# A Technique for Classification of VoIP Flows in UDP Media Streams using VoIP Signalling Traffic

Tejmani Sinam, Irengbam Tilokchan Singh,
Pradeep Lamabam, Ngasham Nandarani Devi
Department of Computer Sciences
Manipur University
Imphal, India - 795003
Email: {tejmani,tilokchan,deeplamabam,nandaraningasham}[a]gmail.com

Sukumar Nandi
Department of Computer Science and Engineering,
Indian Institute of Technology, Guwahati,
Guwahati, India - 781039
Email: sukumar[a]iitg.ernet.in

***Abstract*—VoIP applications are becoming popular these days. A lot of Internet traffic are being generated by them. Detection of VoIP traffic is becoming important because of QoS issues and security concerns. A VoIP client typically opens a number of network connection between VoIP client and VoIP client, VoIP client and VoIP server. In the case of peer to peer VoIP applications like Skype network, connections may be between client to client, client to Super Node, client to login server, Super Node to Super Node. Typically, VoIP media traffic are carried by UDP unless firewalls blocks UDP, in which case media and signalling traffic are carried by TCP. Many VoIP applications uses RTP to carry media traffic. Notable examples includes GTalk, Google+ Hangouts, Asterisk based VoIP and Apple's FaceTime. On the other hand, Skype uses a proprietary protocol based on P2P architecture. It uses encryption for end to end communications and adopts obfuscation and anti reverse engineering techniques to prevent reverse engineering of the Skype protocol. This makes the detection of Skype flows a challenging task. Although Skype encrypts all communications, still a portion of Skype payload header known as Start of Message (SoM) is left unecrypted. In this paper, we develop a method for detection of VoIP flows in UDP media streams. Our detection method relies on signalling traffic generated by VoIP applications and heuristics based on the information contained in Skype SoM and RTP/RTCP headers.**

***Keywords*—Network Traffic Classification, Skype classification, Media and signal traffic***

## I. Introduction

Nowadays, Voice over IP (VoIP) applications have become very popular on the Internet. Some of the popular VoIP applications are Skype, Gtalk, Google+ Hangouts, Apple's FaceTime and Asterisk based clients. VoIP traffic usually consists of *signalling* and *media*. Different VoIP communication approaches uses multiple protocols namely signalling and media protocols. The media protocols are used to transmit media such as audio and video over IP networks. Media protocols, RTP and RTCP (RFC3550 [1]) are more or less common to all types of VoIP with the exception of Skype. Signalling protocols are responsible for the establishment, preservation and tearing down of call sessions. They are also responsible for the negotiation of session parameters such as codecs, tones, bandwidth capabilities, etc. The main signalling protocol/protocol stack in the IP network are H.323, SIP/SDP (RFC3261 [2]) and XMPP/Jingle ( [3]–[5]).

Most of these protocols are standard and their specifications are in the public domain. Skype on the other hand uses closed and proprietary protocols. And the technology it uses has not yet been disclosed.

Skype has generated lots of interest from network operators, researchers as well as many governments around the world for its many characteristics and considers identifying Skype traffic very important. Skype usage is especially of great interest for mobile service operators as more and more users are adopting it. It is indispensable for network operators to know how many users use VoIP applications especially Skype (being the most popular) and how much they talk. This way they can decide on VoIP tariff strategies [6]. Because of Skype's extensive use of cryptography, obfuscation, and anti reverse-engineering techniques, classical statistical traffic classifiers are not suitable to correctly classify Skype traffic [7]. Skype's bandwidth consumption [8], its encryption, its abilities to traverse firewalls and NATs are major cause of concern for many. In network environments that are subject to strict communication regulations, administrators may want to prohibit Skype to reduce the risk of unauthorized communications [9].

In our earlier work [10], we are able to classify UDP flows as RTP or Skype media streams. In this paper, we further propose a method of identifying RTP by correlating with the identification of RTCP traffic. For Skype we further identify a flow as Skype-media or Skype-signal. To validate these results, *host based information* is also used (subsection IV-E).

The rest of this paper is organised as follows. Section II provides background information about RTP, RTCP and Skype. Section III reviews the works done in this field with more focus on works related with the identification of Skype. Section IV describes the heuristics and methods that are used in detecting Skype and other non-skype VoIP traffic. Section V outlines the data used and how they are collected. Section VI presents some observations and results regarding the experiment. Section VII concludes the paper with some final remarks and suggestions of possible future work.

## II. Background

### A. RTP

RTP is the protocol of choice for VoIP communications that deals with real time data such as audio or video and along with

978-1-4799-2572-8/14/$31.00 ©2014 IEEE

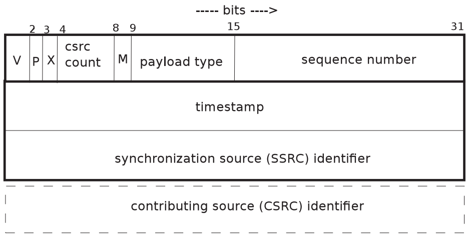

Fig. 1. RTP fixed header (*first 12 octets*)

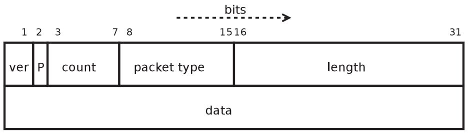

Fig. 2. RTP Control Protocol (RTCP) header

Real-time Transport Control Protocol (RTCP) it provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video conferencing, video on demand, or Push-to-Talk services. The RTP/RTCP protocols are defined in RFC 3550 [1] and RFC 3551 [11].

Applications typically run RTP on top of UDP. In SIP, a RTP session is described by SDP (Session Description Protocol, RFC 4566 [12]), which conveys media details, transport addresses, and other session description metadata such as RTP/AVP (RFC3551 [11]) to the participants. In XMPP/Jingle, RTP session is defined in XEP-0167: Jingle RTP Sessions ( [5]).

What we are interested is the fixed header of RTP packets which includes version (0-2, 2 bits), payload type (9-16, 7 bits), sequence number (16-32, 2 bytes) and ssrc (64-96, 4 bytes) fields of RTP packet header (Figure 1).

### B. RTCP

RTCP (RTP Control Protocol) is used to send periodic control packets to all participants in a RTP session. Its primary function is to provide feedback on the quality of the media connections. RTCP is generally used together with RTP to provide control information and statistics, such as quality of transmission, jitter and packet loss statistics. RTCP sends a description of the source and information about what is being sent, and also collects information about what has been sent.

About 5% of the session bandwidth is reserved for RTCP traffic, so as to allot the major share for media traffic (RTP).About 1.25% allocated to senders, and 3.75% allocated to receivers. The minimum packet transmission frequency is 5 seconds, though a delay is imposed at startup.

### C. SKYPE

Skype uses encryption for end to end communications and adopts obfuscation and anti reverse engineering techniques to prevent reverse engineering of the Skype protocol. This makes the detection of Skype flows a challenging task. Although Skype encrypts all communications, still a portion of Skype payload header known as Start of Message (SoM) is left unencrypted.

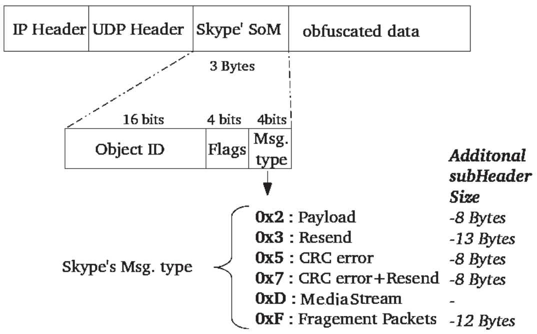

Fig. 3. Skype SoM Header

Skype application payload encapsulates a header called the Start of Message (SoM) [13]. The first three bytes of Skype UDP payload constitute the SoM. Although Skype encrypts all data transfers, SoM is not encrypted. The Skype SoM is of 3 bytes (*first 2 bytes for transaction/object/frame ID number* and *the 3rd byte for packet type, PT*) (Figure 3).

Skype uses different signalling mechanism. Skype Client (SC) exchanges packets with Skype SuperNodes (SNs) and other Skype clients. A SC application activities include startup, login, online, active, idle, call, IM, etc. The activities of a Skype application are sent to the SNs and informed to other SCs. So, some signal are for status information, some are for keepalive status to traverse NAT, some are for initial call setup, some are for call hangup etc.

More details about Skype architecture and its behavior are found in [13], [14] and [15].

## III. Related Work

There is relatively few research studies on SIP based VoIP traffic. Costeux et. al. [16] used a flow-based analysis technique to characterize and compare VoIP traffic specifically Skype and other SIP/RTP based applications. Their results gave interesting insights on traffic patterns and inter-arrivals, connectivity between users, location of sources and performance limitations.

Tstat [17] classification process exploits a finite state machine that perform checks of version field, sequence numbers and *admissible* payload types (for RTP and RTCP) [11] with a check on UDP ports ($>$ 1024) for even/odd port numbers for RTP/RTCP.

*Wireshark* [18] uses the hint derived from the media description protocol such as SIP/SDP and tries to dissect the UDP packets as RTP. *rtpdump* [19] can process RTP data based on even/odd port pairs for RTP and RTCP. *l7-filter* [20] builds RTP signature based on version, the next two bits, four bits for count of "contributing source identifier" and only admissible payload type values (0-34, 96-127) [11] leaving out fields which are randomly generated.

The works of Erdal and Hedayat [21] and Guntur [22] are patented that claims to identify UDP packets as RTP.

Check on UDP ports, i.e., even/odd for RTP/RTCP is not valid today as RFC3550 and RFC3551 has been updated in

RFC5761 [23] where it is specified that RTP and RTCP flows for a single media type can be run on a single port.

Considerable studies has been carried out around Skype. There has been extensive research on various aspects of Skype architecture, quality and reaction to congestion, call relays, firewall traversal, security and its other components such as super nodes, etc. Others have carried out work on detection and blocking of Skype. The following works mainly focuses on Skype's SoM header in order to detect and classify Skype's traffic.

Bonfiglio et. al. [13], S. A. Baset et. al. [15], Sven Ehlert et. al. [24], Bonfiglio et. al. [25] describes in depth about Skype architecture and its traffic. They differentiate between a Skype Client (SC), a Skype SuperNode (SN), Host Cache (HC), Skype Functions: *(Startup, Login, Keep-alive Messages, User Search, Call Establishment and Media Transfer)*, the structure of the Start of Message (SoM) and the different types of possible calls (e.g. with or without relay node).

Davide et. al. [26] have proposed a real-time algorithm to detect and classify Skype's signalling and data traffic. Their method applies both the signature and statistical approaches. Davide et. al. used the characteristic of [13], to describe the Skype UDP Ping and Long Skype UDP Probe to discover SNs and network characteristic. Their approach for detecting and classifying the signalling traffic over UDP consists of the functional field of SoM along with the packets size.

Host behavior based classification ( [27], [28], [29], [30], [31]) looks at the communication pattern generated by a particular host *(who talks to whom)* and to compare it to the behavior patterns representing different activities or applications. The first work in this field is presented in [31] describing the connection patterns of P2P networks, and without relying on packet payload. BLINC [29] captures the profile of a host, in terms of the destinations and ports it communicates with, identifies the applications the host is engaged in by comparing the captured profile with (built-in to BLINC) host behavior signatures of applications servers, and then classififes traffic flows.

Iliofotou, et. al [32] proposed Traffic Dispersion Graphs which model the *social behavior* of the host, where the edges can be defined to represent different interactions (e.g. the exchange of a certain number or type of packets).

Geza [33] in his Ph.D. dissertation observed that these methods has some advantages like it can work with encrypted packets since classification does not require the payload. dkdkdkd [10]

## IV. Methodology

For real-time delivery of audio and video, UDP is preferred over TCP as real-time audio and video are delay sensitive. The overhead incurred in TCP connection negotiation and establishment is huge. TCP's concept of retransmission, flow control or error recovery is not there in UDP. And moreover, TCP doesn't support multicasting. Thus, we consider only the UDP packets for the classification of VOIP traffic.

We examine each packet from a network interface or packet trace and try to characterize whether it belongs to a Skype flow or a RTP flow based on certain heuristics described later. Counters are maintained to count the number of packets in a flow that are labeled as RTP and Skype. These counters are used in a rating algorithm to classify the flow as RTP or Skype. The labeling of the flow is further crossed checked with the signalling information associated with each flow. Signalling information are gathered from associated signalling traffic, in the case of RTP, presence of RTCP signalling activities and in the case of Skype, Skype client to Supernode signalling activities associated with the active Skype Client.

### A. Data Structure

In our method, we maintain two data structure that uses hash table.

1) A *flow hash table* maintains each flow information which consists of *src ip, dst ip, src port, dst port* as *flow ID*. In addition to these information counters of total packets, total bytes size, RTP packets, RTP packets, Skype packets and their corresponding bytes counter are also maintained. These counters are later used when applying the proposed heuristics which will be described in the following sections.
2) A *host/node hash table* keeps track of host behavior of each IP-Node. This hash table records *host ID, number of connected peers, list of peer, list of application protocol detected, boolean function of each (RTP, RTCP, Skype-signal, Skype-media) to indicate their presence/absence , and list of pointers corresponding to the flow hashtables*. The host IP address repesents the host ID of the Node. The number of connected IPs with that host are represented as the number of peers. And the list of connected host IPs are represented in list of peer, (subsection IV-E).

### B. Preprocessing

Before implementing the proposed heuristics, we perform the preprocessing steps. In this step, all the multicasting and broadcast traffic are filtered out along with some well known protocols based on IANA port numbers.

### C. Identification of UDP packets

Many VOIP applications (Gtalk, Google+ HangOut, ekiga, etc,) uses RTP for media transfer over UDP, except for Skype which uses its own protocol called Skype protocol. We report each UDP packet of the corresponding flow as RTP, RTCP or Skype with our proposed heuristics. RTP and RTCP heuristic decides whether a UDP packet is RTP or RTCP based on the RFCs, whereas the Skype heuristic is based on the PT (Payload Type) of Skype's SoM (Start of Message) and the corresponding sub-header size.

*1) RTP/RTCP heuristics:* According to RFC 3550 [1] RTP fixed header are of 12 bytes which includes the *version (0-2, 2 bits), payload type (9-16, 7 bits), sequence number (16-32, 2 bytes) and ssrc (64-96, 4 bytes)*. The version is 2 as of now and in a RTP stream the SSRC should be unique,

In algorithm 1, we check whether a UDP packet of a flow can be considered as candidate RTP packet. If it is a candidate

**Algorithm 1** RTP Heuristic

```
construct the UDP stream
if UDP payload $\geq$ RTP fixed header size then
  if first 2 bits is RTP version number then
    retrieve ssrc, payload type, sequence number
    if current packet belongs to an existing RTP stream and its sequence number is incrementing then
      increment the RTP packet counter
    else
      construct the RTP stream based on ssrc, payload type and sequence number
    end if
  end if
end if
```

**Algorithm 2** RTCP Heuristic

```
construct the UDP stream
if RTCP_version is 2 && RTCP packet type then
  Flow-label $\leftarrow$ RTCP
  increment the RTCP packet counter
  if UDP payload = 4*RTCP_length then
    sub-label $\leftarrow$ Single packet RTCP
    update RTCP_info
    retrive ssrc
  else if UDP payload $>$ 4*RTCP_length then
    sub-label $\leftarrow$ compound RTCP packets
    update RTCP_info
    retrive ssrc
  end if
else
  NOT-RTCP
end if
```

RTP packet then we increment the RTP packet counter of the flow.

Similarly, RTCP identification heuristic (algorithm 2) is also based on its header information (Figure 2) as specified in RFC3550 [1]. The proposed heuristic takes into consideration the *version, packet type and ssrc* fields of RTCP header. The algorithm 2 also checks for RTP and RTCP packet types conflict as mentioned in RFC 5761 [23].

In [1] it is stated that all RTCP packets must be sent as compound packets but [34] allows the use of non-compound RTCP packets in some circumstances. Thus, a check for single or compound RTCP is included in algorithm 2.

The retrieved *ssrc* field is then used to correlate RTP and RTCP flows.

*2) Skype Heuristic:* All the Skype UDP packets starts with SoM (Start of Message) header which is 3 bytes. The last bytes contain the PT (payload type) of the Skype. This PT have their own message and the message format are different. But same PT have same message format and each message have their own sub-header. So, the sub-header size of the corresponding PT are unique.

The Skype heuristic (algorithm 3) is based on the fact that a UDP packet have the possible PT and their corresponding sub-header size (Figure 3). The payload field in Skype SoM is encoded using four bits. The valid values of the payload type are magic numbers (2, 3, 5, 7, 13 and 15). All are prime numbers except 15. Skype media traffic have SoM messages type equal to 13. Media traffic may be voice, video, file transfer between the Skype clients.

We check whether a UDP packet of a flow can be considered as candidate for Skype packets. If it is a candidate Skype packet then we increment the Skype packet counter of the flow.

According to the proposed heuristic (algorithm 3), if all the packets *(100%)* in a UDP flow have SoM message, then the flow is labeled as *SKYPE* otherwise, it is labeled as *NON-SKYPE*. And further, the *SKYPE* flows are identified as Skype-media or Skype-signal. If the SoM message type is 13, then the *SKYPE* flow is identified as Skype-media otherwise Skype-signal.

**Algorithm 3** Skype Heuristic

```
construct the UDP stream
if UDP payload $\geq$ SoM header size then
  if SoM packet type is valid Skype packet type then
    if payload size $\geq$ corresponding payload size then
      increment the Skype packet counter
      Flow-label $\leftarrow$ SKYPE
      if packet type = 13 then
        sub-label $\leftarrow$ media
      end if
    else
      Flow-label $\leftarrow$ NON-SKYPE
    end if
  end if
end if
```

### *D. Detection of flow as RTP, RTCP, Skype-media or Skype-signal*

Based on algorithm 1, 2 and 3 and [10], we identify a flow as,

1) RTP if 60% of the total packets in the flow is RTP packets
2) RTCP if 5% of the total packets in the flow is RTCP packets
3) Skype if 100% of the total packets in the flow is Skype packets and further identified as
   a) Skype-media if the flow is SKYPE and sub-label is Skype-media
   b) else, it is Skype-signal
4) OTHERS

### *E. Correlating Using Host Behavior*

The information gained from studying the behavior of hosts gives additional advantage in the identification of each flow in the flow table. Say, a flow is identified as Skype-media, then we look up the list of application protocol detected in host/node table of the source and destination IP of the current flow. And if the skype-signal is present then we confirm that flow as skype-media. And on the other hand the identification of RTP is endorsed with the RTCP information of the corresponding nodes. This is due to the fact that, in RTP media sessions, periodic RTCP packets are sent among the participants.

TABLE I. EXPERIMENTAL DATA-SETS

| **Applications** | **Pkts** | **Size** |
|---|---|---|
| Gtalk | 38855 K | 7.23 GB |
| Google+ Hangout | 20885 K | 10.09 GB |
| Asterisk based clients | 4285 K | 1.49 GB |
| Skype | 26536 K | 15.41 GB |
| Skype from Tstat's | 323941 K | 586 GB |

TABLE II. RTP IDENTIFICATION RESULTS

| **Threshold** | **RTP** | | **Skype** | | **OTHER** | |
|---|---|---|---|---|---|---|
| | Pkts (%) | Bytes (%) | Pkts (%) | Bytes (%) | Pkts (%) | Bytes (%) |
| 10 | 95.55 | 96.97 | 0.00 | 0.00 | 4.45 | 3.03 |
| 20 | 95.54 | 96.97 | 0.00 | 0.00 | 4.46 | 3.07 |
| 30 | 95.54 | 96.97 | 0.00 | 0.00 | 4.46 | 3.03 |
| 40 | 95.53 | 96.97 | 0.00 | 0.00 | 4.47 | 3.03 |
| 50 | 95.48 | 96.84 | 0.00 | 0.00 | 4.52 | 3.16 |
| 60 | 95.31 | 96.49 | 0.00 | 0.00 | 4.69 | 3.51 |
| 70 | 77.57 | 59.80 | 0.00 | 0.00 | 22.42 | 40.20 |
| 80 | 62.50 | 29.29 | 0.00 | 0.00 | 37.50 | 70.71 |
| 90 | 51.51 | 21.67 | 0.00 | 0.00 | 48.48 | 78.32 |
| 100 | 0.11 | 0.07 | 0.00 | 0.00 | 99.89 | 99.93 |

Thus, our methodology also consists of collecting the host information which is used to reinforce the identification of each flow and correlate skype-media with skype-signal or vice versa and RTP with RTCP or vice versa.

## V. DATA COLLECTION

For our experiments, we utilize network traces obtained from our Testbed Laboratory and packets captured from the edge of our University Network. We also obtained public traces from *Tstat* [35] [36]. Our Testbed Laboratory is setup at DIT Security Lab, M.U. (*Manipur University*). We collected various types of Skype traces such as voice, video, silence call, call within LAN and WAN, etc. Using *gt's* [37] method we collected ground truth application traces. We used a Napatech data capture card, *NT4E-STD* [38] to capture traces at the edge of our University Network. Network traces were also collected at the Asterisk Server which is deployed in the public domain.

Data was collected using the VoIP clients such as Skype (Beta) version 2.2.0.35, linphone 3.5.2 (Windows 7), linphone 3.3.2 (linux mint 13), sipdroid 2.7 beta, Ekiga Softphone 3.3.2, 3CXPhone 6.0.26523.0 (Windows 7), Gtalk in Google Chrome v20.0.1132.57, Gtalk in Google Chrome v23.0.1271.91, Gtalk with Empathy 3.4.2.3, Google+ Hangout and Asterisk 11.0.0-beta1 using Android 4.0.3 (ICS), Android 4.1 (Jelly Bean), Windows 7, linux-mint 13 and Ubuntu 12.04.

For the experiment we were able to collect $\approx$ 34 GB (Table I) of VoIP traces including Skype's 15.41 GB, spread over 5-6 months. And another 3.8 GB of Skype's *anonymized* traffic was obtained from Tstat. We only downloaded end to end Skype UDP traffic from Tstat. From these anonymized traces we extracted about $\approx$ 586 GB of packet size as derived from the header IP lengths.

## VI. RESULTS

### A. Results

From table I, total number of VoIP traces other than Skype is 18.81 GB. If we take RTP detection threshold as 10% then 18.24 GB of VoIP trace is predicted as RTP. Higher the value of threshold, the decrease in RTP prediction. This is due to the fact that media and signal data are transmitted in a single UDP stream. For the low count RTP flows, special care should be taken. It is either false positive (FP) or false negative (FN). The FP and FN of RTP are minimized using host behavior with the correlation of signal traffic.

| | Skype | RTP | OTHERS |
|---|---|---|---|
| Skype | 97.95% | 0 | 2.05% |
| RTP | 0 | 95.31% | 4.69% |
| OTHERS | 0 | 0 | 0 |

TABLE III. CONFUSION MATRIX (PACKETS)

| | Skype | RTP | OTHER |
|---|---|---|---|
| Skype | 99.85% | 0 | 0.15% |
| RTP | 0 | 96.49% | 3.51% |
| OTHER | 0 | 0 | 0 |

TABLE IV. CONFUSION MATRIX (BYTES)

Table II list all the possible values of the threshold and clearly shows that the identification of RTP deteriorates as threshold increases.

If we take the RTP detection threshold as 80, some of the RTP are identified as OTHERS as the threshold value is too high (Table II). Table III and IV shows the confusion matrix for identification of Skype and RTP with threshold = 60% (IV-D).

Total Skype traces consist of 350477 K packets with size 601.41 GB. The detection procedure (IV-D) classified 97.95% of packets and 99.85% of byte size as Skype and the remaining as 'OTHERS' but none was identified as 'RTP'. When we manually looked at the flows that are identified as 'OTHERS', we found that these flows consisted of multicast traffic, local discovery protocols, etc., but they were not Skype.

## VII. CONCLUSION

In this paper, we handled the identification of VoIP traffic in UDP Flows. The heuristics to identify RTP or RTCP in UDP packets is based on their documentations, mainly the RFCs. These heuristics uses information contained in the fixed packet headers. For Skype-media and Skype-signal the heuristic used is based on the Start of Message (SoM). The algorithms use information in packet headers only; no packet payload information is used. The results obtained are further validated using host behavior information.

We performed traffic analysis using several traces. Several analysis results were presented in this paper. We also presented the validation of the identification algorithms based on active test measurements at our university department. The method uses offline and online data as input. Effort is being given to make our method to work with real-time network traffic.

## ACKNOWLEDGMENT

This work was carried out under the project "Network Traffic Classification and Analysis", funded by Department of Electronics and Information Technology (DeitY), Ministry of Communication and Information Technology, Government of India.

## REFERENCES

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC 3550: RTP: A Transport Protocol for Real-Time Applications," IETF, Tech. Rep., 2003, http://tools.ietf.org/html/rfc3550. [Online]. Available: www.ietf.org/rfc/rfc3550.txt

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP: Session Initiation Protocol," IETF, Tech. Rep., 2002. [Online]. Available: www.ietf.org/rfc/rfc3261.txt

[3] "Fun with xmpp and google talk," http://www.adarshr.com/papers/xmpp.

[4] "Jingle," http://xmpp.org/extensions/xep-0166.html.

[5] "Jingle rtp sessions," http://xmpp.org/extensions/xep-0167.html.

[6] S. Molnár and M. Perényi, "On the identification and analysis of skype traffic," *Int. J. Commun. Syst.*, vol. 24, no. 1, pp. 94–117, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1002/dac.1142

[7] D. Adami, C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Skype-hunter: A real-time system for the detection and classification of skype traffic," *Int. J. Communication Systems*, vol. 25, no. 3, pp. 386–403, 2012.

[8] "Skype: A Practical Security Analysis," SANS, Tech. Rep. [Online]. Available: http://www.sans.org/reading_room/whitepapers/voip/32918.php

[9] H. J. Kang, Z.-L. Zhang, S. Ranjan, and A. Nucci, "Sip-based voip traffic behavior profiling and its applications," in *Proceedings of the 3rd annual ACM workshop on Mining network data*, ser. MineNet '07. New York, NY, USA: ACM, 2007, pp. 39–44. [Online]. Available: http://doi.acm.org/10.1145/1269880.1269891

[10] T. Sinam, I. T. Singh, P. Lamabam, and N. Ngasham, "An efficient technique for detecting skype flows in udp media streams," in *Proc. IEEE International Conference on Advanced Networks and Telecommunication Systems (IEEE ANTS)*, Chennai, India, Dec 2013.

[11] H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 3551 (Standard), 2003, http://www.ietf.org/rfc/rfc3551.txt.

[12] M. Handley, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," RFC 4566 (Proposed Standard), Internet Engineering Task Force, Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4566.txt

[13] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: when randomness plays with you," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, Aug. 2007. [Online]. Available: http://doi.acm.org/10.1145/1282427.1282386

[14] H. Wang, "Skype voip service-architecture and comparison," in *INFOTECH Seminar Advanced Communication Services (ASC)*, 2005, p. 4.

[15] S. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *INFOCOM*, 2006.

[16] J.-L. Costeux, F. Guyard, and A.-M. Bustos, "Detection and comparison of rtp and skype traffic and performance." in *GLOBECOM*. IEEE, 2006. [Online]. Available: http://dblp.uni-trier.de/db/conf/globecom/globecom2006qrp.html#CosteuxGB06

[17] S. Arena, R. Birke, M. Mellia, M. Petracca, C. Racca, and D. Rossi, "Multimedia and streaming traffic analysis," 2006, 3rd EuroNGI Workshop on New Trends in Modelling, Quantitative Methods and Measurements, Torino, Italy 22-23 June 2006.

[18] "Wireshark," http://www.wireshark.org/.

[19] "rtptools," http://www.cs.columbia.edu/IRT/software/rtptools/.

[20] "l7-filter," http://l7-filter.sourceforge.net/protocols.

[21] M. Erdal and K. Hedayat, "Real-time transport protocol stream detection system and method," Patent US 8 306 063, 11 06, 2012. [Online]. Available: http://www.patentlens.net/patentlens/patent/US_8306063/en/

[22] R. Guntur, "Technique for identifying rtp based traffic in core routing switches," Patent Application US 2009/0 135 834 A1, 05 28, 2009. [Online]. Available: http://www.patentlens.net/patentlens/patent/US_2009_0135834_A1/en/

[23] C. Perkins and M. Westerlund, "Multiplexing rtp data and control packets on a single port", rfc 5761," 2010, http://tools.ietf.org/html/rfc5761.

[24] S. Ehlert and S. Petgang, "Analysis and signature of skype voip session traffic," Fraunhofer FOKUS Technical report NGNI-SKYPE-06b, Tech. Rep., 2006.

[25] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, and D. Rossi, "Tracking down skype traffic," in *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA*. IEEE, 2008, pp. 261–265.

[26] D. Adami, C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "A real-time algorithm for skype traffic detection and classification," in *NEW2AN*, 2009, pp. 168–179.

[27] G. Dewaele, Y. Himura, P. Borgnat, K. Fukuda, P. Abry, O. Michel, R. Fontugne, K. Cho, and H. Esaki, "Unsupervised host behavior classification from connection patterns," *International Journal of Network Management*, vol. 20, pp. 317–337, 2010.

[28] H. Kim, k. claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices," in *ACM SIGCOMM CoNEXT*. New York, NY: ACM SIGCOMM CoNEXT, Dec 2008.

[29] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *ACM SIGCOMM Conference*, 2005, pp. 229–240.

[30] Y. Himura, K. Fukuda, K. Cho, P. Borgnat, P. Abry, and H. Esaki, "Synoptic graphlet: Bridging the gap between supervised and unsupervised profiling of host-level network traffic," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1284–1297, 2013.

[31] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of p2p traffic," in *Internet Measurement Workshop*, 2004, pp. 121–134.

[32] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese, "Network monitoring using traffic dispersion graphs (tdgs)," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 315–320. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298349

[33] G. Szabo, "Squeezing the most out of traffic classification," 2010.

[34] I. Johansson and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences," RFC 5506 (Proposed Standard), Internet Engineering Task Force, April 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5506.txt

[35] "Tstat - skype traces," http://tstat.tlc.polito.it/traces-skype.shtml.

[36] "Tstat - tcp statistic and analysis tool," http://tstat.tlc.polito.it/index.shtml.

[37] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and kc Claffy, "Gt: picking up the truth from the ground for internet traffic," *Computer Communication Review*, vol. 39, no. 5, pp. 12–18, 2009.

[38] "Napatech," http://www.napatech.com/.