Jestr

Journal of Engineering Science and Technology Review 7 (3) (2014) 151– 157

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Research Article

Traffic Classification Method by Combination of Host Behaviour and Statistical Approach

Ying Hou[1,*], Hai Huang[1], Wenchao Shao[1] and Heqing Huang[2]

[1]*National Digital Switching System Engineering & Techological R&D Center, ZhengZhou 450002, China*
[2]*University of Kentucky,Lexington, KY 40506–0046, USA*

Received 2 February 2014; Accepted 27 July 2014

Abstract

Traffic classification, one of the most active fields in Internet traffic research, is the substructure of network design and management. Generally, there are four techniques to identify the traffic, port-based, payload-based, flow statistic-based, and host-based approaches. In this paper, a hybrid method to classify the traffic was proposed combining the host behaviour and the Affinity Propagation (AP) algorithm. Simple features in the statistical process were selected at the first stage of classification; then, the initial classification results and the host behaviour model were combined to generate the final results. The host behaviour model was updated by the feedback of previous classification. The combining classification approach was evaluated on two real traces. The results indicated that the proposed technique offered improved performance compared with BLINC and independent AP algorithms.

Keywords: Traffic Classification, Affinity Propagation, Statistical Feature, Host Behaviour

1. Introduction

The perception of the traffic application is the important means in the field of network optimizing, resources allocation and abnormal behavior detection etc. It can be considered as the foundation of network management that contributes to deeply understand the nature of the network and effectively master the state of the network. In recent years, a large number of new technologies have been introduced to serve the needs of ever increasing scale of the Internet and its architecture. This development has also posed sever challenges to the traffic classification.

The conventional port-based approach is the fastest and simplest method to identify the applications of traffic. However, some modern applications which select the ports randomly are extremely difficult to be identified by the port-based approaches. Moreover, to trick firewalls, some applications conceal themselves by using standard ports of other applications, such as port 80. The ports are determined by applications, and can be easily changed by the end host. Hence, the port-based approach is less reliable [1].

The payload-based approaches, usually called "Deep Packet Inspection (DPI)" [2], have been widely applied in traffic identification field, particularly in network security. These techniques identify the applications by matching the payload of packets and the characteristic string associated with the applications, called application signature. When a new application appears, the signatures need to be found and labeled. The payload-based approaches are generally reliable except the formidable privacy and laws challenges. Besides, the exploration of the packets payload induces heavy load and high cost. That limits the general use on high-bandwidth links. And now many modern applications with encrypted payloads lead to this technology in vain[3].

The flow statistic-based techniques, also called Deep Flow Inspection (DFI), recognize the applications through a classifier based on machine learning methods. The classifier identifies the application based on the statistical signatures of the traffic[4][5]. The statistical signatures can be packet sizes, connection durations, inter-packet delays, or direction of the flow, etc. The machine learning methods include supervised algorithms, unsupervised algorithms, and semi-supervised algorithms. In the supervised algorithms, the classifier is built on the training samples, which have been labeled the applications. The unclassified traffic is identified by matching the statistical signatures of the classifier established by the training sample. When a new application appears, the classifier must be retrained. in unsupervised algorithms, the classes needn't be predetermined. The algorithms construct distinct classes (e.g., clustering) of traffic and then assign these classes to corresponding applications. In [6], McGregor et al. presented the clustering algorithms (EM) to the identification process. Unsupervised algorithm do not require the information of samples and can identify the obfuscated and encrypted traffic. In Semi-algorithm, The clustering also plays key role and the training samples are taken into account for the application assignment.

The host-based approach is a new branch of traffic classification. It exploits the host behavior to resolve the classification issue. These algorithms apply the heuristics theory to perform the classification effectively, especially for obfuscated traffic. Some of the recent studies have focused on this aspect[7]. In [7], the social network of a given host is correlated with its transport-level interactions to identify peer-to-peer application.

With the development of the Internet applications, a single traffic classification technique can't achieve

* E-mail address: ndschy@139.com

acceptable accuracy[9]. The researchers have started to consider more general and effective techniques that use several classifiers and combine the results of different classifiers by intelligent hybrid algorithms. It is claimed that the combination of a set of classifiers may compensate for the weakness of a single classifier in the classification process[10]. Such a multi-classifier system can achieve higher accuracy compared to a single classifier, and it is more robust to the variation of the sample population, including the nature and mix of applications. Some of the multi-classifier systems pay attention to the combination of same classifiers with different parameters; while others combine several different classifiers (such as port-based classifier and flow statistical classifier). The combination of multi classifiers are usually based on voting, Bayesian probability, or Dempster-Shafer theory. The combination of multi classifiers results in increased computational complexity. However, Alberto[9] has indicated that assuming different classifiers in the combination can execute in parallel, the flexibility offered by combination classifiers facilitates the scalability trade-offs, essential for online techniques.

In this paper, a hybrid approach is proposed that combines the host behavior and DFI classifier to categorize Internet traffic. The proposed approach involves three major steps. In the first phase, the initial classification of the traffic is performed by an iterative statistical algorithm called Affinity Propagation (AP). As second step, the results of initial classification are refined with the host behavior model. Finally, the host behavior model is updated based on the classification results.

The rest of the paper is organized as follows. Section 2 introduces AP algorithm and the features selected in this study. Section 3 describes the methodology to portray the host behavior. Section 4 describes the combination of the two methods. Section 5 discusses the experiments and the evaluation results. Finally, conclusion is presented in Section 6.

2. AP algorithm

AP algorithm[11] is an unsupervised clustering method based on the dissemination of the nearest neighbor information. The goal of this algorithm is to find the optimal representative points that have maximum similarity among a subset of points and test data is classified by finding the nearest representative point. If the similarity is equal to the negative value of the Euclidean distance, the objective function of AP is consistent with K-means; that minimizes the quadratic sum distance of the data point to the nearest representative point. In AP algorithm, considering all the data points as candidates of representative point, avoids the limitations associated with selection of the initial representative point. It is simple and efficient due to the optimization objective function that propagates similarity. Furthermore, it does not rely on the symmetrical nature of the similarity between the data points.

The similarity matrix of n sample points is the key structure of AP algorithm. All of the n sample points are considered as candidate representative point (i.e. potential cluster center). For each point, a $n \times n$ similarity matrix $S_{n\times n}$ is established to indicate its attraction to other points. $s(i,j) = -\left\|X_i - X_j\right\|^2 (i \neq j)$ indicates the degree of X_j being the representative point of X_i. Large $s(i,i)$ means that there is more probability of X_i being a representative point. There are two messages delivered to each point, i.e., responsibility and availability. The parameter $r(i,k)$ describes the degree of k being the representative point of i. While $a(i,k)$ describes the degree of i selecting the data point k as its representative point. The parameters $r(i,k)$ and $a(i,k)$ are set to 0 initially and updated in turn. The update process is as follows:

$$r(i,j) \leftarrow s(i,j) - \max_{k \neq j}\left\{a(i,k) + s(i,k)\right\}$$

$$a(i,j) \leftarrow \begin{cases} \min\left\{0, r(j,j) + \sum_{i' \neq i, i' \neq k} \max(0, r(i',j))\right\}, i \neq j \\ \sum_{i' \neq j} \max(0, r(i',j)), i = j \end{cases} \quad (1)$$

A damping factor λ is introduced to improve the convergence of the algorithm. The weighting update process is described in equation (2):

$$r^{(t)}(i,j) \leftarrow \lambda \cdot r^{(t-1)}(i,j) + (1-\lambda)\cdot\left\{s(i,j) - \max_{k \neq j}\left\{a^{(t-1)}(i,k) + s(i,k)\right\}\right\}$$

$$a^{(t)}(i,j) \leftarrow \begin{cases} \lambda \cdot a^{(t-1)}(i,j) + (1-\lambda)\cdot \min\left\{0, r^{(t-1)}(j,j) + \sum_{i' \neq i, i' \neq k} \max(0, r^{(t-1)}(i',j))\right\}, i \neq j \\ \lambda \cdot a^{(t-1)}(i,j) + (1-\lambda)\cdot \sum_{i' \neq j} \max(0, r^{(t-1)}(i',j)), i = j \end{cases} \quad (2)$$

$$0 \leq \lambda < 1$$

The iterative process given in equation (2) terminates when one of the following conditions is fulfilled: a) current iteration exceeds the maximum number of iterations; b) the step size is under a fixed threshold; c) the representative points are stable.

After the iterative process, $\arg\max_j \left(a^{(t)}(i,j) + r^{(t)}(i,j)\right)$ is calculated, and X_j is selected as the representative point of X_i.

A cluster set $C\{C_1, ...C_k\}$ and representative points set $w\{w_1 ... w_k\}$ are established after clustering. When there are some labeled samples, the application with the maximum samples in a certain cluster is set to that cluster. When there are no samples in the cluster, the cluster is identified as unknown application.

The statistical features in AP algorithm are shown in Table 1.

Tab. 1 Selected features of traffic classification

feasure	description
port	transport-layer ports
P_1~P_N	the first N packets size
pattern($b_1b_2...b_N$)	communication pattern

§The transport layer port is not a major feature in traffic classification. However, it is still one of the important features, especially in classification of some of the traditional applications in Internet. Zhang[12] has proved by statistical methods that it can get relatively high classification accuracy, by using the features of the first four packets. Packets size is commonly used in traffic classification. The communication pattern defined in [13] is used to describe the direction of the first N packets (except the control packets, e.g., SYN, RST, FIN) and signify by a binary integer $b_1b_2\mathrm{L}\ b_N, b_i(i \le N)$; this signifies the direction of *i*th packet. The parameter $b_i = 1$ when the direction of *i*th packet is same as the SYN direction in TCP, otherwise $b_i = 0$. Let *N=4,* in general, the communication pattern of FTP is 0101, and that of HTTP may be 1000-1111.

Tab. 2 Communication pattern of four applications statistic in WIDE data set

pattern	FTP	HTTP	BitTorrent	eDonkey
0000	2.19	0	0	0
0001	1.19	0	0	0
0010	3.43	0	0	0
0011	0	0	0.01	0
0100	0.33	0	0	0
0101	92.84	0	0.01	0
0110	0.02	0	0.01	0
0111	0	0	0	0
1000	0	74.39	0.73	0.2
1001	0	3.34	6.09	84.41
1010	0	15.79	83.73	14.13
1011	0	0.17	6.94	0.04
1100	0	3.44	0.07	0.47
1101	0	1.05	1.47	0.74
1110	0	1.5	0.08	0.01
1111	0	0.32	0.86	0

Table 2 shows the communication pattern of the statistics of four applications from WIDE data set[14]. The FTP and HTTP are representative applications of C/S. It can be seen from the table that 92.84% of the patterns of FTP is 0101, and 74.39% of HTTP is 1000. BitTorrent and eDonkey are P2P applications; their patterns are mainly 1010 and 1001.

AP clustering algorithm keeps looking until the highest assignment probability exceed a predetermined threshold or the maximum detection number is reached. After all the clusters and the representative points are established, the cluster is labeled as the class that has maximum samples with the same communication pattern.

3. Host behavior description

Some researchers have studied the host behavior on residential networks in the field of traffic measurement [15][16]. The applications that run at the host and generate the traffic are part of the host traffic profile. The traffic profile indicates the preferred applications used on the host. Since the preferred information of the host is relevant to the latter application, it can help with the classification of the traffic. For instance, a host browsing the Web is more prone to open consecutive HTTP connections. A host is very likely to receive POP3 flows when it is running POP3 mail server.

The identification of a flow is five-tuple information: source IP address, source port number, destination IP address, destination port number, and protocol type. The source host of a flow is the host sending the first packet, while the destination host is the one receiving it. *F* is denoted as a function that associates a flow between a source and destination to an application *A(i)*. S (or D) is denoted as the generic source (or destination) host of a flow. Thus, F_S and F_D are the functions that assign the data flow to the application A_S and A_D based solely on the traffic of the source and destination, respectively. Let $P(F_S = A_S|S)$ be the probability that the flow is of an application A_S. Then, $P(F_D = A_D|D)$ means the probability that the flow is of an application A_D. The following is the computation of the probability that a flow is of application *A(i)*.

$$\begin{aligned} P(F = A(i)) &= P\left(\left(F_S = A(i)_S\right) \mathrm{I}\ \left(F_D = A(i)_D\right) | A_S = A_D\right) \\ &= \frac{P(F_S = A(i)|S) * P(F_D = A(i)|D)}{\sum_{j=1}^{N} P(F = A(j)|S \mathrm{I}\ D)} \\ &= \frac{P(F_S = A(i)|S) * P(F_D = A(i)|D)}{\sum_{j=1}^{N} P(F_S = A(j)|S) * P(F_D = A(j)|D)} \end{aligned} \quad (3)$$

Equation (1) needs the information of each host. In general, the monitor passively captures the flows between the two hosts and can only records the traffic of one of the two hosts of a flow. Assume a uniform probability for the host that the monitor has no information about it, and then the equation (1) can be simplified to:

$$P(F = A(i)) = P(F_S = A(i)_S|S) \quad (4)$$

or

$$P(F = A(i)) = P(F_D = A(i)_D|D) \quad (5)$$

Table 3 shows the distribution of application in a host as source and destination.

Tab. 3 Distribution of application in a host

Applications	FTP	HTTP	POP3	SMTP	SSH
Source	0.02	0.67	0.06	0.21	0.04
Destination	0.15	0	0.15	0.28	0.42

Th distribution of application of a host is updated after a new flow of the host is classified. The process is as follow. Let $P_{n-1}(A(i))$ denote the probability of *A(i),* calculated by the past *(n-1)* flows. When *n-th* flow is collected and classified, $P(F_s(n) = A(i))$ is the result. Then the probability of *A(i)* is computed as follows.

$$P_n(A(i)) = \beta * P_{n-1}(A(i)) + (1-\beta) * P(F_s(n) = A(i)), \quad 0 \le \beta \le 1 \tag{6}$$

β is the regulatory factors and represents the proportion of that the past distribution affects $P_n(A(i))$. When β is set to be close to 0, most recent flows affect $P_n(A(i))$. When β=0, $P_n(A(i))$ is completely decided by current flow. When β is set to be close to 1, $P_n(A(i))$ is calculated by applications of all previous flows. Initially, all applications are assigned to be uniform distribution. When β=0, $P_n(A(i))$ is completely decided by the previous flows. We will discuss β in section 5 according to the experiments. It should be noticed that, as described in equation (7), the source and destination of the host are computed respectively.

$$P_n(A(i)|S) = \beta * P_{n-1}(A(i)|S) + (1-\beta) * P(F_s(n) = A(i)|S)$$
$$or$$
$$P_n(A(i)|D) = \beta * P_{n-1}(A(i)D) + (1-\beta) * P(F_s(n) = A(i)D), \quad 0 \le \beta \le 1 \tag{7}$$

Practically, the monitor usually stores the host systems inside the ISP network and some host systems outside the network; e.g., obvious server of some applications to support the classification of Internet flows. The distribution of the host traffic, provides a statistical indication of the preferred applications that is running at the host.

4. Combination

This section will introduce the ways of integration of AP results and host behavior algorithm. *P(A(i))* is defined as the probability that the flow is generated by application *A(i)*. The $P(F = A(i))$ is the probability that a flow arrives from the application *A(i)* based on their source or destination host and it is calculated in Eq. (3). $P'(A(i)$ is the AP cluster results. N is the total number of classes got from AP clustering algorithm. The classification process is performed by computing the Euclidean distance between the feature of the new flow and the representative points.

According to the descriptions, *P(A(i))* is expressed by the following formulas:

$$P(A(i)) = \frac{P(F = A(i)) * P'(A(i))}{\sum_{j=1}^{N} P(F = A(j)) * P'(A(j))} \tag{8}$$

The assignment probability *P(A(i))* is calculated by combining the result of the initial classification using AP clustering method, and the result of the classification of the host behavior. The pattern of the hosts can be used to predict the type of application for the following traffic. The assignment probability *P(A(i))* is determined by only $P'(A(i)$ when the application is new in the host. The host application distribution is updated before the following identification steps with hybrid probability method.

The approach predicts the traffic of a host by the application pattern of that host. Compared with other techniques, it is simpler because it needs not research the relations between different hosts. All of the information can be easily obtained.

5. Experiments and results

The proposed hybrid approach was evaluated on two real data traces. The first trace was collected from XINDA university, named XINDA trace; it was collected on five consecutive days from the points connected to Internet. The second one was WIDE trace [14], obtained from a traffic data repository maintained by the MAWI Working Group of the WIDE Project, which survey and analysis the traffic of Internet backbone. We use the traffic trace collected at samplepoint-F. Each packet of the traces contains 40byte payload of anonymous user information that is kept private here.

The applications associated with the traffic were determined with a deep packet inspection method. The traffic associated with five applications was collected and two new traces were obtained. Each of the traces consists of two sets, a training set and a testing set. The training set consists of an equal number of flows per application to ensure that there is no bias in the learning phase. The details of the two traces are illustrated in Table 4 and Table 5.

Tab.4 Detail of WIDE trace

application	training	testing
HTTP	1000	315,309
FTP	1000	184,536
POP3	1000	9,974
eDonkey	1000	87,943
BitTorrent	1000	103,945

Tab.5 Detail of XINDA trace

application	training	testing
HTTP	1000	31,200
FTP	1000	9,003
POP3	1000	19,328
eDonkey	1000	3,902
BitTorrent	1000	4,013

5.1 Metrics

The metrics used to evaluate the performance of the combination approach are recall, precision, and overall accuracy. For application *i*, TP_i is denoted as the number of flows that are correctly classified to application *i*. FP_i is the number of flows that are incorrectly classified to application *i*. FN_i indicates the number of flows of application *i* that are incorrectly classified to other applications. The metrics are given as follows.

Recall of application *i*:

$$R_i = TP_i / (TP_i + FN_i) \tag{9}$$

Precision of application *i* :

$$P_i = TP_i / (TP_i + FP_i) \tag{10}$$

F-measure of application *i* :

$$F_i = 2 * P_i * R_i / (P_i + R_i) \tag{11}$$

Overall accuracy:

$$OA = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} (TP_i + FP_i)} \tag{12}$$

In the above metrics, recall and precision of an application indicates the classification ability of the model to this application. The F-measure signifies the comprehensive classification ability of the classifier to an application. The overall accuracy is given as the weighted average over all the applications and shows the overall classification ability of the classifier.

5.2 Regulatory factor β and precision

The performance of the classification method is discussed when the host behavior is combined with the results computed by the statistical method.

For Trace I, Fig. 1 plot F-measure of each application, when 1 to 10 packets are used to extract classification features, as we discuss in Section 3. In the plot, different lines correspond to the overall accuracy of the combination approach with different β.

The figures verify the validity of the combination approach. Fig. 1(a) plot F-measure of HTTP as a function of the packets' number respectively. with the number of the packets increasing, the precision of the classifier is improved. When $\beta = 0.9$, F-measure can achieve approximate 90% only with two packets. Even when $\beta = 0.1$, means with a small number of host information, we acquires a very high accuracy.

Entdonkey and BitTorrent are P2P applications. As is known to all, for the characteristics of P2P applications, P2P traffic identication is very important for ISP to manage the network. Fig. 1(d) and Fig. 1(e) demonstrate that the combinating approach also obtains good performances in case of P2P traffic identification.

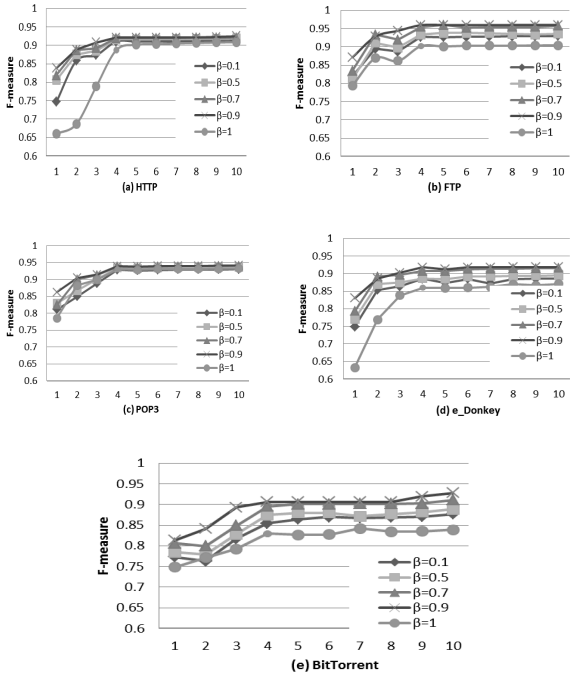

Fig.1 F-measure versus the number of packets (Trace I)

For Trace II,The experimental results of F-measure of each application, versus the number of packets of the flows, are plot in Fig. 2.The different lines in the plot correspond to the overall accuracy of the method with different β.

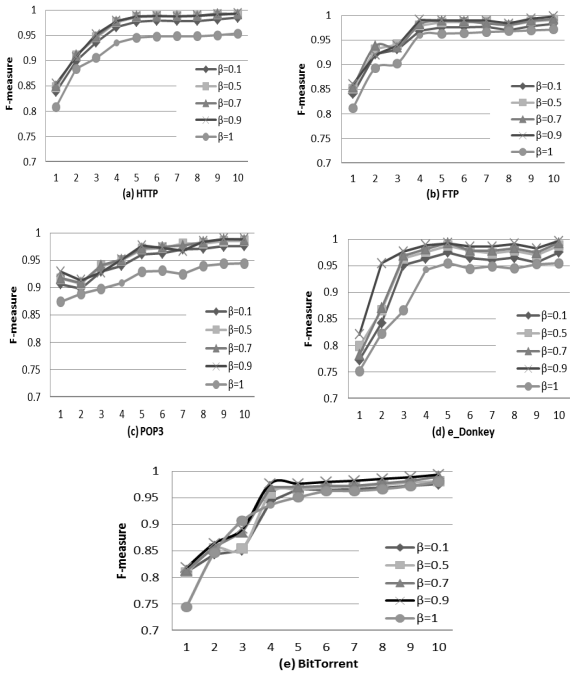

Fig.2 F-measure versus the number of packets (Trace II)

Trace II is collected from local area network (LAN). The number of hosts in the trace is smaller than that in Trace I. Then the host information plays a vital part in the classification.

From Fig.2, we can notice clearly the importance of using the host information. For example, in Fig. 2(a), when β=0.9, after four packets, F-measure exceeds 98.6% and reach 99.36% with ten packets. In Fig.2(d) and Fig.2(e), it can be observed that with more packets we can obtain better precision, especially with four and more packets. This is clear for all values from $\beta = 0.1$ to $\beta = 1$

Fig. 2(c) shows that when recent flows are set more weight, most of the POP3 traffic is correctly classified. The analysis of the traffic shows that the POP3 traffic is predominant in some hosts. That confirms the benefit of the host information. With all values of β, the accuracy of POP3 identification is high.

The experimental results of overall accuracy for Trace I and Trace II, versus the number of packets of the flows, are plot in Fig. 3 and Fig. 4. The different lines in the plot correspond to the overall accuracy of the method with different β.

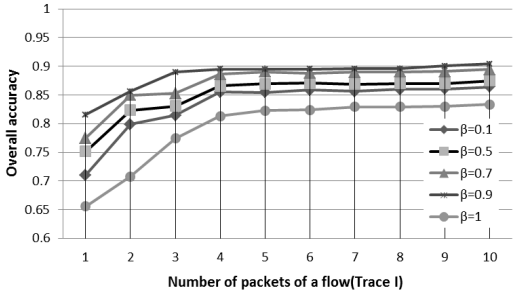

Fig.3 Overall accuracy versus the number of packets (Trace I)

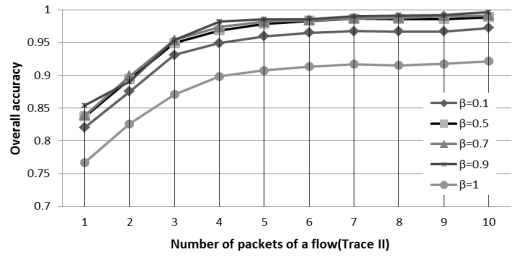

Fig.4 Overall accuracy versus the number of packets (Trace II)

The regulatory factor β can be understood as the ratio of the two parts of the combination approach. When β=1, the behavior of the host is not used, which means each application of a host has a uniform probability. The results demonstrates that the accuracy improves considerably when the prior application distrabution of the hosts is used to judge the application of current flows. When β=0.1, the host behavior is decided by the classification results of the most recent flows. When β=0.9, the classification results of flows during a long period characterize the host behavior. The overall accuracy of Trace I is 90.42% when β=0.9 and the number of packets is 10. From the figures, we can see that the classifier gets the best performance when β=0.9. According to the above results, in the following experiments, β is set to 0.9.

When β=0.5, the overall accuracy of Trace I is 86.59% with four packets and reach 87.48% after ten packets. When β=0.9, the overall accuracy of Trace I is 89.52% with four packets and reach 90.42% after ten packets. When β=0.9, the overall accuracy of Trace II is 98.19% with four packets and reach 99.61% after ten packets. The increase of number of packets does not bring an prominent improvement of the precision. We can conclude that with the first four packets of each flow to calculate the statistical features, the classifer can get high precision. It is worth of paying attention to because that means the appoach can be used in on-line traffic classification.

5.3 Discussion of training set

F-measure of Trace I and Trace II are plotted in Fig. 4 and Fig. 5 respectively, versus the number of flows in training set.

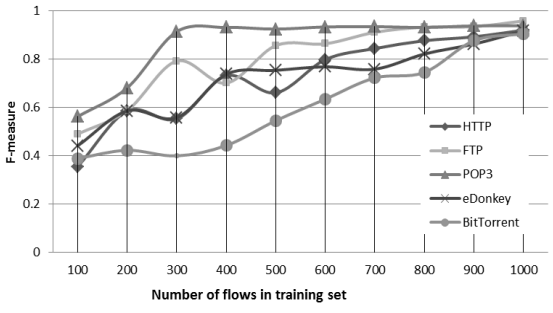

Fig.5 F-measure versus number of flows in training set (Trace I)

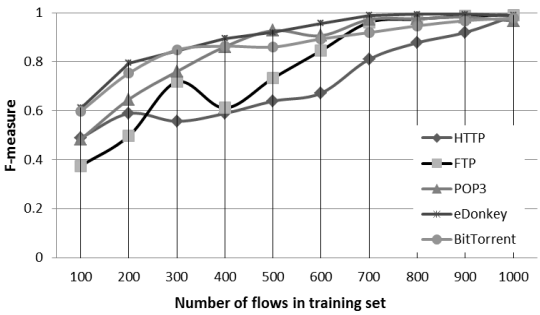

Fig.6 F-measure versus number of flows in training set (Trace II)

It can be seen from Fig.5 and Fig.6 that F-measure fluctuates significantly when the training set is smaller than 600. The performance of the classifier is better when the packets number of the training sets increases. In Trace II, the proportion of the training set is higher than that of Trace I, so that a higher precision is obtained. When there are 1000 in the training set, F-measure of all five applications exceeds 96% and FTP is 99.1% in Trace II. it can be concluded that the ratio of training set and test set affects the results of classification.

5.4 Comparision of algorithms

Based on Trace II, the proposed algorithm was compared with BLINC and AP algorithms. The results are shown in Fig. 7.

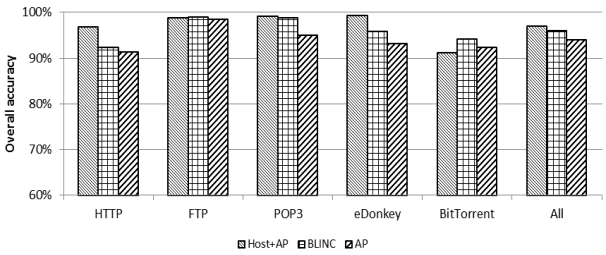

Fig.7 Comparisons of three algorithms overall accuracy

The BLINC algorithm uses host model to classify the applications; whereas, AP algorithm is based on the statistic parameters. The experimental results show that the combination of host behavior model and the statistical methods significantly increases the overall accuracy.

6. Conclusion

This paper introduces a traffic classification algorithm that combines AP algorithm and host behavior. According to the facts that the early application of host can predict the latter traffic, the host application distribution is established. Then, this distribution is combined with the initial results obtained from AP algorithm. The combination provides the final results of traffic classific ation.

The proposed method was applied on two real data traces with very promising results. The results demonstrate that when increasing the size of training data, the performance of the combination algorithm is improved. The comparison with BLINC and AP algorithm indicates that the overall accuracy of the new approach is more satisfying than that of the other two algorithms. In addition, since the statistical features selected in this paper are easy to calculate, the approach offers the prospect of online classification.

Acknowledgment

The corresponding author of this paper is Hai Huang, professor of National Digital Switching System Engineering & Techological R&D Center. And this research was financially supported by a research grant from the National Natural Science Foundation of Chinese government (No.61309019)

References

1. A.Callado, C.Kamienski, et al., "A Survey on Internet Traffic Identification", *IEEE Communications Survey & tutorials*, 11(3), 2009, pp. 37-52
2. A. Moore, K. Papagiannaki, "Toward the accurate identification of network applications," *Proceedings of Passive and Active Measurement*, Boston, USA, 2005, pp. 41-54.
3. T.T.T. Nguyen, G. Armitage, "A Survey of Techniques for Internet Traffic Classification Using Machine Learning," *IEEE Communications Survey & tutorials*,10(4), 2008, pp. 56–76.
4. M.Crotti, M.Dusi, F.Gringoli, L.Salgarelli, "Traffic classification through simple statistical fingerprinting", *ACM-Sigcomm Computer Communication Review*, 37(1), 2007, pp. 5–16.
5. M. Jaber, C. Barakat, "Enhancing application identification by means of sequential testing," *Proceedings of 8th international IFIP-TC 6 Networking Conference*, Aachen, Germany, 2009, pp. 287-300.
6. A. Mcgregor, M. Hall, P. Lorier, J. Brunskill, "Flow clustering using machine learning techniques," *Proceedings of Passive and Active Measurements*, Antibes Juan-les-Pins, FRANCE, 2004, pp. 205–214.
7. T. Karagiannis, K. Papagiannaki, M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," *Proceedings of ACM SIGCOMM*, Philadelphia, USA, 2005, pp. 229-240.
8. A. Dainotti, A. Pescape, K. C. Claffy. "Issues and Future Directions in Traffic Classification, " *IEEE Network*, 26(1),2012,pp. 35-40.
9. G. Aceto, A. Dainotti, W. De Donato, P. Antonio, "PortLoad: Taking the Best of Two Worlds in Traffic Classification," *Proceedings of IEEE INFOCOM*, San Diego, USA, 2010, pp. 1–5.
10. B. J. Frey, D. Dueck, "Clustering by Passing Messages between Data Points" , *Science*, 315(5814), 2007,pp.972-976.
11. H.L.ZHANG, G. Lu, "Machine Learning Algorithms for Classifying the Imbalanced Protocol Flows: Evaluation and Comparison". *Journal of Software*,23(6), 2012,pp.1500−1516.
12. Z. Yang, L.Z. Li, "Network traffic classification using decision tree based on minimum partition distance", *Journal on Communication,* 33(3) 2012,pp.90-102.
13. the MAWI Working Group of the WIDE Project. MAWI Working Group Traffic Archive; 2014. Retrieved June 3, 2014, from: http://mawi.wide.ad.jp/mawi/ .
14. M. Iliofotou, B. Gallagher, T. Eliassi-Rad, G. Xie, M. Faloutsos, "Profiling-by-association: a resilient traffic profiling solution for the Internet backbone," *Proceedings of ACM CoNEXT*, Philadelphia, USA, 2010.
15. M. Pietrzyk, L. Plissonneau, G. Urvoy-Keller, T. En-Najjary, "On profiling residential customers," *Proceedings of 3rd IEEE International Traffic Monitoring and Analysis Workshop*, Vienna, Austria, 2011, pp. 1-14.