INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2011; **24**:94–117
Published online 21 April 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.1142

On the identification and analysis of Skype traffic

Sándor Molnár and Marcell Perényi*, †

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary

SUMMARY

Skype applies strong encryption to provide secure communication inside the whole Skype network. It also uses several techniques to conceal the traffic and the protocol. As a consequence, traditional port-based or payload-based identification of Skype traffic cannot be applied. In this paper, after an overview of the Skype P2P system, network entities and operation, we introduce novel algorithms to detect several types of communications (including voice calls primarily) that the Skype client initiates toward dedicated servers of the Skype network and other peers.

The common point in these algorithms is that all of them are *based on packet headers only* and the extracted flow level information. We do not need information from packet payloads. The identification methods allow us to discover logged on Skype *users and their voice calls*. The whole identification process is scripted in Transact-SQL; it can thus be executed automatically on a prerecorded (offline) data set. We present identification results, analysis and comparison of data sets captured in *mobile and fixed networks*. We also present the validation of the algorithms in both network types. Copyright © 2010 John Wiley & Sons, Ltd.

Received 5 February 2009; Revised 14 February 2010; Accepted 2 March 2010

KEY WORDS: Skype; traffic identification; flow-dynamics; analysis

1. INTRODUCTION

The Skype network is 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), video conferencing (even in high quality) and file transfer are also supported inside the Skype infrastructure. Skype also offers a public application programming interface (API), which can be used by third-party applications for communication over the Skype infrastructure.

We considered the Skype traffic from a network operator's point of view. The operators 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.

Skype usage is especially of great interest for mobile service operators. By the increasing number of smart phones, more and more users have the option to choose between traditional phone calls and the rapidly spreading VoIP services. Operators usually apply a flat-rate pricing in the case of

*Correspondence to: Marcell Perényi, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary.
†E-mail: perenyim@tmit.bme.hu

Contract/grant sponsor: NKTH-OTKA; contract/grant number: CNK77802
Contract/grant sponsor: János Bolyai Research Scholarship of the Hungarian Academy of Sciences
Contract/grant sponsor: Ericsson Hungary Ltd

data transfer, which makes internet-based telephony even more cost efficient. It is indispensable for network operators to know how many users use VoIP applications (e.g. Skype as the most widespread one) and how much they talk. This way they can make a tradeoff between per minute-based pricing of traditional calls and per megabyte-based (or even flat-rate) pricing of VoIP calls. In the current situation, this balancing is crucial for an operator.

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 [1].

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.

We were investigating more (though not all) Skype versions up to the time of the measurements and found that the proposed identification method worked well. However, it is worth to mention that Skype introduced a completely new codec in a later version (after our study was done), where inter-arrival time of voice packets seemed not so characteristic anymore. This renders the call identification more problematic.

In the following subsections, we will give a brief overview of Skype components and operations focusing on those aspects that can be used in our detection algorithms. A more detailed description can be found for instance in [2, 3].

The paper is structured as follows: in Section 1, we describe the main Skype components and operations. Section 2 gives an overview of related work. Section 3 introduces our identification methods, which are verified in Section 5. The used traffic data sets are presented in Section 4. Section 6 gives detailed traffic analysis of Skype traffic in fixed and mobile networks, whereas Section 7 concludes the paper.

1.1. Skype components

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

An ordinary node is a ‘leaf-node’ of the Skype overlay network; it is the equipment of the user used for communication. SNs are the switching elements in the overlay network, responsible for maintaining a Global Index distributed directory that allows users to find each other. Each SN keeps track of a small number (up to 700–800 according to our measurements) of ordinary nodes. SNs can also function as ordinary nodes, every ordinary node with a public IP address and sufficient resources (free CPU, memory, bandwidth capacity) being a candidate to become an SN.

The login server stores the account information of the users. It is responsible for user authentication at the beginning of the session, i.e. when the client is started. According to our observations, there are several login servers storing information in a distributed way.

The update server (212.72.49.131:80) is also contacted by the client after startup to check whether a newer version of the software is available. The indicated IP address of the dedicated Skype servers is valid at the time of writing this paper.

The so-called buddy-list server [4] is responsible for storing the contact list of the users. Although this list is stored locally on the host computer, the role of this entity is to make sure that the contact list is also available if the user logs on from another host. Therefore this server is contacted when changes in the contact list occur, or when the user is logged in from a different host than the previously used one. Known buddy-list servers include 212.72.49.142 and 195.215.8.142 [5].

1.2. Skype operation

When a Skype client is launched, it tries to establish a connection with an SN. For the duration of the session, this SN will be responsible for the client. The client first contacts several (about 20)

DOI: 10.1002/dac

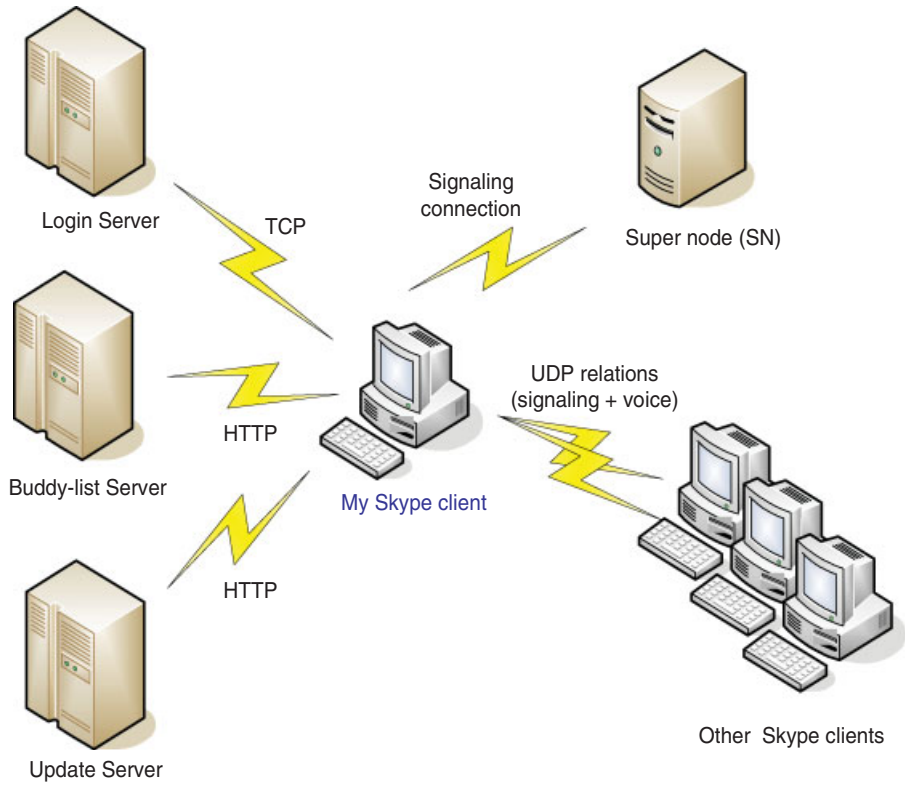

Figure 1. Elements of the Skype P2P overlay network.

SNs to investigate whether they are alive and are ready to accept the client or not. After some message exchange, the client selects one of the SNs to establish a connection.

An initial set of SNs (bootstrap SNs) is hard-coded in the executable. These are operated by Skype. Upon the first run of the client, these SNs are contacted, but a list of other SNs is retrieved and stored by the client. When the client is started for the next and all the subsequent times, the stored SNs are used.

At startup, the client also contacts one of the login servers for authentication. Although there are several login servers, we have found that Skype clients in our measurements always connected to one of the following two servers: 212.72.49.141 or 195.215.8.141. The connection to the login server might be relayed by an SN; this usually happens when the direct connection is blocked (e.g. by a firewall), but it can occur even if there is no such problem.

When the user is online, there is a periodic message exchange between the host of the user and the selected SN.

1.3. Ways of identification

There are several ways for traffic identification. The first question is whether to use payload information or not. Payload-based identification methods can reach higher accuracy. In this case, however, appropriate payload application patterns are necessary, which are not always available. Many applications (especially those carrying illegal or unwanted content) try to hide themselves by encrypting the payload. Payload-based identification is more resource intensive; it is thus not always feasible on high-bandwidth links. In case of Skype, no reliable general purpose payload pattern is available for identification. Furthermore, packet payloads are often not or hardly accessible, since network providers are unwilling to share this information. In many countries, sharing such information is against the law or hurts the internal policy of the company.

DOI: 10.1002/dac

Identification without payload information is more limited, since only packet headers and timing information can be used. However, the storage of logged data requires less disk space, which makes offline processing feasible. This is especially useful, since we can discover connections between flows in different time periods (flow dynamics). In addition, many statistical properties of flows, packets, and timings can still be monitored. In case of Skype, all communication among the network entities is encrypted and hence do not have a recognizable payload pattern; only the initial messages of the login phase have some reliable payload patterns.

Traditional port-based identification cannot be applied either, since Skype chooses a random communication port upon installation. However, once the application is installed the used port is not changed anymore, unless the user changes it manually.

For all these reasons, we decided not to use any payload information in our identification methods.

2. RELATED WORK

The identification of Skype traffic is addressed in several recent publications. Ehlert and Petgang [5] describe a method for identification using traffic patterns and payload information from the Skype login phase. To efficiently block Skype, its activity has to be identified. Many blocking methods were proposed (see e.g. [6]), but these are usually tailored to a given firewall type or security setting, and assume that Skype communication starts after enabling the blocking mechanism. Suh *et al.* [7] present a method for the detection of relayed traffic by comparing input and output traffic patterns.

Guha *et al.* [8] 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 SNs, and general traffic patterns of a single Skype session.

Chen *et al.* [9] 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 the correlation between call duration and voice quality.

The authors of [10] proposed three different techniques to identify Skype voice traffic. The first approach—based on Pearson's Chi-Square test—probes whether payload bytes are truly random or not, a condition that has to be true for encrypted traffic. Their second approach investigates stochastic characteristic properties of Skype traffic, namely packet arrival rates and packet sizes, using Naïve Bayesian Classifier (NBC). These properties are considered in our identification method, but we also involved other characteristics as well (e.g. bandwidth, main mode (defined as the most frequent item of the histogram) of inter-arrival time, and average packet size). The authors found that by combining these two techniques, they can efficiently identify Skype voice traffic. A third technique was also proposed based on deep packet inspection. However, this technique did not prove to be reliable by itself. It was paired with a companion algorithm, which looks for candidate Skype host IP and communication port. The process of looking for candidate Skype host is also an important part of our identification method. However, in our case the searching for candidate host is based on a completely different algorithm.

Paper [11] focused on several aspects of Skype source, including service types, transport protocol, and network conditions. The authors of this paper analyzed three major characteristic properties: bit rate, IPG (inter-packet gap), message size. These properties are also taken into account in our study as filtering criteria. The results of our analysis part are also consistent with their observations for codec properties. However, we concentrated mainly on ISAC codec as the prevailing codec used in our traffic measurements. The authors also present a traffic analysis of a data set captured in a campus network applying one of the proposed identification techniques (the NBC-based one) in [10].

The same authors investigated Skype signaling traffic by means of passive measurement in [12]. They classified Skype signaling traffic into three different categories: probe traffic (to discover

DOI: 10.1002/dac

new nodes), periodic heartbeat flows, and long dialog flows. Heartbeat flows are called 'UDP relations' in our notation (see Section 3.5) and have particular importance in our identification process. An analysis of signaling flow categories in terms of number of generated flows, packets, and affected foreign peers is also presented in [12].

In [13], the authors provide payload patterns that appear with high frequency in the recorded data set generated by an SN and a client. They state that, by using these signatures, Skype traffic can be effectively identified and monitored. However, there is no report on the accuracy of the method. The paper also aims to investigate the structure of the Skype overlay network, including the number and the location of SNs.

Freire *et al.* [14, 15] presented a method for distinguishing web traffic and Skype traffic passing through TCP port 80. Skype may use this spare port if the UDP transfer and default communication port are blocked. They calculated the distribution of certain characteristic parameters (excluding packet payload) of Skype and web traffic using a training data set, and classified flows by Chi-square and Kolmogorov–Smirnov tests.

In [16], the authors suggest a method for identifying BitTorrent-like file sharing applications based on the 'discreteness of remote hosts', i.e. how many hosts a client contacts from various stub networks. They state that by using this representative measure they can differentiate BT-like applications and traditional P2P applications (including Skype).

Wang *et al.* [17] aimed to identify Skype sessions in order to guarantee the bandwidth and assure the quality of service for Skype traffic. They applied the simple K-means method to form clusters and classify flows. However, it is not clear what properties they have chosen to form the feature set; also, the test data set contained very few types of traffic in addition to Skype, which makes clustering easier.

Some commercial products [18] are available for hindering the usage of Skype. These applications prevent the initial connection to the Skype network by seeking for some pattern in the packet payload. The employed patterns are kept in secret. In addition, no data are published about the efficiency of the solutions. Skype blocking was also introduced in IOS version 12.4 (4) released by Cisco in 2006 [19]. Skype can be blocked by a simple policy based on packet filtering, but the details of the algorithms are unknown. Some vendors (e.g. [20]) claim that their products are able to identify Skype, but verifications are not made public.

Traffic patterns are presented in [21] for the detection of Skype-to-Skype communication—including both calls and general (idle) chatter of the program—and Skype-to-phone (SkypeOut) connections. However, the reliability of these patterns is questionable. The patterns are reported to match at least some of the general chatter the client produces when the user is idle or is performing actual phone calls. Unfortunately, the patterns also match a significant fraction of non-Skype (and even random) traffic. This means that both false positive and false negative identifications can occur using the patterns.

The fact that several vendors come out with solutions for blocking Skype may suggest that payload-based blocking is feasible. However, these approaches differ from our objective: we do not want to hinder Skype communication, but rather to monitor and analyze certain properties of the traffic (e.g. intensity of calls and users, bandwidth of the voice calls, etc.). Therefore, obstructing Skype traffic completely by blocking the initial login phase does not serve our needs. However, payload patterns could be used to detect candidate Skype hosts, thus making call identification more reliable.

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.

DOI: 10.1002/dac

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 that seem to be invariant among different software versions. In this section, we first present methods to detect Skype activity even if no calls are made. Then we present our decision-based method for the detection of Skype voice calls.

The decision-based call identification algorithm contains basically two steps: discovering candidate Skype hosts first, and then searching for voice calls. The success of the call identification in the second step depends greatly on the reliability of host identification in the first step. There are three techniques to determine the candidate Skype hosts (described in details in the following subsections):

- Looking for *Skype specific connection*—this is problematic because these connections may be relayed by the SN, or may not be present at all;
- Looking for '*signaling flow*' between the client and the SN;
- Looking for '*UDP relations*'.

The first two techniques permit only to determine the IP address of the host, whereas the third one makes it possible to detect the used communication port too. The second and the third techniques let us detect idle Skype presence as well (i.e. an idle client who does not initiate calls), not only single events. The UDP relation-based candidate Skype host identification proved to be the most successful and reliable among the three techniques. The only disadvantage of it is that it does not work with a client for which UDP communication is completely restricted. Thus, we have to consider whether this situation is likely to occur in the network environment where the measurements were taken. In our case, we dealt with home ADSL users and mobile users where UDP communication is likely allowed. Therefore, we have chosen the UDP-based method to detect candidate Skype hosts. The overall process of the identification and the roles of the methods are depicted in Figure 2.

3.1. Filtering out known applications

The preliminary step of the identification is to filter out traffic of known, non-Skype applications. To do so, we used a database of popular applications, which contains default TCP and UDP communication ports for known applications (see Table I for example). TCP ports 80 and 443 are considered as an exception, since besides HTTP they are also used by Skype.

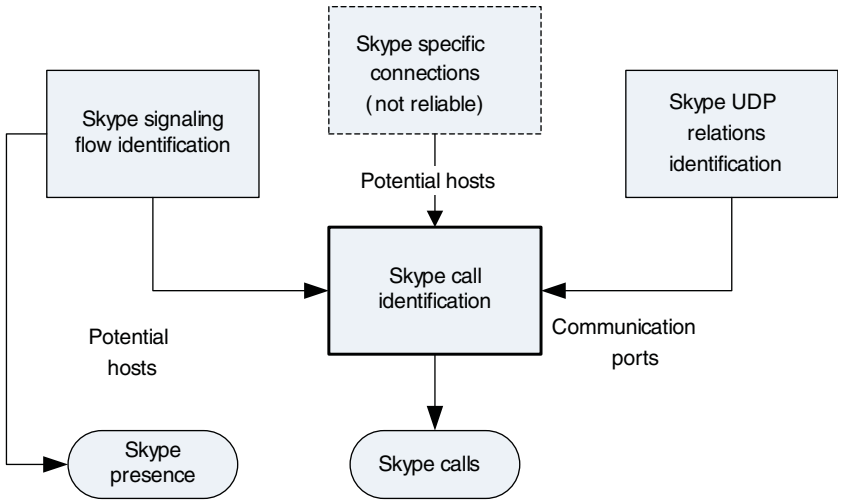

Figure 2. The overall process of Skype traffic identification and the role of the methods.

DOI: 10.1002/dac

Table I. Example: a list of name, default port and protocol type of common applications.

Application	Port	Protocol
MSN messenger	1863	TCP
NETBIOS	135, 137, 139, 445	UDP
DNS	53	UDP
NTP—Network Time Protocol	123	UDP
POP3	110	TCP
SMTP	25	TCP
FTP	20, 21	TCP
RDP—Remote Desktop Protocol	3389	TCP
SSH	22	TCP
POP3	995	TCP/UDP
POP3	110	UDP
IMAP	143, 993	TCP/UDP
…	…	…

3.2. *Skype-specific connections*

The first possible way to identify candidate Skype hosts is to look for Skype-specific connections, such as the connection to a login server, buddy-list server, or bootstrap SN. The occurrence of any of these infers the presence of Skype, since these connections are very unlikely to be initiated by non-Skype applications. On the other hand, these connections are not necessarily present (or visible) during a Skype communication:

- the *connection to the login server* is in some cases relayed through an SN and therefore invisible;
- the *connection to one (or more) of the bootstrap SNs* is necessarily attempted at the first execution of the application, but during subsequent executions the host may choose to contact other SNs;
- the *buddy-list* server is contacted only in the cases mentioned in Section 1.1.

There are also some kinds of connections that are not unique, but characteristic to Skype. We found two of these:

- the connection to the *update server* is initiated at the beginning of the session. However, the same IP and port is used, in some cases, to reach the www.skype.com web server;
- a TCP connection to *port 33033* is likely to be initiated by a Skype client, since this is the default port for SN connections.

It is very unlikely that both of these connections are present if the host does not run a Skype client. Therefore, we decided to conclude on Skype presence if both of these are found.

3.3. *Skype signaling flow identification*

It is possible that the user logged on to Skype before the traffic measurement began. In such a case, we cannot detect login, buddy-list update, or software update attempts. Furthermore, even if these connections can be observed, they give no indication of the end time of the session. Therefore, we investigated Skype network activity to find characteristic traffic patterns that last for the entire duration of the session.

A Skype client maintains one permanent connection to an SN while the user is logged on to the Skype network. At startup the client tries to establish several outbound connections to find an appropriate SN. After a few seconds, these transient connections are terminated, and only one or two permanent TCP connections remain. One of these connections has a traffic pattern that can be relatively easily identified. Data packets are limited in size and both inbound and outbound flows (belonging to the specific TCP connection) have restricted bandwidth and packet intensity.

DOI: 10.1002/dac

In addition, the timing of outgoing packets follows a well-defined pattern. The connection persists as long as the user is logged on to Skype. For these reasons, we selected this flow to be scanned in order to identify Skype clients and constructed a method to identify such flows.

In some cases, we observed that the original signaling connection was replaced by a new one with exactly the same properties. This was possibly caused by the original SN going offline because of a failure or other problem. Nevertheless, the signaling flows are generally long enough to be detected. The long duration of the signaling flow is a key issue, since the proposed algorithm utilizes statistical properties among others.

Based on a widespread analysis of Skype signaling connections, we decided to consider a TCP connection as a Skype signaling connection if it obeys all of the following rules:

1. the outbound and inbound data rates are not larger than 40 bytes/s;
2. the number of packets (in outbound and inbound directions separately) is not larger than 0.4 packets/s;
3. every packet in outbound direction is smaller than 1000 bytes (including IP and TCP headers);
4. a periodicity of 1 min is observable for outbound packets of size between 70 and 250 bytes. E.g. a certain percentage of such packets arrive in a specific, periodic time slot.

The unique time behavior of signaling flows is caught by the fourth rule. We realized that most of the outgoing data packets are rather small, except for some special packets. The exceptional packets have much larger packet size (between 70 and 250 bytes, including IP and TCP headers), and an inter-arrival time of 1 min. We assume that these packets are some periodic keep-alive messages of the client to the SN.

Unfortunately, the picture is not that clear, as some factors make the identification more difficult. E.g. occasionally out-of-period packets appear in the outbound flows, a few of them falling into the interval of 70–250 bytes. Sometimes keep-alive messages happen to drop out, though periodicity is still preserved (no shifting). In some cases (perhaps when the user has few contacts on his/her buddy-list), the size of the keep-alive messages is not significantly larger than that of other packets in the outbound flow, which results in an unclear separation of keep-alive packets. However, when the user initiates a voice call (i.e. it becomes active), the size of keep-alive messages increases and falls into the specified interval in all investigated cases. After finishing the voice call, the size of keep-alive packets usually returns to its original value.

We chose a simple method for detecting periodicity in Rule 4. The modulo 60 remainder is calculated for the arrival time (measured in seconds) of every packet (with packet size between 70 and 250 bytes), which is then rounded to the closest integer. This yields a time slot of 1 s. Then we calculate the distribution (histogram) of modulo 60 remainders (see Figure 3), and mark every

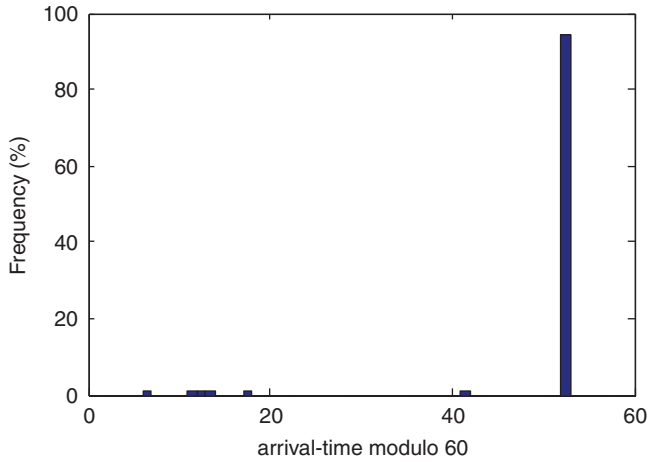

Figure 3. Modulo 60 remainders of arrival times for packets of size between 70 and 250 bytes.

DOI: 10.1002/dac

remainder where the frequency of the remainder in the histogram is over a certain threshold (0.08). The connections considered as Skype signaling are the ones for which only one remainder is over the threshold, i.e. only one remainder is significant. We chose this technique to detect periodicity, because it is adequate, fast, simple, and easy to implement in SQL. However, other techniques could also be applied (e.g. Discrete Fourier Transformation (DFT) or wavelet transformation).

A minor shifting (few 10 ms) was observed in the arrival times of the periodic keep-alive packets, which was most likely due to the inaccuracy of the internal clock of the host computer. This can cause two significant modulo 60 remainders (next to each other) to appear in the histogram. This problem can be avoided by shifting the arrival process by 0.5 s (adding 0.5 to each arrival time), and calculating the above-mentioned histogram for this new process. If the periodicity is present, then either the original or the shifted process will reveal it. The shifting of the arrival process was applied for all flows, but we could find only a few new signaling flows with this technique.

Although in theory other applications might generate similar traffic patterns, in our measurements we could not find any. Therefore, if such a pattern is found we attribute it to Skype.

3.4. Communication between Skype clients

The Skype client communicates not only with SNs and dedicated servers of the Skype infrastructure, but it usually also maintains direct relation with several other Skype clients, including mainly the logged on contacts on the buddy-list of the user, and 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. Usually 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 toward one of the partners on the contact list, and remains active permanently after the call is finished. The main purpose of UDP relations is 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 the size of packets change significantly. Therefore, the early preparation of a call is also a function of a UDP relation. The clients can build up the connection before 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 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.

It can be observed that if we initiate a voice call shortly after logging on to Skype, the call is built up slower. As opposite to this, if we begin the call after a few minutes, the other party is connected immediately and it starts ringing. The reason for this is that in the second case the UDP relation is already built up, which makes call connection time shorter.

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, a 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 and intensities, which facilitate the identification of call sessions and inactive periods.

DOI: 10.1002/dac

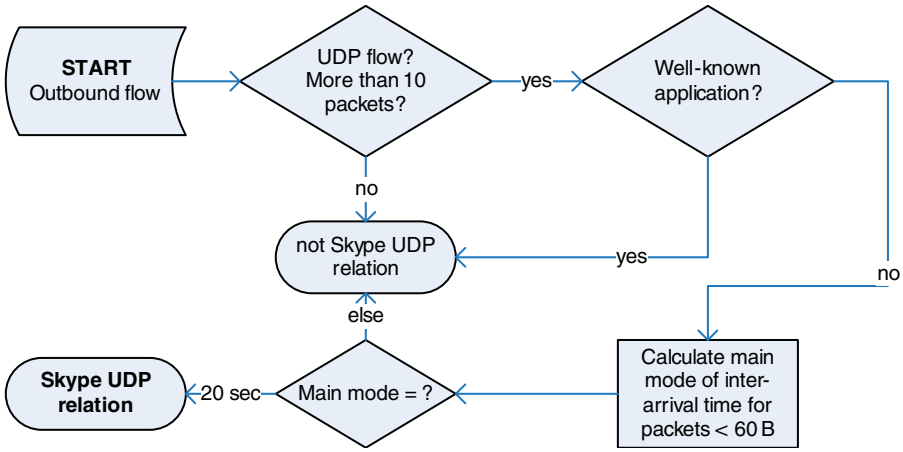

Figure 4. Detection of Skype UDP relations between Skype hosts.

3.5. *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 measurements, 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 lower than 60 bytes.

According to our comprehensive analysis, Skype UDP relations can be detected by a simple identification method [22] that consists of three steps:

1. select each UDP flow that has more than 10 packets and the 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 s.

The first rule is applied in order to get rid of flows that can be unambiguously identified as not being signaling flows; this reduces the computational time needed to verify the second rule. According to the first rule, all flows are discarded that do not contain enough packets to be a UDP relation or have 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 times of UDP ping messages were 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 identification process of UDP relations is depicted in Figure 4.

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 s 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.

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

DOI: 10.1002/dac

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.

3.6. Decision-based 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, whereas 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 that 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 the UDP protocol and are embedded in a UDP relation as described in Section 3.4—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, the identification of calls becomes very problematic. In such a situation, no UDP relations are present, and thus 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, it is necessary to clearly determine the beginning and the end of the calls. Calls rarely begin and end 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.

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

The first and the second cases are covered by the robust identification method based on UDP relations. The third case is quite problematic, since it is possibly the consequence of the complete blocking of UDP communications. In this case, the identification based on UDP relations does not work, and we suggest using our previous identification method introduced in [23]. 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 fourth scenario happens when a dramatic change occurs in the network conditions. In most cases, the client usually adjusts codec parameters only. However, if the 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.

DOI: 10.1002/dac

3.6.2. Call properties. The final decision on whether a flow (or a flow section more precisely) is a Skype call is based on the following characteristics of the section:

- bandwidth (total number of 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 no matter which transport protocol (TCP or UDP) is employed. 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 measurements, the average voice packet size varies from 40 byte to as high as 320 byte, whereas a voice flow in one direction has a bandwidth between 20 to 80 kbps. Therefore, we defined a loose upper bound of 400 bytes for packet size and 100 kbit/s 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, such as 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 or 60 ms in all measurements, which results in a packet rate of 33 or 16 packets/s, 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/s). Finally, we discarded this latter packet rate, since it would induce too many false positive errors, and would provide only a few (if any) right hits. These values were confirmed by several other studies [4, 24] as well.

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 that 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 the changes in network conditions require so. Apart from these, packet rate is still a suitable property to decrease the number of candidate speech flows. A rate of 13 packets/s is chosen as a lower bound and 53 packets/s 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 or 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 5.

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 ports), which was identified in a previous step, is also used to avoid misdetection.

The proposed techniques (e.g. time behavior of flows, relationship between flows) and properties (e.g. flow/packet level characteristics: bandwidth, packet rate, IAT) may be applied for the identification of other Internet applications as well. This can be achieved by systematic investigation of flow behavior, identifying of characteristic parameters (not exclusively those mentioned here), and 'fine-tuning' of them.

DOI: 10.1002/dac

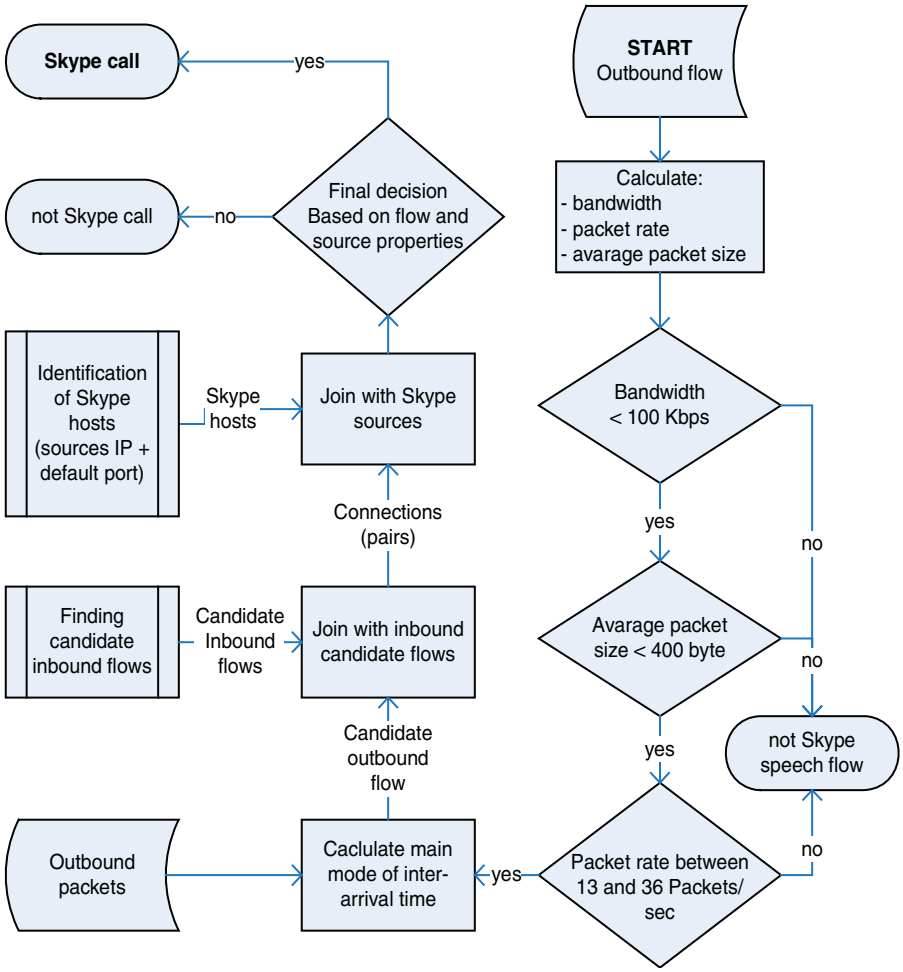

Figure 5. Identification process of Skype calls.

Some properties—which the methods rely on—assume that the identification process is executed off-line, which makes the online application of the methods—without modifications—problematic. By introducing a delay (i.e. a time window), however, all methods can be adapted. The window has to be wide enough to catch the characteristics properties of a flow. We believe that a time window of a few minutes would be absolutely satisfactory (and also acceptable for most of the online applications). The detection of potential Skype hosts is not a time-critical operation (i.e. not strictly online), whereas the identification of calls is time-sensitive.

4. TRAFFIC MEASUREMENTS

We have conducted several traffic measurements in both fixed and mobile environments, for both validation and traffic analysis purposes. The summary of the data sets is presented in Table II.

The first two measurements (called *Callrecords 1* and *Callrecords 2*) were carried out in the fixed network of one of the largest Internet providers in Hungary [25]. 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 Table II.

DOI: 10.1002/dac

Table II. Summary of the collected data sets.

Data sets	Time of measurement From–To	Number of flows	Total traffic
Callrecords 1	22 July 2005 11 h 23 July 2005 11 h	23 796 956	458.04 GB
Callrecords 2	25 April 2006 11 h 26 April 2006 11 h	36 896 516	766.02 GB
Mobile measurement	4 June 2007 18 h 24 7 November 2006 10 h	29 334 814	667 GB
Verification	7 June 2007 15 h 58 8 November 2006 16 h	1 663 752	61.42 GB
Mobile verification	9 November 2007 8 h 50 10 November 2007 7 h 47	30 634	0.8 GB

In the third fixed network measurement (*Verification*), the traffic of our university department was logged, carrying the traffic of about 100 users. We performed this experimental traffic logging to validate our Skype identification method.

We also performed Skype traffic identification and analysis on a data set recorded in the network of one of the largest mobile service providers in Hungary. The trace was captured at the border of the backbone network where the traffic of the mobile users is aggregated. Only users having 3G or HSDPA connection were included in the data set. Some of these users may be able to use Skype using a smart-phone or a laptop with 3G network adaptor. In fact, it is really worth for them to use Skype because it is a cheaper alternative to initiate voice calls. If per minute costs are too expensive compared with per megabyte cost, many users will switch to VoIP applications instead of ordinary mobile calls. Mobile service operators are very much interested in knowing about how many users choose the transferred data based (usually flat-rate) voice calls instead of the per minute (or per second) charged traditional calls. This information can be very useful for the operator to determine the tradeoff in price between the two types of services.

The *Mobile measurement* was almost 3 days long and both inbound and outbound TCP and UDP traffic were logged. The bandwidth of the users was limited in both directions (384 kbps uplink, 384 kbps or 3.6 Mbps downlink). However, according to our measurements, this limitation did not cause a serious problem for Skype, since the typical one-way data rate of speech calls (about 40 kbps) is under this limit. The aggregate traffic of all mobile users in the Budapest area is passed through the measurement point. We also performed a *Mobile verification* measurement (as previously described in Section 5).

In all 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 traffics were logged, since data from both directions are 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 on using our method in edge routers where inbound and outbound traffics flow 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.

DOI: 10.1002/dac

Table III. Identification of default communication ports at our university department.

Host name	IP address	Port	Number of detected UDP relations	Signaling flow
bme-diga	152.66.244.228	25224	266	2
pubi	152.66.244.95	2061	121	1
zed	152.66.244.226	50952	80	2
horus	152.66.244.3	21817	67	7
dzsessz	152.66.244.14	3540	66	No
dhcp76	152.66.244.76	60989	41	24
nirvana	152.66.244.222	65363	21	2
dhcp78	152.66.244.78	28474	16	1
dhcp79	152.66.244.79	7173	14	2
akkord	152.66.244.120	9623	12	1
nokya	152.66.244.2	23169	10	No
dhcp211	152.66.244.211	27661	10	No
core	152.66.244.5	18269	8	2
asus3	152.66.244.224	55635	4	3
dhcp76	152.66.244.76	61099	4	24
hegyi	152.66.244.199	50492	2	69
dhcp206	152.66.244.206	7173	2	No
dzsessz	152.66.244.14	34584	1	No
tico	152.66.244.221	50492	1	No
bme-diga	152.66.244.228	4752	1	2
bme-diga	152.66.244.228	1033	1	2
sb	152.66.244.65	5588	1	No

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 (see Section 4 for details) at our university department. The measurement contained the traffic of about 110 distinct users and a large variety of applications, including other VoIP applications as well. After the logging was finished, we interviewed all the colleagues about whether a Skype client was running on their computer and whether they made any calls during the logging period or not. In addition, we also collected all the history logs of the clients that contain exact information of the calls, e.g. date, time, and call duration.

Then we applied our identification methods 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 UDP relation-based method gives accurate results. Signaling flows were also detected in most of the cases. Update connections sometimes, login-, buddy-list- and SN connections were rarely identified. Table III shows the results of the validation measurement at our university department. The last two columns display the number of detected UDP relations and signaling flows, respectively. The three unmarked rows were not confirmed to be Skype hosts; however, two of these were DHCP hosts.

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, whereas 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.

Apart from the validation in the fixed network, we also performed the validation of our identification algorithms in a mobile environment (for details of the measurement, see Section 4). We used two laptop computers with mobile phones connected to them. The phones were connected to the base station via an HSDPA connection (384 kbps uplink and about 3.6 Mbps downlink bandwidth).

DOI: 10.1002/dac

Table IV. Identification of default communication ports in the mobile network.

Phone IMEI number	Port	Number of detected UDP relations
216301119000054	9942	35
216301119000054	4367	1
216307007161849	12613	43
216307007161849	2993	1

We had a closed test cell; no other phones were thus able to connect to the base station and affect the available bandwidth for the test phones. The application set generating the traffic was limited to Skype, MSN, Fring, web browsing, and other general applications running on Windows.

During the test, we initiated Skype calls between the clients and toward remote clients with fixed network connection. All traffic was captured in the cell and, in addition, we also conducted a Skype activity log, which contained information about calls, sign in and sign out events, including exact times, durations, caller and called parties.

The purpose of this active measurement was twofold. First, we wanted to verify our Skype identification method. Second, we wanted to investigate how Skype behaves in a mobile environment, especially when the available bandwidth is limited. To simulate this, we were able to limit the bandwidth available for users in the test cell—even at the middle of an ongoing call. We have carried out several test calls applying a bandwidth limitation of 128 and 64 kbps. We also wanted to inspect whether limited bandwidth affects the accuracy of the identification algorithm.

After capturing all traffic in the test cell, we applied the same identification algorithm as in the case of the real traffic traces. After this, we compared the identification results with the information in the Skype activity log.

The first step of the algorithm proved to be successful (similar to the case of the fixed network environment). We succeeded in detecting the default communication ports (Table IV) of both clients with high confidence. In addition, we also detected two other ports used by Skype, which are not the default ones. However, this is not considered as a problem here because we only use these ports to look for voice flows originated from one of these ports. Voice flows are originated from the default communication port in case of UDP transmissions and in most of the cases of TCP transmissions.

The second—voice call identification—step of the algorithm successfully identified 12 of the 14 Skype calls. The two unidentified calls were among those where the bandwidth was limited. Limitation (or bad network conditions) may cause interruption in the transmission of voice packets. If there is a longer interruption (timeout), the algorithm cannot accurately determine the beginning and the end of calls, and it may mistakenly think that the call has finished. We experienced no false positive identifications.

As one of the reviewers indicated, there is an arms race going on between Skype developers trying to avoid detection and blocking and identification methods and tools striving to uncover the hidden Skype traffic. Characteristic properties exploiting long-term periodicity (e.g. 60 s for signaling flows and 20 s for UDP relations) are not resistant enough if Skype developers decide to randomize these time periods. On the other hand, it is much more difficult to randomize short-term periodicities and packet-level properties of the Skype traffic since these are reflecting the characteristics of the applied voice codec. To randomize the inter-arrival time of the voice packets, extra delay has to be introduced. If periodicity is eliminated from a traffic flow to conceal it, the inter-arrival time can still be characterized by a distribution (instead of a constant value).

6. TRAFFIC ANALYSIS

In this section, we present the results of our analysis regarding two data sets (called *Callrecords 1* and *Callrecords 2*) captured in a fixed network and one data set captured in a mobile network

DOI: 10.1002/dac

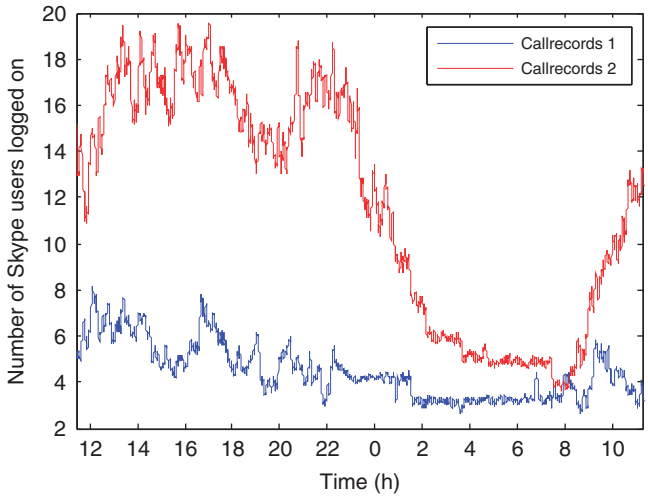

Figure 6. Daily fluctuation of Skype users based on detected UDP relations in Callrecords 1 (upper curve) and Callrecords 2 (lower curve) data sets.

(*Mobile measurement*). Since the number of calls in the fixed network data sets was relatively low, these two data sets were aggregated in some cases to increase the number of samples.

In the analysis, we investigated the general daily profile of call and user activity, the characteristic properties of voice flows, and also the relation between these properties, which makes it possible to diagnose some interesting behavior of the Skype codec.

6.1. Daily profiles and call activity

In Figure 6, the daily fluctuation of Skype users is presented, based on the detected UDP relations. The number of Skype users includes here all the users who are logged in the Skype network, even if they do not initiate any call. The curves are a bit smoothed. Users not sustaining visible UDP relations—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. The other reason for the low ratio of Skype user is that Skype usage was not widespread when the measurements were taken. According to Figure 6, we can see that *Callrecords 1* contains much less calls and active users than the other data set.

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 the Skype client at home. Some users seem to keep their computer switched on during the night period.

The number of voice calls (Figure 7) also follows a similar daily fluctuation. Calls are coming more frequently in the daytime, though we can also recognize some surprising activity in the 01–06 h AM interval, which suggests some ‘night birds’ among the users or the presence of overseas calls.

The calls seem to be shorter in the daytime and definitely longer in the 21 PM–01 h 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 h period in the fixed network. For this reason, we do not want to draw far-reaching general conclusions based only on these results.

The daily fluctuation of speech hours (Figure 8) is constructed as follows: the measurement period was divided into smaller, 1 h long intervals and the sum of speech time by all users was calculated for each interval. The daily fluctuation of speech hours 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 the chat service or just to stay connected and be reachable if needed.

DOI: 10.1002/dac

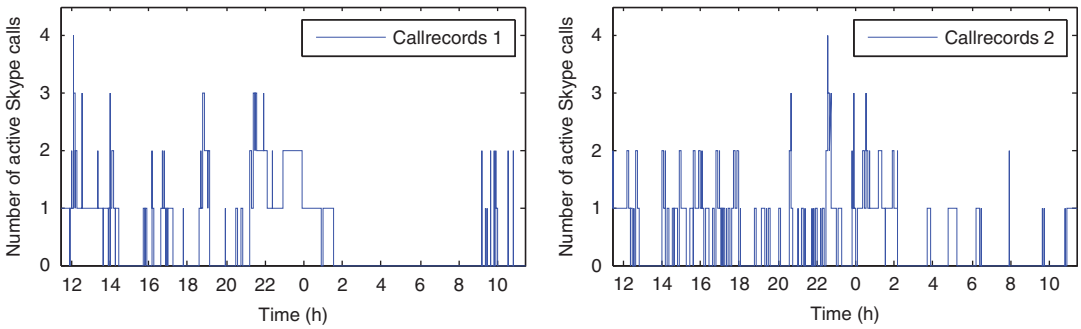

Figure 7. Daily fluctuation of the number of voice calls in the ADSL domains.

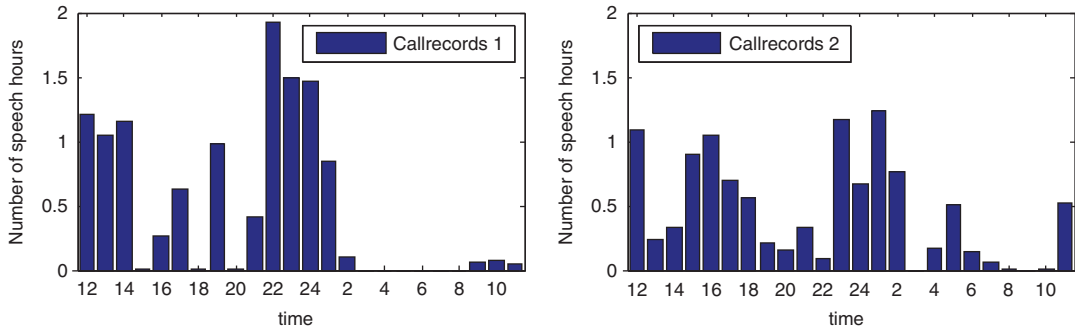

Figure 8. Daily fluctuation of the speech hours in the ADSL domains.

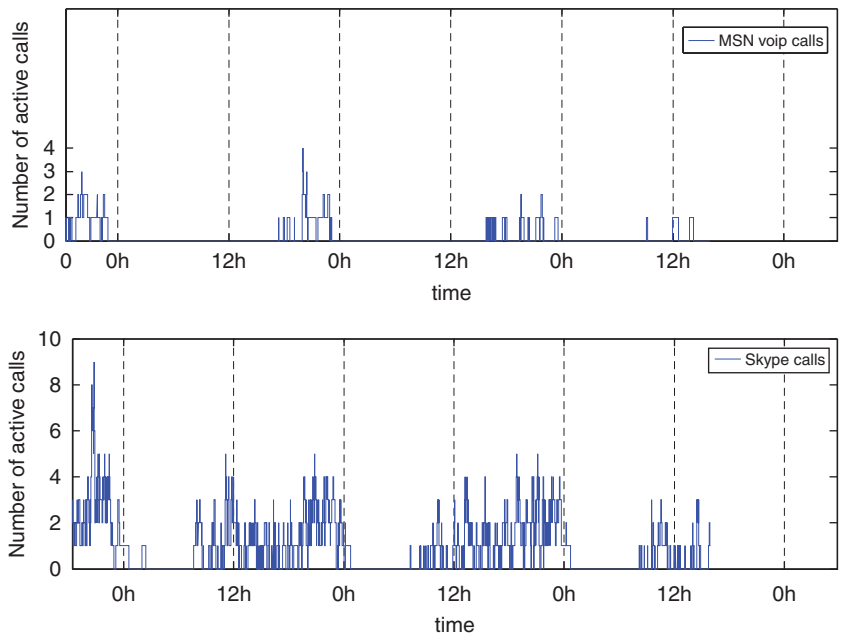

Figure 9. Daily fluctuation of active MSN and Skype calls in the mobile network.

In the almost 3 day long data set captured in the mobile network, we managed to detect 444 Skype calls, which allows us to calculate more precise histograms and other statistics. In addition, 66 MSN voice calls were also detected. First, RTP flows were identified with the method proposed by Szabó

DOI: 10.1002/dac

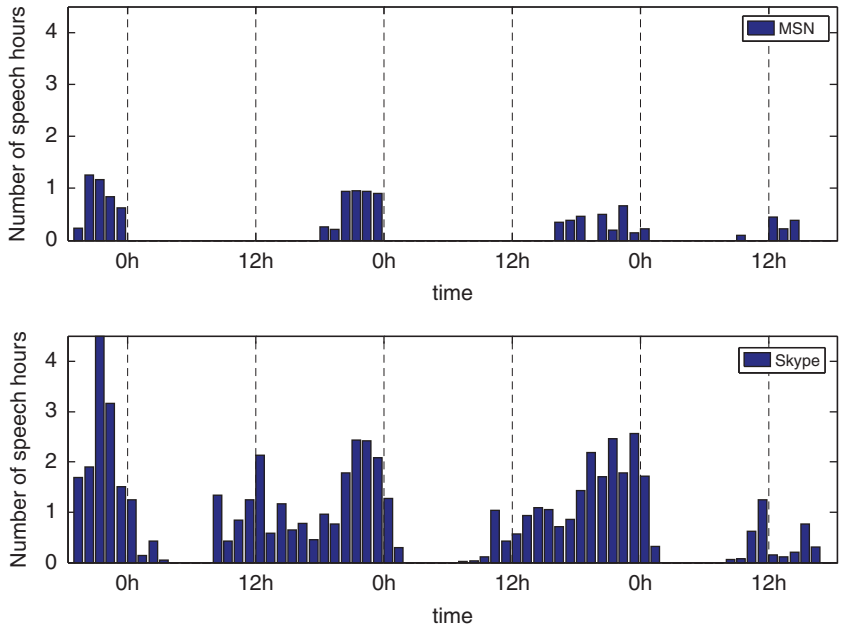

Figure 10. Daily fluctuation of the speech hours in the system during the 3-day long mobile measurement.

et al. [26], and then MSN voice calls were selected based on the content-type field. This way, we can make a comparison between the two popular voice services.

The fluctuation of active Skype calls in the mobile network is presented in Figure 9. The number of Skype calls shows similar daily profiles in all three days of the measurement period. Apart from the typical busy period in the daily hours (occurring in all telecommunication networks), we can recognize a second busy period in the evening and night hours. This suggests that many users use Skype at night for private conversations. We can also see non-busy periods between about 1 AM and 7 AM. MSN calls seem to follow some similar daily profile, although we have not detected enough MSN calls to establish general conclusions.

Figure 10 shows the number of speech hours in each 1 h long interval of the 3-day long measurement. The figure confirms that the main busy hours are at night.

The daily profiles of active Skype calls and speech hours are very similar in both the ADSL domain and the mobile network (compare Figures 7–10). According to these figures, we can clearly state that the main busy hours in case of Skype are in the evening and night hours. This suggests that most people use Skype as a free time activity to talk with friends. It can also indicate that at daytime people either do not have time to conduct private conversations, or they do not choose Skype because they can use the landline phone for free at the workplace. It seems that most Skype users (in Hungary) should be regarded as home users, not business users. Few people or companies deploy Skype for business purposes.

6.2. Basic call characteristics

The next two figures (Figures 11 and 12) show the bandwidth and the packet rate of the detected Skype calls. Figure 11 shows that the bandwidth of Skype calls is usually between 18 and 70 kbps, typically around 40 kbps. Figure 12 shows one prominent and one small peak 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/s) 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.

The average packet size of Skype speech flows is plotted in Figure 13. The figure shows that the typical packet size (including IP and TCP/UDP headers) is somewhere between 100 and

DOI: 10.1002/dac

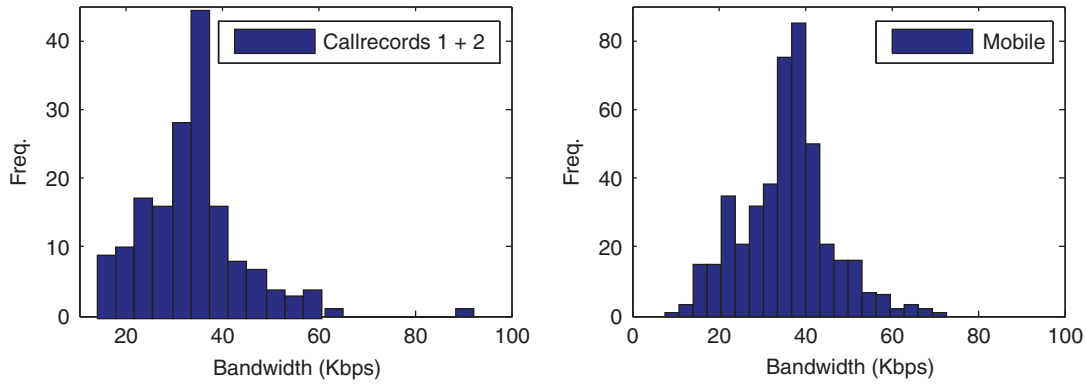

Figure 11. Histogram of the bandwidth of Skype calls in one direction in the fixed ADSL network (left) and in the mobile network (right).

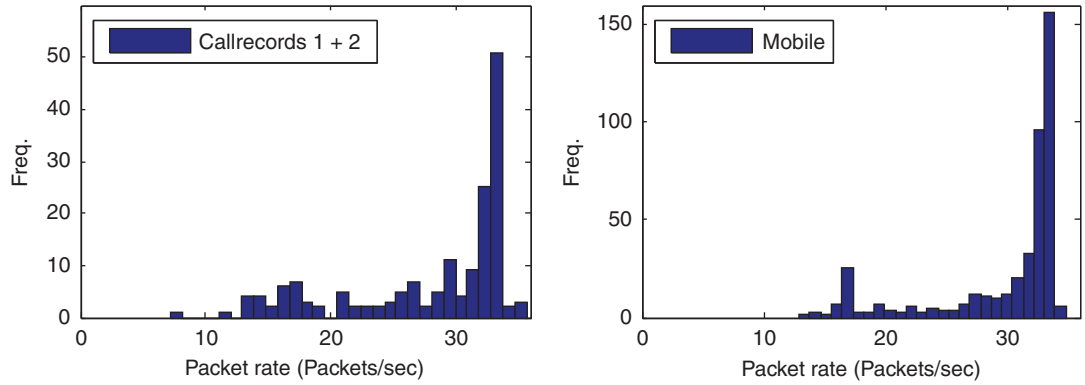

Figure 12. Histogram of the packet rate of Skype calls in one direction in the fixed ADSL network (left) and in the mobile network (right).

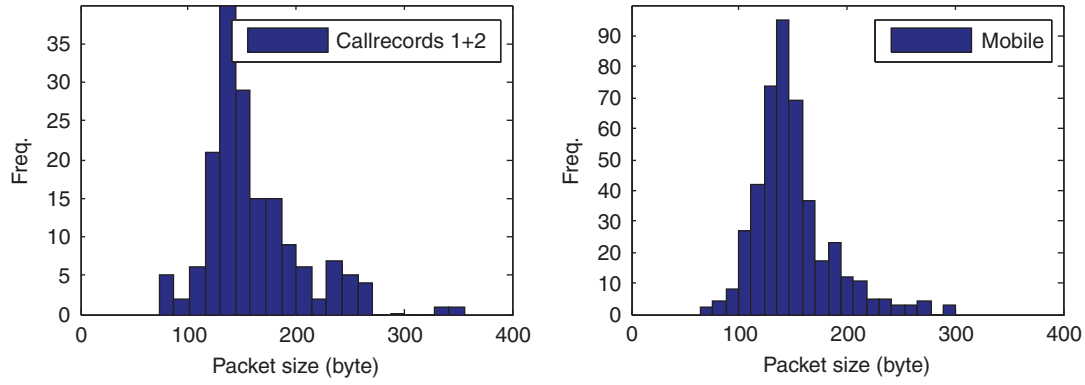

Figure 13. Histogram of the average packet size of Skype speech flows in the fixed ADSL network (left) and in the mobile network (right).

200 byte, which is also confirmed by our test measurements on local computers. Smaller packet size and bandwidth occur in one direction when separate inbound and outbound TCP flows belong to the call.

Figure 14 shows the histogram of the duration of Skype calls. It suggests an exponential-like distribution. However, due to the limited number of samples, we can regard this as a conjecture.

DOI: 10.1002/dac

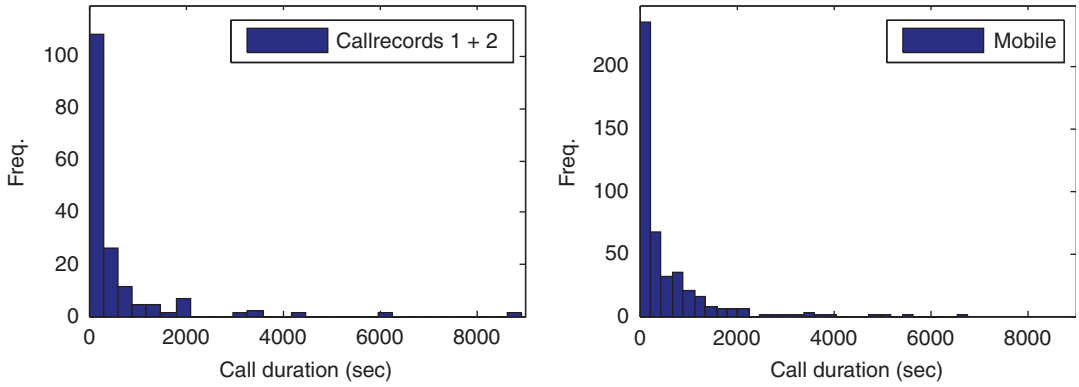

Figure 14. Histogram of the duration of Skype calls in the fixed ADSL network (left) and in the mobile network (right).

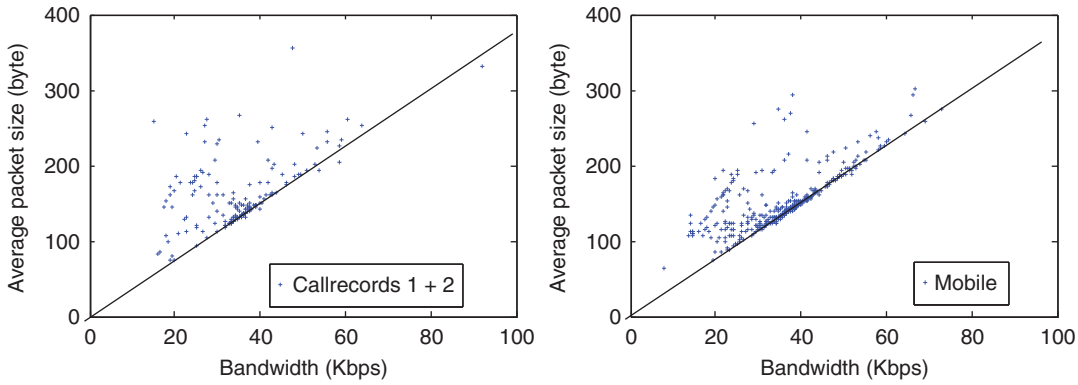

Figure 15. Average packet size as a function of bandwidth for Skype calls (in one direction) in the fixed ADSL network (left) and in the mobile network (right).

6.3. Relations between call characteristics

The following figures depict the correlation between the previous characteristic properties of Skype data flows and voice packets. Figure 15 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 most of 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 16 shows the correlation between packet rate and bandwidth of Skype calls (in outbound direction). One can see two dense areas in the figure correspond to $\frac{16}{33}$ packets/s and 20–30/35–45 kbps, respectively, as indicated in the figure. The two horizontal lines indicate the typical packet rates of Skype flows.

All properties of the detected Skype calls (histogram of bandwidths, packet rates, average packets sizes, and call durations) in the mobile network are very similar to those that we experienced in the ADSL domain. Thus, the mobile environment does not seem to affect these properties. This is not a surprising fact, knowing that the users in the mobile network have 3G or HSDPA connections, which provide higher bandwidth in both uplink and downlink directions than what is required by Skype.

DOI: 10.1002/dac

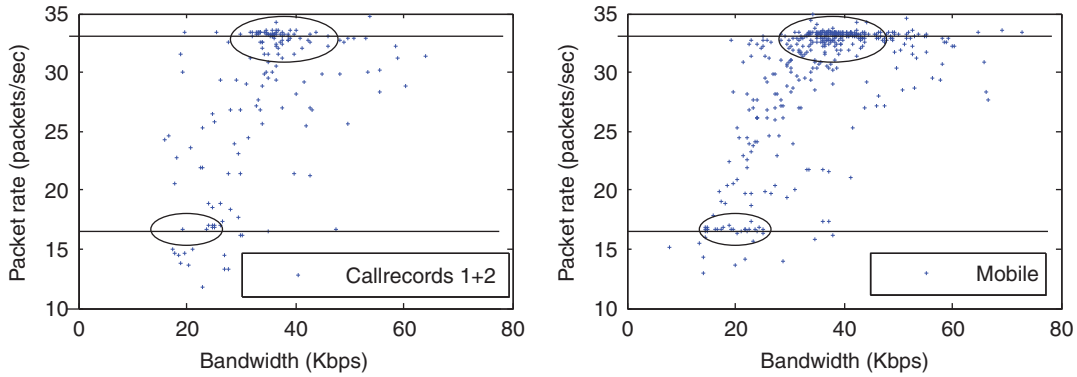

Figure 16. Packet rate as a function of bandwidth of Skype calls (in one direction) in the fixed ADSL network (left) and in the mobile network (right).

7. CONCLUSION

In this paper, we examined the operation of the Skype peer-to-peer overlay network. The numerous connections initiated by the client toward Skype SNs, dedicated servers, and other peers were investigated in details. According to these observations, we proposed heuristic methods based on flow-dynamics and flow- and packet-level characteristics to identify Skype traffic. The algorithms use packet headers only and the extracted flow-level information; no packet payload information is necessary.

The novel decision-based identification algorithm focuses on the 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 connections initiated by the 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. This method can detect logged on but idle Skype clients as well, in addition to calls.

The method expects pre-captured (offline) data as input. However, it could be built into an online identification tool.

We performed traffic analysis in both fixed and mobile network environments in Hungary, using several traces, including two 24 h real data sets measured in an ADSL domain and a 72 h trace captured in a 3 G/HSDPA mobile network. Several analysis results were presented in this paper, including the daily profile of Skype calls and user activity in the aggregate traffic, the characteristic properties of Skype voice calls, and some interesting findings about the Skype codec.

We also presented the validation of the identification algorithms based on active test measurements at our university department and in a mobile test cell.

ACKNOWLEDGEMENTS

The research was supported by NKTH-OTKA grant CNK77802 and S. Molnár was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. 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.

REFERENCES

1. Desclaux F. Skype uncovered—security study of Skype. 2005. Available from: http://www.ossir.org/windows/supports/2005/2005-11-07/EADS-CCR_Fabrice_Skype.pdf.
2. Skype Technologies SA. Skype—guide for network administrators, version 1.01, 2005. Available from: http://www.skype.com/security/guide-for-network-admins.pdf.
3. Biondi P, Desclaux F. Silver needle in the Skype. *Proceedings of Black Hat Europe*, Amsterdam, Holland, 2006.
4. Baset SA, Schulzrinne H. An analysis of the Skype peer-to-peer Internet telephony protocol. *Proceedings of IEEE INFOCOM'06*, Barcelona, Spain, 2006.

DOI: 10.1002/dac

5. Ehlert S, Petgang S. Analysis and signature of Skype VoIP session traffic. *Technical Report NGNI-SKYPE-06b*, Fraunhofer FOKUS, Berlin, Germany, 2006.
6. Ghandour W. Blocking Skype using squid and OpenBSD. *Help Net Security*, 2005. Available from: www.net-security.org.
7. Suh K, Figueiredo DR, Kurose J, Towsley D. Characterizing and detecting Skype-relayed traffic. *Proceedings of IEEE INFOCOM'06*, Barcelona, Spain, 2006.
8. Guha S, Daswani N, Jain R. An experimental study of the Skype peer-to-peer VoIP system. *Fifth International Workshop on Peer-to-Peer Systems* (*IPTPS'06*), Santa Barbara, U.S.A., 2006.
9. Chen K-T, Huang C-Y, Huang P, Lei C-L. Quantifying Skype user satisfaction. *Proceedings of ACM SIGCOMM*, Pisa, Italy, 2006.
10. Bonfiglio D *et al.* Revealing Skype traffic: when randomness plays with you. *Proceedings of ACM SIGCOMM*, Kyoto, Japan, 2007.
11. Bonfiglio D, Mellia M, Meo M, Ritacca N, Rossi D. Tracking down Skype traffic. *Proceedings of IEEE INFOCOM'08*, Phoenix, AZ, U.S.A., April 2008.
12. Rossi D, Mellia M, Meo M. Following Skype signaling footsteps. *Proceedings of Fourth International Telecommunication Networking Workshop on QoS in Multiservice IP networks* (*QoS-IP*), Venice, Italy, February 2008.
13. Yu Y, Liu D, Li J, Shen C. Traffic identification and overlay measurement of Skype. *Computational Intelligence and Security*: *International Conference*, *CIS 2006*, Guangzhou, China, 2006.
14. Freire EP, Ziviani A, Salles RM. On metrics to distinguish Skype flows from HTTP traffic. *5th Latin American Network Operations and Management Symposium*, Petrópolis, Brazil, 2007.
15. Freire EP, Ziviani A, Salles RM. Detecting Skype flows in Web traffic. *Proceedings of IEEE Network Operations and Management Symposium* (*NOMS*) *2008*, Salvador, Bahia, 2008.
16. Cheng WQ, Gong J, Ding W. Identifying BT-like P2P traffic by the discreteness of remote hosts. *Proceedings of 32nd IEEE Conference on Local Computer Networks* (*LCN 2007*), Dublin, Ireland, 2007.
17. Wang J-H, Pan J-Y, Cheng Y-C. Session recognition and bandwidth guarantee for encrypted Internet voice traffic: case study of Skype. *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining* (*CIDM 2007*), Honolulu, HI, U.S.A., 2007.
18. Blocking Skype. Available from: http://voiptelephonyservice.blogspot.com/2006/10/block-skype-hype.html.
19. Blocking Skype. Available from: http://ciscotips.wordpress.com/2006/06/07/how-to-block-skype/.
20. Procera Network PacketLogic tool. Available from: www.proceranetworks.com.
21. Blocking Skype. Available from: http://protocolinfo.org/wiki/Skype.
22. Perényi M, Molnár S. Enhanced Skype traffic identification. *VALUETOOLS*, Nantes, France, 2007.
23. Perényi M, Gefferth A, Dang TD, Molnár S. Skype traffic identification. *Proceedings of IEEE Global Communications Conference* (*GLOBECOM 2007*), Washington, DC, U.S.A., 2007.
24. Sat B, Wah BW. Analysis and evaluation of the Skype and Google-Talk VoIP systems. *Proceedings of Multimedia and Expo 2006*, Toronto, Canada, 2006.
25. Dang TD, Perényi M, Gefferth A, Molnár S. On the identification and analysis of P2P traffic aggregation. *Proceedings of Networking 2006*, Coimbra, Portugal, 2006.
26. Szabó G, Szabó I, Orincsay D. Accurate traffic classification. *Proceedings of World of Wireless*, *Mobile and Multimedia Networks* (*WoWMoM*), Helsinki, Finland, 2007.

AUTHORS' BIOGRAPHIES

Sándor Molnár received his MSc and PhD in Electrical Engineering from the Budapest University of Technology and Economics (BME), Budapest, Hungary, in 1991 and 1996, respectively. In 1995 he joined the Department of Telecommunications and Media Informatics, BME. He is now an Associate Professor and the principal investigator of the teletraffic research program of the High Speed Networks Laboratory. Dr. Molnár has participated in several European research projects COST 242, COST 257, COST 279 and recently in COST IC0703 on 'Traffic Monitoring and Analysis: theory, techniques, tools and applications for the future networks'. He was the BME project leader of the Gold Award winner 2009 CELTIC project titled 'Traffic Measurements and Models in Multi-Service networks (TRAMMS)'. He is a member of the IFIP TC6 WG 6.3 on 'Performance on Communication Systems'. He is a participant in the review process of several top journals and serves on the Editorial Board of the Springer Telecommunication Systems journal. He is active as a guest editor of several international journals such as the ACM/Kluwer Journal on Special Topics in Mobile Networks and Applications (MONET). Dr. Molnár served on numerous technical program committees of IEEE, ITC and IFIP conferences working also as Program Chair. He was the General Chair of SIMUTOOLS 2008. He is a member of the IEEE Communications Society. Dr Molnár has more than 140 publications in international journals and conferences (see http://hsnlab.tmit.bme.hu/∼molnar/

DOI: 10.1002/dac

for recent publications). His main interests include teletraffic analysis and performance evaluation of modern communication networks.

Marcell Perényi received his MSc degree in Computer Science from the Budapest University of Technology and Economics (BUTE), Hungary, in 2005. He was awarded the PhD title from the Doctoral School of Informatics of BUTE in 2009. He has participated in several research projects, supported by the EU and the Hungarian government, in the Department of Telecommunication and Media Informatics. He is a member of the IEEE and HTE. His research interests include simulation, algorithmic optimization and planning of optical networks, as well as identification and analysis of the traffic of IP networks, especially P2P, VoIP and other multimedia applications. He has experience in planning and maintaining of database systems, web services and Microsoft infrastructures. Currently he is working for Ericsson Hungary as System Engineer.

DOI: 10.1002/dac