A Semi-Supervised Clustering Method For P2P Traffic Classification

Bin Liu
Network Centre
HuaZhong University of Science and Technology Wuhan, China
Email: bliu@mail.hust.edu.cn

***Abstract*—In the last years, the use of P2P applications has increased significantly and currently they represent a significant portion of the Internet traffic. In consequence of this growth, P2P traffic identification and classification are becoming increasingly important for network administrators and designers. However, this classification was not simple. Nowadays, P2P applications explicitly tried to camouflage the original traffic in an attempt to go undetected. This paper present a methodology and selection of three P2P traffic metrics and applies semi-supervised clustering to identify P2P applications. Three P2P traffic metrics: IP Address Discreteness, Success Rate of Connections and Bidirectional Connections rate had been proposed and used in this paper. The semi-supervised classification method for P2P traffic consist two steps: Particle Swarm Optimization (PSO) clustering algorithm was employed to partition a training dataset that mixed few labeled samples with abundant unlabeled samples. Then, available labeled samples were used to map the clusters to the application classes. Experimental results using traffic from campus showed that high P2P traffic classification accuracy had been achieved with a few labeled samples.**

***Index Terms*—P2P, Particle Swarm Optimization, P2P Traffic Classification, Semi-Supervised Clustering**

I. INTRODUCTION

Identification of traffic that is generated by P2P applications is an important part of traffic classification, which is a key technology for many network management tasks including application-specific traffic engineering, network planning and design, diagnostic monitoring, service differentiation, and accounting. For example, enterprise network operators would like to limit or even block P2P traffic which could lead to network congestion to ensure business critical applications have enough bandwidth. ISPs and campus network operators have similar requirement to limit P2P traffic to avoid congestion and reduce cost they are charged by upstream ISPs.P2P applications have grown and evolved a lot ever since their emergence in the late 90s of the last century and the task of identifying P2P traffic is becoming more challenging.

The first techniques to detect P2P traffic were port-based. These techniques use layer 4 port number information in the packet header, and a list of known ports to identify the P2P communication packets. Port-based detection is effective for P2P applications which use static ports, however currently most of the P2P applications have dynamic ports. Signature-based detection techniques solve this problem by inspecting the payloads of the packets to locate specific string series which are called signature. Despite its accuracy, signature-based detection has a number of problems. The search for the signatures in the payloads requires mirroring or intercepting the traffic. It is difficult to implement signature-based detection on backbone links. Signature-based detection techniques are only successful for applications whose signatures are available. Hence, the signature databases must always be kept up to date. An additional drawback is that the user privacy is violated by inspecting the packet payload. Furthermore encryption is used increasingly in data transfer, and signature-based detection is not effective for encrypted data.

Traffic behavior-based P2P traffic detection techniques do not have the disadvantages of the signature-based and port-based detection methods. Traffic behavior-based detection uses the information obtained from IP headers. Hence, unlike signature-based techniques, traffic behavior-based P2P traffic detection can scale to high speed links. Furthermore traffic behavior-based detection is not affected by the payload encryption. Traffic behavior-based P2P traffic detection can be formulated as a problem of classifying traffic into P2P and non-P2P. So, this method cannot do an accurate classification for P2P traffic.

Machine Learning, which aims to classify data based on either a priori knowledge or statistical information extracted from raw data, is a powerful tool in data separation in many disciplines. Several machine learning techniques were proposed to classify P2P traffic, each with reasonable successes.

However, there are two main challenges for classifying network traffic using machine learning method. Firstly, labeled samples are scarce and difficult to obtain. With few labeled samples, traditional supervised learning methods often produce classifiers that do not generalize well to previously unseen samples. Secondly, not all types of applications generating samples are known a priori, and new ones may appear over time. Traditional

This paper is supported by the Basic Scientific Research Funds of Central Colleges of China (Grant numbers:c2009m058) and CNGI2008-123.

doi:10.4304/jnw.6.3.424-431

supervised methods force a mapping of each sample into one known classes, without the ability to detect new types of samples.

To address the above-mentioned problems, a semi-supervised learning method is proposed. This method consists of two steps. Firstly, a Particle Swarm Optimization (PSO) clustering algorithm was employed to partition a training dataset that consists of scarce labeled samples combined with abundant unlabeled samples. Secondly, the available labeled samples are used to obtain a mapping from the clusters to the different known classes.

II. RELATED WORK

There has been much recent work in the field of P2P traffic classification. This section will survey the different techniques presented in the literature.

A. Port Number approaches

The traditional method relies on linking a well-known port number with a specific application, so as to identify different Internet traffic. The port-based method is successful because many well-known applications have specific port numbers (assigned by IANA). For example, HTTP traffic uses port 80; Ftp port 21.But with the emergence of P2P application, the accuracy of port-based is declined sharply. Because such application tries to hide from firewalls and network security tools by using dynamic port numbers, or masquerading as HTTP or FTP applications. So the port-based method is no longer reliable. Ref [1,2] analyze and characterize P2P traffic that matches the default port numbers the corresponding P2P applications use. The work makes sense because P2P applications used well-defined port numbers at that time. However, nowadays, more and more P2P applications never use fixed port numbers any more. It is no longer possible to make use of port numbers for identifying P2P applications

B. Signature-based approaches

In order to deal with the disadvantages of the above method, a more reliable technique is to inspect the packet payload. In these methods, payloads are analyzed to determine whether or not they contain characteristic signatures of known applications. This technique can be extremely accurate when the payload is not encrypted. But this assumption is unrealistic because some P2P applications by use of different methods (encryption, variable-length padding), to avoid detecting by this technique. In addition, the demand of high process and storage capacity is discouraged, and privacy is concerned with examining user information.

Sen et al. [3] derive application layer signatures for five file-sharing P2P protocols by examining available documentations and packet-level traces. They implement an online P2P application classifier using these signatures and evaluation results show that the approach has less than 5% false positive and false negative. Karagiannis et al. [4], Bleul et al. [5] and Spognardi et al. [6] have also used protocol signatures in their classifiers to identify P2P traffic. Approaches based on signature-matching provide high accuracy but there are still some limitations. Firstly, signature-deriving is a time-consuming job, because most P2P protocols are proprietary and reverse engineering is needed. Secondly, they can not deal with brand-new applications that use unknown P2P protocols. Thirdly, they are unable to detect P2P traffic that is encrypted, even when only protocol headers are encrypted.

C. Traffic behavior-based approaches

To overcome above limitations, researchers make use of traffic behavior-based heuristics to identify P2P traffic. Karagiannis et al., [7] identify P2P traffic from connection patterns and the concurrent use of UDP and TCP, which can classify 95% of P2P flow and bytes with false positive ranges from 8% to 12%,compared with a signature-based approach. Constantinou and Mavrommatis classify P2P traffic based on connection direction and number of peers in a connected group, achieve false negative around 10% and additional positive around 20%, comparing with a known-port based approach in [8]. In later work, Karagiannis et al. [9] introduce BLINC, a general classification mechanism that classifies hosts based on protocol usage, port usage and connection patterns. These methods rely on behavior that is inherent to P2P applications.

D. Machine learning-based approaches

Machine Learning is an important research direction of modern artificial intelligence. The ability of continually gaining new knowledge or skills, reorganizing knowledge structure to improve their performance, has let it become a widely used method in network traffic classification.

Machine learning techniques can be divided into the categories of unsupervised and supervised. Many prior works with Machine Learning have used supervised learning method to analyze and classify traffic flows based on prior knowledge and statistical information extracted from the raw data. As a representative example, A. W. Moore et al. performed a series of studies using the dataset collected from their own monitor system [10]. A classification method using full packet payload is developed in [11]. Each network flow has 248 feature parameters in [12]. A naïve Bayesian decision method yield an accuracy of 65.26% in [13]. Using kernel density estimation to correct the erroneous Gaussian assumption, the accuracy can be improved to 93.50%. When using fast correlation based filter (FCBF) to select optimal discriminator for each training set, the identification accuracy improved further to 94.26%. However, when the accuracy is accounted using the total number of bytes identified, the accuracy dropped to 84.06%, as some large flows were misclassified. Using SVM methods and optimal discriminator selection, an accuracy of 96.9% is obtained in [14].

From unsupervised learning perspective, Hernandez-Campos et al. applied a hierarchical clustering method to analyze the traffic pattern from a campus network in [15]. Network flows are separated into clusters using the Euclidean distance between their feature vectors. The

results obtained 8 main clusters. The 1st and 2nd clusters are mostly consisted by various P2P applications; applications of http and https are in the 5th cluster; and the 6th cluster includes e-mails applications such as POP3, SMTP. Only the 3rd and 4th clusters have multiple applications. No numerical accuracy number was given in the paper. Expectation-Maximization algorithm was used to cluster the multiple Internet traffic traces in [16] and Kiviat graph was used to visualize the clusters. S. Zander et al. used an unsupervised Bayesian classification method (Auto-class) to classify multiple traffic traces and obtained an average accuracy of 86.5% in [17, 18]. Information entropy along the dimensions of 4 network flow parameters (srcIP, dstIP, srcPort, dstPort) are used to separate the Internet traffic into different clusters. Xu et al. [19] obtained 27 source IP behavior classes from backbone traffic traces. They also showed that the behavior of each class is fairly stable over time and often associate with well-known applications. A. Lakhina et al.found that the information entropy technique is effective in identifying network traffic anomalies in [20].

III. PSO CLUSTERING ALGORITHM

Our P2P traffic classification method consists of two algorithms: Particle Swarm Optimization (PSO) clustering algorithm was employed to partition a training dataset that consists of scarce labeled samples combined with abundant unlabeled samples. Mapping Algorithm used the few available labeled samples obtain a mapping from the clusters to the different known classes.

A. K-means Algorithm

The K-means algorithm is simple, straightforward and is based on the firm foundation of analysis of variances. It clusters a group of data vectors into a predefined number of clusters. It starts with randomly initial cluster centroids and keeps reassigning the data objects in the dataset to cluster centroids based on the similarity between the data object and the cluster centroid. The reassignment procedure will not stop until a convergence criterion is met (e.g., the fixed iteration number, or the cluster result does not change after a certain number of iterations). The K-means algorithm can be summarized as:

(1) Randomly select cluster centroid vectors to set an initial dataset partition.

(2) Assign each document vector to the closest cluster centroids.

(3) Recalculate the cluster centroid vector c_j using the following equation.

$$c_j = \frac{1}{n_j} \sum_{\forall d_j \in S_j} d_j$$

where d_j denotes the document vectors that belong to cluster S_j; c_j stands for the centroid vector; n_j is the number of document vectors that belong to cluster S_j.

(4) Repeate step 2 and 3 until the convergence is achieved.

.

B. PSO Algorithm

PSO was originally developed by Eberhart and Kennedy in 1995, and was inspired by the social behavior of a flock of birds in [22]. In the PSO algorithm, the birds in a flock are symbolically represented as particles. These particles can be considered as simple agents flying through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility. The velocity and direction of each particle moving along each dimension of the problem space will be altered with each generation of movement. The particle's personal experience, P_{id} and its neighbors'experience, P_{gd} influence the movement of each particle through a problem space. The random values rand1 and rand2 are used for the sake of completeness, that is, to make sure that particles explore a wide search space before converging around the optimal solution. The values of c_1 and c_2 control the weight balance of P_{id} and P_{gd} in deciding the particle's next movement velocity. At every generation, the particle's new location is computed by adding the particle's current velocity, $v_{id}(t)$,to its location, $x_{id}(t)$. Given a multi-dimensional problem space, the ith particle changes its velocity and location according to the following equations.

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 rand1(p_{id} - x_{id}(t)) + c_2 rand2(p_{gd} - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

where ω denotes the inertia weight factor; P_{id} is the location of the particle that experiences the best fitness value; P_{gd} is the location of the particles that experience a global best fitness value; c_1 and c_2 are constants and are known as acceleration coefficients; d denotes the dimension of the problem space; rand1, rand2 are random values in the range of (0, 1).

C. PSO CLustering Algorithm

The main drawback of the K-means algorithm is that the result is sensitive to the selection of the initial cluster centroid and may converge to the local optima. Therefore, the initial selection of the cluster centroid affects the main processing of the K-means and the partition result of the dataset as well. We can consider the clustering problem as an optimization problem that locates the optimal centroid of the clusters rather than to find an optimal partition. This view offers us a chance to apply PSO optimal algorithm on the clustering solution.

The objective of the PSO clustering algorithm is to discover the proper centroid of clusters for minimizing the intra-cluster distance as well as maximizing the distance between clusters. The PSO algorithm performs a

globalized searching for solutions whereas the traditional K-means clustering procedures perform a localized searching.

In the PSO clustering algorithm, the multi-dimensional samples vector space is modeled as a problem space. Each feature in the sample dataset represents one dimension of the problem space. Each sample vector can be represented as a dot in the problem space. The whole sample dataset can be represented as a multiple dimension space with a large number of dots in the space. One particle in the swarm represents one possible solution for clustering the sample collection. Therefore, a swarm represents a number of candidate clustering solutions for the sample collection.

Each particle maintains a matrix $C = (X_1, X_2, ...X_M)$, where X_j represents the i[th] cluster centroid vector and M is the number of clusters. According to its own experience and those of its neighbors, the particle adjusts the centroid vector's position in the vector space at each generation. The fitness value is defined as (3) to evaluate the solution represented by each particle.

$$fitness = k / \sum_{j=1}^{M} \sum_{X_i \in \omega_j} d(X_i, X_j) \quad (3)$$

where $d()$ denotes the distance between centroid X_j and sample X_i in cluster ω_j .M stands for the number of cluster. k is a constant.

The PSO clustering algorithm can be summarized:

1) At the initial stage, each particle randomly chooses M different sample vectors from the sample collection as the initial cluster centroid vectors.

2) For each particle:

(a) Assign each sample vector in the dataset to the closest centroid vector.

(b) Calculate the fitness value according to (3).

(c) Using the velocity and particle position to update (1) and (2) and to generate the next solutions.

3) Repeat step 2) until the maximum number of iterations is exceeded.

IV. MAPPING ALGORITHM

The output of the PSO clustering algorithm is a set of clusters, represented by their centroid. Knowing a sample vector most likely belongs to which cluster does not provide the actual classification to one of the application types. Therefore, we need a mechanism to map the clusters found by the PSO clustering algorithm to the different application types.

A probabilistic assignment $P(Y = y_j \mid C_k)$, where j = 1, ..., q (q being number of application types) and k = 1, ...,M (M being the number of clusters), is used to find the mapping from clusters to labels.

To estimate the probabilities $P(Y = y_j \mid C_k)$, the labeled samples in training data that is used.

$$P(Y = y_i \mid C_k) = \frac{n_{jk}}{n_k}$$

where n_{jk} is the number of samples that were assigned to cluster k with label j, and n_k is the total number of labeled samples that were assigned to cluster k.

To complete the mapping, clusters that do not have any labeled samples assigned to them are defined as unknown application types, the representation of previously unidentified application types.

The decision function for classifying a sample feature vector $x \in lable_i$ is the maximum a posterior decision function:

$$label_i = \arg \max_{y_1 \cdots y_q} P(y_i \mid C_k)$$

where C_k is the nearest cluster to x.

V. P2P TRAFFIC METRICS

From remote address distribution, failed connections, the ratio of incoming and outgoing connections, we show inherent P2P behaviors and map such inherent behaviors into metrics: IP Address Discreteness, Success Rate of Connections and Bidirectional Connections. Then, feature vector composed of such metrics would be used by PSO clustering.

A IP Address Discreteness

Figure 1 shows the connections of a host using BT application. Figure 2 shows the connections of a host t using WWW application. It is observed that the number of concurrent connections for BT traffic of a single host fluctuates with the amount and states of peers. There were 23 BT application connections which were distributed in 23 stub network uniformly. On the other hand, the number of concurrent flows for WWW traffic of a single host were focused on few stub networks. There were 210 WWW connections which were distributed in 22 stub networks.

To describe this behavior, we proposed a P2P traffic metric named IP Address Discreteness. With regard to concurrent flows of a single host, the more proportion of flows of which hosts belong to the same stub network, the less discreteness of these flows' hosts. Referring to the entropy principle in information theory, we first define the entropy of IP address $H(p_1, p_2, ...p_k)$ as follows:

$$H(p_1, p_2, \cdots p_k) = \sum_{k=1}^{K} P_k \log \frac{1}{P_k}$$

$$P_K = \frac{m}{n}$$

Where n denotes the total number of concurrent flows at time t, m denotes the number of IP address in the stub network K.The network prefix length of stub networks has be set as 24. K is the number of stub.

Then, IP Address Discreteness E is defined as follows:

$$E = \frac{H(p_1, p_2, \cdots p_k)}{H_{\max}(p_1, p_2, \cdots p_k)} = \frac{H(p_1, p_2, \cdots p_k)}{\log n}$$

IP Address Discreteness E value is in the range of (0, 1).

Figure 3 showed the IP Address Discreteness of two P2P application :ppstream and pplive every 10s.

(001) 218.75.221.*
(001) 62.99.119.*
(001) 210.192.245.*
(001) 218.175.190.*
(001) 64.109.56.*
(001) 68.61.196.*
(001) 90.207.216.*
(001) 124.227.203.*
(001) 69.1.112.*
(001) 84.245.211.*
(001) 61.152.98.*
(001) 85.226.166.*
(001) 84.109.124.*
(001) 84.52.140.*
(001) 123.65.41.*
(001) 87.242.60.*
(001) 121.19.49.*
(001) 221.200.150.*
(001) 125.123.239.*
(001) 89.25.81.*
(001) 82.5.181.*
(001) 218.83.103.*
(001) 219.68.33.*

Figure 1. Connections of a host using BT application.

(080) 121.0.25.*
(071) 60.12.195.*
(059) 218.83.153.*
(056) 60.28.252.*
(026) 121.0.23.*
(021) 203.209.244.*
(019) 202.165.103.*
(019) 202.165.100.*
(014) 218.108.237.*
(012) 211.152.50.*
(009) 124.94.143.*
(003) 202.112.20.*
(003) 202.114.0.*
(003) 61.152.238.*
(002) 59.36.96.*
(002) 59.77.31.*
(002) 221.130.185.*
(002) 203.110.169.*
(002) 202.165.105.*
(002) 222.202.96.*
(002) 220.164.140.*
(002) 124.237.121.*

Figure 2. Connections of a host using WWW application.

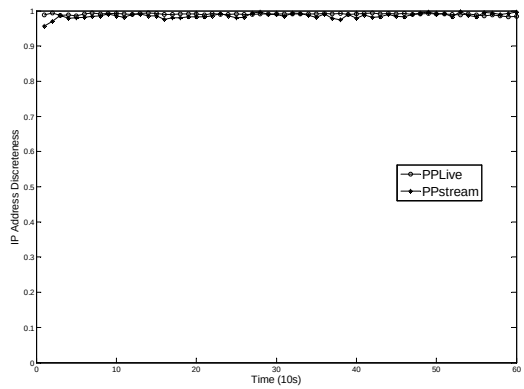

Figure 3. IP Address Discreteness of Pplive and Ppstream.

B Success Rate of Connections

The nature of the P2P traffic lied in the fact every P2P host services as both server and client, conducing to the highly decentralized, self-organizing systems, and the large number of hosts involved and the transient peer membership. These hosts may differ in many aspects, especially the rate of random departure decisions of end-users. Any changes of these aspects can make the connectivity of P2P networks very different at any moment. To keep it's download speed not to decrease, a P2P host can continually initiate TCP connections with others which could be online in a very low probability due to the dynamics of P2P systems.P2P hosts should have a small success rate of initiating connections, and a common (not P2P) host would connect with all kinds of servers in the Internet at a much higher success rate. It is because all the servers must keep their service always acquirable in any client/server systems while peers in P2P system are personal computers: and not always operational and stable. We think the low success rate of TCP connections of peers is the common character of all P2P systems. Based on it, we propose a P2P traffic metric named Success Rate of Connections to describe this behavior. The Success Rate of Connections K is defined as below:

$$M = \frac{successful_{out}}{failed_{out} + successful_{out}}$$

where $failed_{out}$ is the total number of outgoing connections that fail of a host and $successful_{out}$ is the total number of outgoing connections that were successfully established of a host.

Success Rate of Connections value is in the range of (0, 1).

Figure 4 show the Success Rate of Connections of two P2P applications: Ppstream and Pplive every 10s

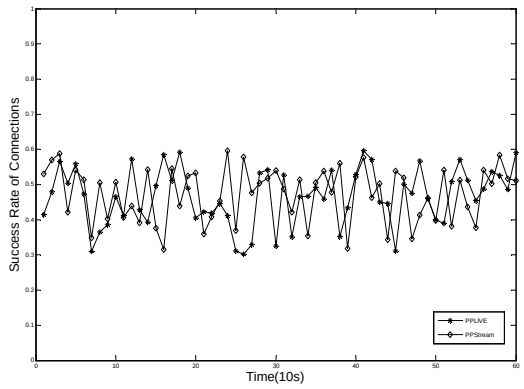

Figure 4. Success Rate of Connections of Ppstream and Pplive.

C Bidirectional Connections Rate

P2P applications not only start connections with peers, but each peer attempts to maintain this network independently. Since peers are equivalent, this means each initiates and receives new connections. Client/server hosts instead primarily either initiate connections (clients)

or receive them (servers). Thus, unlike client-server applications, an inherent behavior of many hosts in a P2P application is a balance of both incoming and outgoing connections.

P2P clients both initiate and receive new connections. We to capture this behavior we use the following ratio of bidirectional connections,

$$P = \frac{successful_{out}}{failed_{out} + successful_{out}}$$

where $failed_{out}$ and $successful_{out}$ is the total number of new, successfully established, incoming and outgoing connections. The metric P will be close to 1 for servers, and close to 0 for clients, and we consider values between 0.2 and 0.9 indicative of P2P hosts. Figure 5 show the Bidirectional Connections Rate of BT and emule application every 10s.

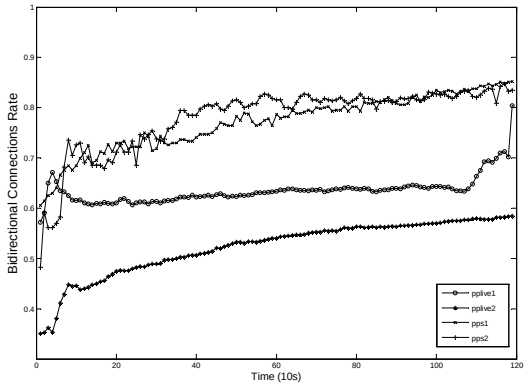

Figure 5. Bidirectional Connections Rate of BT and Emule.

VI. EXPERIMENT

A. Data for experiment

The base truth datasets were established manually. It was obtained from several host running P2P applications including: PpLive, PPStream, Qqlive, Bittorrent, Qq, Msn, Skype, Emule, Xunlei and several host running non-P2P applications including: Dns, ftp, Email, Ssh, telnet, WWW server, FTP server in the campus network of our university. Packets of these host running different applications were firstly captured on a Gbps Ethernet link. Then these packets were hashed into connections according to the five tuples (srcIP, desIP, Prot, srcPort, desPort).Then, three P2P traffic metrics: IP Address Discreteness, Success Rate of Connections and Bidirectional Connections rate for a single host were computed based on the connections every 10s. A three dimensional vector composed of three P2P traffic metric would be used for classification.

We classified captured traffic into seven classes. Table I shows the composition of our dataset.

B. Compared Kmeasns with PSO cluserting

The average distance between samples and the cluster centroid to which they belong was taken as metric. The smaller the average distance value, the more compact the clustering solution is. Table II demonstrated the experimental results by using the K-means, PSO respectively. Seven dataset exacted randomly from Table I were performed for each algorithm. The average distance values are recorded in Table II. As shown in Table II, the PSO approach generates the clustering result with the lowest average distance value for all seven datasets using the Euclidian similarity metric.

TABLE I.
DATASET FOR EXPERIMENT

ID	class	Application	Number of samples
1	P2P file download	Bittorrent,emule,xunlei	4811
2	P2P streaming	Ppstream,Pplive,Qqlive	4320
3	P2P IM	QQ,MSn,Skype	3735
4	Server	ftp server,www server	159
5	client	http,ftp,,https	2320
5	Service	DNS,Email	1364
64	Interactivele	SSH,telnet	598

TABLE II.
AVERAGE DISTANCE OF KMEANS AND PSO CLUSTERING

Dataset	Number of dataset	Average distance of Kmeans	Average distance of PSO clustering
1	2180	9.27	9.1
2	3100	9.45	9.23
3	4200	10.32	8.98
4	5300	11.3	9.76
5	6300	12.32	9.78
6	7350	8.97	8.2
7	8563	10.46	9.34

C. Precision of the classifier

To test the method's effectiveness, precision is calculated as below:

$$precision = \frac{the\ \ number\ \ of\ \ correctly\ \ labeled\ \ samples}{total\ \ number\ \ of\ \ labeled\ \ samples}$$

Those labeled unknown excluded from the precision. TableIII reports the precision of the method.

It is found that increasing the number of unlabelled samples can increase precision with a fixed number of labeled samples. Unlabelled samples are relatively inexpensive to obtain and on the other hand the penalty for incorrect labeling of a sample might be high. Thus, by simply using a large number of unlabelled samples, the precision rate can be substantially increased. This experiment also demonstrates that we can start with a few labeled samples, and over time incrementally label more training samples so as to improve the classification performance.

TABLE III.
CLASSIFICATION PRECISION WITH THE DIFFERENT NUMBER OF UNLABELLED SAMPLES

The number of unlabelled samples	Classification precision using 100 label samples	Classification precision using 1000 label samples	Classification precision using 5000 label samples
100	0.85	0.91	0.92
500	0.87	0.93	0.95
1000	0.9	0.93	0.95
2000	0.93	0.95	0.96
3000	0.93	0.96	0.97
5000	0.94	0.96	0.98

C Choice of number of clusters

The number of clusters impacts the quality of clustering, the time complexity, and the runtime performance of the classifier. To determine a suitable number of clusters, we varied the number of clusters and the number of labeled samples. We changed the number of clusters from 50 to 800, and varied the number of labeled samples in the training data sets from 100 to 5000. Figure 2 shows the results of sample accuracy.

First, sample accuracies in excess of 85% are achieved when using training dataset with 100 or more labeled samples. Second, as the number of clusters increases, the host accuracy also increases. For example, for training data sets with 5000 or more samples, a large number cluster can facilitate samples accuracies around 95%. However, having such large cluster is not practical while this increases the time complexity.

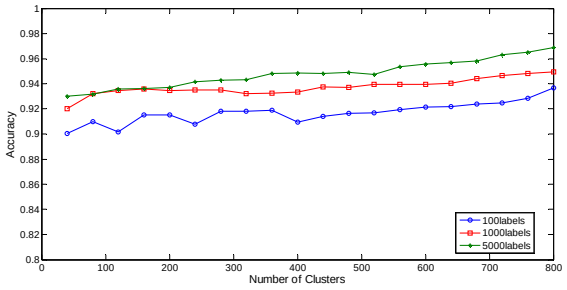

Figure 6. Sample Accuracy.

VI. CONCLUSIONS

This paper proposed and evaluated a semi-supervised clustering for classifying P2P traffic using three P2P traffic metrisc. Comparing with supervised machine learning methods, the proposed semi-supervised approach has two main advantages: First, high precision can be obtained by training with a small number of labeled samples mixed with a large number of unlabelled samples. Adding unlabeled samples can enhance the classifier's performance. Second, this method can handle both seen and unseen applications. Using P2P traffic metrics, this approach identified P2P traffic at host level now. With more P2P metrics studied, it would be developed to classify P2P traffic at connection level in the future work.

REFERENCES

[1] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," IEEE/ACM Transactions on Networking (TON), vol. 12, issue 2,pp. 219-232, April 2004.
[2] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy,An analysis of Internet content delivery systems," in Proceedings of the 5th symposium on Operating systems design and implementation, 2002,pp. 315-327.
[3] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in Proceedings of the 13th international conference on World Wide Web, New York, USA, 2004, pp. 512-521.
[4] T. Karagiannis, A. Broido, N. Brownlee, kc claffy, and M. Faloutsos, " Is P2P dying or just hiding?" in IEEE Globecom 2004 - Global Internet and Next Generation Networks, Dallas, TX, USA, 2004
[5] H. Bleul, E. P. Rathgeb, and S. Zilling, "Evaluation of an efficient measurement concept for P2P multiprotocol traffic analysis," in Proceed-ings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, 2006, pp. 414-423.
[6] A. Spognardi, A. Lucarelli, and R. D. Pietro, "A methodology for P2P file-sharing traffic detection," in Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems - Volume 00 HOT-P2P' 05, 2005, pp. 52-61.
[7] T. Karagiannis, A. Broido, M. Faloutsos, and kc claffy, "Transport layer identification of P2P traffic," in Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, Taormina, Sicily, Italy, 2004, pp.121-134
[8] F. Constantinou and P. Mavrommatis, "Identifying known and unknown peer-to-peer traffic," in Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications, 2006, pp. 93-102.
[9] Thomas Karagiannis, Konstantina Papagiannaki, Michalis Faloutsos, BLINC: Multilevel Traffic Classification in the Dark, ACM SIGCOMM. 35 (4), pp. 229-240, 2005
[10] A. W. Moore J.Hall, C.Kreibich, E. Harris, and I. Pratt. 'Architecture of a Network Monitor.' In Passive & Active Measurement Workshop 2003 (PAM2003), La Jolla, CA, April 2003.
[11] A. W. Moore and D. Papagiannaki. "Toward the Accurate Identification of Network Applications." In Proceedings of the Sixth Passive and Active Measurement Workshop (PAM 2005), March 2005.
[12] A. W. Moore and D. Zuev. "Internet Traffic Classification Using Bayesian Analysis Techniques." SIGMETRICS Perform. Eval. Rev., vol. 33, pp. 50-60, 2005.
[13] A. W. Moore and D. Zuev. "Discriminators for use in flow-based classification". Technical report, Intel Research, Cambridge, 2005
[14] L. Zhu, R. Yuan, and X. Guan, "Accurate Classification of the Internet Traffic Based on the SVM Method." ICC 2007, June 24-28, Glasgow, 2007.
[15] F Hernandez, A B Nobel, F D Smith, and K Jeffay. "Statistical Clustering of Internet Communication Patterns." In Proceedings of Symposium on the Interface of Computing Science and Statistics, 2003.

[16] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. “Flow Clustering Using Machine Learning Techniques.” In PAM, 2004.
[17] S. Zander, T. Nguyen, and G. Armitage. “Automated traffic classification and application identification using machine learning.” In Passive & Active Measurement Workshop, 2005.
[18] S. Zander, T. Nguyen, and G. Armitage. “Self-learning IP traffic classification based on statistical flow characteristics.” In Passive & Active Measurement Workshop, Boston, 2004.
[19] K. Xu, Z. L. Zhang, and S. Bhattacharyya. “Profiling internet backbone traffic: behavior models and applications.” ACM SIGCOMM Computer Communication Review, 2005, 35: 4-16.
[20] A. Lakhina, M. Crovella, and C. Diot. “Mining anomalies using traffic feature distributions.” In SIGCOMM '05: Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2005, 5: 217-228
[21] J. Erman, A. Mahanti, C. Williamson, M. Arlitt, and I. Cohen. A Semi-Supervised Approach to Network Traffic Classification (Extended Abstract). In SIGMETRICS ’07, San Diego, USA, June 2007
[22] J Kennedy, and RC Eberhart, "Particle Swarm Optimization," in Proc.the IEEE International Joint Conference on Neural Networks, Vol. 4, pp 1942–1948, 1995.

Liu Bin was born in Wuhan, China, in 1971. He received his master degree in Pattern Recognition and Intelligent Control from Huazhong university of Science and technology in 1999. He received his Ph.D. degree in Computer Architecture from Huazhong university of Science and technology in 2008. Now, He is lecturer in network centre in Huazhong university of Science and technology. Since 1999 he is also a network administrator at Huazhong university of Science and technology.

From October 2007, he has been involved in several research projects in the field of P2P traffic identification including: unknown P2P streaming identification and signature extract Automatic financed by National High Technology Research and Development Program (grant number 2007AA01Z420), Traffic Classification Method Based on Semi-Supervised Clustering financed by Basic Scientific Research Funds of Central Colleges of China (Grant numbers:c2009m058)

His primary research interests include network measurement, P2P traffic classification and Anomaly detection.