A tale of two outages: a study of the Skype network in distress

Brian Trammell and Dominik Schatzmann, ETH Zurich, Switzerland

***Abstract*—This work applies `snack`, a flow-based algorithm for detecting connections between clients and supernodes to the Skype network, to the study of two separate outages on the Skype network in August 2007 and December 2010. We examine both outages in detail, comparing them to each other, and to previous work on the first outage. We find that a simple metric derived from connection events per unique client as measured by `snack` acts as a proxy for the Skype network, and would have provided forewarning of the 2007 outage, and may have applicability as an indicator of general Skype network health. We also examine the faster onset of and recovery from the 2010 outage, discovering that the distribution of clients per supernode is a proxy measurement for network centralization, and detects the apparent present recovery strategy from such outages. These two metrics may be applied by any network operator to monitor the health of the Skype network.**

I. Introduction

The Internet video- and audio-conferencing platform Skype has grown to be an essential communication tool for many of its users. On any given day more than twenty million users are connected to the network[1]. Given its importance, and its proprietary nature, the authors developed a lightweight method for detecting connected Skype clients using only flow data, applicable both to on-line detection or retrospective analysis in large-scale flow data sets [10].

The Skype overlay network is made up of instances of the Skype client, some of which are promoted to so-called supernodes according to connectivity and accessibility from the open Internet. These supernodes make up the internal nodes of the overlay network, providing metadata forwarding and relay services for NAT traversal to the other Skype clients. This architecture implies that Skype SA[2] does not own most of the infrastructure of the network used to provide the Skype service, enabling the rapid growth and high scalability of the Skype network.

Clients associate with supernodes on each startup in order to connect to the Skype network. The method in [10] detects these to associations in order to detect Skype clients and supernodes in flow data. In this work, we apply this method to a flow data set from a national scale network to examine the behavior of the Skype network during two outage events. The first of these occurred on 16 August 2007, and was caused by a slow collapse of the network, due to mass shutdown of Skype supernodes hosted on Windows machines as those machines rebooted following a "Patch Tuesday" on which the operating system software was updated [8]. We observe a long period of the Skype network attempting to heal itself,

The second of these, on 22 December 2010, was also caused by a mass shutdown of supernodes. This, however, was caused by a bug in a certain revision of the Skype software for Windows, and the shutdown occurred much more quickly, on the order of minutes as opposed to days [6]. Recovery was much more orderly in this case, as Skype SA intervened in the reconnection of the network by building a temporary set of "mega-supernodes".

This paper first reviews the operation of the flow-based Skype detector we used for this study and describes the examined data set. We then take a detailed look at two outages, first in 2007 and in 2010.

[1] as reported by Skype client, 14 January 2011

[2] In this work, *Skype* refers to the overlay network, the application it supports, and the protocol(s) it implements, while *Skype SA* refers to the corporate entity which maintains and further develops Skype.

II. Flow-based Skype detection

This work uses the `snack` flow-based Skype detection algorithm [10]. Snack detects *connection events* between clients and supernodes; its output is a set of assertions that client c contacted supernode s at time t. It does this by detecting the acknowledgment of a *UDP Probe* flow from supernode to client, followed by a *TCP handshake* flow back to the supernode. It is not a general-purpose Skype traffic classifier; its primary use is Skype node detection, so it can be used to identify Skype clients and supernodes on a network, and associations between them. However, in comparison to other work in traffic classification which requires packet headers and some payload, `snack` operates on flow data. This enables it to be efficiently used to study very large portions of the Skype network.

Its application in this work is to provide information about the relative size of the Skype network at given points in time (on the assumption that the connection event rate is proportional to the number of online clients and supernodes), and to examine statistics on connection events themselves. Significantly, we note that a connection event does not necessarily represent a successful connection to the Skype network, that is, that supernodes send positive acknowledgment to a client even if a subsequent complete connection to the Skype network will fail. This property makes connection event statistics useful both for studying outages, and for detecting impending outages.

The flow data set used in this study was collected from the border of SWITCH [9], the Swiss national research and education network. SWITCH operates a production network providing connectivity to the Internet for universities and research laboratories across Switzerland. This network contains about 2.3 million "internal" IPv4 addresses (i.e., hosts for

978-1-4577-9538-2/11/$26.00 ©2011 IEEE

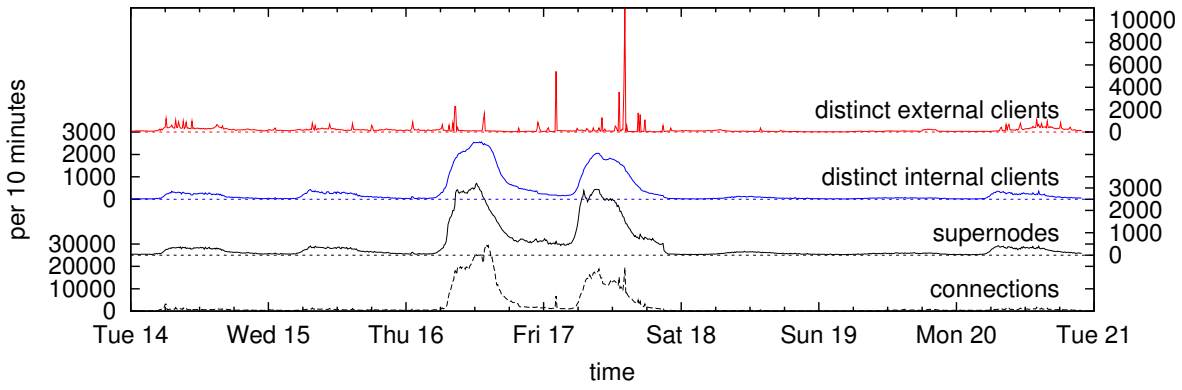

Fig. 1. Activity during outage, August 2007

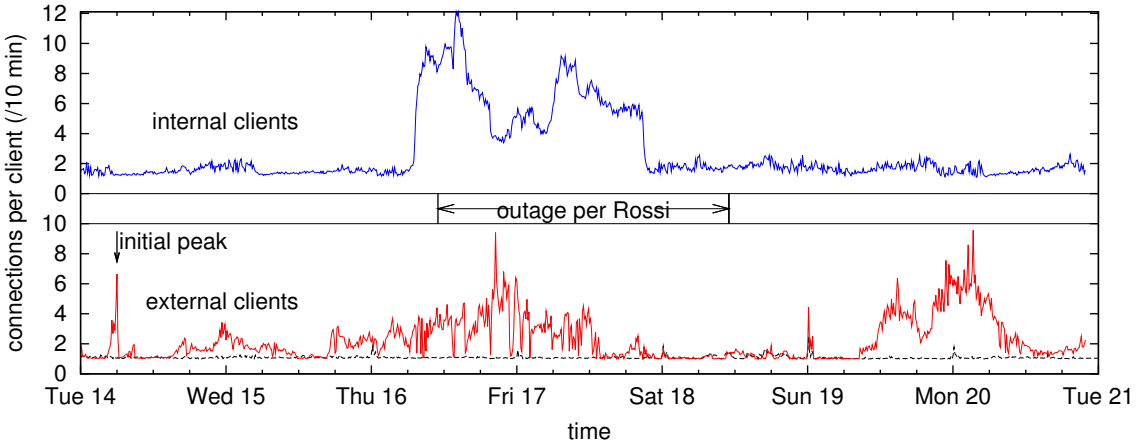

Fig. 2. Connections per internal/external client during outage, August 2007; compared to prior-week baseline for external clients

which SWITCH provides Internet access), and the typical daily traffic volume is between 50 and 100 terabytes. The data set is made up of hourly data files containing on the order of 200 megabytes to two gigabytes of compressed flow data per hour. The data observed here includes one full week of data from August 2007, and four days of data from December 2010, before, during, and after each outage.

III. August 2007: A slow meltdown

In August 2007, the Skype network experienced an outage caused by network instability due to mass rebooting of Windows Skype nodes after a routine Windows Update patch installation [8]. We exploit the fact that a connection event detected by `snack` does not necessarily represent a successful connection to the Skype network to use `snack` to explore the behavior of the connection phase of the signaling plane during the outage.

A. Analysis of outage

Figure 1 gives ten-minute time series counts for connection events, distinct supernodes, and distinct clients internal and external to the SWITCH network for the week beginning 00:00 UTC Tuesday 14 August 2007, two days before the outage is reported by Rossi *et. al.* in [7]. Note in this figure that shortly after 07:00 UTC on Thursday, the number of contacted supernodes begins to rise above its baseline daily peak of less than 400 per ten minutes to a peak of 3205 between 12:20 and 12:30 UTC. This increase in the size of the observed network does not correspond to an actual expansion of the network; instead, it represents many more connection events becoming visible to the `snack` algorithm as Skype clients continue attempting to associate with supernodes in order to connect to the network as a reaction to the network becoming unstable; these supernodes are normally present, but not seen during a given interval because no client is attempting to connect to them.

At the daily scale throughout the time before, during, and after the event, about 1200-1300 distinct internal supernodes are contacted daily.

The distributions of internal and external clients in this figure are wildly different. The internal client distribution rises along with observed supernodes and connection attempts. As the network fails, internal clients begin contacting external supernodes. Since the retry interval for interactive applications is typically less than the measurement interval of ten minutes, we see essentially every internal client attempting to connect

Time UTC	Duration	Distinct Clients
Th 16 Aug 08:32	5 s	2290
Th 16 Aug 13:48	2 s	1660
Fr 17 Aug 02:13	247 s	5422
Fr 17 Aug 13:10	5 s	3577
Fr 17 Aug 14:09	236 s	11093
Fr 17 Aug 16:30	6 s	1719
Fr 17 Aug 16:57	113 s	1560

TABLE I
TIME, DURATION, AND DISTINCT CLIENT COUNTS OF EXTERNAL CLIENT CONNECTION PEAKS DURING OUTAGE, AUGUST 2007

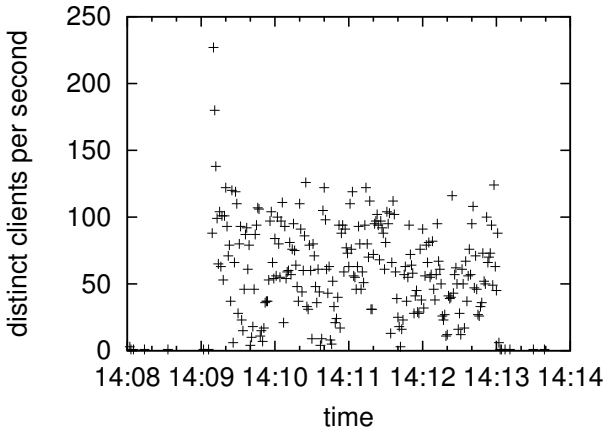

Fig. 3. Clients per second, external client connection flood at 14:09 UTC Friday during outage, August 2007

to the nextwork at the height of the outage. The increase here begins at 06:30 UTC on Thursday, peaking at 2561 at 13:10 UTC on Thursday and again at 2053 at 09:30 UTC on Friday. We interpret this earlier, lower peak on Friday to represent users internal to the network who shut the application down, giving up on attempting to connect as a reaction to the outage.

External client counts, on the other hand, are much noisier, with several peaks filling only one or two ten minute bins. Those with more than 1500 clients per ten minutes are examined at higher resolution in Table I, and the shape of the highest peak is shown at one-second resolution in Figure 3. Instead of persistent retries, we interpret these as mass reconnection attempts. In each case, each peak involves only one internal supernode. This internal supernode's address is distributed to a large number of external clients, which all then connect to the internal supernode within a very short period of time.

B. Measuring reconnection

A careful examination of a different metric, the number of connection attempts per distinct client, provides a different viewpoint on retry activity, and indeed shows that retry activity starts much earlier. Normally, the number of connections per distinct client per hour, as shown in Figure 4, stays reasonably close to one, indicating little client reconnection or connection retry. Using the same data set examined in [10], we measured a long term trend in the 95th percentile of this metric from about 2.13 in 2004 to 1.31 in 2009, indicating increasing network stability over time. During the week of the outage (Tuesday 14 August through Monday 20 August, inclusive), however, the 95th percentile is 26.3 connections per client per hour. Note that the data point for 2007 exclusive of the week out the outage is out of trend, as well, due to longer term recovery from this outage beyond the week removed from the 2007 dataset.

In Figure 2, we compare connections per client per ten minutes for the week of the outage, split by internal and external clients. For external clients, we compare these to baseline activity for the week immediately preceding the outage. Here we clearly see a one-hour increase beginning at 05:00 UTC to peak of 6.6 connection attempts per external client per ten minutes at 06:00 UTC. We interpret this as indicative of instability in a remote part of the Skype network due to the first wave of "patch Tuesday" reboots that caused the outage; this peak is notably absent in the internal network. External retry activity remains elevated over the baseline from about 22:00 UTC Tuesday to about 04:00 UTC Wednesday, corresponding to daytime in Asia. It then rises over baseline again at 17:10 UTC Wednesday, remaining there through Friday.

The outage then spreads into the observed network. Internal client retry activity rises rapidly beginning at 07:00 UTC Thursday, coincident with the increase in number of observed supernodes, plateauing around ten then peaking at about twelve connection attempts per client per ten minutes. This suggests a process with a retry interval of about one minute, to be expected for connection failure in an interactive application. Retry activity from internal clients falls at around 22:00 UTC on Friday, along with the number of connection attempts, observed supernodes, and observed internal clients, and does not peak again.

However, instability is still visible in the external network, with elevated retries returning around 11:30 UTC on Sunday and remaining high until about 12:00 UTC on Monday, corresponding roughly to daytime in Asia. The return of external

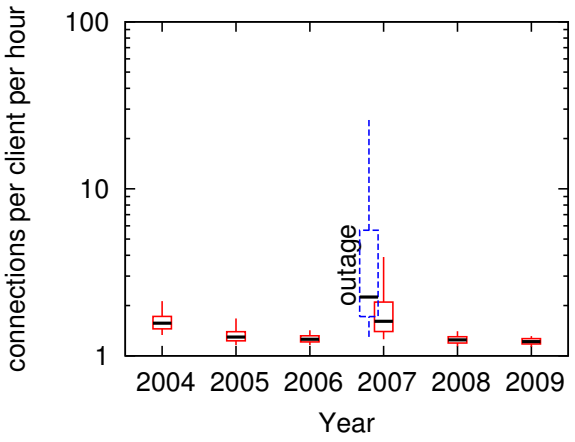

Fig. 4. Hourly connections per client, 2004-2009

client retry activity to near baseline levels is coincident with Skype's official announcement [8] that the disruption was over.

We note that both the distinct supernodes and connections per distinct client metric produced by this approach provide information on this outage event before the previously reported [7] start time of 11:00 UTC on Thursday 16 August 2007, coincident with a Skype UDP traffic spike. We note here that we did see signs of the outage on external clients before those on internal clients, so this may also be impacted by the geographically different and somewhat smaller observation surface used in [7]. Distinct supernode counts began to rise significantly over baseline some four hours before the spike, and anomalous retry activity was detectable a full two days before, back to the original mass reboots which were the original cause of the outage.

Connections per distinct client, a metric which is trivially easy to calculate from the output of `snack`, appears in this case to serve as a proxy for the health of the Skype network overall. We proposed that this could be used as part of a simple scheme to passively detect anomalous retry activity across a measured portion of the Skype overlay network, providing warning of trouble significantly in advance of user-visible disruption or volume-based anomaly detection warnings. However, as we will see in the following section, this is not the case for every outage, and may no longer be applicable given the current configuration of the network or policies for recovering from outages.

IV. An instant collapse: December 2010

According to Skype SA [6], on December 22, 2010 at around 16:00 UTC, the Skype network failed due to a bug in a version 5.0.0.152 of the Windows Skype client. This bug caused about 20% of online Skype clients to crash, reducing the global number of active supernodes by 25-30%. Mass restarts of the crashing Windows clients in turn overloaded the remaining supernodes, which began shutting down to protect their host systems from overload. The resulting positive feedback loop rapidly brought down the network.

We confirm this public report with a look at observed clients and supernodes during the timeframe of the outage in Figure 5. A more or less normal pattern of connection events for midday on a workday is disrupted by a spike in observed clients and supernodes around 15:50, followed by a crash at 16:00. We interpret this spike as the initial reconnection attempts during client restart, as in in the 2007 outage. However, the collapse was already essentially complete at this point.

While the outage is also visible in the connections per distinct client metric, as shown in Figure 6, this metric would not have been as useful in giving advance warning of the outage, given the rapidity of the collapse of the network. Indeed, external retry attempts pick up four hours after the outage is well underway, around 20:30 UTC, peaking around 2.6 at 21:20, falling below 1.5 again after 23:10 UTC. We interpret this as a transient spike in reconnection, similar to those observed in 2007.

Reasons for the observed reduction in retry activity in comparison to the 2007 outage could include the rapidity of

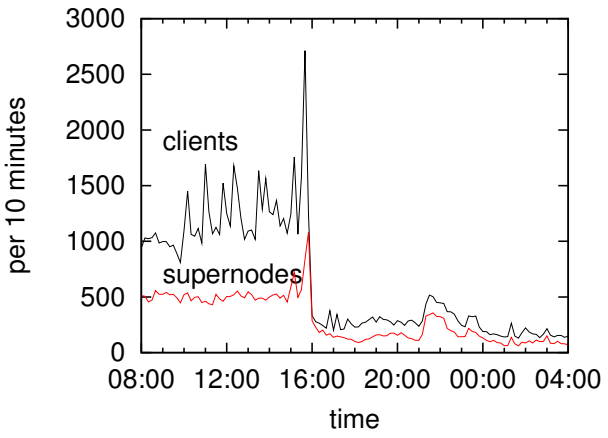

Fig. 5. Observed clients and supernodes at time of outage, December 2010

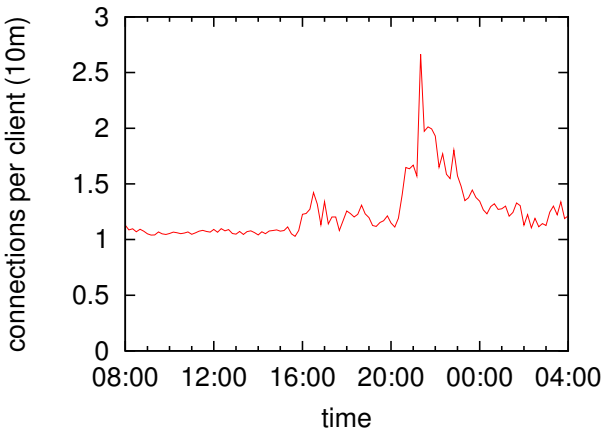

Fig. 6. Connection attempts per distinct external client per ten minutes during outage, December 2010

the collapse, as well as changes to the operation of the Skype network since 2007. However, it would appear that this is also related to the method used for recovery, as well.

A. Centralization and Recovery

Skype SA reported [6] that it was brought a set of "mega-supernodes" online in order to recover from this outage. By doing so, they in effect temporarily centralized the topology of the Skype overlay network, allowing the network to restart itself.

Assuming that clients connecting to these mega-supernodes were more capable of a successful connection to the network[3], a client which connected to one of these mega-supernodes would no longer retry to connect via supernodes located within our network, as these would have a lower probability of eventual successful connection. This would result in a measurable reduction in retry activity, as observed.

We confirm this centralization by looking at the address distribution of supernodes contacted by internal clients. Here

[3]The authors consider this to be a safe assumption; otherwise, additional high-capacity supernodes would be pointless.

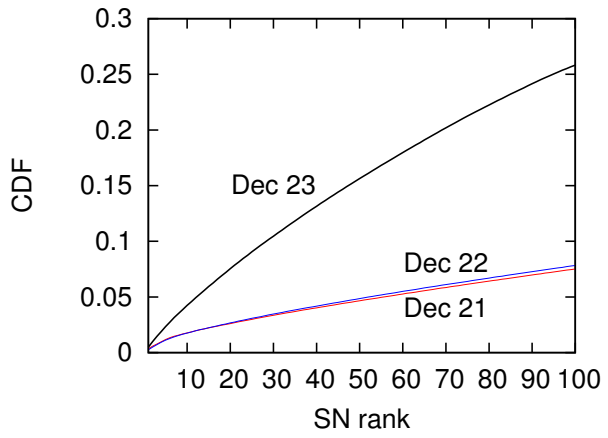

Fig. 7. Distribution of connection events from internal clients by external supernode rank, December 2010

we order external supernodes by rank according to the number of distinct internal clients connecting to them, and show the distribution of connection events during each full day by supernode rank in Figure 7. Here, we see that the connection events for the top 100 supernodes cover about 7% of all connection events both before the outage, on December 21, and during the outage on December 22.

However, on December 23, the top 100 supernodes account for 25% of all connection events. Further examination shows that many new high-volume supernodes appear on December 23 in just two /24 subnetworks: 78.141.181.0/24 in AS6661 (P&T Luxembourg, the Luxembourg national post and telecom enterprise; Skype SA is located in Luxembourg), and 149.5.45.0/24 (registered to Skype SA)[4]. These are presumably the mega-supernodes brought online by Skype SA.

These mega-supernodes disappear from the Top 100 on December 24th, confirming the return of Skype to normal operation as a decentralized, peer-to-peer network[5]. We presume the rapidity of the appearance and subsequent disappearance of these supernodes indicates that Skype SA maintains this capability to heal the network during any future occurrence of a similar outage.

V. Related Work

Research on Skype traffic classification and measurement follows from initial work in reverse engineering [2] and as study of the protocol from the network management viewpoint [3]. Packet-level classification techniques for Skype traffic [4], [5] followed from these. `snack` [10], used by this work for detecting Skype connection events using only unadorned flow data, is inspired by Adami *et al.* [1], who built a packet-level real-time Skype traffic classifier based on reverse engineering of the protocol. Rossi et. al. [7] leveraged this detector in previous detailed study of the August 2007 outage. This study observed a different, and significantly smaller, portion of the network from that we measure in section III of this work.

[4]This information was verified using `rwhois` on 13 January 2011.

[5]Volume metrics for December 24 are not shown, as the SWITCH network on which measurements were made is much less active than normal during the Christmas holiday.

VI. Conclusions

Though this work is focused on the performance of the Skype network during two specific points in time, some general guidance can be taken from from it. In applying our approach to a detailed analysis of the 2007 outage event, we have discovered that the count of connections per distinct client in the Skype overlay network has long-term stability and the potential to be used by external network operators as a general health indicator for the network, though the present strategy used by Skype SA to heal the network – increasing its centralization using "mega-supernodes" under its control – may have led to a reduced peak of this metric in 2010. However, a technique for determining centralization by examining the distribution of clients per supernode itself adds to the full picture of Skype network health.

In the development of this work, we seek to demonstrate the use of open techniques for flow-level network traffic analysis on the analysis of proprietary protocols. As the importance and popularity of applications using proprietary protocols such as Skype continues to increase, we encourage the further application of approaches such as `snack` to allow researchers, network operators, and other third parties to measure the operation of these networks. While we note and applaud that Skype SA was much more forthcoming with details about the 2010 outage as compared to 2007, independent measurement of applications on the open Internet is a useful goal.

VII. Acknowledgments

The authors thank the EU FP7 PRISM and DEMONS projects for their support of this work. We would like to acknowledge SWITCH, the Swiss Research and Education Network, for providing the data used in this study.

References

[1] D. Adami, C. Callegari, S. Giordano, M. Pagano, and T. Pepe. A real-time algorithm for Skype traffic detection and classification. In *9th International Conference on Wired/Wireless Networking*, Sept. 2009.

[2] S. A. Baset and H. G. Schulzrinne. An analysis of the Skype peer-to-peer internet telephony protocol. In *INFOCOM 2006, 25th IEEE International Conference on Computer Communications*, Apr. 2006.

[3] P. Biondi and F. Desclaux. Silver needle in the Skype. In *Black Hat Europe 2006*, Mar. 2006.

[4] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli. Revealing Skype traffic: when randomness plays with you. *SIGCOMM Computer Communications Review*, 37(4):37–48, 2007.

[5] S. Guha, N. Daswani, and R. Jain. An experimental study of the Skype peer-to-peer VoIP system. In *IPTPS'06: The 5th International Workshop on Peer-to-Peer Systems, Microsoft Research*, 2006.

[6] L. Rabbe. CIO update: Post-mortem on the Skype outage. http://blogs.skype.com/en/2010/12/cio_update.html,.

[7] D. Rossi, M. Mellia, and M. Meo. Evidences behind Skype outage. In *IEEE International Conference on Communications 2009*, June 2009.

[8] Skype. Heartbeat (status blog): What happened on August 16. http://heartbeat.skype.com/2007/08/what_happened_on_august_16.html.

[9] SWITCH. The Swiss Education and Research Network. http://www.switch.ch.

[10] B. Trammell, E. Boschi, G. Procissi, C. Callegari, P. Dorfinger, and D. Schatzmann. Identifying Skype traffic in a large-scale flow data repository. In *Proceedings of the Third COST TMA International Workshop on Traffic Monitoring and Analysis (TMA 2011)*, Vienna, Austria, Apr 2011.