A Measurement-based Study of the Skype Peer-to-Peer VoIP Performance

Haiyong Xie Yang Richard Yang
Computer Science Department, Yale University
{haiyong.xie,yang.r.yang}@yale.edu

ABSTRACT

It has been increasingly popular to build voice-over-IP (VoIP) applications based on peer-to-peer (P2P) networks in the Internet. However, many such VoIP applications free-ride the network bandwidth of Internet Service Providers (ISPs). Thus their success may come at a cost to ISPs, especially those on the edge of the Internet. In this paper, we study the VoIP quality of Skype, a popular P2P-based VoIP application. Specifically, using large-scale end-to-end measurements, we first conduct a systematic analysis of Skype supernode network. We then investigate the impacts of the access capacity constraint and the AS policy constraint on the VoIP quality of Skype. We show that even when free-riding is no longer possible for only 20% of supernodes that are located in stub ISPs, the overall VoIP quality of Skype degrades significantly, and a large percentage of VoIP sessions will have unacceptable quality. This result clearly demonstrates the potential danger of building VoIP applications based on P2P networks without taking into account operational models of the Internet. We also study using time diversity in traffic patterns to reduce the impacts of the preceding constraints.

1. INTRODUCTION

Recently Voice-over-IP (VoIP) applications have been proliferating across the Internet. Among them, Skype [18] is the most popular one. As of October 2006, it has been downloaded more than 398 million times and used by more than 29 million users all around the world. Within its first year of inception, Skype has served over 10 billion minutes of VoIP calls. Skype is estimated to carry 25% of annual VoIP traffic [7] and accounts for higher percentage of traffic as its user base increases and more services based on Skype become deployed.

Skype critically depends on the supernodes which form a peer-to-peer (P2P) network. Specifically, the participants in Skype are organized into two categories: standard nodes and supernodes. Any participating node initially is a standard node, and some of them will be promoted to supernodes according to a number of factors including the spare bandwidth and public reachability. Supernodes form an P2P network among themselves, and standard nodes join the network by logically connecting to a number of supernodes. Supernodes play an important role in relaying traffic for standard nodes that are behind NATs and firewalls, and thus are not able to establish connections directly. Skype can also potentially improve VoIP quality by relaying VoIP sessions through supernodes when the direct connections among standard nodes have inferior quality.

Although there are many studies on Skype, the dependency of Skype on the supernode network has not been quantified. We identify two constraints on the supernode P2P network that have major impacts on the VoIP quality of Skype.

Firstly, supernodes may have limited bandwidth available to replay VoIP sessions. We refer to this as the *access capacity constraint*. For example, the access capacity of a supernode with cable/DSL connection is at most its upstream link bandwidth. Further, as NATs and firewalls are increasingly deployed in the Internet, more standard nodes depend on supernodes to obtain VoIP services. The quality of Skype may degrade when the bandwidth demand of active sessions exceeds the available access capacity of the Skype supernode network.

Secondly, supernodes may be subject to the *AS policy constraint*. Specifically, due to Internet economics, autonomous systems (ASes) implement their own policies. A common policy is the no-valley routing policy [4], which specifies that a customer does not relay traffic for its providers. Supernodes relaying traffic may cause violation to this policy constraint [17]. Consider the stub ASes (leaf nodes in the AS topology, *e.g.*, most universities) in the Internet. When a supernode in a stub AS relays traffic for standard nodes outside the stub AS, the stub AS is relaying traffic between its providers. As a result, its financial costs might increase. Consequently, it may have incentives to block supernodes from relaying traffic. Note that this has already happened in the Internet. According to [11], some universities have recently banned Skype from their campuses, while some other universities and government agencies, instead of banning Skype completely, require that users within their networks disable supernode functionality to avoid relaying traffic (see, *e.g.*, [3]).

In this paper, we take snapshots of Skype supernodes and collect statistics. This enables us to make a first effort to evaluate the VoIP quality of Skype supernode network based on large-scale measurements. We then impose the preceding two constraints and evaluate their impacts. Our contributions can be summarized as follows:

- We quantify the VoIP quality of Skype when there is no access capacity and AS policy constraint.
- We evaluate the impacts of the access capacity constraint and the AS policy constraint. We find that they can significantly degrade the VoIP quality of Skype users. This finding points out the potential danger of building P2P applications depending on supernodes' relay capability.
- We propose to use time diversity in traffic patterns to reduce the impacts of the AS policy constraint. We also evaluate its benefits through experiments.

The rest of this paper is structured as follows. In Section 2 we review related work. In Section 3 we present a standard measure of VoIP quality. In Section 4 we describe our experimental methodology. In Section 5 we characterize the supernodes in Skype. In Section 6 we evaluate the impacts of the access capacity and AS policy constraints. In Section 7 we propose and evaluate using time diversity in traffic patterns to reduce the impacts of the AS policy constraint. We conclude and discuss future work in Section 8.

2. RELATED WORK

There have been increasing interests in studying Skype. The existing literature can be largely divided into three categories. The first category is Skype protocol analysis. In particular, Baset *et al.* [1] reverse-engineered the proprietary Skype protocol through experimental analysis. The second category is Skype measurement. In particular, Guha *et al.* [5] studied the statistics of Skype including, for example, the activity of Skype supernodes, and Skype VoIP session bandwidth consumption. The third category is new design for Skype-like VoIP applications. Ren *et al.* [14] proposed an AS-aware peer-relay protocol to improve the quality of Skype. Researchers also proposed many alternative designs (*e.g.*, overlay network [13] and path switching [20]) to improve Skype-like VoIP performance. Our work differs from the existing work. First, we make a systematic effort to characterize the VoIP quality of the Skype supernode network. Second, we identify two realistic constraints and evaluate their impacts. Last, we propose using time diversity in traffic patterns to reduce the impacts of the AS policy constraint.

3. VOIP QUALITY METRIC

At the sender's side in a VoIP session, the voice signals are sampled, digitized, and encoded using a codec (*e.g.*, G.711 and G.729). The encoded data are then assembled into packets and transmitted to the receiver. At the receiver's side, the voice data are reconstructed from the packet stream, and played out after passing through a playout buffer to smooth out the variations of network latency (*i.e.*, jitter).

The quality of a VoIP session is determined by three factors: delay, jitter, and packet losses. Delay is introduced at both the end hosts and the underlying network. In particular, the delay introduced by the end hosts includes codec delay and playout delay. The codec delay is incurred by the encoding and packetization process, and is usually fixed for a given codec. The delay induced by the underlying network consists of the transmission, propagation, and queueing delay in the network.

The quality of a VoIP session is typically measured by the ITU-T E-Model [8]. In this model, a subjective quality score, Mean Opinion Score (MOS), is defined as the metric of the perceived VoIP quality. The score ranges from 1 ("unacceptable") to 5 ("excellent"), and is computed using the nonlinear function established in [2], as shown in Equation (1), at the top of the next page, where R is referred to as the *R-factor*.

The R-factor is determined by a set of parameters. By choosing ITU-T default values, we reduce the calculation of the R-factor to be dependent on two factors I_d and I_e, measuring delay and packet losses respectively:

$$R = 94.2 - I_d - I_e. \tag{2}$$

Here I_d is the impairment caused by end-to-end delay, quantified by

$$I_d = 0.024d + 0.11(d - 177.3) \times \mathcal{H}(d - 177.3), \tag{3}$$

where d is the one-way total delay in milliseconds, and $\mathcal{H}(x)$ is the Heavyside step function taking value 0 when $x < 0$ and 1 otherwise.

The parameter I_e measures the impairment caused by packet losses, and can be quantified by

$$I_e = \gamma_1 + \gamma_2 \ln(1 + \gamma_3 p), \tag{4}$$

where p is the loss rate, and $\gamma_1, \gamma_2, \gamma_3$ are fitting parameters obtained for various codecs. Their values for G.711 and G.729 under random packet losses are shown in Table 1.

Codec	γ_1	γ_2	γ_3
G.711	0	30	15
G.729	11	40	10

Table 1: Fitting parameters for codecs.

Finally, the mapping from MOS to quality ratings is reported in [2] and summarized in Table 2. When a VoIP session receives an MOS score no less than 4.03, its quality is no worse than the current PSTN. An MOS score ranging from 3.10 to 3.60 corresponds to a low quality level, and many users are dissatisfied. Nearly all users are dissatisfied if the score is below 3.10.

R-factor	MOS	Quality of Voice Rating
$90 < R \leq 100$	4.34 – 4.50	Best
$80 < R \leq 90$	4.03 – 4.34	High
$70 < R \leq 80$	3.60 – 4.03	Medium
$60 < R \leq 70$	3.10 – 3.60	Low
$50 < R \leq 60$	2.58 – 3.10	Poor

Table 2: R-factor, MOS, and quality ratings.

4. METHODOLOGY

In order to quantify the VoIP quality of Skype, we first collect statistics of Skype supernodes. We then collect the latency and loss between any standard node and any supernode; thus, we can compute MOS for a relayed VoIP session.

We obtain a snapshot of online Skype supernodes by taking an approach inspired by [5]. Specifically, we maintain a list of uniquely identified supernodes (initially empty). We write a script to run the Skype client program to crawl across the supernode network iteratively. In each iteration, we replace the host cache of the Skype client with an unvisited supernode randomly chosen from the list, label the chosen supernode as visited, and then run the Skype client. This way, the client is forced to use the chosen supernode to obtain a fresh set of supernodes. At the end of an iteration, we kill the client forcing it to dump its current set of supernodes into its host cache. We then add the new supernodes to the list and label them as unvisited. This iterative approach may fail if the supernode network is likely to have disconnected components, or a Skype client always gets the same set of supernodes (*e.g.*, when Skype always returns a set of supernodes that are geographically close to a client). However, the likelihood is low from our experiments. In particular, we tried to test the likelihood by crawling from different starting

$$\text{MOS} = \begin{cases} 1, & R < 0 \\ 1 + 0.035R + 7 \times 10^{-6}R(R-60)(100-R), & 0 \le R \le 100 \\ 4.5, & 100 < R \end{cases} \tag{1}$$

points by running our script in a number of different ISP networks simultaneously. We find that the resulted snapshots are almost identical.

Next we construct a set of Skype standard nodes. Skype uses its own proprietary protocol with encryption; therefore, it is difficult to collect a representative set of standard nodes by tapping into Skype. We follow the approach of Ren *et al.* in [14] to use Gnutella peers to approximate Skype standard nodes. To remove redundancy, we first use BGP routing tables from RouteViews [15] and RIPE RIS [16] to map the collected IP addresses of Gnutella peers to 6860 unique IP prefixes using longest prefix match. Then for all IP addresses belonging to the same prefix, we randomly choose one. We create a standard node with this IP address. Using the same set of BGP routing tables, we also apply the methodology of [21] to look up which AS each standard node belongs to and determine whether or not an AS is a stub AS.

We use the King [6] tool to measure the round-trip time (RTT) between a standard node and a supernode. For two standard nodes a, b, we denote the RTT between a and b through a supernode s by $\mathrm{RTT}^s_{a,b}$. We compute $\mathrm{RTT}^s_{a,b} = \mathrm{RTT}_{a,s} + \mathrm{RTT}_{b,s}$, where $\mathrm{RTT}_{a,s}$ and $\mathrm{RTT}_{b,s}$ are the measured RTTs from the standard nodes a and b, respectively, to the supernode s. Note that in the computation we assume symmetric paths; thus, we approximate one-way delay using one half of the RTT. Note also that we focus on the supernode network; therefore, we do not measure the latencies of direct routes between standard node pairs.

Without a tool to measure end-to-end loss between arbitrary IP pairs, we use the loss data set available in [22], which was obtained through measurements using nodes of PlanetLab [12]. We derive the loss rates for pairs of standard node and supernode as follows. We first use MaxMind [10] to look up the geographical location for each standard node, supernode, and PlanetLab node. Then for each Skype node, we find the representative PlanetLab node that is closest to it in terms of geographical distance. Finally, we approximate the loss rate between a standard node and a supernode by the loss rate between their corresponding representative PlanetLab nodes.

5. CHARACTERIZING SKYPE SUPERNODES

We obtain 15K supernodes by running the crawler script for 2 hours with each iteration being 30 seconds. We have tried other collection parameters.[1] Our experiments show that the number of supernodes is mostly within 5% to 15K. Therefore, we use 2 hours and 30 seconds as the lengths of the experiment and an iteration, respectively. We run the script to obtain snapshots continuously from June 22, 2006 to September 22, 2006.

Figure 1 plots the results. We observe that the number of supernodes in each snapshot is relatively stable, with an average being 14960 in the first 504 snapshots and 15922 in the remaining snapshots. Note that there exists a phase transition at the 504th snapshot, where the number of supernodes increases by about 1K. An investigation shows that the transition took place on August 29, 2006, and almost all of the new supernodes were located in academic institutions.[2] The most likely reason for the phase transition is that new semester started at the end of August, and many students began to run Skype in their institutional networks. Since academic institution networks usually have high-bandwidth Internet connections, the standard nodes in such networks are likely to become supernodes.

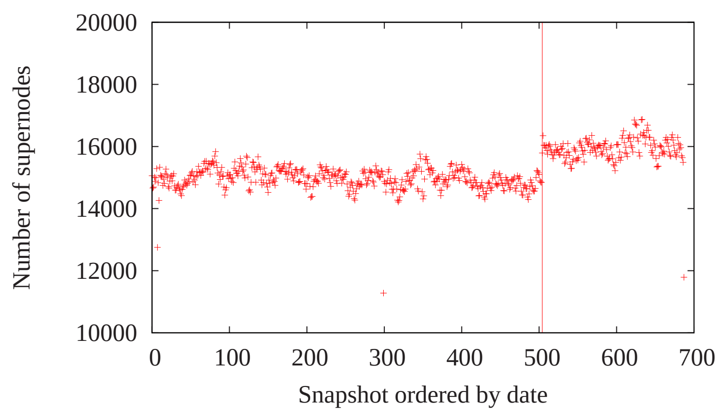

Figure 1: Number of supernodes.

Figure 2 plots the geographical locations of supernodes in the snapshot taken on August 1, 2006. About 60% of the supernodes are located in North America, and about 39% in Europe. Compared with the results obtained by Guha *et al.* [5] in 2005, our results show that North America accounts for a much higher percentage (60% vs. 25%) of supernodes. One possible reason is that starting in 2006, Skype users can make calls to landline and mobile phones for free in the US and Canada; therefore, the number of Skype users has increased dramatically in North America.

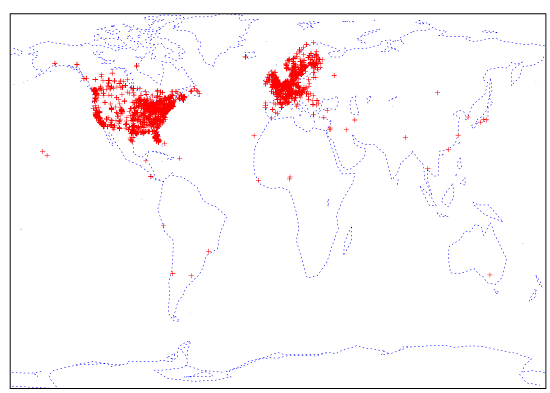

Figure 2: A map of Skype supernodes.

We look up the AS number (ASN) of each supernode and then count the number of supernodes located in each AS. We find that on average, stub ASes account for about 25%

[1] Note that the duration of the experiment should not be too long; otherwise some supernodes may be offline but are still included.

[2] A supernode is identified as new if it never appears in any of the first 504 snapshots.

of supernodes between June 22, 2006 and August 29, 2006, and about 30% afterwards.

We also use reverse DNS lookup to obtain the hostnames for all supernodes. A supernode has cable or DSL connection if its hostname has the keywords cable or DSL, and it belongs to an AS that provides Internet access services through cable or DSL. We find that 57% of the supernodes have cable or DSL connections, and the remaining 43% of supernodes are non-cable/DSL nodes. We also find that most of the cable/DSL supernodes are located in non-stub ASes, and that the non-cable/DSL supernodes are almost equally distributed in stub AS (20%) and non-stub ASes (23%) during the summer time before the phase transition on August 29, 2006 in Figure 1; however, after the new semester starts, stub ASes account for more non-cable/DSL supernodes (25% in stub ASes and 18% in non-stub ASes). Table 3 summarizes the results.

	non-Cable/DSL	Cable/DSL
stub AS	20% (25%)	5%
non-stub AS	23% (18%)	52%

Table 3: Statistics of supernodes.

Note that we do not take into account the potential BGP routing aggregation for small stub ASes. Therefore, we underestimate the number of supernodes in stub ASes since they may be classified as in non-stub ASes. This underestimation of stub-AS supernodes may lead to underestimation of the quality degradation of Skype in our evaluations.

6. QUANTIFYING SKYPE VOIP QUALITY

To characterize the VoIP quality of Skype, we consider two metrics. The first one is MOS computed using Equations (1)-(4). It measures the quality of VoIP sessions. The second metric is supernode load (*i.e.*, the number of relayed sessions carried by a supernode). A supernode has limited bandwidth to relay VoIP sessions, due to bottlenecks at the access, intradomain, or interdomain links. The amount of bandwidth that a supernode can provide is translated to the maximum number of relayed sessions it can support.

6.1 Impacts of Access Capacity Constraint

We first evaluate the impacts of the access capacity constraint. We compare the VoIP quality of Skype when (1) there is no access capacity constraint; that is, all supernodes allow unlimited load; and (2) supernodes with cable/DSL connections have the access capacity constraint; that is, each such supernode allows a maximum load of 7 sessions.[3]

We evaluate the VoIP quality as follows. We randomly choose 2 million pairs of standard nodes. For each pair, we compute the MOS of a relayed session through each supernode; we choose the maximum MOS as the VoIP quality of the session. Then we impose the access capacity constraint on supernodes with cable/DSL connections. We maintain a list of available supernodes; initially the list consists of all supernodes, each of which has the maximum load as its quota. For each supernode, we count the number of relayed sessions it has, and when its quota is used up, the supernode is removed from the list.

Figure 3 plots the results. We observe that when there is no constraint, only 12% of sessions have dissatisfied quality (*i.e.*, MOS is lower than 3.6). However, 16% and 72% of sessions have medium and high level of quality, respectively. This suggests that most Skype users can potentially receive a satisfactory level of VoIP quality. We also observe that after we impose the access capacity constraint on supernodes with cable/DSL connections, the overall quality is still close to the quality without constraint, although a large percentage (57%) of supernodes are affected. This suggests that supernodes with cable/DSL connections do not play an important role in contribution to the good VoIP quality of Skype.

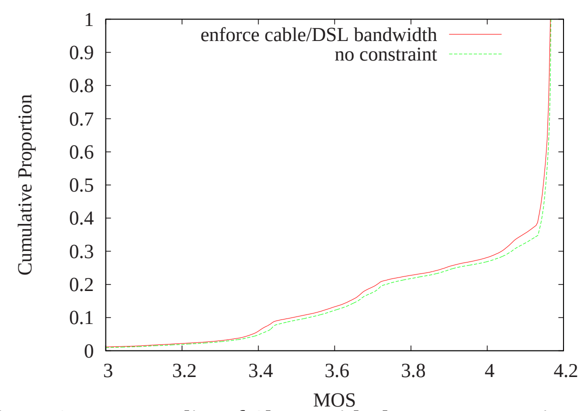

Figure 3: VoIP quality of Skype with the access capacity constraint.

6.2 Impacts of AS Policy Constraint

We next evaluate the VoIP quality of Skype when stub ASes enforce their policy by blocking supernodes from relaying VoIP sessions. There are multiple ways available to an AS to block supernodes inside its network. For instance, an AS can enforce as its policy that Skype clients should not be acting as a supernode, and provide guidelines to configure systems to prevent Skype clients from becoming supernodes (see, *e.g.* [3]). Alternatively, an AS can develop statistical methods to identify relayed VoIP sessions and block them accordingly (see, *e.g.* [19]).

We assign a maximum load to each non-cable/DSL supernode and repeat the experiments in Section 6.1. To evaluate the VoIP quality of Skype when stub ASes completely block relayed sessions, we assign 0 as the maximum load of supernodes in stub ASes; essentially those supernodes are removed from the system. However, when we assign unlimited load to the remaining supernodes, we observe that some supernodes experience extremely high load of 32K relayed sessions, which is equivalent to 2.4Gbps assuming 60Kbps as the session rate. Apparently, it is unlikely that supernodes in non-stub ASes can allocate that much bandwidth or processing capability to handle this high load. Therefore, it is more realistic to evaluate the scenario where only a limited amount of bandwidth is available to supernodes in non-stub ASes. Thus, we evaluate the impact of stub AS policy and available bandwidth in two scenarios where an amount of bandwidth equivalent to T3 and OC3, respectively, is available to supernodes in non-stub ASes.

[3]We assume that these supernodes have residential cable/DSL connections with upstream bandwidth being 384Kbps. We also assume that the rate of relayed sessions is 60Kbps, according to [5]. Thus a supernode with cable/DSL connection can support at most 7 relayed sessions.

Figure 4 plots the results when all supernodes in stub ASes are blocked. We observe that when T3 bandwidth is available to each supernode in non-stub ASes, the MOS of 25% of sessions is lower than 3.6, resulting in 108% increase in the number of unacceptable sessions, as compared to 12% in the case without constraint. The number of sessions with medium level of quality becomes 21%, close to 16% in the case without constraint. Further, the number of sessions with high quality decreases by 25% (from 72% in the case without constraint to 54%). Therefore, the stub AS policy constraint has significant impacts on the VoIP quality of Skype.

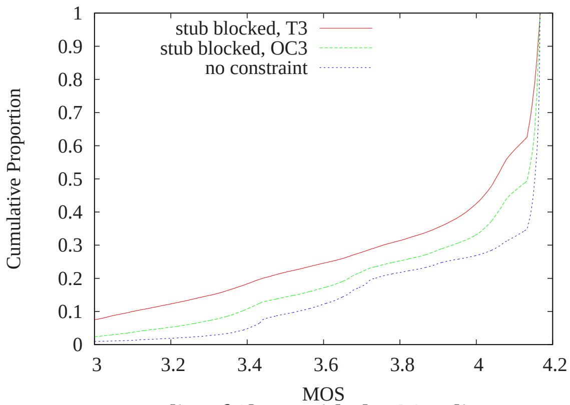

Figure 4: VoIP quality of Skype with the AS policy constraint.

We next evaluate the impacts of access capacity and AS policy constraints when the number of concurrent VoIP sessions changes. We define the relay capacity as the aggregated bandwidth available to all supernodes subject to AS policies and access link capacity. Consider the supernode network used in the preceding experiments where non-cable/DSL supernodes in stub ASes and non-stub ASes have 0 and T3 available bandwidth, respectively. The relay capacity of the supernode network is approximately 150Gbps, equivalent to 2.5 million concurrent relayed VoIP sessions. We vary the number of concurrent VoIP sessions from 0.5 to 2.5 million, and repeat the experiments in Section 6.2 to evaluate its impact on the VoIP quality of Skype .

Figure 5 plots the results. We observe that the VoIP quality degrades as the session volume gradually approaches the relay capacity. In particular, the percentages of sessions with unacceptable quality are almost the same when there are 2 and 2.5 million concurrent sessions; however, the latter case has 5% less sessions with high level of quality.

7. MITIGATING AS POLICY IMPACTS WITH TIME DIVERSITY

We have demonstrated that the access capacity and AS policy constraints have significant adverse impacts on the VoIP quality of Skype. The access capacity constraint is related to criteria of promoting standard nodes to supernodes and the growth of user base. The AS policy constraint differs from the access capacity constraints in that it is completely under the control of ISPs that provide access bandwidth to Skype nodes. Therefore, we are particularly interested in mitigating the impacts of the AS policy constraint.

Note that Internet traffic presents daily patterns (see, *e.g.*, [9]), which have peak rates in the middle of the day and low rates in mid-night. We are inspired by the observation that networks

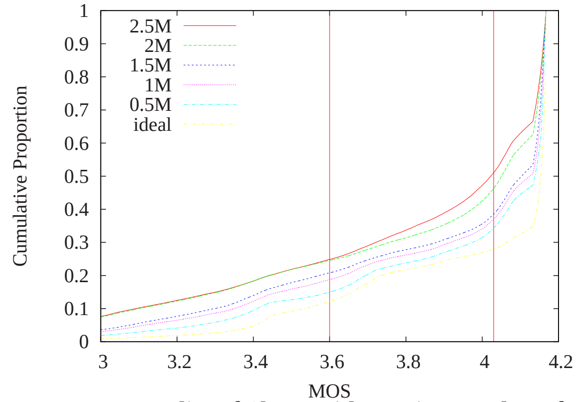

Figure 5: VoIP quality of Skype with varying number of concurrent sessions.

in different geographical locations present time diversity in their traffic patterns. We demonstrate the time diversity using traces collected from the Abilene network (AS 11537). The traces contain 5-minute traffic volumes from Nov. 1, 2003 to Dec. 31, 2003 for a number of universities and enterprises peering with Abilene. We choose the traffic volumes of AS 32, 55, 59, 87 and 237. AS 32 is located at the west coast of North America, ASes 59,87 and 237 at mid-west coast, and AS 55 at east coast. We then compute the cross correlation coefficient for each pair of ASes. Figure 6 plots the results of AS 32. We observe that AS 32 has a strong correlation with itself at a delay of approximately one day (288 5-minute intervals), suggesting that the traffic of AS 32 present daily patterns. We also observe that the traffic patterns of the other four ASes have strong correlation with AS 32 at a delay ranging from 30–40 intervals (approximately 2–3 hours).

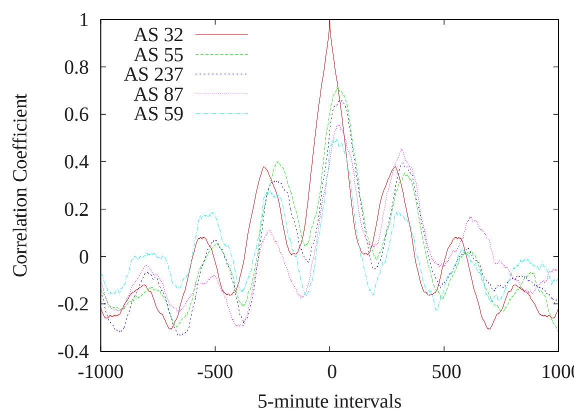

Figure 6: AS 32 has correlated traffic pattern with ASes 55, 237, 87 and 59.

The above observations suggest that when relaying VoIP sessions, supernodes can take advantage of time diversity in traffic patterns to reduce the likelihood of relaying traffic during peak time and thus incurring extra costs to the ASes. We evaluate the potential benefit of time diversity as follows. We randomly choose two consecutive time zones in North American and Europe (most supernodes are in these two regions). All of the supernodes in the chosen time zones are assumed to have peak rates in their traffic patterns. These supernodes should not be used as relay nodes. Thus we remove the supernodes that are both in stub ASes and in the chosen time zones, by setting the maximum load to 0. We also assign a maximum load of 7 sessions to cable/DSL supernodes, and assign T3 bandwidth to each of the remaining supernodes. We then repeat the experiment in the preceding section.

Figure 7 plots the results. For comparison purpose, we also include in the figure both results for the scenario where there is no constraint and the scenario where all supernodes in stub ASes are blocked. We observe that when we use time diversity in traffic patterns in choosing supernodes, the VoIP quality of Skype is approximately in the middle of the two bounding curves. An investigation shows that about 75% of supernodes in stub ASes can still be active after using time diversity in the experiment. Thus, about 15% of previously inactive supernodes become active, leading to approximately 60% increase in the relay capacity, which results in the observed VoIP quality improvement.

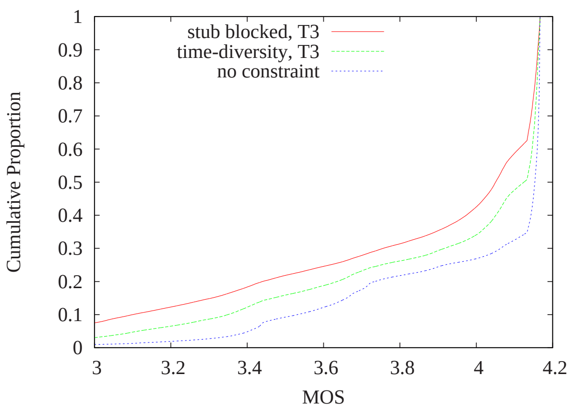

Figure 7: VoIP quality of Skype when using time diversity to choose active supernodes.

8. CONCLUSIONS AND FUTURE WORK

We have characterized the performance of Skype through measurement-based evaluations. Our results suggest that AS policies have significant impacts on the VoIP quality of Skype. Inspired by the widely used charging model and Internet traffic patterns, we have proposed using time diversity in traffic patterns to choose active supernodes, and evaluated the potential performance improvement.

There are several avenues for future work. Firstly, we adopted approximations in the measurements; that is, we use Gnutella peers to approximate Skype standard nodes, and approximate packet losses between Skype node pairs using the packet losses collected on PlanetLab testbed. It is desirable to use a representative set of real Skype standard nodes as well as real latency and loss statistics among them. Secondly, a clean protocol design is necessary for utilizing time diversity to choose supernodes. Extensive evaluations are needed to understand thoroughly how much improvement we can obtain from time diversity. Thirdly, we assume that the global optimal supernode is chosen for a relay session, which may not be achievable. The performance can be more precisely evaluated when we have better understanding of these factors.

Acknowledgments

We thank Hao Wang for valuable comments. This research is supported in part by grants from NSF.

REFERENCES

[1] S. A. Baset and H. Schulzrinne. An analysis of the Skype peer-to-peer Internet telephony protocol. In *Proceedings of IEEE INFOCOM '06*, Apr. 2006.

[2] R. G. Cole and J. H. Rosenbluth. Voice over IP performance monitoring. *SIGCOMM Comput. Commun. Rev.*, 31(2):9–24, 2001.

[3] Skype usage at Fermilab. `https://netweb.fnal.gov/skype/skype.htm`.

[4] L. Gao and J. Rexford. Stable Internet routing without global coordination. In *Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Santa Clara, CA, June 2000.

[5] S. Guha, N. Daswani, and R. Jain. An experimental study of the Skype peer-to-peer VoIP system. In *Proceedings of 5th International Workshop on Peer-to-Peer Systems (IPTPS)*, Santa Barbara, CA, February 2006.

[6] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating latency between arbitrary Internet end hosts. In *Proceedings of Internet Measurement Workshop*, Marseille, France, Nov. 2002.

[7] S. Henshall. VoIP traffic and Skype. `http://www.skypejournal.com/blog/archives/2005/02/thefeature_call.php`.

[8] ITU-T Recommendataion G.107. The E-Model, a computational model for use in transmission planning, Mar. 2003.

[9] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, New York, NY, June 2004.

[10] Maxmind. `http://www.maxmind.com`.

[11] R. Paul. More universities banning Skype. `http://arstechnica.com/news.ars/post/20060924-7814.html`, Sept. 2006.

[12] Planetlab. `http://www.planet-lab.org`.

[13] R. K. Rajendran, S. Ganguly, R. Izmailov, and D. Rubenstein. Performance optimization of VoIP using an overlay network. In *Proceedings of IEEE INFOCOM '06*, Apr. 2006.

[14] S. Ren, L. Guo, and X. Zhang. ASAP: an AS-aware peer-relay protocol for high quality VoIP. In *Proceedings of IEEE ICDCS '06*, Lisboa, Portugal, July 2006.

[15] Routeviews. `http://www.routeviews.org`.

[16] S3: Scalable sensing service. `http://networking.hpl.hp.com/s-cube`.

[17] S. Seetharaman and M. Ammar. Characterizing and mitigating inter-domain policy violations in overlay routes. 2006.

[18] Skype. `http://www.skype.com`.

[19] K. Suh, D. Figueiredo, J. F. Kurose, and D. Towsley. Characterizing and detecting Skype-relayed traffic. In *Proceedings of IEEE INFOCOM '06*, Apr. 2006.

[20] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang. Improving VoIP quality through path switching. In *Proceedings of IEEE INFOCOM '05*, Apr. 2005.

[21] H. Wang, H. Xie, Y. R. Yang, L. E. Li, Y. Liu, and A. Silberschatz. Stable egress route selection for interdomain traffic engineering: Model and analysis. In *Proceedings of the 13th International Conference on Network Protocols (ICNP)*, Oct. 2005.

[22] P. Yalagandula, P. Sharma, S. Banerjee, S.-J. Lee, and S. Basu. S3: A scalable sensing service for monitoring large networked systems. In *Proceedings of ACM SIGCOMM Workshop on Internet Network Measurement (INM)*, Pisa, Italy, Sept. 2006.