Hiding in Plain Sight: Exploiting Broadcast for Practical Host Anonymity

Craig A. Shue*
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
shueca@ornl.gov

Minaxi Gupta
School of Informatics and Computing
Indiana University at Bloomington
minaxi@cs.indiana.edu

Abstract

Users are being tracked on the Internet more than ever before as Web sites and search engines gather pieces of information sufficient to identify and study their behavior. While many existing schemes provide strong anonymity, they are inappropriate when high bandwidth and low latency are required. In this work, we explore an anonymity scheme for end hosts whose performance makes it possible to have it always on. *The scheme leverages the natural grouping of hosts in the same subnet and the universally available broadcast primitive to provide anonymity at line speeds. Our scheme is strongly resistant against all active or passive adversaries as long as they are outside the subnet. Even within the subnet, our scheme provides reasonable resistance against adversaries, providing anonymity that is suitable for common Internet applications.*

1 Introduction

Common Internet protocols and applications are not designed with anonymity in mind. Consequently, Web users have little privacy: The sites they visit can track their behavior even if the users do not intentionally provide any self-identifying information. A commonly used mechanism enabling this tracking is *Web bugs*, where a third party contracted by the Web site puts an invisible Web object on the page visited by the user. A retrieval of this object allows the third party to record information about the user. This technique is a concern since it allows the third party to track the user across a large number of sites. Internet service providers (ISPs) possess similar capabilities. In fact, ISP partnerships with advertising companies to mine user behavior in order to show them targeted advertisements was recently the subject of a US Congressional hearing due to user privacy concerns [1]. Search engines pose similar threats to user privacy. In a well-publicized case, a correlation of search terms entered by Internet users was demonstrated to be sufficient to identify individuals, undermining their privacy [2]. Anonymity fosters freedom of speech, especially in sensitive matters, such as political dissent, where users might otherwise be oppressed for sharing their views. Thus, anonymity in blogging and messaging applications is also a highly desirable feature. Another prominent class of applications that can benefit from host anonymity are peer-to-peer applications, where lack of anonymity and over-reliance on IP addresses to identify individuals has caused innocent users to battle litigation concerning copyright infringement [3].

Many worry that anonymity can be misused by perpetrators of illegal acts, such as spammers, phishers, and denial-of-service (DoS) attackers. However, since these criminals already have tools, such as massive botnets, to launch their attacks anonymously, the lack of anonymity mechanisms only hurts other Internet users. Thus, similar to other works on the topic, we take the view that anonymity is a desirable feature in the Internet.

Anonymity is a well-researched area with many related works. Broadly speaking, the related works can be divided into two categories. The first category of solutions utilize proxies that forward client requests to the servers as their own [4, 5]. Network Address Translation (NAT)[6] and Port Address Translation [7] can also mask the hosts behind the trans-

*This submission was sponsored by a contractor of the United States Government under contract DE-AC05-00OR22725 with the United States Department of Energy. The United States Government retains, and the publisher, by accepting this submission for publication, acknowledges that the United States Government retains, a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this submission, or allow others to do so, for United States Government purposes.

lator. These solutions offer anonymity to the clients from the servers but not from adversaries that can monitor incoming and outgoing traffic at the proxy or translator. Further, legal necessities or a system compromise can force a proxy or translator to divulge information that can compromise client identities. The second category of solutions provide stricter anonymity guarantees, often for both parties involved in the communication [8, 9, 10]. However, those guarantees come at the cost of sacrificing performance: these approaches introduce extra hops in the communication path, additional latency, or both. For example, Herbivore [11], which claims efficiency in bandwidth and latency, used the Internet2 network for performance evaluation and reported achieving bandwidths of 6.25 to 12.5 KBytes/s with latencies around 1 second. These fall far short of the Mbits/s bandwidth and millisecond latencies many Internet users expect.

Our work is motivated by the desire to provide anonymity as a basic service to all Internet users rather than to a select few who take the time and effort to join existing anonymity networks. At the same time, we wish to do so with minimal changes, with little or no performance degradation, so that anonymity can be a *default-on* property of Internet communications. Our approach is based on the observation that users belonging to the same organization or ISP are naturally grouped with the other users of the same organization or ISP. This group can be used to create an *anonymity set*, a group of users that allow each other to be anonymous. We leverage the *broadcast* primitive already available in subnets to offer users k-Anonymity [12] proportional to the group size. Our scheme replaces the source IP address in packets with the broadcast IP address of the subnet. This makes individual hosts indistinguishable to anyone outside the subnet but allows packets to be forwarded in the Internet without requiring any modifications to the functionality of routers outside the subnet. The responses are broadcast to all members of the anonymity set, allowing the intended destination to receive it. To make the reception of packets modified under our scheme efficient, we introduce the notion of *ephemeral session identifiers*, which also provides unlinkability of packets of a connection. The salient features of our scheme are the following:

- **Adversarial model:** Our approach provides strong resilience against all active or passive adversaries as long as they are located outside of the broadcast network. While adversaries within within the broadcast subnet may pose more of a threat, we show that the overhead incurred by the attackers make such efforts impractical, especially given the limited payoffs associated with common Internet usage.

- **Deployment and anonymity:** Our scheme provides anonymity to both end points of the connection if they so desire. Each end point can make an independent decision to be anonymous or not. Even if only one end point chooses to be anonymous, it requires cooperation from the other end point to reap the full benefits of anonymity and unlinkability offered by our scheme. Without this cooperation, our scheme can still provide anonymity, but cannot provide unlinkability.

- **Changes required:** Only minor modifications are required to be made at routers belonging to the subnet to invoke anonymity in our scheme: They must enable broadcast packets from external machines and allow internal hosts to put the broadcast address in outgoing packets. The hosts need to use the subnet's broadcast address as the source IP address of outgoing packets. They also need to be able to identify the broadcast responses directed to them.

- **Performance:** Some existing residential broadband networks, including cable broadband, already employ broadcast to reach their hosts. Our approach incurs no additional overheads in such networks. Even on well provisioned point-to-point last-mile networks, such as Gigabit Ethernet and fiber to the premise, our approach provides anonymity at line speeds. However, our approach is not suitable for networks with high intra-subnet traffic, such as corporate intranets, or networks with last-mile bottlenecks, such as dial-up and digital subscriber loop (DSL) connections.

The rest of this paper is organized as follows. We discuss the details of our anonymity scheme in Section 2, including our adversarial model. The performance aspects of the proposed scheme are presented in Section 3 and a discussion of practical issues is provided in Section 4. We review related work in Section 5 and present concluding remarks in Section 6.

2 Details of Our Anonymity Scheme

2.1 Definitions

In our work, we use Samarati and Sweeney's definition of k-Anonymity [12]: "A table provides k-anonymity if attempts to link explicitly identifying information to its contents ambiguously map the information to at least k entities." In our work, the set of entities which this information can match, which Pfitzmann and Hansen [13] label the *anonymity set*, is the subnet to which a host belongs. The size of this set, or k, is dependent upon the size of the subnet the host's ISP uses.

Pfitzmann and Hansen [13] define unlinkability from an attacker's perspective; if data are unlinkable, an attacker will be no more able to relate these data after his observation than he was without this observation. If the property of unlinkability is assured, an attacker would be unable to link flows between a targeted host and a given destination with communication between that targeted host and any other destination.

2.2 Adversarial model

In our adversarial model, the adversary may control any or all parts of the network outside the host's broadcast network, including the remote connection end point. We provide provably strong guarantees against both passive and active attacks launched by the adversaries. We exclude hosts inside the subnet from the adversary's control and consider these insider attacks separately.

As with all anonymity approaches, we must make some assumptions about the payload of anonymous packets. Users can break the anonymity of any system by simply disclosing their identity in the payload of a packet. In order to make this problem tractable, we assume that users will actively conceal their identity. Further, protocol normalizing approaches can prevent the hosts from unintentionally divulging their identity through cookies or operating system specific behavior [14]. Accordingly, we assume no information in the application layer or packet payload undermines the host anonymity.

Claim 1. *Any adversary or a collection of adversaries outside of the broadcast network may arbitrarily observe, alter, or insert traffic at any or all links and nodes without violating the k-anonymity of the host within the broadcast network.*

Proof. All packets leaving the subnet router have identical data link headers and indistinguishable physical properties because all packets from the anonymity set originate from the same router. By ensuring that the source IP address contained in the packets is the broadcast address for the subnet, our scheme removes a critical distinguishing feature from the network layer. Accordingly, the only remaining fields that can be used to determine the identity of the source are in the transport and application layers. Since we assume proper neutralization of the transport and application layers as well as no admission of identity in the packet payload, packets emerging from the broadcast network will not have any properties that can distinguish a host from others in the broadcast network.

When receiving traffic from the outside network, all packets will be sent to all hosts in the broadcast network. Accordingly, regardless of any modifications to the packets by an adversary, any response from the anonymous host could be issued by any of the members of the broadcast network since they would each be operating on identical information. □

Depending on the underlying structure of the broadcast network, adversaries within the subnet may weaken host anonymity. However, even if successful, the attacks may be too costly and the payoff too low in order to be worthwhile for attackers. For example, on bus networks, such as for cable broadband, colluding adversaries straddling a node may be able to determine which packets that node adds to the wire, subverting its anonymity. However, without obtaining topology and subscriber information from the ISP, these colluding nodes would be uncertain how many nodes separated them or the identity of an individual subscriber. Even with these efforts, the colluding nodes 1) cannot decrypt the payload without colluding with the remote host, 2) must capture every packet to make the analysis effective since they do not know which packets to target in advance, and 3) may not be able to benefit from breaking anonymity on most traffic since most data is not sensitive. Accordingly, while colluding adversaries within a subnet may be able to weaken host anonymity, it is unclear such attacks against most users would be economically viable.

2.3 A basic bootstrapping scheme

The key observation behind our anonymity scheme is the following: Hosts in a subnet can be collectively addressed by a broadcast and they naturally form an *anonymity set* under which all the subnet hosts can

be anonymous. To leverage this observation, we replace the source address in packets originating at a subnet to contain a broadcast address representing all hosts of that subnet. The routers send these packets as they do today and the receiver of these packets can respond, oblivious to any change. When the response packets reach the subnet, the router receiving them will broadcast them to all hosts in the subnet. Hosts then find the responses to their messages by looking at the IP address of the sender, source and destination ports, and the protocol field present in the IP packet. This filtering operation would have to be performed in software on the host system until support was added to network interface cards (NICs) to use hardware filtering.

In some cases, multiple hosts may simultaneously communicate with the same server, over the same protocol, and with identical source and destination port numbers, resulting in a conflict. Most TCP implementations at the servers resolve this conflict by issuing a connection reset (RST) to the second client. While some TCP implementations will attempt to retry to establish failed connections [15], others may require a kernel patch to the networking stack to assure this behavior. This strategy works for connection-oriented protocols, such as TCP but not for connectionless protocols, such as UDP. In those cases, the application layer must resolve the conflicts. Though far from ideal, in many cases, the applications already possess enough information to disambiguate. For example, in case of DNS queries, which use UDP as transport layer, the *queryID* field reduces the probability of confusing the application substantially.

Any ISP can deploy the basic scheme in its subnet to offer broadcast anonymity to its hosts from the rest of the Internet, without requiring any support from external end points. This helps the approach's deployment. However, the basic scheme provides no confidentiality. Even though the broadcast address prevents hosts from knowing who received the packets, they can observe the contents. Fortunately, adding confidentiality is straightforward in many cases. Protocols, such as transport layer security (TLS), are in wide use to provide confidentiality and authenticity to Internet communication. These protocols have widespread client and server support on the Internet. If the remote host supports such a protocol, authenticity and confidentiality can be assured without any other modification. In fact, some sites, such as `https://www.cia.gov`, already require hosts to use TLS when visiting the site.

2.4 Adding unlinkability to the basic scheme

While easily deployed, the basic scheme described in Section 2.3 has some limitations: 1) simultaneous communication between members of the anonymity set and the same remote host could result in ambiguity if they use the same source and destination ports, 2) without TLS or other security protocols, all members of the subnet would be able to see the payload of all the broadcast traffic sent to others in the network, and 3) an outside eavesdropper could analyze traffic flows and learn how long flows last between the remote host and the anonymous host.

To eliminate these deficiencies, we require packet encryption and authentication and introduce the notion of *ephemeral session identifiers*. These identifiers are used by a responding host to demultiplex traffic from multiple parties in the subnet and by hosts in the subnet to filter out traffic destined to them. While static session identifiers could be used for the duration of the connection, an eavesdropper could perform statistical profiling on the flow. To preserve unlinkability, these session identifiers must change regularly. To do so, the clients send two session identifiers: the identifier for the current set of packets, which we denote `A`, and an identifier for the next set of packets, labeled `B`. In each packet, the value of `A` is transmitted in the clear, but the value of `B` is encrypted, ensuring eavesdroppers do not learn the value for the next set of packets. The hosts regularly change the session identifiers by replacing the current identifier, `A`, with the value for the next packet set, `B`. The host generates a new random session identifier for the following set of packets, `C`, which is again encrypted when transmitted. Since each communicating host knows the next session identifier the other host will use, it can continue to identify the session when the identifier advances.

The process begins with the initiating party, `I`, which randomly generates both the current, `A`, and next session identifiers, `B`, and includes them in its message to a responding host, `R`. The responding host (`R`) uses the initiator's current session identifier (`A`) for its own current session identifier; however, it generates its own random next identifier, `X`. The responding host includes both session identifiers (`A` and `X`) in its response. The initiating host confirms the responder's reply by comparing the current session identifier to the one it created (`A` = `A`). Both hosts then advance their session identifiers, to `B` for `I` and to `X` for `R` and generate new identifiers, `C` and `Y` respectively, and include them in the encrypted payload of

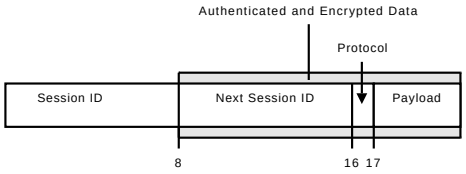

Figure 1: Header for Approach 1.

the packet. The hosts continue to regularly advance the session identifier for the duration of their connection. After the first round trip, the identifiers for the two hosts are independently generated from random, making them unlikely to match. Accordingly, packets from one host to the other cannot be linked to packets traveling in the reverse direction using the session identifiers. The session identifiers can be embedded in a new layer, the *anonymity layer*, that sits between the network layer and transport layers (shown in Figure 1). The anonymity layer consists of three fields: an 8 byte identifier for the current session, an 8 byte identifier for the next session, and a one byte protocol field, which indicates the type of protocol above the anonymity layer.

Finally, confidentiality of packets in this modified scheme works similar to that in the basic scheme; however, this scheme requires that the next session ID and protocol fields in the anonymity layer be kept confidential. Otherwise, an eavesdropper could observe the next session identifier in order to track flows and perform statistical attacks.

3 Performance

While previous research in anonymity networks has led to some strong anonymity properties, these benefits come at the cost of high latency, low bandwidth, and introduces a series of traffic bottlenecks. However, our scheme introduces no extra latency and offers line-speed bandwidth in several types of modern network deployments. In this section, we examine these properties further.

A key tool in our architecture, message broadcasting, raises important performance concerns for networks. In networks that already employ a broadcast model, such as token ring, satellite, or residential cable networks, the approach does not result in significant performance differences because no extra hops or packets are added to the Internet communication. The only overheads in those cases are packet overheads, which are in the form of 17 bytes of anonymity layer. Further, irrelevant traffic must be filtered in these broadcast networks to avoid hosts processing traffic that is not destined to them. This is often done in the NIC hardware using the destination MAC address in many shared-media networks. In our case, the MAC address will be the broadcast address. Accordingly, this filtering would need to be done in software in the kernel or the NIC would need to be modified to filter based on the session identifiers. While hardware support in the NIC would be little different than filtering on MACs, it may require new NICs rather than simply a firmware update.

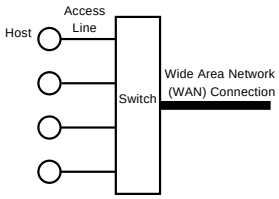

Figure 2: Line speed example network.

In point-to-point networks, performance needs to be carefully considered because broadcast is not a native primitive of such access technologies. To facilitate broadcasts in such networks, the line speed for every component in the broadcast network must be equal to or greater than the line speed of the wide area network (WAN) connection from the broadcast network. In Figure 2, we provide an example network. If the downstream line speed on the access lines is greater than or equal to the downstream line speed on the WAN connection, then the incoming traffic on the WAN connection can be mirrored across all the access lines without causing congestion on the access lines, allowing straight-forward deployment. In cases where this condition does not hold, the subnet can be divided, resulting in less traffic, until the line speed requirements can be met by the access lines. If the subnet's downstream traffic exceeds the access line speed, some packets must be discarded, signaling congestion and ensuring the subnet does not exceed it's proportion of available bandwidth. Fortunately, most modern Ethernet networks are provisioned in a manner than satisfies these conditions. However, in networks where there is significant intra-subnet traffic, such as corporate network with a busy file server, the access lines may be overwhelmed by the sum of the intra-subnet traffic and WAN traffic. These networks would need to provision accordingly. Additionally, these point-to-point networks would have to implement filtering approaches to avoid processing irrelevant traffic, which was not previously required. This filtering must be done efficiently; however, line-speed filtering has been extensively explored in firewalls and is unlikely to be a substantial barrier to deployment.

In other access technologies, such as with dial-up modems or DSL lines, the access line speed is the connection bottleneck. For example, if the access lines in Figure 2 offered only 25% of the download bandwidth of the WAN connection a single host receiving broadcast traffic could saturate the access links. Accordingly, even unicast traffic destined to any of the other hosts would contend for bandwidth with the host receiving broadcast traffic. A broadcast network of `n` nodes and access line-speed `s` and no additional intra-subnet traffic would function if downstream clients only requested broadcast at a maximum of $\frac{s}{n}$, a significant performance penalty for even moderate sized networks. Further, these networks would be vulnerable to a single client, intentionally or otherwise, saturating the access line-speed with broadcasted traffic. Unfortunately, fair queuing would be of limited value, since the broadcast traffic would be to an anonymous destination, preventing per-host traffic limits. While a fair queuing mechanism could be placed inside the broadcast network before traffic is aggregated across multiple hosts, it would only be able to fair-queue upload traffic, which may be asymmetric to the amount of download traffic generated by those messages. Additional approaches may be possible for hosts with access-line bottlenecks, such as a filtering agent on the opposite side of the access line.

4 Discussion

4.1 Presence of legacy hosts

Though a likely scenario for deployment of our proposed anonymity scheme is one where all hosts within a subnet are updated to enable anonymity, we must also consider scenarios where some hosts do not apply the update. The latter is especially true for residential Internet users who are known to ignore patches even when they address serious security concerns. Accordingly, the implications of broadcast in such cases need to be properly considered. While many legacy hosts will ignore unsolicited broadcast messages [16], some may react to the packet as if they were the intended destination, especially in the case of the basic approach discussed in Section 2.3, since the packets are otherwise unmodified. In the case of TCP, these hosts may issue a reset (RST) packet to broadcast messages for connections that were not established by them. If any such legacy hosts exist on the broadcast network, it may cause the remote hosts to reset the TCP connection with each message they receive. This could potentially cripple the communication for participating hosts. Further, in extreme cases, adversaries could launch such resetting attacks, even if there were no legacy hosts. The advanced approach discussed in Section 2.4 does not have these limitations: legacy hosts would not understand the anonymity layer and would simply drop, possibly sending an ICMP protocol unreachable message, which could be filtered by the router. Further, if protocols to ensure confidentiality and authenticity were deployed, attackers would be unable to create valid reset messages in the advanced approach.

4.2 Properties of the anonymity set

The size of the anonymity set is an important consideration in our scheme, much like with any other anonymity schemes. If only a single host wishes to be anonymous, the broadcast address could still uniquely identify the host, especially if the total size of hosts in the subnet is known. It may be challenging for residential broadband networks, where users administer their own machines, to obtain a sizable anonymity set. In such cases, ISPs could market the anonymity service as a value-added proposition.

The diversity of the anonymity set is also an important consideration. If the set lacks diversity, as in the case of a broadcast network for a single organization, the entire organization may targeted. In a diverse set, such targeting is impossible. Further, individuals and organizations may not wish to have their replies duplicated to others in the anonymity set, which may have competing interests. However, this traffic may be encrypted and completely unlinkable with the remote destination with destination support, posing little risk[1]. These organizations may choose to employ the anonymity approach only with remote entities that aid in concealing their identities to avoid leaking any usable information.

4.3 Changes required to routers and end hosts

Our scheme places two simple requirements on the edge network's router: it must broadcast messages from remote hosts via the router's broadcast address and it must not filter packets from the local network that use the broadcast address as their source address. Most networks already support this required functionality. For those which block broadcast packets at the subnet firewall, a simple exception would suffice. This provisioning is not without concerns,

[1] In cable broadband networks, this traffic is already shared with others in the broadcast network, whose cable modems filter the traffic.

however. For example, some broadcast messages could lead to reflector attacks, in which a message with a spoofed source address of a victim machine causes each member of the broadcast network to reply to the spoofed address, multiplying the number of packets and bandwidth the victim machine would use to fend off the attack. These networks could block any packets that would lead to reflection attacks or even restrict the type of packets that would be broadcast, causing only anonymity-aware hosts to reply.

In order to filter packets properly, deploying hosts would need a kernel patch that modifies the networking protocol stack to use the broadcast address as the source address in anonymous packets, as well as the processing and creation of the anonymity headers. For hardware filtering, the NIC would require updates to support filtering on session identifiers.

4.4 Mutual anonymity

When both communicating end points are deploying broadcast networks and leverage anonymity, both hosts remain anonymous to each other and to any node outside of their respective broadcast networks. Since both hosts deploy the approach, we can assume they support the anonymity layer. Accordingly, they can leverage the ephemeral session identifiers to provide unlinkability of packets from the same connection. This is particularly useful for anonymous peer-to-peer communication. Peers could either use a directory service to discover each other or incorporate such a directory into existing peer-to-peer systems.

While our scheme can provide anonymity to both end points of the communication, it is much easier for clients to reap the benefits of anonymity. This is because they can deploy the basic version of our scheme without any cooperation from the servers. Even if the server cooperates to enable ephemeral session identifiers to clients, they may not be able to deploy anonymity themselves in order to continue serving non-deploying clients. Further, servers that are known by their identity, rather than the content they provide, may not be able to make themselves anonymous. This is because accessing these servers today requires DNS lookups to map their name to IP address. A NAT box or a firewall may mask the true identity of the server by mapping the publicly available IP address to the real IP address, but in general strong anonymity is not a realistic goal for such servers.

Servers that are known by the content they serve, not their name, can be anonymous using our scheme. Examples of such servers include peer-to-peer network hosts or the servers users visit after performing keyword searches. In many such cases, the users are interested in specific content and care little about the identity of the content provider. For such servers, rendezvous mechanisms would have to be developed. This concept has been proposed in several related works listed in Section 5. The rendezvous mechanism could be in the form of a directory service which stores records for servers. To announce its services, an anonymous server would anonymously post its entry to the directory service.

When operating as a server in a deploying broadcast network, the anonymous server must be able to filter messages. To do so, the server can bind to a specific longer-term session ID, which it uses to receive initial queries. When issuing replies, the server can use a different, ephemeral session ID, allowing the hosts to communicate. Once the server replies, both the client and the server can begin communicating anonymously, causing eavesdroppers to only be able to detect the two subnets have communicating participants.

5 Related Work

Anonymity has long been a topic of interest, both in the security and networking communities. A variety of themes have developed for tackling anonymity. These approaches can be broadly classified as mix-based strategies or broadcast-based and each approach comes with its own set of strengths and limitations. We highlight our contributions by comparing our scheme with each.

5.1 Mix Networks

In creating Mix-Net, Chaum laid the foundation for modern anonymity systems [8]. In Mix-Net, public key cryptography was used to encapsulate packets and route them through a series of intermediaries, called mixes. At each mix, the messages could be re-ordered, delayed, and decrypted in order to make unclear which source sent the message. To ensure high degrees of anonymity and resilience against attack, some systems increased the latency of messages in order to have better mix properties [17, 18]. Due to the high latency, these approaches cannot be used for some interactive applications.

Low latency approaches have also been explored. The most simplistic anonymizing mix network approach is to use a single Web proxy [4, 5] or NAT device. Unfortunately, since the proxy or translator must maintain state about the client's request

and identity, these proxies are a single point of failure and must be fully trusted by the end-users. The Crowds approach [9] used a mixed network of peers that would probabilistically forward messages or send them to a destination recipient. Accordingly, when receiving a message, peers in the network would be unable to distinguish whether the previous peer originated the message or merely forwarded it based on a coin toss. The destination server would see the IP of the node that decided to forward the packet, but would not know which host actually originated the message. The Tor approach [10] demonstrated that mix networks can be used in practical communication and has a well-deployed network of users, but does sacrifice resilience against some attacks in favor of practical deployment. Tor uses the *onion routing* approach to provide anonymity. In Tor, hosts create a circuit of onion routers who act as a chain of proxies for the original source. When creating the circuit, the users negotiate shared keys with each circuit member. The end-host then symmetrically encrypts the packet payload using the symmetric key of the last onion router, followed by the next to last onion router, and so on. It then transmits the packet to the first router, which decrypts the packet, examines the packet header to find the next destination, and sends the packet on to that destination, at which point the process repeats.

Mix networks have fundamental properties that limit their usage. Sending traffic to a mixing node before reaching the destination incurs extra latency, which is increased with each additional node. Further, since mixing nodes aggregate traffic in order to create anonymity sets, they also become congestion points for traffic. Accordingly, these services cannot offer line-speed bandwidth to a large number of users without being substantially over-provisioned. Our scheme introduces no extra latency and does not aggregate traffic at mixes, eliminating these bandwidth bottlenecks and allowing high bandwidth usage.

5.2 Broadcast/Multicast Approaches

Chaum [19] demonstrated the power of broadcast messages to cause unconditional untraceability using multiple parties. While described in terms of human speech, such an approach holds value in a communication network where broadcast messages can also be enacted. In a later work, Chaum [20] generalizes this approach to multiple parties and describes how it can be realized in key exchanges. He further discusses how a network ring topology enables such broadcast messages. Unfortunately, this protocol allows only one message to be sent on the network at a time, leading to coordination and bandwidth limitations. The Herbivore approach [11] extends the work by Chaum to create a network that provides coordination through a series of rounds, allowing the transmission to occur while obtaining better performance. Unfortunately, even when used on a well provisioned network, Herbivore offers bandwidth rivaling a dial-up connection and incurs latency nearing one second. While useful for low bandwidth tasks, Herbivore cannot be used for a variety of today's Internet activity. In the P^5 work [21], machines are arranged into broadcast cliques, which perform hop-by-hop encryption to hide the identity of the original node that transmitted a message. The protocol relies upon all nodes transmitting a packet; when they have no packet to transmit, they must transmit a "noise" packet. Like Herbivore, P^5 provides only modest bandwidth even on well provisioned networks (about 2 KBytes/s on a 1.6 Mbits/s connection with low loss rates or about 25 KBytes/s on the same network with a 40% loss rate). Unlike these broadcast approaches, our architecture allows line-speed communication in broadcast networks and adds no network latency while requiring only slight packet header overhead.

6 Concluding Remarks

Third party Web sites which aggregate user behavior across multiple Web sites are in a unique position to track users today. This holds for popular search engines and ISPs as well. Any leakage of information from these entities, intentional or unintentional, can have adverse consequences for users. Examples of these consequences include revelation of medical conditions, targeted phishing attacks (commonly known as *spear phishing*), and targeted advertisements. To counter the potential harm enabled by day-to-day activities on the Internet, we presented a scheme that can provide anonymity to end hosts at line speeds, without degrading performance.

While our scheme enables an important feature for Internet users, it has side effects. It decreases the accuracy of any application or research project relying on the identity of hosts, including topology-mapping efforts that leverage the geographical location of Internet hosts. While such efforts could previously make conclusions about the end hosts, under an anonymity system, they would only be able to do so at the IP prefix granularity. In fact, in many cases, classless inter-domain routing (CIDR) would make it challenging to even determine when the results are

for anonymized hosts. It is debatable whether this is harmful or not. We only note that the Dynamic Host Configuration Protocol (DHCP) already interferes with services trying to locate the hosts.

Some other Internet technologies may also be limited by the anonymity. Fair-queuing efforts or other resource sharing approaches may be hindered by the aggregation of anonymous traffic. Accordingly, other mechanisms may be required to provide these services. Some protocols, such as NAT or IPSec, which leverage the IP address of the host in order to demultiplex flows would require modification to accommodate anonymous traffic, since several hosts may be aggregated under a single source address.

References

[1] C. Albanesius, "Can Internet activity ever be truly anonymous?" Jul. 2008, http://www.pcmag.com/article2/0,2817,2325253,00.asp.

[2] A. Jesdanun, "AOL: Breach of privacy was a mistake," Aug. 2006, http://www.washingtonpost.com/wp-dyn/content/article/2006/08/07/AR2006080700790.html.

[3] E. Bangeman, "RIAA drops file sharing case," Oct. 2006, http://arstechnica.com/news.ars/post/20061015-7990.html.

[4] Anonymizer, Inc., "Anonymous proxy servers," http://www.anonymizer.com/.

[5] Citizen Lab, "Psiphon," http://psiphon.civisec.org/.

[6] K. Egevang and P. Francis, "The IP network address translator (NAT)," IETF RFC 1631, May 1994.

[7] H. Yeom, J. Ha, and I. Kim, "IP multiplexing by transparent port-address translator," in *USENIX Tenth System Administration Conference*, 1996.

[8] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudo-nyms," *Communications of the ACM*, vol. 4, no. 2, Feb. 1981.

[9] M. K. Reiter and A. D. Rubin, "Crowds: Anonymous connections and onion routing," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, Jun. 1998.

[10] R. Dingledine, N. Matthewson, and P. Syverson, "Tor: The second generation onion routing," in *USENIX Security Symposium*, 2004.

[11] S. Goel, M. Robson, M. Polte, and E. Sirer, "Herbivore: A scalable and efficient protocol for anonymous communication," Cornell University Computing and Information Science, Tech. Rep. TR2003-1890, Feb. 2003.

[12] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," in *IEEE Symposium on Research in Security and Privacy*, 1998.

[13] A. Pfitzmann and M. Hansen, "Anonymity, unlinkability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology," Feb. 2008, http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf.

[14] Privoxy Developers, "Privoxy - Home Page," http://www.privoxy.org/.

[15] S. Floyd, "Inappropriate TCP resets considered harmful," IETF RFC 3360, 2002.

[16] Microsoft Corporation, "ICF in Windows XP SP1 and Windows Server 2003 blocks unsolicited inbound unicast, multicast, and broadcast traffic," Knowledge Base Article 329928, http://support.microsoft.com/kb/329928/.

[17] C. Gülcü and G. Tsudik, "Mixing emails with Babel," in *Internet Society Network and Distributed System Security Symposium (NDSS)*, 1996.

[18] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in *IEEE Symposium on Security and Privacy*, 2003.

[19] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[20] ——, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *Journal of Cryptology*, vol. 1, no. 1, pp. 65–75, 1988.

[21] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, "P^5: A protocol for scalable anonymous communication," in *IEEE Symposium on Security and Privacy*, 2002.