

Evidences Behind Skype Outage

*Original*

Evidences Behind Skype Outage / Rossi, D; Mellia, Marco; Meo, Michela. - STAMPA. - (2009), pp. 1-6. (Intervento presentato al convegno Communications, 2009. ICC '09. IEEE International Conference on tenutosi a Dresden, Germany nel 15 June 2009) [10.1109/ICC.2009.5199012].

*Availability:*

This version is available at: 11583/1953991 since:

*Publisher:*

IEEE

*Published*

DOI:10.1109/ICC.2009.5199012

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# Evidences Behind Skype Outage

Dario Rossi<sup>1</sup>, Marco Mellia<sup>2</sup>, Michela Meo<sup>2</sup>

<sup>1</sup>*TELECOM ParisTech, France*  
dario.rossi@enst.fr

<sup>2</sup>*Politecnico di Torino, Italy*  
firstname.lastname@polito.it

**Abstract**—Skype is one of the most successful VoIP application in the current Internet spectrum. One of the most peculiar characteristics of Skype is that it relies on a P2P infrastructure for the exchange of signaling information amongst active peers. During August 2007, an unexpected outage hit the Skype overlay, yielding to a service blackout that lasted for more than two days: this paper aims at throwing light to this event.

Leveraging on the use of an accurate Skype classification engine, we carry on an experimental study of Skype signaling during the outage. In particular, we focus on the signaling traffic *before, during* and *after* the outage, in the attempt to quantify interesting properties of the event. While it is very difficult to gather clear insights concerning the root causes of the breakdown itself, the collected measurement allow nevertheless to quantify several interesting aspects of the outage: for instance, measurements show that the outage caused, on average, a 3-fold increase of signaling traffic and a 10-fold increase of number of contacted peers, topping to more than 11 million connections for the most active node in our network – which immediately gives the feeling of the extent of the phenomenon.

## 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 accounts for more than 4.4% of total VoIP traffic [2] and recently reached over 170 millions of users, about 10 millions of which are online at the same time.

One of the most interesting Skype peculiarities is that it relies on a P2P infrastructure to support the control and management functions, which makes the system highly scalable, avoiding single points of failure. A first key to Skype success is its business model, which offers a large spectrum of useful and free services in addition to optional features that can be purchased at competitive rates. Another key to its success is due to user-friendliness, which for many users translated into the ease of configuration. A third characteristic stems from its robustness: although running over the uncontrolled Internet, Skype has been very reliable, almost like a PSTN network.

However, despite the Skype overlay has been fully functional 24/7 during the last years, Skype suffered an unexpected outage during the summer of 2007. Quoting the official company blog [3], “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 re-booted after receiving a set of patches through Windows Update.” By exploiting our previous work in which we devised a methodology that successfully tackles the problem of Skype traffic identification [4], this work aims at contributing to the understanding of the Skype outage by providing several insights into Skype signaling traffic. To gather a more comprehensive view of the phenomenon, we compare it with “normal” signaling patterns, that were observed before and after the outage.

To the best of our knowledge, this work is the first to explore *an extreme event such as the disruption of an Internet scale overlay*. While it is impossible from pure passive measurement to state ground truth about the *reasons* that lead to the outage, we can still manage to precisely quantify some interesting properties of the signaling traffic during the outage. For example, on average we observe a 4-fold increase of the number of signaling flows, a 3-fold increase of the packet sending rate and a 10-fold increase of the number of contacted peers. At the same time, the most active peer in our network experienced a 10-fold increase of traffic amount and a 30-fold increase of number of contacted peers, topping to more than 11 million of signaling connections during the whole 3-days outage period – behaving thus as a lightning rod in a storm.

In addition, while the outage is said [3] to be related to the distribution of a Windows update that forces a large number of computers to reboot during the same period of time, our investigation of the geolocation of the contacted peers does not enlighten an evident synchronization in the occurrence of anomalous peer behaviors. This possibly stems from the asynchronous reboot events, due to delay in patch download and installation. Finally, our measurement show that the outage event actually entailed some changes in the Skype overlay maintenance after the outage: for example, an interesting change in the geographical location of the peers can be observed, which results in an increase of its smoothness, stability and geographical clustering of the overlay.

## II. SKYPE SIGNALING PRIMER

Skype offers end users several free 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 services, in which case a fee is applied.

The main difference between most VoIP services and Skype is that, except for user’s authentication which is performed under a classical client-server architecture, the latter operates on a P2P model. After the user (and the client) authentication, all further signaling is performed on the P2P network, so that Skype user’s informations (e.g. contact list and status, preferences, etc.) are entirely decentralized and distributed among P2P nodes. This allows the service to scale very easily to large sizes, avoiding a costly centralized infrastructure.

From a protocol perspective, Skype uses a proprietary solution which is difficult to reverse engineer due to extensive use of both cryptography and obfuscation techniques. Though Skype may rely on either TCP or UDP at the transport layer, both signaling and communication data are preferentially carried over UDP, although when a UDP communication is impossible, Skype falls back to TCP. As a single random port is selected during application installation (and it is never changed, unless forced by the user), we introduce the following definitions:

- A Skype *client* is identified by the (host IP address, Skype UDP/TCP port) pair.
- A Skype *flow* is identified by using the traditional tuple (IP source and destination addresses, UDP/TCP source and destination ports, IP protocol type), in which at least one endpoint is known to be a Skype client.

A flow starts when a packet with a new flow tuple is first observed, while it is ended by either *inactivity timeout*<sup>1</sup> or, in case of TCP, by observing the tear-down sequence if present. In this work, we focus on signaling flows carried over UDP only, which constitute the largest and most interesting part of Skype signaling traffic. Skype clients are identified as in [4], so that it is possible to identify all flows that are either originated by or directed to a Skype client.

Let us start the investigation of Skype signaling traffic by depicting in Fig. 1 a few patterns that are representative of a peer activity. We select two specific peers from our traces, namely the most active peer (peer A, top plots) and a randomly chosen active peer (peer B, bottom plots), inspecting half an hour of their typical behavior before (left plots) and during (right plots) the outage. Let  $p$  be the peer under observation. Each dot in the picture corresponds to a packet in the trace: the x-axis represents the packet arrival time since  $p$ ’s first packet is observed, and the y-axis reports the ID of the peer that exchanged the packet with  $p$ : positive IDs are used for peers that received a message from  $p$ , negative IDs for peers that sent a message to  $p$ . The y-range corresponds thus to the number of different Skype peers whom  $p$  is exchanging messages with.

Several interesting remarks can be gathered from Fig. 1: indeed, though the *semantic* of the signaling activity cannot

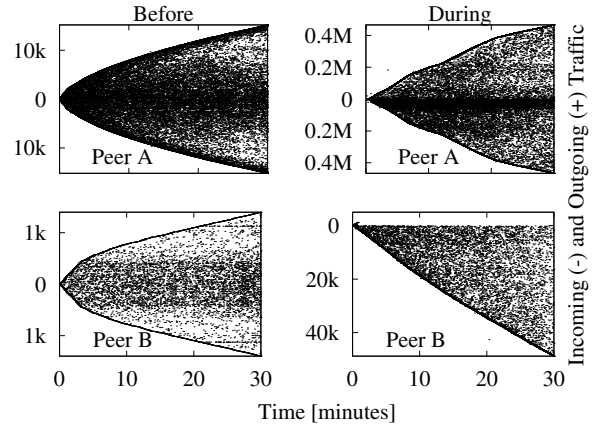


Fig. 1. Pictorial representation of Skype signaling activity

be inferred from purely passive measurement, its *form* can be further differentiated [11]. A linear growth of the number of contacted peers is noticeable: this hints to P2P network discovery being carried on during most of the peer lifetime. Interestingly, this part of the signaling activity is mainly carried out by the transmission of a packet, to which (most of the times) some kind of acknowledgment follows. Also, the fact that  $p$  knows the address and port of valid (but previously un-contacted) Skype peers means that the above information is carried by signaling messages.

Still, some of the peers are contacted on a regular basis. In the activity plot, any point lying in the zone below the “discovery line” states that a peer is contacted several times during  $p$  lifetime. This suggests the existence of two very different kinds of the Skype signaling, namely:

- **Probe** traffic, which aims at network discovery: probe flows are made of a single packet sent toward a peer, to which a single reply packet possibly follows, but *no further message* is exchanged between the peers pair.
- **Non-probe** traffic, which aims at the network maintenance, including overlay and contact information management: non-probe flows are either flows longer than one packet or sequence of single-packet probes, separated by a time gap larger than the inactivity timeout.

We now focus on what happened during the outage. The activity pattern during this critical period exhibits a massive (peer A) and possibly unbalanced (peer B) growth over time, which was on the contrary rather symmetrical before the outage. During the outage, peer A *contacts almost half a million different clients in just 30 minutes* (top-right plot), whereas it used to contact about 15 thousand peers in normal operation (top-left plot) – which corresponds to an almost 30-fold increase. Considering peer B pattern (bottom plots), it can be seen that the amount of incoming traffic experienced a roughly 20-fold increase during the outage, but peer B no longer replied, yielding to the asymmetric traffic profile (bottom-right plot). However, notice that peer B was “alive” during that time period, as the host (which we verified not to be behind a NAT) actually had ongoing traffic.

<sup>1</sup>Results reported in this paper are derived setting the timer to 200s: this choice is justified by the fact that the largest inter packet gap within the same flow that was ever observed is 180s which is likely used by Skype as keep-alive message to force the refresh of possible NAT entries.

TABLE I  
FLOW, PACKETS AND BYTES TRAFFIC AMOUNT: UDP, SKYPE, TOP-10 AND TOP-1 SKYPE CLIENTS

	Flows				Packets				IP Bytes			
	UDP	Skype%	Top-10 [%]	Top-1 [%]	UDP	Skype%	Top10%	Top1%	UDP	Skype%	Top-10 [%]	Top-1 [%]
Before	$3.3 \cdot 10^7$	75	75	22	$7.5 \cdot 10^7$	57	71	12	$2.1 \cdot 10^{10}$	23	71	4
During	$1.1 \cdot 10^8$	94	97	73	$1.3 \cdot 10^8$	89	95	67	$9.0 \cdot 10^9$	69	94	50
After	$3.4 \cdot 10^7$	69	94	27	$8.0 \cdot 10^7$	64	62	15	$1.1 \cdot 10^{10}$	51	60	12

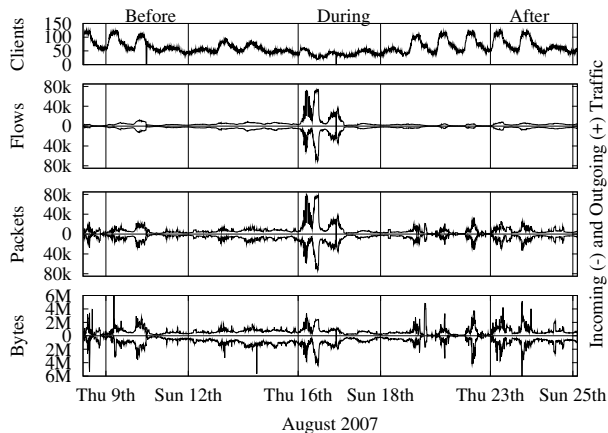


Fig. 2. The amount of Skype flows, packets and bytes for the periods under analysis (1 minute window), considering UDP protocol only

### III. THE SKYPE OUTAGE

In this section, we report results that were collected by passive monitoring the campus access link at Politecnico di Torino, where about 7000 different hosts are regularly used by both students and staff members. It took more than two days before the Skype engineer team managed to get the situation back to normal since the problem was first acknowledged<sup>2</sup> Concerning the outage period, we consider as start time 11AM GMT which corresponds to the time instant at which we begin to observe an anomalous (and massive) increase in the amount of UDP traffic, see Fig. 2. The time at which the Skype engineer team has blogged that the situation was back to normal is instead considered as end time. For reference, we also consider two different time intervals during August 2007, i.e., the period corresponding to one week forward and one week backward with respect to the outage period:

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

#### A. Amount of Signaling Traffic

The amount of traffic that we observed during these periods is depicted in Fig. 2, which reports the number of clients, flows, packets and bytes observed over 1 minute long time windows. First, it has to be noted that the traffic amount is low, which was expected as August is a typical Italian vacation period, with the 15th of August being the typical holiday peak.

<sup>2</sup>See “The words we’ve all been waiting for”, posted August the 18th at 11h00 GMT on [3] and “Problems with Skype login”, posted the August the 16th at 14h02 GMT.

At the same time, the number of internal active Skype clients (top plot) is very similar before and after the outage, which allows us to fairly compare measurements during different weeks. A drastic reduction in the number of Skype clients is observed during the outage event (although a slight decrease was already in progress), to which it corresponds an anomalous increase of UDP traffic (in term of flows and packets). During this period *UDP traffic largely outweighs TCP traffic, so that it accounts for almost all incoming traffic*. Before and after the Skype outage, UDP traffic amount is normally much less than the amount of TCP traffic typically observed. However, after the Skype outage, the amount of Skype UDP traffic is still little larger than before the anomalous event: this hints to a possible more aggressive settings of the Skype protocol parameters, in order to more promptly react to disruption of the overlay network and thus to quickly recover from massive distributed failures.

Tab. I reports further details concerning the amount of traffic, expressed in terms of flows, packets and bytes. Some interesting insights can be gathered considering the workload that Skype clients generated during the outage. Table reports the percentage of aggregate Skype traffic relatively to the total UDP traffic (Skype% in the table). In addition, the percentage of Skype signaling traffic that was directed to/generated by the 10% most active (Top10%) or to the 1% most active (Top1%) Skype clients are reported as well. Considering the first row of the table, it can be seen that, before the outage, Skype normally accounted in our campus network for 23% of the IP bytes, carried over 57% of the total UDP packets and 75% of flows. The ten most active clients generated roughly 70% of Skype flows, packets and bytes, while the most active client was responsible for 22% of the flows, 12% of the packets and only 4% of the exchanged bytes.

Conversely, during the outage, Skype traffic accounted for almost the totality (94%) of UDP flows and for a very significant portion of UDP packets and IP bytes (89% and 69% respectively). Furthermore, almost all the traffic was generated/received by the 10 most active clients, with the most active Skype node accounting for the 73% of all flows, 67% of all packets and 50% of all bytes – a clear overload situation. A first lesson is thus that the outage manifested itself *as a communication storm of significant amplitude, with few peers that possibly acted as “lightning rods” during the storm*.

#### B. Type of Signaling Traffic

We now investigate the *type* of signaling during the three periods, reporting in Tab. II a breakdown of the external peers that were contacted with non-probe versus probe traffic,

TABLE II  
BREAKDOWN OF EXTERNAL PEER TYPE

External Peer	Before		During		After	
	No.	%	No.	%	No.	%
Probe	$0.6 \cdot 10^6$	17%	$5.6 \cdot 10^6$	14%	$0.7 \cdot 10^6$	19%
Replied-Probe	$2.5 \cdot 10^6$	69%	$33.4 \cdot 10^6$	83%	$2.3 \cdot 10^6$	67%
Non-Probe	$0.5 \cdot 10^6$	14%	$1.5 \cdot 10^6$	4%	$0.5 \cdot 10^6$	14%
Total peers	$3.6 \cdot 10^6$		$40.5 \cdot 10^6$		$3.5 \cdot 10^6$	
Top-1 peers	$0.4 \cdot 10^3$		$11.3 \cdot 10^6$		$0.3 \cdot 10^3$	

discriminating in the latter case whether probe traffic received a reply. Replied-probes represent the vast majority of the traffic flows exchanged on the Skype overlay both during normal conditions (roughly 70%) and during the anomalous event (83%). Another interesting figure can be gathered concerning the total number of external contacts: during the anomalous event, internal peers exchanged traffic with about 40 millions peers, which is 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 millions of peers, a more than 30-fold increase compared to the normal operation point (300k peers). From the above observations, and given that each peer not only issued an higher number of probes, but also received a reply to the probe in most of the cases, we can conclude that *the meshedness of the overlay likely increased during the outage, and a few peers explored a very large portion of the overlay.*

Finally, notice that during the Skype outage the *relative* importance of non-probe traffic diminishes, as joint effect of (i) smaller number of stable connections and (ii) exploding number of probe connections, which exceeds a 10-fold increase. At the same time, the *absolute* amount of non-probe traffic during the outage also grows by a factor of 3: therefore, we can conclude that *packet probing represents the most important, though not the only, component of the storm.*

### C. Internal Peers in the Storm

We next turn our attention to the *internal peers*, in the attempt of understanding whether specific nodes (e.g., possibly supernodes) have been more heavily concerned by the storm. Since the majority of the traffic is constituted by probes, we focus for the time being on the external probed received by Skype peers in our network. As we have no reliable means to identify Skype super-nodes, we prefer to measure the level of peer signaling before the outage, and to compare peer activity during the outage with their reference level.

More precisely, our aim is to assess whether the amount of traffic increase has been roughly proportional to the “normal” activity that peers carried out before and after the breakdown, or if some peer might have experienced some unexpected and unprecedented activity. On the one hand, we argue that it would be reasonable for the normally most active peers to be contacted by a larger number of external peers (since higher activity also likely translates into higher popularity). At the same time, overlay disruption has been possibly caused by the failure of many of such peers: therefore, it would also be reasonable for peers that appear to be available and

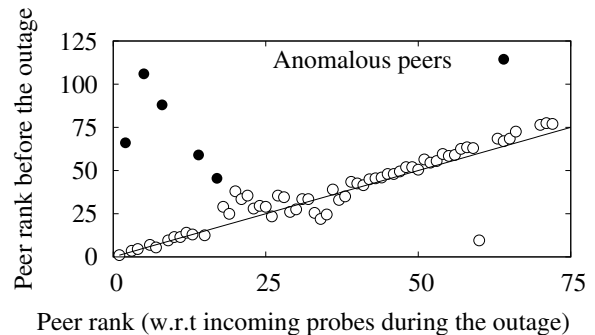


Fig. 3. Peer rank (according to the number of external probes received) during the outage (x-axis) and mean rank before and after the outage (y-axis)

responsive during the outage to experience sudden popularity (in the overlay attempt to react to the failure of highly popular nodes). Our results, as we will show shortly, confirms that both phenomena are indeed present.

Fig. 3 reports a scatter plot of the peer rank  $r$  measured according to the number of external probes received, where the lower the rank the higher the contact (i.e.,  $r = 1$  correspond to the most heavily contacted peer). Each  $(r_D, r_B)$  point in the plot compares the rank of a peer during and before the outage. It can be seen that in most of the cases the rank is roughly maintained (since most points fall around the  $y = x$  line): in other words, peers that were the most contacted before the storm, are also the most contacted during the outage.

At the same time, a few peers, represented in the picture with black bold dots, exhibit an *anomalous* behavior: more precisely, we report a peer behavior as anomalous when its rank ratio exceeds a given threshold  $r_B/r_D > 1.5$ . Such “jumps” in the rank space correspond to peers that were not popular before the outage, but that during the outage received packet probes by an extremely high number of external peers. Most notably, the second (5th) probe target during the outage, receiving about 12% (2%) of all incoming probe packets is a peer which ranked only 68th (107th) before the outage.

Thus, during the outage *the number of incoming probes roughly followed the normal peer popularity, except for a few peers that may nevertheless attract a significant fraction of the incoming probes.*

### D. External Peers in the Storm

Next, we turn our attention to *external* peers. We stress that, since our analysis only involves a single vantage point, our knowledge of external peers behavior is consequently limited: indeed, we can observe only the portion of their traffic which crosses the edge of the campus network (but are not able to observe, e.g., the totality of their probe traffic).

As such, we are forced to focus on longer conversational flows, and limitedly consider those external “Heavy Signalers” (HS) clients that generated at least 25 flows directed to our campus LAN. Tab. III reports several metrics relevant to HS peers, namely: (i) the number of flows generated per HS and directed to our internal clients, (ii) the number of different

TABLE III  
HEAVY SIGNALER, FLOW AND PACKET SIZE STATISTICS

Heavy Signaler (HS)	Before		During		After	
	Mean	p95	Mean	p95	Mean	p95
Flows	106	11k	133	19k	117	11k
Internal Contact	3.8	232	5.7	133	4.4	218
Flows per Internal Contact	60.8	154	37.1	75	64.2	170

Traffic Statistics		Mean	p95	Mean	p95	Mean	p95
Flow Size (packets)	In:	1.68	3	1.08	1	2.06	3
	Out:	1.69	3	1.18	3	2.21	3
Packet Size (Bytes)	In:	86.8	346	36.0	38	95.3	349
	Out:	99.9	435	60.1	53	100.3	430

internal peers contacted by each HS and the (iii) number of flows sent by the same HS toward the same internal peer.

During the outage, these HS peers generate roughly 30% more signaling flows than usual, with top 5% HS peers generating more than 19k of flows toward internal clients, which is almost the double with respect to the normal behavior. Thus, outage apparently triggered another type of reaction, as it actually intensifies the flow of information between peers that were reachable during the storm.

However, while HS peers contact on average *more* internal peers than usual (5.7 versus 3.8–4.4), the top 5% HS peers actually contact *fewer* internal peers than usual (133 versus 218–232). Moreover, both the average and the 95<sup>th</sup> percentile of the number of flows sent by HS peers to their internal contacts is halved with respect to normal situation. Taking into account packet size and flow length statistics as well, reported in the bottom part of Tab. III, we can conclude that, during the outage a *smaller amount of information, carried by shorter flows constituted by smaller packets, was spread out to a greater number of peers*. This is possibly due to the fact that peers keep asking questions (e.g., which reliable super-node should I contact?) each other but get no answer.

### E. Traffic Source Geolocation

In this section, we analyze the geographical location of the contacted peers, in the attempt to collect evidence of a “outage wave,” which possibly occurred as a consequence of contemporary restart of PCs running Skype. To geolocate the Skype clients, we queried the geographical location of the above addresses using HostIP [12], a public, open and free IP address database. Due to the significant number of external hosts, sampling has been applied to reduce the size of the dataset to explore. More specifically, the initial 14 hours of each period are considered, in which the first 1,000 external peers observed every 5 minutes are geolocated, yielding to a total of 168k samples per period.

The distribution of the peers timezone is depicted in Fig. 4, which reports the percentage of peers as a function of the time-lag with respect to the GMT; for reference purpose, the top x-axis also reports a few reference countries. Though the breakdown is very similar across the three time periods, an interesting discrepancy arises. First, as already observed in [11], the picture strongly suggests that peer selection is geographically driven. For all time periods, more than 40%

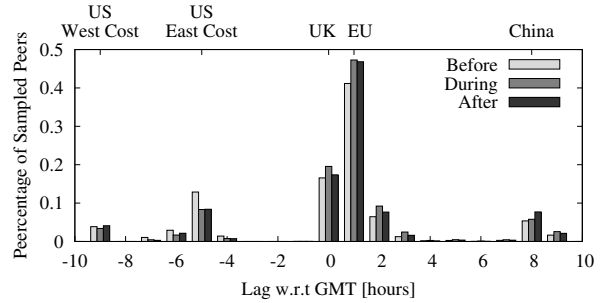


Fig. 4. Peers geolocation: lag breakdown of the sampled peer population

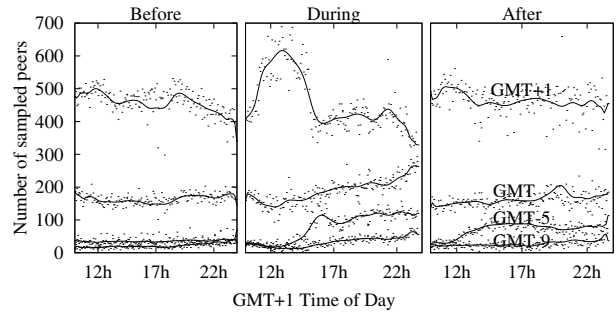


Fig. 5. Time evolution of the lag breakdown

external peers have the same timezone of the vantage point in Italy (i.e., GMT+1, which is common to most European countries). However, a closer look at the picture reveals that, before the outage, the geographical clusterization among GMT+1 peers was significantly smaller, since there were a higher number of peers with time zones different from GMT+1 (e.g., see the GMT-5 peak corresponding to US East cost, Brazil, etc.). Thus, a *Skype client preferably selects “closer” peers, and this preference has been strengthened after the outage, possibly due to different parameter settings*.

Finally the time evolution of the GMT-lag is depicted in Fig. 5 for the most significant time zones early observed (namely, GMT-9, GMT-5, GMT and GMT+1). Each dot in the picture represents the number of peers in each timezone measured every 5 min period, where a smoothed bezier fitting of the dataset is reported for the ease of visualization. It is possible to notice that before and after the outage, the peer distribution is almost uncorrelated with the time. On the contrary, a clear transient effect is present during the Skype outage, with an evident increase in the number of contacted peers at GMT+1, to which it corresponds a decrease of peers at GMT-5. This transient phase reaches its maximum at 1 PM GMT+1, after which the number of GMT+1 peers start decreasing and the number of peers at GMT-5 starts increasing (this may be due to peers at GMT-5 slowly coming back after the windows update process, i.e., at 7 AM for GMT-5). Still, one would expect the same behavior to show at different times when looking at other set of peers with common time zone (e.g., GMT and GMT-9), which is however not true.

As such we therefore cannot find clear evidence of the massive, synchronized PC restart as root cause for overlay disruption. A possible reason to the “desynchronization” may be due to the fact that PC have to download and install windows updates, which introduces different delays spreading the reboot process over time.

#### IV. RELATED WORK

As the use of Skype spreads, the interest of the research community increased as well [4]–[11]. The identification of Skype traffic has been the object of [4], [5]. In our previous work [4], we devised a classification engine for the identification of Skype sessions, by exploiting both stochastic characteristics of the voice flow and statistics of the packet payload. As a side effect of the fine-grained classification, we are able to discriminate the different flavors of the VoIP traffic generated by the Skype application, and to precisely individuate signaling traffic. Authors of [5] only focus on relayed traffic, using Skype as an example of application. Other work, such as [6]–[8] focus on the characterization of Skype traffic and its users: [6] presents an experimental study of the Skype churn rate and the network workload, limitedly considering relayed sessions only. Authors in [7] focus instead on the evaluation of the QoS level provided by Skype calls, whereas our previous work [8] investigates the Skype congestion control mechanism, i.e., how does Skype react to varying network condition.

Related to this work are [9]–[11], which concern mainly the Skype protocols. Authors of [9] provide a very deep understanding of Skype internals, including many details gathered from a partial reverse engineering of Skype protocol and application, with a special focus on security issues. From a low-level inspection of Skype datagram, authors argue that almost everything is ciphered. An active methodology is instead used on [10] to provide an overview of Skype design and functions under several different network setups. Authors uncover functions such as the bootstrap, the authentication and login phases, the traversal of NATs and firewalls, the call establishment and tear-down. Finally, in [11], we present a preliminary investigation of Skype signaling traffic.

To summarize, all previous papers completely ignore Skype signaling traffic except [10], [11], whose focus is however rather different from the one of the present work, since they focus on the *normal* behavior of the Skype network, as opposite as the analysis of a pathological situation, which is indeed the main objective of this work.

#### V. CONCLUSIONS

This paper investigated the Skype outage event, adopting a passive measurement approach that focused on monitoring Skype signaling traffic before, during and after the phenomenon. While the collected measurement allow to precisely quantify several interesting aspects of the phenomenon, it is however very difficult to gather clear insights, that either confirm or deny any strong evidence on the root cause of the breakdown itself. Nevertheless, we believe that this work

quantifies and describes a number of interesting aspects concerning the disruption of an Internet-scale P2P overlay, which we briefly summarize here.

First, the outage manifested itself as a UDP communication storm of significant amplitude: during the outage we observe, on average, a 4-fold increase of the number of flows, a 3-fold increase of packet sending rate and a 10-fold increase of the number of contacted peers with respect to normal P2P operation. Moreover, during the storm some peer may have acted as “lightning rods,” incurring in exceptionally intense signaling activity – e.g., in our network, the most active client experienced a 10-fold increase of traffic amount, and a 30-fold degree of number of contacted peers, topping to more than 11 million of overlay connections. The most important component of the storm is thus constituted by an intense probing activity, which both increased the overlay meshedness (as the number of contacts per peer increases) as well as producing some severe hotspot (as the number of contacts explodes for some peer). Moreover, while we verified that the number of incoming probes roughly followed the “normal” peer activity, we also observed that a few peers may nevertheless have experienced unprecedented popularity, attracting a significant fraction of the incoming probes.

At the same time, packet probing does not represent the only component of the storm: indeed, the absolute amount of non-probe exchanges increased significantly, as the storm also intensified the flow of information between reachable peers. More precisely, we observed that a smaller amount of information, carried by shorter flows constituted by smaller packets, is spread out to a greater number of peers during the outage. Finally, our findings also suggest that some parameters of the Skype overlay maintenance may have changed: for instance, the peer selection mechanism exhibits an increased geographical clustering after the outage.

#### REFERENCES

- [1] Skype Web site, <http://www.skype.com>
- [2] “International carriers’ traffic grows despite Skype popularity”, Tele-Geography Report and Database, Available on line <http://www.telegeography.com/>, Dec. 2006.
- [3] Skype Heartbeats Archives, <http://heartbeat.skype.com> <http://heartbeat.skype.com/2007/08/>
- [4] D.Bonfiglio, M.Mellia, M. Meo, D.Rossi, P.Tofanelli, “Revealing Skype Traffic: when randomness plays with you”, *ACM Sigcomm’07*, Kyoto, Japan, Aug. 2007.
- [5] K. Suh, D. R. Figueredo, J. Kurose, D. Towsley, “Characterizing and detecting relayed traffic: A case study using Skype.”, *IEEE Infocom’06*,
- [6] 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, IP2PS’06*, Santa Barbara, CA, Feb. 2006.
- [7] K. Ta Chen, C. Y. Huang, P. Huang, C. L. Lei “Quantifying Skype User Satisfaction”, *ACM Sigcomm’06*, Pisa, Italy, Sep. 2006.
- [8] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca and D. Rossi, “Tracking Down Skype Traffic,” *IEEE Infocom’08*, Phoenix, AZ, USA, Apr. 2008.
- [9] P. Biondi, F. Desclaux, “Silver Needle in the Skype.” *Black Hat Europe’06*, Amsterdam, the Netherlands, Mar. 2006.
- [10] S. A., Baset, H. Schulzrinne, “An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol.” *IEEE Infocom’06*, Barcelona, Spain, Apr. 2006.
- [11] D.Rossi, M.Mellia, M.Meio, “Following Skype Signalling Footstep”, *IEEE IT-NEWS’08 (QOS-IP)*, Venice, Italy, Feb. 2008.
- [12] <http://www.hostip.info>