A Security/Efficiency-Optimized Infrastructure for Wide-Area Distributed Systems

Haiying Shen
Department of Computer Science and Computer Engineering
University of Arkansas, Fayetteville, AR 72701

Abstract - *Due to lack of a central management, wide-area distributed systems are severely threatened by a variety of malicious users in today's Internet. Current reputation-based and anonymity technologies for node communication enhance system security. However, most of these methods gain security at the cost of efficiency degradation. On the other hand, many technologies that aim to improve system efficiency neglect malicious participants in the system. This paper presents a P2P-based security/efficiency-optimized infrastructure for node communication. The infrastructure jointly treats security and efficiency in its operation to meet the high performance requirements of wide-area distributed systems. The infrastructure includes a policy: trust-based adaptive response forwarding. It enhances overall system performance by harmonious tradeoffs between security and efficiency. Simulation results show the superiority of the infrastructure in achieving both high security and high efficiency in comparison with other related approaches.*

Keywords: Structured peer-to-peer, Distributed Systems, Distributed hash table, Efficiency, Security

1 Introduction

The immense popularity of the Internet has produced a significant stimulus to the tremendous advance of wide-area distributed systems such as peer-to-peer (P2P) and Grid systems. In such a wide-area distributed system, nodes need to communicate with each other. In a high performance system, the messages should arrive at their destination efficiently and securely without dropping or corruption. *Security* encompasses many aspects such as data privacy and anti-Byzantine defenses. In this paper, we use *security* to represent a system environment that excludes malicious or selfish nodes in its operations and offers anonymity between communication nodes.

Ubiquitous users in the system have posed a challenge of secure communication. During communication, malicious or selfish users may drop messages. To survive in such an environment, distributed systems must function correctly even though some participants are malicious. Currently, numerous protocols and systems have been proposed to offer the users some form of security. However, most technologies [1, 2] gain high security at a cost of performance efficiency degradation. On the other hand, to enhance communication efficiency, some algorithms [3] allows the nodes to exchange security for efficiency. Moreover, some algorithms neglect the existence of malicious or selfish nodes and may lead to unsuccessful communication.

In this paper, we propose a P2P-based infrastructure to supply security protection in terms of malicious users avoidance and anonymity provision, and meanwhile guarantee high efficiency in node communications for wide-area distributed systems. The infrastructure includes trust-based adaptive response forwarding policy.

The rest of this paper is structured as follows. Section 2 introduces the P2P-based infrastructure design including the policy. Section 3 shows the performance of these policies with comparison of other related policies using a variety of metrics. Section 4 presents a concise review of related work. Section 5 concludes this paper.

2 Security/Efficiency-Optimized Infrastructure

2.1 Infrastructure Overview

Before we begin detailed discussion of the P2P-based infrastructure, we briefly describe structured P2P overlay networks. Structured P2P overlay networks [4, 5, 6, 7, 8] is a class of decentralized systems that partition ownership of a set of objects among participating nodes, and can efficiently route messages to the unique owner of any

given object. The overlay network provides two main functions: `insert(key,data)` and `lookup(key)` to store the data to a node responsible for the key, and to retrieve the data. The message for the two function is forwarded from node to node through the overlay network until it reaches the data's owner. Each node maintains a routing table recording its neighbors in the overlay network for message routing based on the P2P routing algorithm.

The goal of anonymity is to allow data sharing between clients and servers in such a manner that no one can determine the identities of them. P2P middleware contributes to achieving anonymity in the application level. It obscures the physical locations of nodes from each other, and restricts a node's view only to its neighbors. In addition, initial messages and forwarded messages are constructed and processed similarly, so nodes cannot differentiate message forwarding neighbors from initial message generating neighbors. Furthermore, tunnelled communication provides certain protection to two endpoints.

In this paper, we take a distributed file sharing system as an example for the wide-area distributed systems. We assume the existence of a reputation system in the system that is scalable and can accurately calculate the trustworthiness for each node. We define a *downstream* as a tunnelled path from a client to a server, and a *upstream* as a tunnelled path from a server to a client. For *upstream* communication, we propose trust-based adaptive response forwarding policy for the data forwarding in the upstream. For *downstream* communication, we adopt the FairTrust policy [9] for security and efficiency request routing on the infrastructure. FairTrust is a trust-based fairness-oriented server selection policy for a peer to choose anther peer to interact. FairTrust takes into account both reputation and capacity factors in server selection. Thus, in each step of a routing, a node chooses its neighbor relying on FairTrust policy to ensure the high efficiency and security of message routing.

2.2 Trust-based Adaptive Response Forwarding Policy

A general approach to achieving anonymous on overlay networks are to construct an indirect path between a client and a server. For instance, in Freenet [1] and Mute [2], after a file is located, it is sent back along the nodes in the request routing path. However, an indirect routing comes at the cost of high communication overhead and leads to inefficient communication. To reduce the overhead, Mantis [3] allows the clients to exchange anonymity for download efficiency. It uses anonymous communication to search for files and to send control signals, while allowing the data be sent directly from the server to the client using a return address spoofed UDP. Instead of providing full anonymity, Mantis only protects the privacy of servers.

Trust-based adaptive response forwarding policy (*TrustAdp*) reduces communication overhead by shrinking paths in size based on client reputation. The extend to which a server needs protection from a client is based on the trustworthiness of the client. On the other hand, more tunnelled upstream transfers provide higher anonymity. To provide anonymity and while guaranteeing high efficiency, *TrustAdp* adapts the extent to which a tunnelled upstream shrinks to the client trustworthiness. Particularly, the higher reputation of a client, the less hops are needed in the upstream, and vise versa.

TrustAdp maps the trustworthiness values of nodes in the reputation system to integer values. After a server gets a request from a client, it checks the trustworthiness value of the client in the reputation system. We assume that a server checks the reputation value based on a code contained in the query, which does not reveal the client's identifier. Let t denote the client's trustworthiness level. If t is high and the client is the server's neighbor, then the server sends back data to the client directly. Otherwise, the server sends the requested data to the upstream node that is t hops away. Rather than relying on static hop-by-hop or direct communication in response forwarding, *TrustAdp* dynamically adjusts path length based on client trustworthiness, ensuring high anonymity protection and efficiency.

3 Performance Evaluation

This section demonstrates the distinguishing properties of the infrastructure for secure and efficient node communication in a wide-area distributed system. We compared *TrustAdp* with Freenet [1], Mute [2] and Mantis [3]. Simulation results verified the superiority of *TrustAdp* policies toward achieving high security and efficiency.

In the simulation, the file requests are consecutively generated with a random source node and a random target. Table 1 lists the parameters of the simulation and their default values, unless otherwise specified. We assumed a bounded Pareto distribution for the capacity of nodes. This distribution reflects the real world situations where machine capacities vary by different orders of magnitude. We assume that there are 5 levels of trustworthiness from 0.2 to 1 with 0.2 increase in each level. Each node is randomly assigned as one of the three types according to a uniform distribution. We assume that a reputation system is employed which accurately reflects

Table 1. Simulated environment and algorithm parameters.

Environment Parameter	Default value
Number of nodes n	Fixed at 2048
Node capacity c	Bounded Pareto: shape 2 lower bound: 500 upper bound: 50000
Number of reputation levels	5

each node's trustworthiness.

Anonymity is important in building a secure environment. The probability that a server will be at a risk of identifier exposure to a client depends on three factors. First, whether the client is a malicious node. Second, the number of proxies forwarding a response back from a server to a client. Third, whether the proxies are malicious. We assume that the proxies chosen for routing are trustworthy. The first factor depends on the client trustworthiness, and higher trustworthiness leads to lower possibility of exposure. Since node trustworthiness is in $[0, 1]$ in the test, we define the exposure probability due to the first factor as $1 - t$ where t is client trustworthiness. Because more proxies lead to lower exposure possibility, we define the probability due to the second factor as: $\frac{1}{l}$, where l denotes the number of proxies. The product of the two probabilities is the probability that a server's identifier is disclosed to a malicious client. In this experiment, we compare the privacy protection and performance efficiency of the policies.

3.1 Performance in Security

Figure 1(a) shows the average, 1st and 99th percentiles of exposure probabilities versus client trustworthiness level. A number of important observations can be made from this figure: (1) As expected, the exposure probability of each policy decreases as trustworthiness increases. When clients have the highest trustworthiness of 1, the exposure probability is 0, which means there is no response forwarding that makes a server exposed; (2) Mantis has the highest exposure probability, *TrustAdp* reduces the probability more than one third, and Freenet generates the least exposure probability; (3) Freenet/Mute exhibits a larger variance than *TrustAdp*, while Mantis has no variance at all.

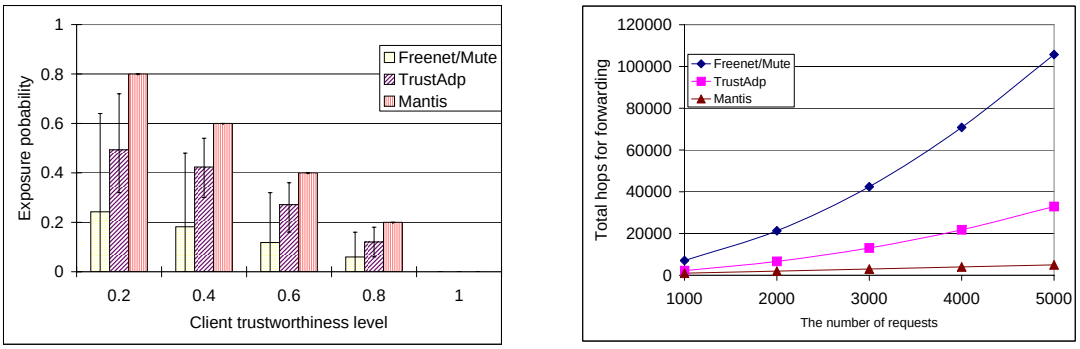
As predicted, more hops along the upstream result in higher server privacy protection. Mantis has the highest exposure probability due to its direct communication between a client and a server. It does not exhibit variance in its exposure probability because its exposure rate depends on the client trustworthiness directly without the involvement of proxies. On the other hand, Freenet/Mute has the lowest exposure probability due to its static hop-by-hop routing along the upstream. However, its benefits are outweighed by its high overhead and efficiency degradation in forwarding. By adjusting the number of hops to the client trustworthiness adaptively, *TrustAdp* improves the privacy protection of Mantis significantly, and reduces the overhead of Freenet/Mute considerably.

3.2 Performance in Efficiency

Figure 1(b) shows the number of hops for response forwarding along the upstream versus the number of requests. We can observe that Freenet/Mute has the highest hops, while Mantis has the least hops. The results imply that Freenet/Mute incurs a high overhead due to hop-by-hop forwarding, leading to efficiency degradation. *TrustAdp* decreases the overhead dramatically, and Mantis achieves highest efficiency in response forwarding. Combining the results in Figure 1(a) and (b), we can conclude that Freenet/Mute achieves high security at a high cost, while Mantis cannot provide privacy protection though it has high efficiency. *TrustAdp* achieves an optimized tradeoff between anonymity and efficiency by reducing overhead while maintaining a high level of privacy to nodes.

4 Related Work

In the past few years, a variety of security technologies have been proposed in order to provide a secure environment for distributed systems. One group includes anonymity protocols which hide the relationship between an observable action and the identity of the users involved with this action. Anonymity systems Mute [2], ANTS [10] and Mantis [3] implement Ants protocol in which a requester broadcasts a query, and the response is forwarded back along its query route. APFS [11] and Tor [12] provide anonymity using Onion routing protocol [13] in which messages are randomly routed, and each router obtains no information about the message routing path other than the identity of the following router using encryption technology. MIXes [14] provides anonymity by letting nodes wait until they have received a number of messages and then forward them mixed up. Crowds [15] randomly routes each message through a crowd of nodes until one node decides to pass it to the server. Freenet [1] is a searchable P2P system which makes it impossible for an attacker to find all copies of a particular file by letting each node store all the files that pass across it. Freenet uses response forwarding along



(a) Exposure probability

(b) Total hops for forwarding

Figure 1. Efficiency of different anonymity response forwarding policies.

its query route to provide provider anonymity, and uses broadcast to obscure the intended recipient. Other systems rely on broadcasting to achieve anonymity such as DC-nets [16] and XOR-trees [17].

Recently, a variety of reputation systems have been developed with different characteristics [18, 19, 20, 21, 22]. Meanwhile, many studies have also been devoted to efficient message routing. Castro et al. [23] proposed a neighbor selection algorithm to direct most traffic to high capacity nodes. Some algorithms [24, 25, 26] let the query being forwarded to high capacity nodes in the routing to enhance communication efficiency.

This proposed infrastructure is developed based on FairTrust policy [9]. FairTrust is a trust-based fairness-oriented server selection policy for a peer to choose anther peer to interact. FairTrust takes into account both reputation and capacity factors in server selection. FairTrust aims to improve the security and high performance of P2P networks. The infrastructure introduced in this paper adopts the FairTrust policy for secure and efficient routing in the downstream. In addition to selecting a server securely and efficiently, the infrastructure also offers anonymous protection for both clients and servers.

5 Conclusions

It's important to guarantee that the wide-area distributed systems function correctly and efficiently even though some participants are malicious. Some security technologies enhance system security, but at the cost of performance efficiency degradation. On the other hand, some methods improve system efficiency, but cannot guarantee successful operations due to the neglect of security aspect. This paper presents a P2P-based security/efficiency-optimized infrastructure that offers both high security and high efficiency in node communication. The infrastructure not only offers a secure environment with communication in a anonymous fashion, but also enhances overall system performance by optimized tradeoffs between security and efficiency. Simulation results illustrate the superiority of the infrastructure compared with other related approaches, and show the effectiveness of each policy component in the infrastructure.

Acknowledgements

This research was supported in part by the Acxiom Corporation.

References

[1] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. International Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66, 2001.

[2] The mute file sharing systems. http://mute-net.sourceforge.net/.

[3] S. Bono, C. Soghoian, and F. Monrose. Mantis: A lightweight, server-anonymity preserving, searchable P2P network. Technical report, TR-2004-01-B, Information Security Institute, Johns Hopkins University, 2004.

[4] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Transactions on Networking*, 1(1):17–32, 2003.

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable net-

work. In *Proc. of ACM SIGCOMM*, pages 329–350, 2001.

[6] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[7] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz. Tapestry: An Infrastructure for Fault-tolerant Wide-Area Location and Routing. *IEEE Journal on Selected Areas in Communications*, 12(1):41–53, 2004.

[8] H. Shen, C. Xu, and G. Chen. Cycloid: A scalable constant-degree P2P overlay network. *Performance Evaluation*, 63(3):195–216, 2006. An early version appeared in Proc. of International Parallel and Distributed Processing Symposium (IPDPS), 2004.

[9] H. Shen and Y. Zhu. FairTrust: Toward Secure and High Performance P2P Networks. In *Proc. of the 13th International Conference on Parallel and Distributed Systems (ICPADS)*, 2007.

[10] Ants. http://antsp2p.sourceforge.net.

[11] V. Scarlata, B. Levine, and C. Shields. Responder anonymity and anonymous peer-to-peer file sharing. In *Proc. of IEEE International Conference on Network Protocols (ICNP)*, 2001.

[12] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. of the 13th USENIX Security Symposium*, 2004.

[13] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. *IEEE Symposium on Security and Privacy*, 1997.

[14] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), 1981.

[15] M. Reiter and A. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[16] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untranceability. *Communications of the ACM*, 24(2), 1988.

[17] S. Dolev and R. Ostrovsky. Xor-trees for efficient anonymous multicast and reception. *ACM Transactions on Information and System Security*, 3(2):63–84, 2000.

[18] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems*, 2007.

[19] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proc. of the 10th International Conference on Information and Knowledge Management (CIKM)*, pages 310–317, 2001.

[20] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[21] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proc. of the 12th International World Wide Web Conference (WWW)*, 2003.

[22] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proc. of the 3rd International Conference on Peer-to-Peer Computing (P2P)*, September 2003.

[23] M. Castro, M. Costa, and A. Rowstron. Debunking some myths about structured and unstructured overlays. In *Proc. of the 2nd Symposium on Networked Systems Desiging and Implementation (NSDI)*, 2005.

[24] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *Proc. of ACM International Conference on Supercomputing (ICS)*, 2001.

[25] Q. Lv, S. Ratnasamy, and S. Shenker. Can heterogeneity make gnutella scalable? In *Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[26] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *Proc. of ACM SIGCOMM*, 2003.