Enhanced Skype Traffic Identification

Marcell Perényi, Sándor Molnár
Budapest University of Technology & Economics, Department of Telecommunications & Media Informatics, Budapest, Hungary
{perenyim, molnar}@tmit.bme.hu

ABSTRACT

Skype applies strong encryption to provide secure communication inside the whole Skype network. The communication ports of clients are chosen randomly. As a consequence, traditional port based or payload based identification of Skype traffic cannot be applied. In this paper we present a novel flow dynamics based identification method to discover both Skype hosts and voice calls. The method is based only on packet headers and extracted flow level information. This method is the second algorithm from our research. It has a significant improvement over our first method [1]. It can detect the randomly selected communication port of the Skype client, which makes the identification more reliable. The whole identification process is scripted in Transact-SQL, thus it can be executed automatically. We also present the validation of the new algorithm together with the analysis of the identification results.

Keywords

Skype, traffic identification, analysis

1. INTRODUCTION

Skype consists of a world-wide P2P VoIP network. The users can initiate and receive voice calls to/from other Skype users, or even PSTN users using SkypeOut/SkypeIn. Moreover, instant messaging (chat) and file transfer is also supported within the Skype infrastructure.

We considered the Skype traffic from a network operator point of view. They are usually interested in the nature of the traffic carried by their network in order to optimize network performance, forecast future needs and also for marketing purposes by identifying and studying popular applications and services.

As a first step of the analysis the Skype traffic should be identified. The identification is not a simple task, since there is no unique standard port for Skype traffic, the protocol is not public, the data is encrypted and different software versions behave differently. Moreover, the Skype binary uses a variety of techniques to prevent reverse engineering [2].

Our goal was to detect Skype traffic even if the Skype client was started before the traffic measurement, in which case we cannot rely on some typical login traffic patterns or payload information. We also decided to detect all Skype traffic regardless of software version and to avoid the use of payload information, which is in many cases not available. We constructed our identification method to be based on properties and time behavior of data flows and packets. This work is an improved and extended version of our previous publication in the area of Skype identification [1]. The focus is now more on the analysis results. However, the identification method also became more reliable. Similarities and difference between the two identification methods are discussed in Section III.

The identification of Skype traffic is addressed in several recent publications. Ehlert et al. [3] describe a method for identification using traffic patterns and payload information from Skype login phase. For efficient blocking Skype activity has to be identified. Methods for blocking are published (see e.g. [4]), but these are tailored to a given firewall type or security setting and assume that Skype communication starts after enabling the blocking mechanism. Suh et al. [5] present a method for the detection of relayed traffic by comparing input and output traffic patterns.

Guha et al. [6] present some results about Skype usage patterns. Their results are based on active measurements and provide global information about the number of clients, the number of super nodes (see next Section) and general traffic patterns of a single Skype session.

Kuan-Ta Chen et al. [7] describe an identification method for relayed Skype flows. Some of the characteristic flow properties they examine to select Skype voice sessions are similar to ours. They aim to detect relayed flows only, and use the collected data for investigating correlation between call duration and voice quality.

2. SKYPE OVERVIEW

The Skype P2P network consists of the following elements: ordinary nodes (clients), super nodes (SNs), login servers, update servers and buddy-list servers. The elements of the Skype network are depicted in Figure 1.

An ordinary node is a leaf-node of the Skype overlay network; it is the equipment of the user that is used for the communication. SNs are the switching elements in the overlay network, responsible for maintaining a Global Index distributed directory which allows users to find each other. Each SN keeps track of a small number of ordinary nodes. SNs can also function as ordinary nodes, and in fact every ordinary node with public IP address and sufficient capabilities (free CPU, memory, bandwidth capacity) is a candidate to become a SN. This is out of the control of the user.

The login server stores the account information of users. It is responsible for user authentication at the beginning of the session. The update server is also contacted by the client at startup to check whether a newer version of the software is available. The so-called buddy-list server [8] is responsible for storing the contact list of the users.

Valuetools'07, October 23-25, 2007, Nantes, France.
Copyright 2007 ICST 978-963-9799-00-4

When a Skype client is launched it tries to establish a connection with a SN. For the duration of the session this SN will be responsible for the client. It also contacts one of the login servers for authentication.

All communication between the Skype network entities is strongly encrypted. RSA algorithm is used for key exchange, while AES encryption is applied for ciphering traffic.

A detailed description of Skype operation and components can be found among others in [5, 9, 10].

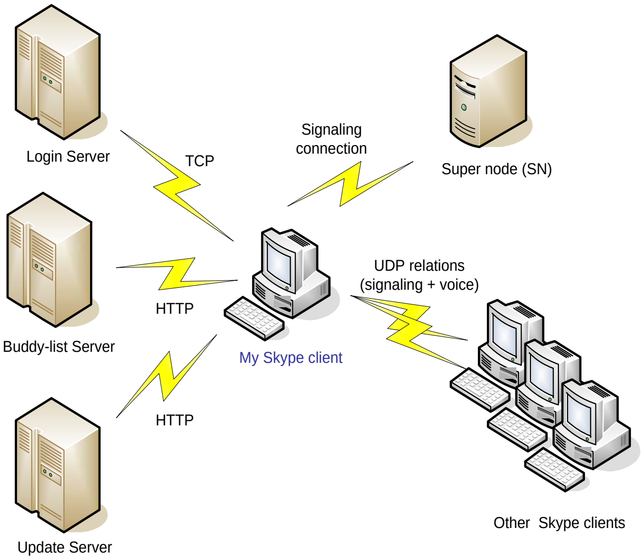

Figure 1. Elements of the Skype P2P overlay network

3. SKYPE IDENTIFICATION

In spite of the fact that the application-layer protocol of Skype is concealed, we can still monitor the network and transport layer protocols and analyze the used IP addresses and ports. The statistical characteristics of the Skype data flows and packets can be studied as well, including flow bandwidths, packet sizes and several other properties. Our proposed identification method is based on these observable open parts of the Skype communication.

The difficulty is that the regular check for software updates does not guarantee that every client runs the latest version of Skype, therefore we need to deal with the behavior of different versions of clients. Although we did not have a chance to analyze each client we based our identification algorithm on those properties which seem to be invariant amongst different software versions. In this section we first present a method to detect Skype activity even if no calls are made, then we present our method for the detection of Skype voice calls.

The identification method introduced in this paper has several advantages over our first version described in [1]. The identification process contains basically two steps in both methods: discovering candidate Skype hosts first, then searching for voice calls. The second step is almost the same in both cases applying flow and packet level properties as filtering criteria. However, the first steps are significantly different. In the first method [1] only the IP addresses of the hosts could be identified – based on connections initiated by the hosts towards other dedicated Skype servers, in addition to a special signaling flow between the host and the SN.

Our second method, which we present in this paper, focuses on the communication between the Skype hosts. It can identify not only the source IP address, but the possible communication port as well, which makes this method more reliable. In addition, this method is simpler, faster and more robust, thus it is easier to deal with larger data sets.

A further technical improvement of the second method is that it better separates several consecutive calls (if exist) in a single UDP flow (UDP relation).

The disadvantage of the second method is that it may not detect calls in such a network configuration, where UDP communication is completely restricted. In this case it is worth using our first method to identify calls based on TCP protocol only.

Both of our methods have the important advantage that none of them use packet payload information in the identification process.

3.1. Communication between Skype clients

The Skype client communicates not only with super nodes and dedicated servers of the Skype infrastructure. In addition, it usually maintains direct relation with several other Skype clients including mainly the logged on contacts on the buddy-list of the user, but other unknown clients as well. The relation cannot be considered as a real connection, since it consists of UDP packets only. In most of the cases the packets are originated from the default communication port of the Skype client. On the other hand, sometimes random UDP ports are used. However, in these cases the communication is terminated in the default port of the Skype client of the other party. Sometime the default ports are used on both sides.

After the client logs on to the Skype network the UDP relations are soon built up with most of the contacts on the buddy-list of the user, who are also logged on to the Skype network at that time. The UDP relation is usually already established when the user initiates a call towards one of the partners on the contact list, and remains there permanently after the call is finished. The purpose of UDP relations is likely to constantly monitor the connectivity between the two sides. The client checks whether the other party is present and reachable. It also examines whether UDP communication is available or not. Hence the packets of the UDP relation can be considered as “UDP ping messages” between the clients.

Note that if a call is indeed initiated between the two sides the same UDP relation is used with the same communication ports, and only the intensity and size of packets change significantly. Therefore the early preparation of a call is also a function of an UDP relation. The clients can build up the connection beforehand the call is initiated which speeds up the establishment of the call. However, the UDP relation is always built up if UDP communication is not restricted – by a firewall, NAT, etc. – irrespectively of the fact whether a call is initiated later or not.

The clients can also earlier recognize if UDP communication is restricted and try TCP or relayed connection.

When the UDP relation does not transfer a call it has well defined characteristics, which makes it possible to construct a robust identification method for UDP relations. If several UDP relations are found for a certain Skype client we can reliably determine the Skype communication ports used by that client.

It is clear from the above description that the separation of call sessions and inactive periods is not trivial within a UDP relation. In the flow level traffic information an UDP relation appears as a single UDP connection. Thus it is necessary to accurately determine the beginning and the end of a call (or several consecutive calls) within the UDP connection. This can be performed by using the related packet level database. Fortunately, speech packets and "UDP ping packets" have distinct sizes, which facilitate the identification of call sessions and inactive periods.

3.2. Identification of UDP relations

A UDP relation has well defined and distinct characteristics when it conducts a voice call and when it is idle. The two states can be separated based on the size of packets. According to our experiments in case of a voice call the average voice packet size varies from 70 bytes to as high as 320 bytes. On the other hand – when the relation is idle – the size of packets (UDP pings) is always less than 60 bytes.

According to our widespread analysis Skype UDP relations can be detected by the following simple identification method consisting of three steps:

1. Select each UDP flow which has more than 10 packets whose source or destination port does not belong to a well-known application.
2. For the remaining flows calculate main mode of the inter-arrival times of data packets smaller than 60 bytes.
3. The flow is likely a UDP relation if the main mode equals to 20 seconds.

The first rule is applied in order to get rid of flows, which unambiguously cannot be signaling flows and so reduce the computational time needed to verify the 2nd rule. By the 1st rule all flows are discarded which do not contain enough packets to be an UDP relation, or has a source or destination port of a well-known application (typically DNS queries and responses).

UDP relations have a specific time behavior: packet arrivals show a certain periodicity. For this reason the inter-arrival time of UDP ping messages was found to be the most characteristic property of UDP relations – in addition to packet size, which was applied as a filter in the previous step. The whole process of the detection of UDP relations is depicted in Figure 2.

A Skype client establishes several UDP relations. The number of such relations depends on the number of logged on users on the buddy-list of the user. In many cases, however, we observed more UDP relations than the number of users on the list, which suggests that UDP relations are established with foreign Skype peers as well. This behavior, fortunately, only helps identification.

UDP ping messages have a specific inter-arrival time, which is generally equal to 20 seconds on average. To avoid the error resulting from the deviation of the inter-arrival time the histogram of the inter-arrival time is calculated, and the main mode of this histogram is selected. The main mode is defined as the most frequent item of the histogram.

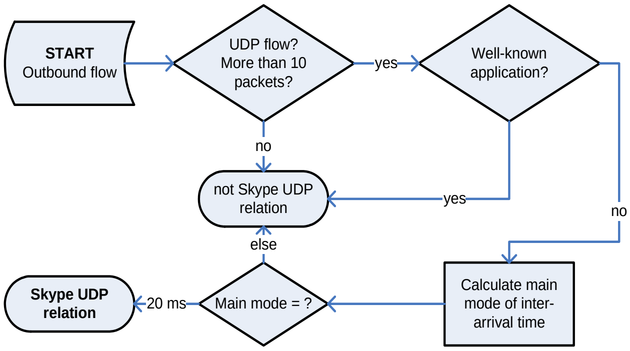

Figure 2. Detection of Skype UDP relations between Skype hosts

Although the main mode of the inter-arrival time sometime differs from the value of 20 seconds, we could detect at least a few UDP relations with a main mode of 20 seconds in all examined Skype clients.

From a detected UDP relation we can determine the IP address of the Skype host and the default communication port of the Skype client, especially if numerous UDP relations are discovered for a certain IP address and port pair.

3.3. Identification of calls

In the previous section we identified Skype UDP relations, which resulted in a list of IP address and communication port pairs. The IP address is the network address of the host computer, while the port is the randomly selected communication port of the Skype client, which is normally used for both UDP and TCP protocol based voice calls.

All flows originated from one of the IP-port pairs are likely generated by Skype. Some of these flows are the UDP relations already identified. Other flows may contain a few UDP relations which could not be detected and other types of Skype communication as well. Flows conveying voice calls are also originated from one of the IP-port pairs.

Most of the calls are based on UDP protocol and embedded in a UDP relation as described in Section 3.1 – i.e. there are one or more sections in the UDP flow (UDP relation), when the relation is not idle but conveying a voice call.

A small part of calls are transmitted over TCP. It is also possible that voice in one direction is sent over TCP and the reverse direction is served by UDP. This is a lucky situation, since at least one direction of the call can be detected based on the list of IP-port pairs of UDP relations.

However, if UDP communication is completely restricted by a firewall or a NAT device that makes identification of calls very problematic. In such a situation no UDP relations are presented at all since we cannot determine the default communication port of the Skype client. Voice calls are transmitted over TCP in both directions.

Whatever transport protocol is assumed clear determining of the beginning and the end of calls are necessary. Calls are rarely begins and ends at the same time when the corresponding UDP or TCP connection starts or finishes. This problem is obvious in the case of UDP calls. However, it also applies for TCP connections. In addition, finding the section(s) within a TCP or UDP connection where a call exists indeed is also required to accurately calculate some characteristic properties of the call (e.g. holding time, bandwidth, packet rate, etc.). These properties are needed for the identification of Skype calls.

3.4. Communication protocols

Skype prefers UDP as the primary transport protocol, and switches to TCP when UDP communication is restricted. It adapts quickly to changing network conditions by switching voice codec and transport protocol even in the middle of a call. Several scenarios are possible for establishing a voice call:

1. UDP protocol is used in both direction
2. UDP is used in one direction, TCP in the other
3. TCP is used in both direction
4. Switching communication protocol in one or both direction from UDP to TCP at the middle of the call.

The 1st and the 2nd cases are covered by the robust identification method based on UDP relations. The 3rd case is quite problematic, since it is possibly the consequence of the complete blocking of UDP communication. In this case the identification based on UDP relations does not work and we suggest to use our first identification method introduced in [1]. However, this method is somewhat less reliable, since the communication port of the client cannot be identified, only the source IP address. These TCP based calls are very rare and these calls do not modify the results and the statistics significantly. The 4th scenario happens when dramatic change occurs in the network conditions. In most cases the client usually adjusts codec parameters only. However, if UDP protocol is no longer available, the client switches to TCP. Nevertheless, this behavior could be proved in our simulations only and we suppose that this case almost never happens in an average actual use.

3.5. Call properties

The final decision whether a flow (precisely a flow section) is a Skype call or not is based on the following calculated properties of the section:

- Bandwidth (total transmitted bytes divided by the holding time of the call)
- Packet rate (total number of transmitted packets per holding time)
- Average packet size
- Main mode of inter-arrival time of voice packets

ISAC and iLBC codecs are used in both TCP and UDP cases. Both codecs adapt their transfer rate and packet size to the available link capacity. Consequently we can only set up a lower and an upper threshold as preliminary filter conditions for voice flows. According to our experiments the average voice packet size varies from 40 bytes to as high as 320 bytes, while a speech flow in one direction has a bandwidth of 20 Kbit/sec to 80 Kbit/sec. Therefore we defined a loose upper bound of 400 bytes for packet size and 100 Kbit/sec for flow bandwidth. Flows failing to match any of these criteria are discarded.

In order to discover real Skype flows we wanted to find some more characteristic properties. Skype codecs have basically constant bit rate, even if the parameters of the codec, like packet size, bit rate, inter-arrival time, might be dynamically modified as a reaction to high delay, jitter or packet loss. The inter-arrival time of voice packets was either 30 ms or 60 ms in all measurements, which results in a packet rate of 33 or 16 packets per second, respectively. In case of a TCP connection and obsolete Skype clients (Linux versions) we also detected an inter-arrival time of 20 ms (50 packets per second). Finally, the packet rate of 50 was not considered, since it would induce so many false positive errors, and would result only a few (if any) right hits. These values were confirmed by several other studies [7, 11] as well.

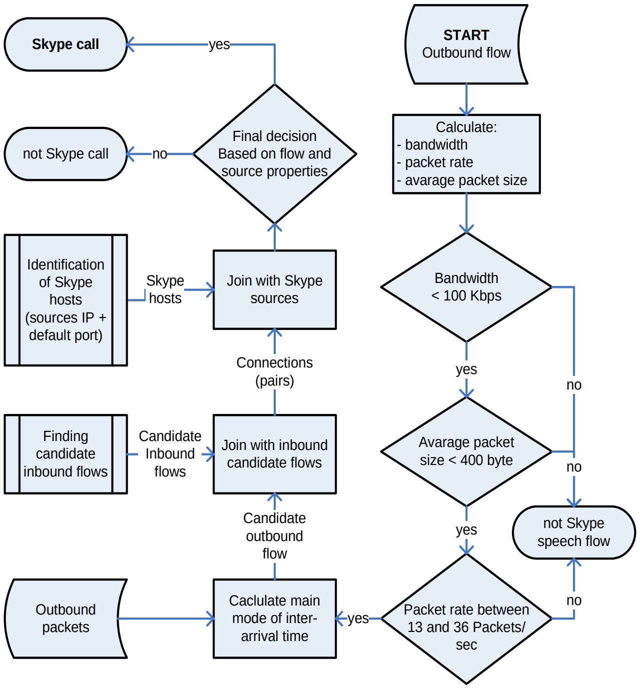

Figure 3. Identification process of Skype calls

Fortunately, the packet rate can be calculated and checked at flow level, knowing the arrival time, end time and the number of packets in the flow. Thus flows which do not correspond to this condition can be discarded. However, we cannot expect that packet rate will be exactly 33 or 16 for all Skype flows, which makes identification of speech flows somewhat problematic. The main reason is that there is some transient behavior at the beginning of the session, when the bandwidth, packet rate (and packet-size) differ significantly from the properties in steady state. The used codec and the occupied bandwidth might also change during a call when necessitated by the changes in network conditions. Apart from these, packet rate is still a suitable property to decrease the number of candidate speech flows. A rate of 13 packets/sec is chosen as a lower bound and 36 packets/sec as an upper bound. Flows not corresponding to the packet rate condition are discarded.

The efficiency of identification can be improved by using not only flow-level properties, but including some packet-level characteristics as well. We found inter-arrival time as the most characteristic property. We calculate the histogram of inter arrival times for each remaining flow and mark the main mode of the histogram.

Afterwards inbound and outbound flows are paired to one another to create voice sessions. The terms of pairing are the following: arrival time and end time of inbound and outbound flows are required to be close to one another, and also source address, source port, destination address and destination port should correspond to each other.

If the inbound and outbound directions of a voice session are served by different TCP connections the similarity of source and destination ports is not required.

In the last step those connections are selected for which the main mode of both inbound and outbound flows has a value of 20, 30 (ms) and source IP-port pair is among the previously identified Skype client IP-port pairs. The whole identification process is shown in Figure 3.

It is possible that some non-Skype flows meet some of the conditions. However, it is unlikely that flows other than Skype (even flows generated by other VoIP applications) meet all the conditions. In addition, the list of Skype sources (together with source port), which was identified in a previous step, is also used to avoid misdetection.

4. TRAFFIC MEASUREMENTS

Two traffic measurements were conducted; the summary of the data sets is presented in Table 1.

Table 1. Summary of the collected datasets

Data sets	Time of measurement From - To	Number of flows	Total traffic
Callrecords 1	22nd Jul. 2005 11h 23rd Jul. 2005 11h	23 796 956	458.04 GB
Callrecords 2	25th Apr. 2006 11h 26th Apr. 2006 11h	36 896 516	766.02 GB
Verification	7th Nov. 2006 10h 8th Nov. 2006 16h	1 663 752	61.42 GB

The first two measurements (called *Callrecords 1 and 2*) were carried out at one of the largest Internet providers in Hungary. In the chosen network segment, the traffic of about 1000 ADSL subscribers is multiplexed before entering the ATM access network. The logging was performed in one of the routers at the border of the access and the core networks. Further details of the measurement configuration are presented in [12].

In the third measurement (*Verification*) the traffic of our university department was logged, carrying the traffic of about one hundred users. We performed this experimental traffic logging to validate our Skype identification method.

In both measurements only IP and TCP/UDP headers were logged. Flow level information was extracted from the traces including source addresses, ports, packet number, transmitted bytes, start time and end time of the flow. Packet level information (packet size, packet arrival-time) was also preserved and used for the identification.

Both inbound and outbound traffic were logged since data from both directions is necessary for accurate identification. However, our method can also be applied if only one direction is available. In this case the reliability decreases since inbound and outbound speech flows cannot be paired to each other. Therefore, we recommend to use our method in edge routers where inbound and outbound traffic flows through the same router. This is not necessarily true in backbone routers because of asymmetric routing.

5. VALIDATION

The validation of the identification algorithm raises a couple of questions. For an exhaustive validation of our algorithm we need a large number of verified Skype signaling and voice flows from several clients. It is not easy to build such a managed environment.

The purpose of this validation is to verify the parameters of the identification method. These parameters were determined based on several local measurements on single computers in different types of network environments (e.g. LAN access, ADSL, dial-in access, etc).

We carried out an experimental traffic measurement in our university department. After the logging has finished we interviewed all the colleagues whether a Skype client was running on their computer and whether they made any calls during the logging period. In addition, we also collected all the history logs of the clients which contain exact information of the calls, e.g. date, time and call duration.

Then we applied our identification method to the experimental data set to detect Skype hosts together with Skype calls. Based on the comparison of detected Skype hosts and known Skype hosts from the user feedback we state that both host and voice call identification methods work well. Especially, the signal flow identification method got good marks: we could not observe any mistakes. Update connections were also detected in most of the cases. Login-, Buddy-list- and SN connection were rarely identified.

All Skype calls extracted from history logs were detected as well. We did not experience any false positive or false negative mistakes. False positive means that a non-Skype flow is mistakenly identified as Skype, while false negative means that a real Skype flow is not detected.

The validation study, however, cannot be considered as an exhaustive verification of the identification methods, since all Skype voice calls were made in an ideal network environment (100 Mbit Ethernet). As a result we suppose that always the best-quality Skype codec was used by the clients.

In addition, we also created a comparison table of commonly used VoIP applications. Table 2 shows the characteristic flow and packet level properties of the codecs used in these applications. As Table 2 confirms the characteristic properties of Skype differ from the properties of other VoIP applications.

Therefore it is very unlikely that our algorithm will mistakenly identify other VoIP flows as Skype.

Table 2. A comparison of characteristic parameters for different VoIP applications

	Gtalk	MSN Messenger	Yahoo Messenger	AOL Messenger	Skype
Average bandwidth (Kbps)	113	60	112.1	321.4	35-45
Average packet size (byte)	166.2	94.5	166.7	165.1	100-200
Packet rate (1/sec)	25.56	50.3	24.03	31.83	32.61
Packets inter-arrival time (sec)	0.038	0.020	0.041	0.033	0.031

6. TRAFFIC ANALYSIS

In this section we present the results of our analysis of two datasets, called *Callrecords 1* and *Callrecords 2*. The number of detected calls in both datasets was relatively low; therefore the two datasets were aggregated in some cases to increase the number of samples.

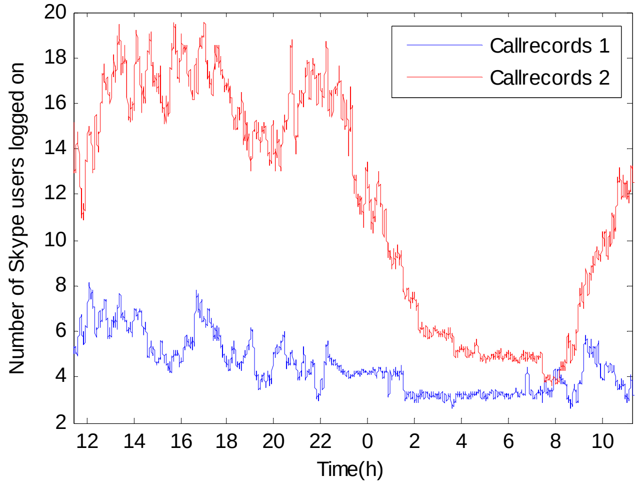

Figure 4. The daily fluctuation of Skype users based on detected UDP relations

In Figure 4 the daily fluctuation of Skype users is presented, based on the detected UDP relations. The curves are a bit smoothed. Users not sustaining visible UDP relation – probably because UDP traffic is blocked on their computer – cannot be taken into account, therefore the real number of logged-on Skype users can be somewhat higher. According to Figure 4 and Figure 5 we can see that *Callrecords 1* contains much less calls and active users than the other dataset.

We can realize that the number of Skype users logged on to the network follows the general daily tendency of the total number of users, which suggests that a certain ratio of users use a Skype client at home. Some users seem to keep their computer switched on during the night period.

The total number of active calls (Figure 5, 6) also follows similar daily fluctuation. Calls are coming more frequently in the daytime, though we can also recognize some surprising activity in the 1.00-6.00 AM interval, which suggest some "night birds" among the users or overseas calls.

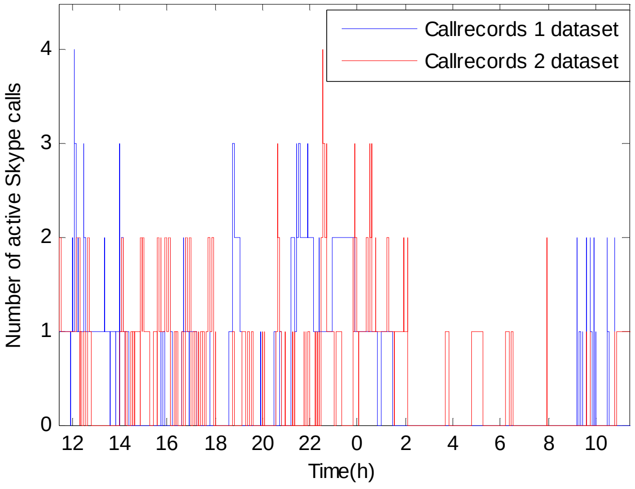

Figure 5. The number of active calls in the system

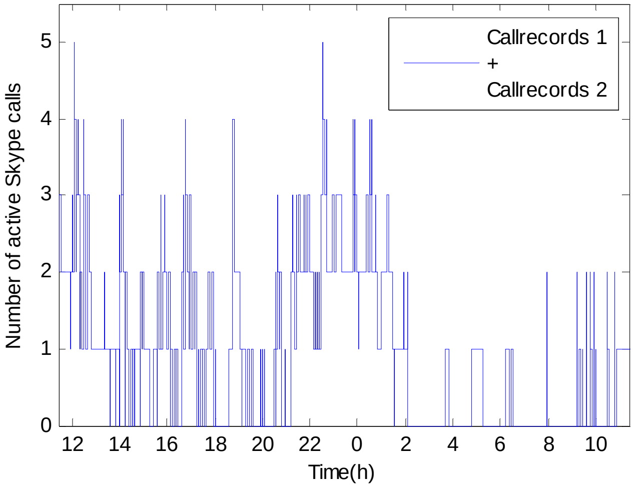

Figure 6. The number of active calls in the system (the two datasets are aggregated)

The calls seem to be shorter in the daytime and definitely longer in the 21h PM-01h AM period, which could be reasonable, because the users have more free time for chatting at night. However, we could detect only about 130 calls during the 24 hour period. For this reason we do not want to draw far-reaching general consequences.

The daily fluctuation of speech hours in Figure 7 also confirms the assumption that the calls are longer at the late night period and shorter at daytime. Thus it seems that the call activity and the busy hours of Skype are different from the pattern experienced in PSTN networks.

There is only a small ratio of active Skype users who initiate calls indeed. Most of the users seem to prefer chat service or just to stay connected and reachable if needed.

The next two figures (Figure 8, 9) show the bandwidth and the packet rate of the detected Skype calls. Figure 8 shows that the bandwidth of Skype calls is usually between 18 and 70 Kbps, typically around 40 Kbps. Figure 9 shows one prominent and one small peaks in the histogram of the packet rate of Skype speech flows which correspond to the typical inter-arrival times (30 and 60 ms). It can be seen that packet rates smaller than the typical ones (16 and 33 packets/sec) also occur. The reason for this is that the termination of a flow cannot be determined accurately in some cases, and the codec may switch rate at the middle of a call.

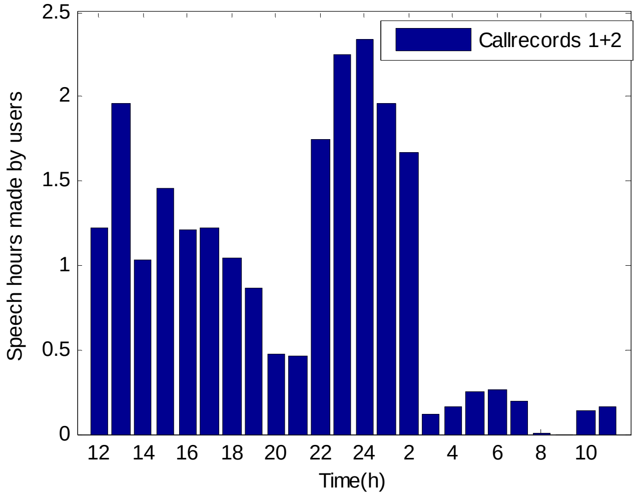

Figure 7. Daily fluctuation of the total speech hours in the system(the two datasets are aggregated)

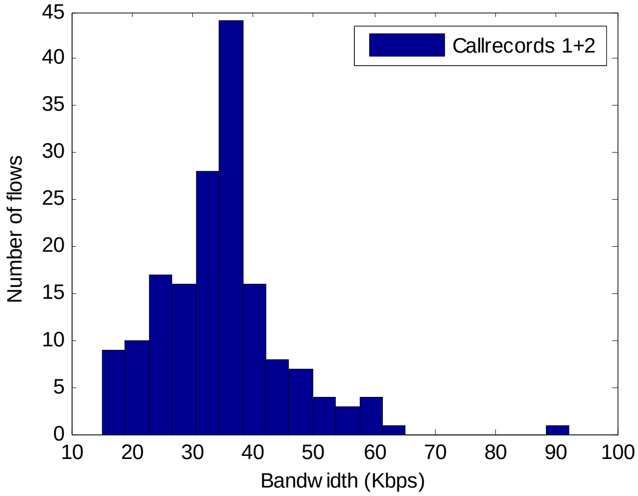

Figure 8. Histogram of the bandwidth of Skype calls in one direction in the aggregate dataset

The average packet size of Skype speech flows is plotted in Figure 10. The figure shows that the typical packet size (including IP and TCP/UDP headers) is somewhere between 100 bytes and 200 bytes, which is also confirmed by our test measurements. Smaller packet size – and bandwidth – occurs in one direction when separate inbound and outbound TCP flows belong to the call.

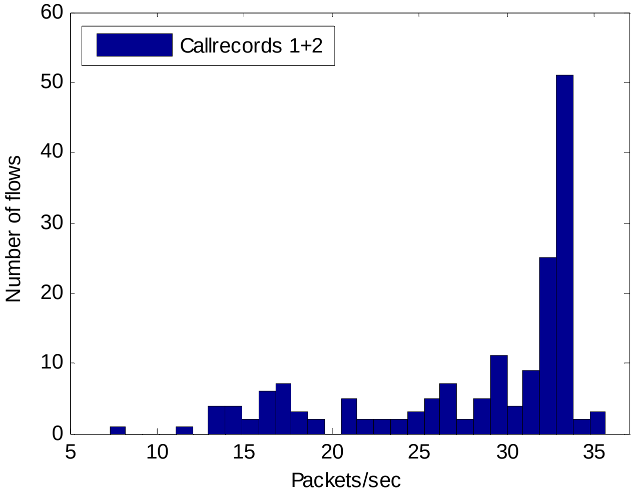

Figure 9. Histogram of the packet rate of Skype calls in one direction in the aggregate dataset

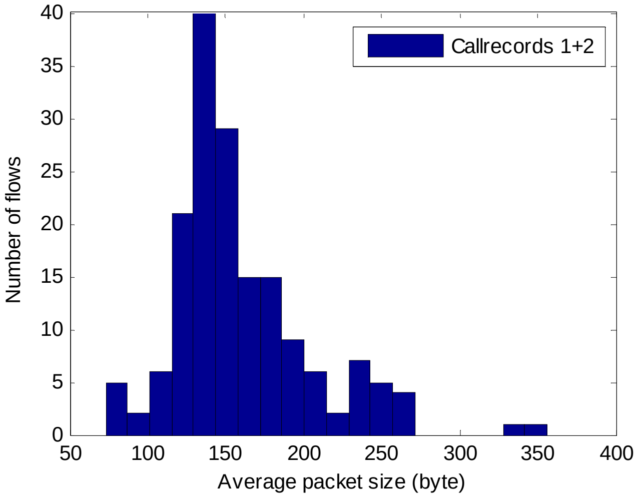

Figure 10. Histogram of the average packet size of Skype speech flows in the aggregate dataset

Figure 11 shows the histogram of the duration of Skype calls. Because of the few samples (about 170 calls in sum) it is hard to determine the exact distribution, but it seems to be an exponential-like distribution.

The following figures depict the correlation between the previous characteristic properties of Skype data flows and voice packets. Figure 12 shows an approximately linear relationship between bandwidth and average packet size of Skype flows. Each data point corresponds to a Skype flow (in outbound direction). One can see that all the points are on or over the linear line which has a gradient corresponding to an inter-arrival time of 30 ms. Data points over the line have higher average inter arrival time (between 30 and 60 ms). This figure tallies with the observed behavior of the Skype codec.

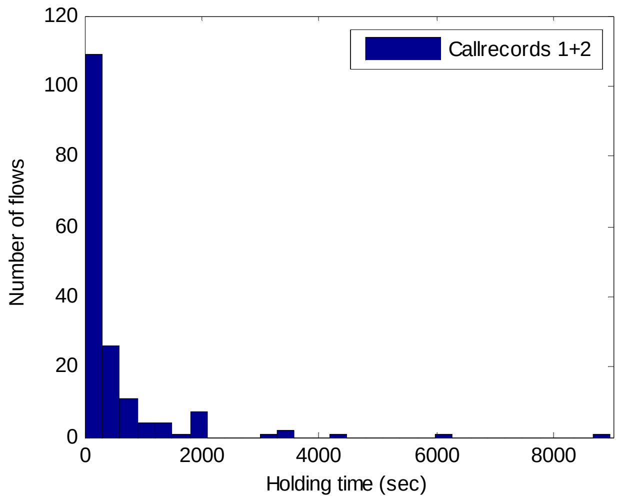

Figure 11. Histogram of the duration of Skype calls in the aggregate dataset

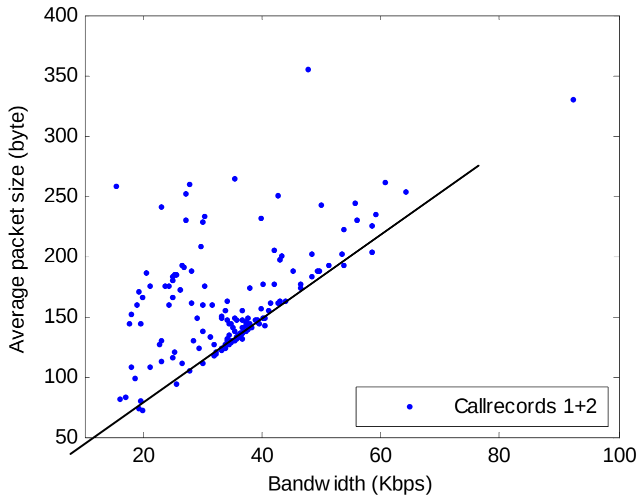

Figure 12. Average packet size as a function of bandwidth of Skype calls (one direction)

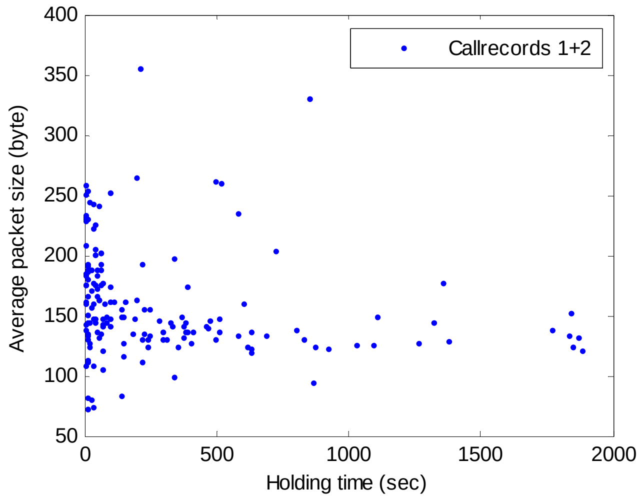

Figure 13. Average packet size as a function of the holding time of Skype calls

Figure 13 shows the average packet size as a function of the holding time of Skype calls. One can see the obvious fact that the variation of packets size decreases as the holding time of the call (and the number of samples) increases. In addition, that high dispersion of packet size in case of short calls can be explained by transient behavior upon call build-up. The typical average packet size of long holding time calls is about 140 bytes.

Figure 14 proves that the Skype codec can adjust the voice quality not only altering the packet rate but by changing the size of voice packets. In the figure the two vertical lines indicate the typical values based on our observations.

Figure 15 draws similar conclusions for bandwidth as Figure 13. Figure 15 shows that the typical bandwidth of long holding time calls (in one direction) is about 36 Kbps.

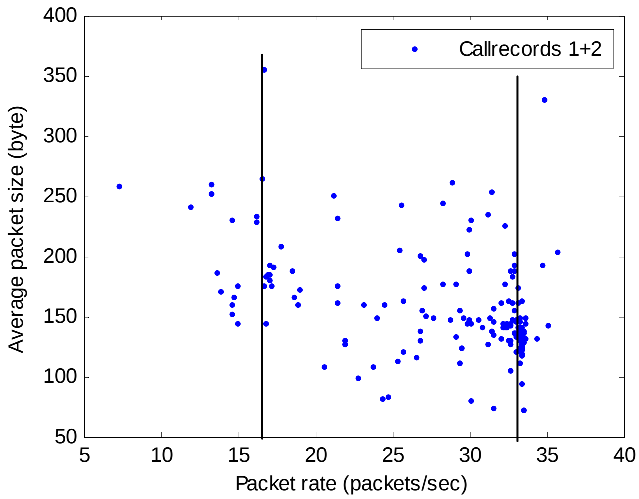

Figure 14. Average packet size as a function of the packet rate of Skype calls

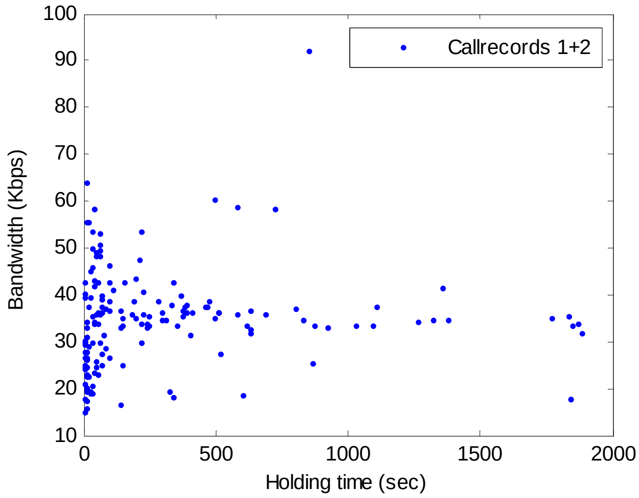

Figure 15. Bandwidth of Skype calls (in one direction) as a function of the holding time

Figure 16 shows the correlation between packet rate and bandwidth of Skype calls (in outbound direction). One can see two dense areas in the figure corresponding to 16/33 packets/sec and 20-30/35-45 Kbps, respectively, as indicated in the figure by ovals. The two horizontal lines in the figure indicate the typical packet rates of Skype flows.

Figure 17 depicts the packet rate as a function of holding time of Skype calls (in outbound direction). The figure also shows that there is a transient section at the beginning of each call where the parameters are not stable. It clearly infers that the longer a Skype call last, the easier can be detected. The figure also confirms our observations that Skype starts a call with a packet rate of 16 packets/sec, but in a little while (if network conditions are satisfactory) switches and stabilizes at 33 packets/sec.

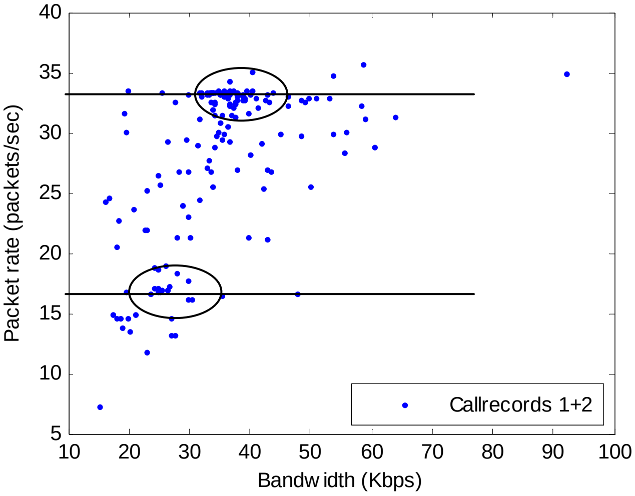

Figure 16. Packet rate as a function of bandwidth of Skype calls (in one direction)

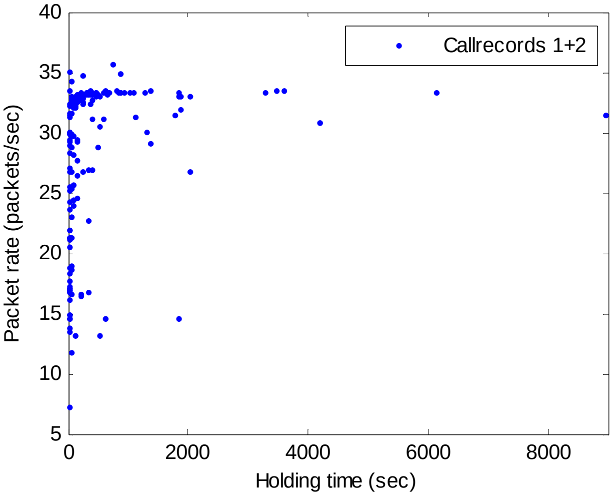

Figure 17. Packet rate as a function of holding time of Skype calls (in one direction)

7. CONCLUSION

We proposed a novel Skype identification algorithm based on observable behavior of the Skype protocol. First, candidate Skype hosts are detected using traditional IP and port-based detection together with the identification of special UDP relations between Skype peers. The identification of UDP relations allows the accurate determination of the randomly selected communication port of each Skype client. Skype calls are then discovered by exploiting the properties of speech flows, timing of voice packets and source host information found in the first step. The algorithm uses only packet headers and the extracted flow-level information and no packet payload information is necessary. It expects logged (offline) data as input.

We also presented the validation of the identification of the algorithm based on a test measurement in our department. We also showed traffic analysis results of two 24h real data sets measured from an ADSL domain in Hungary.

Our future work addresses the real time implementation of the proposed algorithm and a large-scale measurement study in different network environments as well.

8. ACKNOWLEDGEMENT

The authors are grateful to Ericsson Hungary Ltd for the financial support and to P. Varga and L. Kovács for their help in the traffic measurements.

9. REFERENCES

[1] M. Perényi, A. Gefferth, T. Dinh Dang, S. Molnár, “Skype Traffic Identification”, accepted to GLOBECOM 2007 and based on the downloadable technical report: M. Perényi et al. Identification and Analysis of Skype Traffic, http://yossarian.tmit.bme.hu/download/techrep.doc

[2] Fabrice Desclaux, “Skype uncovered – Security study of Skype”, *EADS*, 2005

[3] S. Ehlert, S. Petgang, “Analysis and Signature of Skype VoIP Session Traffic”, *Technical Report NGNI-SKYPE-06b*, Fraunhofer FOKUS, Berlin, Germany

[4] W. Ghandour, “Blocking Skype Using Squid and OpenBSD”, *Help Net Security (www.net-security.org)*, 2005

[5] K. Suh, D. R. Figueiredo, J. Kurose, D. Towsley, “Characterizing and Detecting Skype-Relayed Traffic”, *in Proc. of INFOCOM'06*, Barcelona, Spain, 2006

[6] S. Guha, N. Daswani, R. Jain, “An Experimental Study of the Skype Peer-to-Peer VoIP System”, *in Proc. of IPTPS'06*, Santa Barbara, USA, 2006

[7] Kuan-Ta Chen, Chun-Ying Huang, Polly Huang, Chin-Laung Lei, “Quantifying Skype User Satisfaction”, *in Proc. of SIGCOMM*, Pisa, Italy, 2006

[8] Skype Technologies S.A., “Skype - Guide for Network Administrators Version 1.01”, 2005, http://www.skype.com/security/guide-for-network-admins.pdf

[9] P. Biondi, F. Desclaux, “Silver needle in the Skype”, *EADS*, 2006

[10] S. A. Baset, H. Schulzrinne, “An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol”, *in Proc. of INFOCOM'06*, Barcelona, Spain, 2006

[11] B. W. Wah, B. Sat, M. Gannon, “Analysis and Evaluation of the Skype and Google-Talk VoIP Systems”, *Multimedia and Expo 2006*, Toronto, Canada, 2006

[12] T. Dinh Dang, M. Perényi, A. Gefferth, S. Molnár, “On the Identification and Analysis of P2P Traffic aggregation”, *in Proc. of Networking 2006*, Coimbra, Portugal, 2006