ReDS: Reputation for Directory Services in P2P Systems

[Extended Abstract]

Matthew Wright, Apu Kapadia, Mohan Kumar, and Apurv Dhadphale
mwright@uta.edu,
kapadia@indiana.edu,
mkumar@uta.edu,
apurv.dhadphale@mavs.uta.edu

***Abstract*—Peer-to-peer (P2P) architectures are gaining popularity and importance for applications ranging from massive-scale Internet content delivery to mobile social networks. Such P2P systems must provide *directory services* for locating peers with the desired content and services. These directory services are themselves decentralized, such as with *distributed hash tables* (DHTs), which allow for efficient locating of objects without any centralized directory. Being a distributed system over a diverse set of untrusted nodes, however, such directory services must be resilient to adversarial behavior. Otherwise, the entire P2P system can be crippled by manipulating or simply denying access to resources. We propose *Reputation for Directory Services (ReDS)*, a framework for using reputation management to enhance the security of finding information in distributed systems. While previous reputation systems have addressed several specific applications of P2P networks (e.g., by identifying peers who share bad files), directory services form the backbone of P2P systems and have unique properties with respect to reputation that make them worth investigating. In this extended abstract, we motivate our investigation of ReDS and describe preliminary results that show its effectiveness in the Salsa P2P system.**

Key Words: reputation systems, directory services, trust, detection, peer-to-peer systems

I. Introduction

Peer-to-peer (P2P) architectures are gaining popularity and importance for applications ranging from massive-scale Internet content delivery to mobile social networks. For example, content delivery networks such as Akamai already deliver large quantities of their traffic using a distributed network, and are extending their reach to ordinary Internet clients using a P2P model.[1] The popular Skype Voice-over-IP system employs a P2P model, with a global decentralized user directory and calls are routed through peers.[2] P2P architectures also show great promise for a wide range of other applications such as anonymous communications [5], [6] and social networking [9]. Further, the P2P model can be employed in mobile and pervasive environments to facilitate greater information and resource sharing.

As a means to efficiently locate other peers and resources in a distributed setting, P2P systems must provide *directory services*. These directory services are themselves decentralized, such as with *distributed hash tables* (DHTs) [7], [8], [10], which allow for efficient locating of objects without any centralized directory. Since P2P architectures inherently must distribute functionality to a diverse set of nodes (or "peers") across various network domains out of the control of any single authority, these peers cannot be fully trusted. The lack of a central authority for fundamental tasks such as routing and directory services means that the network must be resilient to adversarial behavior to be usable by honest participants. Numerous reputation systems have been proposed to detect misbehaving peers and punish them or block them from using the the system. Hoffman et al. [3] provide an extensive survey of such systems. Most of these systems are focused on application-level information, such as the quality of resources (e.g. files) that a peer provides. Directory services, however, form the backbone of P2P systems. An unreliable directory service can render the entire P2P network unusable since adversaries can prevent wholesale access to files, or even selectively censor access to specific data. Furthermore, directory services have unique properties with respect to reputation that make them worth investigating separately from general-purpose reputation systems. *We propose Reputation for Directory Services (ReDS), a framework for using reputation management to enhance the security of locating information in distributed systems.*

We have previously built reliable DHT systems (Salsa [6] and Halo [4]). We are extending such

[1] Akamai NetSession Interface, http://www.akamai.com/html/misc/akamai_client/netsession_interface_overview.html

[2] http://www.skype.com/help/guides/p2pexplained/

directory-service systems to incorporate reputation information for tolerating adversarial behavior. This work leverages the specific structural aspects of directory services for more efficient computation and dissemination of reputation information. DHTs such as Chord [10], CAN [7], or Pastry [8] are structured specifically so that lookups are routed through the network in a predictable fashion. Based on the success or failure of various lookups, therefore, the original querier can make inferences about the reputation of nodes along the lookup paths. We are identifying metrics to compute such reputation scores and seeking ways to allow nodes to utilize other peers selectively for different types of queries

We will now briefly sketch our high-level system and attack models (§II), provide background on the Salsa P2P directory service (§III), and then illustrate how the ReDS framework can be applied to Salsa with very good results (§IV) from our preliminary investigation.

II. Directory-Service and Attack Models

System Model. We define a distributed directory service to be a P2P service that allows peers to find information about how to access resources in the system, including all types of data and services. We will initially focus on structured P2P directory services with redundancy, such as Salsa [6], Halo [4], and Cyclone [1]. These systems inherently have many desirable features for the ReDS framework and provide a rich environment to study these features in detail.

Attacker Model. We will assume an adversary who seeks to manipulate the results from directory service lookups. The adversary's goal could be to cause peers to use attacker-controlled nodes for services and information, for example as a way to spread spam or malware or to exploit peers requesting service. It may, however, simply be a denial of service strategy in which lookup results lead to invalid or incorrect nodes. To achieve these ends, the attacker's most effective strategy against a highly distributed network is to control a large number of peers in the system. Social-network-based anti-sybil techniques such as SybilInfer [2] and SybilLimit [11] may be employed to prevent the number of malicious peers from growing without bound. Nevertheless, we expect that through social engineering, the attacker may be able to inject a constant fraction of the total number of peers into the network without detection. We expect such an attacker to both directly manipulate lookup results as well as try to deceive any attempt at using reputation or malicious node detection. Thus, we must devise system designs that are robust to both types of attack.

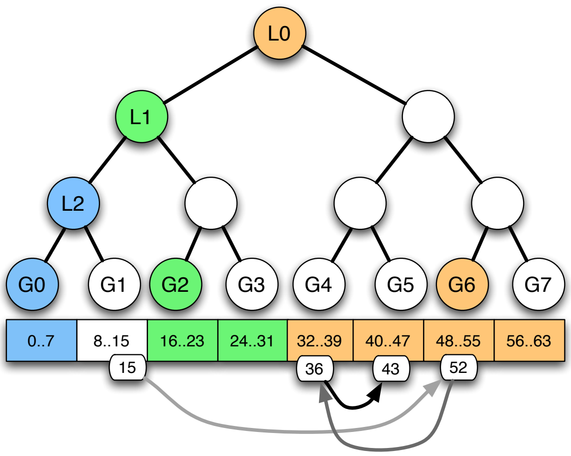

Fig. 1. **Salsa Virtual Tree:** Node 15 in group G1 has global contacts in groups G0, G2, and G6. Arrows on the bottom show a recursive lookup path; the lines get darker as they get closer to the target.

III. Salsa Overview

Salsa. Salsa is a fully distributed directory service for node discovery that was originally aimed at anonymity systems [6]. Salsa is a structured P2P system, similar to Chord [10], in which there is an ID space that is mapped to by a consistent hash function (e.g. SHA-1 mapping to a 160-bit ID space). The Salsa architecture is based on a virtual balanced binary tree, as depicted in Figure 1. Nodes are placed into groups created by dividing the ID space into contiguous regions — group $G1$ in the figure contains nodes with IDs in the range 8 to 15. All nodes know the peers in their group (their *local contacts*), as well as a small set of *global contacts* that are outside of their group. In particular, global contacts are selected randomly, but based on the virtual tree structure, as indicated by the colors in Figure 1. By using the global contacts in a recursive lookup process, as depicted by the arrows in the figure, a lookup will reach a member of the target's group in $O(\log_2 G)$ steps,

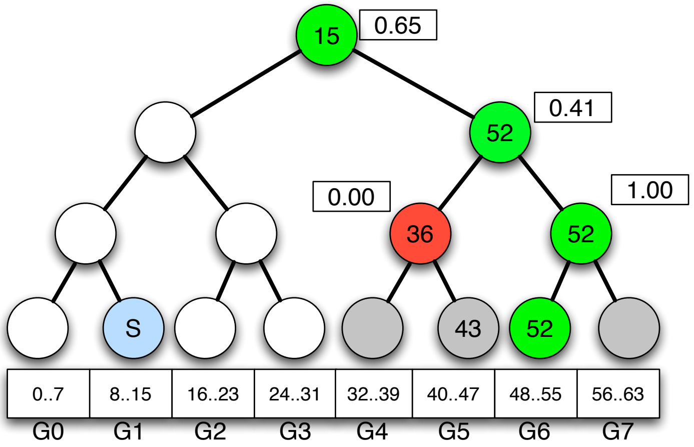

Fig. 2. **Salsa-ReDS**: Node S' reputation tree for node 15. Green nodes are believed to be good and the red node is believed to be bad. The numbers in the boxes are representative reputation scores.

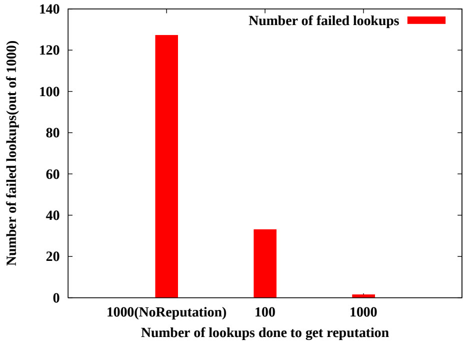

Fig. 3. Salsa-ReDS simulations: The no. of failures out of 1,000 lookups. The leftmost bar shows the number of failures when reputation is not used.

where G is the number of groups in the system. To protect against malicious nodes manipulating lookups, Salsa uses *redundant lookups*, in which the requesting node asks a subset of its local contacts to perform lookups for the same target. Randomness in global contact selection provides *path diversity*, which prevents the redundant lookups from using the same node in the lookup path, since using the same node would negate the benefits of redundancy. A useful benefit of the structured ID space is that, when a requesting node receives conflicting results, the actual target owner is the closest to the target by definition. These benefits make Salsa a potentially useful directory service for many applications in which reliable directory service is needed, not just for anonymity.

Limitations. Salsa and Halo (a similar system, not described due to lack of space) both use redundancy and path diversity to reduce lookup failures. While effective, these techniques require substantial communication overhead, e.g. $O(\log n)$ *times* as much communication in Halo for lookups than in Chord. Additionally, there remain a number of failed lookups in both Salsa and Halo for reasonable redundancy levels. While small as a fraction of the total number of lookups, the additional failures can hurt system integrity. *Reducing the amount of redundancy while also reducing failure rates is an important goal of our proposal.* Further, Salsa and Halo require the use of structured P2P. We would like to improve directory services in systems that cannot use structured P2P. We propose to investigate reputation to improve directory services.

IV. Preliminary Results

In this section, we describe our initial investigation into reputation for directory services applied to Salsa. We present the design of a system that we call *Salsa-ReDS* that employs reputation in a way that takes advantage of and benefits the Salsa system specifically. Further, we show the results of simulation experiments that demonstrate the potential of the ReDS approach.

Salsa-ReDS Design. Salsa-ReDS takes advantage of redundant lookups and the ability of the requesting node to determine which of several results is correct. If the requesting node gets conflicting results from the redundant lookups, it assumes that the closest one is correct. The local contacts that provided correct results gain positive reputation and those that provided incorrect results lose reputation. While this simple idea would likely identify which local contacts were more reliable, we also take advantage of the Salsa tree structure. Specifically, we note that while the requesting node does not know about the specific path of global contacts used in a lookup, it does know in which part of the Salsa virtual tree the target address is located. Thus, it can keep track of each local contact's lookup performance for each part of the ID space, as lookups in the same sub-tree will share nodes in the lookup path. In Figure 2, node S's local contact, 15, has a malicious node in its lookup path to the subtree to G4 and G5; all lookups through 15 to this subtree will fail. If there are no other malicious nodes, then lookups through 15 will be successful 75% of the time on average. However, by taking the tree structure into account, S can use 15 for three-fourths of the tree with a 100% success rate and stop using 15 for the part of the tree that fails. To handle system dynamics, such as nodes leaving the system, and to adapt to attackers that change their behavior, the reputation score is taken as an exponentially weighted moving average at each level of the tree. To account for information available at all levels of the tree, the reputation scores for a specific lookup are taken from the relevant lookup path in the tree and multiplied together. Thus, a low score at any level will result in a low overall score for the local contact.

Simulation Results. To evaluate Salsa-ReDS, we implemented it in our existing Salsa simulation environment [6]. For the results we present here, we simulate systems with 10,000 nodes, 20% of which were attackers, and 256 groups. We generate a new Salsa system and select a random node N to make (redundant) lookup requests in each simulation run. We present the number of failed lookups out of 1,000; we ignore lookups in which the target owner is an attacker, as the attacker would behave correctly in those lookups. In Figure 3, node N first makes a number of lookups (100 or 1,000) purely for obtaining reputation. Then we observe the result of 1,000 lookups, during which the reputation scores are not updated. Although not realistic, this allows us to take a snapshot of the utility of reputation after a set amount of information is gathered. The redundancy is fixed to four. We see from the figure that the number of failed lookups drops from 127 out of 1,000 without reputation to 33 with reputation after 100 lookups to just 1.5 after 1,000 lookups. *The latter is nearly a 99% reduction in the failure rate*. Without reputation, much higher redundancy would be required to reach such a low failure rate.

V. Conclusions

In this extended abstract, we described the problem of securing directory services in open P2P systems. To help solve this problem, we proposed ReDS, which applies the main techniques of reputation systems to directory services. While many reputation systems have been investigated, ReDS uniquely utilizes the structure of certain P2P systems. We believe that there remains a great deal to investigate to understand the principles of ReDS designs and apply them in other contexts.

References

[1] M. S. Artigas, P. G. Lopez, J. P. Ahullo, and A. F. G. Skarmeta. Cyclone: a novel design schema for hierarchical DHTs. In *Proc. P2P '05*, 2005.

[2] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, 2009.

[3] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42(1):1–31, 2009.

[4] Apu Kapadia and Nikos Triandopoulos. Halo: High-Assurance Locate for Distributed Hash Tables. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS)*, pages 61–79, February 2008.

[5] Prateek Mittal and Nikita Borisov. Shadowwalker: peer-to-peer anonymous communication using redundant structured topologies. In *ACM Conference on Computer and Communications Security*, pages 161–172, 2009.

[6] Arjun Nambiar and Matthew Wright. Salsa: a structured approach to large-scale anonymity. In *ACM Conference on Computer and Communications Security*, pages 17–26, 2006.

[7] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp, and Scott Shenker. A scalable content-addressable network. In *SIGCOMM*, pages 161–172, 2001.

[8] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329, 2001.

[9] Daniel R. Sandler and Dan S. Wallach. Birds of a fethr: Open, decentralized micropublishing. In *Proceedings of the 8th International Workshop on Peer-to-Peer Systems (IPTPS '09)*, Boston, MA, April 2009.

[10] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.

[11] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 3–17, 2008.