Analysis of the Match Probabilities for the iTrust Information Network with Message Forwarding

L. E. Moser and P. M. Melliar-Smith
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106 USA
moser@ece.ucsb.edu, pmms@ece.ucsb.edu

***Abstract*—The iTrust system is a completely distributed and decentralized information publication, search and retrieval system, that is designed to defend against censorship of information in the Internet. In this paper, we investigate the iTrust system with message forwarding, which spreads the responsibility of message distribution more widely across the nodes in the network. We present an analysis of the match probabilities of the iTrust system with message forwarding, in terms of the forwarding fanout, the number of levels of forwarding, and the forwarding probability. We show that, with a forwarding probability of 1.0, relatively small values of the forwarding fanout and the number of levels of forwarding suffice to achieve a high match probability and a reasonable message cost.**

***Keywords*-distributed and decentralized information publication, search and retrieval; message forwarding; match probability; iTrust**

I. Introduction

The free flow of information is one of the cornerstones of a free society. Modern societies have come to depend on the Internet for distribution of, and access to, information. Unfortunately, centralized mechanisms, such as search engines, are very vulnerable to censorship of information. Even in societies in which censorship is not currently practiced, there is no guarantee of the free flow of information in the future, as evidenced by history and recent events.

The iTrust system is a completely distributed and decentralized information publication, search and retrieval system, that is designed to defend against censorship of information in the Internet. In the iTrust network, the nodes distribute metadata and requests (queries) for information to subsets of the participating nodes chosen at random. The nodes that receive the requests try to match the keywords in the requests with the metadata they hold. If a node has a match, it then responds to the requesting node with the URL of the corresponding document and the requesting node then uses that URL to retrieve the document.

In [17], we showed that, if the metadata and request messages are distributed to $2\sqrt{n}$ nodes, where n is the number of nodes in the iTrust network, then the match probability is high (0.9817). Instead of the source node's distributing its metadata and the requesting node's distributing its request *directly* to $2\sqrt{n}$ nodes, other nodes might forward the metadata and request messages they receive to further nodes, thus spreading the responsibility of message distribution more widely across the nodes. In [18], we presented an algorithm that calculates the probability density function (pdf) for the number of nodes reached when forwarding messages in an arbitrary network. The iTrust system does not need to flood the metadata and request messages to all of the nodes in the network to achieve a high match probability; rather, it needs to distribute those messages to only $2\sqrt{n}$ nodes. In this paper, we investigate the match probability (and also the number of nodes reached and the message cost) in the iTrust network with message forwarding, as a function of the forwarding fanout, the number of levels of forwarding, and the forwarding probability.

The rest of the paper is organized as follows. Section II presents an overview of the design of iTrust. Section III presents the match probabilities for iTrust without message forwarding. Section IV discusses the probability density functions for the number of nodes reached and the corresponding number of messages required. Section V presents an algorithm that combines the match probabilities for iTrust with the probability density functions for the number of nodes reached to obtain the match probabilities for iTrust with message forwarding, along with results obtained from that algorithm. Section VI discusses related work, and Section VII presents conclusions and future work.

II. Design of iTrust

The iTrust system is a completely distributed system that involves no centralized mechanisms and no centralized control. We refer to the nodes that participate in an iTrust network as the *participating nodes* or the *membership*. Multiple iTrust networks may exist at any point in time, and a node may participate in multiple iTrust networks at the same point in time.

In iTrust some nodes, the *source nodes*, produce information, and make that information available to other participating nodes. The source nodes produce metadata that describes their information, and distribute that metadata to a subset of the participating nodes chosen at random, as shown in Figure 1. The metadata are distinct from the information they describe, and include a list of keywords and the URL of the source of the information.

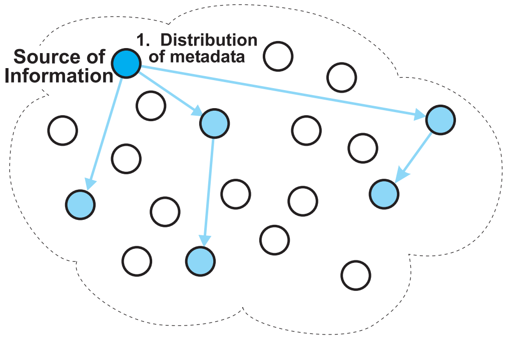

Figure 1. A source node distributes metadata, describing its information, to randomly chosen nodes in the network.

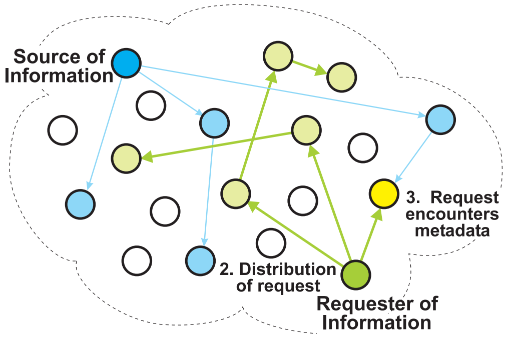

Figure 2. A requesting node distributes its request to randomly chosen nodes in the network. One of the nodes has both the metadata and the request and, thus, an encounter occurs.

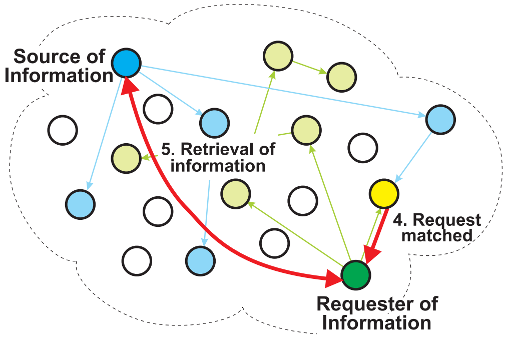

Figure 3. A node matches the metadata and the request and reports the match to the requester, which then retrieves the information from the source node.

Other nodes, the *requesting nodes*, request and retrieve information. Such nodes generate requests (queries) that contain keywords for the desired information, and distribute their requests to a subset of the participating nodes chosen at random, as shown in Figure 2.

The participating nodes compare the metadata in the requests they receive with the metadata they hold. If a node finds a match, which we call an *encounter*, the matching node returns the URL of the information to the requesting node. The requesting node then uses the URL to retrieve the information from the source node, as shown in Figure 3.

In iTrust, a match can be an exact match or a partial match, or it can involve synonyms.

III. Match Probabilities of iTrust without Message Forwarding

First, we provide the match probabilities for iTrust *without* message forwarding. We assume that all of the participating nodes in an iTrust network have the same membership set and that any node can connect *directly* to any other node. The primary parameters that determine the match probabilities of iTrust without message forwarding are:

- n: Number of participating nodes (*i.e.*, the size of the membership set), $1 < n$
- m: Number of participating nodes to which the metadata are distributed, $1 \leq m \leq n$
- r: Number of participating nodes to which the requests are distributed, $1 \leq r \leq n$
- x: Proportion of the n participating nodes that are operational, *i.e.*, respond to requests when they have a match, $0 < x \leq 1.0$.

Note that, if $x < 1.0$, a node is not reporting a match when it has a match, either because it has crashed or because it is malicious. In [3], we presented algorithms for detecting and defending against malicious nodes in the iTrust network.

Our probabilistic analysis of iTrust is based on the hypergeometric distribution [8], which describes the number of successes in a sequence of random draws from a finite population *without* replacement. Thus, the probability of *exactly* k matches is given by:

$$\begin{aligned} P(k) &= \frac{\binom{mx}{k}\binom{n-mx}{r-k}}{\binom{n}{r}} \\ &= \frac{\left(\frac{mx}{k}\frac{mx-1}{k-1}\cdots\frac{mx-k+1}{1}\right)\left(\frac{n-mx}{r-k}\cdots\frac{n-mx-r+k+1}{1}\right)}{\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1}} \end{aligned}$$

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

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

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

Thus, the probability of *one or more* matches (*a match*) is given by:

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

for $mx + r \leq n$.

For an iTrust network without message forwarding, with $n = 1000$ nodes and $x = 1.0$, 0.8, 0.6 operational nodes, Figure 4 shows the match probabilities, obtained from Equation (1), as the number of nodes to which the metadata and the requests are distributed increases. Note that, if $n = 1000$ and $x = 1.0$, distribution of the metadata and the requests to $2\sqrt{n} \approx 62$ nodes results in a high match probability (0.9817).

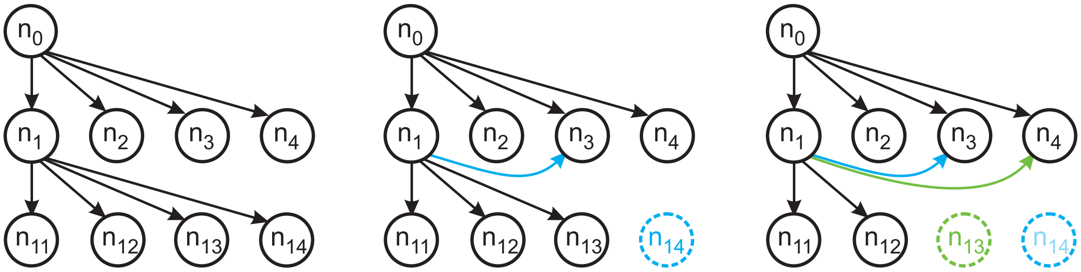

Figure 5. Three examples of nodes at levels $l = 0, 1, 2$ with $c = 4$ and $f = 1.0$.

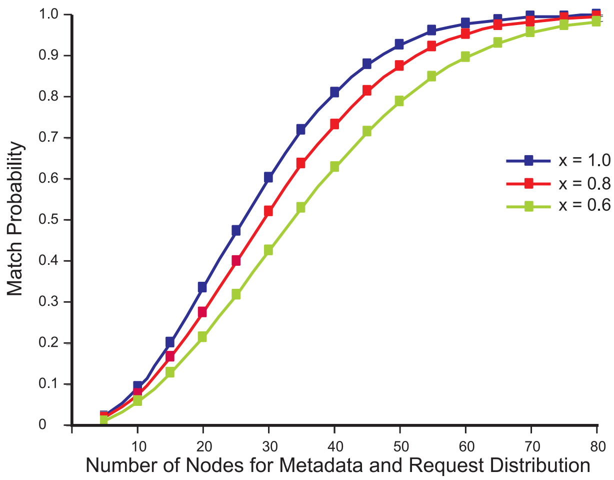

Figure 4. Match probabilities for iTrust without message forwarding, as the number of nodes to which the metadata and the requests are distributed increases, for various proportions x of operational nodes when $n = 1000$.

IV. Probability Density Functions for iTrust with Message Forwarding

Next, we discuss the probability density functions for the number of nodes reached for iTrust *with* message forwarding. We assume that any node can connect to any other node, and consider the following parameters of the forwarding algorithm:

- n: Number of nodes in the network, $1 < n$
- c: Forwarding fanout, $1 \leq c < n$
- l: Level of message forwarding, $0 \leq l$
- f: Probability of message forwarding, $0 < f \leq 1.0$.

Note that, if $c = 1$, then the message follows a random walk and, if $c = n - 1$ and $l = 1$, then the message is broadcast. Also note that a node forwards a message to c nodes with probability f and does not forward the message at all to any other node with probability $1 - f$.

We investigate settings of the parameters c, l and f in order to control the number of nodes reached, the number of messages sent, and the match probability. Other papers, such as [19], have investigated settings of f to control the number of nodes reached.

A node n_0 sends a message to c randomly chosen nodes $\{n_1, n_2, \ldots, n_c\}$ at level 1, other than itself, with forwarding probability f. Each node n_i, $1 \leq i \leq c$, forwards n_0's message to c randomly chosen nodes $\{n_{i1}, n_{i2}, \ldots, n_{ic}\}$ at level 2, other than itself, with forwarding probability f, and so on. Note that n_{ij} might be n_0 or $n_{i'}$, $1 \leq i' \leq c$, $i' \neq i$, and that n_{ij} might be $n_{i'j'}$, $1 \leq i' \leq c$, $1 \leq j' \leq c$, $i' \neq i$. We refer to such nodes as *duplicate nodes*.

Figure 5 shows three examples of nodes at levels $l = 0, 1, 2$ with $c = 4$. In the middle example, $n_3 = n_{14}$ is a duplicate node. In the example on the right, $n_3 = n_{14}$ is a duplicate node and $n_4 = n_{13}$ is a duplicate node.

In [18], we presented a general algorithm for an arbitrary network that determines the probability density function $pdf[i], 1 \leq i \leq n$, for the number of nodes reached within l levels of message forwarding for specific values of c, l and f. In calculating the pdf, the algorithm eliminates duplicate nodes. Using that algorithm, we found that the number of nodes reached, within a specific number of levels of message forwarding, exhibits a wide range, particularly for smaller values of c and smaller values of f.

Figure 6 shows the probability density functions for the number of nodes reached up through level l, for various values of c, l and f in a network with $n = 1000$ nodes. It can be seen that, with $f = 1.0$, appropriate choices of the parameters c and l can be made to distribute the messages to desired numbers of nodes, and that the variability in the number of nodes reached is reasonable. However, when $f < 1.0$, there is substantial variability in the number of nodes reached.

The algorithm, presented in [18], for calculating the probability density function for the number of nodes reached is easily extended to calculate the probability density function for the number of messages required to reach specific numbers of nodes.

Figure 7 shows the probability density functions for the number of messages required (*i.e.*, message cost) up through level l, for various values of c, l and f in a network with $n = 1000$ nodes. It can be seen that, with $f = 1.0$, the messages cost is quite well defined. However, with $f = 0.8$ and $f = 0.6$, there is substantial variability in the message cost.

Note that, for $f = 1.0$, there is less variability in the number of messages required than in the number of nodes reached. The reason is that the larger number of nodes at level l to which the message is sent makes it more likely that a duplicate node will be encountered, whereas the number of messages sent at level $l-1$ and prior levels is not affected by the duplicate nodes at level l.

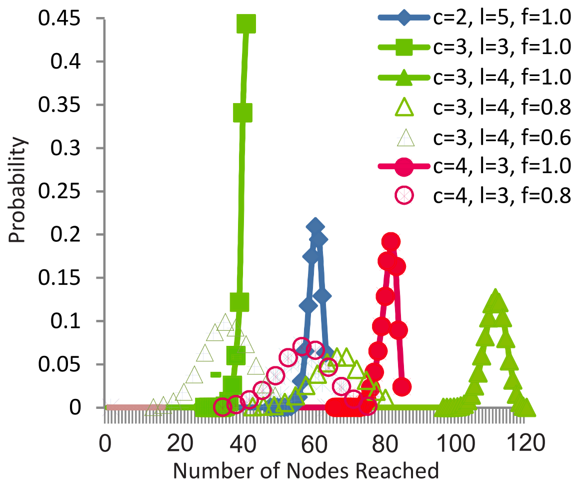

Figure 6. Probability density functions for the number of nodes reached up through level l for various values of c, l and f when $n = 1000$.

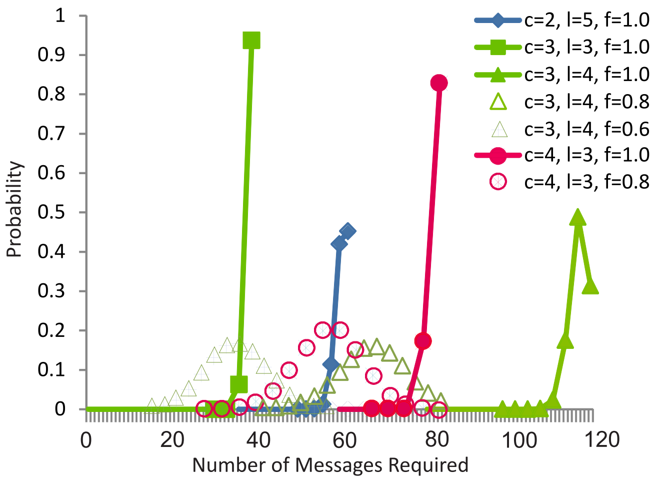

Figure 7. Probability density functions for the number of messages required up through level l for various values of c, l and f when $n = 1000$.

In Figure 6 and Figure 7, it can be seen that, for $n = 1000$ nodes, setting the forwarding fanout $c = 2$, the level of forwarding $l = 5$, and the forwarding probability $f = 1.0$ results in the distribution of the metadata and request messages to approximately 60 nodes with corresponding message cost, to achieve a high match probability, as shown in Figure 9. With $c = 4$, $l = 3$ and $f = 1.0$, the forwarding algorithm delivers the messages to more nodes with higher message cost, which might be acceptable. However, with $c = 3$, $l = 3$ and $f = 1.0$, the forwarding algorithm delivers messages to too few nodes whereas, with $c = 3$, $l = 4$ and $f = 1.0$, the forwarding algorithm delivers messages to too many nodes, with a much higher message cost.

Figure 6 and Figure 7 also show the probability density functions for several values of c and l with $f < 1.0$. It can be seen that, for $f < 1.0$, fewer nodes are reached and fewer messages are required, but the probability density functions exhibit much more variability than when $f = 1.0$. This variability results in reduced match probabilities, as shown in Figure 10. The reason is that sometimes the metadata or request messages reach too few nodes, which is not offset by the other times when the metadata or request messages reach more nodes.

V. Match Probabilities for iTrust with Message Forwarding

The method for finding the match probability (one or more matches) for iTrust with message forwarding is given in Figure 8. It uses the probability of a match $P(k \geq 1)$ given by Equation (1) and the probability density functions $pdf[m]$ and $pdf[r]$ given by the algorithm in [18] for the number of nodes reached when forwarding the metadata to m nodes and the request messages to r nodes.

Figure 9 shows the match probabilities for various values of c, l and x in an $n = 1000$ node iTrust network with message forwarding when $f = 1.0$. The solid lines represent the match probabilities when all of the nodes are operational,

```
findMatchProbability($n$, $c$, $l$, $f$, $x$)
matchProb = 0.0
    for $m$ = $c + 1$ to $n$ do
        for $r$ = $c + 1$ to $n$ do
            matchProb = matchProb +
                            $P(k \geq 1) \times pdf[m] \times pdf[r]$
return matchProb
```

Figure 8. The method for finding the match probability in the iTrust network with message forwarding.

i.e., when $x = 1.0$. It can be seen that, when the forwarding fanout $c = 2$, at least $l = 5$ levels of forwarding are needed to achieve a high match probability. Moreover, if the forwarding fanout $c = 3$, then $l = 4$ levels of forwarding are required and, if $c = 4$, then $l = 3$ levels of forwarding suffice. Furthermore, if $c = 3$, then $l = 3$ levels of forwarding result in a match probability that is not great but that might suffice in some circumstances.

The match probabilities, shown in Figure 9, are traded off against the number of messages required (message cost), shown in Figure 7. For example, with $c = 3$, $l = 4$ and $x = 1.0$, the match probability is high but the message cost is also high whereas, with $c = 3$, $l = 3$ and $x = 1.0$, the message cost is lower but the match probability is also lower.

Figure 9 also shows the match probabilities when some of the nodes are not operational ($x = 0.8$ and $x = 0.6$). With $c = 3$ and $l = 4$ or with $c = 4$ and $l = 3$, the iTrust network with forwarding provides a high degree of resilience to crashed or malicious nodes, but that resilience comes at the cost of more messages, as shown in Figure 7. With $c = 3$ and $l = 3$ and with $c = 2$ and $l = 5$ (which are economical in the use of messages), there are substantial reductions in the match probabilities in the presence of crashed or malicious nodes. That reduction in the match probabilities can be prevented by increasing the number of levels of forwarding when non-operational nodes are detected.

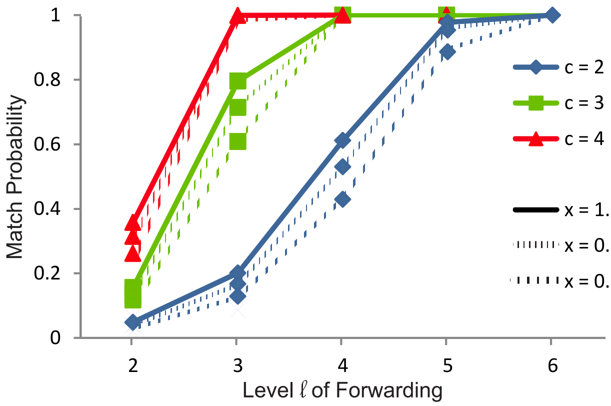

Figure 9. Match probabilities for iTrust with message forwarding, for various values of c, x and l when $n = 1000$ and $f = 1.0$.

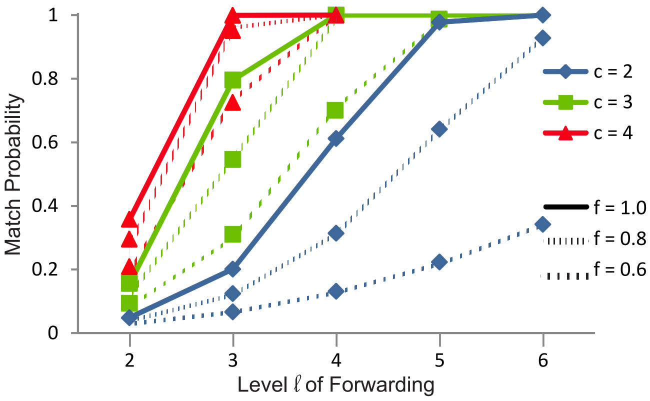

Figure 10. Match probabilities for iTrust with message forwarding, for various values of c, f and l when $n = 1000$ and $x = 1.0$.

An alternative strategy, in the presence of non-operational nodes, is to increase the forwarding fanout c. Such a strategy might be appropriate in moving from $c = 3$ and $l = 3$ to $c = 4$ and $l = 3$. With $c = 2$ and $l = 5$, a simple increase in the forwarding fanout is excessive. It might be more appropriate to move from $c = 2$ and $l = 5$ to $c = 3$ and $l = 4$, increasing the forwarding fanout and reducing the number of levels of forwarding.

Some forwarding systems [19] use the forwarding probability parameter f to reduce the number of nodes reached. Figure 6 shows the effect of varying the forwarding probability f on the number of nodes reached. In comparison with the c and l parameters which are more deterministic in their effect on the number of nodes reached, the f parameter introduces more variability into the number of nodes reached.

The variability in the number of nodes reached with a forwarding probability $f < 1.0$, shown in Figure 6, has, in iTrust, a significant detrimental effect on the match probability, shown in Figure 10. The match probability is traded off against the number of messages required (message cost), shown in Figure 7. For example, with $c = 3$, $l = 4$ and $f = 0.6$, the mean message cost is about the same as with $c = 3$, $l = 3$ and $f = 1.0$, but the variability in the message cost is greater, and the match probability is lower, in the former case than in the latter case. However, sometimes setting $f < 1.0$ might be useful. For example, with $c = 3$, $l = 4$ and $f = 0.8$, the match probability is still high but the message cost is much less than with $c = 3$, $l = 4$ and $f = 1.0$. Nonetheless, this case must be compared with $c = 2$, $l = 5$ and $f = 1.0$, which has an equally high match probability and a lower message cost with less variability.

VI. Related Work

Message forwarding has been used as an alternative to multicasting by a single source node, and in gossiping protocols. Hedetniemi *et al.* [13] present a survey of the theory of gossiping and broadcasting in communication networks. Shah [20] provides a comprehensive discussion of gossip algorithms. Farley [6] presents algorithms that construct broadcast networks with approximately the minimum number of links. He also determines upper and lower bounds to broadcast m messages throughout a network of n nodes [7], but he does not consider probability density functions.

Deering and Cheriton [5] provide a survey of multicast routing, using spanning trees along which messages are forwarded in internetworks and extended local-area networks. Lin *et al.* [15] exploit directional information for gossiping in wide-area networks. Castro *et al.* [1] describe a scalable, decentralized multicast infrastructure based on gossiping, which is extended to a hierarchical infrastructure in [12].

Gnutella [11], one of the first unstructured information sharing networks, uses flooding of requests to find information. A node makes a copy of a file when it receives the file it requested. If the query rate is high, nodes quickly become overloaded and the system ceases to function satisfactorily. Freenet [4] is more efficient than Gnutella, because it learns from previous requests. In Freenet, nodes that successfully respond to requests receive more metadata and more requests, making it easy for a group of untrustworthy nodes to gather most of the searches into their group.

Other peer-to-peer systems use random walks to improve on the flooding of Gnutella. Random walks correspond to $c = 1$ with larger values of l, in our notation. For example, Ferreira *et al.* [9] use random walks to replicate the metadata and the requests to the square root of the number of nodes in the network. Lv *et al.* [16] use random walks and 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. 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. Gia [2] employs biased random walks to direct queries towards high-capacity nodes. In contrast, iTrust distributes the metadata and the requests to nodes that are chosen uniformly at random.

BubbleStorm [21] replicates both queries and data, and combines random walks and flooding to perform exhaustive search. Leng *et al.* [14] present mechanisms for maintaining the desired degree of replication in BubbleStorm. The iTrust system does not use exhaustive search but, rather, distributes the metadata and the requests to $2\sqrt{n}$ nodes to achieve a high probability of a match.

Gkantsidis *et al.* [10] show that, for searching, random walks achieve better results than flooding in two cases: (1) when peers form two-tier clusters, and (2) when clients reissue queries repeatedly. Generally, random walks are more susceptible to node and communication faults than flooding, and can result in higher latency for responses to requests. The iTrust network with forwarding uses larger values of c and smaller values of l than random walks, with less vulnerability to malicious nodes that do not forward.

VII. Conclusions and Future Work

We have presented an analysis of the match probabilities of the iTrust information network with message forwarding. We have shown that, in iTrust, relatively small values of the forwarding fanout c and the forwarding level l with a forwarding probability $f = 1.0$ result in the metadata and request messages being distributed to $2\sqrt{n}$ nodes and, thus, a high match probability and a reasonable message cost. With $f < 1.0$, the number of nodes reached and the match probabilities exhibit much greater variability, with detrimental effects on the match probabilities. Thus, for the iTrust network with forwarding, it is preferable to adjust c and l and to keep $f = 1.0$.

In future work, we plan to investigate the match probabilities of iTrust in the presence of message forwarding for networks that are not completely connected. In particular, many networks, particularly social networks, contain neighborhoods that are completely connected locally but where connections to other neighborhoods are sparse. Such networks can achieve very large sizes, without requiring every node to have knowledge of a large number of other nodes. Forwarding within neighborhoods, and between neighborhoods, can achieve greater scalability of the iTrust information network.

Acknowledgment

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

References

[1] M. Castro, P. Druschel, A. M. Kermarrec and A. I. Rowstron, “Scribe: A large-scale and decentralized application-level multicast infrastructure,” *IEEE Jour. Selected Areas Communications*, vol. 20, no. 8, 2002, pp. 1489–1499.

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham and S. Shenker, “Making Gnutella-like P2P systems scalable,” *Proc. 2003 ACM SIGCOMM Conf.*, Aug. 2003, pp. 407-418.

[3] Y. T. Chuang, I. Michel Lombera, P. M. Melliar-Smith and L. E. Moser, “Detecting and defending against malicious attacks in the iTrust information retrieval network,” *Proc. 26th Intl. Conf. Information Networking*, Feb. 2012, pp. 263–268.

[4] I. Clarke, O. Sandberg, B. Wiley and T. Hong, “Freenet: A distributed anonymous information storage and retrieval system,” *Proc. Workshop Design Issues in Anonymity and Unobservability*, July 2000, pp. 46–66.

[5] S. E. Deering and D. R. Cheriton, “Multicast routing in datagram internetworks and extended LANs,” *ACM Trans. Computer Systems*, vol. 8, no. 2, 1990, pp. 85–110.

[6] A. M. Farley, “Minimal broadcast networks,” *Networks*, vol. 9, iss. 4, 1979, pp. 313–322

[7] A. M. Farley, “Broadcast time in communication networks,” *SIAM Jour. Applied Mathematics*, vol. 39, no. 2, Oct. 1980, pp. 385–390.

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

[9] R. A. Ferreira, M. K. Ramanathan, A. Awan, A. Grama and S. Jagannathan, “Search with probabilistic guarantees in unstructured peer-to-peer networks,” *Proc. 5th IEEE Intl. Conf. Peer-to-Peer Computing*, Aug. 2005, pp. 165–172.

[10] C. Gkantisidis, M. Mihail and A. Saberi, “Random walks in peer-to-peer networks: Algorithms and evaluation,” *Performance Evaluation*, vol. 63, no. 3, Mar. 2006, pp. 241–263.

[11] Gnutella, http://www.gnutella.wego.com

[12] I. Gupta, A. M. Kermarrec and A. J. Ganesh, “Efficient and adaptive epidemic-style protocols for reliable and scalable multicast,” *IEEE Trans. Parallel and Distributed Systems*, vol. 17, no. 7, 2006, pp. 593–605.

[13] S. M. Hedetniemi, S. T. Hedetniemi and A. L. Liestman, “A survey of gossiping and broadcasting in communications networks,” *Networks*, vol. 18, 1988, pp. 319–349.

[14] C. Leng, W. W. Terpstra, B. Kemme, W. Stannat and A. P. Buchmann, “Maintaining replicas in unstructured P2P systems,” *Proc. ACM CoNEXT 2008 Conf.*, Dec. 2008, pp. 19:1–19:12.

[15] M. J. Lin and K. Marzullo, “Directional gossip: Gossip in a wide area network,” in *Proc. Third European Dependable Computing Conf.*, LNCS 1667, Springer, 1999, pp. 364–379.

[16] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, “Search and replication in unstructured peer-to-peer networks,” *Proc. Intl. Conf. Supercomputing*, June 2002, pp. 84–95.

[17] P. M. Melliar-Smith, L. E. Moser, I. Michel Lombera and Y. T. Chuang, “iTrust: Trustworthy information publication, search and retrieval,” *Proc. 13th Intl. Conf. Distributed Computing and Networking*, LNCS 7129, Springer, Jan. 1012, pp. 351–366.

[18] L. E. Moser and P. M. Melliar-Smith, “Probabilistic analysis of message forwarding” *Proc. IEEE Intl. Conf. Computer Communications and Networks*, July-Aug. 2013, pp. 1–8.

[19] K. Oikonomou and I. Stavrakakis, “Performance analysis of probabilistic flooding using random graphs,” *Proc. 2007 IEEE Intl. Symp. World of Wireless, Mobile and Multimedia Networks*, June 2007, pp. 1–6.

[20] D. Shah, “Gossip algorithms,” *Foundations and Trends in Networking*, vol. 3, no. 1, 2008, pp. 1–125.

[21] W. W. Terpstra, J. Kangasharju, C. Leng and A. P. Buchman, “BubbleStorm: Resilient, probabilistic, and exhaustive peer-to-peer search,” *Proc. 2007 ACM SIGCOMM Conf.*, Aug. 2007, pp. 49–60.

[22] M. Zhong and K. Shen, “Popularity-biased random walks for peer-to-peer search under the square-root principle,” *Proc. 5th Intl. Workshop Peer-to-Peer Systems*, 2006.