Presentation of

A. Villoing

Last update: September 2006

Contents

1 Introduction

The first stable version of Skype has been released in July 2004, since then the number of users kept on growing. Nowadays Skype claims having more than 20 millions accounts and between 4 and 6 millions of users simultaneously connected. Bought by eBay in October 2005 Skype's popularity does not need to be demonstrated. If Skype is the name of a VoIP software it is also by extension the name of its protocol. Skype protocol is a P2P encrypted proprietary protocol, therefore the traffic generated is not easy to detect and it is still a very important issue in for many companies. In this part I will present consecutively the features of the software and the protocol.

2 Software features

If the main purpose of Skype is providing VoIP service it also has many other features, some are close to classic instant messaging services and some emulate a real phone service. Only some of them are free. Only the features that can affect Skype's network behavior are described below.

VoIP from computer to computer This the most used feature especially because it is *free*.

VoIP from computer to regular phone (Skype Out) By registering on Skype's website it is possible buy credit and then call all over the world with very interesting rates compared to rates applied by phone companies.

VoIP from regular phone to computer (Skype In) Skype sells phone numbers available for three month or a year. Those numbers can be issued from 14 different countries. This way, no matter the country where the customer leaves it is possible to have an Americain or a French phone number.

Video conferencing Introduced in Skype2.0 in 2006 this feature implies a modification Skype traffic especially since this option is enabled by default as soon as a video device is detected.

Instant Messaging This feature is comparable to many other instant messaging clients like MSN Messenger, Yahoo! Messenger, Google Talk, etc. The main difference is that Skype does not tell the user whether the person he is chatting with is typing or not. This is due to the P2P design of the Skype network.

File Transfer The Skype network design has a big influence on the quality of file transfers. It can make it very fast ($\sim$1Mbps) or very slow ($\sim$3 kbps).

3 Protocol features

To summarize in few words Skype's protocol we could describe it as a structured Peer-to-Peer, distributed and encrypted protocol.

Structured Peer-to-Peer There are 3 kinds of nodes in the Skype network: the authentication server, the super nodes (SN) and the normal nodes. The two last ones are Skype Clients (SC) run by users on their personal computer, depending on their network configuration the SC will modify its behavior.

The authentication server is used by users to create, login to or modify an account. During the login process a connection is established with this server once and then, in case of successful login, the SC becomes part of the Skype network and does not communicate anymore with this server. Beside this connection there is a version verification done on ui.skype.com that shares its IP (212.72.49.131) with www.skype.com. Indeed, the SC sends a number of version and then the server allow the authentication or force the user to upgrade its SC. The last version is not always required but this ensure that a minimal version is installed. This is the only centralize aspect of the Skype protocol.

SN are SC run by users that have a "good" Internet connection and a "good" computer. Having a good Internet connection means having a public IP address, without a NAT and without firewall restrictions. A good computer is a machine that can forward other users' communications and handle many connections. SN have a role of relay in the network, this is the reason why they need a better connectivity and better performances. As you can see on figure 1 SN are used to connect SC together. It is not possible to establish a connection to a SC behind a NAT unless if some ports are forwarded, but usually it is not the case. So in order to make the P2P network reachable by anybody, nodes with a better connectivity have to be known and have to act as relays for other nodes.

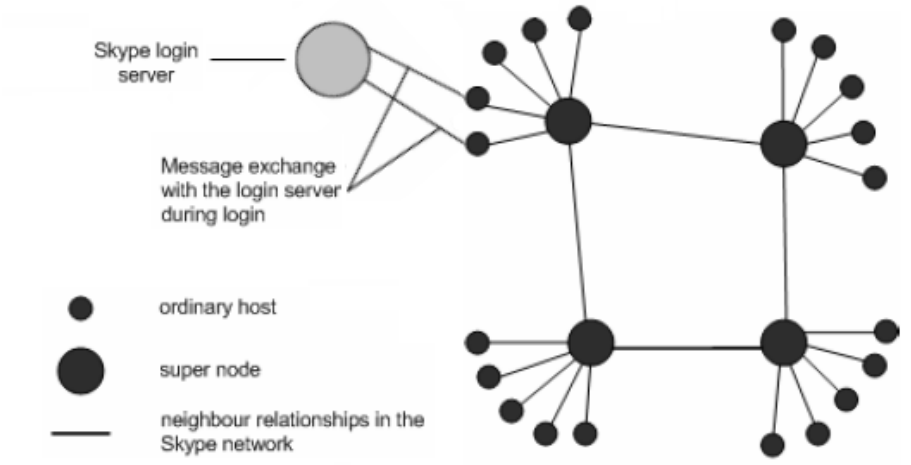

Figure 1: The three main entities of the Skype network

Distributed Skype network is designed in a distributed way. Very few informations are local to users' computers or stored on servers. Only the login/password pairs are stored on a server (the login server). Then apart from the general preferences of the software most of the information about the user are stored on the network. Note that there is also a local copy of everything but when installing Skype on a new computer, a user will retrieve his profile and his buddy list for example. Since version 2.0 local data are stored in C:\Document and Settings\[username]\Application Data\Skype

Encrypted According to Skype publicist, both user identities and software distributions are digitally signed by an RSA private key. The matching RSA public key is embedded into every Skype executable. The asymmetric algorithm used for key exchange is RSA with a 1024 bit key for computer-to-computer communications and a 1536 bit key for communications involving non-free services. Then bulk encryption is performed using a 256 bit AES algorithm.

Since neither the protocol nor the software are known Skype involves lots of security issues but this is not what this paper is about.

A Word About the Skype Voice Codecs Skype uses its own proprietary codec based on three different codecs in order to compress the voice sent in IP packets. Two codecs, iLBC and iSAC are licensed from Global IP Sound and are publicly available. The third one is the property of Skype Inc. and therefore not available. It seems to be a wide-band codec (frequency between 50 and 8000 Hz. [1]) that allows Skype to work with a constant bit rate protocol. It is not feasible to identify Skype communications in the packets payload, as one cannot decipher Skype communications. Hence, an analysis of the Skype protocol must be based on the header of the packets and/or the behavior of the protocol.

4 Protocol presentation

Skype Connection steps during the establishment of the communication with another peer
1. Startup
2. Login sequence
3. Connection to a bootstrap node
4. Determination of the network setup
5. Call establishment to another user
6. Voice communication and/or chat messages
7. Call teardown

Explanation of these steps:

1. Startup For the 1st startup, the Skype client makes a HTTP 1.1 GET request with "installed". The subsequent startups, the Skype clients request a HTTP 1.1 GET with "getlatestversion". In these packets there is also the current version number. So that if there is a critical update the server can force the user to upgrade. In the preferences there is an option "Checking for updates" even if it is disable the HTTP packet is sent. This option just makes a difference between updating for every new version or only for the critical ones.

2. Login Sequence Skype clients directly connect to login servers, whose IP addresses are hard coded within the software. In this connection the login name and the version are sent in clear text format.

3. Connection to a bootstrap node First, the Skype Client tries to connect to 5 SN sending a UDP packet to IP addresses of super nodes randomly chosen in the host cache. Host cache contains a list of 200 potential SN's IP addresses, that is updated if a SN is unreachable. Then the SC tries to connect using TCP connections to the same SN. Otherwise, the SC sends packets to the super node at port 80 (HTTP), and if the SN is not available at this port it retries an attempt to the same node with port 443 (HTTPS). When the client finds a super nodes to connect to, it refreshes its list of active and available super nodes in host cache. When SC is installed the first time it come with a list of SN to connect to. There are 200 IP addresses so they are very few chances for all of them not to be available especially if we assume that some SN are maintained by Skype Inc.

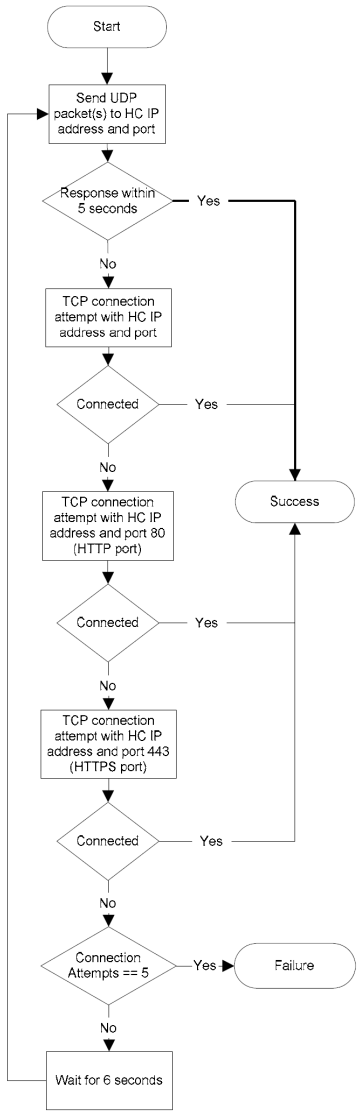

Figure 2: Logical Skype login sequence

4. Determination of the network setup This operation is done by the super nodes. Skype clients seem to use a variant of

the STUN protocol (Simple Traversal of UDP through NATs) [3] to determine the type of firewall or NAT between the newcomer and the Skype network.

5. Call establishment to another user Routing in the Skype overlay network is done by the SN. When a SC tries to establish a call it first ask the its SN (if it is not a SN itself) where is the callee and tries to connect directly to it. If the SC is restricted because of NAT or firewall then it will connect to the callee using a SN as a relay. Once the two SC that want to communicate are connected every kind of data (voice, video, text...) goes through the connection established, relaid or not. We found out that in the case of two SC on the same LAN there will be connected directly together even if they are NATed and have to go through a SN. This means that there must be a mechanism for them to exchange their private IP addresses through the SN and then connect directly. This kind of mechanism can be used every time a SN sees two SC having the same public IP address that want to be connected.

6. Voice communication and/or chat messages Data are sent through the path established before. Note that the network configuration can change along the communication. I have seen a communication over UDP being stopped and continued over TCP when we added a file transfer in parallel of a voice communication. Since TCP is more suitable for file transfers this makes sens. When the SC is still on after a communication with somebody from the budylist Skype uses keep-alive messages every 20 seconds to maintain UDP bindings at NAT and in the case of sending TCP messages to avoid being dropped by congestion window, maybe also to keep track of users status.

7. Call teardown Call teardown is done using the same protocol used for the communications.

"Skype port" During the first installation of Skype the software chooses a random port (hereafter called Skype port). This port can be changed by users in the preferences but nobody does. This port is the one used by the SC to create the UDP and the TCP connections to the SN. So a SN listens on this port over UDP and TCP and on ports 80 and 443 over TCP. If we know this port for every single SC we can tell that all the traffic going on this port for this specific SC is due to Skype. But since it is random it is hard to know.

5 Detecting Skype Clients

Looking at the protocol we can note that when a Skype client starts it always exchange some HTTP packets with ui.skype.com server that can be found at the following IP address: 212.72.49.131. The client sends its version number to the server. Even if the option "check for available update" is disabled this verification is done to ensure that critical releases are widely distributed and that all clients have a minimum version.

This seems to be a very good signature to know who is using Skype. But it appends only once along a whole Skype connection (which is different from a Skype call). So if a user initiates a connection to "ui.skype.com" he has a SC. Here we cannot use the IP address because www.skype.com is on the same server.

So to know who is using Skype we detect "Host: ui.skype.com" in a HTTP packet.

References

[1] Salman A. Baset and Henning Schulzrinne, *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol*, Columbia University, New York, NY, USA, September 2004

[2] K. Suh, D. R. Figueiredo, J. Kurose, D. Towsley, *Characterizing and detecting relayed traffic: A case study using Skype*, University of Massachusetts, Boston, MA, USA, July 2005

[3] Rosenberg, et al., *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, http://www.ietf.org/rfc/rfc3489.txt, March 2003