Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 888 – 900

doi:10.3906/elk-1501-126

An effective empirical approach to VoIP traffic classification

Nazar Abbas SAQIB[1,*], Yaqoot SHAKEEL[1], Moazzam Ali KHAN[1], Hasan MAHMOOD[2], Muhammad ZIA[2]

[1]College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad, Pakistan

[2]Department of Electronics, Quaid-i-Azam University, Islamabad, Pakistan

Received: 26.01.2015 • **Accepted/Published Online:** 07.03.2016 • **Final Version:** 10.04.2017

Abstract: It is beneficial for telecommunication authorities and Internet service providers (ISPs) to classify and detect voice traffic. It can help them to block unsubscribed users from using their services, which saves them huge revenues. Voice packets can be detected easily, but it becomes complicated when the application or port information in the packet header is hidden due to some secure mechanism such as encryption. This work provides effective voice packet classification and detection based on behavioral and statistical analysis, which is independent of any application, security protocol, or encryption mechanism. First we have made initial assessments through packet feature analysis followed by the implementation of a voice detection algorithm to perform statistical analysis for classifying traffic over IP networks. The proposed voice detection algorithm is executed in three phases: registering of packet flow traces, signature-based analysis, and voice classification. In the first phase, new packets are registered. In the second phase, registered packets are tested if they are already marked as detected. In the third phase, the voice detection algorithm works at distinguishing encrypted and nonencrypted voice flows by fine-tuning the parameters, which are chosen after a detailed statistical analysis of datasets on security protocols such as secure socket layer, secure session initiation protocol, and secure real-time transport protocol. Our results demonstrate a high true positive rate (TPR) and very low false alarm rate (FAR). The proposed methodology achieves a TPR of 93.6% for offline traces, 100% for the self-configured voice setups, and 95% for the online traffic. The FAR is 0.000084% for offline traces and 0.00020% for online traces, which shows that the proposed methodology is highly efficient and can be incorporated in contemporary telecommunication systems.

Key words: Traffic classification, Voice over Internet Protocol, secure socket layer, encrypted and decrypted voice

1. Introduction

Similar to other services on IP networks, Voice over Internet Protocol (VoIP) traffic can be eavesdropped, hacked, or spoofed, which results in serious security threats to VoIP users as well as to service providers. VoIP users can use security protocols on both the sending and receiving sides to guard their media sessions, whereas VoIP service providers cannot use security protocols. They can protect their services by detecting illegal voice traffic and stopping or blocking it. In this way, VoIP providers can save additional profits that cannot be earned otherwise.

The practice of typecasting computer network traffic based on diversified constraints such as port numbers and protocols into distinct traffic classes is called traffic classification. Several traffic classification techniques reported in the literature use deep packet inspection to identify specific VoIP applications by creating a reference

*Correspondence: nazar.abbas@ce.ceme.edu.pk

database, which requires a continuous updating of signatures [1–3]. Pattern-based techniques are dependent on signaling mechanisms as each application has its own signaling patterns [4]. Traffic classification techniques based on source and destination IP addresses or source and destination ports [5] are limited to some extent due to hiding transport and network layer information through encryption mechanisms.

In this work, we propose a more precise generic voice packet classification and detection technique based on packet features and statistical analysis. The proposed method is independent of an application, security protocol, or encryption mechanism for voice traffic over IP networks with low probability of false alarm. It first makes initial assessments through packet feature analysis and then performs statistical analysis on the acquired data by following the proposed voice detection algorithm. We have performed analysis on different voice and nonvoice applications such as Skype, Yahoo messenger, Gmail, MSN messenger, user voice setups with different configurations, and Facebook. We also analyze other applications such as Hotmail, Yahoo, Gmail, media sessions, online gaming, torrent downloading, and online live TV. The aforementioned applications along with additional applications like Tencent QQ messenger, Trillian IM, and TEAMtalk have been used for testing purposes. Test results demonstrate that the proposed voice traffic detection technique not only exhibits low false alarm rates but it is also adaptable by telecommunication authorities and ISPs to detect voice traffic. The rest of the paper is organized as follows: Section 2 provides the related work. The proposed strategy for traffic classification is presented in Section 3. Details about our experimental setup and datasets are provided in Section 4. Results and performance analysis are presented in Section 5. Finally, conclusions are presented in Section 6.

2. Related work

Traffic classification techniques are helpful to categorize diverse applications and protocols that exist in a network and to detect the voice traffic over IP networks. These classification techniques are organized as port-based, pattern analysis, statistical analysis, deep packet inspection, heuristic analysis, and numerical analysis techniques. In port-based techniques [1], packets are classified based on the fields of the packet header such as the source or destination ports at the transport layer. It is an ultimate, customary, and simple technique but less accurate [5]. Pattern-based techniques perform analysis on the signaling pattern where certain patterns such as bytes, characters, or strings are embedded in the application. These patterns help in identifying applications or protocols [4]. This technique entirely depends on the call signaling mechanism of VoIP traffic. Standard and proprietary protocols can be powerfully exploited by inspecting the signaling patterns. The drawback of this technique exists due to the fact that the pattern for every application is different and therefore not generic. In signature-based analysis/deep packet inspection, every application has its own unique signature that symbolizes its unique characteristics and a reference database is created. This reference is then used to identify the application and needs to be updated periodically [1–3]. Techniques based on numerical analysis involve numerical attributes of the traffic like payload size, offsets, and number of response packets [6]. Techniques using behavior or heuristic analysis investigate the behavior [5,7,8] and heuristics [4] of the network traffic, which sometimes yield better insight to identify the application. In [9], port-based analysis is used as helping information to detect VoIP. Pattern-based detection usually involves machine learning techniques, which were first used in traffic classification for intrusion detection [10]. In [11], supervised learning was used with three extractable properties of IP packets: packet length, interarrival time, and order of arrival. The parameters are used to develop protocol fingerprints. In [12] a decision algorithm was used to classify traffic by categorizing the Internet flowe into classes such as web-browsing, email, bulk FTP, and P2P. The accuracy achieved by this algorithm is around 99%.

Most of the research in the literature includes SSH or SS traffic rather than purely focusing on tunneling approaches like IPSec. In [7], the authors proposed an empirical method based on the hidden Markov model for the type of applications using tunneled protocols and demonstrated an accuracy of 20%. Some heuristic methods have been also explored based on the characteristics of the host behaviors. In [11], an approach based on the behavior of P2P peers was proposed. In [13], a multilevel traffic classification named BLINC was developed. Similarly, in [14], a methodology based on data mining and information theoretic techniques was proposed to discover the behavioral patterns of the hosts and the services provided by the hosts. In [15], the authors used a machine learning algorithm on subflows using features based on mean packet length, autocorrelation, and the ratio of data transmitted on both sides to identify variable rate VoIP flows. In [16], the authors used the Skype framing structure and exploited randomness during the encryption process, and secondly proposed a naïve Bayesian classifier by characterizing Skype traffic in terms of packet arrival rate and packet lengths. These algorithms are application-specific and successfully identify those applications even with variable data rates or different versions of them. These heuristic algorithms assume that only one network application has been in execution at one host. In reality, multiple applications may coexist.

In this work we propose a time-efficient voice packet classification and detection technique, which is a hybrid approach based on behavioral and statistical analysis of an input packet. All the packets are sorted into voice and nonvoice packets by using behavioral analysis. The packets are then further classified and confirmed as voice or nonvoice using statistical analysis. That helps in reducing the false alarm rate. Once a packet is classified correctly, it is marked as a voice or nonvoice packet. That makes our algorithm time-efficient as no statistical analysis is performed when packets of the same flow enter the next time. The main contribution of this paper is to analyze voice and nonvoice traffic, which is not application-specific, and then to classify incoming packets in the future as voice or nonvoice without any prior information.

3. The proposed strategy for identifying voice traffic

In our work, we first take IP traffic, separate out distinct flows, and analyze their features based on their flow parameters. In the second step, statistical approaches further improve the results to classify the voice traffic. We first work on the test data by capturing voice and nonvoice traces in our own setup for a number of applications to train the system. We have validated our proposed algorithm on datasets collected from 3rd party websites. A systematic procedure to execute the proposed flow-based analysis followed by our statistical model applicable to voice and nonvoice flows is presented below. We first present packet size distribution (PSD) to examine packet size and minimum and maximum packet size to define the range and variation in packet size with respect to their frequencies for both voice and nonvoice traces. PSDs of bidirectional flows for voice-based applications such as Zfone-Asterisk-Xlite, Yahoo messenger, MSN messenger, Google Talk (Gtalk), Eyebeam-Asterisk-Blink, and Skype are illustrated in Figure 1. By looking at PSDs of all voice traces, we can observe that the PSD of Skype voice traces has more variations as compared to Gtalk. Applications like Eyebeam-Asterisk-Blink and Zfone-Asterisk-Xlite show much less deviation because they are voice-specific applications and use codecs like G.711, which maintain almost constant data rates. The PSDs of voice-based traces have low packet sizes. For example, Skype has a packet size of 45–216 bytes, Gtalk has a packet size of 67–230 bytes, and MSN messenger has a packet size of 117–147 bytes.

In Figure 1, it can be seen that there are packets of small sizes, but the most frequent packets of Yahoo messenger (90 and 131 bytes), Eyebeam-Asterisk-Blink (224 bytes), and Zfone-Asterisk-Xlite (218 bytes) have packet sizes in the range of 90–220 bytes. As voice flows have small packet sizes, they are in low ranges. These low variations in packet sizes guarantee the low jitter and delay in delivering voice packets at the receiving end.

Figure 1. Packet size distributions of the voice flows.

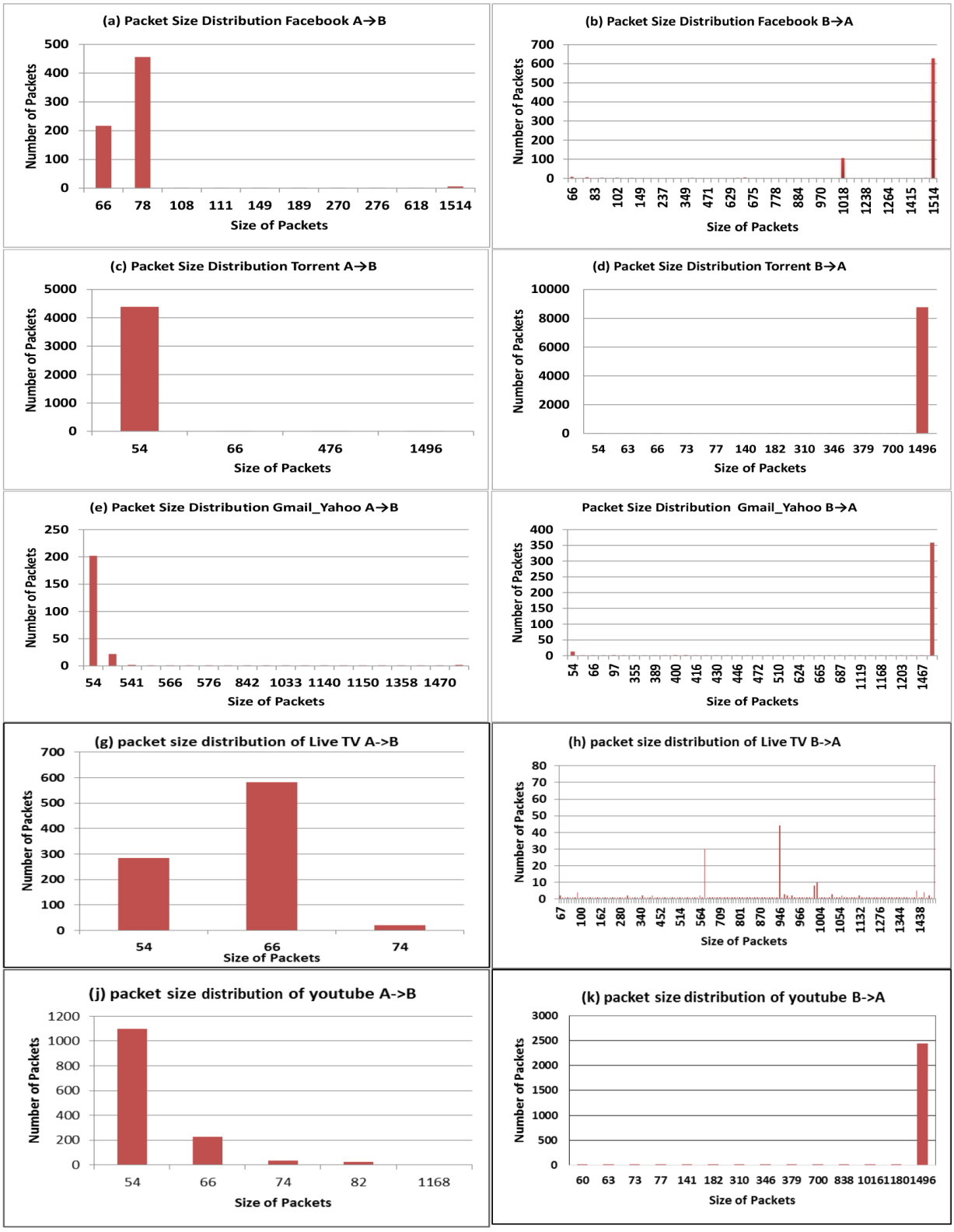

Figure 2. Packet size distributions of the nonvoice flows.

Bidirectional PSDs of nonvoice traces provide two observations. As shown on the left-hand side in Figure 2, most frequent packets of flows A to B have low packet sizes. For example, Facebook has 66 bytes, Gmail has 54 bytes, Yahoo Mail has 54 bytes, Online Live TV has 54 and 66 bytes, Torrent has 54 bytes, and YouTube has 54 bytes. The packet size ranges from 54 to 66 bytes, which is low as they all are client requests. As shown in Figure 2 on the right-hand side, the packet size lies between 1450 and 1550 bytes, and our second observation is that the flows from B to A have very large packet sizes as they start exchanging data such as pictures, movies, audio, and video after the login requests by the clients are accepted. Next we examine the packet rate (PR) of all the traces for each incoming voice and nonvoice packet. PR in flow-based techniques becomes significant as it provides the whole statistics of how may packets (Ps) a flow sends in every second (T). PR defines the average packet size for a specific time.

$$PR = \frac{\Sigma P_i}{\Sigma T_i} \tag{1}$$

Figures 3a and 3b show that the PRs of voice-dependent applications are quite high.

High packet rates depict the best quality of voice service. As voice applications are real-time applications they are sensitive to the delivery rate. Every data packet should reach its destination within its allowed time, or else it has no significance and is discarded.

Figure 4 presents the PRs of nonvoice application traces in both directions from A$\rightarrow$B and B$\rightarrow$A. As it is shown, PRs of nonvoice traces like YouTube, Torrent, and Online TVs are high in contrast to voice traces but their PSDs are different from the voice flows. PR and PSD provide vital information for traffic classification like the number of packets, interarrival time of packets, variation in packet size, packet rate, maximum and minimum packet size, and the range of packet size. Packet feature analysis on PRs and PSD of voice and nonvoice traces can lead to three main observations: 1) voice application traces have small packet sizes of $\sim$60 bytes as compared to nonvoice packets of $\sim$90 bytes; 2) voice application traces have low variations in packets, within the range of 0 to 100, whereas nonvoice applications can have variations of greater than 100; 3) voice application traces have high packet rates, greater than 12.

3.1. Voice detection algorithm

The behaviors of both voice and nonvoice traffic vary when the traffic is tunneled or encrypted through protocols like SRTP, SSL/TLS, MGCP, SIP, SMIME, or IPSec. That requires defining more precise boundaries for differentiating voice and nonvoice traffic. For this, we have extended our analysis by including more features like number of packets, total flow time, arithmetic mean, standard deviation, packet rate, and minimum and maximum size of the packet. Based on our deep analysis for traffic using secure protocols, the following algorithm works well for classification of voice and nonvoice traffic.

As shown above, the algorithm starts by measuring the total flowe time and total number of packets from the first packet to the current packet when flowe F is not yet detected as voice or nonvoice flowe. The algorithm keeps updating the flowe parameter values such as total flowe time, total flowe packets, sum of packet lengths and sum of packet lengths squares, minimum packet size, and maximum packet size until the flowe time exceeds 10 s or the total number of flowe packets exceeds 120. When both the conditions are met it computes values such as average of packet sizes, variation of packet sizes (standard deviation), packet rate, maximum packet size, minimum packet size, and range of packet size of the flowe. The packet is declared as a voice packet when the computed values stand within threshold constraints as described in Step 4 of the voice detection algorithm, but if statistical values of that flowe do not match the system threshold parameters then it is identified as nonvoice

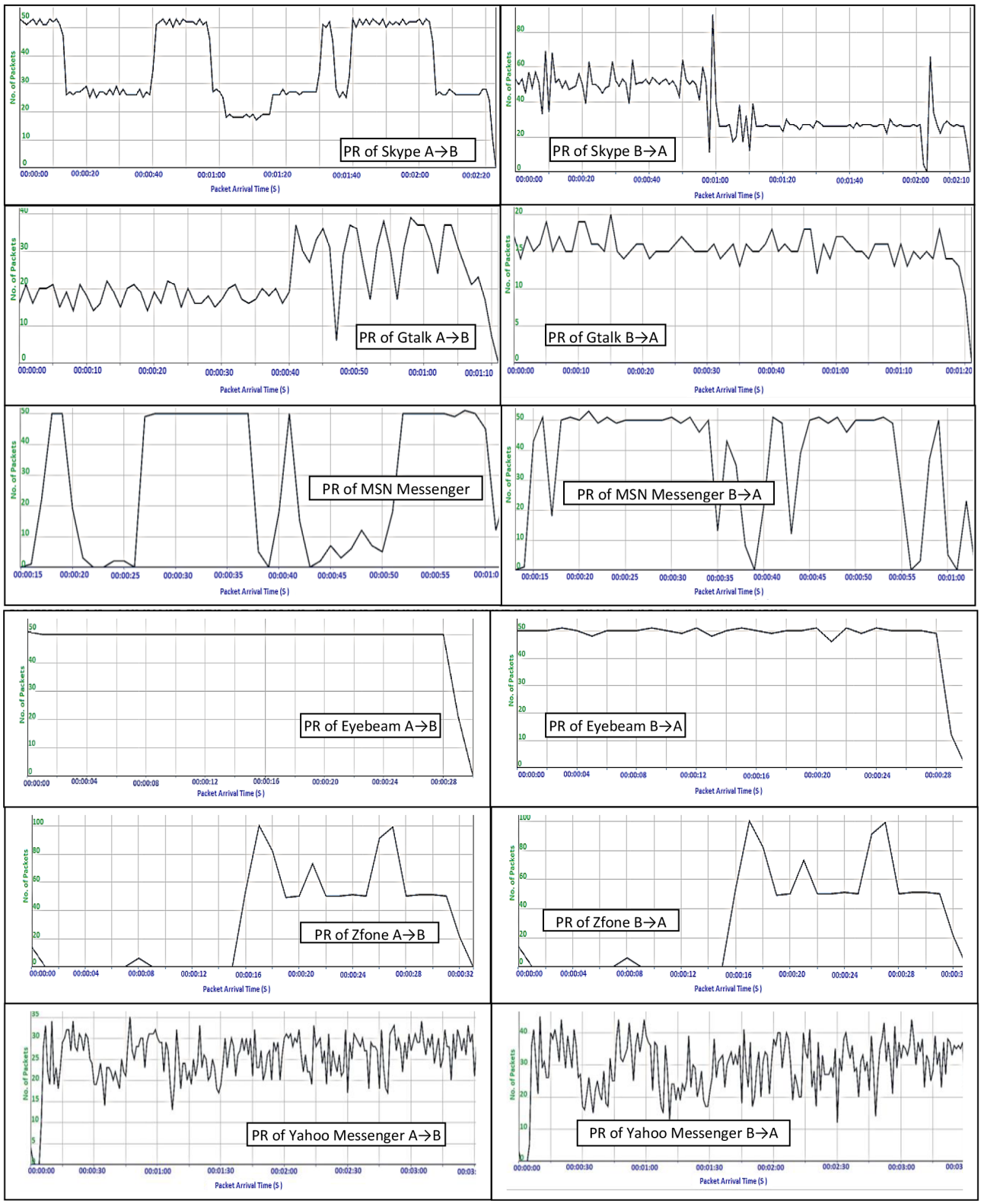

Figure 3. a) Packet rate of voice-based traces: Facebook, Gtalk, MSN Messenger. b) Packet rate of voice-based traces: Blink, Xlite, Yahoo messenger.

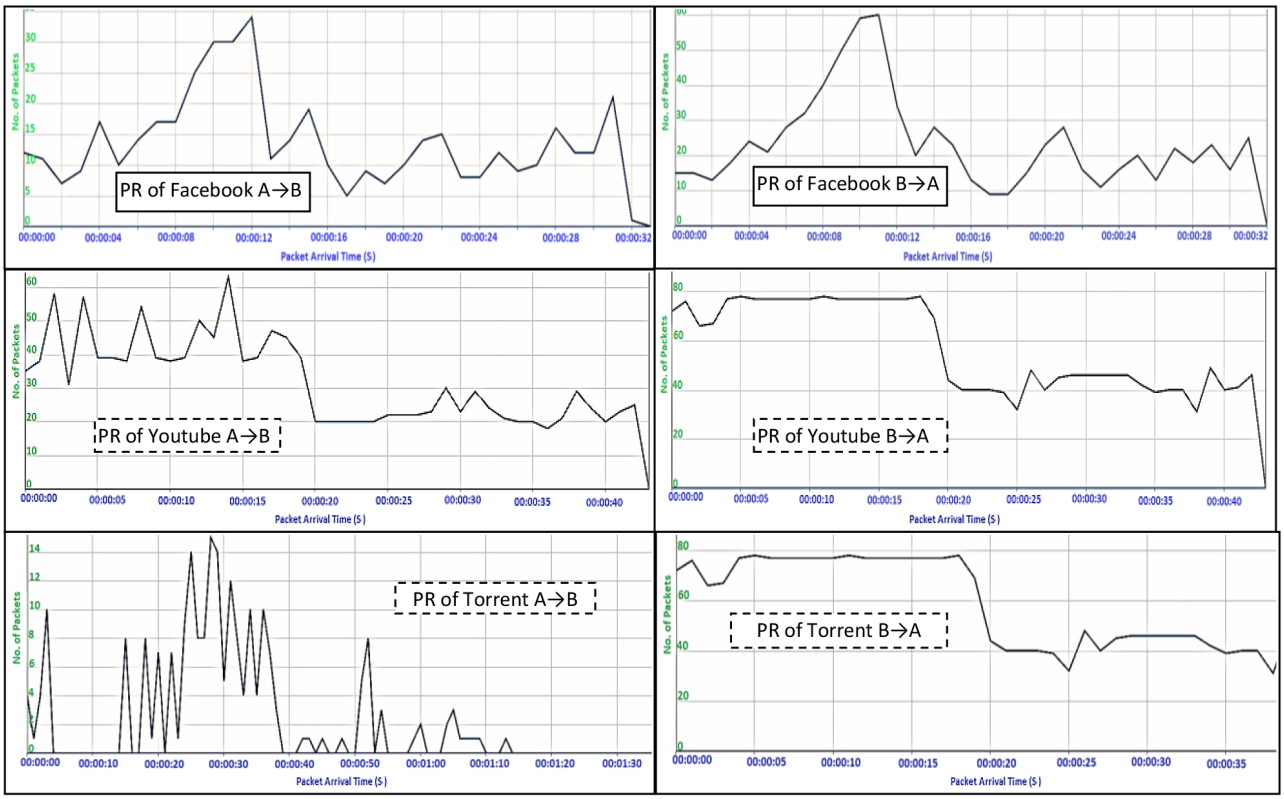

Figure 4. Packet rate of nonvoice traces.

flowe. Flows that have less than 120 packets within 10 s will be detected as nonvoice flowes by default through the proposed solution.

4. Experimental setup

For the voice detection algorithm and analysis of the captured traces we use a dual core 2.26 GHz processor with 6 GB RAM under the Windows 7 operating system. The test setup captures both online and ofline traces. Online traces are captured by observing the communication between various voice and nonvoice traces for almost 3 h. Ofline traces are collected from 3rd party websites such as Wireshark (https://wiki.wireshark.org/SampleCaptures) and Skype (http://tstat.polito.it/traces-skype.shtml). These traces comprise voice-based application traces, their own voice setup traces, nonvoice traces, and mixed traffic. Voice-based application traces like Skype version 6.1.129, Yahoo! messenger version 11.5.0.228, Gtalk Beta Version, MSN messenger 7.0 and 8.5, ooVoo Version 3.5.8.22, Tencent QQ messenger ver 1.75.2548.0, Trillian IM ver 5.3.0.15, and Team Talk 4 are used for voice communications. A simple voice setup is also developed by using Asterisk as voice server and Eyebeam, Blink, Zfone, and Xlite as voice clients that perform conversation for encrypted and nonencrypted voice using SRTP and ZRTP protocols for testing purposes. Moreover nonvoice traces are collected for testing by using applications like Facebook, BitTorrents, YouTube Session, downloading audio and video, Online Live TV, and gaming. Mixed traffic consists of both voice and nonvoice traffic simultaneously. Table 1 shows the summary of those testing traces used in testing phase.

```
Algorithm:
INPUT: Flow F, Input packet Pi, where 0 < I < N
OUTPUT: Flow F is categorized as voice or nonvoice flow
Step 1: Read Pi                    (Read a new Packet)
Step 2: Test Pi ∈F(r) or Pi ∈ F(ur)   (Test if a packet belongs to a registered F(r) or unregistered flow F(ur)
   (a) If (Pi ∈F(ur)) Then                    (packet received first time, to be registered)
        Register(S-IP, D-IP, S-Port, D-Port)
        Initialize (T(flow), N(pkts), MIN(pkt size), MAX(pkt size),SUM(pkt length), SUM
        (Sqr pkt length))
   (b) If ( T(flow) < 10 OR N(pkts) <120 ) Then
        GoTo Step 1
       Else
        GoTo Step 3     (packet is registered but not detected)
Step 3: Test ( Pi ∈F(voice) OR Pi ∈F(nonvoice) )
   (a) If ( Pi ∈F(voice) OR Pi ∈F(nonvoice) ) Then
        GoTo Step 1
       Else              (Already identified, read the new packet)
        GoTo Step 4                    (Go for flow identification)
Step 4: Calculate T(flow) & N(pkts)                    (for a required time interval)
Step 5: Test ( T(flow)< 10 OR (N(pkts)<120 )
   (a) If (True)
        Update(T(flow), N(pkts), SUM(pkt length), SUM(Sqr pkt length), MIN(pkt size),
        MAX(pkt size))  (Keep updating until total flow time < 10 or total number of packets < 120)
       Else
        Calculate AVG(pkt size), VAR(pkt size), PR, RANGE(pkt size)
   (b) If ( (PR > 12) & (50 < AVG(pkt size) < 250 ) & ( 0 ≤VAR(pkt size) < 100)) Then
        If ( 50 < AVG(pkt size) < 60 ) Then
          If (PR ≤ 200) Then
                F ∈Flow(voice)
          Elsif (PR < 500) Then
                F ∈Flow(voice)
          Else
                F ∈Flow(nonvoice)
          End if
        End if
       End if
End algorithm
```

5. Performance evaluation of the proposed solution

In this work we have applied statistical analysis to both voice and nonvoice flowes to compute parameters such as number of packets, total flowe time, arithmetic mean, standard deviation, packet rate, and minimum and maximum packet size of the captured traces.

As shown in Figure 5a, our three previous observations about a voice flowe are true, i.e. high packet rate and low mean and standard deviation of packet size. Moreover, the range of packet size is also minimum in the

Table 1. Test traces.

#	Application	Type	No. of traces	Max. duration	Max. size	Transport layer
1	Skype version 6.1.129	Voice	11	980	8	UDP
2	Yahoo Messenger 11.5.0.228	Voice	5	332	2	TCP&UDP
3	Gtalk Beta Version	Voice	4	504	3	UDP
4	MSN Messenger 7.0, 8.5	Voice	4	88	0.6	UDP
5	Eyebeam-Asterisk-Blink	Voice	8	478	103	UDP
6	ooVoo Version 3.5.8.22	Voice	1	123	4	UDP
7	Zfone-Asterisk-Xlite	Voice	2	2630	4.5	UDP
8	Tencent QQ 1.75.2548.0	Voice	1	57	1	UDP
9	Trillian IM ver 5.3.0.15	Voice	1	133	1.8	UDP
10	TEAMtalk 4	Voice	1	238		UDP
11	Mixed	Voice & nonvoice	1	1023	4	TCP&UDP
12	Facebook	Nonvoice	2	370	65	TCP&UDP
13	Email server (Hotmail, Yahoo Mail, Gmail)	Nonvoice	2	156	16	TCP&UDP
14	Online game	Nonvoice	1	108	3	TCP&UDP
15	Online live TV (Geo, Duniya)	Nonvoice	2	204	4	TCP&UDP
16	Torrent download	Nonvoice	3	2043	150	TCP&UDP
17	Video clips (YouTube, others)	Nonvoice	4	297	9	TCP&UDP
18	Mixed nonvoice	Nonvoice	2	1331	112	TCP&UDP

	Gtalk(Home)		Gtalk (Lab)		Skype(Home)		Skype (Lab)		Yahoo Messenger (Home)		MSN Messenger(Lab)		Eyebeam-Asterisk-Blink Set-up		MSN Messenger (Home)		Yahoo Messenger (Lab)		Zfone-Asterisk-Xlite Set-up	
Direction	A -> B	B -> A	A -> B	B -> A	A -> B	B -> A	A -> B	B -> A	A ->B	B -> A	A -> B	B -> A	A -> B	B -> A	A -> B	B -> A	A -> B	B -> A	A -> B	B -> A
Transport Protocol	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP	TCP(SSL)	TCP(SSL)	UDP	UDP
No. of Packet	121	121	121	121	121	121	121	1	121	121	121	121	121	121	121	121	121	121	121	121
Total Flow Time (S)	6.369	9.581	6.49	7.48	2.31	2.28	2.36	0	7.68	7.34	2.37	3.01	2.39	2.4	2.77	5.4494	5.39	3.85	9.73	2.33
Mean	167.36	60.24	123.02	85.27	91.98	100.66	125.59	50	83.56	179.39	111.53	115.87	210	210	116.12	115.21	68.81	83.09	205.09	202.28
Standard Deviation	54.52	14.77	68.97	41.86	11.81	26.12	19.59	0	14.91	34.3	11.72	21.74	0	0	14.81	13.654	34.2798	41.42	29.54	35.13
Packet Rate	18.997	12.63	18.63	16.18	52.46	52.96	51.33	1	15.75	16.47	50.95	40.21	50.55	50.41	43.66	22.204	23.759	31.4	12.43	51.75
Minimum Packet Size	53	53	53	53	69	31	92	50	49	49	103	41	210	210	41	41	40	40	120	56
Maximum Packet Size	220	96	244	206	140	154	158	50	90	228	146	146	210	210	145	132	151	400	512	512
Range	167	43	191	153	71	123	66	0	41	179	43	105	0	0	104	91	111	360	392	456

	Facebook		Yahoo Mail & Gmail		Hotmail				Online Live TV		Torrent		Media Session				Torrent (3 sessions)	
Direction	A -> B	B -> A	A -> B	B->A	A -> B	B -> A	C -> D	D -> C	A -> B	B -> A	A -> B	B -> A	A -> B	B -> A	C -> D	D -> C	A -> B	B -> A
Transport Protocol	TCP	TCP	TCP	TCP	TCP	TCP	TCP	TCP	UDP	UDP	TCP	TCP	TCP	TCP	TCP	TCP	TCP	TCP
No. of Packet	121	121	121	121	121	121	121	121	121	69	121	1	121	121	121	121	121	121
Total Flow Time (S)	9.231545	6.55	8.770659	7.504825	6.37	7.46	5.91	7.32	8.37	5.13	9.14	0	2.77	1.64	1.69	2.89	8.18	5.1
Mean	50.876033	1365.5	1435	1368	1323.34	126.93	1334.55	85.63	1322.71	49.16	1327.31	156	52.84	1349.6	1349.59	53.46	71	1304.84
Standard Deviation	61.054314	199.9	287	382	426.87	293.27	410.48	160.68	377.11	5.86	273.82	0	100.67	400.68	400.73	101.99	190.66	455.97
Packet Rate	13.1	18.48	13	16	18.99	16.22	20.49	16.54	14.46	13.45	13.22	0	43.68	73.66	71.75	41.89	14.77	23.68
Minimum Packet Size	40	40	40	40	40	40	40	40	55	40	273	156	40	40	40	40	40	40
Maximum Packet Size	653	1400	1500	1500	1500	1500	1500	1194	1500	60	1400	156	1154	1482	1482	1169	1482	1482
Range	613	1360	1460	1460	1460	1460	1460	1154	1445	20	1127	0	1114	1442	1442	1129	1442	1442

Figure 5. a) Statistical analysis of voice applications. b) Statistical analysis nonvoice applications.

case of voice flowes. Based on this analysis, threshold values of the statistical parameters are fine-tuned for the proposed voice detection algorithm.

Figure 5b shows the statistical parameter values of different flowes on different nonvoice applications. Mainly those applications are considered that have higher packet rates, similar to voice flowe, so that we make a

clear distinction between voice and nonvoice flowes based on other parameters when the packet rate constraints match. There is only one flowe, i.e. torrent flowe from A→B, that may be confused with voice flowes but in such a case we use the range determinant that will differentiate such flowes from voice flowes. Thus, this article describes a thorough investigation of both nonvoice and voice flowe parameter values by using prevalent voice and nonvoice applications. Moreover, it is also observed that the measurements of the values of these evaluated statistical factors for both voice and nonvoice flowes are quite different. Using the empirical analysis carried out in this research and with the above statistics, voice flowes can easily be identified.

The following are the conventional evaluation parameters used to test the proposed system accuracy: 1) true positive (TP) - flows that are correctly identified as voice flows; 2) false positive (FP) - flows that are incorrectly identified as voice flows; 3) false negative (FN) - flows that are incorrectly identified as nonvoice flows; 4) true negative (TN) - flows that are correctly identified as nonvoice flows.

Table 2 presents total voice calls for applications, which is the product of the number of traces multiplied by the number of flowes. For example, the Eyebeam-Asterisk-Blink Voice Setup has a total of 8 traces used in our testing phase multiplied by 2 as there are two flowes involved in a single trace, i.e. A. → gB and B. → gA. The proposed solution truly detects 103 voices traces out of 110 voice traces. Table 3 presents the evaluation results of the proposed solution for nonvoice flowes. As shown, the algorithm correctly identifies 59,466 traces out of 59,471 nonvoice traces. However, 5 nonvoice traces are identified as voice traces. Mixed traces are both voice and nonvoice traces, which are captured simultaneously.

Table 2. Performance evaluation of voice traces.

Application	Total voice calls	Our solution	
		TP	FN
Eyebeam-Asterisk- Blink	16	16	0
Gtalk	8	8	0
MSN Messenger	8	8	0
Yahoo Messenger	10	10	0
Skype	22	19	3
ooVoo	2	1	1
Mix_Voice_Calls	46	44	2
Tencent QQ Messenger	2	2	0
Trillian IM	2	2	0
aZfone-Asterisk-Xlite	8	7	0
TeamTalk	2	2	0
Total	110	103	7

Table 3. Performance evaluation of nonvoice traces.

Application	Total voice calls	TN	FP
Facebook	20,764	220,763	1
Mixed nonvoice	8405	8403	2
Mail servers (Hotmail,YahooMail,Gmail)	253	253	0
Online game	19	19	0
Online live TV (Geo Sports, Duniya)	249	249	0
Torrents	22,847	22,846	1
Video clips (YouTube, others)	1428	1427	1
Total	59,471	59,466	5

Table 4 shows the evaluation results by applying the algorithm in terms of TP, TN, FP, and FN rates. As shown, there are 2 TPs and 5506 TNs, demonstrating good detection estimates. We have added two more parameters to summarize the overall results by the proposed voice detection algorithm. These two parameters are the direct rate (DR) and false alarm rate (FAR) used in [17] to measure the correctness of the system. There should be a minimum FAR value and maximum DR value for an efficient and accurate system. According to [17], the ideal system has 100% DR and 0% FAR. DR provides the percentage of correctly identified voice flowes and is defined as: DR = TP / (TP + FN) & FPR = FP / (FP + TN). Table 5 summarizes the performance results of the proposed voice detection algorithm.

Table 4. Performance evaluation of mixed traces.

Mix traces	Total flows	TP	FN	FP	TN
Mix voice and nonvoice	5508	2	0	0	5506

Table 5. Absolute performance evaluation of the proposed methodology.

Traces	TP	FN	FP	TN	DR	FAR
Offline traces	103	7	5	59,466	93.6%	0.000084%
Own voice setup traces	16	0	-	-	100%	-
Online traffic	19	1	1	4855	95%	0.00020%

6. Conclusion

We have proposed a new voice packet classification and detection strategy over IP networks. In the first step, we perform packet feature analysis on voice, nonvoice, and mixed traces for a number of applications and provide more precise boundaries for packet size distribution and packet rate to separate distinct flowes for voice and nonvoice applications. We then propose a voice detection algorithm to further improve our results. The algorithm is based on statistical analysis of both the voice and nonvoice flowes. It is independent of any protocol, security mechanism, and application. Statistical analysis is performed on the basic parameters to set threshold constraints on captured datasets for IP traffic and is equally applicable for applications making use of security protocols such as TLS, SRTP, ZRTP, or SIPS. Evaluation of the system is based on online and ofline test suites and data captured in different environments. Our proposed technique has 93.6% TP for ofline traces, 100% TP for the self-configured voice setups, and 95% TP for the online traffic. Our research work has a low FAR of 0.000084% for ofline traces and 0.00020% FAR for online traces. Future work includes parallel implementation of the proposed algorithm to achieve high-speed gains for real-time traffic.

References

[1] Renals P, Jacoby GA. Blocking Skype through deep packet inspection. In: 42nd Hawaii International Conference on System Sciences; 5–8 January 2009; Waikoloa, HI, USA. New York, NY, USA: IEEE. pp. 1-5.

[2] Zhou LX, Zhi HJ, Song HM, Peng YF. Identification of P2P streaming traffic using application signatures. Appl Res Comp 2009; 6: 2214-2216.

[3] Dodge RC Jr. Skype fingerprint. In: 41st Hawaii International Conference on System Sciences; 7–10 January 2008, Waikoloa, HI, USA. New York, NY, USA: IEEE. p. 485.

[4] Karagiannis T, Papagiannaki K, Faloutsos M. BLINC: Multilevel traffic classification in the dark. In: 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications; 22–26 August 2005; Philadelphia, PA, USA. pp. 229-240.

[5] Lee S, Kim H, Barman D, Lee S, Kim C, Kwon T, Choi Y. NeTraMark: a network traffic classification benchmark. Comput Commun Rev 2011; 41: 22-30.

[6] Bernaille L, Teixeira R, Salamatian K. Early application identification. In: 2nd Conference on Future Networking Technologies; 4–7 December 2006; New York, NY, USA. pp. 1-6.

[7] Wright CV, Monrose F, Masson GM. On inferring application protocol behaviors in encrypted network traffic. J Mach Learn Res 2006; 7: 2745-2769.

[8] Bernaille L, Teixeira R, Akodkenou I, Soule A, Salamatian K. Traffic classification on the fly. Comput Commun Rev 2006; 36: 23-26.

[9] Alshammari R, Zincir-Heywood AN. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? Comp Netw 2011; 55: 1326-1350.

[10] Yildirim T, Radcliffe P. VoIP traffic classification in IPSec tunnels. In: 2010 International Conference on Electronics and Information Engineering; 1–3 August 2010; Kyoto, Japan. pp. 151-157.

[11] Crotti M, Dusi M, Gringoli F, Salgarelli L. Traffic classification through simple statistical fingerprinting. Comput Commun Rev 2007; 37: 7-16.

[12] Li W, Moore AW. A machine learning approach for efficient traffic classification. In: 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems; 7–9 August 2012; Arlington, VA, USA. pp. 310-317.

[13] Karagiannis T, Broido A, Faloutsos M, Claffy KC. Transport layer identification of P2P traffic. In: 2004 Internet Measurement Conference; 25–27 October 2004; Taormina, Italy. pp. 121-134.

[14] Alshammari R, Zincir-Heywood AN. Machine learning based encrypted traffic classification: Identifying SSH and Skype. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications; 8–10 July 2009; Ottawa, ON, Canada. New York, NY, USA: IEEE. pp. 1-8.

[15] Branch P, But J. Rapid and generalized identification of packetized voice traffic flows . In: 2012 IEEE 38th Conference on Local Computer Networks; 22–25 October 2012; Clearwater Beach, FL, USA. New York, NY, USA: IEEE. pp. 85-92.

[16] Bonfiglio D, Mellia M, Meo M, Rossi D, Tofanelli P. Revealing skype traffic: when randomness plays with you. In: 2007 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications; 27–31 August 2007; New York, NY, USA. pp. 37-48.

[17] Alshammari R, Zincir-Heywood AN. An investigation on the identification of VoIP traffic: case study on Gtalk and Skype. In: 6th International Conference on Network and Service Management; 25–29 October, 2010; Niagara Falls, ON, Canada. pp. 310-313.