On the Impact of Packet Sampling on Skype Traffic Classification

P.M. Santiago del Río, D. Corral, J.L. García-Dorado, J. Aracil
High Performance Computing and Networking
Universidad Autónoma de Madrid, Spain
Email: pedro.santiago@uam.es,diego.corral@estudiante.uam.es,{jl.garcia, javier.aracil}@uam.es

Abstract*—Nowadays, traffic classification technology addresses the exciting challenge of dealing with ever-increasing network speeds, which implies more computational load especially when on-line classification is required, but avoiding to reduce classification accuracy. However, while the research community has proposed mechanisms to reduce load, such as packet sampling, the impact of these mechanisms on traffic classification has been only marginally studied. This paper addresses such study focusing on Skype application given its tremendous popularity and continuous expansion. Skype, unfortunately, is based on a proprietary design, and typically uses encryption mechanisms, making the study of statistical traffic characteristics and the use of Machine Learning techniques the only possible solution. Consequently, we have studied Skypeness, an open-source system that allows detecting Skype at multi-10Gb/s rates applying such statistical principles. We have assessed its performance applying different packet sampling rates and policies concluding that classification accuracy is significantly degraded when packet sampling is applied. Nevertheless, we propose a simple modification in Skypeness that lessens such degradation. This consists in scaling the measured packet interarrivals used to classify according to the sampling rate, which has resulted in a significant gain.

Keywords*—*Skype; Traffic Classification; Packet sampling; High-speed networks.

I. Introduction

Both the research community and network operators have dedicated extensive effort to the development of traffic classification technologies given their relevance in management tasks as important as network design and engineering, security, advertising, or DiffServ mechanisms [1]. Similarly, traffic classification allows analyzing changes in the Internet, understanding the behavior of different applications and the traffic generated by them. Specifically, on-line traffic classification has proven useful for a set of tasks that require taking measurements on-the-fly. Examples of such tasks are intrusion detection, accounting, quality of service (QoS) or quality of experience (QoE) management and lawful-interception.

Nonetheless, the ever-increasing data transmission rates have become traffic classification in an exciting challenge. In multi-10Gb/s networks, very common nowadays, traffic classifiers have to be able to capture and analyze up to several tens of millions of packets per second. In spite of improvements on capture capabilities and efforts to optimize and relieve classification mechanisms of burden [2], to date many network monitoring systems only deal with packet sampling data in an attempt to reduce such burden. That is, traffic classification systems are not provided with all the traffic but only a fraction of the packets are taken into account.

The relationship between traffic classification and packet sampling was first pointed out in [3]. In such work, the monitoring system first sampled at packet level, then generated Netflow records, and finally the records were classified using machine learning (ML) techniques [2] (specifically, decision trees). Note that Netflow data records only comprise information about the source and destination IP addresses, port numbers, protocol and counters of bytes and packets. Similarly, the authors in [4] proposed to use packet-sampled flow records that included a more extensive set of features, e.g., RTT or number of ACKs. Both studies concluded that sampling entails a significant impact on the classification performance, especially, in terms of volume in bytes and packets.

Differently, this paper does not analyze packet-sampled flows but assumes a monitoring system fed with a sample of the total packets traversing the monitored link. The advantages are twofold: the accuracy increases, and it is possible to classify on-the-fly. Note that flow-based classifying requires that flows end before being analyzed. This is unacceptable in VoIP applications where operators have to apply measurements, such as accounting, improve quality or, conversely, blocking if some VoIP applications are not allowed by contract, while the call is in course, and not after its finalization.

Specifically, we turn our interest to Skype classification given its tremendous popularity and continuous expansion between the clients of VoIP [5]. In fact, Skype has also attracted the attention of the research community, which has characterized its behavior [5] and proposed several detection algorithms [6], [7]. In this paper, we have evaluated the impact of sampling on the classification of Skype using *Skypeness* [7] over both synthetic and real traces from public repositories. Skypeness is a commodity off-the-shelf system to Skype traffic detection at multi-10Gb/s rates based on the functionality of Tstat Skype module [8] but with a simpler software implementation to allow its on-line execution.

The results show that Skype detectors are affected by sampling because the statistical characteristics that they are based on, such as interarrival times, are distorted by sampling. However, we propose a simple modification in the detection algorithm to mitigate such effects. Particularly, the observed interarrivals that Skypeness uses to make a decision are scaled according to the sampling rate. With this modification, the results are similar to those with unsampled traffic, although at the expense of a small increment in the false positive ratio. Consequently, this study proves that sampling is not a definitive pitfall to track Skype at multi-10Gb/s.

978-3-901882-50-0 ©2013 IFIP

As an additional contribution of this work, we have made public for the research community as open-source the code of Skypeness, the programs that we have used to sample packets in traces, as well as the Skype traces used as testbed[1].

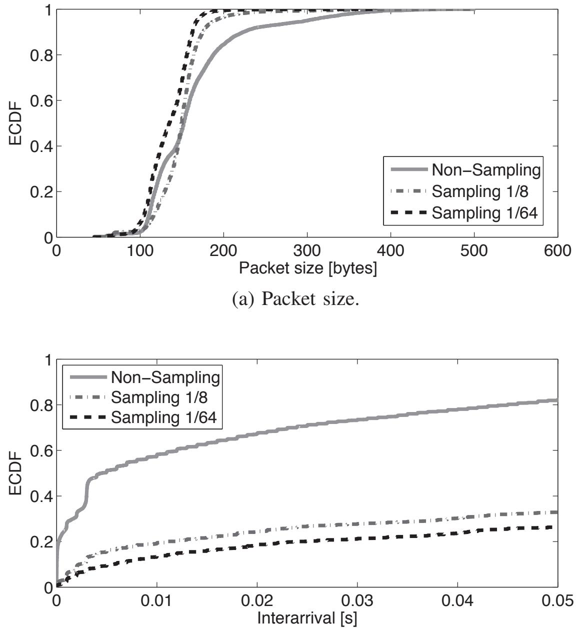
II. Skype traffic classification

Skype traffic, unlike traditional services and protocols, cannot be detected using well-known ports or applying deep packet inspection (DPI) techniques because Skype uses a proprietary, obfuscated and encrypted protocol that employs per-session random ports. The answer of the research community has been the use of statistical traffic characteristics and ML techniques [2].

The authors in [6] presented a Skype traffic detection algorithm based on two statistical techniques: First, they infer a probability distribution of both packet length and inter-arrival time from audio and video codecs used by Skype. Then, it is checked if the empirical distributions of a given flow fit with the hypothesized ones, using a Bayesian classifier. Second, as Skype traffic is encrypted, it is checked if the payload of a given flow follows a uniform distribution, using Pearson's Chi-Square estimator. The algorithm is implemented as a module of Tstat. [8]. However, Tstat documentation explains that the Bayesian classifier configuration requires a fine parameter configuration and significant computation load limiting its applicability to multi-10Gb/s networks.

In this light, we borrowed Tstat's proposals and developed Skypeness [7], a high-performance Skype traffic classifier based on three intrinsic characteristics of Skype traffic, namely: delimited packet size, nearly constant packet interarrival times and bounded bitrate. Specifically, Skypeness computes the mean values of these three features (packet size, interarrival time and bitrate), averaging in windows of 10 packets, for each flow. If the ratio of packet windows whose mean values are inside of a given interval is greater than a given threshold, such flow is marked as Skype. For instance, Fig. 1 shows the appropriate interval and threshold values for audio Skype calls, specifically it shows the empirical cumulative distribution functions for packet size and interarrival time increments from 44 Skype audio calls when no sampling is applied (continuous line). Thus, packet size is well delimited (between 60 and 200 bytes more than 75% of the packets) and more than 60% of the interrarival increments are less than 15 ms. Table I shows all intervals and thresholds corresponding to the different classes of Skype traffic, namely, only audio calls, video (and audio) calls and file transfers. Note that the detector only considers UDP flows that have more than 30 packets (three packet windows). Skype typically uses only UDP as transport-layer because it is more suitable in real-time applications. However, it is uncommon but possible that Skype shifts to TCP in an attempt to evade firewalls or other similar restrictions. As we leverage on packet interarrivals assuming they are fairly constants, and TCP can modify this depending on its configuration, we have focused on UDP traffic.

Although packet size is not affected by packet sampling (Fig. 1a), interarrival time is distorted when sampling is applied (Fig. 1b) and, therefore, the expected interval values are no longer valid. Thus, Skypeness detection accuracy is reduced to nearly zero in presence of packet sampling. This fact will be analyzed in more detail in Section V.



(a) Packet size.

(b) Interarrival time increments.

Fig. 1: Empirical CDF for packet size and interarrival times in audio Skype calls.

TABLE I: Intervals and threshold values used by Skypeness detector.

Media	Characteristic	Interval	Threshold
Audio	Packet size [Bytes]	$[60, 200]$	0.75
	Interarrival [ms]	$[i_{n-1} \pm 15]$	0.6
	Bitrate [Kbps]	$[0, 150]$	0.75
Video	Packet size [Bytes]	$[150, 1200]$	0.19
	Interarrival [ms]	$[i_{n-1} \pm 15]$	0.6
File Transfer	Packet size [Bytes]	$[480, 540] \cup$ $[950, 1050] \cup$ $[1310, 1380]$	0.44

III. Methodology

A. Classification accuracy metrics

In order to measure the detector accuracy, let us define the following metrics:

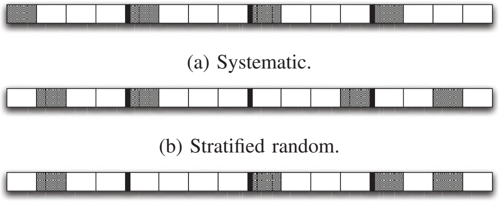
- True (False) positive, TP (FP): amount of Skype (Non-Skype) traffic classified as Skype traffic.
- True (False) negative, TN (FN): amount of Non-Skype (Skype) traffic classified as Non-Skype traffic.

Such metrics can be counted using bytes, packets or flows. The choice of the unit (packets, bytes or flows) depends on the purpose of the classification.

B. Packet sampling policies

Packet sampling techniques allows choosing a fraction of the total amount of packets, following a given criterion to

[1] http://www.eps.uam.es/~psantiago/skypeness.html



(a) Systematic.

(b) Stratified random.

(c) Simple Random.

Fig. 2: Packet Sampling Policies.

reduce the computational burden of any subsequent analysis. Figure 2 shows the three main packet sampling policies [9], namely:

- Systematic: data are split in cycles of n packets and the first element of each cycle is deterministically chosen.
- Stratified random: data are also split in cycles of n packets but one element of each cycle is randomly chosen.
- Simple random: each packet is randomly chosen with a given probability $1/n$.

Sampling techniques can be implemented using mechanisms based on either events or timer [9]. That is, each cycle can be either an amount of packets or a time interval. In our case, the cycle is an amount of packets (equal to the inverse of the sampling rate) due to its better performance.

Other packet sampling policies could be applied, such as window-based sampling (i.e., capturing packets during a given period, then, waiting during another time interval without sniffing, and so forth). However, such approaches require capturing all packets (zero losses) in the active period, which is not often suitable in high-speed capturing context.

IV. Datasets

We have made use of four different traces of UDP traffic, Table II shows an overview of the datasets. The first and second traces, named as Trace 1 and Trace 2 in the following, contain Skype traffic captured on the access link of Politecnico di Torino [10]. The set of users are students, faculty and administration staff. The capture duration is 96 hours in May/June 2006. Trace 1 only contains end-to-end Skype audio and video calls whereas Trace 2 only contains Skype end-to-out calls. Trace 1 and Trace 2 contain 40M and 3M packets respectively. The third trace, named as Trace 3, contains Skype traffic generated in our laboratory at Universidad Autónoma de Madrid in May 2010. The trace contains 700K packets from end-to-end Skype voice (3A) and video (3B) calls, as well as file transfers (3C). The last trace used, named as Trace 4, is a trace generated and captured in our laboratory that contains 5K UDP packets of P2P traffic from several applications, such as eMule and BitTorrent. With this in mind, traces 1, 2 and 3 are useful to estimate accuracy in terms of FN ratio because such traces only contain Skype traffic. TP ratio is estimated with Trace 4 as this trace does not contain Skype traffic.

TABLE II: Datasets.

Trace		Skype	Non-Skype	Skype Media
Trace 1	Bytes	8,381,658,970	0	Audio and Video
	Packets	39,458,562	0	
	Flows	1059	0	
Trace 2	Bytes	231,257,652	0	Audio
	Packets	3,049,148	0	
	Flows	159	0	
Trace 3A	Bytes	30,950,000	0	Audio
	Packets	230,100	0	
	Flows	44	0	
Trace 3B	Bytes	108,700,000	0	Video
	Packets	217,300	0	
	Flows	46	0	
Trace 3C	Bytes	162,800,000	0	File transfer
	Packets	254,300	0	
	Flows	46	0	
Trace 4	Bytes	0	1,098,935	-
	Packets	0	5312	
	Flows	0	52	

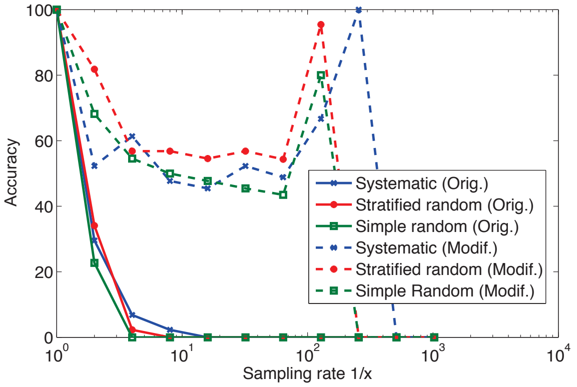


Fig. 3: Skypeness (original and modified versions) accuracy (in bytes) applying different sampling policies and varying sampling rate over Trace 3A (audio calls).

V. Performance Evaluation

To assess the effect of packet sampling on the accuracy of Skypeness detector, we have applied the three sampling policies (see Section III-B), varying the sampling rate between $1/2^0$ (no sampling) and $1/2^{10}$ over the four packet traces. In the following, accuracy in the case of Skype traces means FN ratio, whereas in the case Non-Skype trace means FP ratio.

As an example, Fig. 3 shows the accuracy of Skypeness (continuous line) for trace 3A, while Table III reports the results for all traces (roman fonts). For space constraints, we only show the results for the cases of sampling rates, $s \in \{1/8, 1/64, 1/128\}$. Note that in the case of Trace 4 $s \in \{1/2, 1/4, 1/8\}$, because there is no enough packets when greater sampling rates are applied (recall that we only consider UDP flows with more than 30 packets).

The accuracy suffers a significant cut even when a sampling rate of only $1/8$ is applied for both audio and video traces. This is because mean packet interarrival times do no longer fall inside of the expected intervals assuming unsampled traffic. That is, flows are not identified as Skype calls as packet interarrival time is proportionally incremented with sampling rate, as shown in Fig. 1b. Conversely, in the case of trace 3C

TABLE III: Accuracy (% of bytes) of Skypeness detector original version (roman fonts) and modified version (italic fonts).

Trace	**Non-Sampling**	**Systematic**			**Stratified Random**			**Simple random**		
		1/8	**1/64**	**1/128**	**1/8**	**1/64**	**1/128**	**1/8**	**1/64**	**1/128**
Trace 1	99.59	3.87	0.15	0.04	0.61	0.02	1.23	0.05	11.92	0.17
		90.72	*95.02*	*95.53*	*90.27*	*93.55*	*94.98*	*87.65*	*91.60*	*93.20*
Trace 2	94.22	35.24	0.54	0.00	24.66	0.00	0.00	9.07	0.00	0.00
		75.32	*85.36*	*90.86*	*73.85*	*92.36*	*96.52*	*65.14*	*71.63*	*88.92*
Trace 3A	100	2.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		54.51	*56.20*	*72.19*	*63.02*	*63.99*	*94.75*	*56.35*	*58.30*	*78.49*
Trace 3B	81.38	5.96	0.00	2.68	6.05	1.40	10.40	0.00	0.85	2.76
		84.48	*81.97*	*82.75*	*86.16*	*91.51*	*82.70*	*88.73*	*86.55*	*70.82*
Trace 3C	95.83	96.24	94.76	96.09	96.29	95.29	94.99	95.98	95.64	96.69
	Non-Sampling	**Systematic**			**Stratified Random**			**Simple random**		
		1/2	**1/4**	**1/8**	**1/2**	**1/4**	**1/8**	**1/2**	**1/4**	**1/8**
Trace 4	100	100	100	100	98.87	100	100	100	100	100
		83.00	*95.67*	*100*	*97.19*	*95.53*	*79.92*	*77.26*	*68.04*	*76.85*

(file transfer), packet sampling does not have impact on the accuracy because, in this case, the classifier is only based on packet sizes—and packet size distribution is not affected by packet sampling, as shown in Fig. 1a.

In order to adapt the detector to packet sampling, we multiply the observed interarrival times by the sampling rate, thus reducing their values up to the expected intervals when no sampling is applied. Table III shows the accuracy obtained by such modified version of Skypeness detector (italic fonts). The detector is able to correctly classify, applying systematic or stratified sampling over the Trace 1 (the best case), more than 90% of the traffic regardless the sampling rate. Note that this implies that the detector is able to classify with only 1 out of 128 packets, indeed the results show that the detector after the modification is practically insensitive to the sampling rate. The rest of the traces show also significant accuracy (but the Trace 3A), such accuracy ranges between 73% and more than 95%. In the case of Trace 3A, its accuracy ranges between 54% and 95%, we are investigating on the reasons of this behavior. In Fig. 3, it is shown the accuracy of such trace in dashed lines. Note that there is a spike in the accuracy when sampling rate is greater than 1/100. This fact may be due to that high sampling rates reduce the number of seen flows removing the more unstable (and difficult to identify) ones. Finally, we observe that the false positive ratio, shown in Trace 4, presents also good results, that is, only a moderate increase.

VI. Conclusion

We have empirically studied the impact of packet sampling on the open-source Skype traffic detector *Skypeness*, which is based on three statistical features of Skype traffic: delimited packet sizes, nearly constant interarrival times and bounded bitrates. We analyze the effect on the detector accuracy of two packet sampling factors, namely: the sampling rate and the sampling policy.

Accuracy dramatically decreases when packet sampling is applied, even with the smallest sampling rates ($1/8$) due to distortion on the observed interarrival times. We have proposed a simple modification in the detector (to multiply the observed interarrivals by the sampling rate), which lessens the accuracy reduction, at the expense of a moderated increment on the false positive ratio. Thus, this work shows that sampling is not a definitive drawback to identify Skype at multi-10Gb/s rates. Particularly, Skypeness would be able to detect Skype traffic at more than 300 Gb/s with notable accuracy, given a sampling rate of $1/8$ [7].

References

[1] A. Dainotti, A. Pescapè, and K. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, 2012.

[2] T.T.T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surv. Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[3] V. Carela-Español, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Sol-Pareta, "Analysis of the impact of sampling on Netflow traffic classification," *Computer Networks*, vol. 55, no. 5, pp. 1083–1099, 2011.

[4] D. Tammaro, S. Valenti, D. Rossi, and A. Pescapè, "Exploiting packet-sampling measurements for traffic characterization and classification," *Int. J. Netw. Manag.*, vol. 22, no. 6, pp. 451–476, 2012.

[5] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed analysis of Skype traffic," *IEEE Trans. Multimed.*, vol. 11, no. 1, pp. 117–127, 2009.

[6] 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, 2007.

[7] P.M. Santiago del Río, J. Ramos, J.L. García-Dorado, J. Aracil, A. Cuadra-Sánchez, and M. Cutanda-Rodríguez, "On the processing time for detection of Skype traffic," in *Wireless Communications and Mobile Computing Conference*, Istanbul, Turkey, July 2011.

[8] A. Finamore, M. Mellia, M. Meo, M.M. Munafò, and D. Rossi, "Experiences of Internet traffic monitoring with Tstat," *IEEE Network*, vol. 25, no. 3, pp. 8 –14, 2011.

[9] K.C. Claffy, G.C. Polyzos, and H.-W. Braun, "Application of sampling methodologies to network traffic characterization," *SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, pp. 194–203, 1993.

[10] Telecommunication Networks Group Politecnico di Torino, "Skype traces:," http://tstat.tlc.polito.it/traces-skype.shtml.