Knowledge-Based Systems 82 (2015) 152–162

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Robust application identification methods for P2P and VoIP traffic classification in backbone networks

Tao Qin [a,*], Lei Wang [a], Zhaoli Liu [a], Xiaohong Guan [a,b]

[a] MOE KLINNS Lab, Xi'an Jiaotong University, Xi'an 710049, China
[b] Department of Automation and TNLIST Lab, Tsinghua University, Beijing, China

ARTICLE INFO

Article history:
Received 6 November 2014
Received in revised form 16 January 2015
Accepted 1 March 2015
Available online 6 March 2015

Keywords:
Backbone network
Application identification
P2P and VoIP
Packet Size Distribution
Robustness
Sampling method

ABSTRACT

Application identification plays an essential role in network management such as intrusion detection and security monitoring. But the continuous growth of bandwidth and massive amount of packets pose serious challenges for efficacious and accurate application identification. In this paper, we develop a new method to reduce the number of packets being processed while achieving the goal of accurate P2P and VoIP application identification. Firstly, we employ the Bi-flow model to aggregate traffic packets into Bi-flow, which can capture the exchange behavior characteristics between different terminals. Then we employ the signature of Packet Size Distribution (*PSD*) to capture flow dynamics, which is defined as the payload length distribution probability of the packets in one Bi-flow. Secondly, we collect *PSD* of several different P2P and VoIP applications and the analysis results show that *PSD* of different applications are different with each other, which can be used as features to perform traffic identification. We also find the *PSD* characteristics of one Bi-flow can be captured by its first few packets, which demonstrate our methods can identify the Bi-flow quickly after its establishment. We employ the Renyi cross entropy to perform identification by calculating the similarity between *PSD* of the Bi-flow being identified and that of specific application. If the similarity is higher than a selected threshold, the Bi-flow being identified is classified to the specific application. Finally, as the *PSD* is a type of probability feature which is not sensitive to packet lose, we integrate the Poisson sampling method into our framework to process the massive data in backbone networks. Experimental results using the artificial and actual traces collected from monitoring platform in the Northwest Center of CERNET (China Education and Research Network) verify the accuracy and robustness of our method.

1. Introduction

Application identification is one of the most important parts of network management. Accurate and efficacious application classification is the key stone of network monitoring, and on the basis of the classification results network administrators can design various policies to enhance the network security. However, it is a challenging task to classify the applications based on the traffic characteristics due to the massive data in high-speed networks. The traditional identification methods based on port numbers [1] have been proved useless for applications in network today, because of the dynamic ports policy adopted in P2P applications. The DPI (Deep Packet Inspection) methods are employed to perform application identification based on the signature extract from the payload. As the signatures extracted from the payload are different with different applications, the identification accuracy is very high. The main research works on DPI recently are focused on how to process the massive data by adopting different methods [2–5]. However, more and more applications, such as Skype and QQ, adopted cryptography and the DPI methods will lose its efficiency in this situation. To solve this new challenge, the methods based on flow statistical characteristics attract many attentions [6–8]. But this kind of methods require accurate capturing process without packet lose and the flow features (e.g. packet interval) used are sensitive to network environment, thus those methods cannot be deployed in different networks [9–13]. Furthermore accurate packet capturing in backbone networks is still an open question.

Focus on those problems, in this paper we develop a new robust method for P2P and VoIP application identification by analyzing the characteristics of *PSD*, which is defined as the payload length distribution probability of the packets in one Bi-flow and has been proved to be an effective feature for traffic classification in [14]. As

* Corresponding author. Fax: +86 82664603.
E-mail address: tqin@sei.xjtu.edu.cn (T. Qin).

http://dx.doi.org/10.1016/j.knosys.2015.03.002
0950-7051/

PSD is a type of probability feature, focusing on *PSD* can not only increase the robustness of the identification methods, but also reduce the number of packets being processed by combining sampling methods. The goal of this paper is to design traffic classification methods which are robust and efficacious enough to process massive data in backbone networks. To capture the application dynamics by analyzing its traffic packets, we employ the Bi-flow model to aggregate packets with same source and destination addresses into one Bi-flow, including the forward and backward packets. The TCP Bi-flows are terminated after the packet including FIN/RST is captured or after timeout, conservatively to be 30 min, while UDP Bi-flows are terminated only after timeout. Based on the Bi-flow model developed, we extract the *PSD* of the Bi-flow as the feature to capture flow dynamics, which is defined as the payload length distribution probability of those packets in the Bi-flow. We collected many Bi-flows of different popular P2P and VoIP applications and extract their *PSD*s based on the definition, we find *PSD* of different applications have different statistical characteristics, thus we can employ them as features to perform traffic identification. We also find that the *PSD* characteristics of specific Bi-flow could be captured using first few packets of the entire Bi-flow, which shows that the method developed can identify the Bi-flow quickly after its establishment.

After obtaining *PSD* of the applications selected, we can perform identification by measuring the similarity between the *PSD* of specific application and that of the Bi-flow being identified. If the Bi-flow is generated by Skype, its *PSD* would be much more similar with that of the Skype application, and we employ the Renyi cross entropy method to calculate the similarity. As the *PSD* is a type of probability feature, packet lost during the capture process will not cause obvious changes to the feature. Thus we integrate the Poisson sampling method into our framework to process the massive data in high-speed networks and develop an online system for network monitoring. Using suitable parameter selected, several experiments based on traffic trace collected from the Northwest Region Center of CERNET demonstrate the efficiency and accuracy of our methods. Results of online performance evaluation also verify the robustness of our methods.

Our contributions of this paper can be summarized as follows: Firstly, since the *PSD* is a kind of probability features, which are not sensitive to packet lose during capture process, our method is robust and can be easily deployed in different network environment. Secondly, our method can not only identify the popular P2P and VoIP traffic, but also give the detailed application type information, such as Skype. Finally, as we only use the first few packets of an entire Bi-flow to extract the *PSD*, the proposed methods can identify the Bi-flow quickly after the connection establishment. Moreover, we integrated the Poisson sampling method into our framework, which can greatly reduce the difficulty and computation complexity of online traffic identification in backbone networks.

The remainder of this paper is organized as follows: Section 2 briefly presents the related works, and Section 3 gives the framework of the developed methods. In Section 4, we present the measurement results on the *PSD*. Experimental results using artificial and actual traces are presented in Sections 5 and 6. Conclusion and feature works then follow.

2. Related work

Traffic classification has attracted many attentions in the past decade, researchers do lots of work on popular application identifications and many interesting results have been obtained.

Since early popular applications use default network ports, traffic identification and analysis methods based on network ports are presented in [1]. The port based method is simple and effective, but it fails to identify the current P2P applications using dynamic port assignment. The protocol characteristics such as SIP and RTP are also employed to identify SIP and RTP based VoIP traffic [15,16]. But those kinds of methods are limited by the protocol characteristics obtained, since it is a hard job for getting characteristics of private protocols. To overcome this difficulty, authors have conducted many researches on the payload characteristics and this kind of methods usually named as DPI (Deep Packet Inspection) [17]. Authors developed a method to identify the P2P traffic based on the signature of payload, and five major P2P applications are employed to verify the efficiency of their method. Many methods based on DPI focus on how to develop quickly identification methods [2–5]. However, DPI based methods cannot deal with applications adopted cryptography, which is widely used in the network today.

Another new kind of methods is based on the patterns of host behavior at the transport layer [18]. They pay attention to all the flows generated by a specific host, and can accurately associate each host with the services it provides or uses (such as application server and web client). Authors in [19] investigated some fundamental characteristics of P2P applications, such as the huge network diameter and the presence of many hosts acting as both servers and clients, to identify the P2P traffic. However, this method cannot identify a single flow and is time-consuming, since it must gather information from several flows for each host before it can decide the role of a host. To identify application using cryptography, many researchers adopted the statistical properties of traffic to classify flows. Some works classify network traffic based on summarized flow information such as packet length, flow duration, number of packets, packet length, packet size and mean packet inter-arrival time [6–13]. Usually this type of pattern recognition methods need a training process with labeled samples, in many cases due to their sophisticated algorithms, they hardly could be used for real time identification [20,21].

In order to classify traffic in real time, some clustering and machine learning methods based on the first few packets of an entire flow have been presented [22–24]. In [23] the authors apply the One-Against-All Approach (OAA) for two online classification strategies based on statistical features of TCP sub-flows. In [24] the author compare four identification methods and develop a methods based on few packets. Although the proposed algorithms can be implemented with a part of the traffic flow and deployed easily, they cannot identify the detailed application type information. Actually multiple applications may coexist at the same host simultaneously and these algorithms are difficult to achieve a satisfied identification accuracy. Some researchers try to classify packets only use the first N packets or first M bytes of each flow. Authors [25] used the sizes of the first five packets in each TCP flow to classifying traffic. This method opens a new avenue for on-line traffic classification. But the first N packets and first M bytes may have significant differences for the same P2P application, since some peers may have to re-send some packets due to unreliable network connection and the packets in same position of the corresponding flows may also be different. Therefore the positions of packets and bytes are not reliable information sources to classify a flow and the methods will fail in this case. Methods using the signatures of key packets are proposed in [26], the authors only employ the payload length of the first few control packets to achieve fast P2P traffic identification. But this method is sensitive to the packet captured efficiency. If one of the control packets is not captured and the method will generate a negative result.

Enlightened by prior works and focus our attention on reducing the number of packets being processed and developing robust

identification methods, in this paper we developed a new framework to identify applications based on the characteristics of *PSD*. The effectiveness and efficiency of our method are demonstrated using several actual traffic traces collected from the Northwest Region Center of CERNET.

3. Framework of the developed methods

The framework of the proposed methods is described in Fig. 1, which can be divided into multi-steps as follows:

Step 1: Traffic collection. There are two kinds of traces used for performance evaluation in this paper. The first kind is the traces only content the packets of specific applications, and this kind of traffic is generated manually. Another kind of traces is actual traffic collected from the ingress router of the Northwest Centre of CERNET with bandwidth of 10 Gbps.
Step 2: Poisson sampling. To process the massive data in backbone networks and reduce the computation complexity of identification module, the traffic is sampled using suitable sample ratio.
Step 3: Bi-flow establishment and features extraction. After sampling the traffic packets are aggregated based on the Bi-flow model and *PSD* is extracted.
Step 4: Application identification using Renyi cross entropy. The entropy is employed to calculate the similarity between *PSD* of the being identified Bi-flow and that of the specific application. If the similarity is higher than a threshold, the Bi-flow can be classified as the corresponding application.

3.1. Bi-flow model

To perform traffic identification, we need flow model to extract the features which can reflect the flow dynamics. The NetFlow model is widely used for traffic monitoring in the past several years. But it cannot reflect the dynamics of exchange behaviors between end terminals, which is more important for traffic identification.

In this paper, we employ the Bi-flow model, which is defined as the set of traffic packets with the same source and destination IP addresses, including the forward packets and backward packets. The Bi-flow lifetime is decided by the following policies: TCP flows are terminated after the packet including FIN/RST is captured or after timeout, conservatively to be 30 min, while UDP flows are terminated only after timeout. The definition of Bi-flow model is shown in Fig. 2, which has been proved efficiency in user's behavior monitoring in our prior work [35]. In this paper, we employ the Bi-flow model to extract the traffic dynamics for application identification. Compared with the NetFlow model, the Bi-flow model can greatly reduce the number of traffic flow records while reflecting the users' behavior dynamics.

3.2. Packet Size Distribution

Most previous work uses the packet length as one of the signatures to classify the traffic, but we find it is affected by trailer and option field of TCP header. The trailer is actually padding within the Ethernet frame to satisfy request for minimum frame size. The options may occupy space at the end of the TCP header and are a multiple of 8 bits in length. The lengths of trailer and options are variable according to current network configuration. Therefore, we choose the payload length as packet size, which is a more appropriate attribute than packet length.

The *PSD* of each Bi-flow can be expressed in Eq. (1), where the variable x refers to the packet size with value in the range of $[0,n]$, $p=(p_0,p_1,\ldots,p_k,\ldots,p_n)$ is the probability distribution of the variable x, where p_k is the probability of the variable x with value k, and n is the maximum value of the packet size appeared in the flow, which is different according to applications type. But for each application, we can select an appropriate value based on our experimental experiences.

$$\begin{pmatrix} x \\ p \end{pmatrix} = \begin{pmatrix} 0 & 1 & \ldots & k & \ldots & n \\ p_0 & p_1 & \ldots & p_k & \ldots & p_n \end{pmatrix} \tag{1}$$

3.3. Robust application identification methods

In order to further identify the type of applications rather than only distinguish the traffic, we employ the Renyi cross entropy to calculate the similarity of *PSD* of being identified flow with the *PSD* samples and identify the detailed application type. The Renyi cross entropy has been introduced to detect abnormal traffic in [27] and shows high performance in similarity measurement. The Renyi entropy measures a proper probability distribution of order α using Eq. (2):

$$H_\alpha(p) = \frac{1}{1-\alpha}\log_2\sum_r p_r^\alpha \tag{2}$$

where $0<\alpha<1$ and p is a discrete stochastic variable, p_k is the distribution function. The Shannon entropy measure is a special case of Renyi entropy for $\alpha\to 1$. From Eq. (2)we can get that the Renyi cross entropy of order α is:

$$I_\alpha(p,q) = \frac{1}{1-\alpha}\log_2\sum_r \frac{p_r^\alpha}{q_r^{\alpha-1}} \tag{3}$$

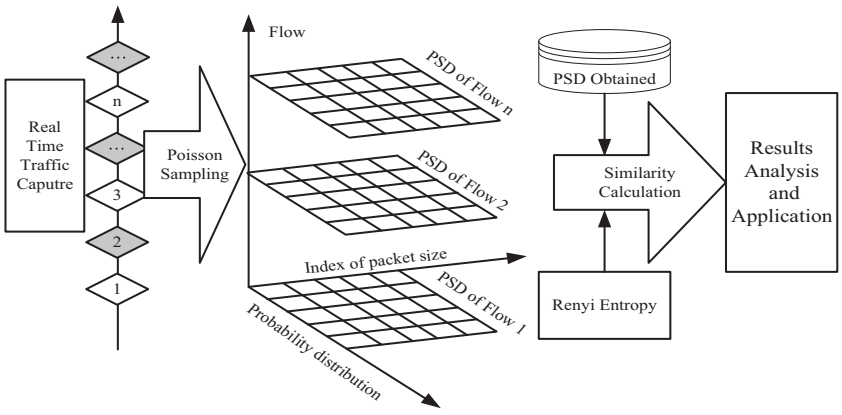


Fig. 1. Framework of the proposed method.

where p and q have the same distributions, p_r and q_r are their distribution functions. The K–L distance is a special case of Renyi cross entropy for $\alpha \to 1$. One important property of the Renyi cross entropy is that if p and q are the same variables, then $I_\alpha \to 0$. If we choose $\alpha = 0.5$ in Eq. (2), then the entropy measure in Eq. (3) is symmetric, which means that $I_\alpha(p,q) = I_\alpha(q,p)$. This symmetric measurement method is suitable for traffic identification as the similarity we want to measure is symmetric. Chose $\alpha = 0.5$, the Renyi cross entropy can be rewritten into Eq. (4).

$$I_{0.5}(p,q) = 2\log_2 \sum_r \sqrt{p_r q_r} \tag{4}$$

Method based on the Renyi cross entropy to identify a specific flow can be expressed as follows:

$$\begin{cases} R = I_{0.5}(p,q) = 2\log_2 \sum_{k=0}^{N} \sqrt{p_k q_k} \\ p = (p_0, p_1, \ldots, p_k, \ldots, p_n) \\ q = (q_0, q_1, \ldots, q_k, \ldots, q_m) \\ N = \min\{m,n\} \end{cases} \tag{5}$$

where p is one of the *PSD* samples in the database while q is the *PSD* of the flow being identified. It is clear that if p and q have the same or similar distributions, the Renyi cross entropy R will be equal or close to 0. On the contrary, the larger the difference between p and q is, the farther R is from 0. Therefore we can select a suitable threshold η, and identify the application type by measuring whether $|R| \leqslant \eta$.

3.4. Integrated with Poisson sampling methods

Sampling method is one of the efficacious ways to process massive data, which attracts many researchers' attention and some interesting results are obtained [28–31]. Poisson sampling is one of the simple but efficacious sampling methods, as the sample intervals are independent, all packets processed would have equal chance of selection [32]. Another advantage of the Poisson sampling is that the duration length ratio of stages is captured in the ratio of number of observations taken during the stages, allowing proportional sampling of the different stages of the observed stochastic process. This can solve the changes of packet arrival process characteristics due to some factors such as time of day and day of the week. The random feature of sampling interval is generated by the negative exponential distribution with parameter λ, which can be obtained from Eq. (6).

$$F(x) = \begin{cases} \int_{-\infty}^{x} f(y)dy = 1 - e^{-\lambda x} & x \geqslant 0 \\ 0 & x < 0 \end{cases} \tag{6}$$

If x follows uniform distribution on $[0,1]$, we can obtain the sampling interval function in (7):

$$G(X_i) = -\frac{1}{\lambda}\ln(X_i) \tag{7}$$

In Eq. (7), $G_i(X_i)$ follows negative exponential distribution with λ and the sampling interval generated using equation (7) follows the negative exponential distribution, in other words, it is Poisson sampling. The process for Poisson sampling can be implemented using the following step shown in Fig. 3:

Step 1: λ Selection. From the PDF of the negative exponential distribution we can obtain that the mean value of the $F(x)$ is:

$$\mu = 1/\lambda \tag{8}$$

where μ is the mean sampling interval, if the value of λ is bigger, the mean will become smaller and the sampling accurate will be higher. For actual networks, the value of λ should be selected according to the network environment.

Step 2: Generation of X_i. Generate the uniform distribution on $[0,1]$ by employing the rand () of library function.

Step 3: Calculate the sampling interval G_i using Eq. (7).

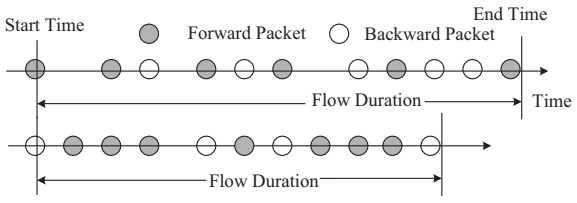


Fig. 2. Bi-flow model.

4. Measurement on the Packet Size Distribution

4.1. Artificial traffic collection

Firstly, we collect one trace with labeled flows with hardly manual efforts. That is, we open only one particular application, disable all other network applications in a host and mark the application flows out of all traffic flows. The flows of each application are collected at different time periods in five different days. Also the collection points are different from Lab, dormitory and company. We selected eight popular applications and for each application and we sample 100 flows each day with a total number of 500. As we label each flow's application, it is accurate and can be used as the benchmark to evaluate the performance of the developed framework. The detailed information of the collected traces is listed in Table 1. In the table, the first four applications are popular P2P applications. PPTV and Qvod are online videos while PT (Private Tracker) and Thunder are applications for file downloading. All of those applications use dynamic port assignment with encryption payload. The traditional methods have been proved

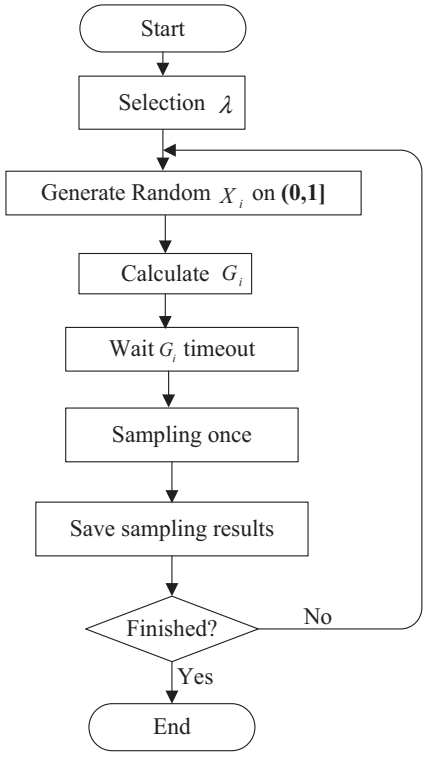


Fig. 3. Process of Poisson sampling.

Table 1
Description of traces of different applications.

No.	Application	# of flow	Lasting time (h)
1	PPTV	500	4
2	Qvod	500	4
3	PT	500	4
4	Thunder	500	4
5	Skype	500	4.5
6	MSNtalk	500	4.5
7	Gtalk	500	4.5
8	QQTalk	500	4.5
9	Others	12,000	5

useless for those applications. The lower part is four popular applications selected from VoIP, which is also using encryption and cannot be identified using DPI easily. We also collect 12,000 flows of other applications, including HTTP, FTP, etc., and mixed them together as artificial trace for performance evaluation.

4.2. Simple measurement on the feature of packet size

Based on the trace collected, we analyze the characteristics of packet size of each application and the analysis results are shown in Fig. 4. In the figure, the red[1] dot is the packet size of each packet in one selected flow and the blue dot is that of the average value of the entire flow. From the results we can get that the packet size of the VoIP applications usually have small packet size due to they aim to delivery real time voice clearly. While packet size of the P2P applications is much bigger, as they are mainly used for file downloading or online videos. For the VoIP applications, the packet intervals are very small and usually there are thousands of packets per minute. While the number of packets in one Bi-flow of P2P application is also huge, but the packet intervals may be larger than VoIP. Furthermore, from the results we can get that the statistical packet size characteristics of different services are different, even different applications of the same VoIP services.

4.3. Analysis on the Packet Size Distribution

We extract the *PSDs* of different applications and the results of their characteristics are shown in Fig. 5. In the figure, the top row is the results of VoIP and low of P2P. As the figure shows, the *PSD* of each application are different with each other obviously. As for the VoIP applications, there are about 60% of the total packets of MSN are centralized around 90 Bytes while the distribution of Skype is more decentralized. Those of the P2P applications are mainly differed by the maximum of the packet size and the distribution of the small packets. Those differences can be characterized obviously using the cumulative distribution functions as shown in Fig. 6. All the analysis results show that the *PSD* can be employed as features for traffic identification.

4.4. PSD generation using first few packets

For real-time application identification and traffic control, how to identify the flow at the early age of the connection establishment is one of the most important problems. For example, one company want to prohibit their employers use P2P applications, if we can identify the application quickly after the connection establishment, the connection can be closed quickly. To verify the *PSD* is a suitable feature for traffic identification, we analyze whether the *PSD* of the first few packets in an entire flow can capture the characteristics of that of the whole flow. We give the

[1] For interpretation of color in Fig. 4, the reader is referred to the web version of this article.

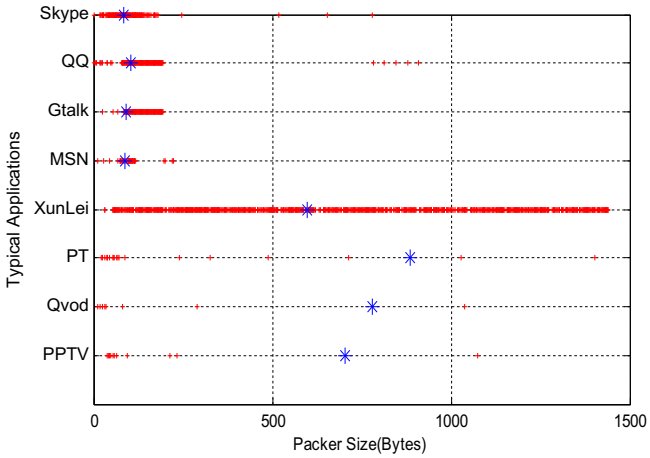


Fig. 4. Packet size of different applications (one Bi-flow selected randomly).

packet size in one Bi-flow sequentially and the results are shown in Fig. 7. As the figure shows, different packet sizes basically distribute uniformly in the whole flow, which verify *PSD* of the whole flow can be captured by the first few packets of the flow, in turn, the *PSD* can be employed for early application identification. The number of first few packets used for identification can be selected by experimental results.

5. Performance evaluation using artificial traces

5.1. Selection of the number of first few packets

We want to select a suitable number of first few packets to construct a sub-flow whose *PSD* is similar with that of the entire flow. We employ the Renyi cross entropy to measure the *PSD* similarity between the *PSD* of the first few packets and that of the entire flow, the analysis results are shown in Fig. 8. In the figures, the *X*-axis is the length of the sub-flows with different number of first few packets selected from the beginning of the flow and the *Y*-axis is the absolute value of Renyi cross entropy. The smaller the Renyi cross entropy is the two *PSDs* are more similar with each other. As the analysis results shows, the number of first few packets used for identification different applications should be different. As the P2P and VoIP flows usually carry thousands of packets, to obtain more stable *PSD* and achieve the goal for real-time identification we select the number of packets as 2000 for all the applications.

5.2. Selection of the threshold of Renyi cross entropy

We employ the false negative ratio (*FNR*) and false positive ratio (*FPR*) of different thresholds η to select suitable η in order to obtain lower *FNR* and *FPR*. The *FNR* and *FPR* are widely used in academic for performance evaluation, which is defined in Eqs. (6) and (7), respectively. Where N_{flow} is the number of the flow being identified, N_{detected} is the total number of flows identified using the methods in this paper, N_{Ture} is the total number of flows correctly identified. *FPR* and *FNR* are smaller when the testing methods have better performance. The analysis results are shown in Fig. 9, in the figures the *X*-axis is the value of Renyi cross entropy and the *Y*-axis is the values of *FNR* and *FPR*. Based on the results we select η equal to 1 for better identification results for all the selected applications.

$$FNR = \frac{N_{\text{detected}} - N_{\text{true}}}{N_{\text{detected}}} \tag{9}$$

$$FPR = \frac{N_{\text{Flow}} - N_{\text{Ture}}}{N_{\text{detected}}} \tag{10}$$

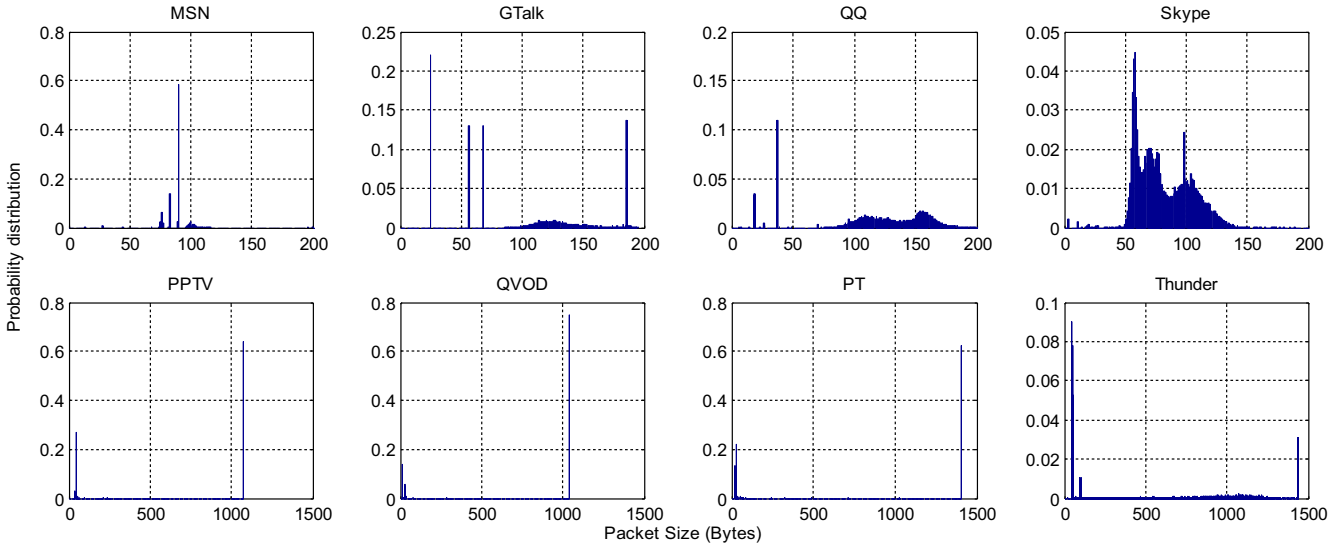


Fig. 5. Packet Size Distribution of different applications (one flow selected randomly).

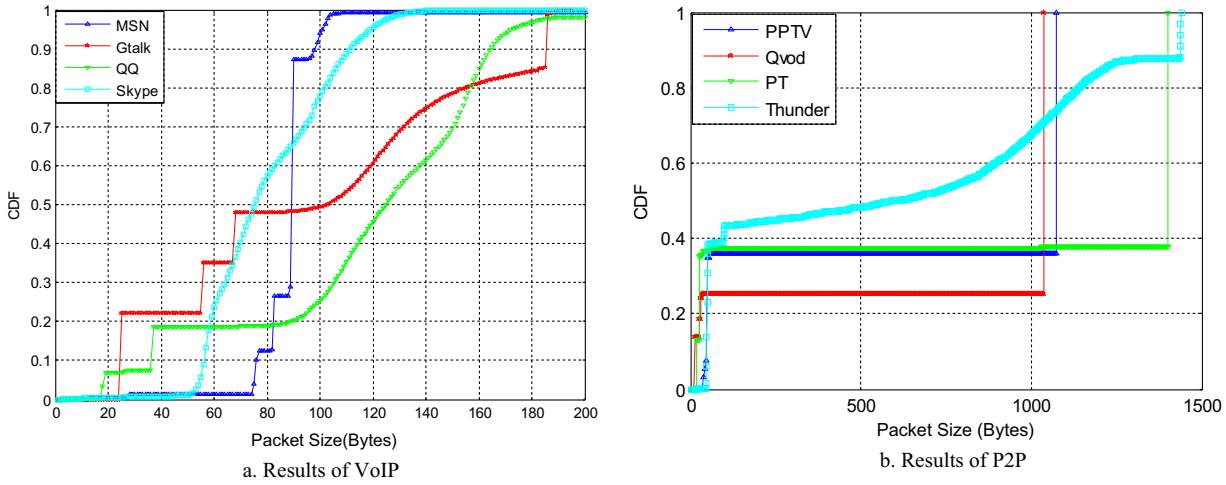


Fig. 6. The CDF of packet size for different applications (one flow selected randomly).

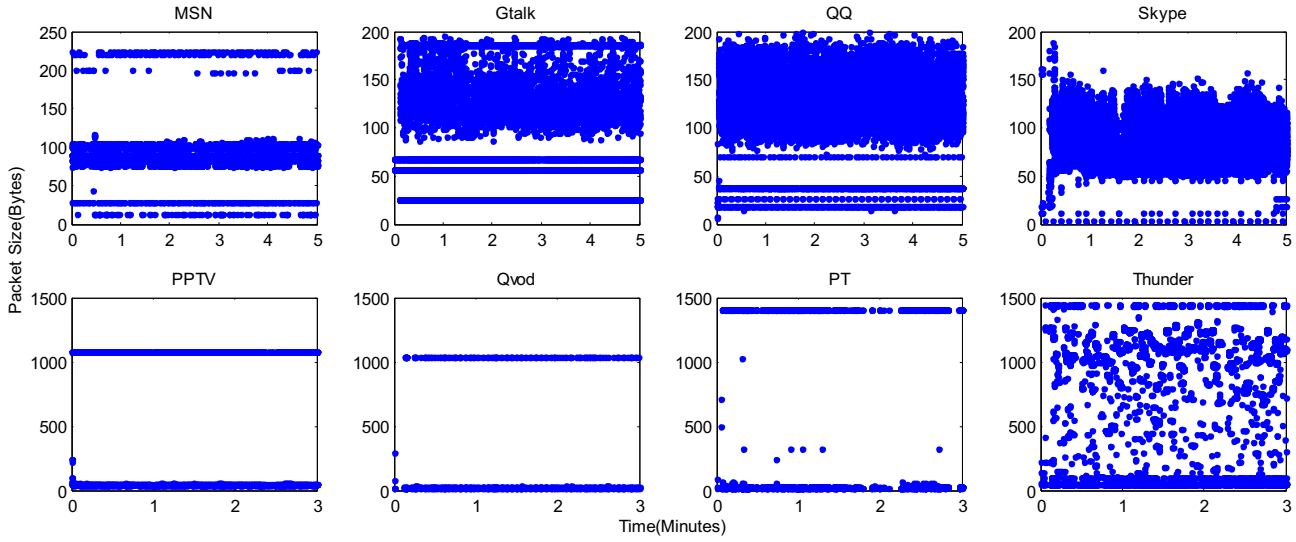


Fig. 7. Sequential of the packet size of different applications (one flow selected randomly).

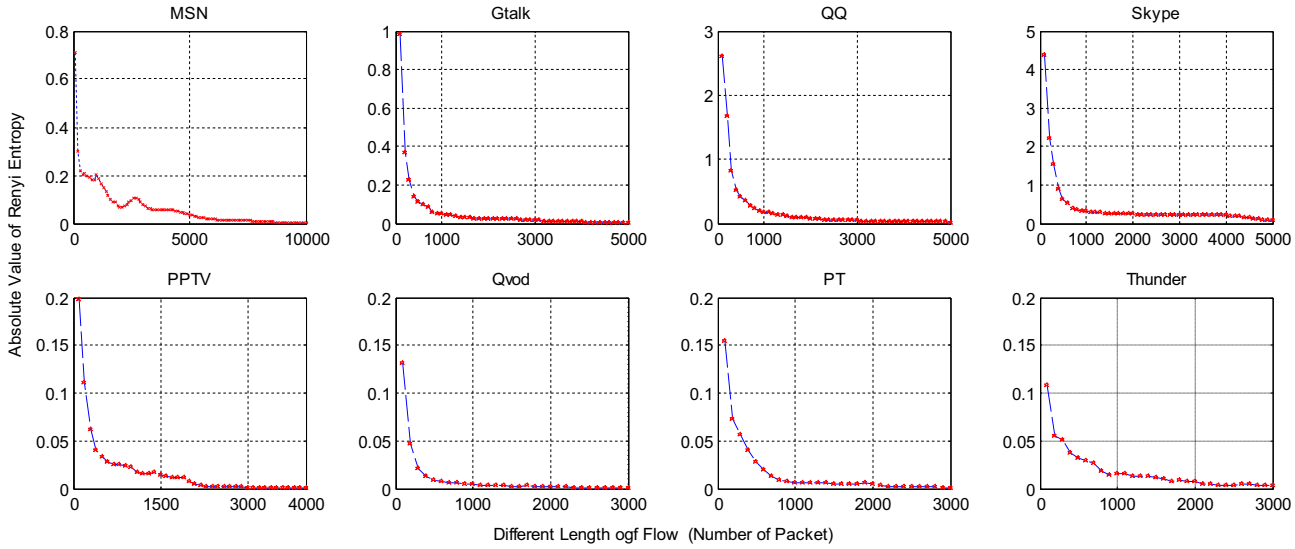


Fig. 8. Selection of parameter N based on the similarity of *PSD* (one Bi-flow selected randomly).

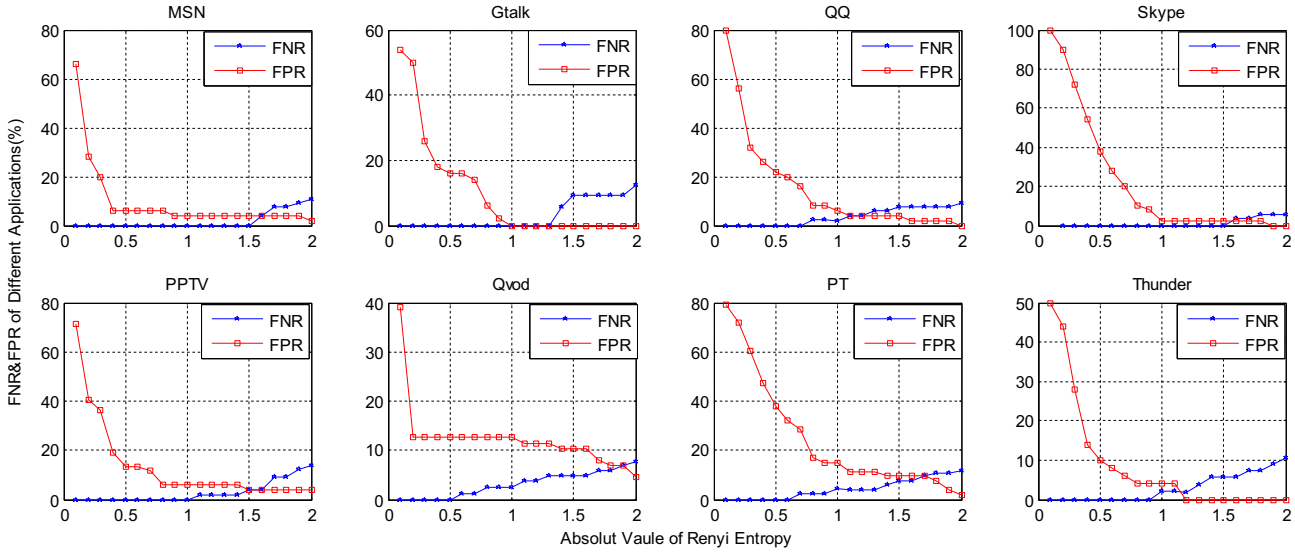


Fig. 9. Influences of parameter η on the *FNR* and *FPR* (one Bi-flow selected randomly).

5.3. Performance evaluation using artificial trace

Based on the parameters selected above, we employ the artificial trace as benchmark to evaluate the identification performance based on *PSD*. In addition, we select the HBC-FLS algorithm proposed in [12] and SKP based methods proposed in [26] for comparison with our methods. The HBC-FLS methods are focus on identify the VoIP traffic while the SKP based methods are focus on P2P flow identification. The parameters required by the HBC-FLS algorithm have been assigned according to [12] and that of SKP based methods are selected according to [26]. Both of them are listed in Table 2. The experimental and empirical parameters for our methods are selected based on analysis above: the number of the first few packets N is selected as 2000 and the threshold η is selected as 1.

All of the three methods are tested using the Bi-flows shown in Table 1, while our methods are tested using the *PSD* extracted from the 2000 packets from the head of the flows. The HBC-FLS methods are tested using the information extracted from the entire Bi-flow, the SKP based methods are tested with the key packets information extracted from 30 packets in front of the flow, while the information of the control packets are extracted using the methods proposed in [26]. The false negative ratio (*FNR*) and false positive ratio (*FPR*) are employed to evaluate the performance. The evaluation results are shown in Table 3. Where INF is the number of identified flows using our methods and CINF is the number of correctly identified flows. It shows that our scheme can identify the applications accurately based on the *PSD* of the first few packets in an entire flow. Both the *FPR* and *FNR* of our methods are small and basically equal to zero.

Table 2
Parameters of compared algorithms.

HBC-FLS methods						SKP based methods		
a	α	β	γ	δ	n	N	θ_T	T
0.9	0.3	0.8	0.45	5	20	30	80%	180 s

Table 3
Experimental results on typical applications.

App	Flow	Our method (%)				HBC-FLS (%)				SKP based method (%)			
		INF	CINF	*FPR*	*FNR*	INF	CINF	*FPR*	*FNR*	INF	CINF	*FPR*	*FNR*
Skype	500	496	492	0.81	1.61	487	426	12.52	15.19	463	401	21.38	13.39
Gtalk	500	502	496	1.19	0.79	496	443	10.68	11.49	427	398	23.88	6.79
QQ	500	497	488	1.81	2.41	498	470	5.62	6.02	418	345	37.08	17.46
MSN	500	491	486	1.01	2.85	513	453	11.69	9.16	526	423	14.63	19.58
PPTV	500	509	492	1.37	1.57	502	476	5.17	4.78	483	461	8.07	4.55
Qvod	500	500	500	0.00	0.00	506	492	1.58	1.58	491	424	15.47	13.64
PT	500	496	490	1.21	2.01	481	451	6.23	10.18	493	451	9.93	8.51
Thunder	500	497	492	1.01	1.60	479	432	9.81	14.19	489	462	7.77	5.52
Others	12,000	12,012	11,948	0.43	0.53	12,038	11,681	2.90	2.64	12,210	11,365	5.20	6.92

On the other hand, the performance of the HBC-FLS is not as good as that of our methods. The HBC-FLS method is based on statistical characteristics of the flow, including the packet interval and packet sequence. But those features are not stable under different network environments, such as the packet interval will become bigger when the network becoming congestion. Also the packet sequence is not stable as there are some resend packets due to packet lose. Furthermore, performance of the SKP based methods for identifying VoIP applications is even worse due to there are thousands of packets per seconds, it is hard to identify key packets from them. The performance for identifying P2P flows is also not so good compared with our methods. The key packets are strongly correlated with the version of the application software and the key packet sequences are different for the same application. Thus the methods have worse performance. If there are some packets lose during capturing process, the performance would become even worse.

In conclusion, our method is based on the *PSD*, which is not sensitive to packet lose as the *PSD* is a kind of probability feature. We also find the *PSD*s of different versions only very a little for specific software, such as Skype, which is important for detailed application identification.

6. Parameters selection for sampling method

6.1. Raw traffic collection

The actual testing beds are placed in the Northwest Regional Center of CERNET. The network being monitored contains more than 3,000,000 end users with self-governed IP addresses, including students, faculty members and contract personnel from service providing companies. The users in the monitored network usually use the VoIP to contact their friends outside of China, due to its easy using and free charging. They also use the internet for file download, online games, online movies, email, news, etc. The network environment is relatively complex and suitable for performance evaluation. Based on the testing beds, we collected several traces from an egress router with a bandwidth of 10 Gbps using the traffic collection tool – Coral Reef [33]. The detailed description of the traffic traces are listed in Table 4. The traces contain number of flows of the normal applications in current network, such as FTP, P2P, and E-mail. Furthermore, to give more reasonable results, the traces are collected at different time of different days.

6.2. Parameters selection of sampling methods

To select suitable parameters λ for the Poisson sampling method, we firstly analyze the number of packets reduced with different λ and the results are shown in Fig. 10a. As the figure shows, along with the incensement of λ, the sampling interval is becoming small and more packets are sampled. Suitable selection of λ not only can capture enough number of packets for application identification, but also reduce the number of packets being processed. Secondly, the similarly of the *PSD* between that of the total packets and that of sampled packets with different λ are shown in Fig. 10b, where the similarity are measured using Renyi cross Entropy. Based on the results of Fig. 10a and b, we select λ equal to 40,000 in this paper, which can reduce more than 80% of packets being processed while the *PSD*s between the sampling and un-sampling Bi-flows are similar with each other. Based on the λ selected, the differences of *PSD*s before and after sampling are analyzed, and the results are shown in Fig. 10c. As the figure shows, after sampling the percent of packet size between [200 1400] are increased. This is because there are mainly two kinds of traffic packets, the size smaller than 200 bytes and larger than 1400 bytes, usually the smaller one is used for connection establishment and control while the larger one is used for data transfer. After sampling, the percent of those packets will decrease.

Table 4
Traffic trace descriptions.

Type	Duration (h)	Begin time	Volumes (GB)	# of packets
1	5	2013.3.18 11:14	34.0	62,324,759
2	5	2013.4.21 09:27	37.1	67,509,581
3	5	2013.5.18 20:12	39.5	66,117,892
4	5	2013.6.11 08:10	37.5	65,730,275

We also analyzed the influence of the sampled methods on the *PSD* of the typical application selected and the analysis results are shown in Fig. 11. The results show there is no obviously changes of the *PSD* after sampling by selecting $\lambda = 40{,}000$ for Poisson sample.

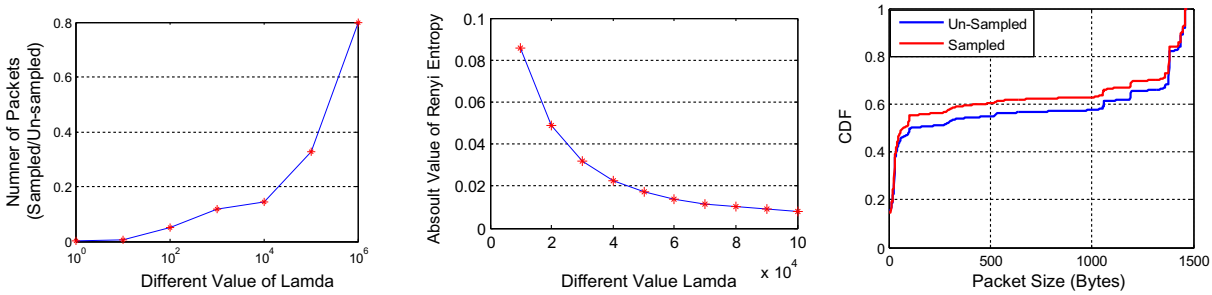
7. Performance evaluation

7.1. Performance evaluation of sampling method using artificial traffic

Firstly, the artificial traces are employed as benchmark to evaluate performance of identification methods combined with sampling. We run the same identification process using the sampled and un-sampled artificial traces and the results are shown in Table 5, as the table shows the sampling methods nearly have no influence on the results. These results also verify that the *PSD* is not sensitive to packet lost. The most important advance of combining with sampling methods is that after sampling the number of recorded data to be processed is reduced obviously, about 80% are reduced with λ equal to 10^4, which will greatly reduce the difficulty of traffic monitoring and management in high speed or backbone networks.

7.2. Performance evaluation of the proposed framework using actual traces

To verify performance of the developed framework in backbone networks, we apply our methods with sampling methods to the



a. Number of packets changed b. The similarity of *PSD* c. The sampling results of trace one

Fig. 10. Experimental results with different sampling ratio.

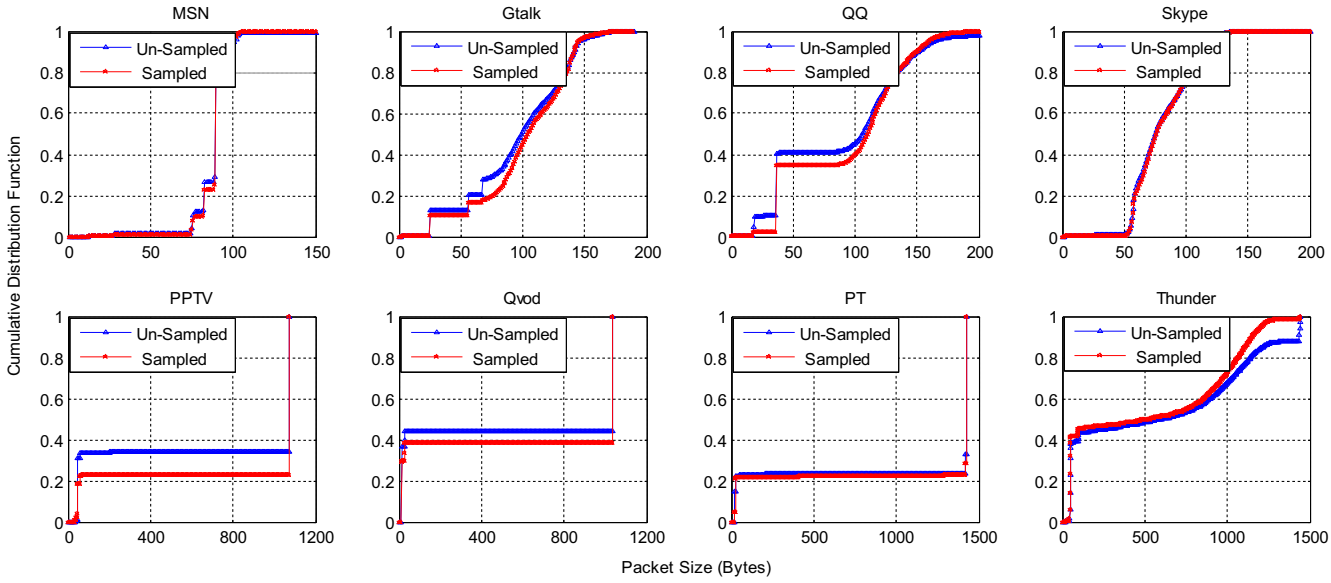


Fig. 11. Similarity of *PSD* of different application (one Bi-flow selected randomly).

Table 5
Identification results comparison.

Type	Un-sampled		Sampled	
	FPR (%)	*FNR* (%)	*FPR* (%)	*FNR* (%)
Skype	0.81	1.61	0.20	2.24
Gtalk	1.19	0.79	1.18	1.58
QQTalk	1.81	2.41	2.01	2.82
MSNTalk	1.01	2.85	1.41	2.01
PPTV	1.37	1.57	2.97	1.98
Qvod	0.00	0.00	0.79	0.39
PT	1.21	2.01	1.40	1.60
Thunder	1.01	1.60	0.81	2.85
Others	0.43	0.53	0.64	0.51

Table 6
Identification results using actual network trace one.

Type	INF	PFTF (%)	PPTP (%)	PBTB (%)	*FNR* (%)
Skype	2	0.0078	0.0259	0.0865	0
Gtalk	7	0.0274	0.8543	0.2017	0
QQ	157	0.6152	5.8726	1.9025	2.08
MSN	5	0.0196	0.9825	0.7310	0
PPTV	926	3.6285	4.2105	9.0043	0
Qvod	215	0.8425	2.7804	4.0258	4.26
PT	403	1.5792	3.9736	7.8972	2.56
Thunder	2741	10.7406	8.9058	15.3561	2.04

actual traces collected from CERNET, and the identification results are shown in Table 6, where INF is the identification number of flows of each application, PFTF is the percent of flow of specific application to the total number of flow in the trace, PPTP is the percent of packets of specific applications to the total number of packets in the trace, and PBTB is the percent of bytes of the specific applications to the total bytes in the trace. From the results we can get P2P applications occupied many of the traffic volumes in the campus network. The students also use some VoIP applications to connect with their friends, such as use Skype to connect with their friend outside of China. For this trace, we cannot obtain the *FPR* due to we do not know the accurate flow number of specific application in the raw data. But we can obtain the *FNR* using our methods. Based on the framework developed, we can obtain the specific number of flows of different applications. Then we employ the DPI methods to verify the identification results and calculate *FNR*. The results show that our methods have high accuracy for application identification in actual networks.

7.3. Discussion on the effect of NAT or proxy

The actual traces are collected from our CERNET, where the NAT is widely used and usually there are several computers behind one public IP address. In this kind of network environment, our methods also have high identification accuracy due to the following reasons: (1) Packets with the same source and destination IP

Fig. 12. Simple framework of online traffic monitoring system.

addresses are aggregated into Bi-flow, the destination addresses for different applications are usually different from each other. In this case, the packets belong to different applications are aggregated into different Bi-flows. Thus we can obtain the correct *PSD*s and perform accurate application identification. (2) If two users behind the NAT use the same application simultaneously and the destination addresses are same with each other. In this case, packets belong to the same application are also aggregated into one Bi-flow. Thus we can obtain the correct *PSD*s and perform accurate application identification. (3) If two users behind the NAT use two different applications simultaneously and the destination addresses are same with each other coincidentally. Packets belong to different applications will be aggregated into one Bi-flow. We cannot obtain the correct *PSD*s and the developed methods will loss efficiency in this situation, but the occurred probability of this situation is very small.

Accordingly, if the users only use the proxy, we can obtain the correct *PSD*s at most cases. But if the users use the proxy and change the payload size of the packets, we cannot obtain the correct *PSD*s and perform identification. To overcome this difficulty, we can integrate DPI methods into our framework and then update the *PSD*s of different applications frequently.

7.4. System implementation for online traffic monitoring

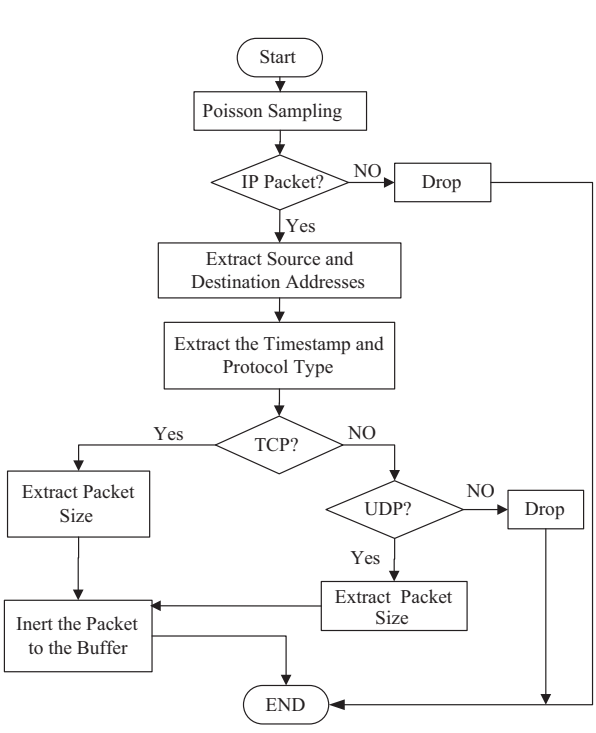
For online monitoring applications, we develop a traffic monitoring system based on the methods proposed in this paper. In Fig. 12, we give the simple framework of the monitoring system, we firstly employ the Poisson sampling method to reduce the number of packets need to be processed. Then we aggregate the sampled packets into Bi-flows and extract their *PSD*s, the identification process then follows. The traffic collection server and application identification server are powerful workstations configured with 8 Intel Core I7 980X and 16G RAM.

7.4.1. Data collection module

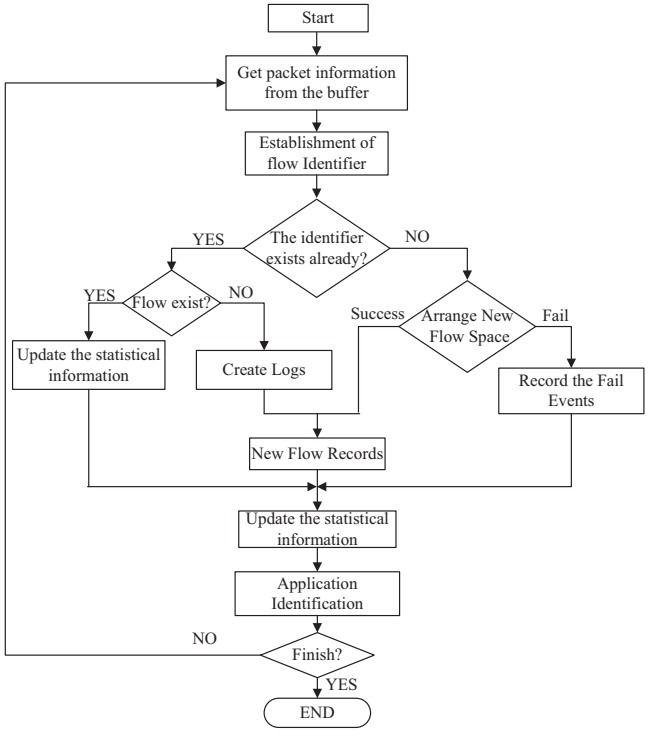
We collect the packets at the ingress router of the campus network of Xi'an Jiaotong University and the detailed collection process is shown in Fig. 13a. Firstly, we only focus on the IP packets and other kinds of packets are dropped directly and we only collected the packets with transfer protocol TCP and UDP. The features used for identification only include the source address, destination address, protocol, capturing timestamp and the payload size. After extracting those features the raw packets are dropped. To ensure that all of the packets are processed and their information is used to establish Bi-flows, we create a buffer with 5G to store packet information temporary. After the packet information is used and insert into Bi-flows, they are removed from the buffer immediately.

7.4.2. Flow establishment and identification module

Process of the flow establishment and identification module is shown in Fig. 13b. We first get packet information from the packet buffer and establish the Bi-flow based on the model defined. For those flows end with less than 2000 packets, we will drop them and not perform identification process. For those flows with more than 2000 packets, we only employ the *PSD* characteristics of the first 2000 packets to perform identification. For a long time testing last about one month, we find our methods can greatly reduce the computation complexity as we only employ about 20% of the total packets to perform identification.



a. Process of packet capturing



b. Flow establishment and identification

Fig. 13. Process of the monitoring.

But there is one limitation of our methods. Since of using the sampling methods, we only can identify the Bi-flows with long duration and hold huge number of packets. This is to say we can only identify the elephant flows in the traffic. But as the prior works claim that the elephant flows occupy more than 80% of the total traffic [34], thus our methods can deal with most of the traffic and the results obtained are enough for traffic control and management.

8. Conclusions and future works

In this paper, we develop a robust traffic identification method based on *PSD* of the first few packets in an entire flow. We find the *PSD*s are basically stable over the whole connection time and we employ the first few packets to capture the flow dynamics. We find the *PSD*s of different applications are different with each other, which can be used to identify the detailed application type. We also combined our methods with the Poisson sampling method to reduce the data to be processed. The experimental results based on traffic traces collected from the platform placed in the campus network of Xi'an Jiaotong University show that the proposed methods have identification accuracy above 97%. The testing results also verify that the proposed methods are robust and can be used for real-time traffic monitoring. As using the sampling methods, we only can identify the Bi-flows with long duration and hold huge number of packets. But as the prior works statements our methods can deal with more than 80% of the total traffic packets and the results obtained are enough for traffic control. In future works, we will evaluate performance of our methods when combined with different sampling methods, furthermore we will collect several different traces including NAT or proxy to evaluate the performance of our methods.

Acknowledgements

The research presented in this paper is supported in part by the National Natural Science Foundation of China (61221063, 61103240 and 61403301), the National Science & Technology Support Program of China (2011BAK08B02), the Application Foundation Research Program of SuZhou (SYG201227) and the Fundamental Research Funds for the Central University. The authors would like to thank Professor Don Towsley, Weibo Gong and Lixin Gao from University of Massachusetts, Professor John C.S. Lui from Chinese University of Hong Kong and Professor Zhili Zhang from University of Minnesota for their valuable comments and suggestions.

References

[1] S. Sen, J. Wang, Analyzing peer-to-peer traffic across large networks, IEEE/ACM Trans. Networking 12 (2) (2004) 219–232.
[2] N. Hua, H. Song, T. Lakshman, Variable-stride multi-pattern matching for scalable deep packet inspection, in: Proceedings of IEEE INFOCOM 2009, Rio de Janeiro, 19–25 April 2009, pp. 415–423.
[3] M. Bando, N. Sertac Artan, H.J. Chao, Scalable look ahead regular expression detection system for deep packet inspection, IEEE/ACM Trans. Networking 20 (3) (2012) 699–714.
[4] Z. Li, G. Xia, H. Gao, Y. Tang et al., NetShield: massive semantics-based vulnerability signature matching for high-speed networks, in Proceedings of the ACM SIGCOMM 2010, August 30-September 3, 2010, New Delhi, India, pp. 279–290.
[5] Y. Xu, L. Ma, Z. Liu, H.J. Chao. A multi-dimensional progressive perfect hashing for high-speed string matching, in: Proceedings of the ACM/IEEE Symposium on Architectures for Networking and Communications Systems, Brooklyn, October 2011, pp. 167–177.
[6] A. Dainotti, K. Pescape, Claffy, issues and future directions in traffic classification, IEEE Network 26 (1) (2012) 35–40.
[7] M. Korczynski, A. Duda, Classifying service flows in the encrypted Skype traffic, in: The Proceedings of 2012 IEEE International Conference on Communications, June 2012, pp. 1–5.
[8] M. Korczynski, A. Duda, Markov chain fingerprinting to classify encrypted traffic, in: The Proceedings of IEEE International Conference on Computer Communications, April 2014, pp. 1–9.
[9] G. Xie, M. Iliofotou, M.R. Keralapura, M. Faloutsos, A. Nucci, SubFlow: towards practical flow-level traffic classification, in: Proceedings of IEEE INFOCOM 2012, Orlando, FL, March 2012, pp. 2541–2545.
[10] Y. Jin, N. Duffield, J. Erman, P. Haffner, S. Sen, Z.-L. Zhang, A modular machine learning system for flow-level traffic classification in large networks, ACM Trans. Knowl. Disc. Data 6 (1) (Mar. 2012) 1–34.
[11] M. Iliofotoua, H. Kimb, M. Faloutsosa, M. Mitzenmacherc, Graption: a graph-based P2P traffic classification framework for the internet backbone, Comput. Netw. 55 (8) (2011) 1909–1920.
[12] B. Li, M. Ma, Z. Jin, A VoIP traffic identification scheme based on host and flow behavior analysis, J. Netw. Syst. Manage. 19 (2011) 111–129.
[13] W. de Donato, A. Pescapè, A. Dainotti, Traffic identification engine: an open platform for traffic classification, IEEE Network 28 (2) (2014) 56–64.
[14] Y. Lin, C. Lu, Y. Lai, W. Peng, P. Lin, Application classification using packet size distribution and port association, J. Netw. Comp. Appl. 32 (5) (2009) 1023–1030. 9.
[15] M. Chen, G. Zhang, J. Bi, Research of SIP-based VoIP traffic identification methodology, Appl. Res. Comp. 24 (4) (2007).
[16] Y. Wang, Y. Xiang, W. Zhou, S. Yu, Generating regular expression signatures for network traffic classification in trusted network management, J. Netw. Comp. Appl. 35 (3) (2012) 992–1000.
[17] S. Sen, O. Spatscheck, D. Wang, Accurate, scalable in-network identification of P2P using application signatures, in: Proceedings of the 13th International Conference on World Wide Web, 2004, pp. 512–521.
[18] M. Jaber, G. Cascella, C. Barakat, Using host profiling to refine statistical application identification, in: Proceedings of IEEE INFOCOM 2012, pp. 2746–2750.
[19] F. Constantinou, P. Mavrommatis, Identifying known and unknown peer-to-peer traffic, in: Proceedings of Fifth IEEE International Symposium on Network Computing and Applications, 2006.
[20] S. Tapaswi, A. Gupta, Flow-based P2P network traffic classification using machine learning, in: Proceedings of 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2013, pp. 402–406.
[21] H. Liu, W. Feng, Y. Huang, X. Li, A peer-to-peer traffic identification method using machine learning, in: Proceedings of Networking, Architecture, and Storage, 2007.
[22] A. Elnaka, Q. Mahmoud, Real-time traffic classifications for unified communication networks, in: Proceedings of 2013 International Conference on Selected Topics in Mobile and Wireless Networking, 2013, pp. 1–6.
[23] A. Ribeiro, R. Filho, J. Maia, Online traffic classification based on sub-flows, in: Proceedings of 2011 IFIP/IEEE International Symposium on Integrated Network Management, pp. 415–421.
[24] S. Zhao, X. Yu, Z. Chen, S. Jing, L. Peng, K. Liu, A novel online traffic classification method based on few packets, in: Proceedings of 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4.
[25] L. Bernaille, R. Teixeira, I. Akodkenou, Traffic classification on the fly, Comp. Commun. Rev. 36 (2) (2006) 23–26.
[26] P. Wang, X. Guan, T. Qin, P2P traffic identification based on the signatures of key packets, in: Proceedings of 14th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, 2009, pp. 1–5.
[27] F.H. Edward, Measuring network change: Renyi cross entropy and the second order degree distribution, in: Proceedings of Passive and Active Measurement Conference, 2006, April 2006.
[28] N. Duffield, C. Lund, M. Thorup, Flow sampling under hard resource constraints, in: Proceedings of ACM SIGMETRICS, New York, USA, 2004, pp. 85–96.
[29] X. Wang, X. Li, D. Loguinov, Modeling residual-geometric flow sampling, IEEE/ACM Trans. Networking 21 (4) (2013) 1090–1103.
[30] X. Ma, C. Hu, J, Jiang, J. Wang, S3: smart selection of sampling function for passive network measurement, in: Proceedings of IEEE 36th Conference on Local Computer Networks, 2011, pp. 416–423.
[31] C. Hu, B. Liu, S. Wang, J. Tian, Y. Cheng, Y. Chen, ANLS: adaptive non-linear sampling method for accurate flow size measurement, IEEE Trans. Commun. 60 (3) (2012) 789–798.
[32] M. Roughan, A comparison of poisson and uniform sampling for active measurements, IEEE J. Sel. Areas Commun. 24 (12) (2006) 2299–2312.
[33] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, The architecture of CoralReef: an internet traffic monitoring software suite, in: Proceedings of the Passive and Active Measurement Workshop, Amsterdam, Netherlands, 2001.
[34] T. Pan, X. Guo, C. Zhang, W. Meng, B. Liu, ALFE: a replacement policy to cache elephant flows in the presence of mice flooding, in: Proceedings of 2012 IEEE International Conference on Communications, pp. 2961–2965.
[35] T. Qin, X. Guan, W. Li, P. Wang, M. Zhu, A new connection degree calculation and measurement method for large scale network monitoring, J. Netw. Comp. Appl. 41 (15–26) (2014) 5.