The Design of a Versatile, Secure P2PSIP Communications Architecture for the Public Internet

David A. Bryan and Bruce B. Lowekamp
College of William and Mary / SIPeerior Technologies, Inc.
Williamsburg, VA, USA
{bryan, lowekamp}@cs.wm.edu

Marcia Zangrilli
SIPeerior Technologies, Inc.
Williamsburg, VA, USA
marcia@sipeerior.com

Abstract

Communications systems, encompassing VoIP, IM, and other personal media, present different challenges for P2P environments than other P2P applications. In particular, reliable communication implies that each resource (person) is unique and must be reliably located, without false negatives. Because of their prevalence in real deployments, the overlay must use endpoints behind NATs as peers and must be resilient against DoS attacks that attempt to disrupt the system's routing properties or DoS a particular person. We have designed and implemented a P2P communications system that addresses these issues, now deployed as both a commercial and academic project, which has resulted in a leading proposal for a P2PSIP standard in the IETF. We present the design tradeoffs necessary to meet the requirements of a reliable communications system and provide guidance on appropriate choices for designers of other similar systems in the future. In particular, the practical issues of non-transitive routing, NAT traversal required by our endpoints, and the prevention of DoS attacks have proven to be more critical than strict performance metrics in selecting DHT identifiers, topology, and routing algorithms. Where a central authority exists, certificates can be stored in the overlay and allow more efficient DHT algorithms to be used. We explain how security and routing schemes can help preserve the integrity, scalability, and performance of P2PSIP communication Systems.

1. Introduction

Communications systems are a natural application for P2P technology. SIP [17], the dominant standard for VoIP and IM, already supports intelligent endpoints capable of end-to-end media connections. The challenge is to replace SIP's server-based registration, naming, and security with a P2P-based distributed location service. Designing a P2P-based communications system, particularly one developed on top of open-standards and intended for the open Internet, presents different service requirements than other P2P applications. In developing and deploying such a system [5, 20], we first identified the unique requirements for a P2P communications system, then determined what optimizations needed to be made to supporting algorithms and mechanisms.

1.1 P2P Communications Requirements

In filesharing applications, many peers may offer copies or versions of a file, and the identity of the provider is often deliberately obfuscated or unimportant. Except for special purpose systems, communications systems typically require that a unique individual be contacted. The lack of anonymity and the availability of the desired resource (person) at only a single location, coupled with a need for high reliability for communications, leads to our first requirements:

- The P2P algorithm must return the location of a resource, if present. False negatives are unacceptable.
- The user-agent must be able to verify the integrity of the resource and identity of the remote party.

The requirement to avoid false negatives eliminates many unstructured techniques, strongly suggesting a DHT or other structured P2P algorithm be used. Furthermore, the algorithm must contain sufficient protection against attackers manipulating the overlay routing to gain control of a particular portion of identifier space in order to prevent DoS attacks against targeted users. While also a problem, false positives are more easily handled at the application (SIP) level, and do not require DHT support.

Meeting the integrity and identity requirement requires a security mechanism that allows peers to sign resources, verify the signatures on resources, and the identity of remote parties.

- The system must provide connectivity between NATed peers and include them in the DHT when possible.

Deploying an Internet-scale P2P communications systems dictates that it must support the full range of deployed

networking technologies. This includes consumer broadband network users and businesses located in many countries. Unlike universities, these endpoints are frequently behind Network Address Translators (NATs), introducing the problem of non-transitive connectivity. Freedman et al. [9] explored some issues of Non-Transitivity in a PlanetLab based system, where the cause of these non-transitive connections is generally ephemeral problems or difficulties in Internet1 systems communicating with Internet2 systems. In our Internet-scale system, non-transitive connectivity caused by NATs is the rule, rather than an exceptional failure condition. Because we support deployment scenarios where the majority of devices are behind NATs, we must utilize them as peers in the DHT rather than relying on a sufficient number of super-peers on the public Internet.

- The DHT algorithm must not amplify DoS attacks.
- The DHT algorithm must prevent an attacker from gaining control of a particular portion of the identifier space by manipulating Peer-ID assignment.

As with any other publicly deployed P2P network, the overlay must guard against malicious peers and be resilient against DoS attacks. In particular, a P2P communications network is subject to both general DoS attacks against the entire overlay and DoS attacks targeting specific users.

1.2 Contributions

In this paper we present the approaches we have developed to address the practical issues of deploying a fully functional P2PSIP communications system. An implementation of this system is in use commercially, and this system has formed the foundation of a leading proposal for the IETF P2PSIP WG's standard. In particular:

- Despite emerging standards, securing a system using NATed peers remains challenging, as the mechanisms proposed by most DHTs for generating Peer-IDs facilitate several attacks. We demonstrate that, although hashing can be made more secure, providing provable security in light of an attacker's capabilities is unlikely to be successful. In particular, NATs and the adoption of IPv6 impose constraints on Peer-ID generation for Internet-wide deployments, as discussed in Section 3.1. Section 3.2 presents a certification authority (CA) based solution to these problems. We show that such a system can achieve security properties for a P2P system comparable with SIP's server-based security, using no centralized operations after a one-time enrollment.
- The routing scheme employed by a P2P algorithm impacts both its performance and resistance against attacks. In Section 4 we review the characteristics of several routing techniques and discuss the costs, benefits, and selection criteria of each. We analyze the NAT's influence on the cost of each algorithm, and show that the most efficient routing algorithms are also more vulnerable to attacks.
- We present an algorithm for locating STUN servers in an overlay that lacks a service provider to deploy and configure the necessary components for NAT traversal.

2. Background

Based on our requirements, we focus on structured P2P networks. In particular, we focus on extensions to the Chord algorithm [23], although most of our discussions apply to any DHT-based solution.

The majority of communications systems, including the traditional phone network, have been based on client-server principles, with more recent architectures moving some, though not all, of the functionality to the endpoints.

2.1 P2P Communications Systems

SIP is sometimes called a P2P application because much of the intelligence resides in the endpoints and media flows directly between them. SIP also allows endpoints to communicate directly to enable certain call features, but centralized registrars and proxies are generally required for registration, routing, security, and presence. Current work to create "P2PSIP," including this work and the efforts of the IETF P2PSIP working group, use the term P2P in the more traditional academic sense of eliminating or significantly reducing the role of servers. In addition to endpoints communicating directly, run-time services for resource location and NAT traversal, traditionally centralized in SIP, are handled in a P2P fashion.

The best known P2P communications system is the application Skype [21], which offers free computer-to-computer calls and charges for computer to PSTN (Public Switched Telephone Network) calls. Skype is a closed system, using encrypted messages and obfuscated binaries, therefore what is known about it has been discovered by reverse engineering [2, 3]. Skype is a hybrid system, relying on centralized authentication servers to establish identity at log-in and when calls are placed. Central servers also appear to sometimes be used for resource location. Skype uses a super-peer architecture of media relays to enable NAT traversal, and any peer can be elevated to a super-peer at any time without the user being aware. While there have been some licensed third parties, Skype's proprietary nature means few others create Skype clients.

Around the time Skype was launched, early work in P2PSIP was started. This early work led to our own

SOSIMPLE [6] and Columbia's SIPpeer [19] emerging as research projects aimed at removing the central servers from a SIP architecture. Over time, SOSIMPLE has evolved into the RELOAD P2PSIP peer protocol.

The RELOAD peer protocol defines how peers communicate for DHT maintenance, resource management, authentication, and overly routing. This peer protocol is modular, allowing various DHTs, routing mechanisms and even security types to be used with the base protocol. RELOAD uses a light-weight binary protocol to exchange DHT routing information between peers. The baseline DHT proposed for RELOAD is a modification of Chord. RELOAD provides primitives that allow a variety of routing techniques to be employed, support for standards-based NAT traversal between all peers, and security mechanisms intended to secure DHT maintenance and routing messages, as well as to facilitate P2P security of the communications signaling and media channels. For communications session establishment, the SIP protocol is used with minimal, backward-compatible modifications. As with conventional SIP, RTP [18] is used for media transport. RELOAD's flexibility enables the appropriate security and routing schemes to be selected to meet the security and performance goals of various P2PSIP application scenarios [4].

3. Securing the System Using Certificates Issued by an Offline CA

In this section we explain why IP address hashing as a mechanism for Peer-ID generation is insecure, and show that a credential-based system to authenticate the user and establish identity addresses a large number of security problems for a large-scale, publicly accessible DHT. The mechanism presented requires the CA to be consulted only at the time the peer first joins the overlay and not each time the user logs in or needs to verify a user or resource.

3.1 Problems with IP-based Peer-IDs

Due to a lack of IPv4 addresses, the Internet relies on NATs to establish a multi-level network. Behind a NAT, a number of devices use private addresses drawn from a pool of IP addresses reserved for this purpose. A single or small number of public addresses is shared among the devices by mapping available public ports to the devices behind the NAT. NATs create particular challenges for any applications that require connections directly between endpoints—such as SIP or P2P applications—because the internal devices cannot use their local IP address as an identity on the public Internet. Techniques to provide connectivity in such environments are documented in [8, 10], and the IETF is working to standardize NAT traversal techniques in the form of STUN, STUN relay, and ICE [16]. Using these techniques, basic connectivity can be achieved, but without readily identifiable global IP addresses, security in the overlay becomes a challenge.

Many DHTs propose hashing the IP address of the peer, neglecting the port, to generate a Peer-ID. Devices behind different NATs may erroneously believe they have the same (actually private) address. If techniques such as STUN are used by the devices to obtain the public address of the NAT, multiple devices behind the same NAT may use this address to generate a Peer-ID. In both cases, duplicate Peer-IDs will be produced.

Combining the public address and a port number can ameliorate the problem, but does not protect against malicious peers. Appending the port number to the address before hashing produces a different Peer-ID for each peer behind the NAT, but enables a placed location attack against the overlay, as an attacker can simply hash each of the 64K host/port pairs for that NAT's global IP address in advance, and select the closest Peer-ID to mount the attack. Replacing the least significant bits with the port number after hashing forces the possible IDs to be contiguous, reducing the area of DHT space that can be attacked, but has another problem. Chord places redundant storage of information on sequential peers, meaning an attacker can compromise all replicas of a particular resource. While this problem too can solved with alternate replication schemes (for example by placing replicas by appending replica keywords before hashing resources), it points to a broader issue of peers self-generating IDs.

Generating Peer-IDs based on hashing of a value that can be controlled in any way by the end user is inherently insecure. Attackers with access to many Peer-IDs will be able to mount attacks relatively easily, and port choice or possessing many IP addresses (through the use of bot nets or simply as a result of a move to IPv6) enable this attack.

3.2 Securing the System with Certificates

We propose a system where both Peer-IDs and user identities are secured using a public-key certificate model. Small scale ad-hoc systems, particularly those limited to a small number of trusted users, may still use an approach incorporating hash-based Peer-ID generation and a shared secret, but our deployments have shown that such systems are not practical for Internet-scale deployments. Our proposed certificate-based mechanism solves the following problems:

- In a communications system, individuals need to be reachable in a deterministic and repeatable way. Two conversations with "Alice" must be with the same individual. Accordingly, a user must be able to uniquely assert their identity over an extended period spanning multiple sessions. Repeatable identity is solvable with-

out a central server, but verification of initial identity is difficult without a centralized mechanism.

- Communications systems require messages (text or other media) to be exchanged between users, and for offline users, stored for future retrieval. Because messages may be stored or relayed by arbitrary peers, all content must be encrypted.

- While not unique to communications systems, Sybil attacks [7] involving one entity attempting to masquerade as multiple peers in the overlay must be addressed. Certificates issued by a central authority employing a rate-limiting technique can help detect and mitigate these attacks by restricting the number and range of IDs a peer may assume in the overlay.

This CA based approach differs significantly from a hybrid approach like Skype, where the authentication server is contacted frequently to verify users. The centralized certificate authority (CA) we propose is not involved on every transaction, but only the first time a user (or peer) wishes to join the overlay.

The server generates two types of certificates:

- User certificates assert that a user is valid and owns the specified username. The server ensures the uniqueness of usernames.

- Peer certificates asserts that a peer is authorized to join the system and has the Peer-ID specified in the certificate. This Peer-ID is a unique, random number generated by the server, and is not tied to IP.

A user must obtain both a user and peer certificate from the CA before joining the system. We allow the certificates to be separate, since a particular device, for example an IP phone, might have multiple users or "lines," requiring several User-IDs to be asserted, but only one Peer-ID. Similarly, a user may be present on multiple devices, requiring multiple Peer-IDs but only one User-ID. Finally, devices such as gateways, which allow VoIP calls to terminate onto the PSTN, might be peers but not be associated with a particular user. In addition, each peer in the system obtains a copy of the CA's root certificate.

The public portion of the certificate for each user is stored persistently in the overlay, ensuring they are available even if the user is offline. The certificate is used to encrypt media sessions as well as to secure messages (voicemail) left for the user. Because the chain for each certificate can be traced to the well-known, shared CA root, the authenticity of any certificate can be verified without contacting the central server.

3.3 DHT Operations with Certificates

When a peer attempts to join, it must present a valid certificate for the asserted Peer-ID, which is verified by the peer currently responsible for that region of the overlay. By limiting the rate at which certificates are generated, the CA can reduce the effectiveness of a Sybil attack. Rate limiting mechanisms can include a minimal charge or requiring a valid credit card number that while not charged, can only be used to obtain an ID once in a given period, perhaps 24 hours. As a result, obtaining the very large number required to corrupt an overlay would be costly or impossible. Because the Peer-IDs are randomly selected by the CA, it is very difficult mount a placed location attack against the overlay.

Signatures provide two important benefits to the system:

- If an attacker does obtain a legitimate ID and launches a flooding attack, the sender's signature allows the attack to be detected, the malicious peer to be identified, and messages from that peer discarded.

- A peer sending a request can confirm that a response is delivered from the responsible peer, rather than a compromised peer along the route, by verifying the signature of the response. This allows such "man-in-the-middle" attacks to be identified, and alternate routing can be used to circumvent them.

Furthermore, a user storing the address of the device on which they can be contacted must sign the registration, incorporating an expiration time. A peer retrieving the registration does not need to trust the peer that stored the message, as they can directly verify the signed registration. An attacker, either the storing peer or a compromised peer along the query's route can fail to return a registration or provide an outdated (but validly signed) registration, but cannot spoof one. By simultaneously querying several replicas of the resource and employing a comparison algorithm as discussed in [15, 14], the severity of these types of attack can be reduced.

In spite of questions about the cost of public key security [22], creating a new signature is necessary only when establishing a new connection and registering a resource. Encryption can be used after the initial connection, and messages can be routed to their destination without hop-by-hop validation. Therefore, devices with low CPU power can still participate in such an overlay.

4. Selecting a Routing Technique

The choice of routing algorithm—iterative, recursive, etc.—might at first appear to be merely a question of the number of hops taken by a message. However, if we choose

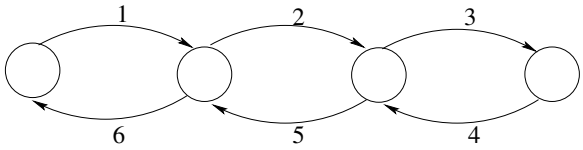

Figure 1. A Recursive Routing Flow

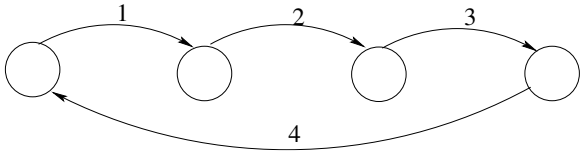

Figure 2. A Semi-Recursive Routing Flow

to utilize peers behind NATs, we must select an algorithm that allows them to function efficiently in the overlay, while considering how the algorithms protect against DoS attacks from compromised peers.

With recursive routing, the initiator issues a message to the peer it is aware of nearest the target. If the recipient peer is not the target, the message is forwarded to the nearest peer the recipient is aware of, and the process repeats until the target is reached.

Recursive algorithms can route the response in three ways. In a true recursive algorithm (Figure 1) the response is returned simply by reversing the original path. Many recursive algorithms shortcut by sending the response directly back to the initiator, an approach termed semi-recursive routing (Figure 2). A response can also be routed forward-only, with the response being routed back to the initiator in the same manner a new message from the responder to the initiator would be routed.

In contrast, with iterative routing (Figure 3) the peer receiving the message replies suggesting a nearer peer, rather than forwarding the message. The initiator then sends a new request to the recommended (and nearer) peer, repeating until the target is reached.

The first column of Table 1 lists the total number of messages passed in a message traversing i peers (including the initiator and target). The semi-recursive technique is the most efficient in terms of basic message count, but as we discuss further, there are some drawbacks, particularly in the presence of NATs.

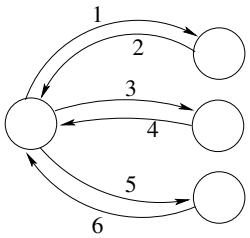

Figure 3. An Iterative Routing Flow

4.1. Implications of NATs in the Overlay

One of our primary functional requirements is that the overlay be able to function—and provide good service—when the majority of peers are behind NATs. This requirement dictates that we select routing algorithms that allow peers to participate in the overlay even when non-transitive connectivity exists because of NATs.

We use ICE [16] to establish connections in the presence of NATs. ICE relies on a rendezvous service to exchange information between the two devices attempting to form a direct connection. When used in a P2P overlay, the two peers can form a direct connection by relying on the overlay to route the ICE offer/answer messages that allow them to establish a direct connection. Using ICE to open a new connection requires the round-trip open request/response routed along the overlay plus at least one pair of STUN request/response messages sent directly between the two peers in each direction.

For our analysis in this paper, we assume that the connections to immediate neighbors are established in advance and keep-alives occur during periodic DHT maintenance, therefore we look only at the cost of contacting a peer not in our set of neighbors. Ultimately the cost and reliability of NAT traversal depends on the proportion of users behind NATs and the types of NATs they are behind. The distribution, and thus cost and success rates of various routing algorithms, varies based on the user population of the overlay. For example, college students in dorms typically have IP addresses on the public Internet, whereas most residential broadband customers are usually behind one or more P2P-friendly (endpoint-independent mapping) [8, 1] NATs. P2P-friendly NATs use the same public IP:port pair for all outbound traffic from a particular IP:port pair behind the NAT. Corporate networks are almost always behind NATs and frequently behind non P2P-friendly NATs.

Using recursive routing, messages are proxied through the overlay and no new connections must be opened through NATs. Forward-only routing is similar, as only established connections are used.

Semi-recursive routing attempts to deliver the response directly to the querying peer. If the peer is behind a NAT that allows the traffic through, then the direct response is more efficient, but otherwise, the direct response requires an ICE exchange to open the connection. The cost in column 3 of Table 1 is derived by counting the normal i messages, plus a round-trip recursively routed ICE open request ($2(i-1)$ hops) plus the STUN connection checks (4).

A possible optimization would be to allow the direct re-

Table 1. Comparison of costs associated with various routing algorithms

Algorithm	Messages	New Connections Established	Messages Counting NAT Traversal	DoS Risk	Messages Processed by Interior Peers
Recursive	$2(i-1)$	0	$2(i-1)$	Yes	4
Semi-recursive	i	1	$i+2(i-1)+4$	Yes	2
Forward-only	$2(i-1)$	0	$2(i-1)$	Yes	2 to 4
Iterative	$2(i-1)$	$i-2$	$2(i-1)+8(i-2)$	No	2

sponse when the querying peer has a reason to believe that a direct response is possible (not behind a NAT or firewall) or when a direct connection has previously been established between the two peers. Detecting the first case is complex and prone to failure [12]. In the second case, if a direct connection was previously established the query would have been routed along that path in the first place, thus the case degenerates into simple recursive routing.

For an iterative algorithm, many more connections must be established. While the first peer where a message is sent is the direct neighbor of the source, all subsequent connections may be with non-neighbors, and a NAT connection must be opened for each. In all, as many as $i-2$ new connections may need to be established. By routing the ICE open request through the referring peer, each ICE request/response will take exactly 4 hops, with the 4 additional STUN connection check messages. This cost makes iterative routing infeasible in overlay networks where significant numbers of peers are behind NATs.

Thus, we conclude that recursive or forward-only routing is most effective in overlays with peers behind NATs.

4.2. Routing in Unstable Overlays

We now extend our analysis to unstable overlays. While peer failures induce instability, a peer joining the overlay (either during initial joining or while healing a previous link failure) can also introduce momentary instability. In both cases, the admitting peer is communicating with a peer that is not currently reachable via overlay routing.

The first conclusion of this challenge is that forward-only routing does not work in an unstable overlay. For example, the response from an admitting peer to a joining peer cannot reach the joining peer if it is forwarded around the overlay because the joining peer is not yet reachable via overlay routing.

The second conclusion is the requirement that there must be some state used in recursive routing, whether it is each peer storing the peer from which it received the message or adding a "Via-List" to the message itself. Overlay topologies with bi-directional routing also do not address this problem because if a peer is not yet fully reachable via overlay routing, only the specific path followed by the original message will reach the querying peer.

In conclusion, to deal with unstable overlays, only a true recursive routing algorithm is both reliable and reasonably efficient in an overlay relying on peers behind NATs. Establishing direct connections with peers with whom messages are frequently exchanged (such as resource queries or updates) is a useful optimization that does not affect reliability. Utilizing direct responses (semi-recursive routing) or iterative routing when NATs do not block particular direct communication is feasible, but detecting when such operations will complete and when they will fail is sufficiently complicated to create doubt that a reliable routing protocol could be formed around such rules.

4.3. Locating STUN servers

A complexity of relying on ICE for NAT traversal in P2P environments is that we cannot assume that the overlay is well-provisioned with STUN servers on the public Internet. One option is to use a service discovery algorithm, such as ReDiR [13] in which all peers that believe themselves to be on the public Internet without being firewalled register themselves as STUN servers. This approach is feasible, but has a few drawbacks:

- There is no guarantee the overlay is on the public Internet. It might be on a large internal network that is entirely within the 10.x.x.x address prefix, for example.
- To facilitate NAT traversal in multi-level NAT topologies, a peer should identify a STUN server at each level of NAT. Solving this problem with a service discovery algorithm would require both the ability to identify a multi-NATed topology from the middle layer and the ability to discover other peers based on their specific location.

Fortunately, there is no requirement that a STUN server be on the public Internet. The NAT in front of the STUN server does not change the address of an incoming request. Furthermore, all peers have a STUN service on their P2P ports, because all peers implement ICE for NAT traversal. As a peer forms connections with new peers, it will send STUN queries to each as a part of ICE. To identify useful

peers to serve as STUN servers, the peer simply groups the other peers it has contacted into sets indexed by the reflexive address (IP port pair the NAT assigns them on the public Internet) they return to STUN queries.

Applying this algorithm, almost all peers will learn only two reflexive addresses: their local interface and the reflexive address allocated to them by their NAT. In the case of multi-level NATs, they may learn multiple addresses, but the addresses will be fixed and converge to a small number of sets.

Only in the case where a NAT without endpoint-independent mapping (non P2P-friendly) is in its outbound path to other peers will the set of reflexive addresses grow unbounded. In this case, the peer should not collect all addresses and should not advertise them as ICE candidates because they are unlikely to be of any use for other peers.

Unfortunately, without a service provider carefully provisioning a NAT at each level of NATing in the topology, there is no deterministic way to identify a reflexive address at each level of the NAT topology. Our algorithm allows a peer to learn those addresses it can in a probabilistic manner: if it happens to contact a peer that gives it a new reflexive address it will remember that peer and use it as a STUN server for future transactions.

5. DoS Attack and Parallelization Risks

Recursive routing allows for peers behind NATs to participate in the overlay, but introduces the hazard of DoS amplification attacks. Recursive algorithms allow peers to serve as surrogates in DoS attacks against a particular target. In Figure 4, we assume an attacker A can send requests to intermediate peers I searching for target peer T. With iterative routing, the intermediate peer I replies with a closer peer to use to attempt to reach T, but the attacker must directly send to T to attack it. In the recursive approach, attacker A can send to many intermediate peers I, all of which relay the message to T. The intermediate peers essentially serve as DoS surrogates for A and may assist in obfuscating the source of the attack. The ability of A to multiply the attack is limited only by the number of intermediate peers I that are reachable from A. If we take Chord as an example, each peer has $lg(m)$ immediate neighbors, where $m = 160$ is the size of the address space. If the attacker sends to every neighbor, $lg(m)$ messages will reach the target. Ironically, sending parallel messages to a single peer using different paths also can be used as a defense against routing attacks [22].

Another problem with recursive algorithms is that combining them with parallelization can further aggravate the DoS Attack. Some iterative DHT algorithms, such as EpiChord or Kademlia [11, 13], improve average case performance using parallel search, where each initiator sends the request to k intermediate peers, and the algorithm parallelizes by sending the request to an average of p neighbors at each level of the search. Combining parallelization with a recursive algorithm creates an explosive DoS multiplication attack risk. In a DHT where queries have a worst case depth of $lg(N)$ (where N is the number of peers in the system), and all messages are delivered to the target, the number of messages M_T arriving at the target T can in the worst case be $M_T = k\ p^{lg(N)}$. For even small values of k and small branching factors such as the $p = 3$ used by EpiChord, taking a recursive approach is disastrous. If we assume $N = 100,000$, and $k = 3$, we have $M_T \approx 3\ 3^{16.6} \approx 250$ million. Techniques can be applied to detect and drop duplicate messages, but the target may still receive N messages.

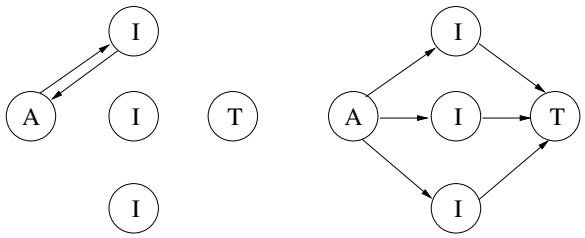

Figure 4. Surrogate Attackers in a Recursive Approach

In a generic DHT, the fact that intermediate peers have relayed the messages can make determining who sent the message impossible, leaving no simple way to find the attacker. Signing of the overlay messages, including signing each entry in the "Via-List" can prevent obfuscation, but cannot prevent the attack completely. In essence, the target of the attack would be able to determine who had attacked them, however such an approach is computationally expensive, requiring that certificates are checked far more frequently than we propose in our system. Exploring how distributed reputation services and certificates might be combined to reduce the impact of this problem is an area of future work.

Our results show that there is no clear solution for a routing algorithm that is both efficient in the presence of NATs and does not present an opportunity for a DoS amplification attack. We are studying other solutions, such as limiting the depth of recursion or expending more effort to determine when iteration can be used without excessive NAT traversal. Regardless, this challenge causes us to reiterate our belief that certificate-based security is necessary in spite of its costs. Ultimately, in an open Internet-scale systems where peers cannot be trusted, despite its cost, iterative routing must be used to defeat DoS amplification attacks. A more advanced system might determine which approach to use at

run-time depending on its current load.

6. Conclusion

We have presented solutions to several of the problems we have encountered moving DHT-based P2P communications systems from the lab to a real-world setting with combinations of residential, commercial, and academic systems. In short, the P2P algorithms must be chosen based on the control exerted over participation in the network.

- If a CA-based system can be used to establish identity and monitor peer additions, and peer behavior can be monitored (such as in a closed system) then routing algorithms can be chosen based on efficiency (generally based on the prevalence of NATs and firewalls in the environment) and Peer-IDs can be assigned with a simple uniform random distribution. However, if policing of peer behavior is unavailable or peers are easy to add, iterative routing must be used.

- For system with open enrollment, the routing choices are more restricted. As the ease of obtaining Peer-IDs and challenges of monitoring the network increases, compromised peers become more likely and more efforts must be made to defend against attacks. A gradual transition from recursive to iterative routing, such as limiting the number of recursive hops, is possible, as are more active techniques to identify and remove compromised peers. Furthermore, replication schemes and alternative routing techniques become more important as the proportion of compromised peers grows.

- Ultimately, combining a certificate-based approach with iterative routing is not the fastest, but is the most resilient to attack approach for a global-scale deployment. By combining those techniques with other work to identify and revoke the IDs of compromised peers, such a system can be used as a reliable means of communication.

References

[1] F. Audet and C. Jennings. RFC 4787 - Network Address Translatoin NAT Behavioral Requirements for Unicast UDP. `http://www.ietf.org`, Jan. 2007.

[2] S. Baset and H. Shulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *INFOCOM2006*, Apr. 2006.

[3] P. Biondi and F. Desclaux. Silver Needle in the Skype. Presentation at Black Hat Europe 2006, Feb. 2006.

[4] D. Bryan, E. Shim, B. Lowekamp, and S. Dawkins. Application scenarios for peer-to-peer session initiation protocol (p2psip), Nov. 2007.

[5] D. Bryan, M. Zangrilli, and B. Lowekamp. draft-bryan-p2psip-reload-01, July 2007.

[6] D. A. Bryan, B. B. Lowekamp, and C. Jennings. SOSIMPLE: A serverless, standards-based, P2P SIP communication system. In *Proceedings of the 2005 International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA 2005)*. IEEE, 2005.

[7] J. R. Douceur. The Sybil attack. In *Proceedings of the IPTPS02 Workshop*, Cambridge, MA, USA, Mar. 2002.

[8] B. Ford, P. Srisuresh, and D. Kegel. Peer-to-peer communication across network address translators. In *Proceedings of USENIX 2005*, pages 179–192, Anaheim, CA, April 2005.

[9] M. Freedman, K. Lakshminarayanan, S. Rhea, and I. Stoica. Non-Transitive Connectivity and DHTs. In *Proceedings of the 2005 Usenix Workshop on Real, Large-scale Distributed Systems (WORLDS05)*, 2005.

[10] S. Guha, Y. Takeda, and P. Francis. NUTSS: A SIP-based Approach to UDP and TCP Network Connectivity. In *Future Directions in Network Architecture (FDNA-04)*, SIGCOMM '04 Workshops, Portland, OR, Aug. 2004.

[11] B. Leong, B. Liskov, and E. Demaine. Epichord: Parallelizing the chord lookup algorithm with reactive routing state management. In *Proceedings of the 12th International Conference on Networks (ICON 2004)*, Nov. 2004.

[12] D. MacDonald and B. Lowekamp. draft-ietf-behave-nat-behavior-discovery, Nov. 2007.

[13] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the XOR metric. In *Proceedings of IPTPS02*, Mar. 2002.

[14] A. Muthitacharoen, S. Gilbert, and R. Morris. Etna: A fault-tolerant Algorithm for Atomic Mutable DHT Data. Technical Report MIT-LCS-TR-993, MIT-LCS, June 2005.

[15] R. Rodriques and B. Liskov. Rosebud: A scalable byzantine-fault-tolerant storage architecture. Technical Report TR/932, MIT CSAIL, Dec. 2003.

[16] J. Rosenberg. draft-ietf-mmusic-ice-19, Oct. 2007.

[17] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC 3261 - SIP : Session initiation protocol, June 2002.

[18] H. Shulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889 - RTP: a transport protocol for real-time applications. `http://www.ietf.org/rfc/rfc1889.txt`, Jan. 1996.

[19] K. Singh and H. Schulzrinne. Peer-to-peer internet telephony using sip. In *Proceedings of the 2005 Network and Operating System Support for Digital Audio and Video (NOSSDAV 2005)*, 2005.

[20] SIPeerior Technologies, Inc. `http://www.SIPeerior.com/`.

[21] Skype Technologies, S.A. `http://www.skype.org/`.

[22] M. Srivatsa and L. Liu. Vulnerabilities and security threats in structured overlay networks: a quantitative analysis. In *Proceedings of Computer Security Applications Conference*, 2004.

[23] I. Stoica, R. Morris, D. Karger, M. Kaashock, F. Dabek, and H. Balakrishman. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Neworking*, 11(1), 2003.