Empirical Tests of Anonymous Voice Over IP

Marc Liberatore[b,*], Bikas Gurung[a], Brian Neil Levine[b], Matthew Wright[c,*]

[a] *Qualcomm, Inc. 5775 Morehouse Drive, San Diego, CA 92121 USA*
[b] *Department of Computer Science, University of Massachusetts Amherst, Amherst, MA 01003, USA*
[c] *Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington, TX 76019, USA*

Key words: anonymity, voice over IP, privacy, traffic analysis

Abstract. Voice over IP (VoIP) is an important service on the Internet, and privacy for VoIP calls will be increasingly important for many people. Providing this privacy, however, is challenging, as anonymity services can be slow and unpredictable. In this paper, we propose a method for extending onion-routing style anonymity protocols for supporting *anonymous VoIP* (aVoIP) traffic with reasonable performance. We report the results of extensive experimentation across 210 globally placed PlanetLab proxies which shows that paths for reasonable aVoIP quality would need to be selected carefully. Our design includes an algorithm for the measurement and selection of paths for reasonable aVoIP performance and an analysis of the potential for attackers to take advantage of this algorithm to improve existing attacks. We show that aVoIP could be developed in an onion routing system with reasonable performance guarantees and a modest increase in risk to its users as compared to the standard path selection algorithm.

1. Introduction

Voice-over-IP (VoIP) is a tremendously popular and important application on the Internet. The popular Skype P2P VoIP service had over 28 billion minutes of phone calls in 2006 [1]. While calls over the Public Switched Telephone Network (PSTN) are secured by physical barriers to phone lines and equipment, VoIP calls are often routed through volunteer, intermediary peers that can easily inspect data. Although the call's data may be encrypted end-to-end, preventing third parties from linking participants in a call is more challenging. We define VoIP privacy as the protection of the knowledge of who is communicating from all parties except the two end-points of the call.[1]

In this paper, we introduce a method for *anonymous VoIP* (aVoIP) that is secure and has reasonable Internet performance. Techniques for proxy-based anonymous routing are well known; our contributions include a measurement study showing the feasibility of such a system on the Internet and a security analysis of threats posed by VoIP quality-of-service requirements.

[*] Corresponding author

Email addresses: `liberato@cs.umass.edu` (Marc Liberatore), `brgurung@gmail.com` (Bikas Gurung), `brian@cs.umass.edu` (Brian Neil Levine), `mwright@cse.uta.edu` (Matthew Wright)

[1] This definition is different from that typically used by anonymizing services that also protect the identity of the initiating user from receivers. Unfortunately, one-sided anonymous phone calls encourage harassment.

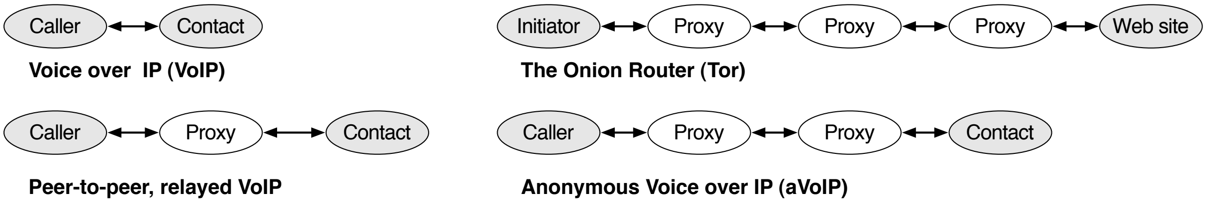

Figure 1: Path architecture comparison.

End-to-end delays in voice communication greater than 100–150 ms are detectable by humans, and delays greater than 200 ms can become intolerable and impede interactive communication [2]. These performance requirements make it difficult to provide an anonymized end-to-end connection for VoIP using a series of proxies. We performed an extensive measurement study of about 210 globally-placed proxies over a 10-week period. Our measurements demonstrate that aVoIP is currently possible on the Internet as a viable, robust, real-time service if we select paths carefully. Specifically, our measurements show that connections between a series of proxies located on the same continent could support aVoIP with one-way end-to-end delays of under 150 ms over 90% of the time. However, intercontinental paths of proxies could not support aVoIP traffic as well: of paths with one intercontinental hop, 35% show acceptable performance, while paths with two and three hops result in only 7% and 1% of the paths exhibiting acceptable performance, respectively.

These performance results dictate we build paths differently for aVoIP than for less delay-sensitive applications. As Kesdogan and Palmer point out, it is critical to consider the impact on network performance and user experience when building anonymity systems [3]. We propose to conduct occasional measurements of the connection quality between pairs of nodes and use these measurements to help users select a path with sufficient quality of service (QoS) for phone calls. While intuitive, this method could help attackers to modify path selection by manipulating the measurements. For example, the attacker can attempt to remove some pairs of nodes to improve the odds of users selecting his nodes for their paths. This strategy in turn strengthens attacks such as the predecessor attack [4, 5].

We use trace-based evaluations of these security threats to measure the limits of an attacker against our approach. We find that attackers have a nontrivial ability to reduce the measured number of paths with sufficient quality of service. When the attacker is powerful, with many attacker-controlled proxies (e.g., 33% of the proxies in the system), he can greatly reduce the work necessary to perform the predecessor attack; e.g., by 50% in the scenarios we examined. Even with a more modest number of attacker-controlled proxies (e.g., 5% of the proxies), the attacker can reduce his work by 20%. Nevertheless, we can still assure aVoIP users that they will receive most of the security benefits of existing, low-latency anonymous systems [6].

We begin with a description of our path selection method and our measurement methodology and present the results of our PlanetLab measurements. We then provide an analysis of the potential attacks on our proposed measurement and path selection methods.

2. Proposed Architecture and Evaluation Methodology

In this section, we describe our architecture for anonymous VoIP in terms compatible with previous work. We also present our specific evaluation methodology, including performance metrics of interest.

2.1. An Architecture for aVoIP

We begin by describing our proposed architecture for an aVoIP system. Figure 1 illustrates the differences between our approach and related systems, namely plain VoIP, peer-to-peer VoIP (such as Skype), and an onion-routing-based anonymity system. Each of the circles in this figure represents an application layer-proxy running on a distinct host, and each of the links represents a network hop between proxies. The VoIP illustration is a straightforward two-way connection; the peer-to-peer (or relayed) VoIP connection relies upon an intermediate host to relay traffic. This latter setup is typically required to circumvent traffic filtering at one or both end hosts.

In anonymity systems based upon onion routing, an *initiator* (the user) connects through a chain of intermediate *proxies*, which are remote hosts that are running application-layer software that relay encrypted traffic and hide routing information from observers. These proxies relay the traffic to the user's intended recipient, called the *responder*; this recipient is often a Web site. The responder's traffic is returned over the same path. Tor is an example of an onion-routing system, and it uses 1500 public Internet proxies and has an estimated 200,000 distinct users.[2] In Tor, the list of all available and active proxies is provided by a trusted directory server.

We propose the use of an onion-router based anonymity system for aVoIP, with the additional restriction that paths are selected by the initiator such that a minimum quality of service is provided at each hop. To ensure this constraint is met, the trusted directory server takes measurements of streaming data network performance by sampling from all possible paths. Tang and McKinley [7] show that not all paths need to be probed to get good results; examining this in the presence of an attacker is a challenging open problem. We assume full measurements of the system. The measurements are reported to each initiator, who can then estimate the quality of a randomly selected path before use, including its own link quality to other proxies as observed over time. We report on our collection of such measurements in Section 3.

The path length is variable. For example, by default, the Tor system uses three proxies between the initiator and the responder. Our measurements in Section 3 include scenarios with two and three proxies. Some may prefer the additional security provided by a path of three proxies. However, in our model, the two communicating parties both want privacy from third parties but are not concerned with privacy from each other. In this case it is reasonable to assume that the responder is running the final proxy on her own host, reducing the number of intermediate proxies to two.

For two reasons, we suggest the shorter path length. First, because aVoIP is more sensitive to the quality of links, the difficulty of finding a sufficiently high-quality path diminishes with longer paths. Second, this

[2] See `http://metrics.torproject.org` for current Tor statistics.

R-score	90<R≤100	80<R≤90	70<R≤80	60<R≤70	50<R≤60
MOS	4.34–4.5	4.03–4.34	3.60–4.03	3.10–3.60	2.58–3.10
Quality	Best	High	Medium	Low	Poor

Table 1: The mapping between quantitative and qualitative VoIP scores.

Set	**Hosts**
asia	40
america	49
europe	121

Table 2: The average number of PlanetLab hosts in our measurements active at any given time.

reduction in the path length proxies does not, by itself, lead to a major reduction in security. For example, if the attacker has compromised some of the proxy nodes, timing analysis can allow an attacker that controls the first and third proxies on a path to link the sender with the receiver. Prior work [8] has shown that this kind of attack is very accurate when no countermeasures are in place. Actively perturbing the timings of aVoIP flows is even more effective, with a 99% true positive rate and a 0% false positive rate [9]. Thus, regardless of whether two or three proxies are used, the adversary needs to control only two proxies to link the initiator with the responder.

Integration with Tor. Our designs and analysis are focused on an onion-routing version of aVoIP and not of a Tor-version of aVoIP in particular. The latter requires a lengthy discussion of specific Tor subtleties (e.g., guard nodes) that are best left for a document devoted to development issues. In this paper, we focus on broad security and performance issues that are not specific to any one system, and offer only brief comments on Tor below.

The proposed aVoIP architecture can be built into the Tor system and leverage Tor's mature design in the use of directory servers, the building of paths, and other operations. For example, the system's *control traffic* — including key exchange and path set up — could use TCP while data is sent over UDP. aVoIP could be deployed in one of two ways: separately from Tor or as part of Tor. If it is deployed separately, it will be simpler but requires a fresh set of volunteer node operators. If it is deployed as part of Tor, alongside existing TCP traffic, it will require integration and more extensive testing.

We do not evaluate the consequences of introducing UDP and TCP data streams in one protocol, except to make the following points. First, we note that there are several attacks in the literature that make use of congestion to compromise anonymity [10, 11]. We note, however, that these attacks are possible without the introduction of UDP. Second, Tor ensures some level of fairness through token-bucket-based bandwidth limits and window-based throttling at both the circuit and stream levels. These methods would need to be carefully modified for use in UDP streams, and selecting the best techniques and corresponding parameters is beyond the scope of this work.

2.2. Performance Metrics

Deadline-oriented applications like VoIP require that the *path between the initiator and the responder* provide sufficient quality of service for real-time voice communications. Three low-level metrics are typically used to characterize path quality for streaming multimedia applications such as VoIP: latency, loss, and jitter. Additionally, the International Telecommunication Union (ITU) has developed a family of metrics for telephony. We briefly describe each of the metrics we use in this study and their relevance below.

Latency: Latency is the delay between when a signal is transmitted and when it is received. According to ITU-T G.114 [2] recommendations, an upper bound on one-way latency is 150 ms for acceptable voice quality. A delay of 150–400 ms can be tolerable for international callers, demonstrating that QoS requirements also depend on user expectations. Privacy-conscious users may accept these higher latencies for anonymity, much as they do when browsing the Web with Tor.

Loss rate: The fraction of packets lost in transit, packet loss, is the result of either transmission errors or full queues at routers along the path. The use of Forward Error Correction (FEC) and the human ear's tolerance for small disruptions in audio means that the relationship between loss rate and perceived quality is not straightforward. Small amounts of loss may have no perceptible impact on quality, but once a codec-dependent threshold is reached, quality can drop off dramatically.

Jitter: Jitter is an estimate of the statistical variance in packet interarrival time. In this paper, we define instantaneous jitter as the difference between when two packets' actual and expected interarrival times. VoIP decoding occurs in close to real time, and it requires that packets be available by a playout deadline; late packets are effectively lost. The primary way to counter the effects of jitter is to buffer packets at the receiver ahead of playback, but this has the secondary effect of increasing latency.

E-model: The E-model, described in ITU-T G.107, is a tool to assist in planning and evaluating telephony systems running on a computer network. Ideally, such systems are evaluated by a panel of experts who use the system and make judgments about its quality, resulting in a Mean Opinion Score (MOS) ranging from one to five. Since such evaluations are expensive and time-consuming, the quantitative E-model was designed to approximate subjective judgments. A host of factors, ranging from loss and latency to loudness and room noise are measured and input into the model, which outputs a composite R-score. This score can be mapped onto the one-to-five scale of the MOS and closely approximates the score a panel of experts would likely give. Table 1 provides a well-known mapping from MOSs and R-scores to subjective terms.

Cole and Rosenbluth [12] examined the E-model in the context of VoIP on the Internet. Their goal was to simplify the E-model's twenty terms, many of which are irrelevant in network evaluation, to a more useful number. Their result is given by:

$$\begin{aligned}R \quad \sim \quad & 94.2 - 0.024(d_{lat} + 85) \\ & -0.11(d_{lat} - 92.3)H(d_{lat} - 92.3) - 11 \\ & -40\ln[1 + 10(e_{loss} + (1 - e_{loss})e_{de\text{-}jitter})]\end{aligned}$$

where d_{lat} is the end-to-end one-way network latency, $H(x)$ is the Heaviside step function (equal to 1 if $x >= 0$, and 0 otherwise), e_{loss} is the packet loss rate, and $e_{de\text{-}jitter}$ is the buffering delay incurred to removed jitter from the stream (de-jitter). We use this quantitative estimator of the R-score to evaluate paths in our measurements.

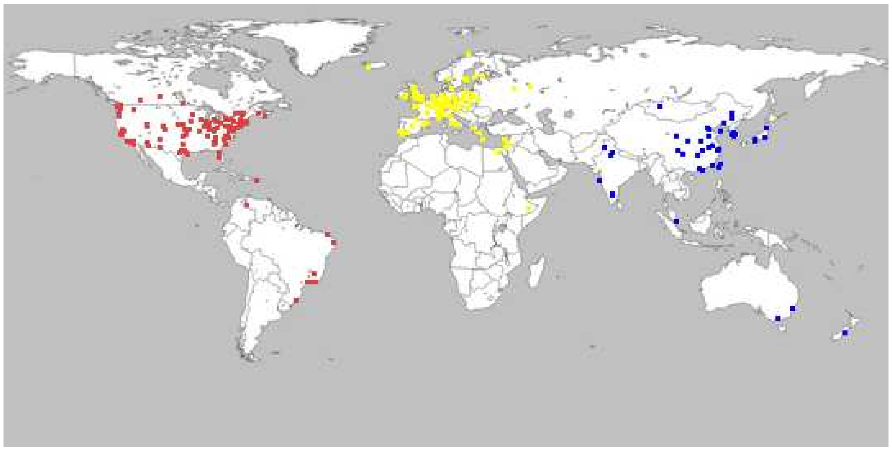

Figure 2: Location of PlanetLab nodes used in our experiments, coded by set: red for `america`, yellow for `europe`, and blue for `asia`.

2.3. Measurement Setup

Supporting aVoIP on the Internet requires forming multi-proxy paths with sufficiently low latency, loss, and jitter. To see if such paths are available on the Internet we used the PlanetLab testbed of programmable hosts (see `http://www.planet-lab.org`). As we focus here on network performance, we developed and used a lightweight measurement tool that mimicked the behavior of aVoIP proxies carrying UDP traffic. PlanetLab is composed of hosts of varying capability spread across the Internet and the world. In that sense, it is representative of hosts on the Internet in general. On the other hand, virtually all PlanetLab hosts are in academic autonomous systems (ASes), and this fact may bias the data in subtle ways — for example, many academic ASes in the US are connected through the high-speed Internet2 backbone. Also, the use of PlanetLab nodes is moderated by virtualization, so our measurement process may unable to communicate for periods when virtual nodes are inactive. Despite these limitations, our findings, which show the presence of acceptable multi-hop paths on PlanetLab, shows that good paths exist on the Internet. They certainly between PlanetLab hosts and likely elsewhere. In that sense, our results are a lower bound on the feasibility of aVoIP on the current Internet.

The paths we constructed in PlanetLab are representative of aVoIP paths, consisting of an *initiator*, two or three *proxies*, and a *responder*, as depicted in Figure 1. We are interested in performance within intracontinental and intercontinental geographic areas. We generated three sets of PlanetLab hosts, corresponding to hosts located in Asia, Europe, and the Americas (see Figure 2). Each set typically had at least 40 active hosts, as shown in Table 2. As detailed below, we pruned inactive hosts from the lists dynamically. We refer to these sets as `asia`, `europe`, and `america` in the remainder of this paper.

We performed measurements under four scenarios. For the first scenario, we constructed paths by choosing four hosts uniformly at random without replacement from `asia`. The second and third scenario were formed analogously, using hosts chosen from `europe` and `america`, respectively. The final scenario, called `intercontinental`, was generated choosing a set from among `asia`, `europe`, and `america` uniformly at random for each of the four hosts, and then choosing each host from the corresponding set. The `intercontinental` scenario is thus a superset of the other three scenarios and encompasses varying numbers of intercontinental

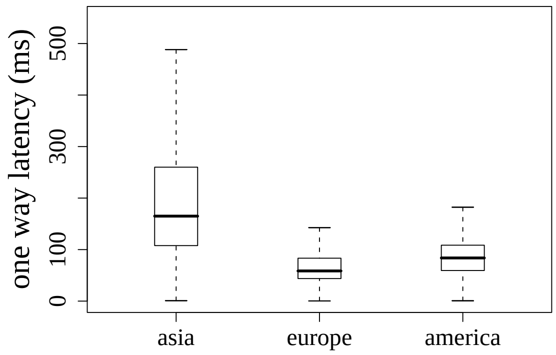

Figure 3: **Intracontinental:** Distributions of mean estimated one-way latency for paths within regions.

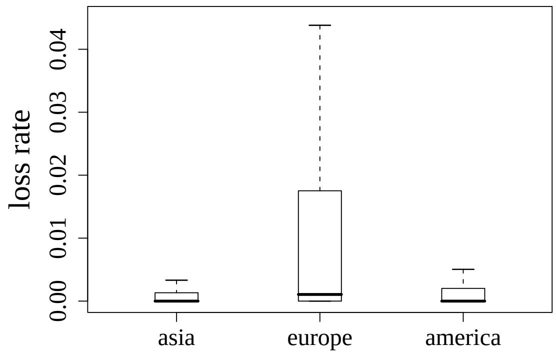

Figure 4: **Intracontinental:** Distributions of mean loss rate for paths within regions.

application-layer hops.

For each of the four scenarios, we performed measurements from Feb. 22–May 4, 2007 and examined the case of three proxies in a smaller study from Feb. 1–14, 2008. A master process performed the following steps continuously: *(i)* Choose a new set of hosts according to the scenario. *(ii)* Perform a series of 10 pings, consisting of UDP packets relayed by application layer proxies running on each host. At each host along the path, record the sending and arrival times of each packet. *(iii)* Perform a test of one-way streaming data, sending packets with an interarrival time of 20 ms, with an effective bitrate of 16 kbps; this bitrate is representative of the most common VoIP codecs and encapsulations. The actual bandwidth consumed was approximately 28.5 kbps due to IP and UDP headers. As with the pings, these UDP packets are relayed by application-layer proxies running on the hosts; each host records the sending and arrival time for each packet. Hosts went up and down over the course of our measurements. We dynamically removed hosts from the set when connections to them failed, and we occasionally probed them in a separate process, returning them to the set of active hosts if they came back up. Several PlanetLab hosts had to be permanently removed from our sets due to apparent ingress or egress filtering on their Internet connections stopping our UDP measurements. Otherwise, failures tended to be transient, and we observed an average of 210 of our 250 chosen hosts up at any one time.

3. Performance Measurements

In this section, we present the results of the measurement study described in Section 2 and discuss the implications of these results.

Measurement Results. As described in Section 2.3, we set up multi-hop paths between proxies running on PlanetLab hosts and performed measurements on these paths. From the measurements, we computed round trip times, loss rates, and jitter. We found that the clocks on PlanetLab systems are synchronized, but with an insufficient level of precision to meaningfully compute millisecond-granularity metrics. Thus, we assume that one-way latency is half of the round trip time; this is a simplifying assumption, as paths on the Internet can be asymmetric in terms of delay. If the paths are asymmetric in latency, then our measurements

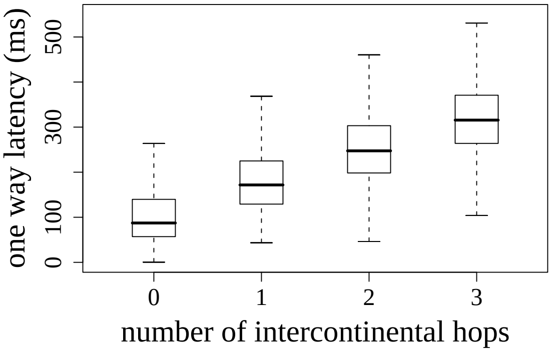

Figure 5: **Intercontinental:** Distributions of mean estimated one-way latency for a given number of hops across regions.

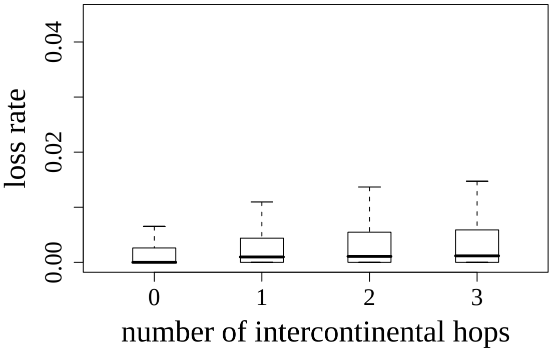

Figure 6: **Intercontinental:** Distributions of mean loss rate for a given number of hops across regions.

of latency and R-scores under-estimate the variances but do not affect the averages. Notably, we base our conclusions about aVoIP largely on the averages we observed.

The box-and-whisker plot in Figure 3 shows the distribution of the means of the *per-path one-way latency* we observed for the three intracontinental scenarios. The whiskers show the min and max values, the box is the semi-interquartile range, and the thick bar inside the box represents the median of the distribution. Figure 4 shows the *loss rate distributions* for the three intracontinental scenarios. In all three scenarios, we observe that the majority of one-way latencies are within the ITU's limit for adequate quality, 150 ms. In the `europe` and `america` scenarios, virtually all paths we observed had this property. We conjecture that the higher latencies in the `asia` set were due to higher loads on the hosts and on the links between the hosts; unfortunately, the virtualization software used by PlanetLab makes accurate measurement of these loads infeasible. We qualitatively observed that most hosts appeared to be fully loaded at all times. Loss rates were also quite low: the median loss rate was zero in the `asia` and `america` scenarios, and 0.001 in the `europe` scenario.

Figures 5 and 6 show the corresponding data for the intercontinental measurements, arranged by number of intercontinental hops. As expected, latency and loss rate increase with more hops between continents. Some of this effect can be attributed to the greater physical distance between hosts, and we conjecture that some is due to load on the network links.

Figures 7 and 8 show the distribution of the observed mean instantaneous jitter over intra- and intercontinental scenarios as a PDF. We found that the mean jitter was quite low in both scenarios: over 97% of all observed jitter was less than 3 ms, implying jitter was not a serious problem on the hosts.

As described in Section 2.2, the R-score is calculated on the basis of latency, loss, and the de-jitter buffer. To calculate the R-score for each measurement of a path, we assumed a static de-jitter buffer of 3 ms, on the basis of the jitter we observed (see Figs. 7 and 8). Any packets that were delayed longer than 3 ms due to jitter were added to the loss rate for the measurement in question. We did not assume that loss was hidden by the transport layer, i.e., we assumed no packet-level forward error correction. As a result, the R-scores we show here are conservative estimates of the quality that could be supported on the paths we examined; aVoIP deployments would likely implement adaptive jitter buffers.

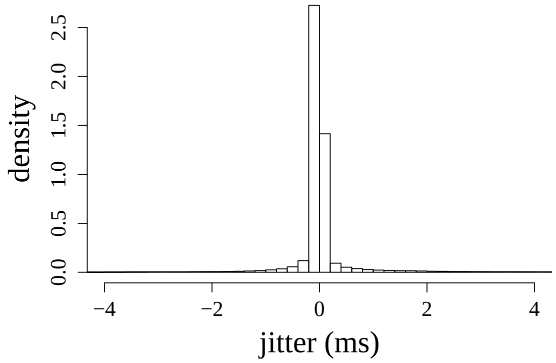

Figure 7: **Intracontinental:** Distributions of mean instantaneous jitter across all intracontinental measurements.

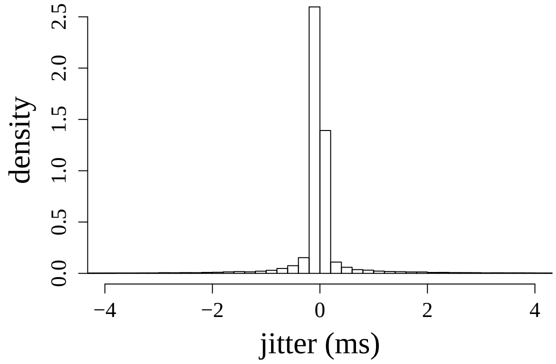

Figure 8: **Intercontinental:** Distributions of mean instantaneous jitter across all intercontinental measurements.

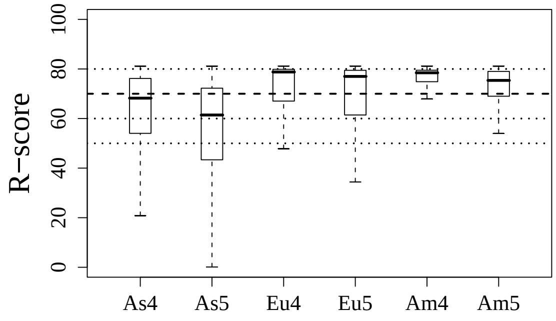

Figure 9: **Intracontinental:** R-score for paths with a given number of hops within each region.

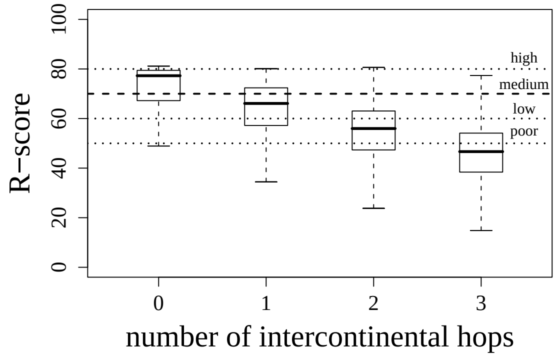

Figure 10: **Intercontinental:** R-score for paths with a given number of hops across regions.

Figures 9 and 10 show the distributions of the calculated R-scores for paths formed according to intra- and intercontinental scenarios, respectively. The path length, in terms of the number of nodes including proxies and end points, is indicated by the numerical suffix (e.g. five nodes for "As5"). The horizontal lines demarcate the varying quality levels, as described in Table 1. Of particular note is the line at $R = 70$, which is considered to be the cutoff between "medium" and "low" estimated quality. We use this cutoff to decide if connection quality is acceptable to support aVoIP.

Discussion. Our measurements suggest that aVoIP systems are feasible in most scenarios within continents and in some scenarios on the global Internet. Among intracontinental paths, 71% of all observed paths had good enough quality of service to support aVoIP. More specifically, the percentage of measurements with acceptable quality was 46% in `asia`, 71% in `europe`, and 86% in `america`. When the number of intercontinental proxies increases to three, the variance in quality increases and the average quality decreases; this decrease in path quality motivates our choice of two proxies.

The results for intercontinental quality, as shown in Figure 10, were less promising. As the number of intercontinental hops along the path increases, the percentage of our randomly formed paths with acceptable performance drops. For paths of length four (that is, with two proxies) with one such hop, 35% show acceptable performance, while two and three intercontinental hops result in only 7% and 1% of the paths exhibiting acceptable performance. This decline is clearly due to the increased latency and loss along these paths. The corresponding data for paths of three proxies is worse; overwhelming majority of paths with one or more intercontinental hops have unacceptable performance.

These results have two important implications. First, choosing proxies at random from among all such proxies in the world to form paths for aVoIP is unlikely to result in paths with acceptable performance. Thus, it will critical that aVoIP systems allow paths to be formed on the basis of path performance. Second, there is a tradeoff present between path performance and the increased security that geographically dispersed proxies provide, subject to the concerns outlined by Feamster and Dingledine [13]. Murdoch and Zielinski show that this concern is real by experimentally determining that a single Internet exchange (IX) in the UK could observe as many as 27% of Tor connections [14]. Callers located on the same continent who wish to capitalize on location diversity must utilize an even number of intercontinental hops (i.e., zero or two), and thus limit the fraction of paths available to them. Whether this limit reduces their anonymity more than the increase afforded by location diversity depends upon the threat model.

4. Attacker Measurement Manipulation

aVoIP, as we have described it in Section 2, is subject to the same threats as any similar p2p anonymous communication system, including Sybil, timing, and predecessor attacks, topics studied in depth in previous work. In this section, we focus on a vulnerability that is unique to our proposed method. To form paths with reasonable performance for aVoIP, users will have to rely on a quantitative metric of path quality, such as the R-score (see Section 2.2). A user cannot rely upon proxies to self-report performance, as malicious proxies can falsely claim higher performance then they can actually deliver [5]. Thus, we assume that users will test paths or utilize the test results provided by trusted hosts, such as the directory servers in Tor.

Even with actual network measurements conducted by a trusted party, attackers can modify performance test results so that other proxies appear to perform poorly. By doing so, the attackers become a larger percentage of the peer group providing acceptable performance. As a result, attackers are more likely to compose the full path (creating an *attacker path*), thereby speeding up a successful predecessor attack. In this section. we quantify such an attack. In particular, we show that an attacker with limited resources is not very effective in making the predecessor attack practical, but that more powerful attackers can actively leverage their position to compromise user anonymity in about half the time a passive attack would require. These results assume that the attacker can optimally degrade all mixed paths while attaining high performance for all attacker paths — a real attacker may have to adjust his attack to avoid detection and may not have high performance between all pairs of attacker-controlled proxies. We note that these attacks only affect users who form paths on the basis of path performance, and that the increased risks are a necessary cost of ensuring adequate performance for telephony.

Can the attack we describe in this section be prevented? The proposed measurement and path selection system is a distributed reputation system, and from one viewpoint, our attacker is a Sybil, that is, a single entity that controls a group of proxies. This view allows us to directly apply Cheng and Freidman's [15] result that states that this attack is impossible to defeat unless we use a reputation system that becomes ineffective for anonymous systems: one where initiators form paths with only proxies which are trusted by a central authority that is completely trusted by each initiator. Such a scenario implies a very small set of anonymous peers and accordingly little effective anonymity. The Sybil attack cannot be defeated in an inexpensive fashion [16]. However, our approach is no more susceptible than existing systems to the Sybil attack, which largely already rely on a trusted directory server.

4.1. Attacker and Measurement Model

For our attacker model, we use an adversary that can control a subset of proxies. We are primarily concerned with the predecessor attack, as it is among the most powerful attacks against onion-router-like path formation in this attacker model. Further, the use of non-random path formation affords the attacker an opportunity to optimize the attack by influencing user measurements. To conduct the predecessor attack, the attacker controls a subset of the proxies and simply waits for the user to randomly select attacker-controlled proxies for the first and last proxy on the path. The attacker, with c out of n proxies, gets this position on $\frac{c^2}{n^2}$ of the user's paths [4], and he can use timing analysis on the user's packet stream to confirm the connection between the initiator and the responder.

We model the attack on a system of n proxies with onion-router path formation, where performance measurements are taken periodically by a trusted entity (e.g., the directory server), and provided to users of the system. We assume that there are h honest proxies and c attacker-controlled proxies, such that $n = h + c$. We assume that attacker proxies all have good connections to all other proxies on the Internet, while any honest proxy may have varied levels of connection quality to the other proxies. However, we do not distinguish any honest proxies as having generally good or bad Internet connections. An honest proxy with a bad Internet connection may become isolated from other honest proxies, but we are only concerned with average performance and attacker success rates.

4.1.1. Two-proxy paths

We first evaluate paths with two intermediate proxies, as described in Figure 1. Our goals are to: (1) determine the number of expected calls (and path setups) that can take place before the attacker will occupy both end points of a path; and (2) quantify how an attacker's manipulation of measurement results can reduce the expected number of calls.

Let us divide the set of p valid paths among the n proxies into three groups: p_h *honest paths* consisting of two honest intermediate proxies, p_a *attacker paths* consisting of two attacker-controlled intermediate proxies, and p_m *mixed paths* consisting of one honest intermediate proxy and one attacker-controlled intermediate proxy. We note that $p_h/p = \frac{h\cdot(h-1)}{n\cdot(n-1)}$, $p_a/p = \frac{c\cdot(c-1)}{n\cdot(n-1)}$, and $p_m/p = \frac{2\cdot h\cdot c}{n\cdot(n-1)}$. Following on the results of our performance measurements, let $p^* \leq p$ paths provide acceptable performance (i.e., they have adequate R-scores) according to the user's requirements. Breaking this down further, $p_h^* \leq p_h$ honest paths, $p_a^* \leq p_a$ attacker paths, and $p_m^* \leq p_m$ mixed paths have adequate performance.

We assume that the user has the measured performance values from the directory server and selects only paths with adequate performance. The chance that the attacker controls a given path would normally be

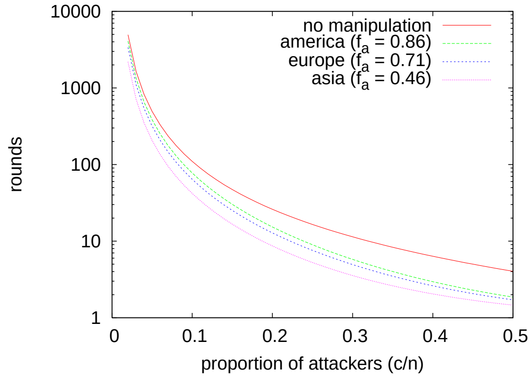

Figure 11: **Manipulating Measurements:** For different scenarios, the expected number of aVoIP calls between a pair of users required for the attacker to control the path once.

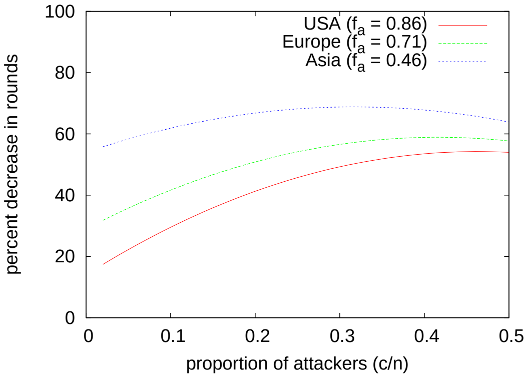

Figure 12: **Attacker Improvement:** For different scenarios, the percentage decrease of expected aVoIP calls between a pair of users required for the attacker to control the path once.

given by

$$P_{normal} = \frac{p_a^*}{p^*} = \frac{p_a^*}{p_a^* + p_h^* + p_m^*}. \tag{1}$$

On expectation, this attack would take $Calls_{normal} = 1/P_{normal}$ aVoIP calls between the same initiator and responder.

We now evaluate the attacker's attempts to manipulate measurements to increase his chances of controlling the user's paths. The attacker's ability to modify the system measurements is limited. When an attacker path is being measured, he can attempt to ensure that the path quality is adequate. Let us conservatively assume that he is always successful and that $p_a^* = p_a$. We assume that he cannot influence the measurement of honest paths. However, he can influence mixed paths. From Equation 1, we see that the attacker can increase his chance of controlling the path by lowering p_m^*. Let us assume that he is also always successful in this effort and that $p_m^* = 0$. The proxy performing the measurements could attempt to discover which proxies are manipulating measurements, but this task is difficult for several reasons. First, exhaustively measuring all paths may be infeasible, leaving the proxy with insufficient information to detect attackers. Second, the performance of paths on the Internet can fluctuate, increasing the uncertainty of detecting manipulation of the measurements. Finally, the attacker proxies can attempt to thwart detection by varying their performance over time.

Based on an attacker manipulating the measurements as described above, we see that the chance that he controls a given path is $P_{manip} = \frac{p_a}{p_a + p_h^*}$. This gives us the expected number of aVoIP calls required for the attacker to succeed as

$$E[Calls_{manip}] = \frac{p_a + p_h^*}{p_a}. \tag{2}$$

We apply Equation 2 to three scenarios based on our measurements of f_a, the fraction of adequate paths, presented in Section 3: `americas` ($f_a = 0.86$); `europe` ($f_a = 0.71$); and `asia` ($f_a = 0.46$).

Figure 11 shows the time required to conduct the attack with and without manipulation of path quality data. We see that the attacker succeeds more quickly for scenarios where the fraction of adequate paths is smaller. Accordingly, the attack succeeds more quickly for paths constructed of only Asian proxies than the other continents. This is to be expected, as we have assumed that the attacker paths are always adequate, leading to their higher proportion in the pool of adequately performing paths. We also notice that the attacker has limited gains when the attacker has few proxies in the system. For example, for 86% adequate paths and 5% attacker-controlled proxies, the attacker reduces the expected number of aVoIP calls needed for the attacker to control the path from 495 to 385, a 22% decrease. While this is a significant decrease, either number is a large number of phone calls.

When the attacker is stronger, however, manipulation of the measurements makes the predecessor attack much more effective. When the attacker controls 33% of the proxies, we get a drop from 9.4 calls to 4.6 calls — over 50%. For such an attacker, however, the number of calls is rather small in either case. Even without the attacker being able to manipulate the measurements, the system is insecure. Nevertheless, there are clearly intermediate fractions of attackers for which measurement manipulation is beneficial to accelerate the predecessor attack. A concerned user would need to limit the number of aVoIP calls she makes to the same party or to a group of parties to avoid exposure.

4.1.2. Paths with three proxies

A user who desires more security and is willing to tolerate decreased performance may choose to use three intermediate proxies. The user can construct these paths by chaining together short paths with good measurements into longer paths. In this case, the attacker's job becomes harder. If he wants to control the first and last proxy on a three-proxy path, he should not negatively manipulate the measurements of mixed two-proxy paths. In fact, the attacker should maximize the performance of such paths.

We use the same notation as above for two-proxy paths that are measured by the directory server. We also divide the set of p_3 valid three-proxy paths into three groups: p_{3a} attacker paths in which at least the first and last proxy proxies are controlled by attackers and p_{3n} *non-attacker paths* in which at least one of the first or last proxies is honest.

The user selects a two-proxy path (e.g. A to B) that defines the first two proxies, and then selects any of the two-proxy paths that starts with B and doesn't end with A. If all paths have adequate performance, the user selects an attacker path with probability

$$Pr_{3a} \quad = \quad \frac{p_a \cdot (c-2)}{p \cdot (n-2)} + 1/2 \cdot \frac{p_m \cdot (c-1)}{p \cdot (n-2)}. \tag{3}$$

The user selects a non-attacker path with probability $Pr_{3n} = 1 - Pr_{3a}$.

Since Pr_{3a} decreases if p_m decreases, Pr_{3n} necessarily increases if p_m decreases. Thus, the attacker benefits from keeping p_m as high as possible. If the attacker manipulates the measurements to make $p^*_m = p_m$ as well as $p^*_a = p_a$, then the chance that the user selects an attacker path is given by

$$Pr^*_{3a} \quad = \quad \frac{p_a \cdot (c-2)}{p^* \cdot (n-2)} + 1/2 \cdot \frac{p_m \cdot (c-1)}{p^* \cdot (n-2)}. \tag{4}$$

We can calculate $p^* = p_a + p_m + p^*_h$. We note that p^*_h may be lower than for two-proxy paths, as the path quality requriments may be higher when paths are combined.

Table 3: Comparison with Related Measurement Studies

Study	**Protocol tested**	**Geographic Breadth**	**Application-layer hops**	**Primary Metric**
Markopoulou et al. [17]	VoIP	US-only	0	MOS
Furuya et al. [18]	VoIP	Emulated	0	PESQ MOS
Barbosa et al. [19]	Skype and GoogleTalk	Emulated	0	PESQ MOS
Suh et al. [20]	Skype	International	1	N/A
Skevik et al. [21]	Video streaming	International	0	Various
Tang et al. [22]	Video	International	1	Various
aVoIP (this work)	Emulated VoIP	International	2-3	R-score

To see the effect of this manipulation, we apply the these probabilities to an example scenario as above. For example, for 86% adequate paths and 5% attacker-controlled proxies, the attacker reduces the expected number of aVoIP calls needed for the attacker to control the path from 495 to 432, a 13% decrease. This is somewhat less of a decrease than for the two-proxy paths, and it assumes that the attacker can increase the performance of mixed paths, which is unlikely in general. If the attacker controls 33% of the proxies, the number of calls drops from 9.4 to 8.8 calls — only 6.5%.

5. Related Work

In this section, we first discuss related work in measuring the performance of streaming media, especially VoIP, on the Internet. We then compare the proposed aVoIP architecture with other recent work in anonymity systems.

1) Voice over IP. Deploying aVoIP does not present significant challenges to standard VoIP signaling, encoding, transport, or gateway control. We refer the reader to Goode [23] and Mehta and Udani [24] for more details on these systems. The main problem in deploying VoIP over paths consisting of multiple proxies is a degraded quality of service in terms of latency, jitter, and packet loss. Degraded QoS can also be the result of potential aVoIP security measures not present in VoIP (nor in our aVoIP design), including purposeful delays to defeat watermarking or timing attacks [8, 25, 9].

A number of papers have analyzed the performance of streaming multimedia communication over Internet. We now compare the study presented in this paper with the most relevant work; the comparison is summarized in Table 3. Markopoulou et al. [17] measured VoIP quality on the backbone path connecting five US cities in terms of a Mean Opinion Score (MOS), which is related to our use of the E-Model (see Section 2). Furuya et al. [18] perform network emulation experiments to test the effect of different network parameters on VoIP quality. They use the Perceptual Evaluation of Speech Quality (PESQ), which they note is well-correlated with MOS. Barbosa et al. [19] compare the performance of Skype and Google Talk under varied network packet loss and delays using a network emulator. Rossi et al. [26] study the Skype signaling mechanism in detail. They report that the mechanism uses latency as a key metric for probing peers, but they focus primarily on the breakdown of when peers probe rather than the probe results (which are internal to the Skype system). Skevik et al. [21] study P2P video-on-demand streaming performance using PlanetLab. Their proposed architecture uses pings to estimate round trip times between peers.

The primary distinction of our work from most studies is the use of multiple proxies along the path for anonymity and the large scale of our measurement study. To our knowledge, no previous work has evaluated an aVoIP network scenario. The study that is perhaps closest to ours is by Suh et al. [20] which characterized Skype traffic sent through one proxy, with the goal of reliably detecting the traffic. Their study was limited to a single proxy and did not determine how the relative locations of proxies affects performance. In fact, to the authors' knowledge, our work appears to be unique in its comparison of different performance in different regions. Tang et al. [22] actually show that using one application-layer hop in an overlay can *improve* performance for multimedia; intuitively, the overlay finds good paths that route around transient network failures and congestion without waiting for BGP convergence. Since we must balance privacy considerations with performance, their techniques and results are not applicable to our problem.

2) Low-Latency Anonymity Systems. There are many designs for *low-latency* anonymous systems [6, 27, 28], where the term low-latency is relative to very high-latency, anonymous email message systems. Ren and Wu provide a useful survey of the field [29]. The performance of real-time applications with stricter performance requirements than Web browsing has not been measured over a multi-proxy anonymity network. The use of TCP as the underlying transport protocol in such systems, including Tor, presents a significant obstacle. This is due to TCP's congestion control mechanisms, which adapt to packet loss to ensure fairness of resources and can stop sending depending upon network conditions. Multimedia applications, ranging from games to video conferencing to VoIP, usually rely on UDP streams and application-level congestion control.

Table 4 describes the differences between aVoIP and prior work. Tor [6] does allow users to be aware of the available bandwidth of proxies within the system. At present, the proxies self-report their bandwidth, rendering it unreliable and and generally unused in path formation, as an attacker might claim high bandwidth in order to facilitate an attack. Bauer et al. [5] describes such an attack. Snader and Borisov [30] describe a measurement-based approach to using bandwidth for path selection. While these measurements can be useful for finding good nodes, it may not be suitable for VoIP. The relative bandwidth costs of VoIP are low, while the latency between two high-bandwidth nodes may be too high for voice. The system we present in Section 2 also measures path performance rather than self-reported proxy performance, but it is designed to ensure high voice quality.

Reardon and Goldberg [31] describe an improvement to Tor that multiplexes TCP streams over UDP. While they report that the approach substantially improves delays over Tor for TCP connections, it does not use path selection ensure a low-latency connection. Further, connections still go over TCP, meaning that streaming media does not get handled efficiently. Sherr, Blaze, and Loo [32] describe the framework most similar to our latency-measurement approach. They propose using a latency-measurement technique that includes secure network coordinates as a basis. However, support for streaming media like VoIP is not included, as connections go over TCP.

In an approach that differs from these anonymity systems, Karopoulos et al. present study of identifier

Table 4: Comparison with Related Systems

System Name	**Support for VoIP**	**Performance enhancements**	**Performance concerns**
Tor [6]	Only VoIP over TCP	Reported bandwidth	Random path selection; streaming over multi-hop TCP
Snader-Borisov [30]	Only VoIP over TCP	Bandwidth-based selection	No assurance of a low-latency path; streaming over multi-hop TCP
Reardon-Goldberg [31]	Only VoIP over User-level TCP	Reduced congestion control between Tor nodes	Random path selection; streaming over multi-hop TCP
Sherr-Blaze-Loo [32]	Only VoIP over TCP	Latency measurement	Streaming over multi-hop TCP
aVoIP (this work)	VoIP over UDP	Latency measurement	None

privacy for the Session Initiation Protocol (SIP) used in VoIP [33]. While the identifiers are protected in their proposed system, the IP addresses can still be used to monitor the participants; thus, their approach should be layered on top of the protocol we propose in this paper.

6. Conclusion

VoIP is a popular service on the Internet that can easily provide confidentiality but not anonymity. We have investigated the performance and security issues involved in offering an anonymous VoIP service over the Internet using a Tor-like infrastructure. Our performance evaluation of this system is based on ten weeks of measurements over paths of multiple proxies on PlanetLab. These measurements show that aVoIP has sufficient voice quality for proxies that are located within the same broad geographic area. However, our study also suggests that multihop intercontinental paths, which have a likely advantage in protecting against powerful eavesdroppers, cannot be supported on the current Internet. In sum, aVoIP paths cannot be constructed in a way that is oblivious to the performance of the underlying network.

We evaluated the security of path selection for aVoIP, given that path quality introduces a new attack. Our analysis demonstrates that attackers can leverage their position to make predecessor attacks faster; fortunately, this optimization is most powerful only when the predecessor attack is already very efficient. Overall, we have shown that it is possible to make aVoIP with acceptable performance a reality while retaining most of the security benefits of onion routing against compromised proxies.

References

[1] R. Le Maistre, Skype: Revenue Doubles, Growth Slows, *Light Reading*, `http://www.lightreading.com/document.asp?doc_id=122229` (April 27, 2007).

[2] Intl. Telecommunication Union, ITU-T G.114, One-way transmission time, `http://www.itu.int/ITU-T/publications/recs.html` (May 2003).

[3] D. Kesdogan, C. Palmer, Technical challenges of network anonymity, Computer Communications 29 (3) (2006) 306 – 324.

[4] M. Wright, M. Adler, B. Levine, C. Shields, The predecessor attack: An analysis of a threat to anonymous communications systems, ACM TISSEC 4 (7) (2004) 489–522.

[5] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, D. Sicker, Low-resource routing attacks against Tor, in: ACM WPES, 2007.

[6] R. Dingledine, N. Mathewson, P. Syverson, Tor: The next-generation onion router, in: Proc. of the USENIX Security Symposium, 2004.

[7] C. Tang, P. K. McKinley, Topology-aware overlay path probing, Computer Communications 30 (9) (2007) 1994 – 2009.

[8] B. Levine, M. Reiter, C. Wang, M. Wright, Timing attacks in low-latency mix systems, in: Proc. Financial Cryptography, 2004, pp. 251—265.

[9] X. Wang, S. Chen, S. Jajodia, Tracking anonymous peer-to-peer VoIP calls on the Internet, in: Proc. ACM CCS, 2005.

[10] S. J. Murdoch, G. Danezis, Low-cost traffic analysis of Tor, in: Proc. IEEE Symp. on Security and Privacy, 2005.

[11] N. Hopper, E. Y. Vasserman, E. Chan-Tin, How much anonymity does network latency leak?, in: Proc. ACM CCS, 2007.

[12] R. G. Cole, J. H. Rosenbluth, Voice over IP performance monitoring, SIGCOMM Computer Commununications Review 4 (3).

[13] N. Feamster, R. Dingledine, Location diversity in anonymity networks, in: Proc. WPES, 2004.

[14] S. J. Murdoch, P. Zieliński, Sampled traffic analysis by Internet-Exchange-Level adversaries, in: Proc. Privacy Enhancing Technology, Springer, 2007.

[15] A. Cheng, E. Friedman, Sybilproof reputation mechanisms, in: ACM Workshop on the Economics of Peer-to-Peer Systems, 2005, pp. 128–132.

[16] J. Douceur, The Sybil attack, in: Proc. IPTPS, 2002.

[17] A. P. Markopoulou, F. A. Tobagi, M. J. Karam, Assessment of VoIP quality over Internet backbones, in: Proc. of IEEE INFOCOM, 2002.

[18] H. Furuya, S. Nomoto, H. Yamada, N. Fukumoto, F. Sugaya, Experimental investigation of the relationship between IP network performances and speech quality of VoIP, in: Proc. International Conference on Telecommunications (ICT 2003), 2003.

[19] R. Barbosa, C. Kamienski, D. Mariz, A. Callado, S. Fernandes, D. Sadok, Performance evaluation of P2P VoIP applications, in: Proc. ACM NOSSDAV, 2007.

[20] K. Suh, D. R. Figueiredo, J. Kurose, D. Towsley, Characterizing and detecting relayed traffic: A case study using Skype, in: Proc. of IEEE Infocom, 2006.

[21] K.-A. Skevik, V. Goebel, T. Plagemann, Evaluation of a comprehensive P2P video-on-demand streaming system, Computer Networks 53 (4) (2009) 434 – 455.

[22] L. Tang, Z. Chen, H. Yin, J. Li, Y. Li, CORS: A cooperative overlay routing service to enhance interactive multimedia communications, Journal of Visual Communication and Image Representation 21 (2) (2010) 107 – 119, special issue on Network Technologies for Emerging Broadband Multimedia Services.

[23] B. Goode, Voice over Internet protocol (VOIP), in: Proc. of the IEEE, Vol. 90, 2002, pp. 1495–1517.

[24] P. C. Mehta, S. Udani, Overview of voice over IP, Technical Report MS-CIS-01-31, University of Pennsylvania (February 2001).

[25] G. Danezis, The traffic analysis of continuous-time mixes, in: Proc. Privacy Enhancing Technologies, Vol. 3424 of LNCS, 2004, pp. 35–50.

[26] D. Rossi, M. Mellia, M. Meo, Understanding Skype signaling, Computer Networks 53 (2) (2009) 130 – 140.

[27] O. Berthold, H. Federrath, S. Köpsell, Web MIXes: A system for anonymous and unobservable Internet access, in: Proc. Privacy Enhancing Technology, 2000.

[28] A. Pfitzmann, M. Waidner, Networks without user observability, Computers & Security 6 (2) (1987) 158 – 166.

[29] J. Ren, J. Wu, Survey on anonymous communications in computer networks, Computer Communications 33 (4) (2010) 420 – 431.

[30] R. Snader, N. Borisov, A tune-up for Tor: Improving security and performance in the Tor network, in: Proc. NDSS, 2007.

[31] J. Reardon, I. Goldberg, Improving Tor using a TCP-over-DTLS tunnel, in: Proc. USENIX Security Symposium, 2009.

[32] M. Sherr, M. Blaze, B. T. Loo, Scalable link-based relay selection for anonymous routing, in: 9th Privacy Enhancing Technologies Symposium (PETS '09), 2009.

[33] G. Karopoulos, G. Kambourakis, S. Gritzalis, E. Konstantinou, A framework for identity privacy in SIP, Journal of Network and Computer Applications 33 (1) (2010) 16 – 28.