Computer Networks xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Understanding Skype signaling

Dario Rossi [a,*], Marco Mellia [b], Michela Meo [b]

[a] TELECOM ParisTech, INFRES, 37/39 Rue Dareau, 75014 Paris, France
[b] Politecnico di Torino, Italy

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Internet traffic measurements
Skype

ABSTRACT

Skype is without a doubt today's VoIP application of choice. Its amazing success has drawn the attention of both telecom operators and the research community. There is a great interest in characterizing Skype's traffic, understanding its internal mechanisms, and learning about its users' behavior. One of the most interesting characteristics of Skype is that it relies on a P2P infrastructure for the exchange of signaling information that is distributed between peers.

Leveraging the use of an accurate Skype classification engine that we recently designed, we now report the results of our experimental study of Skype signaling based on extensive passive measurements collected from our campus LAN. We avoid the need to reverse-engineer the Skype protocol, and we instead adopt a black-box approach. We focus on signaling traffic in order to infer certain interesting properties regarding overlay maintenance and, possibly, the overlay structure as well.

Our results show that, even though the signaling bandwidth used by normal peers is exiguous, it may nonetheless account for a significant portion of the total traffic generated by a single Skype client. Skype performs peer discovery and refresh by using a large number of single packet probes. This may be as effective for the purpose of overlay maintenance as it is costly, at least from the viewpoint of layer-4 network devices. At the same time, single-packet probes account for only a minor fraction of all signaling traffic: therefore, we wish to explore more deeply the traffic that is exchanged among the more stable peers, in an attempt to learn how the peer selection mechanism actually operates.

Measurements were collected during April and August 2007. In particular, during the second month of sampling, the Skype network suffered a worldwide service outage. We compare the results collected during the two time periods, and we demonstrate the striking impacts on the signaling network as a result of the outage.

1. Introduction

In the last few years, VoIP telephony has gained tremendous popularity, with an increasing number of operators offering VoIP-based phone services. Skype [1] is the most remarkable example of this new phenomenon: developed in 2002 by the creators of KaZaa, it recently reached over 170 million users, and it accounts for more than 4.4% of total VoIP traffic [2]. As the most popular and successful VoIP application, Skype has attracted the attention of the research community and of multiple telecom operators as well.

One of the most interesting features of Skype is that it relies on a P2P infrastructure to exchange signaling information in a distributed fashion, with a twofold benefit of making the system both highly scalable and robust. The natural question is as follows: how costly is the P2P overlay maintenance, and how great is the signaling overhead needed to exchange information about users' reachability in a distributed fashion? The objective of this paper is to provide answers to these questions. To the best of our

[*] Corresponding author. Tel.: +33 145817563.
E-mail address: dario.rossi@enst.fr (D. Rossi).

1389-1286/$ - see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.comnet.2008.10.013

knowledge, this work is the first investigation of Skype signaling traffic: indeed, the study of Skype is made very complex by the fact that protocols are proprietary, that the system makes extensive use of cryptography, obfuscation and anti reverse-engineering techniques [4], and that it uses a number of techniques to circumvent NAT and firewall limitations [3].

By building on our previous work in which we devised a methodology that successfully tackles the problem of Skype traffic identification [8], we aim to contribute to the understanding of Skype's operation. We follow an identical methodology to that in our previous research, which relies on protocol ignorance. This is because Skype's proprietary design and its adoption of cryptography mechanisms makes it almost impossible to decode. Consequently, we did not perform any reverse engineering of the protocol. We propose a simple classification of Skype signaling traffic, isolating different components of signaling activity that pertain to different tasks (such as network discovery, contact list refresh and overlay maintenance). Our results show that, despite the fact that the signaling bandwidth used by normal peers is exiguous, it may nonetheless constitute a very significant portion of the total traffic generated by a Skype client. Also, we observe that Skype performs peer discovery and refresh using a large number of single packet probes. At the same time, the bulk of the signaling traffic is carried by a relatively small number of longer flows, exchanged with more stable contacts. We therefore explored the traffic exchanged among such peers, in an attempt to understand how the peer selection mechanism works: in the following, we will show that the selection is driven by both network latency and user preferences.

Our study uses measurements collected during April and August 2007. During the second month of sampling, the Skype network suffered a worldwide service outage that lasted two days. We compare the results collected during the two time periods, and we report on the striking impacts on the signaling network as a result of the outage.

Despite the attention of the research community and telecom operators, [3–9], all previous papers but [3] have completely ignored Skype signaling traffic. [3] focuses on the login phase, and on how Skype traverses NAT and firewalls. By contrast, our aim is to provide quantitative insights into the volume and characterization of Skype signaling traffic. Moreover, we evaluate the cost of typical P2P mechanisms, such as network discovery, overlay maintenance, and information diffusion.

2. Skype Overview

In this section, we overview Skype behavior.

Skype offers end users several (free) services: (i) voice communication, (ii) video communication, (iii) file transfer and (iv) chat services. Communication between users is established using a traditional end-to-end IP paradigm, but Skype can also route calls through a SuperNode to ease the traversal of symmetric NATs and firewalls. Voice calls can also be directed toward the PSTN using Skypein/Skypeout services, in which case a fee is applied.

The main difference between most VoIP services and Skype is that the latter operates on a P2P model, except for user authentication, which is performed under a classical client–server architecture by means of public key mechanisms. After the user (and the client) has been authenticated, all further signaling is performed on the P2P network, so that Skype's user information (e.g., contact lists, status, and preferences) is entirely decentralized and distributed among nodes. This allows the service to scale very readily, thereby avoiding a centralized (and expensive) infrastructure. Peers in the P2P architecture can be either normal nodes or SuperNodes. The latter are selected among peers with large computational power and good connectivity (considering bandwidth, uptime and absence of firewalls). They take part in a decentralized information distribution system that is based on a DHT.

From a protocol perspective, Skype uses a proprietary solution that is difficult to reverse engineer due to its extensive use of both cryptography and obfuscation techniques [4,3]. Though Skype may rely on either TCP or UDP at the transport layer, both signaling and communication data are preferentially carried over UDP. A single random port is selected during application installation, and it is never changed (unless forced by the user). When a UDP communication is impossible, Skype reverts to TCP, listening to the same random port, and to ports 80 and 443, which are normally left open by network administrators to allow Web browsing. We introduce the following definitions:

- A Skype *client* is identified by the endpoint address, i.e., the (IP address, UDP/TCP port) pair.
- A Skype *flow* is the bidirectional set of packets having the same tuple (IP source and destination addresses, UDP/TCP source and destination ports, IP protocol type). A flow starts when a packet with a given flow tuple is first observed, and it is ended by either an *inactivity timeout* (set to 200s as later discussed) or, in case of TCP, by observing the tear-down sequence, if present. We further refer to the *sender* and *receiver* unidirectional flows to distinguish among the stream of packets coming from the same source and going to the same destination.

3. Measurement results

We report results that were collected by passively monitoring the campus access link at Politecnico di Torino for more than a month, starting from 22 April 2007.

Our methodology is as follows. Through the use of the classification framework [8], we were able to reliably identify individual voice/video calls initiated by Skype peers. As previously explained, all Skype communication events are multiplexed over the same transport layer port, so that a pair (IP,port) uniquely identifies a Skype peer. Since we monitored the campus network *continuously*, we were able to build a list of Skype peers that actively placed voice/video calls in our network. By using such a list, we obtained a subset of the traffic that originated from (or was transmitted to) the various Skype endpoints. Moreover, by means of [8] we were able to reliably filter out from the subset any voice/video calls, thereby allowing us to focus exclusively on the analysis of Skype signaling traffic.

During our measurement period, we observed about 1700 distinct internal Skype clients, out of the more than 7000 different hosts used by both students and staff members (in total, this comprised about 50,000 people). We present a subset of those results, namely, the first week during which we monitored Skype peers' activities, during which internal Skype clients contacted nearly 305,000 external peers, exchanging about 2.5 million flows for a total of 33 million packets.

Fig. 1 shows the changes during the week-long observation period in number of clients, flows, packets and bytes (from top to bottom, respectively) observed during 5-min time windows. Given the number of active clients (top plot), we observed a typical day–night periodicity during weekdays. A minimum of about 80 Skype clients were active at any given time, with a maximum of 260 during weekdays, and 120 during weekends. A similar periodicity was present in the number of flows, packets and bytes. However, the latter showed higher burstiness.

3.1. Signaling overhead

We note that the average signaling bitrate, i.e., the total number of signaling bits transmitted by a client over its whole lifetime, was very low. The left plot of Fig. 2 illustrates the Cumulative Distribution Function (CDF) of the average signaling bitrate. It is clear that the required signaling bandwidth is less than 100 bps in 95% of cases, while only very few nodes generate more than 1 kbps (these may have been SuperNodes).

Since the signaling bitrate is exiguous, its relative importance vanishes if weighted on the grounds of VoIP call traffic. For about 5% of the Skype clients, signaling accounted for only 5% of the total traffic (i.e., including voice and video calls). At the same time, since clients were left running for long periods without VoIP services being active, the signaling traffic dominated communication in 80% of all cases, accounting for more than 99% of the traffic generated by an average Skype client.

Let $C(p,i)$ be the number of different peers contacted by peer p considering the ith time interval since the start of peer activity, where time intervals are 5 min long. Distribution of $C(p,i)$ over all internal peers, and over all measurement intervals, is shown in the right-hand plot of

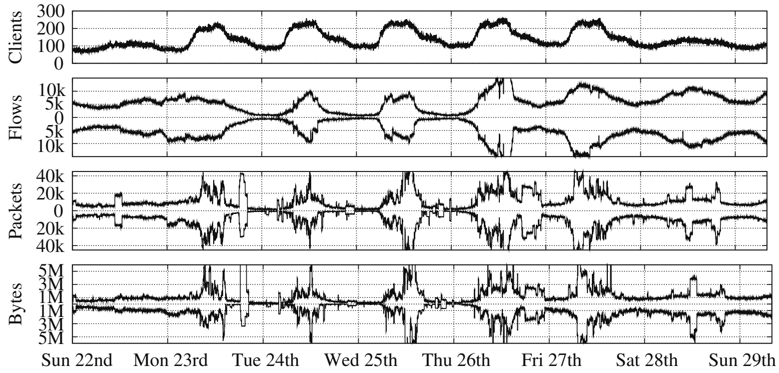

Fig. 1. Number of active clients, flows, packets and bytes observed every 5 min during the measurement week.

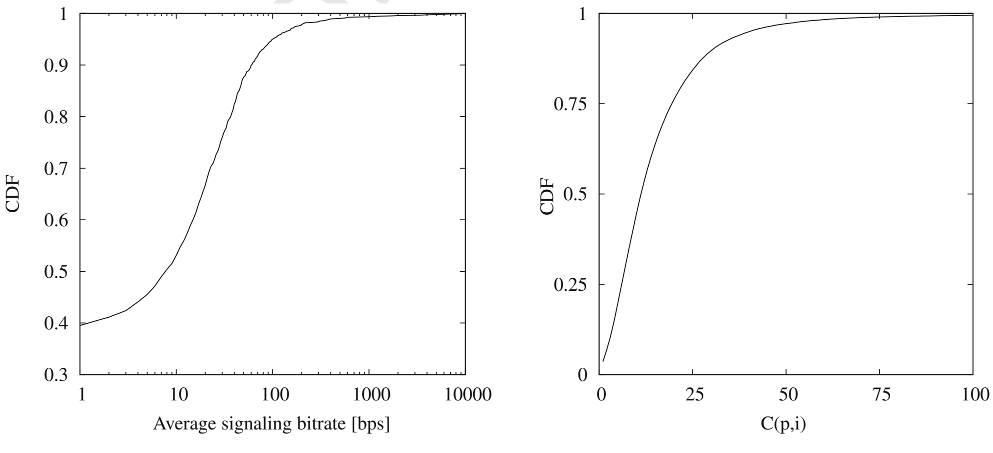

Fig. 2. Signaling bitrate (left) and number of contacted peers per unit time(right).

Please cite this article in press as: D. Rossi et al., Understanding Skype signaling, Comput.Netw. (2008), doi:10.1016/j.comnet.2008.10.013

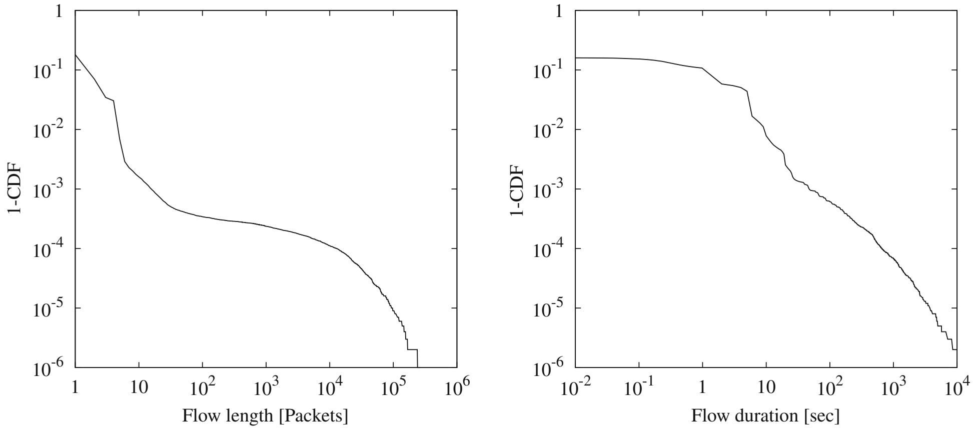

Fig. 3. Complementary distributions of the signaling flow volume (left) and duration (right).

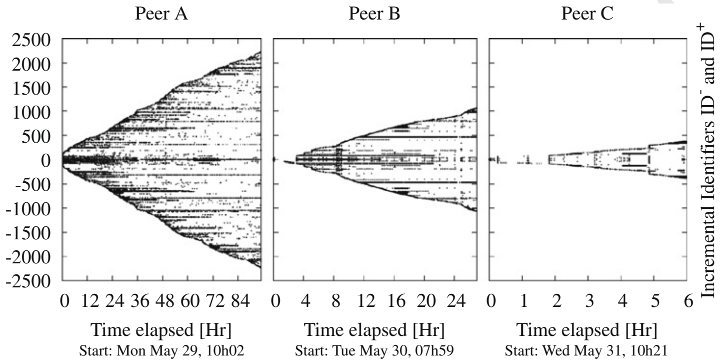

Fig. 4. Skype signaling activity: contacted peers over time.

Fig. 2. Every 5 min, a peer exchanges data on average with 16 other peers, and no more than 30 other peers are contacted in 90% of cases. Still, $C(p,i)$ can grow larger than 75 in 1% of the cases, which may constitute a burden for certain layer-4 devices that maintain flow states (e.g., a entry in a NAT table, a lookup in a firewall ACL table). We will show that many signaling flows are single-packet probes that create new temporary soft-state entries, many of which are rarely used thereafter.

3.2. Signaling flow classification

We wish to observe the signaling traffic that a Skype client exchanges. In particular, we examined measurements at the transport (flow) layer. The *semantic* of the signaling activity cannot be inferred from purely passive measurements, but the *form* of signaling activity can be differentiated. Let us consider the *source* signaling flow length (in packets) and the duration (in seconds) as a complementary distribution function (1-CDF) shown in Fig. 3 on a log–log scale. About 80% of the signaling flows consist of single packet probes, and 99% of the flows are shorter than 6 packets. At the same time, some persistent signaling activity is present, transferring a few MBytes of information over several thousand packets and lasting many hours. This is shown by the long tails in Fig. 3. Indeed, the single-packet flows account for less than 5% of all bytes exchanged.

We consider the schematic representation of typical Skype signaling activity. Let p be the observed peer and $I_p(x,t)$ be an indicator function that takes the value 1 if peer x sends/receives a packet to/from peer p at time t. $I_p(x,t)$ represents the P2P overlay topology evolution over time as seen by peer p. Peers $\{x\}$ will be identified by increasing numbers of identifiers (IDs) consistent with their arrival order. N is the total number of peers observed during the lifetime of peer p. Positive IDs are used for packets that were sent from p, negative IDs for packets sent to p.

Fig. 4 reports $I_p(x,t)$ considering three different peers, namely, the most active peer A that does not perform any voice calls during the observation period (left plot in the figure), a random peer B that generates only signaling traffic (center plot), and a randomly chosen peer C that has both signaling and voice flows (right plot). The figure shows that A has contacted (was contacted by) about 2500 other peers during its lifetime, whereas B and C (whose lifetimes are admittedly shorter) were contacted by about 1100 and 450 other peers respectively.

Three observations hold. First, the number of peers contacted exhibits an almost linear growth over time, suggesting that P2P network discovery was carried out during most of the peers' lifetimes. This part of the signaling activity is mainly carried out by transmission of a single packet, which (most of the time) is followed by some kind of acknowledgment. The fact that p knows the IP address and UDP/TCP port number of valid (but previously uncontacted) Skype peers means that the above information is acquired by some signaling message. Since some of the unknown contacted peers may have gone offline before *p* actually probes them, the positive and negative ID ranges are not exactly symmetrical. Second, some of the peers are contacted on a regular basis: in the activity plot, horizontal elements indicate that the same peer is periodically contacted during the lifetime of *p*. Finally, periodic information refreshments can be distinguished in the form of vertical patterns (clearly visible in the right-hand side of Fig. 4 about once an hour).

These observations suggest the existence of different types of signaling flows, which we classify depending on their *length* and *periodicity* as:

- *One-time probe*: Any packet sent to an unknown peer, to which a single reply packet possibly follows, but *no further packet* is exchanged between the peer pair. For the sake of brevity, hereafter we refer to one-time probes simply as probes.
- *Heartbeat*: A sequence of periodically exchanged one-time probes, separated by a time gap larger than the inactivity timeout, so that they are identified as different flows.
- *Dialog*: Any flow constituted by more than one packet.

In Fig. 4, heartbeats and dialogs can be easily recognized as dotted horizontal patterns and solid horizontal segments, respectively. Periodic information refresh operations, responsible for the vertical patterns, involve both heartbeats toward peers that are already known and discovery probes that target new peers.

Notice that the above definitions are sensitive to the setting of the end-of-flow inactivity timer, e.g., by setting the timeout to infinity, heartbeats will be classified as dialogs. However, we experimentally verified that the results are only very marginally affected by the choice of inactivity timer period, so long as it is smaller than a few minutes. Results reported in this paper were all generated by setting the timer to 200 s. This choice is justified by the fact that the largest regular inter-packet-gap that we ever observed was 180 s.

For the sake of simplicity, we distinguish signaling traffic depending on the kind of signaling activity in:

- *Probe* traffic, which is associated with probe flows;
- *Non-probe* traffic, which is associated with heartbeats and dialog flows.

3.3. Signaling flow characterization

We now analyze and characterize signaling traffic based on the proposed flow classifications. We focus on internal peers, and we investigate the resulting flows[1]. Table 1 summarizes the average amount of traffic due to external peers that exchange with a single peer (i) only probe flows (labeled as ‘probe’ in the table), (ii) only heartbeats, (iii) only dialogs or (iv) a mix of heartbeat and dialog flows. Results are reported considering the number of peers, flows and packets. Clients generate one-time probes with more than 50% of contacted peers. But only 8% of all observed flows are one-time probes, accounting for just 1% of signaling packets. Subsequently, internal clients exchange heartbeats alone with about 15.8% of their external contacts, which corresponds to about 26% (3%) of the flows (packets). By contrast, dialogs represent the only signaling activity for a quarter of all the peers (25.1%), accounting for a relatively modest percentage of flows (6.2%), but corresponding to a large number of packets (12.6%). Finally, a mixture of heartbeats and dialogs is exchanged with about 8% of all peers, which builds the bulk of the signaling activity in terms of flows (59.5%) and packets (83%).

Table 1
Per-source signaling traffic classification.

Level	Probe%	Heartbeat%	Dialog%	Mix%	Total No.
Peers	51.2	15.8	25.1	8.0	390126
Flows	8.0	26.3	6.2	59.5	2505622
Packets	1.0	3.1	12.6	83.3	18274451

Our results suggest that probe and non-probe traffic correspond to different kinds of signaling activity (possibly network discovery and network maintenance). To further confirm this intuition, the distribution of the UDP payload size reported in Fig. 5 shows that different information is carried by probe and non-probe traffic. The figure shows that probe traffic typically exhibits smaller packet size than non-probe traffic. Although it is not possible with a purely passive measurement technique and without reverse engineering of the protocol to make statements about Skype signaling operations, it is possible to conjecture that: (i) network discovery, carried out by means of probes, is a continuous activity; (ii) heartbeats are used to continuously ping contacts and friends, and to notify others of changes in their status; (iii) dialogs may be used to maintain the overlay, during call setup, and to update user information, etc.

4. Further insights into Skype signaling

In this section, we gather further insights into Skype signaling traffic, inferring some interesting properties of the Skype overlay network such as the churn rate, the geolocation of peers and their selection process, and the correlation of signaling traffic over time.

4.1. On the Skype churning rate

One of the parameters that affects P2P systems in general is the churn rate, i.e., the peer arrival and departure processes that force the P2P overlay to be updated. In order to understand the churn process in the Skype network, we performed a measurement of peer lifetime and deathtime. In particular, a peer is considered dead if no packet is sent

[1] We restrict our attention to internal peers, since we do not have access to all the traffic generated by external peers.

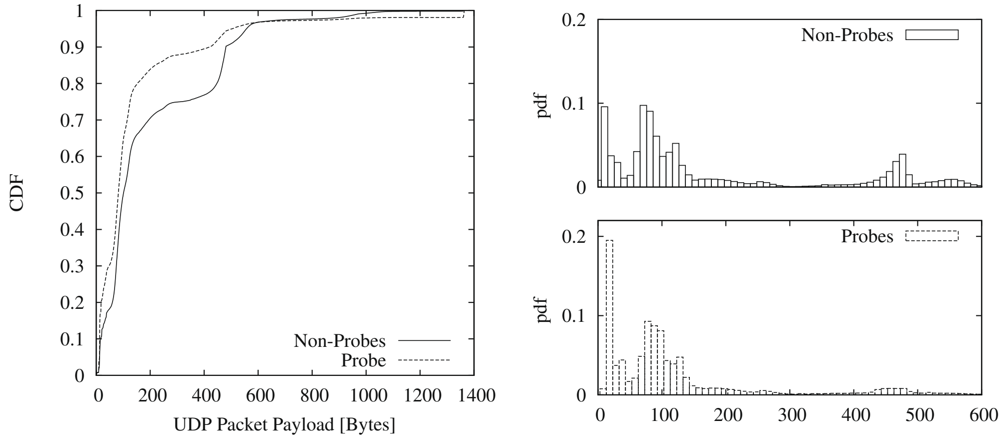

Fig. 5. Probe versus non-probe traffic packet size.

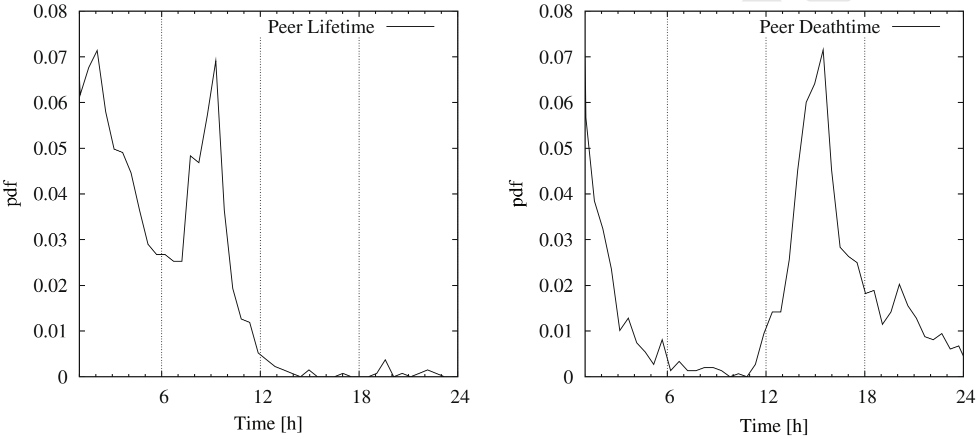

Fig. 6. Peer lifetime (left) and deathtime (right) pdf.

for a period of time longer that an idle time τ. Otherwise, the peer is considered alive. We experimentally verified that any value of τ larger that 200 s has a minimal impact on the lifetime measurements, and, therefore, we conservatively selected $\tau = 500s$. Fig. 6 shows the probability density function (pdf) of peer lifetime (left plot) and deathtime (right plot). We note that peer lifetimes are either short (one or two hours) or much longer (7–10 h). About 95% of peers disappear after 10 h of activity. However, the remaining 5% of peers have a lifetime that is much longer, with more than 1% remaining alive during the whole week. In respect of peer deathtime, we observed that the death period was either shorter than 2 h or longer than 11 h. The pdf also exhibits a heavier tail, indicating that about 2% of peers remained idle for more than 72 h.

The intuition behind this is that the majority of individuals run Skype by default, so that peer lifetime matches PC operation schedules; i.e., it is on during the day and off during the night. Nonetheless, some PCs are left running during the night as well, so in these cases, the Skype lifetime can be much longer. This confirms our intuition that Skype's churn rate is very low, which contributes to limiting the P2P overlay maintenance costs and update rates.

4.2. On the geolocation of Skype peers

We now consider the geographical locations of Skype peers. In our dataset, we observed 263,886 different IP addresses. We queried the geographical locations of the above addresses using `HostIP` [10], a public, open and free IP address database.

The resulting geolocation is shown in Fig. 7, which depicts the subset of about 10k peers (out of about 264k que-

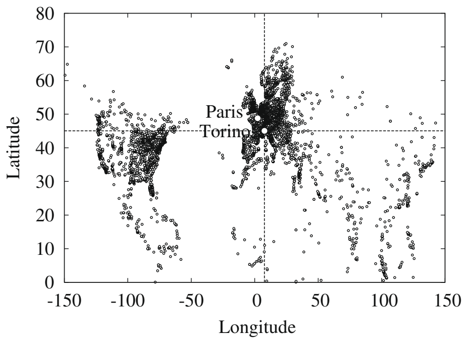

Fig. 7. Peer geolocation: graphical representation.

ries), for which *precise* longitude and latitude information was available. From the graphic, one can readily recognize the shapes of the different continents, especially Europe and North America. Two white landmarks identify the cities of Paris, France and Torino, Italy.

Further details on the geolocation of the entire Skype peer dataset are given in Table 2. We stress that, in this case, the number of successful geolocation events increases significantly since continent and country information are more easily identified with respect to the precise longitude and latitude information used early in Fig. 7.

The table shows a breakdown, considering probe versus non-probe traffic, in peers per continent (left), European country (center) and Italian city (right). The locations are sorted by decreasing percentage, and only the eight top locations are listed. The total number of Non-Probe and Probe traffic events is reported at the bottom of the Continent Breakdown table. Elements in bold represent those that are geographically closest to the measurement point, i.e., the Politecnico di Torino campus.

Two important considerations can be taken from Table 2. First, the probing mechanism tends to treat nearby hosts preferentially: indeed, nearly half of the probed IPs (45.4%) were located in Europe, nearly four times as many as in North America (11.9%). This means that the probing mechanism tends to discover network hosts that are geographically close. Second, the geographical location is much less important for non-probe traffic: indeed, as the percentage of peers that are located in Europe actually decreases (38.2%) with respect to probe traffic, the percentage of North American peers nearly doubles (23.1%). Considering that users resort to Skype to lower communication fees and to keep in contact with others who are very distant, we are not surprised that non-probe traffic is more geographically dispersed. Indeed, the relationship among users forces Skype peer selection with respect to non-probe traffic. By contrast, the peer discovery mechanisms implemented by one-time probes are driven by the physical properties of the underlying network. Similarly, probe traffic is roughly distributed consistent with the population sizes of Italian cities. Non-probe traffic, on the other hand, is influenced by user social networks and favors peers in Torino. The breakdown by EU countries does not show significant differences between probe and non-probe traffic.

4.3. Peer selection criteria

Fig. 8 shows the pdf of Round-Trip Time (RTT) between two peers, measured as the time elapsed between observing the packet leaving the campus LAN and the response packet (if any) being returned. In the case of non-probe traffic, the first sent–received packet pair is used to estimate the RTT. This measurement takes into account both the network and the application latency.

The information in the graphic confirms our previous intuition: the latency of probe traffic is lower than that of non-probe traffic. Given Torino's location, RTTs shorter than 100 ms are typical of nodes within the European Union, while RTTs of above 100 ms are typical of nodes outside the EU. Our measurement results suggest that the probing mechanism is *latency driven*: the Skype client probes for peers based on the information received by other peers, so that low latency peers are more likely to be selected than higher latency ones. Conversely, the peer selection mechanism is *preference driven*, where the preference criterion depends on the user relationships.

We also investigated the degree of "clustering" of the Skype overlay network. For a given peer *p*, let the *popularity* be the number of peers that *contacted* it; i.e., an internal (external) peer has a popularity *x* if it is contacted by *x* external (internal) peers. The popularity distribution is depicted in Fig. 9, showing probe and non-probe traffic separately. Consistent with earlier considerations, non-probe traffic popularity pertains to the degree of clustering of users at Politecnico di Torino. Conversely, probe popularity

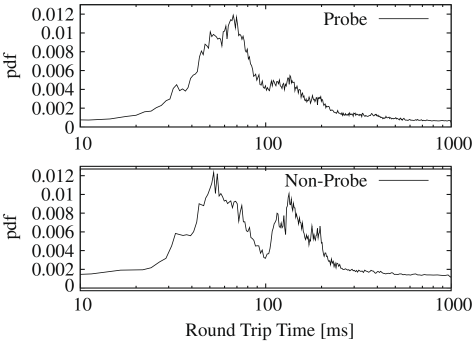

Fig. 8. Probe versus non-probe traffic: round-trip times.

Table 2
Peer geolocation: percentage breakdown by continent, European country and Italian city.

	Continent breakdown		EU countries breakdown				Italian cities breakdown			
	Non-probe	Probe	Non-probe		Probe		Non-probe		Probe	
Europe	38.2	45.4	17.9	FR	21.4	DE7	32.5	Torino	30.0	Roma
America NO	23.1	11.9	15.4	DE	17.5	PL	21.7	Milano	23.8	Milano
Asia	12.1	11.7	13.4	IT	15.0	FR	18.9	Roma	17.1	Torino
America SO	3.0	2.7	10.4	NL	11.2	IT	8.9	Bologna	8.1	Bari
Africa	1.8	2.2	10.0	SW	8.2	ES	4.7	Bari	5.9	Firenze
Oceania	0.8	0.7	9.0	BE	6.5	BG	4.7	Napoli	5.8	Bologna
UNKNOWN	21.0	25.4	8.4	PL	6.4	SW	4.4	Firenze	4.8	Padova
TOT number	51358	212528	6.3	FI	5.8	BE	4.2	Moncalieri	4.4	Napoli

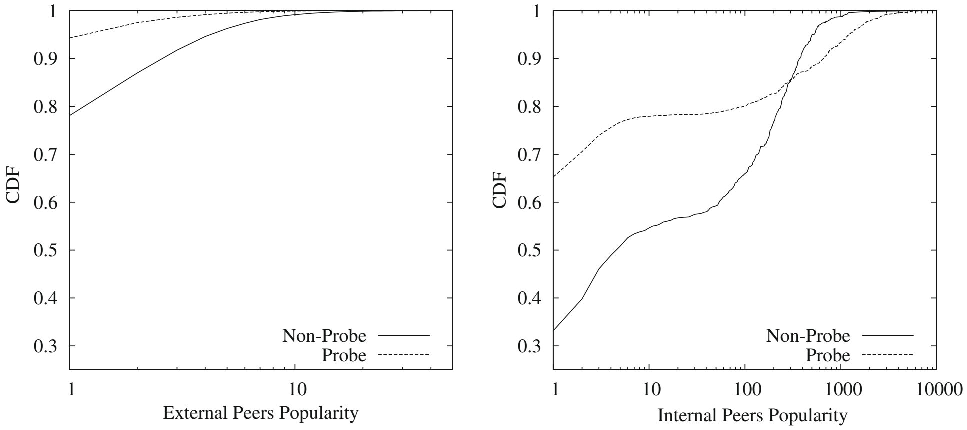

Fig. 9. Peer popularity of external (left) and internal (right) peers.

may help to reveal SuperNodes that are probed more frequently that random peers. Interestingly, this clearly emerges from the external flows directed toward internal peers (right plot of Fig. 9). Indeed, for probe traffic, the popularity metric is 1 in about 65% of the cases; i.e., the internal peer has been contacted by a single external peer. The CDF then increases until the popularity reaches 10. It remains constant thereafter until the popularity reaches much higher values (100 or more). This suggests that the internal peer is a SuperNode that attracts substantial signaling traffic from external peers (note that PCs in the campus LAN are not protected by firewalls and use public IP addresses, so it is very likely that some PC can be elected as a supernode); the phenomenon is similar for probe traffic.

Conversely, in the case of traffic directed toward external peers, the phenomenon is no longer evident since the number of internal clients is much smaller (1700) with respect to the external clients (305,000).

4.4. *Time correlation over peers*

Another interesting property of signaling activity in P2P systems is the possible periodicity that may be present when contacting other peers. To highlight such periodicity, we extended our definition of time correlation to consider that the activity pattern $I_p(x,t)$ of peer p evolves both over time t and for different peers x.

Let $\mathbb{I}_p(x,i)$ be an indicator function that takes the value 1 if peer p is active during the ith time interval

$$\mathbb{I}_p(x,i) = \begin{cases} 1 & \text{if } I_p(x,t) > 0, \quad t \in [i\Delta, (i+1)\Delta) \\ 0 & \text{otherwise.} \end{cases}$$

$\mathbb{I}_p(x,i)$ is the discrete time equivalent of the activity pattern $I_p(x,t)$, where Δ is the quantization time step. Let $\underline{s}_p(i)$ be the vector of peers that exchange a packet with peer p at interval i, where N is the total number of peers that exchange packets with p:

$$\underline{s}_p(i) = \langle \mathbb{I}_p(1,i), \mathbb{I}_p(2,i), \ldots, \mathbb{I}_p(N,i) \rangle.$$

We can then define the normalized *peer correlation* $C(j)$ as

$$C(j) = \frac{1}{kM}\sum_{i=1}^{M}\langle \underline{s}_p(i) \cdot \underline{s}_p(i+j) \rangle \quad j \neq 0 \tag{1}$$

$$k = C(0) = \frac{1}{M}\sum_{i=1}^{M}\langle \underline{s}(i) \cdot \underline{s}(i) \rangle, \tag{2}$$

where $\langle \underline{u} \cdot \underline{v} \rangle$ is the scalar product between vectors $\underline{u}$ and $\underline{v}$ and M is the number of time intervals over which peer correlation is averaged.

$C(j)$ represents the average number of peers at interval i that are also active at interval $i+j$. The peer correlation is defined by averaging over M different time intervals and is normalized to the average number of active peers $C(0)$. Intuitively, large values of $C(j)$ indicate that a large fraction of peers are also active after j time intervals. By contrast, other values of $C(j)$ indicate that, after j time intervals, the set of active peers is very different. Finally, if $C(j) = 0$, then no currently active peer is still active after j time intervals.

The normalized correlation function is shown in Fig. 10 for the same two peers p_1 and p_2[2] whose activity pattern is plotted in Fig. 4; $\varDelta = 1$s. Spikes at $j = 20, 40, 60, \ldots$ show that peers periodically poll previously contacted peers every 20 s, which was not obvious from looking at the activity pattern. Notice that the most active peer (left plot) features smaller spikes, since the average number of active peers, $C(0)$, is rather large. By contrast, peer p_2 exchanges information with a more limited number of peers and periodically re-contacts about 1/3 of them every 20 s.

Clearly, by definition, the periodic polling involves dialogs. Moreover, since the time at which external peers are first contacted is jittered, periodic polling does not result in noticeable load spikes that are tied to signaling traffic.

5. Measurements during Skype's summer outage

As previously stated, one of the characteristics that made Skype a very successful application stems from its robustness: Skype has been very reliable, almost like a PSTN network.

[2] Results for other peers are very similar and are not reported here.

Please cite this article in press as: D. Rossi et al., Understanding Skype signaling, Comput.Netw. (2008), doi:10.1016/j.comnet.2008.10.013

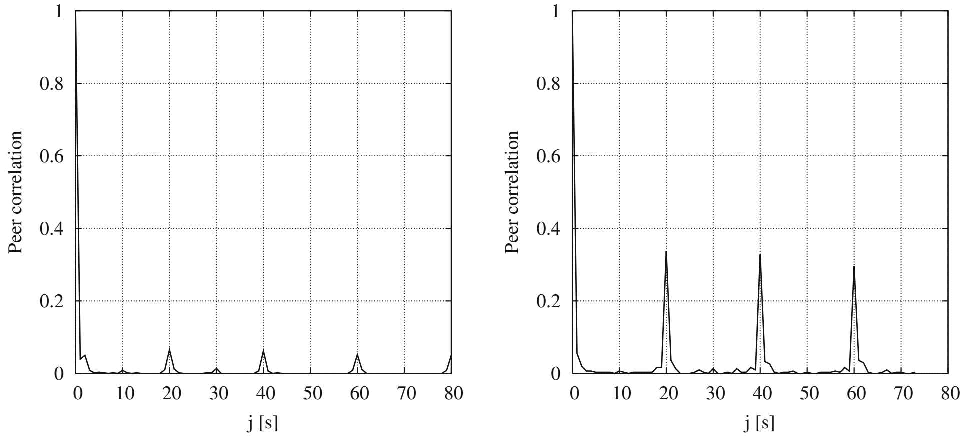

Fig. 10. Time correlation of the most active peer (left plot) and a random peer (right plot).

However, despite the fact that the Skype overlay has been fully functional 24/7 over the past few years, thereby confirming the effectiveness of its self-healing capabilities, Skype suffered an unexpected outage last summer. Quoting the official company blog [11], “On Thursday, 16th August 2007, the Skype peer-to-peer network became unstable and suffered a critical disruption. The disruption was triggered by a massive restart of our users’ computers across the globe within a very short time frame as they rebooted after receiving a set of patches through Windows Update.” As we monitored the campus network during that period, we were able to observe the outage: given its extreme nature, i.e., the disruption of an Internet-scale overlay, it is interesting to investigate this event.

We report interesting measurements that were observed before, during and after the Skype outage. It took more than two days before the Skype engineering team managed to get the situation back to normal (see “The words we have all been waiting for”, posted August the 18th at 11h00 GMT on [11]) after the problem was first acknowledged (see “Problems with Skype login”, posted the August the 16th at 14h02 GMT). The start time was 11AM GMT, which corresponds to the instant at which we begin to observe an anomalous (and massive) increase in the amount of UDP traffic. The time at which the Skype engineering team blogged that the situation was back to normal is considered to be the end time. For reference, we also considered two different time intervals during August 2007, one week before and one week after the outage period:

- Before: from Thu 09 (11AM) to Sat 11 (11AM) of August.
- During: from Thu 16 (11AM) to Sat 18 (11AM) of August.
- After: from Thu 23 (11AM) to Sat 25 (11AM) of August.

5.1. Traffic volumes

The volume of traffic that we observed during these periods is reported in Fig. 11, which includes the number of clients, flows, packets and bytes observed over 1-min time windows. Comparing Fig. 11 with Fig. 1, similar trends are observed during the normal Skype operation period. However, we note that traffic volumes are smaller in Fig. 11, since August is a typical Italian vacation period (with 15 August being the typical holiday peak). At the same time, the number of internal active Skype clients (top plot) is very similar before and after the failure, which

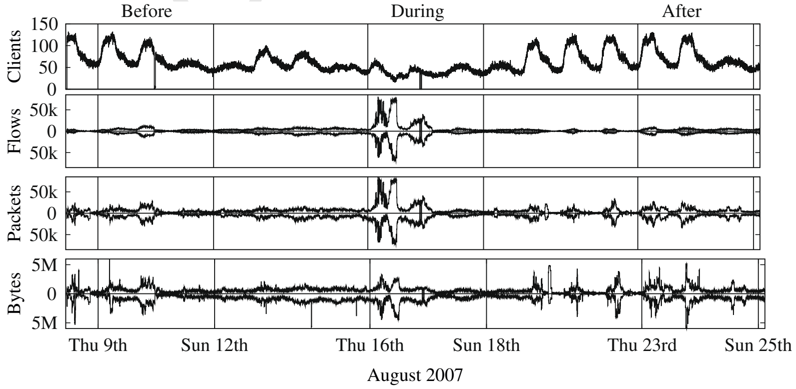

Fig. 11. The volume of Skype flows, packets and bytes for the periods before, during and after the outage (1 min windows).

allows us to accurately compare measurements during different weeks.

Focusing on the two days of Skype outage, a drastic reduction in the number of Skype clients is observed (although a slight decrease was already in progress). This corresponds with an anomalous increase of UDP traffic (in terms of flows and packets). During this period, UDP traffic largely outweighs TCP traffic, so that it accounts for almost all packets passing into our campus network. Before and after the Skype outage, UDP traffic volumes were much lower than the amount of TCP traffic.

Indeed, during the outage, Skype traffic accounted for almost all (94%) UDP flow and for a very significant portion of UDP packets and bytes (89% and 69%). At the same time, almost all this traffic was generated/received by the 10 most active clients; furthermore, the most active Skype node was responsible of 50% of all bytes, 67% of all packets, and 73% of all flows: a clear overload situation. Thus, during the outage we observed more than an average 3- and 4-fold increase in number of packets and flows, respectively, and this may increase by up to an order of magnitude for the most active clients.

5.2. *Traffic properties*

We now distinguish between probe and non-probe traffic to quantify the type of observed signaling traffic. The top part of Table 3 distinguishes between probe and non-probe traffic, reporting how many external peers have contacted an internal peer with a single-probe packet and have (or have not) received a reply. *Replied probes* represents the vast majority of the traffic exchanged on the Skype overlay during normal conditions (before 69.2%, after 67.1%), and during the anomalous event (82.5%). Notice that, during the outage, internal nodes are contacted by more than 40 million peers, far larger (almost 20 times) than under normal conditions.

Another interesting figure can be gathered from these data: the total number of external peers with which our campus exchanged traffic during the anomalous event is about 40 million, more than one order of magnitude larger than during normal functioning. Even more interesting is the fact that the most active internal client contacted more than 11 million peers, a more than 30-fold increase compared to the normal operation point (300k peers).

The reported numbers show that the cost of maintaining a P2P database may not be negligible, and that, in adverse conditions, a single peer can generate the same amount of traffic as is normally generated by all peers across an entire campus network.

Table 3
External peer type, internal contacts and further traffic details.

	Before		During		After	
External peer type	No.	%	No.	%	No.	%
Probe	620k	17.3	5.62M	13.9	663k	19.1
Replied-probe	2.47M	69.2	33.4M	82.5	2.33M	67.1
Non-probe	484k	13.5	1.45M	3.6	481k	13.8
Total peers	3.57M		40.5M		3.48M	
Top-1 peers	376K		11.3M		333K	

6. Conclusions

In this paper, we investigated Skype signaling traffic by means of passive measurements, providing insights into Skype signaling mechanisms, and allowing for a better understanding of the cost and complexity of managing a P2P architecture. In particular, we observed that Skype signaling traffic comprises the following: (i) probe traffic flows, in which a pair of packets are exchanged between two peers and are used to discover new nodes; (ii) periodic heartbeats flows that are used to exchange information about the status of peers of interest in the user's contact network, (iii) long dialog flows that carry the most signaling information and support the maintenance of the overlay network.

Our results offer empirical evidence of the fact that Skype prefers to flood the network with short single-probe events that target many hosts. This may be as effective for the purpose of overlay maintenance as it is costly from the viewpoint of layer-4 network devices.

Interestingly, Skype performs network discovery by accounting for geographical peer location (i.e., in terms of latency), while the overlay network is also influenced by the user's network of contacts.

Finally, we report measurements collected during a Skype outage event that lasted two days. During the outage, we observed a 4-fold increase in the number of flows, a 3-fold increase in the packet sending rate and a 10-fold increase in the number of contacted peers. At the same time, the most active peer in our network experienced a 10-fold increase in traffic and a 30-fold increase in the number of contacted peers, topping 11 million signaling connections. This gives the sense of the complexity of the algorithms required to maintain a P2P system.

Acknowledgements

The Italian team was funded by the Italian Ministry of University, Education and Research (MIUR) through the PRIN project RECIPE.

References

[1] Skype web site, <http://www.skype.com>.
[2] International carriers' traffic grows despite Skype popularity, TeleGeography Report and Database. Available from: <http://www.telegeography.com/>, December, 2006.
[3] S.A. Baset, H. Schulzrinne, An analysis of the skype peer-to-peer internet telephony protocol, IEEE Infocom'06, Barcelona, Spain, April 2006.
[4] P. Biondi, F. Desclaux, Silver Needle in the Skype, Black Hat Europe'06, Amsterdam, The Netherlands, March 2006.
[5] S. Guha, N. Daswani, R. Jain, An experimental study of the skype peer-to-peer VoIP system, 5th International Workshop on Peer-to-Peer Systems, Santa Barbara, CA, February 2006.
[6] K. Ta Chen, C.Y. Huang, P. Huang, C.L. Lei, Quantifying skype user satisfaction, ACM Sigcomm'06, Pisa, Italy, September 2006.
[7] K. Suh, D.R. Figuieredo, J. Kurose, D. Towsley, Characterizing and detecting relayed traffic: a case study using Skype, IEEE Infocom'06, Barcelona, Spain, April 2006.
[8] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, P. Tofanelli, Revealing skype traffic: when randomness plays with you, ACM Sigcomm'07, Kyoto, Japan, August 2006.
[9] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, D. Rossi, Tracking down skype traffic, IEEE Infocom'08, Phoenix, AZ, 15,17 April 2008.
[10] <http://www.hostip.info>.
[11] Skype Heartbeats Archives, <http://heartbeat.skype.com/2007/08/>.

Dario Rossi was born in Torino, Italy, on a sunny sunday of March 1977. He received his M.Sc. degree in Electronic Engineering in 2001 and his Ph.D. in Electronic and Telecommunication Engineering in 2005, both from Politecnico di Torino. Between September 2003 and August 2004 he held a visiting researcher position in the Computer Science division at University of California, Berkeley. Since October 2006, he is an Associate Professor at the INFRES department at Ecole Nationale Superieure de Telecommunications (ENST), in Paris, France. He has coauthored over 30 papers in leading conferences and journals and currently holds 3 patents. He participated in the program committees of several conferences including IEEE ICC, IEEE IPCCC and IEEE ISCC. He is responsible for several European research projects, such as FP7 NAPA-WINE, Celtic TIGER and Celtic TRANS. His research interests are in the fields of peer-2-peer networks, Internet traffic measurement, sensor and vehicular networks.

Marco Mellia received his Ph.D. in Telecommunications Engineering in 2001 from Politecnico di Torino. From March to October 1999 he was with the CS department at Carnegie Mellon University as visiting scholar. From February to March 2002 he visited the Sprint Advanced Technology Laboratories Burlingame, California, working at the IP Monitoring Project (IPMON). Since April 2001, he is with Electronics Department of Politecnico di Torino as Assistant Professor. He has co-authored over 100 papers published in international journals and presented in leading international conferences, all of them in the area of telecommunication networks. He participated in the program committees of several conferences including IEEE Infocom, IEEE Globecom and IEEE ICC. His research interests are in the fields of all-optical networks, traffic measurement and modeling, peer to peer systems.

Michela Meo received the Laurea degree in Electronic Engineering in 1993, and the Ph.D. degree in Electronic and Telecommunication Engineering in 1997, both from Politecnico di Torino. Since November 1999, she is an Assistant Professor at Politecnico di Torino. She co-authored more than 100 papers, about 40 of which are in international journals. She edited six special issues of international journals, including ACM Monet, Performance Evaluation Journal and Computer Networks. She was in co-chair of three editions of ACM MSWiM, program co-chair of IEEE QoS-IP 2005, Movenet 2007, IEEE ISCC 2009. Her current research interests are in the field of traffic characterization and classification, peer-to-peer systems, energy efficient networks.