AP3: Cooperative, decentralized anonymous communication

Alan Mislove[†] Gaurav Oberoi[†] Ansley Post[†] Charles Reis[‡] Peter Druschel[†]
Dan S. Wallach[†]

[†] Rice University, Houston, TX, USA [‡] University of Washington, Seattle, WA, USA

Abstract

This paper describes a cooperative overlay network that provides anonymous communication services for participating users. The Anonymizing Peer-to-Peer Proxy (AP3) system provides clients with three primitives: (i) anonymous message delivery, (ii) anonymous channels, and (iii) secure pseudonyms. AP3 is designed to be lightweight, low-cost and provides "probable innocence" anonymity to participating users, even under a large-scale coordinated attack by a limited fraction of malicious overlay nodes. Additionally, we use AP3's primitives to build novel anonymous group communication facilities (multicast and anycast), which shield the identity of both publishers and subscribers.

1 Introduction

In anonymous communication, the identity of the sender or the receiver involved in an information exchange remains hidden. There are many legitimate reasons why the parties involved in an information exchange might wish to remain anonymous. For instance, a user who wishes to gather information on a medical condition might wish to remain anonymous to protect his privacy and avoid embarrassment or economic disadvantage. Citizens who voice criticism of a repressive government wish to remain unknown to avoid prosecution. An employee reporting abuses within a corporation needs to protect his identity to avoid exposure as a "whistleblower". Voters involved in an on-line election should remain anonymous to ensure their vote reflects only their conscience. Finally, in decentralized systems, auditing is an effective mechanism to enforce the system's policies [12]; however, for the audit to be effective, the auditor's identity often has to remain hidden from the one being audited.

Different applications require very different guarantees regarding the degree of anonymity. In this paper, we use the terminology defined by Reiter and Rubin [15] to describe levels of anonymity. For instance, a "whistle-blower" might require *beyond suspicion* anonymity, where he is no more likely to be the informant than any other employee. In an election, on the other hand, *probable innocence* anonymity may suffice, where the probability that a given citizen cast a certain vote is less than the probability that the citizen did not cast the vote. Finally, for distributed auditing, it is often enough that the identity of an auditor cannot be ascertained, an anonymity level known as *possible innocence*.

Our system, AP3, provides a cooperative, distributed anonymous communication service. AP3 is completely decentralized, self-organizing, it does not require any trusted nodes to provide anonymity and it scales to large and dynamic groups of participants. It is designed to provide at least *probable innocence* for the participating users without requiring a dedicated or trusted infrastructure. Additionally, AP3 is designed to maintain probable innocence even under a large-scale coordinated attack by participating nodes. For example, we will show that even under an attack consisting of 20% of the network conspiring to destroy anonymity, AP3 still provides probable innocence to non-malicious nodes while only incurring an overhead of an expected four extra forwarding hops, regardless of network size.

The AP3 service allows users to communicate anonymously by providing three simple primitives: (i) anonymous message delivery, (ii) anonymous channels, and (iii) secure pseudonyms. Building on these primitives, users are able to send and receive unicast, multicast and anycast messages anonymously. Additionally, users can create secure persistent pseudonyms, allowing them to build a reputation under a recognizable pseudonym while protecting their real-world identity. This may be useful, for instance, to a corporate whistleblower or a "mole" in a position of power, who may not want to reveal his or her identity but wishes to engage in a dialogue with the public, the press or judicial authorities.

The outline of the rest of this paper is as follows. Section 2 discusses background material, including p2p overlays and end system multicast. Section 3 describes the design of AP3 in detail and analyses the level of anonymity that AP3 provides. Section 4 discusses how anonymity can be extended to multicast. Section 5 outlines related work, and Section 6 presents our conclusions.

2 Background

Structured peer-to-peer overlays [11, 13, 17–19] provide a self-organizing, scalable and fault tolerant substrate for cooperative peer-to-peer applications. In such overlays, every node and every object is assigned a unique identifier, referred

to as a *nodeId* and *key*, respectively, which is chosen from a large, sparse identifier space. Each key is dynamically mapped to one of the live nodes, such that the number of keys mapped to each node is statistically balanced. Given a message and a key, these overlays efficiently route the message to the node whose nodeId is numerically closest to the key. Generally, such overlays maintain $O(\log N)$ state and provide routing paths of $O(\log N)$ expected hops, with N where N is the number of nodes in the network.

One type of system built on such overlays is end-system multicast (ESM) [2, 3], where hosts on the edge of the network form a multicast tree and provide multicast services using only the unicast service provided by the network layer. This is in contrast to conventional network-layer multicast, such as IP multicast [4], where the IP routers form a multicast distribution tree. A number of cooperative ESM systems have been designed based on structured overlays [2, 20]. In Scribe [2], each group has a 160 bit *groupId*, which serves as the address of the group. The current subscribers to each group form a multicast tree, which consists of the Pastry routes from all group members to the node that is currently responsible for the groupId. Scribe supports large numbers of groups based on the same overlay, group sizes ranging from one to all participants, and highly dynamic groups. Proximity neighbor selection [9] lends Scribe low link stress and low delay stretch [2].

3 Design

In this section, we describe the architecture of AP3 and discuss each of the primitives that AP3 provides: (i) anonymous message delivery, (ii) anonymous channels, and (iii) secure pseudonyms. AP3 is built on top of Pastry [17], but could in principle be implemented on other structured p2p overlays as well. Additionally, AP3 is designed to require very little extra processing when a node joins or leaves the overlay, which means that AP3 can support networks with relatively high rates of node churn. Throughout the paper, we assume a defense against the Sybil Attack [6], such as the one presented by Castro et al. [1].

3.1 Anonymous Message Delivery

Our strategy for providing anonymous message delivery is similar to that implemented by Crowds [15] and Tarzan [8], in that it relies on a network of peers to forward messages attempting to hide the originator. In AP3, a node along the request path does not know whether the node from which it received a message is the message's originator or simply another forwarding peer. Consequently, the destination of the message only learns the identity of the peer that handed it the message.

When a node wishes to anonymously send a message, it first creates an anonymous request object comprised of the message itself and the address of the intended recipient. Obviously, the message must not contain any information that can reveal the originator's identity; if a user gives himself away all anonymity properties are lost. This request is then forwarded to a node in the overlay selected by drawing a random key. The underlying routing substrate ensures efficient delivery to the node responsible for this key. Upon receiving a request, an AP3 node performs a weighted coin toss to decide whether to fulfill the request and send a message to the intended recipient, or to forward the message to another randomly selected peer. The decision to forward is made with probability p_f, the *forward probability*. This mechanism essentially provides a random path through the p2p network built from a variable number of random hops. It obscures the originator's identity from both the intended recipient and any malicious peers hoping to expose the originator's identity. Figure 1 below shows an example of anonymous message delivery.

If the weighted coin flip determines that the node should forward the message to another node, the node first chooses a random key k in the id space, using a secure random number generator. However, the node cannot simply use overlay routing to send the message to the node nearest k. Doing so would allow the node's overlay neighbors to observe all of the node's forwarded messages and facilitate a traffic analysis attack. Instead, the node first determines the current live node n closest to k by routing a lookup request with the target k. Once n responds to this lookup, the node then forwards the anonymous message directly to n.

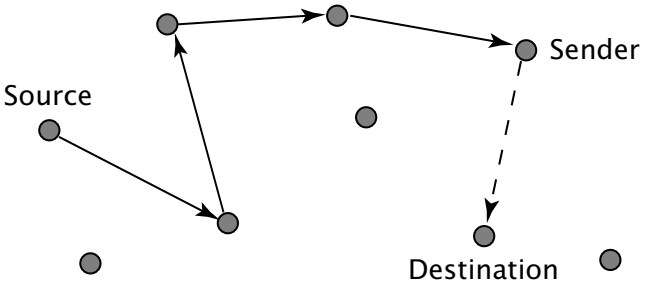

Figure 1: Example of anonymous routing. The destination only sees the dashed part of the route, so the source of the request appears to be 'Sender'. Each node along the path performs a weighted coin toss to choose whether to forward the message or deliver it.

To provide probable innocence, p_f needs to be at least 0.5, otherwise the sender of a message is more likely than not the originator of the message. On the other hand, p_f clearly needs to be below 1 for the routes to be of finite length. We have determined values between 0.5 and 0.9 to be practical. The impact of the forward probability on performance and guarantees is discussed in detail in Section 3.4.

3.2 Anonymous Channels

While anonymous routing allows nodes to send requests without divulging their identity, anonymous routing alone is insufficient to support a request-response communication

in which the requester does not wish to divulge his identity. Since destinations receiving a message do not know the identity of the sender, they are unable to reply. In order to allow for this functionality, AP3 provides anonymous channels that allow a node to specify a return location for a message without divulging their identity.

When a node wishes to construct an anonymous channel, it first picks a random id, the address of the channel. Messages sent to this channel id are then forwarded anonymously back to the receiver, and nodes who send messages to the channel are unaware who is the actual recipient. Thus, if a node wishes to anonymously send a request and receive a response, it first creates an anonymous channel and then includes the address of the channel in the anonymously routed request.

To establish an anonymous path between the endpoint and the source, the source picks a random id L and then establishes a path by sending an anonymous message through the network in the same manner as was described above. In this case, however, each node in the forwarding chain remembers the node from which it received the message in a local table called the *forwarding table*. The message is eventually delivered to the node closest to L, the endpoint, which in turn constructs the channel by agreeing to forward any messages sent to L back along the anonymous path. Using this mechanism, anonymity is preserved as no node along the channel know if the previous node is the originator of the channel or just another intermediate node. An example of an anonymous channel is shown in Figure 2.

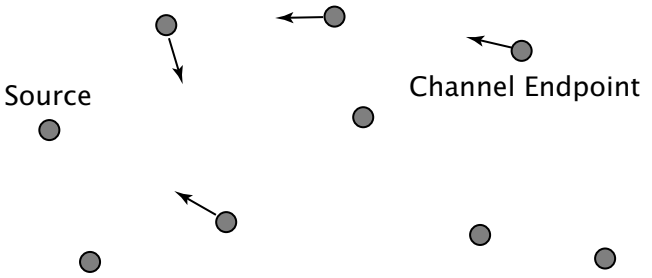

Figure 2: Example of anonymous channels. Nodes maintain back pointers along the anonymous path (shown as the short arrows), and the first node on the chain serves as entrance to the channel. Messages sent to the channel are forwarded back to the source node.

Additionally, when a path is established, the receiver specifies an expiration time that defines the period during which entries remain in the forwarding tables. Thus, forwarding table entries naturally expire over time. If a given channel has expired, the source node can simply create a new and different anonymous path to serve the anonymous channel.

The expiration time must be chosen taking into account the churn rate of the overlay network. As soon as one of the nodes along the channel leaves the network, the channel is unusable since messages sent to the channel will not make it back to the originator. The originating node must then periodically refresh the channel with a frequency on the order of the average node lifetime in the system, or risk not receiving messages sent to the channel.

3.3 Secure Anonymous Pseudonyms

AP3 allows users to have secure, persistent online identities that cannot be tied to a real-world identity. Providing persistent pseudonyms can be achieved by having users in the system generate public/private key pairs (K_{pub}, K_{pri}). Each key pair corresponds to one pseudonym, and users can easily generate more pseudonyms as required. Users can have different pseudonyms, such that receivers cannot tell that messages sent by the user under different pseudonyms are in fact from the same user. Note the no public key infrastructure (PKI) is needed; nodes are able to generate additional keypairs without contacting any central authority.

In order to allow other users to securely send messages to a pseudonym, the owner of a pseudonym establishes an anonymous channel at the location $H(K_{pub})$ where H is a secure hash function such as SHA-1. The node owning the pseudonym must also periodically refresh the anonymous channel associated with the pseudonym, since nodes along the channel may have died.

When another user wishes to communicate with the pseudonym, he first encrypts the message using the pseudonym's public key and then sends the message (anonymously, if desired) to the anonymous channel. This ensures that only the user who owns the pseudonym is able to read messages sent to it. In a similar manner, all messages which are sent from the pseudonym can be signed, which prevents other users from forging messages from the pseudonymous user.

3.4 Anonymity Guarantees

In order to analyze the anonymity guarantees that AP3 provides, let us assume for the time being that there is a system-wide forwarding probability p_f, and let us also assume that all nodes in the network follow the AP3 protocol (we will also consider the case of malicious nodes below). We will show that AP3 provides probable innocence for the originator with respect to all nodes along the anonymous path. Moreover, under the assumption that the destination does not conspire with a node along the path, AP3 provides anonymity beyond suspicion with respect to the destination.

Under these assumptions, the probability that an anonymous path is of length i is exactly $(1-p_f){p_f}^{(i-1)}$. A node receiving a message can assert that the previous node in the path is the originator with the same probability that a path is of length one, i.e. $(1-p_f)$. Similarly, the node can assert that the previous node is not the originator with probably p_f. This shows that for $p_f > 0.5$, AP3 provides probable innocence since the previous node on the path is less likely to be the originator than not. Additionally, since the originator of an anonymous message always forwards it at least one hop, the ultimate destination of the message knows that the node from which it received the request is no more likely to be the source than any other node. Thus, AP3 provides anonymity

beyond suspicion for the originator with respect to the destination, unless the destination conspires with a node along the anonymous path.

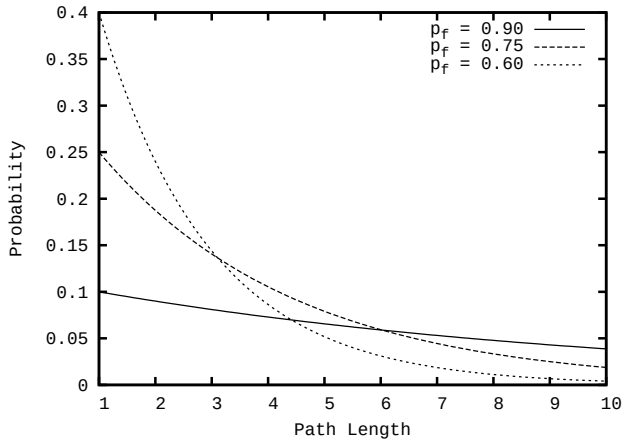

Figure 3: Distribution of path length probabilities with forward probabilities 0.6, 0.75, and 0.9.

The level of probable innocence anonymity provided by AP3 is directly proportional to the forwarding probability p_f. It can easily be seen that the average path length A is

$$\begin{aligned} A &= \sum_{i=0}^{\infty}(i+1){p_f}^i(1-p_f) \\ &= \frac{1}{(1-p_f)} \end{aligned}$$

which grows inversely proportional to the forwarding probability. This demonstrates the direct tradeoff between efficiency and the level of anonymity. The probability distribution of path lengths is shown in Figure 3, with forwarding probabilities of 0.6, 0.75, and 0.9.

AP3 is designed to provide anonymity guarantees even in the face of a large-scale attack by a coordinated set of malicious nodes. For simplicity, let us assume that a percentage f of all nodes are malicious, and that these nodes are evenly distributed throughout the network and in routing tables. In our analysis, we allow for the worst case attack where the malicious nodes work together and share information about routing requests, with the goal of uncovering the originator of a message. Figure 4 shows the path length distribution with 20% malicious nodes, assuming all malicious nodes misbehave by immediately forwarding requests to the destination rather than flipping a weighted coin.

Similar to the path length distribution equation above, the probability that an anonymous path is of length i is

$$[f+(1-f)(1-p_f)]{p_f}^{(i-1)}(1-f)^{(i-1)}$$

This shows that, even under a large-scale coordinated attack on anonymity involving 20% of all nodes and a forward probability $p_f = 0.75$, the group of malicious nodes can only assert that an incoming route request was originated by the previous node with 0.40 probability. Thus, it is still more likely that the request came from a different node than the one from which the malicious node received the message, which preserves probable innocence with respect to the path members.

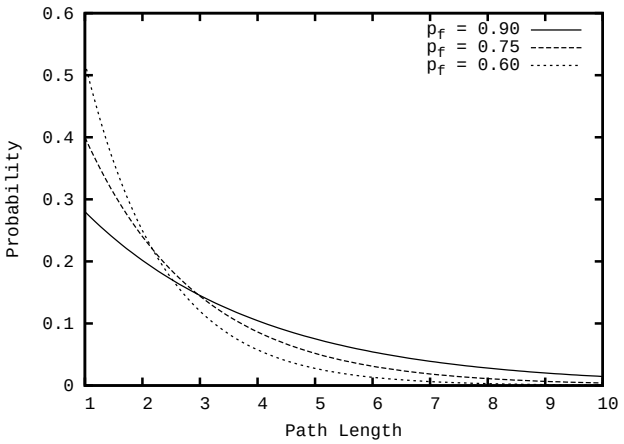

Figure 4: Distribution of path length probabilities with $f = 0.2$ and forward probabilities 0.6, 0.75, and 0.9.

If the ultimate destination of the message is not part of the coordinated attack, AP3 still preserves beyond suspicion with respect to the destination. However, if the destination is part of the coordinated attack, AP3 provides the anonymity guarantee of probable innocence, since a malicious node along the path can relay the identity of the previous node the to the destination.

The maximal coordinated attack that AP3 can withstand while providing probable innocence with a fixed forward probability p_f is described by the equation:

$$f < 1 - \frac{1}{2p_f}$$

which is derived from the fact that the probability of a path of length one is not greater than 50%. It follows that, with a coordinated attack consisting of a fraction f of the network, the forward probably must satisfy the equation below in order to maintain probable innocence.

$$p_f > \frac{1}{2(1-f)}$$

4 Group Communication

In this section, we describe how the primitives discussed in Section 3 can be used to build a novel anonymous group communication service. The service provides the scalability, self-organization, and low cost of p2p end-system multicast systems like Scribe [2] while providing probable innocence to nodes using the group. Such a service would be desirable, for instance, for a news-feed under an oppressive government,

where neither the publisher nor any of the receivers would want their identity divulged.

P2p multicast is usually implemented by forming a subscription tree from the union of all member node routes to the root, and then using reverse path forwarding to publish content. In this context, one goal of AP3 is to provide publisher anonymity, so that any node receiving content cannot determine who published it. Also, AP3 aims to provide subscriber anonymity, meaning that no node, including the publisher or the root, can determine whether a given node is subscribed to the group or received the content. Additionally, no node should be able to determine the set of subscribers.

4.1 Publishing

In order to publish content anonymously, the publisher uses anonymous message delivery to send a message to the group's root. Since the request is sent anonymously, the root of the multicast tree cannot determine whether the node that sent the publish request was the originator of the content. Subsequent publish requests sent to the group will come via different anonymous paths, and thus neither the root nor any subscribers can determine if one publisher is publishing multiple times or if there are many distinct publishers.

4.2 Subscription

In the normal operation of a p2p multicast system like Scribe, the membership in the tree can be determined by interior nodes in the tree or by any node overhearing join requests. When membership must remain anonymous, efforts have to be made to protect the identity of subscribers. To that end, we use anonymous channels to allow anonymous subscriptions to the group. Any node wishing to receive content without divulging its identity can subscribe through a random set of proxy nodes, the last of which actually joins the multicast tree. Once content is published to the group, the message is passed back along the anonymous route to the subscriber. Thus, the apparent subscriber to the group is likely not the actual node that joined the group, so no node in the multicast tree can determine the identity of any subscriber.

Interior nodes in the tree join and forward on behalf of others in the overlay. They may also be receiving the content, but since nodes are compelled to join the tree upon an anonymous subscription there are some nodes in the tree that may not have asked to receive the content. So nodes in the tree have a reasonable excuse to be forwarding the content and thus they are afforded plausible deniability if accused of subscribing to the group.

While providing anonymity for receiving nodes, these subscription paths will increase the latency for content to reach the endpoints. Likewise, the link stress on the underlying physical network increases. The increase is related to the average path length, which is in turn controlled by p_f and reflects a tradeoff between cost and the degree of anonymity. Since a random node is used as a proxy subscriber, the tree is formed as usual and all load balancing properties are preserved within the interior of the tree. A diagram of an anonymous multicast group is shown in Figure 5, where the jagged lines denote a random anonymous path to the multicast tree, which is highlighted.

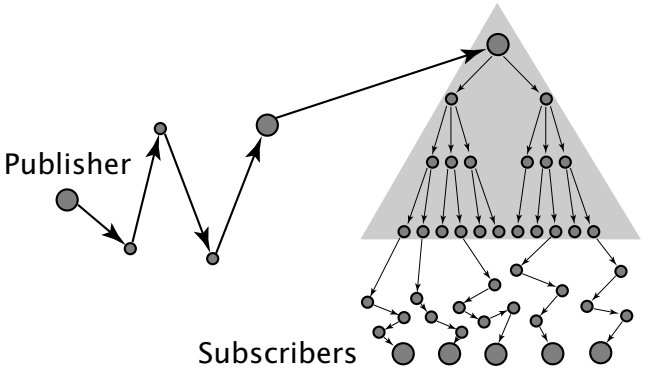

Figure 5: Diagram of an anonymous multicast group. Jagged lines represent anonymous paths, and the nodes behind the grey triangle are in the traditional multicast group.

4.3 Anycast

Anycast is a group communication primitive that is typically used to locate a node with a given property. Such nodes all join a multicast group; other nodes looking for a matching node send an anycast to the group. If the group contains at least one member, the message is delivered to at least one of the subscribers. For example, a distressed individual seeking counsel about a sensitive issue may wish to locate a qualified professional but with both remaining anonymous for reasons of liability or privacy. Implementing such a system is done in the same manner as the group multicast: the sending node sends the anycast request through an anonymized route and the subscribers are subscribed with anonymous channels.

5 Related Work

Onion Routing [5, 14] is based on a dedicated set of onion routers with complete knowledge of all other onion routers. Request initiators first pre-determine the path their messages will take, and then encrypt them in layers such that routers at successive hops can decipher exactly one layer. Onion Routing's design cannot adapt to rapidly changing networks, since the frequent arrival and departure of onion routers requires significant communication among all routers. Onion routing provides beyond suspicion anonymity with no compromised routers but if routers are malicious then anonymity may be sacrificed. A second version of Onion Routing [5] has been recently proposed that attempts to address some of the shortcomings in the original scheme. The newer scheme relies on directory servers who agree on the set of onion routers, these directory servers again may vulnerable to certain attacks. The

newer scheme also add support for a primitive similar to the anonymous channels presented in this paper.

Another system, Tarzan [8], is based on the peer-to-peer paradigm. Therefore, it does not share Onion Routing's reliance on a small set of fixed nodes. However, requesters in Tarzan must also pre-determine message paths, which requires them to have knowledge of a significant portion of the network. To accomplish this, peer discovery in Tarzan is implemented using a gossip-based protocol with the aim of producing a fully connected network of nodes. Such an architecture limits Tarzan's scalability, especially when considering the rapid flux in network topologies common to peer-to-peer architectures. Significant overhead is also incurred during route creation due to Tarzan's encryption mechanism, which requires key exchange. MorphMix [16] is another peer-to-peer solution that differs from ours in that it focuses on the problem of providing a low latency socket.

Crowds [15] is an application-level anonymization solution that implements routing in a similar fashion to AP3. Routes in Crowds are determined dynamically as nodes make random decisions to either forward or fulfill requests. Unlike AP3, subsequent requests in Crowds follow the same path until a periodic path reformation occurs, usually hourly. Crowds also provides admission control by using a centralized server, known as a "blender". This dependence on a single node restricts Crowds' scalability.

Hordes [10] is an application level anonymization system similar to Crowds, which adds support for anonymous multicast receivers. Hordes relies on the deployment of IP multicast, a technology that has yet to receive wide scale adoption for a variety of reasons. Furthermore, Hordes does not provide an anycast primitive.

Recent analysis of attacks based on hostile ASes (Autonomous Systems) [7] have shown that if a large AS such as an ISP is hostile than there are a large number of attacks possible on many anonymization systems. Our system would share these vulnerabilities .

6 Conclusions

AP3 provides a cooperative, distributed anonymous communication service. It is built on top of untrusted nodes, gracefully handles node arrival and departure and provides a flexible, lightweight, generic mechanism for anonymizing unicast and group communication.

Acknowledgments

This research was supported by Texas ATP (003604-0079-2001), by NSF (ANI-0225660) and a gift from Microsoft Research. We thank the anonymous reviewers for their helpful comments.

References

[1] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach. Security for structured peer-to-peer overlay networks. In *Proc. of the Fifth Symposium on Operating System Design and Implementation (OSDI 2002)*, Boston, MA, December 2002.

[2] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communication (JSAC)*, 20(8), Oct. 2002.

[3] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang. A case for end system multicast. *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast*, 20(8).

[4] S. Deering. RFC 1112: Host extensions for IP multicasting, Aug. 1989.

[5] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the Thirteenth USENIX Security Symposium*, San Diego, CA, Aug. 2004.

[6] J. Douceur. The Sybil attack. In *Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Cambridge, Massachusetts, Mar. 2002.

[7] N. Feamster and R. Dingledine. Jurisdictional diversity in anonymity networks. `http://freehaven.net/doc/routing-zones/routing-zones.ps`.

[8] M. J. Freedman, E. Sit, J. Cates, and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS 9)*, Washington, D.C., Nov. 2002.

[9] R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of DHT routing geometry on resilience and proximity. In *Proc. ACM SIGCOMM'03*, Karlsruhe, Germany, 2003.

[10] B. N. Levine and C. Shields. Hordes: A protocol for anonymous communication over the internet. *ACM Journal of Computer Security*, 10(3), 2002.

[11] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Cambridge, Massachusetts, Mar. 2002.

[12] T. Ngan, P. Druschel, and D. S. Wallach. Enforcing fair sharing of peer-to-peer resources. In *Proceedings for the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA, Feb. 2003.

[13] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.

[14] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communication: Special Issue on Copyright and Privacy Protection*, 16(4), May 1998.

[15] M. K. Reiter and A. D. Rubin. Anonymous Web transactions with Crowds. *Communications of the ACM*, 42(2):32–48, Feb. 1999.

[16] M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-peer based anonymous internet usage with collusion detection. In *Proceedings of the Workshop on Privacy in the Electronic Society*, Washington, DC, USA, Nov. 2002.

[17] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM Middleware 2001*, Heidelberg, Germany, Nov. 2001.

[18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proc. ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.

[19] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-resilient wide-area location and routing. Technical Report UCB//CSD-01-1141, U. C. Berkeley, April 2001.

[20] S. Zhuang, B. Zhao, A. Joseph, R. Katz, and J. Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. In *In Proc. of the Eleventh International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2001)*, June 2001.