Pictures from the Skype

Dario Rossi,
ENST, France
dario.rossi@enst.fr

Silvio Valenti,
ENST, France
silvio.valenti@enst.fr

Paolo Veglia,
ENST, France
paolo.veglia@enst.fr

Dario Bonfiglio,
Politecnico di Torino, Italy
bonfiglio@tlc.polito.it

Marco Mellia,
Politecnico di Torino, Italy
mellia@tlc.polito.it

Michela Meo
Politecnico di Torino, Italy
meo@tlc.polito.it

ABSTRACT

This demo focuses on the online characterization and classification of Skype traffic, a very popular and fashionable VoIP application nowadays. Building over previous work on the field, we aim at illustrating the classification process of Skype calls in an interactive fashion using a controlled testbed. The demo also focuses on interesting characterization of Skype traffic, such as representing the traffic patterns Skype generates during a call and while idle, or the geographical localization of Skype peers. Finally, the demo provides interesting insights on the actual Skype usage by users, showing the classification engine running live, and showing the persistent monitoring of real networks.

Categories and Subject Descriptors

C.4 [**Computer Communication**]: Measurement Techniques; C.2.5 [**Computer Communication Network**]: Internet

General Terms

Demo, Experimentation, Measurement

1. INTRODUCTION TO SKYPE

VoIP is currently becoming a synonym for telephony as the increasing number of operators that are offering VoIP-based phone services to users suggests. Skype is beyond any doubt the most amazing example of this new phenomenon: developed in 2003 by the creators of KaZaa, it recently reached over 100 millions of users, about 10 millions of which are on-line at the same time, becoming so popular that people indicate Skype IDs in their business cards. The goal of this demo proposal is to allow people

- to observe Skype traffic generation, by showing in real time the packet generation process of a Skype VoIP/video call
- to understand state of the art techniques to classify Skype traffic from passive measurements
- to monitor Skype usage by users in operative networks.

In [1] we proposed a classification framework to reveal the presence of Skype traffic within traffic aggregates. The classification engine is based on two different and complementary techniques. The first approach is based on a stochastic characterization of Skype traffic in terms of inter-packet gap (IPG) and packet length, which are used as features of a decision process based on *Naive Bayesian Classifiers* (NBC): while the above features successfully allow to identify VoIP traffic, they are not representative of the application that generated it. Therefore, a second technique is needed in order to detect Skype fingerprint from the packet framing structure: this novel traffic identification methodology, based on Pearson's Chi Square test and agnostic to VoIP-related traffic characteristics, is baptized as *Chi-Square Classifier* (CSC).

Based on the above classification framework, we setup an interactive demo that aims at assessing two complementary aspects on the life of Skype peers. First, as described in Sec. 1.1, we illustrate the classification mechanism in an interactive fashion using a controlled active testbed. The output of the packet generation process and of the two classifiers is shown and updated during live Skype calls, so that users can appreciate the variability of the traffic pattern Skypes generates, and the effectiveness of the classifiers. The demo also focuses on the characterization of Skype traffic during idle operation. Interesting aspects of the Peer-to-Peer overlay maintenance will be shown, such as the representation of the traffic patterns Skype generates, or the geographical localization of Skype peers. Finally, as Sec. 1.2 reports, the demo shows the output of continuous and persistent network monitoring of Skype traffic: by running the classification engine on operative networks, we provide more insights on the actual Skype usage in real environments.

1.1 Skype Classification

The NBC and CSC techniques are jointly used in [1] to take a classification decision at the flow end. However, this choice is can be taken early, as soon as enough packets are collected. To show this, we use a simple testbed in which a PC is connected to the network via a Linux box serving as a router, traffic monitor and classifier. A first aim of the demo is thus to illustrate the Skype call traffic generation and classification processes in an interactive fashion by means of the controlled testbed.

For reason of space, the whole process is only briefly described here: we refer the reader to [1] for a more thorough description of the classification engine and to [2] for further details concerning the demo and the software itself. The demo software is a wrapper around the classification framework which we implemented in TSTAT [3], a flow level logger and traffic analyzer originally developed at Politecnico di Torino.

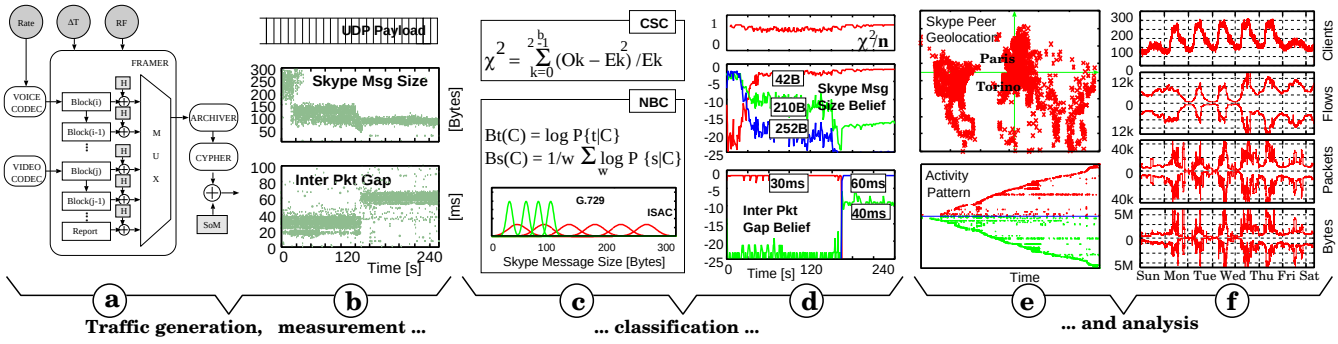

Figure 1: Synopsis of Skype Traffic Generation, Measurement, Classification and Analysis Process

The whole process is sketched in Fig. 1, where different subsequent phases are reported from left to right: traffic generation, measurement, classification and analysis. Whenever Skype uses UDP as transport level protocol, the client assembles a message by multiplexing eventually several voice/video/chat/data blocks into the payload of a single transport layer segment as sketched in Fig. 1-(a). Encryption is applied to the resulting message to protect the information and to hide protocol details. Since Skype preferred transport layer protocol is UDP, we mostly ignore TCP traffic for the sake of simplicity. A stream of UDP Skype messages is depicted in Fig. 1-(b): from top to bottom, we are interested in the UDP packet payload, message size and inter-packet gap (IPG).

Let us focus on the UDP payload first. As UDP offers only a connectionless unreliable service, there is no guarantees that data will be delivered entirely and in-sequence: therefore, a Skype receiver must be able to extract information to detect and deal with possible incorrect cypher stream lining (e.g., due to the loss of a message) directly from the received packet. It follows that an UDP message must contain an application layer header, that cannot be ciphered but rather only be *obfuscated* [4]. As a consequence, it turns out that a few bits of the Skype message are actually *deterministic*, whereas all the others are *random* due to the encryption process: this fact is exploited by the Pearson CSC classifier at the top of Fig. 1-(c), which intuitively quantifies the amount of randomness in groups of bits of the UDP payload.

Let us now focus on the message size and IPG, shown in bottom part of Fig. 1-(b), which considering voice/video services have rather distinctive properties. In the case of Skype, these are determined by both the voice/video encoder, and by the Skype framing and congestion control algorithms: as shown in bottom plots of Fig. 1-(b), these metrics may vary widely over time during the same call. Bottom part of Fig. 1-(c) sketches a simplified NBC classifier, which computes the "belief" that a sequence of messages is a voice stream, by comparing real measurements with a description of known codecs. This description is given as the Probability Density Function (PDF) of the typical Skype message size and IPG for different "modes" of any given codec. The belief varies over time as well as in Fig. 1-(d), reflecting the different framing and congestion control policies adopted by Skype during the call as seen early in Fig. 1-(b).

The CSC "payload randomness" and the NBC "voice likelihood" information are then (conservatively) combined to eventually classify the current flow as a VoIP flow which has been generated by Skype. All the above mentioned figures are shown in the demo. In particular, users of the testbed setup are invited to place a Skype call, and observe the packet generation process (as in Fig. 1-(b)), the output of the classifiers (as in Fig. 1-(c)), and the final classification decision.

In addition, the demo aims at showing the Skype activity pattern when idle. Indeed, being Skype a P2P application, lot of signalling messages are exchanged even if no call is in place. By observing the traffic generated by the Skype peer, the demo allows to extend the analysis to other interesting aspects of the P2P overlay maintenance, such as representing the Skype traffic pattern, and the geographical localization of Skype peers, as shown in Fig. 1-(e).

1.2 Skype Characterization

The demo provides interesting insights on the actual Skype usage by users, adopting a complementary approach based on a live and persistent passive monitoring of real networks: TSTAT currently monitors a number of networks in realtime, and a subset of the collected performance figures is browsable through the Web [5]. The demo provides then access to the data, as well as more detailed characterization of the current Skype traffic flowing on the monitored link.[1]

An example of the available data is given in Fig. 1-(f), which reports the typical time evolution of a one week long observation period in the Politecnico di Torino campus LAN. The metrics reported in the figure are, from top to bottom, the number of clients, flows, packets and bytes observed during 5 minutes long time windows.

2. REFERENCES

[1] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi and P. Tofanelli, "Revealing Skype Traffic: When Randomness Plays with You," *ACM SIGCOMM'07*, Kyoto, Japan, Aug. 2007

[2] Skype demo, `http://www.enst.fr/~drossi/SkypeDemo`

[3] M.Mellia, R.Lo Cigno, F.Neri, "Measuring IP and TCP behavior on edge nodes with Tstat", *Computer Networks*, Vol. 47, No. 1, pp. 1–21, Jan 2005

[4] P. Biondi, F. Desclaux, "Silver Needle in the Skype." *Black Hat Europe'06*, Amsterdam, the Netherlands, Mar. 2006.

[5] TSTAT website, `http://tstat.tlc.polito.it/`

[1]This possibly includes the conference WiFi network, provided that privacy and technical issues can be dealt with the local organizer.