22nd IEEE International Conference on Distributed Computing Systems, pp. 5–14 (2002).
21. 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, pp. 247–252 (2003).
22. Zhong, M., Shen, K.: Popularity-Biased Random Walks for Peer-to-Peer Search under the Square-Root Principle. In: 5th International Workshop on Peer-to-Peer Systems (2006).