Detection and Tracking of Skype by exploiting Cross Layer Information in a live 3G Network

Philipp Svoboda[†], Esa Hyytiä[‡], Fabio Ricciato[‡],
Markus Rupp[†], Martin Karner[*]

[†]INTHFT Department, Vienna University of Technology, Vienna, Austria
[‡] Forschungszentrum Telekommunikation Wien, Vienna, Austria
[*] mobilkom austria AG, Vienna, Austria

Abstract. This paper introduces a new method to detect and track Skype traffic and users by exploiting cross layer information available within 3G mobile cellular networks. In a 3G core network all flows can be analyzed on a per user basis. A detected Skype message is therefore related to a specific user. This information enables user profiles that provide a relationship between the mobile station and the characteristics of the corresponding Skype instance, which remain unchanged for long periods of time. Based on this information, our computationally lightweight method is able to classify Skype flows accurately. Moreover, the method is, by design, robust against false positives. Based on test traces from a live network, our new method achieves a similar detection performance as publicly available tools, yet with much less complexity.

1 Introduction

Nowadays, 2009, the traffic in the packet switched domain is increasing fast. Therefore, the operators are interested which services are present in the PS domain and as a next step how to optimize the network accordingly. In previous studies we classified traffic based on the port numbers found in the traffic flows. However, over the time the share of traffic we could identify reliably has started to decrease. In addition to this, we want to be able to discriminate background noise originated from Skype nodes probing for other nodes from port scans and attacks against the network elements [1]. Therefore, we have started to research in more advanced traffic classification for the traces from the measured 3G core network.

In this work we focus on the detection of Skype traffic in a 3G core network. The core network of a mobile operator offers various additional signaling information, which can be used to analyze traffic of each user. More specifically, the signaling information relates each IP packet with a mobile host, and therefore with a specific user or mobile station (MS). Our approach is different from other studies, and is in some sense more practical, as the major part of the work takes place when the signaling traffic is analyzed. Note that we gain here as the signaling load represents only a small fraction of the flow arrival rate. After that, the classification of an individual data flow translates to a simple query from a user profile database.

Several methods to detect Skype traffic on a network link have been proposed [2–4]. The first method [2] is similar to our approach, but for Internet backbone

links. In fact, our focus is not in traffic classification, but in the user satisfaction with the service Skype. The presence of a Skype user is detected via a call to the update server. The Skype port is then set to the most active UDP port.

The second method [3], and its advanced implementations [4], are based on statistical classification. Firstly, a voice communication has certain unique characteristics and, therefore, VoIP flows will have, e.g., a constant rate and small packet size. Secondly, Skype packets have certain structure, and the classifier can check that the first two byte and the data area have a high entropy, as they are encrypted, and that the third byte has a low entropy, as it is signaling.

Our method is combining well known facts from the other papers to gather information on Skype traffic. However, to the best of our knowledge, the particular idea to store in a central database information on the services a user accesses and the settings the client of the user, as discussed in the paper have not been proposed in the literature.

2 Measurement Setup

The reference network scenario is depicted in Fig. 1. As most access networks, the 3G mobile network has a hierarchical tree-like deployment. The mobile stations and base stations are geographically distributed. Going up in the hierarchy (see [5]) the level of concentration increases, involving a progressively smaller number of equipments and physical sites. In a typical network there are relatively few Serving GPRS Support Nodes (SGSN) and even fewer Gateway GPRS Support Nodes (GGSN). Therefore it is possible to capture the whole data traffic from home subscribers on a small number of Gn/Gi links. For further details on the structure of a 3G mobile network refer to [5].

Measurement System: We used the monitoring system described in [6]. This system supports all protocols of the packet switched domain in a 3G core network, MS tracking per packet, and user mobility. Independent modules, so called metrics, can be attached to this system working with the derived data sets. The measurement modules run online to avoid the storage of user critical payload data. To meet privacy requirements traces are anonymized by replacing all fields related to user identity at the lower 3G layers with unique identifiers which cannot be reversed, while the user payload above the TCP/IP layer is removed after the checking. Therefore, our system is able to associate packets and to reconstruct flows.

Captured Traces: In this work we captured two traces in the live network of a mobile operator at one Gn interface. The Gn interface connects a SGSN with a GGSN. The protocol at the Gn interface is the GPRS Tunneling Protocol (GTP). This protocol allows to analyze data packets on a per user base. For details of 3G architecture, see [5].

Two traces, $\texttt{TR}_1$ and $\texttt{TR}_2$, were recorded in the last week of August and September 2008, respectively. Both traces span four hours including the busy

hour in the network of the operator. This allows to extract a sufficient statistic. All numbers presented are renormalized by an undisclosed value. The length of the traces was chosen in order to allow reasonable fast processing on one hand, and to offer enough input data on the other hand.

3 Detection Method

Our detection method is based on some assumptions closely related to 3G core networks. The start of a data transfer in UMTS is similar to dial-up session. The user initiates a so called Packet Data Protocol (PDP)-context, which enables him to transfer data on the IP layer. The measurement software is tracking such PDP-context creations. Therefore, we are able to identify the start of a data session, which itself addresses a unique MS by the related mobile host.

The network under test offers dynamically allocated public IP addresses for each active PDP-context. In this work the term "local" always refers to the parameters of the 3G mobile device, e.g., the public IP address. In such a case, where no network address translation takes place, Skype is mainly communicating via UDP. We focus here on the detection in such a scenario. Functionally the setting is similar with dial-up connections when, e.g., PPP protocol is used to authenticate and assign a dynamic IP address.

Structure of Skype Packets: A typical Skype packet is depicted in Fig. 2(a) [7, 8]. The first two byte of the packet represent the ID of the packet. The ID for each packet is chosen randomly. It defines a packet in a unique way, e.g., allowing retransmission requests. The next byte indicates the type of the packet, this can be interpreted as a signaling setting. There are random bits added to this byte in order to obfuscate the detection. The real function is obtained by applying a bit mask `0x1F` to the byte. Table 1 gives the known byte values (Fall 2008). The rest of the packet is encrypted Skype payload.

Detecting Skype Flows: In older versions of Skype the first packet a client did send had some special properties. Following [8] the public IP address parameter in the ciphering is set to *0.0.0.0*. Note that according to [8] the public IP address

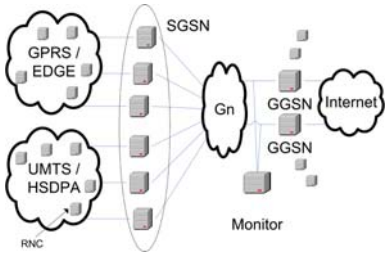
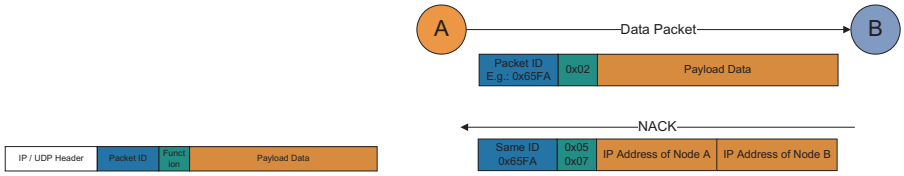


Fig. 1. Measurement Setup.

of the sender is part of the encryption function. Therefore, the receiver, which used IP address stored in the IP-header, was not able to decode the arriving packet in a proper way and triggered a NACK packet. The UPD payload of this packet contains the public IP address of the client in plain text. This message identifies network address translations by the Skype software. This procedure is depicted in Fig. 2(b).



(a) Header of a UDP Skype Packet. (b) Initial Packets of Node-A.

Fig. 2. Packet Structure of Skype Messages.

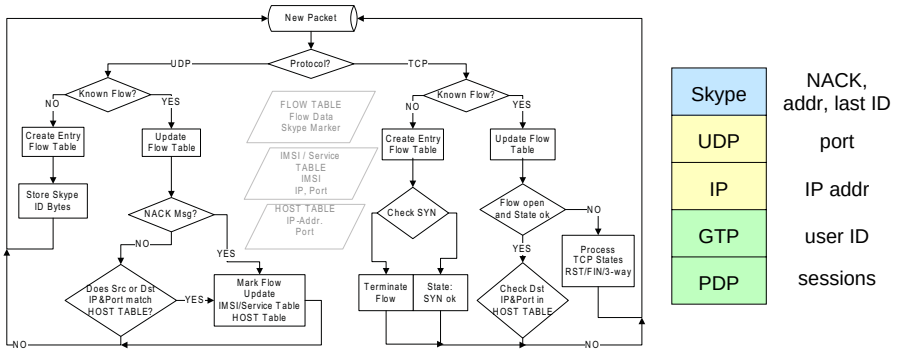
Function	Value	Description
Enc	`0x02`	Initial packet for encoding
NACK	`0x05` or `0x07`	Packet of given ID could not be decoded
Resend	`0x03`	Retransmitted packet
Data	`0x0D`	Normal data packet

Table 1. Description of the Values of the Signaling Byte.

In the current version (3.1.0) the algorithm of Skype has improved compared to older versions, see [8]. After the startup the client encrypts the packets based on its last known public IP address. Therefore, in an scenario with static IP addresses the NACK message will only occur once after the software installation. However, in our network, for each PDP-context creation, a user is assigned a new IP address out of the address pool of the operator. Moreover, in our measurement period we did not observe IP re-usage for an individual user. Earlier measurements did show similar results [9]. This behavior facilitates the detection procedure.

We start tracking at the beginning of a PDP-context. Therefore, we monitor those two login packets. The detection algorithm executes for each UDP packet the following (simplified) steps:

1. *Check if the third payload byte matches any Skype function.*
   - Yes: go to **2**
   - No: go to **4**
2. *Check local UDP port with the database.*
   The database contains the IP address and port of every detected Skype user.
   - Hit: Flow is marked as Skype traffic, go to **3**
   - Otherwise: go to **3**



(a) Flow-diagram for Skype Detection. (b) Layer Information.

Fig. 3. Principles of the Cross Layer Based Skype Detector.

3. *Check if the packet is a NACK messages.*
   This is the case if and only if the UDP payload matches the follwing:
   - The first two byte (ID) match with last seen ID to this destination.
   - The third byte (function) matches to `0x05` or `0x07`.
   - The Skype payload length is 12 byte.
   - The Skype payload contains the client IP address in plain text (4 byte).

   If the packet contains a NACK message we store the following information:
   - Local user and port as active Skype client, Skype port respectively
   - Remote address and port as active Skype client, Skype port respectively
4. *Wait for the next UDP packet.*

Based on these steps the method is able to mark Skype related flows. The method matches 7 byte of information in the actual packet of which 4 byte are cross-layer information (the IP address of the client) and two bytes from the previous packet (ID). The flow-diagram of the method is depicted in Fig. 3(a). Regarding the fact that one NACK packet reveals two Skype nodes it is not mandatory to catch all login packets of each client. The method will work fine as soon as the database is populated with so-called super-nodes.

The authentication process is achieved via a TCP connection, see [7]. We detect this via pattern matching like proposed in [7]. Further detection of the TCP flows is possible using the method presented in [7]. Based on the lookup table for active Skype nodes, we are also able to mark possible Skype related TCP flows, note these flows use the port related to "Skype" too. With this information we can already classify most of the TCP flows. Without the user related signaling information present in the mobile network it is hard to detect the shut down of a Skype node see [10]. In our case, we can rely on information from the lower layers, namely 3G signaling, and assume that the Skype software is active from the first monitored login packets until the termination of the PDP-context or until a packet with a non-matching third byte (function of Skype) arrives at the client. Figure 3(b) depicts the different layers we exploit and the information we gain at each layer.

Based on the fact that we exploit data on different layers to classify Skype flows we called the method "Cross Layer Based Skype Detection" (CLBSD).

4 Measurement Results

The results of this section are derived from three traces, one test trace and two live traces, $\mathtt{TR}_1$ and $\mathtt{TR}_2$. In the test trace two Skype nodes, both connected via the radio access network of the operator generated a VoIP call and a file transfer. In this setup the ground truth is known as both terminals did only offer the Skype service.
Our method reached a detection performance of 97.1 % with respect to volume in byte and 95.2 % in terms of packets. Analyzing the non-classified flows did show that more than 95% of the remaining flows were due to port scans and P2P "background traffic". The other flows had a destination port equal to 80 or 443. Traffic on these two ports is not classified as Skype traffic at the moment as the mis-classification rate on these ports was too high. As a solution to this problem we propose a *white list* of known servers, e.g., Google, news pages and so on, traffic of which is excluded in advance.

In the next step we analyzed $\mathtt{TR}_1$ and $\mathtt{TR}_2$ with our method and compared the results with TSTAT `v1.72b`[1] [11]. Both traces were taken on a Tuesday afternoon including the busy hour around 8 p.m., following [12]. As the traffic is recorded on a packet level we had to create flows or connections. Regarding the term "connection", in case of TCP traffic it will refer to the plain TCP connection, for UDP traffic we define a connection as the union of all packets seen with the same quadruple (source / destination addresses and ports) with a maximum inter-packet spacing of ten minutes, see [12]. Based on this definition the traces contain on average more than ten million flows per hour (note that we cannot disclose more detailed numbers).

Table 2 presents the numbers of detected flows, packets and byte for all three methods. The flows in the table are accumulated over the tracing period. The values are normalized using our proposed method as reference, to 100%. The number of flows was in the order of 10^5.

The performance of TSTAT is slightly lower compared to our solution. As we are not allowed to store payload in any way, we are not able to post analyze the differences on the packet level. However, an investigation of the flow table of classified Skype traffic showed that the TSTAT method had problems detecting some long flows. From the duration, average packet size and data-rate of the flows, often larger than 10 minutes, we hypothesize that these flows are undetected voice or video calls. We assume that the statistical method of the used version has problems to cope with the silence suppression implemented into the new versions of Skype.

In order to get a better understanding we generated an artificial test trace between two mobile terminals. We then initiated a voice call between the two

[1] `http://tstat.polito.it/download/tstat.v172beta.tgz`

nodes. The test call included longer periods of silence on both terminals. In this setup TSTAT did not detect a voice call, while our method was able to detect this traffic based on the port mapping.

Trace	$\mathtt{TR}_1$			$\mathtt{TR}_2$		
Method	Flows	Byte	Packets	Flows	Byte	Packets
CLBSD	100%	100%	100%	100%	100%	100%
TSTAT v1.72b	89%	93%	91%	93%	94%	89%

Table 2. Classification of Skype Traffic.

5 Summary and Conclusions

In this paper, we present a new method to detect Skype traffic flows and users tailored for a 3G network with dynamic IP addresses allocation.[2] Traces from a Gn interface of a 3G network allow a unique mapping between data packets and users. This feature allows to track Skype users in an efficient manner.

In contrast to existing methods, we do not make use of the statistical properties of the traffic flows, but rather focus on the cross-layer information within signaling of Skype. We exploit the fact that at the startup a pair of special packets is generated. If the node has received a new IP address since the last startup of the program, which is the case in most 3G networks, we are able to detect these messages. From this first flow we gain the knowledge about the presence of a Skype node, and the port it is listening on. Note that Skype uses persistently the same port under normal conditions. In the following all flows that originate or terminate at this node and port are accounted as Skype traffic. In addition to this, the TCP authentication message is traced via a pattern matching.

The advantage of the new method is the fact that a flow can be classified already when the first packet arrives. Therefore, this approach can be directly used for quality of service settings at a low cost. However, the proposed method relies totally on the detection of special signaling events that, on one hand, may change at some point in time, and on the other hand, need a change of the client IP address as a trigger. The former constraint is the same for all Skype detection methods.

The detection performance of this method is comparable to an publicly available tool, TSTAT, but offers a higher performance in terms of accuracy and computational burden. Note that we consider the low scores for TSTAT in the table to be a part of a change of the Skype codec, rather than a restrictive design of the detector. We only mark flows based on a match of 7 byte of which 4 byte are containing dynamic cross-layer information, e.g., the local IP address of

[2] Note, in fact this method could be used in any network which offers a dynamic allocation of public IP addresses, e.g., ADSL access networks, if the relation between customer and IP address is known, e.g., by sniffing Radius messages.

the client. This is a much stronger restriction than what is found in commercial firewalls and other publications [2–4]. We believe that the mis-classification rate of our method is close to zero.

In our further work we want to change the method accordingly for networks not offering public IP addresses.

Acknowledgments

This work was part of the DARWIN+ project at the ftw. This project is supported by the COMET (Competence Centers for Excellent Technologies) initiative of the city of Vienna and hosted at the ftw in Vienna. The views expressed in this paper are those of the authors and do not necessarily reflect the views within the partners of the project.

Bibliography

[1] F. Ricciato and P. Svoboda and E. Hasenleithner and W. Fleischer. "On the Impact of Unwanted Traffic onto a 3G Network". *Proc. of the SECPERU'06*, vol. 36(4), pp. 49–56, 2006.

[2] C. Kuan-Ta, H. Chun-Ying, H. Polly, and L. Chin-Laung. "Quantifying Skype user satisfaction". *Proc. of the SIGCOMM'06*, vol. 36(4), pp. 399–410, 2006.

[3] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli. "Revealing Skype traffic: when Randomness plays with you". *Proc. of the SIGCOMM'07*, vol. 37(4), pp. 37–48, 2007.

[4] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, and D. Rossi. "Tracking down Skype traffic". In *Proc. of Infocom'08*, pp. 5, 2008.

[5] H. Holma and A. Toskala. *" WCDMA for UMTS, Radio Access For Third Generation Mobile Communications, Third Edition"*. Wiley, 2004.

[6] F. Ricciato, P. Svoboda, J. Motz, and W. Fleischer. "Traffic monitoring and analysis in 3g networks: lessons learned from the METAWIN project". *e&i Elektrotechnik und Informationstechnik*, vol. 123(7-8), pp. 22–28, July 2006.

[7] S. A. Baset and H. G. Schulzrinne. "An analysis of the skype peer-to-peer internet telephony protocol". *Proc. of 25th IEEE ICC*, vol. 1, pp. 1–11, April 2006.

[8] P. Biondi and F. Desclaux. "Silver needle in the skype". In *Proc. of Black Hat Europe06*, vol. 1, pp. 25, 2006.

[9] F. Ricciato, F. Vacirca, and P. Svoboda. "Diagnosis of Capacity Bottlenecks via Passive Monitoring in 3G Networks: an Empirical Analysis". *Computer Networks*, vol. 57, pp. 1205–1231, March 2007.

[10] D. Rossi, S. Valenti, P. Veglia, D. Bonfiglio, M. Mellia, and M. Meo. "Pictures from the skype". In *Proc. of ACM SIGMETRICS Demo Competition*, vol. 1, pp. 7, 2008.

[11] M. Mellia, A. Carpani, and R. Lo Cigno. "Measuring IP and TCP behavior on edge nodes". In *Proc. of Globecom'02*, vol. 1, pp. 5, 2002.

[12] P. Svoboda and F. Ricciato. "Composition of GPRS and UMTS traffic: snapshots from a live network". *Proc. of the IPS MoMe 2006*, vol. 4, pp. 42–54, 2006.