# Detailed Analysis of Skype Traffic

Dario Bonfiglio, Marco Mellia, Michela Meo,
Politecnico di Torino – Dipartimento di Elettronica
email: name.surname@polito.it

Dario Rossi
TELECOM ParisTech – INFRES
email: dario.rossi@enst.fr

***Abstract*—Skype is beyond any doubt *the* VoIP application in the current Internet application spectrum. Its amazing success has drawn the attention of telecom operators and the research community, both interested in knowing its internal mechanisms, characterizing its traffic, understanding its users' behavior.**

**In this paper, we investigate the characteristics of traffic streams generated by voice and video communications, and the signaling traffic generated by Skype. Our approach is twofold, as we make use of both active and passive measurement techniques to gather a deep understanding on the traffic Skype generates. From extensive testbed experiments, we devise a source model which takes into account: i) the service type, i.e., SkypeOut calls or calls between two Skype clients, ii) the selected source Codec, iii) the adopted transport layer protocol, and iv) network conditions. Leveraging on the use of an accurate Skype classification engine that we recently proposed, we study and characterize Skype traffic based on extensive passive measurements collected from our campus LAN.**

## I. Introduction

The last few years witnessed VoIP telephony gaining a tremendous popularity, as testified by the increasing number of operators that are offering VoIP-based phone services. Skype [1] is beyond doubt the most amazing example of this new phenomenon: developed in 2002 by the creators of KaZaa, it recently reached over 170 millions of users, and accounts for more than 4.4% of total VoIP traffic [2].

Being the most popular and successful VoIP application, Skype is attracting the attention of the research community [3]–[10], and of the telecom operator as well. Many interesting questions related to its internal mechanisms, the traffic it generates and the users' behavior remain, to date, unanswered. The complexity stems from the fact that Skype protocols are proprietary, and that an extensive use of cryptography, obfuscation and anti reverse-engineering techniques [4] are adopted by Skype creators. Finally, Skype implements a number of techniques to circumvent NAT and firewall limitations [5], which add further complexity to an already blurred picture.

In previous work, we devised a methodology that successfully tackles the problem of Skype voice traffic identification [3]. We extend here the methodology to identify also video-calls and voice calls generated by the newly deployed SVOPC Codec. Moreover, via a wider set of active and passive measurements we investigate Skype users' behavior and some internal mechanisms. A preliminary version of this paper appeared in [6]. In this version we complete the characterization of Skype voice Codecs, and add several details to both the signalling traffic study, and users' behavior measurement.

The main contributions of this paper are the following. First, we characterize the traffic generated by voice and video calls, by observing the time evolution in terms of bit rate, inter-packet gap, packet size. Besides distinguishing among various voice Codecs that Skype uses, we also unveil the different behavior of the traffic source based on the adopted transport layer protocol. Second, we observe how Skype reacts to different and changing network conditions. Third, we focus on the users' behavior by analyzing the number of flows generated in the time unit, the call duration – which unsurprisingly is very much related to the tariff policies – and the churning process. Fourth, we analyze the signaling traffic generated by a Skype client, considering the number of different clients that are contacted by a peer, which gives a feeling about the cost of maintaining the P2P architecture.

While many details about the Skype protocols and internals can be found in [4], [5], few papers deal with the issues of Skype identification [3], [7], and traffic and users' characterization [8]–[10]. In [7], authors focus on the identification of relayed[1] traffic only, using Skype as an example of application: little results are therefore presented about Skype source characterization. Authors of [8] present an experimental study of Skype, based on a five month long measurement campaign. Lacking a reliable Skype classification engine, authors are forced to limit the scope to relayed sessions, and they restrict furthermore their attention to the case of UDP transport layer only. Works closest to ours are [9], [10]. In [9], authors focus on the evaluation of the QoS level provided by Skype calls. As the adopted VoIP traffic classification criterion is fairly simple, authors cannot distinguish between video and voice, end-to-end and SkypeOut calls, and cannot account for the impact of transport protocols. Authors in [10] instead investigate the Skype congestion control algorithm considering video-calls, exploring Skype reaction to variation of the available bandwidth and its TCP friendliness. Finally, all previous papers completely ignore Skype signaling traffic except [5], although the focus is different – i.e., authors analyze the login phase, and how Skype traverses NAT and firewalls rather than providing quantitative insights into Skype signaling traffic.

## II. Skype Premier

The main difference between Skype and other VoIP clients is that Skype is based on a P2P architecture, rather than a more traditional client-server model. Only user's authentication is performed under the classical client-server model, using

[1] A session is *relayed* if packets from a source to a destination are routed through an intermediate node which acts as an application layer relay.

TABLE I
NOMINAL CHARACTERISTICS OF SKYPE CODECS.

| Codec | Frame Size [ms] | Bitrate [kbps] |
|---|---|---|
| ISAC* | 30,60 | 10 ÷ 32 |
| ILBC | 20,30 | 13.3, 15.2 |
| G.729 | 10 | 8 |
| iPCM-wb* | 10,20,30,40 | 80 (mean) |
| EG.711A/U | 10,20,30,40 | 48,56,64 |
| PCM A/U | 10,20,30,40 | 64 |
| SVOPC* | 20÷60 | 20÷50 |
| TrueMotion VP7 | Unknown | $> 20$ |

$*$ *wideband Codecs*

public key mechanisms. After the user (and the client) has been authenticated, all further signaling is performed on the P2P network, so that Skype user's informations (e.g. contact list, status, preferences, etc.) are entirely decentralized and distributed among P2P nodes. This way the service scales very easily to large sizes, and the costs of a centralized infrastructure are avoided. Peers in the P2P architecture can be normal nodes or supernodes. The latter ones are selected among peers with large computational power and good connectivity (considering bandwidth, uptime and absence of firewalls) and they take part to the decentralized information distribution system which is based on a DHT.

Skype offers end users several services: i) voice communication, ii) video communication, iii) file transfer and iv) chat services. The 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 service, in which case a fee is applied. In the following, we denote by *End-to-End (E2E)* any voice/video call between two Skype clients, and by *End-to-Out (E2O)* any call involving a Skype peer and a PSTN terminal.

From a protocol perspective, Skype uses a proprietary solution which is difficult to reverse engineer due to extensive use of cryptography and obfuscation techniques [3]–[5]. Though Skype may rely on either TCP or UDP at the transport layer, both signaling and communication data are preferentially carried over UDP.

Considering voice services, Skype chooses a Codec from a list according to an unknown algorithm. It is however possible to force Codec selection and we exploit this feature to observe the different behavior of the Skype source when using different Codecs. The supported Codec name, nominal frame size and bitrate are reported in Tab. I, where Wide-band Codec (offering 8 kHz bandwidth) are labeled by a "$*$" symbol. All Codecs are standard except the ISAC one, which is a proprietary solution of GlobalIPSound [11]. Some are Constant Bitrate (CBR), while others are Variable Bitrate (VBR) Codecs. G.729 Codec is preferred Codec for E2O (SkypeOut) calls, while ISAC has been the preferred one for E2E (End-to-end) calls until version 3.2, starting from which Skype prefers the SVOPC [12] Codec. Considering video services, Skype adopts TrueMotion VP7 Codec, a proprietary solution of On2 [13], which provides a variable bitrate stream with minimum bandwidth of 20 kbps. No other detail is available.

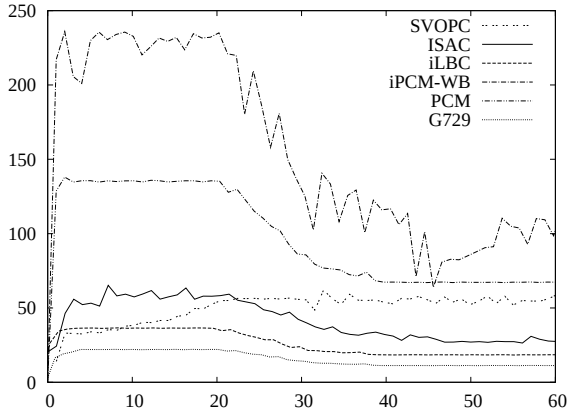


Fig. 1. Bitrate traces versus time for different voice Codecs - UDP at the transport layer, no artificial delay and loss.

## III. VOICE AND VIDEO STREAMS CHARACTERIZATION

In order to derive a source model, we performed several experiments in a controlled environment: the testbed involved several PCs connected by a Linux NAT / Firewall / Router / Traffic-Analyzer box. The typical experiment involves two PCs at a time (a sender and a receiver); different PCs with specific hardware and software characteristics were used to test various versions of Skype and different operating systems such as Windows, Linux and Pocket-PC. Several network scenarios were emulated by the Linux router using NIST Net [14] to enforce various combinations of delay, packet loss and bottleneck bandwidth, and to observe how Skype reacts to different network conditions. Different types of access technology (i.e., WiFi, Ethernet, UMTS) were also investigated, as well as their combination. Overall, our active experimental campaign comprises about 100 experiments corresponding to more than 17 hours worth of Skype traffic, in which we generated and captured a traffic volume of 776 Mbytes, exchanged in nearly 5 million packets. A subset of the above testbed traffic is made available to the research community [15].

A monodirectional flow is identified by using the traditional tuple (IP source and destination addresses, UDP/TCP source and destination ports, IP protocol type)[2]. A flow starts when a packet with the flow tuple is first observed, and ends when an inactivity timeout expires (the timeout is conservatively set to 200 s) or, in case of TCP, by observing the connection teardown sequence. Flow characterization is provided by the following measurement indexes, which are typical of streaming services over packet networks:

- Bitrate ($B$): amount of bits generated at application layer in a time interval of 1 second.
- Inter-Packet-Gap ($IPG$): time between two consecutive packets belonging to the same flow.
- Payload length ($L$): number of bytes transported in the TCP or UDP payload; the corresponding IP packet size can be determined by adding the transport and network layer overheads.

[2]We separately analyze and track monodirectional flows, so that each call is built by two flows.

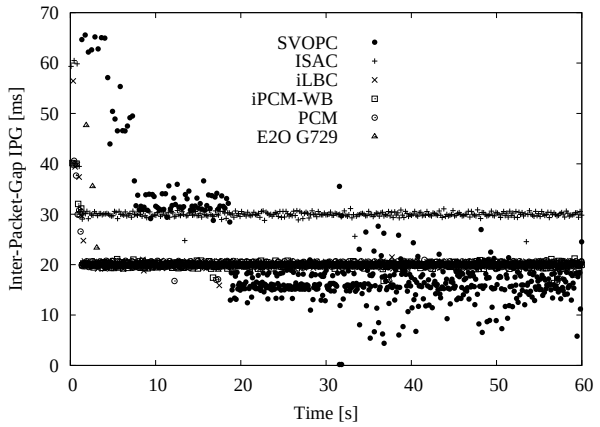


Fig. 2. $IPG$ traces versus time for different voice Codecs - UDP at the transport layer, no artificial delay and loss.

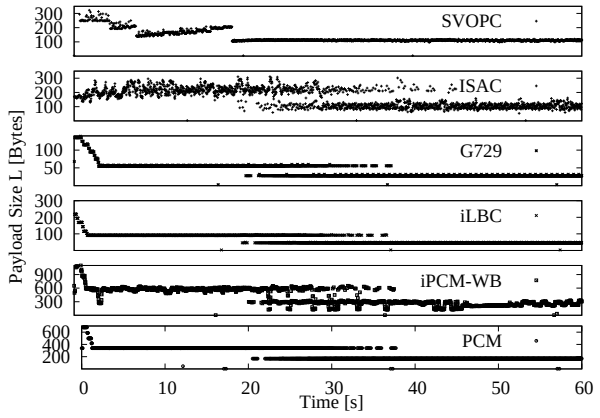


Fig. 3. $L$ traces versus time for different voice Codecs - UDP at the transport layer, no artificial delay loss.

We use the Skype source model proposed in our previous work [3]. According to this model, three parameters determine the characteristics of the generated traffic: i) *Rate* is the bitrate used by the source, e.g., the Codec bitrate; ii) $\Delta T$, that represents the Skype message framing time, is the time elapsed between two subsequent Skype messages belonging to the same flow; iii) *RF* is the Redundancy Factor, i.e., the number of previous blocks that Skype retransmits, independently from the adopted Codec, along with the current encoded block. The above parameters may change during an ongoing phone call: as we show in the following, Codec *Rate* and *RF* are the preferred knobs used by Skype to react to changing network conditions; however, changes of $\Delta T$ are also possible.

### A. Voice flows characterization

In this section we analyze the traffic generated by voice flows. We perform a first set of experiments by generating voice calls between two PCs directly connected by a LAN with no interfering traffic, and no imposed artificial delay or packet loss; one experiment for each available Codec was performed- we focus on the case of flows transported by UDP, the preferred transport protocol.

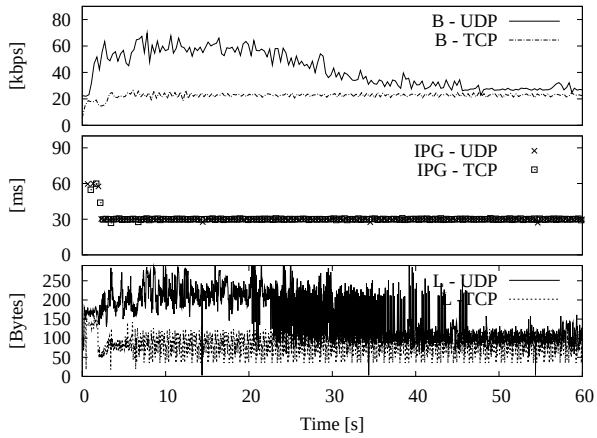


Fig. 4. $B$, $IPG$ and $L$ traces versus time for a voicecall - UDP or TCP at the transport layer, no artificial loss, ISAC Codec.

Figures 1, 2 and 3 report, respectively, the bitrate, $B$, the inter-packet-gap, $IPG$, and the payload length, $L$, versus time for different voice Codecs. Due to the different characteristics of each Codec, a Skype voice call can consume up to 230 kbps and as few as 11 kbps. Neglecting SVOPC for the moment, we notice that independently from the adopted Codec, three phases can be easily distinguished in the traces: during the first 20 s, the bitrate is high; then, a transient period between 20 and 40 s follows, during which the bitrate smoothly decreases; finally, during the third portion of the trace, $t \geq 40$ s, the bitrate is roughly half the one at the trace beginning. This is likely due to an *aggressive* initial setting of the redundancy factor $RF = 2$. This setting is typical of bad network conditions, when packet losses are present, and it aims at reducing the impact of possible losses: it is apparently used during the flow initial phase, when network conditions are unknown, in order to aggressively enforce high quality: after a short time, when Skype realizes that network conditions are good, $RF$ is set to 1. Conversely, in the case of SVOPC, the service bitrate shows a different behavior, slowly increasing from 25 Kbps to about 50 Kbps: on the one hand, the adoption of SVOPC makes Skype more "network friendly" during the initial probing phase; on the other hand, it makes it also more "bandwidth eager" over time, since SVOPC generates a bitrate that is about twice the one of the ISAC Codec after the transient.

Considering the $IPG$ measurements reported in Fig. 2, we observe that, for all Codecs but SVOPC, $IPG$ is almost constant during the three phases, meaning that the bitrate variability is not obtained modifying the $IPG$. During the very beginning of the traces (roughly 1 s), Skype performs a frame size tuning, reflected in the $IPG$ taking values in 30, 40, 60 ms before assuming the regime value which is equal to 30 ms for ISAC and 20 ms for all the other Codecs. SVOPC, on the contrary, exhibits a much more variable IPG, since it tunes it to achieve the desired bitrate.

Skype varies the bitrate by modifying the message size $L$, as Fig. 3 shows. Indeed, messages double their size during the initial trace portion with respect to the last portion. In the case of SVOPC, which is tailored for frame-erasure channels,

messages unlikely carry old replicas of previous blocks (i.e., $RF$ =1). Instead, given that SVOPC $IPG$ is higher in the initial phase with respect to the other Codecs, Skype conservatively tries to reduce the packetization overhead by bundling together several voice blocks and delaying their transmission. This is coherent with the increased network-friendliness early noticed concerning the SVOPC offered bitrate during the initial probing phase. During the central transient phase Skype applies reframing to include more than one block in the same message ($RF = 2$) but not to all the blocks, so that a mix of double- and single-sized messages is present. Notice that VBR Codecs, such as ISAC and iPCM-wb, exhibit larger message size variance, while CBR Codecs (e.g., G.729, iLBC and PCM) generate almost constant size messages: in this case, the small but noticeable variability is due to report blocks piggybacked into the same message. Notice also that during the transient period, the bitrate exhibits a smooth decrease, whereas only two message sizes are possible. This means that Skype controls $RF$, so as to shape the resulting bitrate. Finally, note that at the very beginning $L$ is larger, being the initial framing $\Delta T$ larger too.

We now consider the case of a voice flow transported by TCP. We use the same testbed scenario previously described and repeat all experiments presented above, after having imposed TCP as the transport protocol by means of a firewall rule. The results are presented in Fig. 4 for the ISAC Codec only. When using TCP, Skype sets $RF = 1$ from the beginning of the flow; indeed, since TCP recovers packet losses, there is no need for setting $RF$ to 2. A couple of additional observations are also worth: first, the SoM header is not present, reducing the message size of 4 bytes, as reflected by the smaller value of $B$; second, $\Delta T$ is still variable, as shown at the initial portion of the trace.

Interestingly, TCP congestion control and segmentation algorithms do not alter $L$ and $IPG$. This is due to the fact the during the test, no loss was present, so that the TCP congestion window was unbounded. We also suspect that Skype uses the `TCP_NODELAY` socket option to disable Nagles's algorithm, so that the time delay between messages is maintained.

### *B. Video Flows Characterization*

In order to analyze the traffic generated by video flows, we repeated the experiments of the previous section, enabling the video source after about 5 s. Voice Codec is left to the default ISAC choice and UDP is used as transport protocol; neither artificial delay nor loss are imposed.

Results are presented in Fig. 5. The variability of the bitrate (top plot) significantly increases with respect to the case of voice flows, ranging from a few kbps up to 800 kbps. The $IPG$ (middle plot) is less regular than in the voice-only case. Indeed, a large number of $IPG$ samples is about 30 ms (the preferred ISAC $\Delta T$), while many other $IPG$ samples are very small. This is due to the fact that Skype is multiplexing voice and video blocks: the first ones are produced by the corresponding voice Codec at a very regular rate; the latter ones are instead bigger, have a larger frame size and are chopped and transmitted using multiple back-to-back messages. This is reflected by $L$ plot at the bottom

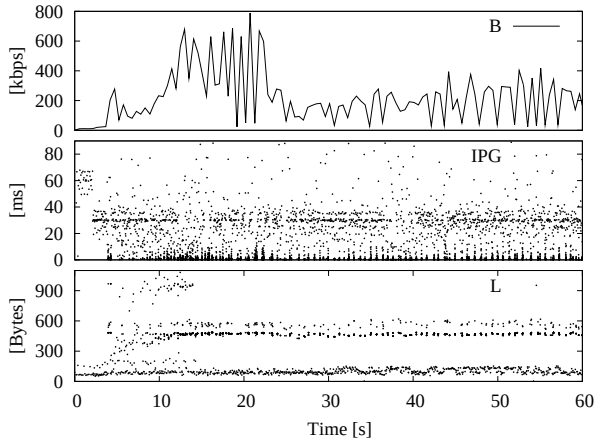


Fig. 5. $B$, $IPG$ and $L$ traces versus time for a videocall - UDP at the transport layer, no artificial loss, ISAC Codec.

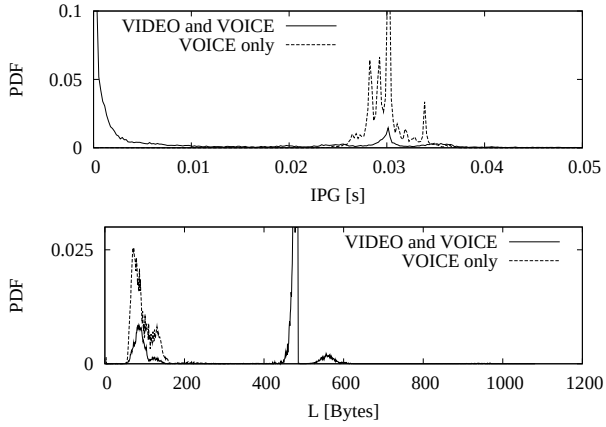


Fig. 6. $L$ and $IPG$ PDFs for voice only or video and voice streams - UDP at the transport layer, no artificial loss, ISAC Codec.

of Fig. 5. Let us first focus on the period $t > 20$ s, when $RF = 1$. It is possible to identify three typical message sizes: i) $L \in [0, 150]$ Bytes, for messages containing a voice block only , ii) $L \in [350, 490]$ Bytes, for messages containing a video block only, and iii) $L \in [491, 500]$ Bytes when a voice and a video blocks are multiplexed in a single message. The message size doubles if $RF = 2$, e.g., when $t \in [5, 15]$ s, as also shown by the probability density functions (PDFs) of $L$ and $IPG$ of a voice only and video plus voice flows reported in Fig. 6.

### *C. Impact of Different Network Conditions*

Let us now investigate the impact on the traffic generated by Skype of different network conditions, namely: i) available end-to-end bandwidth, ii) loss probability, and iii) source-destination path delay.

Figs. 7 and 8 report measurements obtained during a voice call during which we artificially limited the available bandwidth between the two clients, for the ISAC and SVOPC Codecs, respectively. Top plot reports $B$ and the imposed bandwidth limit; middle and bottom plots report $IPG$ and $L$, respectively. Let us consider the ISAC case first. The usual 20 s long initial period is present with $RF = 2$. When the available

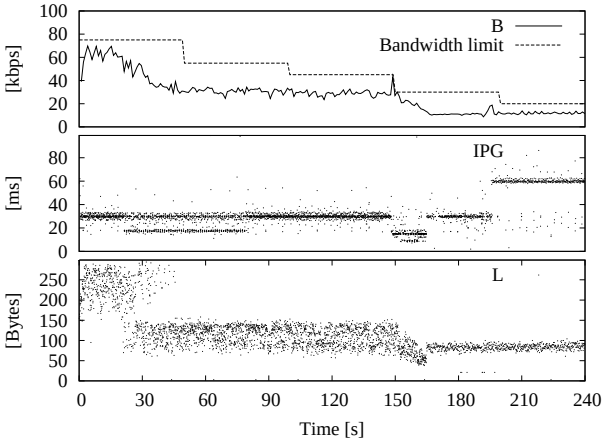


Fig. 7. $B$, $IPG$ and $L$ during a voice call under decreasing available bandwidth - UDP at the transport layer, ISAC Codec.

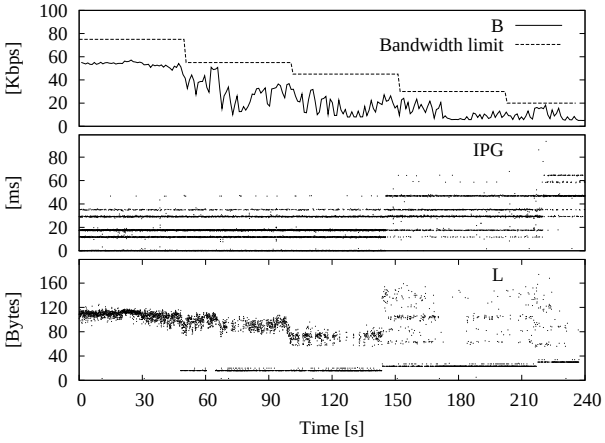


Fig. 8. $B$, $IPG$ and $L$ during a voice call under decreasing available bandwidth - UDP at the transport layer, SVOPC Codec.

bandwidth is larger than the actual bitrate, no changes are observed with respect to the typical source behavior shown in Fig. 1. As soon as the available bandwidth limit kicks in (after about 150 s), the source adapts $B$ to the new constraints: $L$ takes smaller values, which suggests that the Codec selects a low-bitrate state (recall that the ISAC Codec is a VBR Codec), and $IPG$ changes to 20, 30 or 60 ms, hinting that the Skype framer modifies the framing time to reduce the protocol overhead. We can then state that Skype implements a congestion control protocol that acts on the $RF$, $\Delta T$ and Codec bitrate. When using the SVOPC Codec, whose results are depicted in Fig. 8, Skype seems to implement more complex mechanisms to adapt to network conditions: different values of $IPG$ are used, under both good and bad network conditions; $L$ can grow to significantly larger values than the with the other Codecs, especially under tight bandwidth limits.

We perform a second set of experiments to assess the impact of network packet losses. Fig. 9 plots the message size $B$ observed during a voice call when artificial packet losses are introduced. In particular, periods with no losses alternate to periods during which 5% or 10% loss probability is enforced. Results considering a UDP-E2E and TCP-E2E

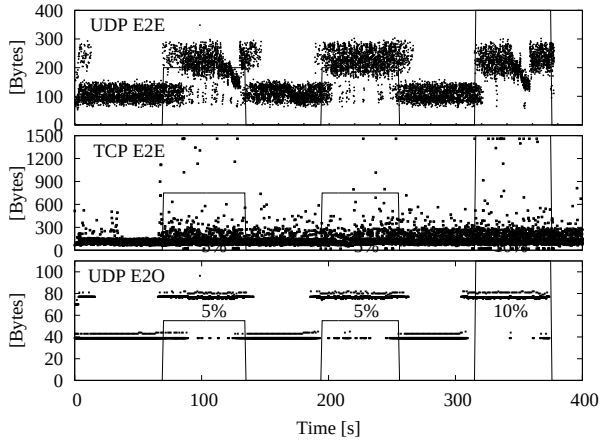


Fig. 9. $L$ during voice calls under on-off artificial losses – UDP and TCP E2E calls (ISAC Codec) and UDP E2O call (G729 Codec).

flow (VBR ISAC Codec), UDP-E2O flow (CBR G.729 Codec) are reported. In the UDP case, when some losses are detected, Skype greedily compensates them by retransmitting past voice blocks into the same message, i.e., setting $RF = 2$; on the contrary, when no loss is detected, Skype sets $RF = 1$. This holds for both E2E and E2O, and for both voice and video calls (the latter E2E video case is not reported for the sake of brevity). Conversely, if TCP is adopted, no loss concealment mechanism is implemented by Skype, which completely relies on TCP loss recovery mechanism. This results in a much more complex $L$ pattern, since TCP congestion control and segmentation algorithms impose a different framing pattern to the application stream. For example, if a loss is recovered after that the retransmission timeout expiration, data buffered at the socket are immediately sent by TCP in one (or more) larger segments.

We now consider a UDP voice call facing increasing average loss probability from 0% to 10% with 1% step increments every 45 s. Measurements for ISAC and SVOPC Codecs are reported in Figs. 10 and 11, respectively. In the ISAC case, Skype uses $RF \geq 1$ as soon as loss probability exceeds 1% and the relative occurrence of $RF$ values changes as a function of the loss rate: the vast majority of messages use $RF = 1$ until losses exceed 4%, in which case $RF = 2$ is used with few exceptions. If no loss is detected (e.g., at the end of the trace), $RF$ is set to 1 again. For the SVOPC Codec, packet size exhibits a higher variability, especially when loss probability is large; in these cases Skype seems to try different combinations of packet size and $RF$ value to deal with the occurrence of a large number of losses.

Some tests were also performed to assess the impact of network delay: no change was observed and, thus, no results are reported. This is quite intuitive, since there is no major countermeasure that a real-time application can implement if the end-to-end delay is large due to physical constraints such as distance.

Comparing the previous discussed Skype reactions to network conditions, we can conclude that, when using UDP at the transport layer, Skype measures loss probability and

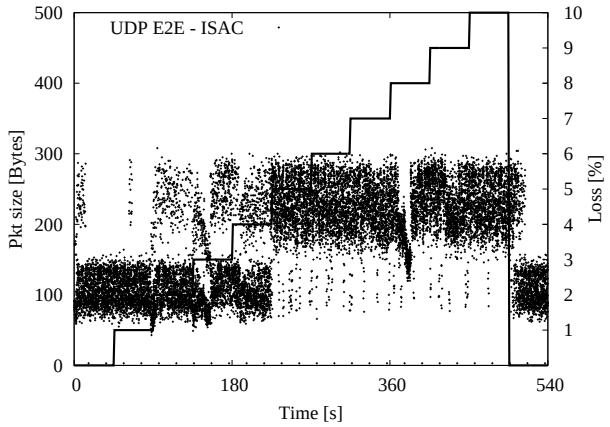


Fig. 10. $L$ during a voice call under increasing artificial losses - UDP at the transport layer, ISAC Codec.

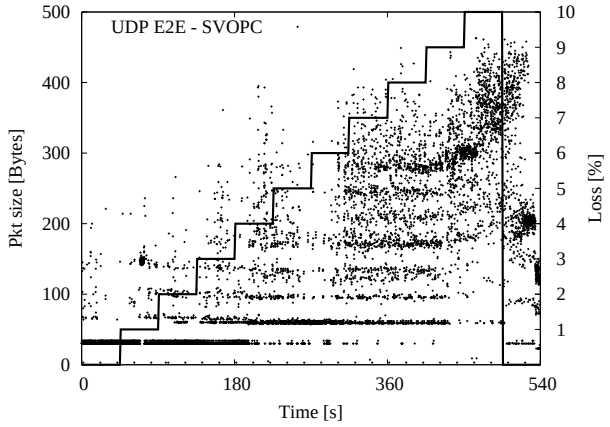


Fig. 11. $L$ during a voice call under increasing artificial losses - UDP at the transport layer, SVOPC Codec.

implements techniques to measure the available bandwidth, so as to effectively react to changing network conditions by either tuning the bitrate or introducing higher redundancy. In the scenario of Fig. 7, through the probing phase after $t = 150$, Skype determines that the low call quality is due to network congestion rather than to path losses, and thus sets $RF = 1$ to avoid overloading the network. Conversely, Fig. 9 shows that some probing phases occur during the time intervals where losses are present and Skype can effectively distinguish that low call quality is due to path losses rather than to network congestion, and therefore sets $RF = 2$ in the attempt to ameliorate call quality.

## IV. User Characterization

In this section we analyze some characteristics of Skype users' behavior, such as the typical service usage workload and the users churning rate. We report results that were collected by passive monitoring Politecnico di Torino campus access link through our classification framework [3]. The classification tool is based on a combination of two different and complementary techniques, namely Naive Bayes classification and Chi-Square statistical test: the first classifier aims at detecting VoIP traffic characteristics, the second one reveals Skype fingerprint from the packet framing structure.

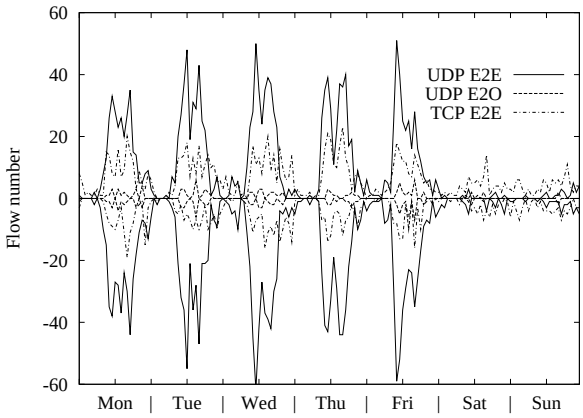


Fig. 12. Number of UDP E2E, TCP E2E and UDP E2O voice calls every 1 hour. Incoming flows on positive values, outgoing flows on negative values.

We monitored the campus access link for more than a month starting from April the 22nd 2007. More than 7000 different hosts are present in the campus LAN, which is used by both students and staff members. The total number of flows that were identified are 17595, 9136, 1393 and 1145 considering UDP E2E, TCP E2E, UDP E2O voice and UDP video calls, respectively. Notice that most of the calls are "free" E2E voice calls, with video enabled in only 6% of cases.

### A. Service usage workload

Fig. 12 reports the number of calls per hour in a typical week, showing outgoing flows (source IP address belonging to the campus LAN, destination IP address not belonging to it) with positive values, and incoming flows with negative values. UDP, that is adopted in 68% of cases, is the preferred transport protocol. Notice that this can dramatically change in a different network setup, e.g., when NAT or firewall are extensively used. As expected, the number of calls is larger during working hours, with a negative bump during lunch time, while during nights and weekends fewer calls are present. The total peak number of calls accounts about 75 Skype calls per hour. Asymmetry is due to the fact that the two directions of the same call can use different transport layer protocols: this happens in about 15% of the cases. Specifically, our campus is more likely to accept UDP connections, whereas for other users in more restrictive network settings Skype is forced to rely on TCP, as can be gathered by the smaller number of UDP E2E incoming flows with respect to the outgoing ones.

### B. Call destination and duration

Fig. 13 reports the call endpoint geolocation, i.e., the location of external IP addresses considering voice flows. We queried the geographical location of the IP addresses using HostIP [16], a public, open and free IP address location database. More than 54% of the E2E voice call endpoints are in Italy, 27% are located in Europe, being UK, France,

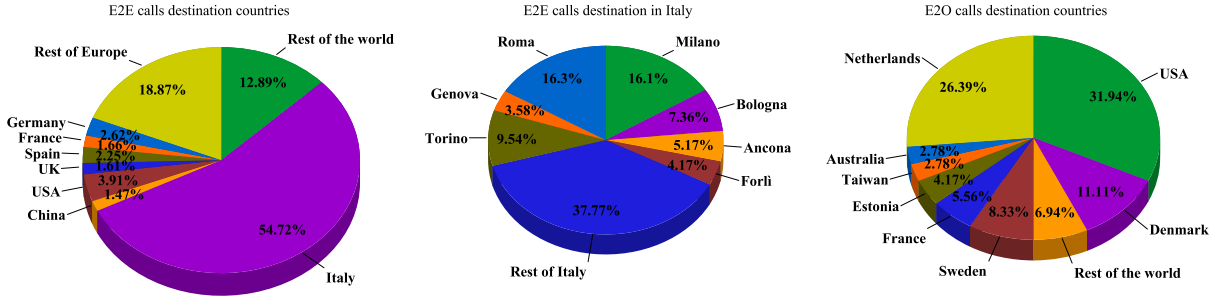


Fig. 13. Geolocalization of peers making E2E and E2O voice calls.

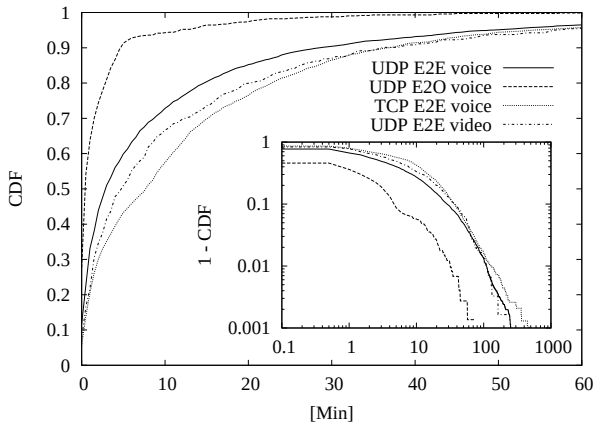


Fig. 14. Call holding time CDF and 1-CDF in the inset.

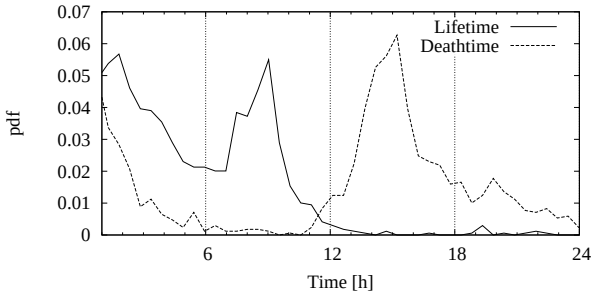


Fig. 15. Peer life-time and death-time PDF.

Spain and Germany the top four destinations. In Italy, destinations are distributed similar to the population, mainly in the largest cities; about 18% of calls for the rest of the world are terminated outside the EU, and only less than 4% are terminated within the US. This picture changes dramatically considering E2O calls. Recalls that E2O calls are subject to a (low) connection fee. The right pie shows that the E2O service is competitive with traditional phone services only when international calls are considered. In this case, about 32% of E2O calls are directed to the US, while the Netherlands accounts for more than 26%, being Denmark the third preferred endpoint. We suspect that connections directed to Countries in which no local Skype gateway is present are terminated either in the Netherlands or in Denmark, where the Skype headquarters are located.

Fig. 14 reports the Cumulative Distribution Function (CDF) of call holding time (i.e., the call duration), defined as the time elapsed from the first to the last packet of the flow. It can be noted that E2E calls last much more than E2O calls, probably because they are free. Interestingly, the measured holding time is slightly larger when the video is enabled.

The larger TCP E2E holding time is at first surprising, since there is no reason for the user to talk more when TCP is adopted. Investigating further, we noticed that Skype delays the TCP tear-down sequence, keeping the connection alive even if the call has been hung up. This affects resource usage on both end hosts and the possible full-state NAT, since the TCP connection must be managed until the tear-down sequence is completed.

### C. Peer life-time and death-time

One of the parameters that affect P2P systems in general is the churning rate, i.e., the peer arrival and departure process that forces the P2P overlay to be updated. In order to understand the churning process in the Skype network, we focus on the peer activity cycle, measuring peers *life-time* (the duration of peers' activity period) and *death-time* (the duration of peers' idle periods). A peer is considered to be idle (or dead) if no packet is sent for a period of time longer that an idle time $\gamma$, otherwise the peer is considered to be alive. Thus, a life-time sample is measured from the instant in which an idle peer generates the first packet until an idle time $\gamma$ is detected. A death-time, on the contrary, is the time interval between the instant in which a peer becomes inactive until the instant in which it generates a new packet again. We experimentally verified that any value of $\gamma$ larger that 200s have minimal impact on the lifetime measurements, thus, we conservatively selected $\gamma = 500s$.

Fig. 15 reports the PDF of peers' lifetime and death-time measured during a week long observation period. Peers' lifetime is either short (1 or two hours) or very long (from 7 to 10 hours): overall, about 95% of peers disappear after 10h of activity, more than 1% of the peer were always alive during the whole week. Considering peer's death-time, we observe, on the contrary, that the death period is either shorter than 2 hours, or larger than 11 hours, with about 2% of peers idle for more than 72 hours.

The above results allow us to refine the picture of the Skype usage pattern in our scenario, that is the typical campus with activity during working hours and users with a high degree of familiarity to communication technologies. We isolate and quantify two well-defined user behaviors in the Politecnico di Torino LAN: namely, Skype *occasional* and *regular* users. Occasional users run Skype only when they actually need to make a call, and quit the application shortly after call completion. Their lifetime is proportional to the call holding time, whereas the death-time depends on the frequency of their calls – being Skype possibly dead for several days. On the contrary, regular users typically run Skype by default, so that the peer lifetime follows their PC lifetime: Skype software is on during the daytime and off outside office hours and during nights. Some PCs are left running during the night as well, so that the Skype life-time is extremely long. For regular users accessing Skype application through laptops, the activity cycle is much faster as such users can possibly turn off their laptops to save battery energy; these users are partially responsible for short death-times.

We stress that the above user behaviors have been observed in a single network setup and may not be valid at a more general extent: e.g., other behaviors could emerge when considering residential or business Skype usage. At the same time, the usage pattern described above suggests that Skype churning rate is very low: we can therefore expect the P2P overlay maintenance and update rate to be limited.

## V. Signaling Characterization

In this section, we focus on signaling traffic generated by Skype, dissecting several interesting aspects, such as signaling overhead, peer geolocalization, Skype overlay selection process.

### A. On the Signaling Overhead

We first consider the overhead that Skype signaling introduces in the network. The average signaling bitrate, evaluated as the total signaling bits transmitted by a client during its whole lifetime, is very low: it is less than 100 bps in 95% of cases (and less than 10 bps in 50% of cases), while only very few nodes, that are possibly supernodes, use more than 1 kbps for signaling.

Since the signaling bitrate is exiguous, its relative importance vanishes if weighted on the ground of VoIP call traffic: for about 5% of the Skype clients, signaling accounts only for 5% of the total (including voice and video calls) Skype traffic. At the same time, since clients may be left running for long periods without VoIP services being actively accessed, the signaling traffic portion is dominating in 80% of the cases.

Let $C$ be the number of different peers contacted by a given peer in a 5 minutes long interval. The CDF of $C$, reported in Fig. 16, shows that a peer contacts about 16 other peers on average, and no more than 30 in 90% of cases. Still, $C$ can grow larger than 75 in 1% of the cases, which may constitute a burden for some layer-4 devices that keep per flow state (e.g., a entry in a NAT/ACL tables). Moreover, many signaling flows are single-packet flows that create new temporary soft-state entries, rarely used later on.

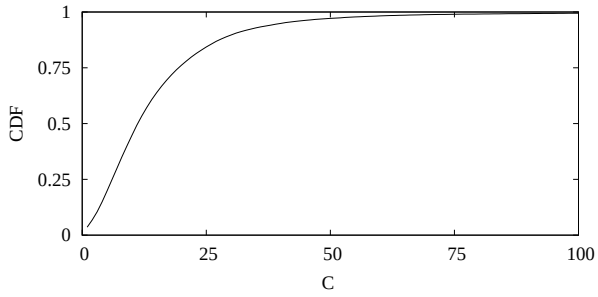


Fig. 16. Distribution of the number of different peers that are contacted by a peer during 5 minute long intervals.

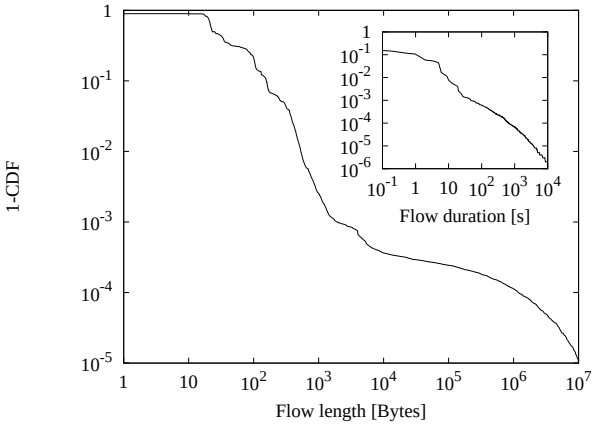


Fig. 17. Distribution of the signaling flow size (outset) and duration (inset).

### B. Signaling Flow Classification

We are now interested in observing the signaling traffic a Skype client exchanges. The *semantic* of the signaling activity cannot be inferred from purely passive measurement, but the *form* of signaling activity can be further differentiated. Let us observe the amount of data sent by the *source* (in packets) and its corresponding duration (in seconds). The complementary distribution functions (1-CDF) are reported in Fig. 17 using a log/log scale. About 80% of the signaling flows consists of single packet probes, and 99% of the flows is 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 for hours, as the tails of the curves in Fig. 17 show: indeed, the single-packet probes account for less than 5% of the total bytes.

Consider now the schematic representation of the typical Skype signaling activity depicted in Fig. 18. We select two peers, namely the most active peer $p_1$ that does not perform any voice call (left plot) and a randomly picked peer $p_2$, having both signaling and voice flows (right plot). Each dot in the picture corresponds to a packet in the trace: the x-axis represents the packet arrival time since the first packet observed for that client; the y-axis reports an ID that uniquely identifies a peer that exchanged a packet with peer $p$. Positive IDs are used for peers that received a packet from $p$, negative IDs for peers that sent a packet to $p$. The range of the y-values corresponds to the number of different peers with whom the

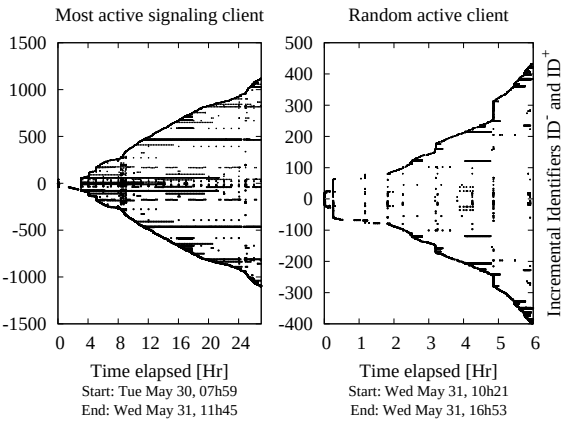


Fig. 18. Pictorial representation of Skype signaling activity for a given client: each dot represents a packet exchanged in a given time by a client with some other peer whose ID is reported on the y-axis; positive (negative) IDs represent peers that has sent (received) a packet to (from) the client.

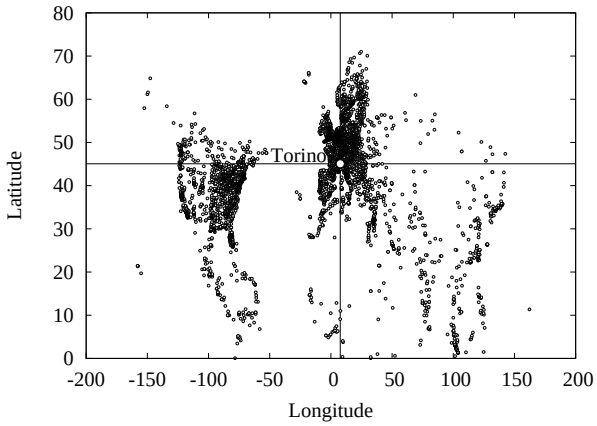


Fig. 19. Geolocation or peers contacted by internal peers.

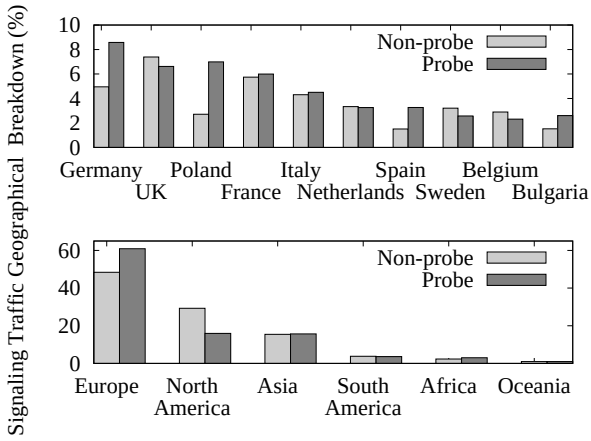


Fig. 20. Geographical breakdown of probe and non-probe signaling traffic, considering all continents (top) and the ten most active European countries.

selected peer is exchanging packets. The figure shows that $p_1$ contacted (was contacted by) about 1100 other peers within its whole lifetime (about 27h), whereas $p_2$ by about 450 in 6 hours.

From the figure we can make three observations. First, the number of contacted peers exhibits an almost linear growth over time, hinting to P2P network discovery being performed during most of the peer life-time. This part of the signaling activity is mainly carried out by the transmission of a single packet, to which (most of the times) some kind of acknowledgment follows. The fact that $p$ knows the address and port of valid (but previously un-contacted) Skype peers means that the above information is exchanged through some signaling messages. 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 symmetric. Second, some of the peers are contacted on a regular basis: in the activity plot, horizontal patterns state that the same peer is periodically contacted during $p$ lifetime. Finally, a periodic information refreshment can be distinguished in the form of vertical patterns (clearly visible at about every hour).

These observations suggest the existence of two types of signaling flows, which we classify as:

- **Probe**: any packet sent toward an unknown peer, to which a single reply packet possibly follows, but *no further packet* is exchanged between the peer pair;
- **Non-Probe**: any flow constituted by more than one packet, including periodically exchanged probes.

In Fig. 18 non-probe traffic is represented by dots inside the triangular shape; the periodic information refreshment, responsible for the vertical patterns, involves both non-probe and probe traffic toward new peers.

Considering the type of flows two peers exchange, in 50% of cases, a probe flow is exchanged; in 15% of cases, two peers exchange only periodical packets, in the remaining cases a variable packet exchange activity is observed. These results confirm that probe and non-probe traffic correspond to different kinds of signaling activity (possibly network discovery and network maintenance).

### C. On the Geolocation of Peers

We now consider the geographical location of contacted peers. In the dataset we consider, we observed 304,690 external peers, corresponding to 263,886 different IP addresses. `HostIP` was used again to perform the geolocation of IP addresses. Fig. 19 reports results for the subset of about 10k peers (out of the about 264k queries) for which longitude and latitude information were available. From the picture, it is easy to recognize the shape of continents, especially Europe and North America. A white landmark helps in locating Torino, that is our vantage point.

Further details on the geolocation of the whole Skype peer dataset is given in Fig. 20, which reports a breakdown, considering probe and non-probe flows per continent (bottom) and per European Country (top). The breakdown is limited to top 10 groups, ranking them by decreasing level of preference.

Two considerations that can be drawn. First, probing mechanism tends to privilege nearby hosts: indeed, 60% of the probed IPs are located in Europe, four times as much as in North America (15%). This suggests that the probing mechanism tends to discover network hosts that are geographically close. Second,the opposite occurs for non-probe traffic: while

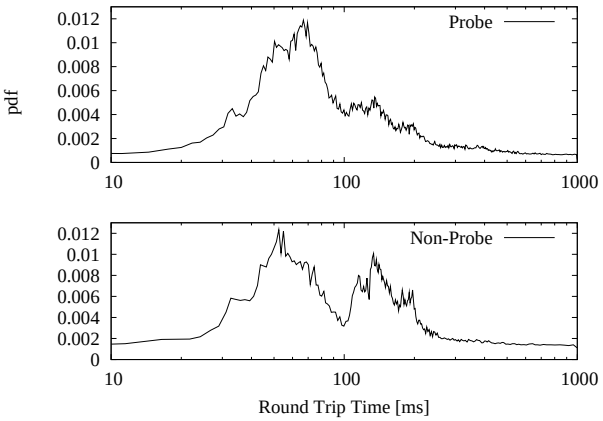


Fig. 21. Probe and non-probe traffic round trip time distribution.

the percentage of peers that are located in Europe actually decreases (48%) with respect to probe traffic, the percentage of North American peers nearly doubles (29%). Considering that users resort to Skype to decrease communication fees and to keep contacts with faraway users, we are not surprised that non-probe traffic is more spread out. Indeed, the relationship among users forces Skype peer selection when considering non-probe traffic. On the contrary, the peer discovery mechanisms implemented by the probes is driven by the physical properties of the underlying network.

### D. On the Peer Selection Criterion

Fig. 21 shows the distribution of the Round Trip Time (RTT) between two peers, measured as the time between the packet probe going out of the campus LAN and the probe response packet (if any). For non-probe traffic, the first sent-received packet pair is used to estimate the RTT. This measurement takes into account both the network and application latency.

Results confirm our previous intuition: the latency of probing traffic is lower than that of other traffic. From Torino, RTT smaller than 100ms are typical of nodes within the Europe, while RTT larger than 100ms are typical of nodes outside it. Measurement results allow us to conjecture that the probing mechanism is *latency driven*: Skype client probes peers based on the information received by other peers so that low latency peers are more likely selected than high latency ones. Conversely, the peer selection mechanism is *preference driven*, where the preference criterion is dependent on the user relationships with other users.

## VI. Conclusions

This paper focused on the characterization of Skype traffic, the most popular VoIP application nowadays. Our contribution is twofold. First, from extensive testbed experiments we investigated several aspects of the Skype source, considering different service types (i.e., SkypeOut, End2End voice and video calls), transport protocols (i.e., TCP, UDP), and network conditions (i.e., loss rate and available bandwidth). Testbed measurements refined the picture on the Skype source model, enlightening the mechanisms and triggering conditions that Skype uses to adapt to network conditions: specifically, when UDP is used at the transport layer, Skype distinguishes and differently reacts to path losses and network congestion. Second, by leveraging on a consolidated methodology for fine-grained Skype traffic classification, we investigated both i) Skype users' behavior and the traffic generated during voice and video communications, and ii) the signaling traffic generated by Skype. Concerning signaling, we have shown that Skype floods the network with short single-probe messages toward many hosts – which may be as effective for the purpose of the overlay maintenance as costly from the viewpoint of statefull layer-4 network devices.

## VII. Acknowledgments

This work was supported by the Italian Ministry of University, Education and Research (MIUR) under the PRIN RECIPE, and partly by a research contract with Vodafone Italia. We would like to thank our Campus LAN Network Administrators for allowing us to monitor traffic.

## References

[1] Skype web site, http://www.skype.com
[2] "International carriers' traffic grows despite Skype popularity", TeleGeography Report and Database. available on line http://www.telegeography.com/, Dec. 2006.
[3] D.Bonfiglio, M.Mellia, M. Meo, D.Rossi, P.Tofanelli, "Revealing Skype Traffic: when randomness plays with you", *ACM Sigcomm'07*, Kyoto, Japan, Aug. 2006.
[4] P. Biondi, F. Desclaux, "Silver Needle in the Skype." *Black Hat Europe'06*, Amsterdam, the Netherlands, Mar. 2006.
[5] S. A., Baset, H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol." *IEEE Infocom'06*, Barcelona, Spain, Apr. 2006.
[6] D.Bonfiglio, M.Mellia, M. Meo, D.Rossi "Tracking Down Skype Traffic", *IEEE Infocom'08*, Phoenix, AZ, Apr. 2008.
[7] K. Suh, D. R. Figueiredo, J. Kurose, D. Towsley, "Characterizing and detecting relayed traffic: A case study using Skype.", *IEEE Infocom'06*, Barcelona, Spain, Apr. 2006.
[8] S. Guha, N. Daswani and R. Jain, "An Experimental Study of the Skype Peer-to-Peer VoIP System", *5th International Workshop on Peer-to-Peer Systems*, Santa Barbara, CA, Feb. 2006.
[9] K. Ta Chen, C. Y. Huang, P. Huang, C. L. Lei "Quantifying Skype User Satisfaction", *ACM Sigcomm'06*, Pisa, Italy, Sep. 2006.
[10] L. De Cicco, S. Mascolo, V. Palmisano "Skype Video Responsiveness to Bandwidth Variations," *ACM NOSSDAV'08*, Braunschweig, Germany, May, 2008
[11] GlobalIPSound web site, http://www.globalipsound.com
[12] J. Lindblom, "A Sinusoidal Voice Over Packet Coder Tailored for the Frame-Erasure Channel" *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 5, Sep. 2005.
[13] On2 web site, http://www.on2.com
[14] M. Carson, D. Santay, "NIST Net: a Linux-based network emulation tool." *ACM SIGCOMM Computer Communication Review*, V.33, N.3, July 2003, pp:111-126.
[15] Skype testbed traces, available at http://tstat.tlc.polito.it/traces-skype.shtml
[16] http://www.hostip.info