Skype Relay Calls: Measurements and Experiments

Wookyun Kho, Salman Abdul Baset, Henning Schulzrinne
Department of Computer Science
Columbia University
wk2153@columbia.edu, {salman,hgs}@cs.columbia.edu

***Abstract*—Skype, a popular peer-to-peer VoIP application, works around NAT and firewall issues by routing calls through the machine of another Skype user with unrestricted connectivity to the Internet. We describe an experimental study of Skype video and voice relay calls conducted over a three-month period, where approximately 18 thousand successful calls were made and over nine thousand Skype relay nodes were found. We have determined that the relay call success rate depends on the network conditions, the presence of a host cache, and caching of the callee's reachable address. From our experiments, we have found that Skype relay selection mechanism can be further improved, especially when both caller and callee are behind a NAT.**

***Index Terms*—Skype, relay.**

I. INTRODUCTION

Skype is a popular peer-to-peer Internet telephony application developed by the founders of the well-known file sharing application KaZaA [1]. One of the mechanisms Skype uses to address network connectivity issues in the presence of NATs and firewalls is by relaying calls through machines of other Skype users with unrestricted connectivity. This mechanism, however, is relatively little known. The Skype users allow the relaying of calls from their machines by agreeing to the Skype end user license agreement (EULA) [2]. The Skype guide for network administrators suggests that relaying of voice and video calls can consume network bandwidth up to 4 KB/s and 10 KB/s for voice and video calls, respectively [3]. Skype does not provide users with any mechanism to disable the forced use of this network bandwidth.

In this paper, we attempt to gain insights into the Skype relay selection algorithm by determining the characteristics of relay calls and machines relaying Skype calls in a black-box manner. During a four month period, we made over 38 thousand Skype calls under three different network setups. The experimental setup forced Skype to use a relay. Our results show that 18 thousand calls were successful and were routed through 9,584 unique relays. 89.3% of these calls were routed through a relay in the US. The call success rate depends on the network setup. For 17.5% of successful Skype relay calls, Skype used a different relay from caller to callee and from a callee to caller.

Our work makes two contributions. First, we have determined that although Skype exhibits geographical locality in relay selection, the median one way call latency is not insignificant, suggesting that further tuning of Skype relay selection mechanism could be beneficial. Second, we discuss factors impacting the success rate of Skype calls. This result can be used for improved future protocol design.

The remainder of the paper is organized as follows. Section II describes the experimental setup and Section III-A discusses factors affecting success rate of relay calls. In Section III-B, we discuss the geographical, ISP, and autonomous system distribution of relay nodes and relay calls and online presence of relay nodes. Finally, we conclude and discuss future work in Section IV.

II. EXPERIMENTAL SETUP

We give a brief description of the Skype software and network architecture and then describe the experimental setup. For a detailed description of Skype architecture, we refer the reader to [4].

Skype software, thereafter referred to as Skype client (SC), maintains a list of other Skype peers called a host-cache (HC) [4]. This list is empty when a SC is run for the first time after installation, is built during the lifetime of a Skype client, and survives the SC termination. During subsequent runs, a SC can contact nodes in the host cache (HC). Additionally, a SC has a built-in list of about seven default Skype nodes, called bootstrap nodes, which it uses to join the Skype overlay for the first time after installation.

Upon a successful join, a SC establishes a TCP connection with the machine of another Skype user or node, referred to as a super node (SN) in this paper. There are two types of nodes in the Skype overlay; super node (SN) and ordinary node. Both SN and ordinary node run the same Skype software. SNs are responsible for detecting online SCs, and transmitting signaling messages between SCs [3]. To establish a call, a SC searches for the callee machine and upon successful contact, either directly exchanges media traffic with the callee machine or through another Skype node. We refer to the node relaying a Skype call as a relay node (RN) and the call being routed as a relay call (RC). A RN is

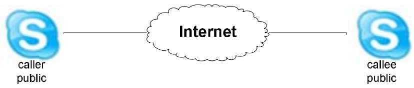

Fig. 1. Unrestricted

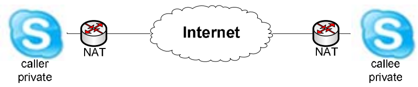

Fig. 2. NAT: caller and callee behind address and port dependent NAT.

Fig. 3. Direct-blocked: packets between caller and callee are dropped.

a SN that has sufficient bandwidth to relay a voice or a video call.

A key aspect of Skype's robust connectivity is its ability to traverse NAT and firewalls by routing a video or a voice call through one or more Skype relays. To study Skype's robust connectivity, we devised an experiment in which a call was established between two Skype clients running on machines in our lab at Columbia University. The RTT between two machines was less than one ms. The network conditions between the caller and callee machines were configured so that they were forced to use a RN. Specifically, the experiment was performed under three different network setups: unrestricted connectivity (Figure 1), caller and callee behind different address and port dependent NAT[1] (Figure 2), and direct-blocked setup, in which caller and callee, having otherwise unrestricted connectivity, could not send packets to each other (Figure 3). The setups are referred to as unrestricted, NAT, and direct-blocked in the paper. For the NAT and direct-blocked setup, the experiment was performed with and without deleting the host cache. The experiments were performed from March 27 to July 3, 2007. We used Skype v3.2 for Windows XP in our experiments.

We wrote a script using AutoIt [5] that uses the Skype API [6] to automate the establishment of Skype calls, checking of call status, and gathering and collection of relay data at caller and callee. The duration of a call was configured to be one minute and during this time, the script gathers the number of packets sent by the caller and callee machines to unique IP address and port number pairs using WinDump [7]. With a packetization interval of 30 ms (as used by Skype), the caller machine should send approximately 1,800 packets to the RN and vice versa. After terminating each call, we parse the WinDump data and label the IP address and port number that most frequently appeared in the WinDump data as a relay node, i.e., the IP address and port number with which a caller or callee exchanged the maximum number of packets.

[1] Consider an internal IP address and port (X:x) that initiates a connection to an external (Y1:y1) tuple. Let the mapping allocated by the NAT for this connection be (X1':x1'). Shortly thereafter, the endpoint initiates a connection from the same (X:x) to an external address (Y2:y2) and gets the mapping (X2':x2') on the NAT. If (X1':x1') equals (X2':x2') only when (Y2:y2) equals (Y1:y1), then such a behavior of the NAT is defined as "Address and Port Dependent Mapping" behavior.

We have determined the Skype success rate for the above three network setups. We also determined how the presence of the host cache (a list of online Skype nodes) affects Skype relay success call rate. Throughout the experiment, we collect IP address and port number of Skype relay nodes and track their online status by sending a specially crafted Skype message to them. We have also determined the geographical, ISP, and autonomous-system (AS) distribution of RNs and RCs using MaxMind [8] and a AS number lookup utility [9] and calculated the RTT for RNs. We use this data to gain insights into the efficacy of Skype relay selection algorithm.

The closest study to ours is an experimental study of Skype SNs conducted by Guha et al. in 2005 [10]. Guha monitored the HC of a SC and tracked the population and presence of SNs. It was not certain if these SNs were also acting as relay nodes. We, however, discovered the RNs by establishing and monitoring calls that were relayed through these Skype nodes. Further, we study the factors impacting Skype RCs success rate, the presence information of RNs and their geographical distribution.

III. RESULTS AND DISCUSSION

In this section, we discuss the results of our experiments: factors impacting sucess rate of Skype RCs and characterization of Skype RCs and RNs.

A. *FACTORS IMPACTING SKYPE RELAY CALLS*

We established over 38,000 calls over a three month period for the three different network setups described in Section II. For 37,761 calls, the network was configured such that Skype was forced to select a relay. Out of these 37,761 calls, approximately 18,000 were successful and the success percentage depended on the network setup. Seventeen percent (3,146) of successful RCs used a different relay from caller to callee and from callee to caller. There was almost no difference between the call success rates of video and voice calls. Table I shows the detailed call statistics for the three network setups.

We pose the following two questions: first, how does network connectivity affects Skype RC success rate and second, whether retention of HC entries from previous runs impacts the call establishment. Observe from Table I that the call success rate for the direct-blocked setup that retains HC after a call trial is lower than the unrestricted and NAT setups. We have observed that when a SC comes online, it sends a notification to the Skype users

TABLE I
STATISTICS FOR SKYPE RELAY CALLS

	Experimental setups					Aggregated
	Unrestricted	NAT		Direct-blocked		
		HC deleted	HC not deleted	HC deleted	HC not deleted	
Call trials	867	4,649	5,658	15,774	11,680	38,628
Successful calls	867	339	5,567	2,586	8,597	17,962
Success rate (%)	100%	7.3%	98.4%	16.4%	73.6%	46.5%
Relays found	N/A	379	2,843	2,718	4,317	9,584
% of calls through US relays	N/A	74%	81.2	91.6%	92.4%	88.2%
% of succ. calls w/two relays	N/A	28.6%	17.7%	31.2%	14.6%	17.5%
One-way call latency (ms)	N/A	29.1	95.7	8.8	13.3	43.6

in its buddy list. Since the direct-blocked setup will drop any packets between caller and callee, this notification is dropped. When a caller initiates a call, it must search for the callee SC in the Skype network. After finding the callee IP address, the caller sends signaling traffic to the call through another Skype user. Further, the caller and callee must find a Skype node for relaying media traffic. However, unlike the NAT setup, the relay search is initiated only during call establishment, as both caller and callee had earlier assumed that they had unrestricted connectivity.

When a SC comes online, it establishes a TCP connection to a SN and publishes its SN information in the Skype network. In our experiments, we found that for some of the failed calls in the direct-blocked setup, the callee search reached the callee SN from the previous trials. Our conjecture is that this is likely due to the caching of the callee reachable address in SNs. When the SC is behind a NAT, it publishes its SN information more frequently than when it is directly connected. However, this does not happen for the direct-blocked case, as the SC is fooled into thinking that it has unrestricted connectivity. Thus, for the direct-blocked setup, less frequent publishing of callee SN and an attempt to find a signaling and media relay at the time of call establishment result in relatively more call failures than the NAT setup.

To address the second question whether retention of HC from previous runs impacts the call success rate, we deleted the HC after every trial for the three network setups and measured the call success rate. In the absence of HC, a SC uses bootstrap nodes to join the Skype network. For experiments in the NAT and direct-blocked setup where HC was deleted after every call trial, we observed a strange phenomena with the call success rate. Initially, the call success rate was 100%, but as time passed, there was a drastic drop in the success rate, and ultimately all calls started to fail. We experimented with different caller and callee identifiers and observed the same phenomena. Interestingly, we did not observe this phenomena for the unrestricted setup. We offer a possible explanation below.

It has been previously studied that the intermediate SNs contacted by the caller during the callee search process cache the results [4]. Since the HC is deleted after every call trial, a caller SC must contact the same set of bootstrap peers during login. It takes time, on the order of minutes, for Skype to build a new HC from scratch. Moreover, a Skype callee behind a NAT and direct-blocked setup must publish a reachable IP address, obtained through STUN [11] and TURN [12] like mechanisms, in the Skype network. It is likely that for a new call trial, callee's reachable address is not updated in the SNs cache and the caller SC always reaches the SNs caching the old reachable address of callee.

Thus, we attribute the Skype RC failures to (1) the stale information about the IP address and port number of the callee and its SN in the cache of other Skype nodes and (2) the inability of Skype to find a relay at the time of call establishment.

B. CHARACTERIZATIONS OF SKYPE RELAY NODES

In this section, we classify the IP address of 9,584 unique relay nodes discovered in Section III-A according to their geographical, ISP, and autonomous-system (AS) level distribution and determine their RTT and uptime. We also present a geographical distribution of relay calls and comment on the efficacy of the Skype relay selection algorithm.

a) Relay distribution: We used MaxMind [8] to determine the geographical distribution of relay node IP addresses and *nslookup* for reverse DNS lookup. Out of 9,584 RNs, 89.36% were located in North America and 11.5% were in Europe. The US-based RNs comprised 82.6% (7,920) of the total relay nodes and 21.2% of the total RNs were in New York state. We also classify the RNs using their domain names obtained from reverse DNS lookup. Table III shows five organizations with a .edu, .net, and .com suffix having the most number of unique RNs, the percentage of calls routed, and median RTT. An interesting aspect is that 22.4% (2150) of RNs had a .edu suffix, which indicates their affiliation with universities. This is a significant percentage and without attempting to answer we pose the question whether the

TABLE II
TOP TEN AS WITH THE LARGEST NUMBER OF UNIQUE RNS

Organization	AS #	% RNs	% succ. calls	Median RTT (ms)	Organization	AS #	% RNs	% succ. calls	Median RTT (ms)
Cable Vision	6128	9.2	6.1	15.3	AOL	1668	3.9	3.4	95.6
RR-NYC	12271	7.3	5.9	16.4	Comcast	33287	3.7	2.8	30.1
Rogers	812	4.5	2.5	52.1	Columbia Univ.	14	3.1	17.4	0.28
SBC	7132	4.4	2.5	46.6	Cox	22773	2.9	1.7	65.6
Comcast	7015	4.4	3.6	16.2	Comcast	33657	2.6	2.0	34.5

TABLE III
TOP FIVE ORGANIZATIONS WITH RELAY NODES

Organization	% of RNs	% of calls	Median Uptime (hours)	Median RTT (ms)
.edu				
Columbia	2.7	15.1	3.3	0.3
Yale	2.1	5.1	3.9	9.8
Georgia Tech.	1.3	5.2	4.1	30.1
MIT	1.1	0.9	6.2	0.9
NYU	1.1	2.3	5.9	2.27
.com				
RR	9.8	7.2	4.6	15.5
AOL	4.0	4.5	4.2	95.6
Mindspring	2.6	1.6	3.4	58.6
Rogers	2.5	1.5	0.2	34.9
Charter	1.6	1.0	3.5	32.8
.net				
Comcast	18.1	12.3	3.9	29.1
Optimum Online	9.2	6.1	3.8	14.9
Cox	2.6	1.4	2.2	81.8
SBC	2.5	1.5	6.7	38.3
Ameritech	1.5	0.7	6.2	40.7

Skype network is being indirectly sustained by high-bandwidth and non-NAT networks of universities?

Besides geographical and domain name classification of 9,584 RN IP addresses, we also used the aslookup [9] utility to discover the AS number of RN IP addresses. The tool contacts one or more *whois* servers to obtain the AS number of an IP address. The aslookup was successfully able to retrieve the AS numbers of 7,954 (83%) IP addresses that belong to 336 unique autonomous systems. The statistics are summarized in Table II and Figure 4. Out of 336 ASes, the top ten AS had 46% of RN IP addresses while the top twenty percent had 90% (8,627) of RN IP addresses. Recall that New York state had 21% of the total RN IP addresses and observe that a New York city ISP (RR-NYC) and Columbia University have 10.4% of the total RN IP addresses. This result gives an indication that a SC attempts to select a RN that is geographically closer to caller and callee.

Observe from Table II that a higher percentage of RNs belonging to one organization does not imply that more relay calls are routed through hosts in that organization. Cable Vision, an ISP, has 9.2% of total RNs but they only relay 6.1% of the calls. Columbia University has 3.1% of the total RNs but they relay 17.4% of the total RNs calls. This result gives another indication that Skype is attempting to optimize the RN selection.

Figure 5 shows the number of unique RNs found for the complete duration of the experiment, i.e., from March 27 to July 3, 2007, and is a linearly increasing line. This result shows that the total population of RN candidates is much larger.

Guha et al. [10] had mined the HC of Skype to obtain a list of 2,081 Skype SNs. We compared our RN list to Guha's SN list and found that the two lists had only six IP addresses in common. One reason for such a minimal overlap is a time gap of more than one year between the two studies. The other reason is that Guha mined SN list from Skype client's HC. The SN list from HC does not necessarily imply that those nodes will be selected as a relay. On the other hand, we established Skype calls and experimentally obtained the RNs.

b) RTT: To gain more insights in the efficacy of the Skype relay node selection algorithm, we also analyzed the RTT of relay nodes. We measured the RTT of RNs by sending ping messages to each RNs. Figure 6 shows the CDF of the RTT for RNs. The average and median RTT for RNs were 52.2 ms and 43.6 ms, respectively. Since both caller and callee machines were located in our lab and have the same RTT to the RN, one-way median network latency for a call is 43.6 ms. For NAT setup, one way median network latency is 95.6 ms and for the direct-blocked setup, it is 13.3 ms. The NAT setup is likely to be a common case on the Internet and a median latency of 95.6 ms between two machines behind NATs, having otherwise a RTT of less than one ms, is not insignificant. Moreover, one-way median latency for NAT setup is significantly higher than the direct-blocked setup. One could argue that for NAT setup, a SC will choose a relay at the time of login, where as in direct-blocked setup a SC chooses a relay at the time of call establishment. Thus, a SC would have more time to optimize relay selection for NAT setup, and consequently, one-way call latency should be lower for NAT setup. However, we did not observe low latencies for calls in NAT setup and the cause remains unclear. We suspect that Skype might try to balance the load on each relay node, and this mechanism causes Skype to not always choose the closest relay node.

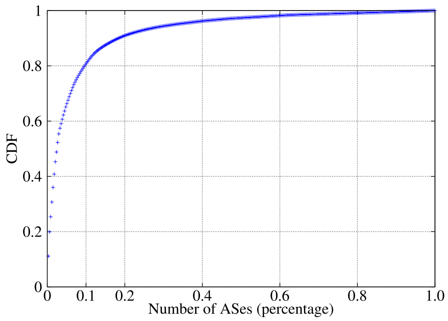

Fig. 4. Number of relays nodes (RN) per AS

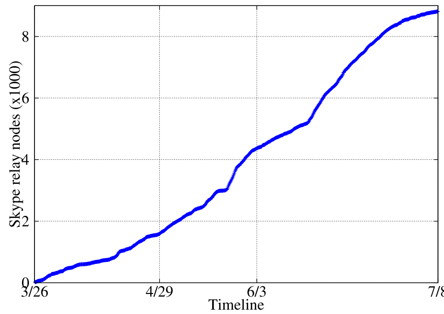

Fig. 5. Number of unique RNs found

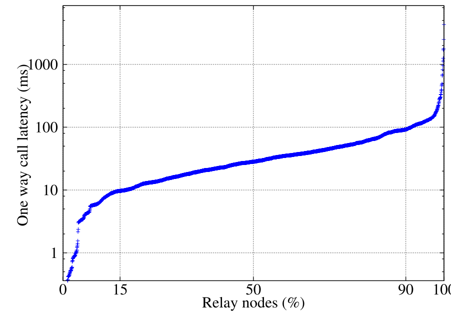

Fig. 6. CDF of RNs RTT

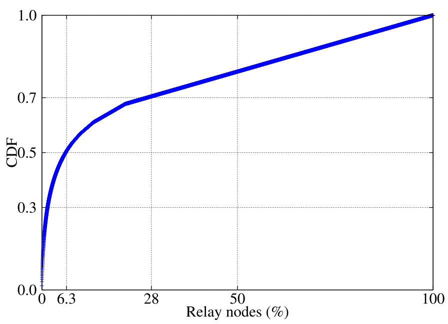

Fig. 7. CDF of relay calls (RCs) per unique RN

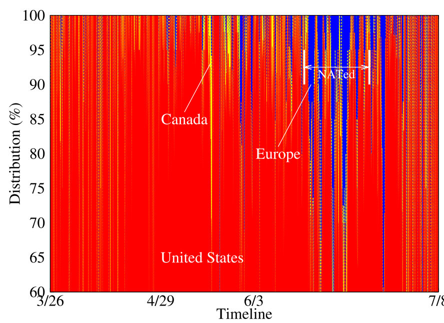

Fig. 8. Geographical distribution of RCs

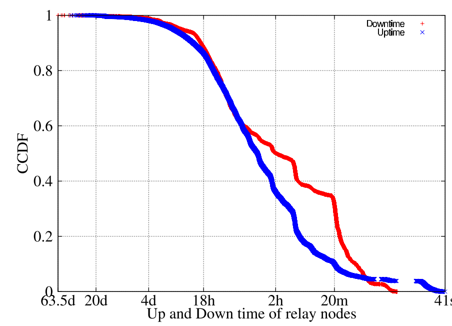

Fig. 9. CCDF of RNs uptime & downtime

c) Call distribution over relays: We characterize the distribution of relay calls per RN and determine whether there is any temporal locality in the selection of a RN, i.e., for how many subsequent calls does a SC use the same relay.

As listed in Table I, there are 9,584 RNs that relay 17,095 calls, so each RN relays approximately two calls. Figure 7 and Figure 8 show the CDF of RCs per unique RN and geographical distribution of RCs. Approximately, 6.3% (603) of the 9,584 nodes belonging to 74 autonomous systems relayed 50% of the total calls. Clearly, a significant portion of relay calls are routed through a small subset of RNs and this refutes any conjectures about Skype relay algorithm selecting a random RN. The result will be more clear when we discuss the uptime of RNs.

As listed in Table I, 88.2% of successful RCs were routed through relays in US, and 3.7% and 6.93% through relays in Canada and Europe. Observe that for the NAT setup, only 81.2% of the calls were routed through US-based relays as compared to 92.4% for direct-blocked setup which indicates that there is room for improvement in Skype's relay selection algorithm. This is also highlighted in Figure 8 which shows an increase in the use of European relay nodes during the month of June when NAT-setup experiments were performed. It is unclear why Skype uses relatively more non-US relays when both caller and callee are behind address and port dependent NAT.

Figure 10 shows the timeline of the number of times the top five RNs were selected as a relay and their uptime. From our results we noted that the maximum number of times a relay node was selected in consecutive trials was seven. These results indicate that Skype is caching the RN lists. Surprisingly, the RN with the largest uptime of 42.5 days was only once selected as a relay during the course of our experiments. It is possible that this node was selected as a relay for calls placed by other users. However, it is difficult to obtain this information due to the closed nature of the Skype.

d) Uptime of relays: We tracked the presence of RNs discovered in the three experimental setups. Every few minutes, we sent a specially crafted Skype message, thereafter called Skype-ping, to RN IP address and port number to which a Skype client across different versions is known to respond. If there was no reply, we consider the Skype node to be offline. We conducted this experiment from March 27 to July 3, 2007.

Our result shows that the uptime distribution of Skype relay nodes follows a diurnal pattern. This result is quite similar to the uptime distribution of supernodes found by Guha [10]. The likely reason as also mentioned by Guha is that there are more users in the Skype network during the day than during the night. Figure 9 shows the CDF of uptime of RNs. A relay node can be online at different times during the study period. The CDF plot shows all the uptimes of a unique RN and not the cumulative uptime. Over the course of our RN study, the maximum and median uptime of RNs was 42.5 days and 3.5 hours, respectively. Similarly, the maximum and median downtime was 63.5 days and 2 hours, respectively. The median lifetime cycle length of a RN is 5.5 hours and uptime probability is 63.6%. The median uptime for RNs in .edu, .net, and .com domains was 4.5 hours, 3.7 hours, and 2.5 hours, respectively. Note the relatively longer uptime for RNs with a .edu suffix and this again raises the question whether the university networks are indirectly supporting the Skype peer-to-peer network.

The median uptime of SNs reported by Guha was 5.5 hours. Perhaps the difference between our RN median uptime and the one reported by Guha is the duration of uptime study. We conducted our uptime study over a five month period and discovered 9,584 unique RNs whereas Guha conducted his study over a period of one month for 2,081 unique SNs.

An aspect of Skype which can impact the uptime statistics is that Skype does not have a standardized listening port. SC picks a random port upon installation and additionally, listens for incoming requests at port 80 and 443, the HTTP and HTTPS ports, respectively. There is no guarantee that a SC will always use the same random port picked at installation. Therefore, it is possible that a Skype-ping message may actually never be received by a SC although it may be online. Thus, the uptime results may not accurately represent the uptime of RNs. We tried sending Skype-ping message to ports 80 and 443, but SCs did not respond.

e) Summary of Results: 82.6% of the RNs were located in US and 6.3% of the total RNs relayed 50% of the calls. This reuse of relay nodes suggests that Skype caches RN information. The median one-way network latency was 43.6 ms and is dependent on the experimental setup. Our results indicate that the mechanisms for relay selection can be improved, especially when both caller and callee are behind a NAT. The RC failures in the absence of HC can be reduced by quickly updating the SN cache for callee's reachable address. Further, since 50% (~9,000) of the calls are relayed by nodes belonging to 74 autonomous systems, it is possible to use the AS number as an approximate metric to search for a relay closer to caller or callee. However, this depends on how large the AS is.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we have experimentally determined various characteristics of Skype RCs. The success rate of Skype RCs depends on the network conditions, the presence of a host cache, and caching of callee reachable address by SNs.

We observed that for 17.5% of the successful calls, Skype used a different relay from caller to callee and vice versa. Further, approximately 80% of the calls were routed through US-based relays showing geographical locality. However, the median one-way network latency for RCs in a NAT setup is 95.7 ms. Since NAT setup is likely to be the common case on the Internet, this latency is not insignificant, showing that Skype relay selection mechanism can be improved by using techniques like network coordinates [13] and ASAP [14]. We also observed that the uptime of RNs follows a diurnal pattern.

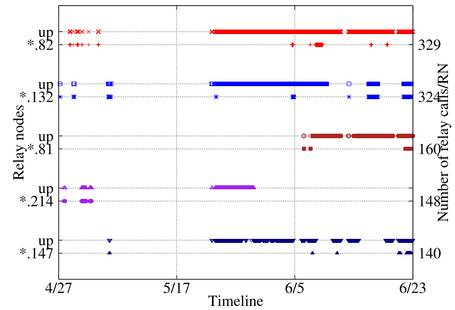

Fig. 10. Relay selection and uptime duration of top five RNs

This paper is only a first step in understanding the nature of Skype relay calls. To gain further insights into the Skype relay selection algorithm, one can gather the relay node data for voice and video calls at different locations around the world, and under different network conditions such as limited bandwidth, NAT, and firewalls. While a reverse engineering of Skype executable will obviously reveal the Skype relay selection mechanism, only such a distributed study can give key insights into the operation of the Skype overlay.

REFERENCES

[1] KaZaA: http://www.kazaa.com.
[2] Skype End User License Agreement: http://www.skype.com/company/legal/eula/.
[3] Skype Guide for Network Administrators: http://tinyurl.com/24nzz8.
[4] S. A. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," in *INFOCOM*, Barcelona, Spain, April 2006.
[5] AutoIt: http://www.autoitscript.com/autoit3.
[6] Skype API: https://developer.skype.com.
[7] Windump: http://www.winpcap.org/windump.
[8] MaxMind: http://www.maxmind.com.
[9] AS number lookup utility: http://www.bugest.net/software/aslookup/index-e.html.
[10] S. Guha, N. Daswani, and R. Jain, "An experimental study of the Skype peer-to-peer VoIP system," in *IPTPS*, Santa Barbara, California, USA, February 2006.
[11] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "STUN - Session Traversal Utilities for NAT," Internet-draft, IETF, February 2008, Work in progress.
[12] J. Rosenberg, R. Mahy, and P. Matthews, "TURN - Traversal Using Relays around NAT," Internet-draft, IETF, February 2008, Work in progress.
[13] F. Dabek, R. Cox, M. F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network coordinate system," in *SIGCOMM*, Portland, Oregon, USA, August 2004, pp. 15–26.
[14] S. R. Lei, L. Guo, and X. Zhang, "ASAP: an AS-aware peer-relay protocol for high quality VoIP," in *ICDCS*, Lisboa, Portugal, July 2006.