An Experimental Study of the Skype Peer-to-Peer VoIP System

Saikat Guha
Cornell University
saikat@cs.cornell.edu

Neil Daswani
Stanford University
daswani@cs.stanford.edu

Abstract

Despite its popularity, relatively little is known about the traffic characteristics of the Skype VoIP system and how they differ from other P2P systems. We describe an experimental study of Skype VoIP traffic conducted over a one month period, where over 30 million datapoints were collected regarding the population of online clients, the number of supernodes, and their traffic characteristics. The results indicate that although the structure of the Skype system appears to be similar to other P2P systems, particularly KaZaA, there are several significant differences in traffic. The number of active clients shows diurnal and work-week behavior, correlating with normal working hours regardless of geography. The population of supernodes in the system tends to be relatively stable; thus node churn, a significant concern in other systems, seems less problematic in Skype. The typical bandwidth load on a supernode is relatively low, even if the supernode is relaying VoIP traffic.

The paper aims to aid further understanding of a significant, successful P2P VoIP system, as well as provide experimental data that may be useful for design and modeling of such systems. These results also imply that the nature of a VoIP P2P system like Skype differs fundamentally from earlier P2P systems that are oriented toward file-sharing, and music and video download applications, and deserves more attention from the research community.

1 Introduction

Email was the original killer application for the Internet. Today, voice over IP (VoIP) and instant messaging (IM) are fast supplementing email in both enterprise and home networks. Skype is an application that provides these VoIP/IM services in an easy-to-use package that works behind NAT/firewalls; it has attracted a user-base of 50 million users, and is considered valuable enough that eBay Inc. recently acquired it for more than $2.6 billion. In this paper, we conduct a measurement study of the Skype P2P VoIP network. While measurement studies of both P2P file-sharing networks [24, 25, 2, 12, 19] and "traditional" VoIP systems [13, 16, 3] have been performed in the past, little is known about VoIP systems that are built using a P2P architecture.

One of our key goals in this paper is to understand how P2P VoIP traffic in Skype differs from traffic in P2P file-sharing networks and from traffic in traditional voice-communication networks. Do Skype users leave their client on for days, or start it just to make a call and close it soon afterwards, like file-sharing users [12]? We find that unlike file-sharing users, Skype users regularly run the client during normal working hours and close it in the evening, leading to different network dynamics. Does the fact that Skype calls are free encourage users to talk for longer than they do on telephones where calls are charged by the minute? We find evidence to the affirmative. Does Skype really need the resources of millions of peers to provide a global VoIP service, or can a global VoIP service be supported by dedicated infrastructure? We find that the median network utilization in Skype peers is very low, but that peak usage can be high.

Overall, our work makes three contributions. First, in §2, we shed light on some design choices in the proprietary Skype network and how they affect robustness and scalability. Second we analyze node dynamics and churn in Skype's peer-to-peer overlay, and the network workload generated by Skype users in §3 and §4 respectively. Third, we provide data on user-behavior that can be used for design and modeling of peer-to-peer VoIP networks. Altogether, we find evidence that Skype is fundamentally different from the peer-to-peer networks studied in the past.

2 Skype Overview

Skype offers three services: *VoIP* allows two Skype users to establish two-way audio streams with each other and supports conferences of up-to 4 users, *IM* allows two or more Skype users to exchange small text messages in real-time, and *file-transfer* allows a Skype user to send a file to another Skype user (if the recipient agrees)[1]. Skype also offers paid services that allow

[1] This is different from *file-sharing* in Gnutella, KaZaA and BitTorrent, where users request files that have been previously published.

Skype users to initiate and receive calls via regular telephone numbers through VoIP-PSTN gateways.

Despite its popularity, little is known about Skype's encrypted protocols and proprietary network. In [10], Garfinkel concludes that Skype is related to KaZaA; both the companies were founded by the same individuals, there is an overlap of technical staff, and that much of the technology in Skype was originally developed for KaZaA. Network packet level analysis of KaZaA [15] and of Skype [1] support this claim by uncovering striking similarities in their connection setup, and their use of a "supernode"-based hierarchical peer-to-peer network.

Supernode-based peer-to-peer networks organize participants into two layers – supernodes, and ordinary nodes. Such networks have been the subject of recent research in [28, 27, 5, 4]. Typically, *supernodes* maintain an overlay network among themselves, while *ordinary nodes* pick one (or a small number of) supernodes to associate with; supernodes also function as ordinary nodes and are elected from amongst them based on some criteria. Ordinary nodes issue queries through the supernode(s) they are associated with.

We observed that in Skype, ordinary nodes send control traffic including availability information, instant messages, and requests for VoIP and file-transfer sessions over the supernode peer-to-peer network. If the VoIP/file-transfer request is accepted, the Skype clients establish a direct connection between each other. To determine this, we ran two Skype clients on separate hosts, and observed the destination and source IP addresses for packets sent and received in response to various application-level tasks. We repeated the experiment for a single client behind a NAT[2], and both clients behind different NATs. We observed that if one client is behind a NAT, Skype uses connection reversal whereby the NAT'ed node initiates the TCP/UDP media session regardless of which end requested the VoIP or file-transfer session. If both clients are behind NATs, Skype uses STUN-like NAT traversal [23, 9] to establish the direct connection. In the event that NAT traversal fails, Skype falls back to a TURN-like [22] approach where the media session is relayed by a publicly reachable supernode.

Consequently, Skype supernodes are chosen from nodes that have plenty of spare bandwidth, and are publicly reachable. In an experiment we conducted, we ran several Skype nodes in various environments and waited two weeks for them to become supernodes. A Skype node behind a saturated network uplink, and one behind a NAT did not become supernodes, while a fresh install on a public host with a 10 mbps connection to the Internet joined the supernode network within minutes. We did not test additional criteria such as a history of long session times, or low processing load as suggested in [27]. As we show later, the population of supernodes selected by Skype, apparently based on reachability and spare bandwidth, tends to be relatively stable. Skype, therefore, represents an interesting point in the P2P design-space where heterogeneity is leveraged to *control* churn, not just cope with it.

[2]We overload NAT to mean NATs and firewalls

3 Methodology

In order to understand the Skype network, we performed three experiments in parallel. In the first experiment, we observed the network activity of a Skype supernode for 40 days. We used ethereal [7] to capture the 3GB of data sent and received by the supernode during this time, including relayed VoIP and file-transfer sessions.

In the second experiment, we discovered IP addresses and port numbers of supernodes. We wrote a script that parses the Skype client's supernode-cache and adds the addresses in the cache to a list. Our script then replaces the cache with a single supernode address from the list such that the client is forced to pick that supernode the next time the client is run. The script starts the client and waits for it to download a fresh set of supernode addresses from the supernode to which it connects. The script then kills the client causing it to flush its supernode-cache. The cache is processed again and the entire process repeated; the result is a crawl of the supernode network which discovers supernode addresses. Our on-going experiment has discovered 250K supernode addresses, and has crawled 150K of them. As a side-effect, the script also records the number of online Skype users each time the client is run, as reported by the Skype client.

In the third experiment, we gathered "snapshots" of which supernodes were online at a given time. We wrote a tool that sends application-level pings to supernodes; the tool replays the first packet sent by a Skype client to a supernode in its cache, and waits for an expected response. For each snapshot, we perform parallel pings to a fixed set of 6000 nodes randomly selected from the set of supernodes discovered in the second experiment. Each snapshot takes 4 minutes to execute. These snapshots are taken at 30 minute intervals for one month.

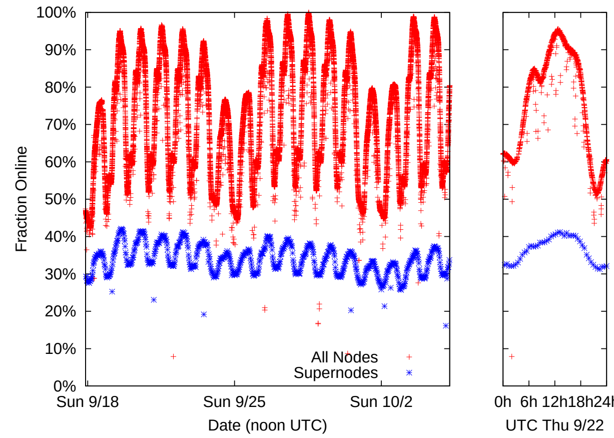
4 Characterizing Skype's Network

Churn in P2P networks, the continuous process of nodes joining and leaving the system, increases routing latency as some overlay traffic is routed through failed nodes, while some new nodes are not taken advantage of. Many peer-to-peer networks handle churn by dynamically restructuring the network through periodic or reactive maintenance traffic. Churn has been studied extensively in peer-to-peer file-sharing networks [24, 25, 2, 12, 6]; the consensus is that churn can be high, average node *session times*[3] can be as low as a few minutes, and that frequent updates are needed to maintain consistency [21]. In this section, we measure churn in Skype's supernode P2P network. We find that there is *very little churn* in the supernode network, and that supernodes demonstrate *diurnal behavior* causing median session times of several hours. Further, we find that *session lengths are heavy-tailed* and are not Poisson distributed.

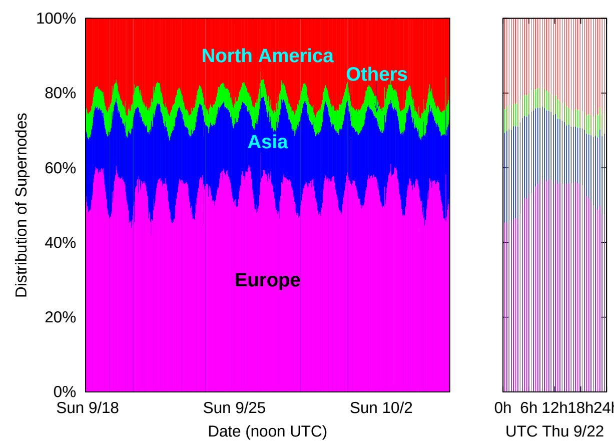
Figure 1(a) shows that the number of Skype supernodes is more stable than the number of online Skype users. The figure is split into two parts for clarity; the plot on the left tracks daily variations in client and supernode populations from Sep. 18 to Oct. 4, while the plot on the right zooms-in on hourly variations on Sep. 22. As mentioned in Section 3, the number of online users is reported by the official Skype client, while online supernodes are determined through periodic application-level pings. There are large diurnal variations with peak usage during normal working hours and significantly reduced usage (40–50%) at night. In addition, there are weekly variations with 20% fewer users online on weekends than on weekdays. The maximum number of users online was 3.9 million on Wednesday, Sep. 28 around 11am EST. In comparison, of the 6000 randomly-selected supernodes pinged, only 2078 responded to pings at least once during our trace, and between 30–40% of them are online at any given time. While client population varies by over 40% on any given day, supernode population is more stable and varies by under 25%.

Figure 1(b) confirms that Skype usage peaks during normal working hours. The graph plots the geographic distribution of active supernodes. Europe accounts for 45–60% of supernodes, with its contribution peaking around 11am UTC (mid-day over most of Europe). North America contributes 15–25% of supernodes, with a peak contribution around noon CST. Similarly, Asia contributes 20–25% and peaks around its mid-day. Combined with the lower weekend-usage from the previous graph, there is evidence to conclude that Skype usage, at least for those nodes that become su-

[3]Time between when a node joins the network, and then subsequently leaves.



(a)



(b)

Figure 1: (a) Percentage of all nodes, and supernodes active at any time. (b) Geographic distribution of supernodes as observed over the duration of our trace.

pernodes, is correlated with normal working hours. This is different from P2P file-sharing networks where users download files in batches that are processed over days, sometimes weeks [12].

Session times reflect this correlation with normal working hours. As has been observed widely for interactive applications like telnet, web, and email [18, 8], node arrivals in Skype are concentrated towards the morning, while departures are concentrated towards the evening (Figure 2). Figure 2 plots the fraction of supernodes joining and leaving the network in consecutive snapshots taken at 30 minute intervals. The median supernode session time from the same experiment is 5.5 hours, as shown in Figure 3. The median is higher than reported in previous studies of file-sharing

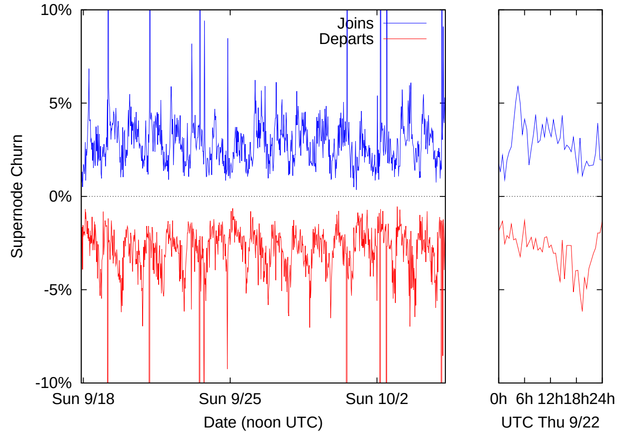


Figure 2: Fraction of supernodes joining or departing the network over the duration of our trace.

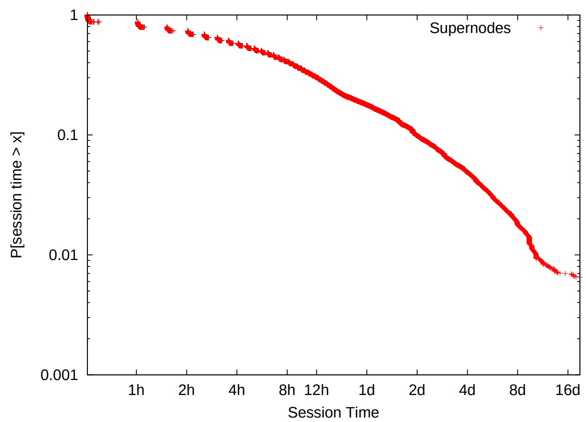


Figure 3: Semi-log plot of the complimentary CDF of supernode session times.

networks [24, 25, 2, 12, 6]; however, these studies measure session times for all nodes and not just the supernodes that form the P2P overlay. We find that the session time for supernodes has a heavy tail, and our data can be modeled as Pareto or Weibull with shape parameter 0.64. This non-exponential nature of the distribution suggests that both arrivals and departures in Skype are not Poisson or uniform. Consequently, results from past work that model churn as fixed-rate Poisson processes [21, 4, 14] may be misleading if directly applied to Skype. One way to model churn in Skype is to model node arrival as a Poisson process with varying hourly rates (higher in the morning), and picking session times from a Weibull or Pareto distribution, similar to the approach in [20]. Nevertheless, since there is little churn in the first place (more than 95% of supernodes persist from one thirty-minute snapshot to the next), we expect periodic updates with an update-rate chosen accordingly to perform well [21]. As an optimization, reactive up-

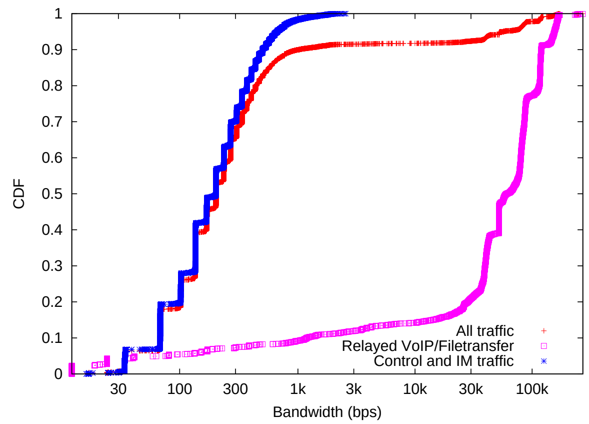


Figure 4: Semi-log plot of CDF of bandwidth used by our supernode.

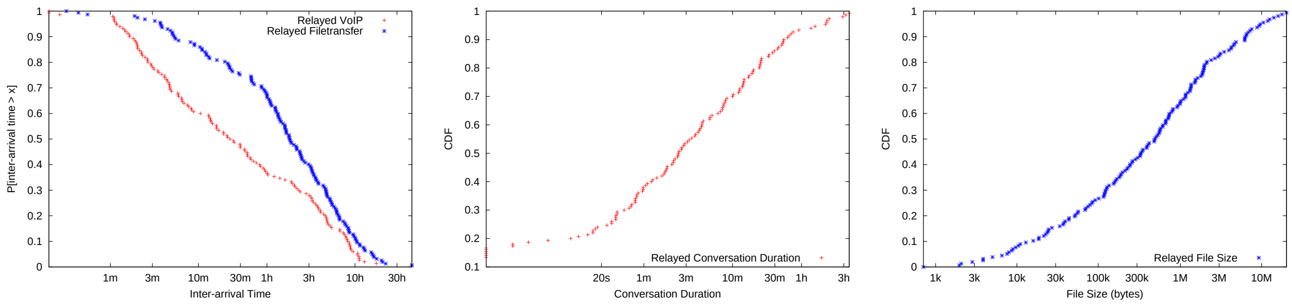
dates can be used to conserve bandwidth at night, when there is little churn.

5 VoIP in Skype

Skype uses spare network and computing resources of hundreds of thousands of supernodes, and little additional infrastructure to handle calls, as compared to traditional telephone companies and wireless carriers who rely on expensive, dedicated, circuit-switched infrastructure. In this section, we analyze the role this peer-to-peer network plays in the context of VoIP. We find that Skype supernodes incur a *small network cost* for participating in the Skype network. In addition, Skype's use of peer-to-peer represents a convenient looking-glass into a global VoIP/IM network. In that regard, we observe that Skype calls *last longer* than calls in traditional telephone networks, and that files transferred are *smaller* than in file-sharing networks.

Figure 4 shows that our Skype supernode uses very little bandwidth most of the time. Bandwidth used by our supernode is plotted for 30 second intervals. Fifty-percent of the time, our supernode consumes less than 205 bps[4]. We separate out low-bandwidth control and IM traffic, and high-bandwidth, relayed VoIP and file-transfer traffic; due to Skype's use of encryption, however, we resort to statistical approaches for this, which may misclassify small file-transfers as control traffic. The supernode is engaged in relaying connections 9.6% of the time. This value is smaller than we expected; it is explained by Skype's use of NAT traversal [26] that successfully establishes direct VoIP/file-transfer sessions through many NATs. For relayed data, the supernode uses 60 kbps in the median. We observe

[4] bits per second



(a) (b) (c)

Figure 5: (a) Semi-log plot of CCDF of inter-arrival time of relayed VoIP and file-transfer sessions. (b) Semi-log plot of CDF of Skype VoIP conversation durations. (c) Semi-log plot of CDF of file-transfer sizes.

that Skype does not use silence suppression and sources 33 packets per second for all VoIP connections regardless of speech characteristics. In [25], Sen et al. find that half the users of a P2P file-sharing network have upstream bandwidth greater than 56 kbps; Skype can take advantage of such nodes, when publicly reachable, to act as supernodes. In addition to the network bandwidth, our supernode consumed negligible additional processing power, memory and storage as compared to an ordinary node.

Figure 5 offers some insights into Skype user behavior. These results are preliminary: first, encryption prevents us from looking into all control traffic, and we therefore are limited to analyzing user behavior only for relayed VoIP/file-transfer sessions and not IM or direct sessions. This potentially introduces an unavoidable bias in our user population. Second, this causes us to miss an estimated 85% of the data (based on [11]), resulting in only 300 data points for 30 days[5]. Not withstanding, we believe that even these preliminary results show interesting trends and, to the best of our knowledge, represent the first publicly available measurements of call parameters in a VoIP network.

Figure 5(a) suggests that inter-arrival time of relayed VoIP sessions and file-transfer sessions are not Poisson. VoIP connection arrivals in our sample show the characteristics of power-law behavior. File-transfers are initiated less frequently; note, however, that file-transfer in this context refers to a user sending a file to another user (much like email attachments), and are different from file-sharing. Figure 5(b) shows that the median Skype call lasted 2m 26s, while the average was 19m 11s. The longest relayed call lasted for 3h 26m. The average call duration is much higher than the 3-minute average for traditional telephone calls [17]. One reason for this difference may be that Skype-to-Skype VoIP is free, while phone calls are charged. The median file-transfer size is 500 kB (Figure 5(c)). The size is similar to documents, presentations and photos, and is much smaller than audio files in file-sharing networks [24]. Altogether, we find that Skype users behave differently from file-sharing users as well as traditional telephone users.

6 Future Work

We have only scratched the surface of understanding how peer-to-peer supports VoIP. More generally, interactive applications such as peer-to-peer web-caching, VoIP, instant messaging, games etc. may demonstrate different characteristics than P2P file-sharing networks and we are interested in understanding these differences. Measuring existing interactive networks including instant messaging networks (AIM, MSN, Yahoo!) and massively multiplayer game networks (World of Warcraft, Ultima Online) can reveal different user behavior. In addition, it would be useful to compare user experience, call setup latency and call quality in Skype and other infrastructure-based telephony services including traditional telephone and cellular networks, and VoIP networks that use SIP and H.323 for signaling. Combined, these would give insights about how peer-to-peer networks for such applications should be built and provisioned.

7 Conclusions

This paper presents the first measurement study of the Skype VoIP system. Skype differs significantly from other peer-to-peer file-sharing networks in several re-

[5] at the time of submission. The final version of this paper will have about 1500 data points from an on-going 5-month trace

spects. Active clients show diurnal and work-week behavior analogous to web-browsing rather than file-sharing. Stability of the supernode population tends to mitigate churn in the network. Skype calls are significatly longer than calls in traditional telephone networks, while files transferred over Skype are significantly smaller than those over file-sharing networks. Supernodes typically use little bandwidth even though they relay VoIP and file-transfer traffic in certain cases. Overall, we present measurement data useful for designing and modeling a peer-to-peer VoIP system. Even though this data is limited due to the proprietary nature of Skype, we believe that this study forms a basis for understanding and discussing the differences between peer-to-peer file-sharing and peer-to-peer VoIP systems.

Acknowledgments

The authors wish to thank Ravi Jain for valuable feedback on early drafts of this paper.

References

[1] BASET, S. A., AND SCHULZRINNE, H. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. Tech. Rep. CUCS-039-04, Computer Science Department, Columbia University, New York, NY, Sept. 2004.

[2] BHAGWAN, R., SAVAGE, S., AND VOELKER, G. M. Understanding availability. In *Proceedings of the IPTPS '03* (Berkeley, CA, Feb. 2003).

[3] CALYAM, P., SRIDHARAN, M., MANDRAWA, W., AND SCHOPIS, P. Performance measurement and analysis of h.323 traffic. In *Proceedings of the 5th International Workshop on Passive and Active Network Measurement (PAM 2004)* (Antibes Juan-les-Pins, France, Apr. 2004).

[4] CASTRO, M., COSTA, M., AND ROWSTRON, A. Debunking some myths about structured and unstructured overlays. In *Proceedings of the NSDI '05* (Boston, MA, May 2005).

[5] CHAWATHE, Y., RATNASAMY, S., BRESLAU, L., LANHAM, N., AND SHENKER, S. Making gnutellalike p2p systems scalable. In *Proceedings of SIGCOMM '03* (Karlsruhe, Germany, Aug. 2003).

[6] CHU, J., LABONTE, K., AND LEVINE, B. N. Availability and locality measurements of peer-topeer file systems. In *Proceedings of ITCom '02* (Boston, MA, July 2002).

[7] COMBS, G. Ethereal: A Network Protocol Analyzer.

[8] CROVELLA, M. E., AND BESTAVROS, A. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking 5*, 6 (Dec. 1997), 835–846.

[9] FORD, B., SRISURESH, P., AND KEGEL, D. Peer-to-peer communication across network address translators. In *Proceedings of the 2005 USENIX Annual Technical Conference* (Anaheim, CA, Apr. 2005).

[10] GARFINKEL, S. L. VoIP and Skype Security, Jan. 2005.

[11] GUHA, S., AND FRANCIS, P. Characterization and measurement of tcp traversal through nats and firewalls. In *Proceedings of the 2005 Internet Measurement Conference* (New Orleans, LA, Oct. 2005).

[12] GUMMADI, K. P., DUNN, R. J., SAROIU, S., GRIBBLE, S. D., LEVY, H. M., AND ZAHORJAN, J. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proceedings of the SOSP '03* (Bolton Landing, NY, Oct. 2003).

[13] JIANG, W., AND SCHULZRINNE, H. Assessment of voip service availability in the current internet. In *Proceedings of the 4th International Workshop on Passive and Active Network Measurement (PAM 2003)* (La Jolla, CA, Apr. 2003).

[14] LI, J., STRIBLING, J., MORRIS, R., KAASHOEK, M. F., AND GIL, T. M. A performance vs. cost framework for evaluating dht design tradeooffs under churn. In *Proceedings of the INFOCOM '05* (Miami, FL, Mar. 2005).

[15] LIANG, J., KUMAR, R., AND ROSS, K. W. The kazaa overlay: A measurement study. *Computer Networks 49*, 6 (Oct. 2005).

[16] MARKOPOULOU, A. P., TOBAGI, F. A., AND KARAM, M. J. Assessing the quality of voice communications over internet backbones. *IEEE/ACM Transactions on Networking 11*, 5 (Oct. 2003), 747–760.

[17] NOLL, A. M. Cybernetwork technology: Issues and uncertainties. *COMMUNICATIONS OF THE ACM 39*, 12 (Dec. 1996), 27–31.

[18] PAXSON, V., AND FLOYD, S. Wide-area traffic: the failure of poisson modeling. In *Proceedings of the SIGCOMM '94* (London, UK, Aug. 1994).

[19] POUWELSE, J., GARBACKI, P., EPEMA, D., AND SIPS, H. The bittorrent p2p file-sharing system: Measurements and analysis. In *Proceedings of the IPTPS '05* (Ithaca, NY, Feb. 2005).

[20] QIAO, Y., AND BUSTAMANTE, F. Elders know best - handling churn in less structured p2p systems. In *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P '05)* (Konstanz, Germany, Aug. 2005).

[21] RHEA, S., GEELS, D., ROSCOE, T., AND KUBIATOWICZ, J. Handling churn in a dht. In *Proceedings of the USENIX '04* (Boston, MA, June 2004).

[22] ROSENBERG, J., MAHY, R., AND HUITEMA, C. Internet draft: TURN – Traversal Using Relay NAT, Mar. 2006. Work in progress.

[23] ROSENBERG, J., WEINBERGER, J., HUITEMA, C., AND MAHY, R. RFC 3489: STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs), Mar. 2003.

[24] SAROIU, S., GUMMADI, P. K., AND GRIBBLE, S. D. A measurement study of peer-to-peer file sharing systems. In *Proceedings of the Multimedia Computing and Networking 2002 (MMCN'02)* (San Jose, CA, Jan. 2002).

[25] SEN, S., AND WANG, J. Analyzing peer-to-peer traffic across large networks. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment* (Marseille, France, Nov. 2002).

[26] TAKEDA, Y. Internet draft: Symmetric NAT Traversal using STUN, June 2003. Work in progress.

[27] XU, Z., AND HU, Y. Sbarc:a supernode based peer-to-peer file sharing system. In *Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC'03)* (Antalya, Turkey, July 2003).

[28] ZHAO, B. Y., DUAN, Y., HUANG, L., JOSEPH, A. D., AND KUBIATOWICZ, J. D. Brocade: Landmark routing on overlay networks. In *Proceedings of the IPTPS '02* (Cambridge, MA, Mar. 2002).