High-Speed Network Traffic Analysis: Detecting VoIP Calls in Secure Big Data Streaming

Mazhar Rathore, Anand Paul*, Awais Ahmad
The School of Computer Science and Engineering,
Kyungpook National University, Daegu, South Korea
(rathoremazhar,*paul.editor) @gmail.com,
awais@knu.ac.kr

Muhammad Imran
College of Computer and Information Sciences King Saud University
Saudi Arabia
dr.m.imran@ieee.org

Mohsen Guizani
Department of Electrical and Computer Engineering,
University of Idaho, USA.
mguizani@ieee.org

***Abstract*—Internet service providers (ISPs) and telecommunication authorities are interested in detecting VoIP calls either to block illegal commercial VoIP or prioritize the paid users VoIP calls. Signature-based, port-based, and pattern-based VoIP detection techniques are not more accurate and not efficient due to complex security and tunneling mechanisms used by VoIP. Therefore, in this paper, we propose a rule-based generic, robust, and efficient statistical analysis-based solution to identify encrypted, non-encrypted, or tunneled VoIP media (voice) flows using threshold approach. In addition, a system is proposed to efficiently process high-speed real-time network traffic. The accuracy and efficiency evaluation results and the comparative study show that the proposed system outperforms the existing systems with the ability to work in real-time and high-speed Big Data environment.**

***Index Terms*— VoIP, Big Data, Tunneling, Hadoop, Spark.**

I. Introduction

Recent advancement in the VoIP for commercial usage is getting attention from the academia as well as research community due to various features, such as cost, effectiveness, and its compatibility with the Public Switch Telephone Network (PSTN). There are two key steps involved in VoIP i.e., signaling and media channel setup. The call is setup and the connection is established using signaling protocols, such as Session Initiation Protocol (SIP) and H. 323, whereas the 2nd step i.e., media channel setup, is responsible for voice transmission between two communicating parties using Real-time Transport Protocol (RTP). In some of developing countries, the usage of VoIP for commercial purpose is prohibited since it incurs huge financial loss to telecom authorities. Therefore, organizations are interested in detecting VoIP calls either to block or prioritize.

One of the big challenge in VoIP calls detection is the advancement in communication technologies, which increases the amount and velocity of data generated over the internet. In 2008, the number of UK houses connected to the internet is increased to 65% [1]. Also, in 2012, the number is reached to 80% and the amount of data generated by computers was recorded as 2.27 zettabytes [2]. Therefore, a system is needed that handles high velocity of data during monitoring and analyzing network traffic in a real-time. Such enormous amount data at a high velocity at different variety is leading us toward the concept of Big Data.

In the literature, four types of VoIP detection methodologies are provided, namely; Port-based, signature-based, pattern-based, and statistical analysis-based techniques [3]. Port-based, signature-based, and pattern-based techniques are not accurate due to complex security and tunneling mechanisms used by VoIP. Statistical analysis-based techniques are useful for tunneled traffic analysis, which consider packet size, arrival time, number of packet, etc., as basic parameters for VoIP flows analysis. However, most of them are specific to some protocols and applications. Whereas others do not provide real-time detection in a high-speed network environment. Various statistical-based techniques are given in the literature, which are used to detect VoIP [4-10]. Host and flow behavior analysis (HFBA) [4] analyzed VoIP by examining the ports and IP addresses. In [5], authors separate VoIP traffic by considering traffic features, which are challenging to modify i.e., packet interval time, packet size, exchange rate, etc. In [6], a mechanism is described that detects hidden VoIP calls in web traffic on port number 80 and 443 by using web request size, web response size, inter-arrival time, etc. Another technique is proposed that detects VoIP traffic using flow features, i.e., size and time using three different machine learning techniques [7]. A mechanism called flow level behavior (FLB) is proposed [8] that detects VoIP using packet size and inter-arrival time. Taner proposed a technique [9] for VoIP identification hidden in the IPsec tunnel by comparing packet size with the threshold. Similarly, Branch and But's statistical technique [10] requires a part of flow for voice classifiers. However, this technique can only be applied in the two-way interface.

We have noticed that exiting statistical techniques fail to meet the basic requirements to be generic, efficient, and independent from VoIP application and protocols. Also, they are not capable of processing ultra-high-speed traffic at real-time. Having the aforementioned schemes and their drawbacks, in this paper, we present a statistical analysis-based solution to all those problems that are addressed above by considering various rules for statistical parameters in order to identify the VoIP media flow. The proposed solution is generic, efficient and accurate by considering real-time scenarios and is able to detect encrypted as well as tunneled VoIP. Furthermore, the proposed scheme is independent of any VoIP application protocol, security mechanisms, or tunneling mechanism.

II. VoIP Traffic Analysis and Discussion

The main purpose of the data traffic analysis is to identify the particular features of the VoIP flows to make a distinction between VoIP and non-VoIP flows. The data is collected from

DOI 10.1109/LCN.2016.128

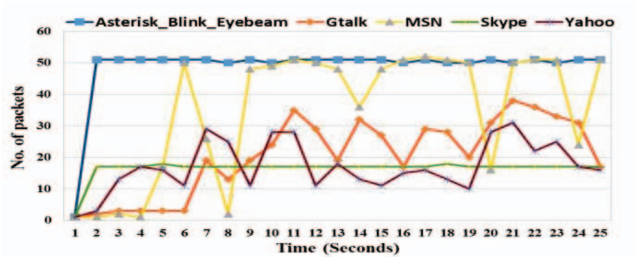

Fig.1. Packer rate analysis of various VoIP applications

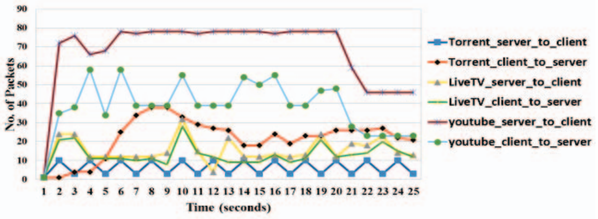

Fig. 2. Packet rate analysis of various high-packet-rate non-VoIP applications

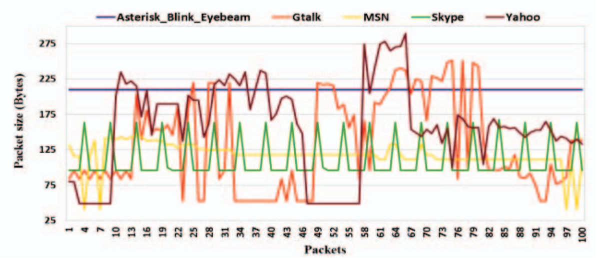

Fig. 3. Packer size distribution of various VoIP applications

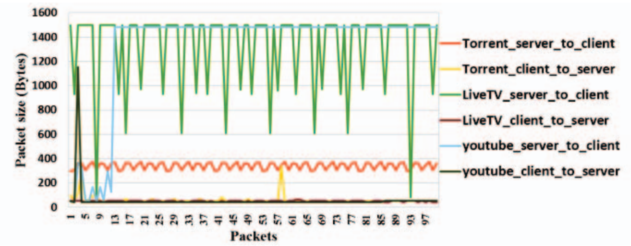

Fig. 4. Packer size distribution of various high-packet-rate non-VoIP apps.

the various location for analysis and evaluation. Most of the datasets are captured from a university lab, from home user's PC, and from two telecommunication authorities' gateways. The telecommunication authorities' datasets are captured in a very high-speed environment in chunks of 1GB, 2GB, and 4GB. We also established our own VoIP setup using Asterisk as a telephone Private Branch Exchange (PBX) by using Zfone, X-Lite, Eyebeam, Blink as a client and SSL/TSL, IPSec, and SRTP as secure tunnels. Skype datasets are also taken for testing purpose from tstate skype repository [11]. Besides, we also use captured traffic of Bluetooth, chatting, DNS, Frame relay, document retrieval, and etc. More than 10 VoIP applications including SKYPE, GTalk, and Yahoo messenger.

The analysis is made on each flow, distinguished by 4-tuples i.e., source-IP, destination-IP, source-port, and destination-port (S-IP, D-IP, S-Port, D-Port). In case of IPsec flows, the flow is recognized as 3-tuples i.e., source-IP, destination-IP, and security parameters index (S-IP, D-IP, and SPI). The delay, latency, jitter, and packet loss are the factors, which affect the quality of voice transmission. More latency, jitter and packet loss degrade the voice quality at the destination end. The packet size is a key element to control these factors; considering all of these facts, the IP layer packet size is considered as a basic parameter for VoIP flows analysis. In addition, reducing the size of packet increases the number of packets per voice transmission. Therefore, we have taken packet size distribution (PSD) and the packet rate (packet sent per second) for initial analysis. Figure 1 shows the packet rates of various VoIP application. The graph clearly shows the high packet rate, mostly from 15-52 packets/sec. yahoo messenger is the only application that has fewer packet rate as compared to other application. However, while analyzing non-VoIP application, we found few non-VoIP flows that also have high data rate as VoIP flows, as shown as packet rate graphs in Figure 2. Therefore, packet rate cannot be the only parameter to judge. Thus, we analyzed both types of high packet rate flows with respect to packet size distribution. The packet size distribution for both VoIP and non-VoIP applications are depicted in Figure 3 and Figure 4 respectively. In the case of the identified non-VoIP high packet rate applications, the packet sizes are quite larger (mostly the communication from server to client) or quite smaller (from client to server), as clearly shown in Figure 4. Hence, based on these findings, we 2) $\bar{X}$ (packet size), and 3) S.D (packet size). It is also a natural process that speech has continuous behaviors, a person continuously speaks until he/she completed his talk. Therefore, a lot of packets are transmitted between both of parties with very short time duration. In view of this factor, we selected three more parameters for analysis i.e., 4) max-diff-time (Maximum time duration between current and previous packet in seconds), 5) $\bar{X}$ (diff-time), and 6) S.D (diff-time) as statistical measures in order to perform time-based analysis to distinguish voice flows. We also took the difference measures of time, such as ($\bar{X}$ (diff-time)-S.D (diff-time)) to cater some of the abnormalities exist in some of the VoIP applications.

III. Proposed System

A. Proposed Rule-based VoIP Flows Detection

The complete pseudocode of the proposed mechanism is presented as algorithm 1, which classify the network flows as VoIP or non-VoIP. The symbols, functions, and the threshold values used in the algorithm are presented in Table I.

The proposed algorithm takes the network packets for each flow, distinct by either 3-tuples (SIP, DIP, SPI) in case of IPsec or 4-tuples (SIP, DIP, Sport, DPort) in case of other type of packet, as input and classifies the flow as VoIP or non-VoIP based on the flow parameters. Two types of rules are formed using threshold values of the selected parameters i.e., basic rule (rule 1-4) and auxiliary rules (5-8). All basic rules are mandatory to be fulfilled in the case of VoIP calls, while among auxiliary rules, at least three rules must be satisfied for a flow to be a voice. The non-VoIP flows are identified by two phases. In the first phase, the non-VoIP flows are directly identified based on basic rules as mentioned in the 6th step of the algorithm. In the second phase, some of the flows are detected as suspected and need to be re-investigated for next five seconds traffic. If any flow is detected as suspected, it is categorized as non-VoIP.

TABLE I. SYMBOLS USED IN ALGORITHM 1

Symbol	Description
Pkt_size	Packet size in bytes
Curr_F	Current flow being captured or processed
Reg_F	Registered flows(not yet detected as VoIP/Non-VoIP).
Pkt_time	Packet transmission time
Curr_F	Current flow being captured or processed
Det_F	List of flows already detected as VoIP/Non-VoIP.
SqF$_i$	Sequence file containing packet parameters of Flow i.
Type()	get the type of the Packet (IPsec/Non-IPsec)
Extract()	Extract the required packet parameters
add_in_SqF()	Add packet parameters into the flow sequence file
No.pkt()	Getting number of packet for a particular flow
F_Duration()	Getting the overall flow duration in seconds
Cal_F_Param()	Calculate the flow parameters
No.Time.Suspected()	Get number of time a flow is detected as suspected.
λ	Threshold (Mn: Minimum, Mx: Maximum)

Algorithm 1: VoIP Flows Detection Algorithm

```
INPUT:  Network Packets
OUTPUT: Flows=VoIP/Non-VoIP
RULES:
Basic Rules
1. pkt-rate > λPrate
2. λMn_X̄Size ≤ X̄(size) ≤ λMx_X̄Size
3. λMn_SDSize ≤ S.D(size) ≤ λMx_SDSize
4. X̄(size) ≤ S.D(size)
Auxiliary Rules:
5. λMn_max_diff_time < max_diff_time ≤ λMx_max_diff_time
6. λMn_X̄diff_time < X̄(diff_time) ≤ λMx_X̄diff_time
7. λMn_SDdiff_time < S.D(diff_time) ≤  λMx_SDdiff_time
8. λMn_mean-SDTime < │X̄(diff_time)-S.D(diff_time) │≤ λMx_mean-
SDTime
STEPS:
ForEach(Packet pkt)
1.    IF (Type(pkt)=IPsec)
          Curr_F=Extract(SIP, DIP, SPI)
      Else
          Curr_F=Extract(SIP, DIP, Sport, DPort)
      EndIF
2.    IF (Curr_F ε Det_F)
          ReturnTo--->Next_pkt
      EndIF
3.    IF (Curr_F ε Reg_F)
          Reg_F:= Reg_F+ Curr_F
          SqFCurr_F= add_in_SqF(pkt_Time, pkt_Size)
          ReturnTo--->Next_pkt
      EndIF
4.    IF (No.pkt(SqFCurr_F) < λMx_VoIP_F_pkt ||
      F_Duration(SqFCurr_F) < λMx_VoIP_F_time)
          SqFCurr_F= SqFCurr_F + add_in_SqF(pkt_Time, pkt_Size)
          ReturnTo--->Next_pkt
      EndIF
5.    Cal_F_Param(SqFCurr_F)
6.    IF(No.Time.Suspected(Curr_F) ≥ 3)
          Curr_F=Non-VoIP
          ReturnTo--->Next_pkt
      ElseIF (Basic Rules=True && atleast3(Auxiliary Rules) = True)
          Curr_F=VoIP
          ReturnTo--->Next_pkt
      ElseIF(Rule 1 = True && atleas1(Rule 2,Rule 3,Rule 4) = False)
          Curr_F=Non-VoIP
          ReturnTo--->Next_pkt
      Else
          Curr_F= Suspected
          ReturnTo--->Next_pkt
      EndIF
End ForEach
```

IV. SYSTEM IMPLEMENTATION AND EVALUATION

A. Implementation environment

The system is implemented using Hadoop ecosystem in a single data node environment at Ubuntu 14.04 LTS coreTMi5 machine with 3.2 GHz x 4 processors and 4 GB memory. The Hadoop performs the parallel processing using its mapper and reducer programming nature and the distributed file system HDFS. The parallel processing nature of Hadoop makes the overall processing too fast. Traditionally, Hadoop is developed for batch processing. However, we used Spark with Hadoop in order to perform real-time streaming analysis by taking the both benefits of parallel processing of Hadoop and real-time processing engine of Spark. We put a separate module to capture high-speed traffic using very fast and high-speed capturing device and driver i.e., RF_RING and TNAPI [12]. We use In-Memory database, which contains two types of information i.e., 1) currently being processed flows and 2) the VoIP/non-VoIP classified flows. The use of In-Memory databases, containing the process information being processed and classified flows, makes the filtration process too fast. PcapInputFormat, TextInputFormat, Hadoop-pcap-lib, Hadoop-pcap-serde [13] libraries are used to process network data in packet's format. The use of In-Memory database, Hadoop ecosystem, SPARK, and the strong filtration mechanism make the overall system more proficient.

B. System evaluation

The system is evaluated by considering three aspects i.e., the accuracy of the system, the efficiency of the system, and the scalability of the system by comparing it with the existing scheme. For accuracy, we use typical parameters for accuracy testing, such as truea positive (TP): is the measure of flows that are correctly identified as VoIP flows, false negative (FN): is the measure of flows that are incorrectly identified as non-VoIP flows, true negative (TN) is the measure of flows that are correctly identified as non-VoIP flows, and false positive (FP): is the measure of flows that are incorrectly identified as VoIP flows. DR reflects how much VoIP flows are correctly identified as VoIP flows and calculated by *TP/(TP + FN)*. FPR reflects how much non-VoIP flows incorrectly identified as VoIP flows and calculated by *FP/(FP + TN)*. Table II shows overall accuracy results on all the offline plus real-time traffic traces. Our system has 97.54 % DR which is quite higher and .00015% FPR, which is quite lower.

The efficiency is measured in terms of VoIP flow detection time. The detection time of voice calls is less than 6 seconds; as we only consider the small part of the flow traffic (i.e. first 60

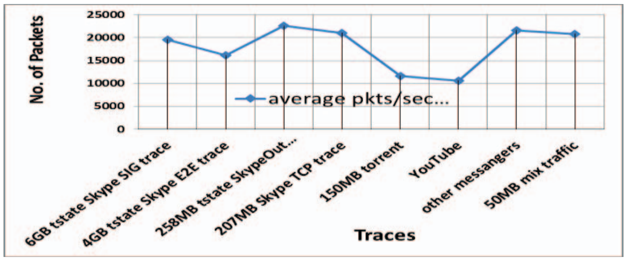

Fig. 5. System Efficiency in terms of average number of packet processed per second

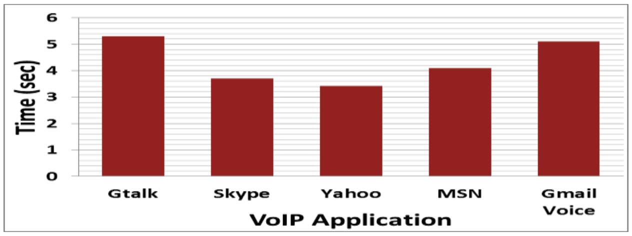

Fig. 6. System Efficiency in terms of VoIP flows detection time on various VoIP applications.

packets or packets within 5 seconds for each flow) for VoIP detection. Figure 5 shows the system's data processing rate as average processed packets/sec and Figure 6 presents the average detection time of voice flows on different VoIP applications.

We compare our technique with HFBA technique [4], threshold-based detection [5], and IPsec VoIP detection[9], FLB [8] in terms of TP, FP, and FN. HFBA [4], threshold-based detection [5], and FLB techniques gave more importance to Skype voice traffic in analysis and testing, so we consider larger size 3.5 GB tstat Skype trace "Internet-Skype-UDP-E2E" [11] for comparing these VoIP detection techniques with our technique. Figure 7 shows overall comparison. We have observed that our system performs better than existing techniques with respect to accuracy and efficiency.

V. Conclusion

The proposed system presented in this paper is generic, does not depend on any VoIP application, protocol, codec, and security mechanism, can detect encrypted tunneled VoIP, and implementable at either one-way or two-way network interface. It meets the need of any organization to detect VoIP flows to either prioritize or block. We test our solution on many traces of more than 10 VoIP applications. The comparisons and results show that our technique is the best among all the existing techniques. This technique has 97.54% TP and .00015% FP. It is the better choice for telecommunication authorities and ISPs to detect VoIP calls in high-speed big Data environment.

TABLE II. Overall system accuracy

	TP	FN	FP	TN	DR	FPR
Online streaming	37	3	1	5000	92.5	.0002
Offline datasets	56	2	2	15000	96.56	.00013
Self-established VoIP Setup	16	0			100	---
Tstate Skype datasets	882	20			97.78	
Overall					97.54	.00015

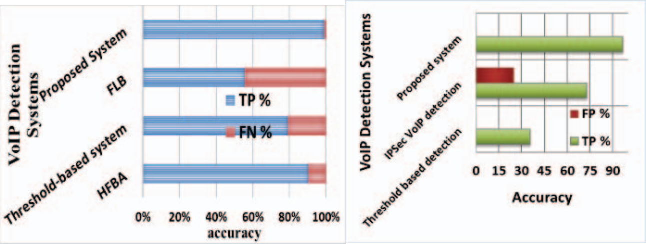

Fig. 7. Accuracy comparison of the proposed system with existing technique

Acknowledgment

This research was supported by Kyungpook National University Bokhyeon Research Fund, 2015. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP). [No. 10041145, Self-Organized Software platform (SoSp) for Welfare Devices].

References

[1] Vegard Engen, "machine learning for network based intrusion," Ph.D. dissertation, Bournemouth Univ., Poole, UK, 2010.

[2] S. Sagiroglu and D. Sinanc, "Big Data: a review," 2013 International Conference on Collaboration Technologies and Systems (CTS), pp. 42-47, IEEE, 2013.

[3] M. M. U. Rathore and T. Mehmood, "Research on VoIP traffic detection," 2012 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Genoa, 2012, pp. 1-5.

[4] Bing Li, Shigang Jin, and Moade Ma, "VoIP Traffic Identification Based on Host and Flow Behavior Analysis," Journal of Network and Systems Management, Volume 19, 2010.

[5] Fauzia Idrees and Uzma Aslam Khan,"A Generic Technique for Voice over Internet Protocol (VoIP) Traffic Detection," IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.2, pp 52-59, 2008.

[6] Emanuel P. Freire, Artur Ziviani and Ronaldo M. Salles, "Detecting VoIP Calls Hidden in Web Traffic," IEEE transaction on network and service management, Vol no. 5, pp- 210-214, 2008.

[7] Riyad Alshammari and Nur Zincir-Heywood, "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?'" ELSEVIER: Computer Networks, vol. 55, pp 1326-1350, 2011.

[8] T. Okabe, T. Kitamura, and T. Shizuno, "Statistical Traffic Identification Method Based on Flow-Level Behavior for Fair VoIP service," Proceeding of IEEE Workshop on VoIP Management and Security, pp. 35-40, 2006.

[9] Taner Yildirim and Dr. PJ Radcliffe, "VoIP Traffic Classification in IPsec Tunnels," 2010 International Conference on Electronics and Information Engineering (ICEIE), Koyoto, Japan, pp VI-151-VI-157, 2010.

[10] P. Branch and J. But, "Rapid and Generalized Identification of Packetized Packetized Voice Traffic Flows," 37th IEEE Conference on Local Computer Networks (LCN12), Clearwater, Florida, October 2012.

[11] http://tstat.tlc.polito.it/traces-skype.shtml

[12] F. Fusco and L. Deri, "High Speed Network Traffic Analysis with Commodity Multi-core Systems," ACM IMC 2010, Nov. 2010.

[13] Hadoop library to read packet capture (PCAP) files. "Available online:https://github.com/RIPE-NCC/hadoop-pcap," Accessed on 1 April, 2016.