2010 Fifth International Conference on Internet and Web Applications and Services

Identification and Analysis of Skype Peer-to-Peer Traffic

Dongyan Zhang
Dept. of Computer Science & Technology
Tsinghua University
Beijing, China
e-mail: zhangdongyan@mail.tsinghua.edu.cn

Chao Zheng
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China
e-mail: zhengchao@software.ict.ac.cn

Hongli Zhang
Dept. of Computer Science & Technology
Harbin Institute of Technology
Harbin, China
e-mail: zhl@pact518.hit.edu.cn

Hongliang Yu
Dept. of Computer Science & Technology
Tsinghua University
Beijing, China
e-mail: hlyu@tsinghua.edu.cn

***Abstract*—More and more applications are adopting peer-to-peer (P2P) technology. Skype is a P2P based, popular VoIP software. The software works almost seamlessly across Network Address Translations (NATs) and firewalls and has better voice quality than most IM applications. The communication protocol and source code of Skype are undisclosed. It uses high strength encryption and random port number selection, which render the traditional flow identification solutions invalid. In this paper, we first obtain the Skype clients and super nodes by analyzing the process of login and calling in different network environments. Then we propose a method to identify Skype traffic based on Skype nodes and flow features. Our proposed method makes the previously hard-to-detect Skype traffic, especially voice service traffic, much easier to identify. We design an identification system utilizing the proposed method and implement the system in a LAN network. We also successfully identified Skype traffic in one of the largest Internet Providers over a period of 93 hours, during which over 30TB data were transmitted. Through experiments, we show that our proposed approach and implementations can indeed identify Skype traffic with higher accuracy and effectiveness.**

Keywords-Skype; traffic identification; logging; voice flow

I. INTRODUCTION

Peer-to-Peer (P2P) technology is adopted in more and more applications, for example, file sharing, multimedia sharing, anonymous communication, telephony and games. As a result, P2P network traffic has increased significantly. The ability to measure, understand and model network traffic has been crucial to the execution of a wide range of network operations such as performance evaluation, traffic engineering, capacity planning, since Web became an Internet application [1,2,3]. Consequently, research activities related to P2P traffic identification are also very important.

Supported by China Postdoctoral Science Foundation.
Supported by Major State Basic Research Development Program of China(973 Program) (No. 2007CB311100).

Early applications of P2P utilized fixed ports. For example, eDonkey2000 used 4371 and 4662 ports; BitTorrent used ports from 6881 to 6889. P2P traffic could be identified based on fixed ports. With advancements in technology, P2P applications started to use random port number selection. Then P2P traffic flow can not be identified by the method of using fixed ports which render the traditional flow identification solution invalid.

Skype is the most popular and successful P2P VoIP application. The communication protocol and source code of Skype are not open. The software uses high strength encryption and random port number selection. Baset [4], Ehlert [5], Guha [6], Marcell Perényi [7] and so on [8, 9, 10, 11] analyzed Skype traffic. They all used certain payload features or flow features to identify Skype traffic. But payload features are distributed in different phases of Skype client's communications, so single payload feature cannot accurately identify Skype traffic. Using flow features helps to avoid the problem of data encryption, but doing so results in ambiguity. High accuracy is not to be expected if Skype traffic is identified using flow features.

In this paper, we focus on the detection of Skype traffic especially voice service traffic. Using both node information and flow features, we propose a method to identify Skype traffic effectively and accurately. We design and implement an identification system using the proposed method. We also apply the system in a LAN network and one of largest ISP network.

The main steps of our method are summarized below.

- Firstly, we analyze the process of login in different network environments using a black-box approach. After observation and experimentation on Skype software, we distill its payload features of different phrases in the login. Then we propose an algorithm to recognize Skype client and built a database to collect the Skype clients using payload features and time order.
- Secondly, during Skype voice calls, utilizing Skype super nodes helps to locate and identify Skype traffic

978-0-7695-4022-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIW.2010.36

quickly. So, we use active and passive methods to establish a Skype super node discovery system for obtaining the super nodes.

- Thirdly, based on the analysis of Skype login, distribution lookup and voice communication, we propose a method to identify Skype traffic using the Skype nodes and flow features which enable us to identify Skype voice traffic easily. We are able to improve the accuracy and performance of Skype traffic identification without adding complexity.

The rest of this paper unfolds as follows. A brief review of Skype is provided in Section 2. Section 3 describes the detailed algorithm to recognize Skype clients based on analysis of Skype login process. We establish and implement a Skype super node discovery system using active and passive method in Section 4. Section 5 describes the method of Skype traffic identification using Skype clients, super nodes and flow features, and we apply the system to identify Skype traffic in a LAN network and backbone network in section 6. Section 7 concludes the paper.

II. SKYPE OVERVIEW

Garfinkel [12] concluded that Skype is related to KaZaA which is a famous P2P filesharing system and it consists of the ordinary nodes (clients), super nodes (SNs) and servers. The Skype P2P network organizes participants into two layers: super nodes, and ordinary nodes. Such networks have been the subject of recent research [13,14,15,16]. Typically, super nodes maintain an overlay network among themselves, while ordinary nodes pick one (or a small number of) super nodes to associate with; super nodes also function as ordinary nodes and are actually elected from ordinary nodes. Ordinary nodes issue queries through the super node(s) with which they are associated [6].

Our own findings and recent research suggest that Skype communication consists of three components: Skype client login, buddy lookup and file/voice/video communication.

First, Skype client will login on to Skype and the login process could be divided into four steps: scanning super nodes, connecting with super nodes, connecting to update servers and login on servers.

Then, Skype clients need to lookup super nodes to obtain the buddy's IP address before they conduct file transfers, chat services, voice and video communications. The lookup can be classified into distribution lookup and concentration lookup. With distribution lookup, a Skype client sends requests to three super nodes which are known alive. The super node which responds to the request will return the buddy's IP address or return the super nodes which might know the buddy's IP address. Before obtaining the IP address of interest, the lookup can repeat at most 6 rounds. The maximum of number of super nodes included in the search process is 18. If the Skype client doesn't obtain the buddy's IP address using distribution lookup, it will use concentration lookup which asks the servers to find the IP address.

Upon learning the buddy's IP address, the Skype client can then communicate with the buddy. We focus on voice communications between two Skype clients transmitted through PCs. The caller tries to establish a TCP connection after it obtains the recipient's address. If the TCP connection is successfully established, the Skype client establishes a direct connection and exchanges the secret key using UDP protocol. In the event that a direct connection fails, Skype uses reverse connection to set up communications. Suppose a caller is behind a NAT. The caller sends a request for connection to a recipient through a servant super node of the caller. The servant super node of the caller finds a servant super node of the recipient. The recipient then sets up a connection with the caller. If such NAT reversal fails or a firewall blocks Skype packets, connections between two Skype clients are relayed by a publicly reachable super node.

III. SKYPE CLIENT IDENTIFICATION

There is no unique, fixed port for Skype traffic. The protocol and source code of Skype are not published and the data is encrypted. Furthermore, the Skype binary uses a variety of techniques to prevent reverse engineering [17]. Under such circumstance, we have to adopt a black-box approach. We simulate the Skype client actions and construct examples to analyze the payload features in different phases during login process. Using payload information, we propose a Skype client identification algorithm that recognizes unknown clients. Meanwhile, we keep records of Skype clients in a database.

A. Analysis of Skype Login Phase

Skype login process can be divided into four steps as described in section 2. After observations from and experiments on Skype, we parse the login process and distill its payload features which distribute in four steps.

Scanning super nodes: Skype client always sends UDP datagram with a size of between 25 bytes and 39 bytes to super nodes which the Skype client cache in local host. We observe that the value of position 12 is 0x02 in the sending packets and it does not change with connections, packet times and versions.

Connecting super nodes: Receiving the first response of super node, a Skype client tries to establish a TCP connection with the super node through a randomly selected port. The connection will last until Skype quits. The super node involved is referred to as "the servant super node" of the Skype client. Every 30 seconds or 70 seconds, there are periodic keep-alive messages from the client to the servant super node. Meanwhile, the client downloads a set of super nodes from the super node to which it connects and refreshes the local super nodes list.

Connecting to ui.skype.com: After establishing the connection, a Skype client sometimes sends an HTTP request to 80 port of ui.skype.com web server in order to get the latest version of Skype. The IP addresses of ui.skype.com web server are as follows:

193.88.6.228

212.187.172.228 ns5.skype.com

212.72.49.136 ns2.skype.com

217.159.236.228 ns1.skype.com

Login on servers: The login servers store the account information of users. A Skype client makes user authentication and obtains the buddy list and so on in this step. There are 0x16 03 01 00 00 contents in TCP packets. There are also 0x17 03 01 contents in the head of TCP packets and the two bytes behind the 0x170301 indicate the position of next 0x170301 in the connections to login servers. But the connection to the login server is, in some cases, relayed through a super node and therefore invisible.

B. A Skype Client Identification Algorithm

We found different payload features within different steps during the login process. So, we propose a Skype client identification algorithm: for each client, we capture the incoming and outgoing packets of the client and identify whether it is a Skype client by using the payload features and the specific time order.

TABLE I. SKYPE LOGIN PROCESS AND PAYLOAD FEATURES

Time	Name	Step	Payload Feature
T1	A	scanning super nodes	UDP packets including 0x02 value
T2	B	connecting super nodes	super nodes database
T3	C	connecting to ui.skype.com	IP addresses
T4	D	login on servers	packets including 0x170301 in the head of TCP

If we find the payload feature of 0x02 in the fixed position in the UDP datagram sent from a node, we believe the node might scan the super nodes and investigate whether the node has a connection with any super nodes which we know from the super node database. We will describe the super nod database in detail in the next section. If there are message exchanges between the node and a super node, we examine whether the node builds connections with ui.skype.com using the method of matching the IP addresses of ui.skype.com web server. Finally, we examine whether the incoming and outgoing packets from the node include 0x170301 in the head of TCP. The connections in the last two steps are not necessarily visible during Skype login communications. So, based on the payload features and time order, we can determine whether the node is a Skype client.

Description of Skype Client Identification Algorithm:

The incoming and outgoing packets from address X form packet set P.

$P=\{p_i\}$, $1 \leqslant i \leqslant N$

T_{pi} is the time when the packet p_i is sending or receiving.

SC denotes Skype client set.

Doubt SC denotes doubt Skype client set.

FA(p): determine whether p belong to A
FB(p): determine whether p belong to B
FC(p): determine whether p belong to C
FD(p): determine whether p belong to D

for (p_i, i from 1 to N)

{

if $FA(p_i)$ = false continue;

for all p_j, $p_j \in P$ && $0 \leqslant (T_{pj} - T_{pi}) \leqslant 70s$

if $FB(p_j)$ = true

{

if $p_{k1}, p_{k1} \in P$, $FC(p_{k1})$ && $0 \leqslant (T_{pk1} - T_{pj}) \leqslant 70s$
then $X \in SC$

if $p_{k2}, p_{k2} \in P$, $FD(p_{k2})$ && $0 \leqslant (T_{pk2} - T_{pj}) \leqslant 70s$

then $X \in SC$

else $X \in$ doubt SC

}

}

IV. SKYPE SUPERNODE DISCOVERY

During the Skype client login process, Skype client send UDP datagrams to a list of super nodes which cache in the local host to obtain the available super node. The client, then, downloads a set of super nodes and refreshes the local super node list from the super node that first responds to the client. We can obtain super nodes during the phase.

We use active and passive methods to obtain super nodes. The active approach is running Skype client interval automatically and records the super nodes during login process [4].

Because the number of super nodes is massive, the speed of super node discovery in single Skype client is slow, and can't show the changes in P2P network. So, we use passive method to detect the super nodes at the same time. We capture the traffic to and from Skype clients which we have identified on all or just parts of the network to extract the super nodes. The number of super nodes increases quickly using passive method.

We develop active super nodes discovery program using Autoit3 script and simulate keystroke, mouse trails and window/control operations to run Skype client automatically. The auto-login program runs Skype client for 5 minutes and then quits from Skype client. The interval of next login is 30 seconds. We have performed the experiment for 36 hours. When using the passive method to discover the super nodes, we carried out an experiment to capture traffic at one of the largest Internet providers in Beijing. The experiment lasted 93 hours. We found 37638 super nodes and observed that a

subset of super nodes was inquired several times. For example, 356 super nodes were inquired more than 10 times.

V. SKYPE TRAFFIC IDENTIFICATION

Skype traffic includes signaling and data/voice/video flows. Utilizing our analysis on the login process, we can identify signaling flows such as update information, heart-beat connections easily using the database sets of Skype client and super node. Here, we mainly discuss the method of voice flow identification. As discussed in section 2, during voice communications between two users, a Skype client needs to exchange messages with super nodes and send periodic keep-alive messages to the servant super node whether in direct connection or in reversal connection. In relay connection, two users will communicate with each other using relayed super node. Based on this understanding, we propose the method of Skype traffic identification by Skype client information, super node information and voice flow features.

A. Skype Voice Flow Features

Voice communications always have a fixed coding mode and are usually continuous. In order to decrease the delay, voice data doesn't cache long time in local storage and are transmitted quickly. Doing so has two implications: (1) the datagram cannot be too large; (2) the speed of packets transmission is stable.

Through experiments, we collected voice packets from real Skype voice connections in different network environments in order to obtain the voice flow features of Skype. The flow features that describe the Skype voice connections have three parameters: packet per second (pps), average packet size (aps) and bytes per second (bps). We use a five-variable array to describe a Skype connection which includes source IP, destination IP, source port, destination port and transport layer protocol. We use wireshark to collect voice packets and estimate the three parameters. The results are as follows:

TABLE II. Flow Parameters on Skype Voice Call

Number	Outbound packet per second(pps)	Average packet size(aps)	Outbound bytes per second(bps)	Description
1	16	167	2691	users in the same city
2	26.6	105.7	2810	users in the same city
3	44	83	3739	users in LAN network
4	16	171	2827	users in the same city through the relay
5	16.72	135.5	2266	users in the same city
6	9	79.9	718	outbound silent call
7	36.3	232	8423.9	users in the same city
8	48	107	5222	users in different provinces
9	12.25	158.7	1944	users in different provinces
10	9.14	237	2167	users in different provinces

From the table, we observed that the size of voice packet varies from 80 B to as high as 320 B except No. 6, while a speech flow in one direction has a bandwidth of 2Kbit/sec to 16 Kbit/sec and average packets range from 12 to 50 in one second. These are the Skype voice call features from our measurement.

We compare the flow properties of Skype voice call with the existing literature. Marcell Perényi [7] also found that the upper bound for packet size is 320B. We believe this feature is related to the Skype protocol. Marcell Perényi [7], however, had different findings on the other two parameters: packet and bytes per second. We believe the discrepancies are caused by the difference in locations where measurements were taken and conditions of network. The discrepancies indicate that voice flow properties are variable and should be measured in local network environment.

B. Skype Traffic Identification Algorithm

We devise a method to identify Skype traffic based on Skype client information, super node information and voice flow features. For packets we obtain from network, we first check whether the source address or destination address is a Skype client's address. If both are Skype client's addresses and flow-level properties between two users match the Skype voice call features, then the traffic is determined as Skype voice flow. If only one is Skype user, we examine the flow properties and detect whether it communicates with super nodes. Then we can identify the Skype voice flow from different types of traffic.

Description of Skype Traffic Identification Algorithm:

The incoming and outgoing packets of addresses between X_1 and X_2 form packet set P in time (T_1, T_2).

SC denotes Skype client set.

SN denotes Skype super node set.

```
if ((X1 ∈ SC) ∧ (X2 ∈ SC))
{
    if (P matches Skype Voice Flow Features)
    then P is Skype Voice Traffic
}
else if (((X1 ∈ SC) ∧ (X2 ∈ SN)) ∨ ((X2 ∈ SC) ∧ (X1 ∈ SN)))
{
    if (P matches Skype Voice Flow Features)
            then P is Skype Voice Traffic
}
else if (X1 ∈ SC)
{
    if (P matches Skype Voice Flow Features)
    {
        if ( Xi ∈ SN, X1 communication with Xi)
        then P is Skype Voice Traffic
    }
}
else if (X2 ∈ SC)
```

```
{
    if (P matches Skype Voice Flow Features)
    {
        if ( Xi ∈ SN, X2 communication with Xi)
        then P is Skype Voice Traffic
    }
}
```

VI. SKYPE TRAFFIC IDENTIFICATION SYSTEM

A. System Design

The major challenge in constructing a Skype traffic identification system in real time is the high demand for memory space and process speed. Analysis of flow properties is more complex than that of payload features and every packet needs to be analyzed. We include optimizations in our design and implementation of the system.

The system consists of four layers in order to parallel process and increase the models' modularity. The four layers are as follows:

- Flow Capture Layer: To capture and collect packets, IP packets are reassembled into IP datagram and sent to flow extract layer.
- Flow Extract Layer: For the datagram whose source or destination is a Skype client's address, this layer establishes an IP unit for the address. Flow features, such as source address, destination address, source port, destination port, protocol, length, time, etc. are extracted.
- Traffic Identification Layer: For an IP unit, the features are calculated for bandwidth, packet size, packet rate, etc. in every interval and recorded feature in hash table. The algorithm identifies Skype traffic based on flow-level properties. Additional optimization is used to maintain the hash table, delete redundancy data, decrease memory cost and update local database.
- Result Express Layer: We use single program and tools to show the result.

In the system, IP list is the core data structure and consists of millions of IP units. IP unit information is categorized into three types: IP basic information, flow state information and running variable. Flow state information includes transfer data bytes, packets number, outbound packet rate, outbound bandwidth, average packet size, start time and end time of the flow and specially marked items. For example, packets with a size of more than 1000byes and destination addresses that are super nodes or Skype clients are marked.

The database adopts SQLite because it embeds in the program and SQLlit engine is not a dependent process. The database operates using API technique. This helps to decrease CPU cost and relay time and enhance stability.

B. Skype Voice Traffic Measurements

Two traffic measurements were taken. The first measurement was taken from a LAN network and the other one was from one of the largest Internet providers in Beijing.

We implement our system on 32-bit and 64-bit Red Hat hosts and adopt language C and Glib. SQLite 3.0 is used as local data storage.

Developed Host: Operation System Red Hat Enterprise AS4
CPU Intel® Xeon™ 3.20G hz
Memory 2GB

Validate Host: Operation System Red Hat Enterprise Server release 5
CPU Intel® Xeon™ 3G hz
Memory 32GB

The first experiment is carried out within a campus LAN. It provides sufficient background traffic and helps to validate the accuracy of our identification system. The LAN is consisted of about 300 different hosts and servers and has eight C IP addresses. The experiment includes 5 subsets. Each subset includes 10 logins. In the experiment, each login was performed at a different host in the LAN and voice call connections with a buddy were established after login. The buddy's addresses are in the Beijing and Heilongjiang. Three of them are behind a NAT. The experiment environment is shown in Fig. 1.

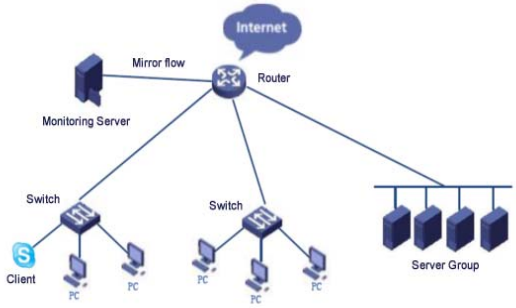

Figure 1. Ethernet Topology

After voice call connections become stable, all Skype traffics are detected using our Skype traffic identification system. We observed that, on average, voice call connections were identified 2 minutes after they were established. We noticed that if we relied solely on Skype voice flow features to identify voice traffic, false positives would occur about 7 times in 50 trials. That amounts to a false positive rate of 14%.

The other experiment was carried out within one of the largest Internet providers of CERNET (China Education and Research Network) and captured the packets passing by the CERNET in real time. We present below the results on Skype client collection, super node collection and Skype traffic identification.

We obtained 22995 Skype clients and the geographical distribution of Skype clients is shown in Fig. 2. Since we obtained only partial network traffic, the corresponding

Skype clients did not represent Skype's global clientele. The figure shows the geographical distribution of the Skype clients we obtained. Most of them were in Taiwan and Mainland China.

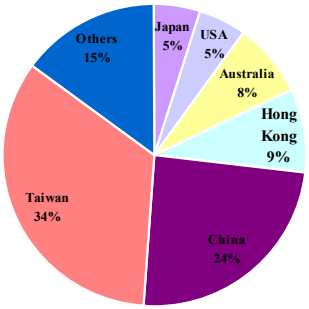

Figure 2. Geographical Skype Client Distribution

We obtained 37638 Skype super nodes and the geographical distribution of Skype super nodes that were out of China is shown in Fig. 3. The figure shows that most of the non-chinese super nodes we obtained were in Spain, American and Italy.

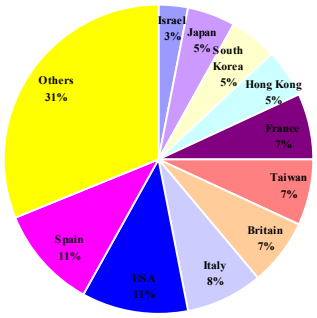

Figure 3. Geographical Distribution of Super Nodes Out of China

We notice patterns in super node discovery (shown in Fig. 4). The observation period can be divided into two phrases. One is from 0 to the 33rd hour. During this period, the number of super node being discovered was increasing linearly with a speed of 200 -- 300 per hour. At this phase, the system just started to run and the number of clients we obtained was small. The number of super nodes discovered linearly corresponds to the number of Skype clients

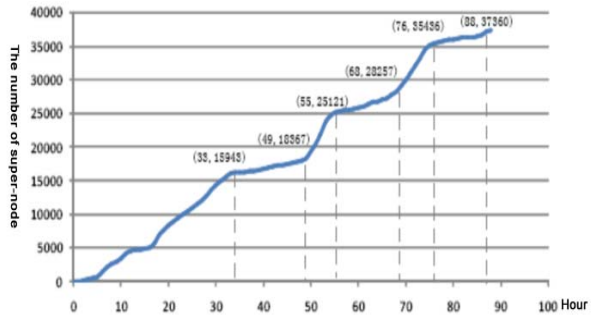

Figure 4. Super Node Growth Curve in Backbone Network

discovered. The second phase is from the 34th hour to the end. Within this second phase, the pattern of super node discovery resembles a step-function. The speed of super node discovery was slow for about 15 hours, and then accelerated for about 6 hours. This cycle repeated itself afterwards.

Our experiment lasted 93 hours. We dealt with a huge data volume that contained over 30TB and identified 21586 flows.

We also investigated the accuracy our Skype traffic identification system. The logins and voice communications of Skype were conducted 20 times within the campus LAN through the CERNET network. Each voice connection lasted for 10 minutes. Each Skype voice traffic was detected accurately by our system and each Skype client information was recorded.

CONCLUSIONS

This paper studies the identification of Skype traffic. We analyze Skype client logins using a black-box approach. We first propose an algorithm to identify Skype clients and build a database to collect the Skype clients based on payload features and time order. With further investigations, we find that through either direct connection, revisal connection, or relay connection, Skype clients must exchange messages with super nodes before or during a voice call communication. So we propose a method of Skype traffic identification using Skype client information, super node information and voice flow features. The method simplifies the identification of Skype voice traffic and enhances accuracy without adding further complexity when compared to the flow-feature-only approach.

Utilizing the proposed algorithm, we design and implement a Skype traffic identification system. The system was applied to identify Skype traffic in a LAN network. We detected all the Skype voice traffic in the LAN network with high accuracy. We also operated the system in one of the largest Internet providers of CERNET for 93 hours. 21586 flows were identified among over 30TB data. We believe our proposed method can accurately and effectively identify previously hard-to-detect Skype traffic, especially voice service traffic.

REFERENCES

[1] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," ACM SIGMETRICS'98, Wisconsin, USA, pp. 151–160, Jun. 1998.

[2] D. Rossi, M. Mellia, and M. Meo, "A detailed measurement of Skype network traffic," In 7th International Worshop on P2P Systems (IPTPS'08), Florida, USA, February 2008.

[3] M. F. Arlitt and C. L. Williamson, "Internet web servers: workload characterization and performance implications," Networking, IEEE/ACM Transactions, vol. 5, no. 5, pp. 631–645, Oct 1997.

[4] S. Baset and H. Schulzrinne, "An analysis of the Skype peer-to-peer Internet telephony protocol," in Proceedings of IEEE INFOCOM'06, Barcelona, Spain, pp. 1-3, Apr 2006.

[5] S. Ehlert, S. Petgang, T. Magedanz, D. Sisalem, and K. Elissa, "Analysis and signature of Skype VoIP session traffic," In 4th IASTED International Conference on Communications, Internet and

Information Technology(CIIT'06), US Virgin Islands, pp. 83-89, Nov/Dec 2006.

[6] S. Guha, N. Daswani, R. Jain, and R. Nicole, "An experimental study of the Skype peer-to-peer VoIP system," in 5th International Workshop on Peer-to-Peer Systems(IPTPS'06), CA, USA, pp. 1-6, February 2006.

[7] M. Perényi, A. Gefferth, T. D. Dang, and S. Molnár, "Skype traffic identification," IEEE Globecom 2007, Washington, DC, USA, Nov. 2007.

[8] W. Ghandour, "Blocking Skype using squid and OpenBSD," Help Net Security (www.net-security.org), 2005.

[9] K. T. Chen, C. Y. Huang, P. Huang, and C. L. Lei, "Quantifying Skype user satisfaction, " in Proceedings Of SIGCOMM'06, Pisa, Italy, 2006.

[10] K. Suh, D. R. Figueiredo, J. Kurose, and D. Towsley, "Characterizing and detecting Skype-relayed traffic," in Proceedings of INFOCOMM'06, Barcelona, Spain, pp. 1-12, 2006.

[11] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed analysis of Skype traffic," IEEE Transactions on Multimedia "1", Vol. 11, No. 1, January 2009, pp. 117-127, doi: 10.1109/TMM.2008.2008927.

[12] S. L. Garfinkel, "VoIP and Skype security," Jan. 2005.

[13] B. Y. Zhao, Y. Duan, L. Huang, A. D. Joseph, and J. D. Kubiatowicz, "Brocade: Landmark Routing on Overlay Networks," In 1th International Worshop on P2P Systems (IPTPS'02), Cambridge, MA, pp. 34-44, Mar. 2002.

[14] Z. Xu and Y. Hu, "SBARC: a supernode based peer-to-peer file sharing system," In Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC'03), Antalya, Turkey, pp. 1053, July 2003.

[15] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutellalike P2P systems scalable," In Proceedings of SIGCOMM '03, Karlsruhe, Germany, pp. 407-418, Aug. 2003.

[16] M. Castro, M. Costa, and A. Rowstron, "Debunking Some Myths about Structured and Unstructured Overlays," In Proceedings of the NSDI '05, Boston, MA, May 2005.

[17] F. Desclaux, "Skype Uncovered," EADS/CRC, Technical Report, November 2005.