A Novel P2P Traffic Identification Model Based on Ensemble Learning

He XU, Suo-ping WANG, Ru-chuan WANG
1College of Internet of Things, Nanjing University of Posts and Telecommunications
Nanjing 210003, China, E-mail: xuhe2046@126.com
2Center for Control and Intelligence Technology and College of Automation
Nanjing University of Posts and Telecommunications, Nanjing 210003, China,
E-mail: wangsp@njupt.edu.cn
3College of Computer, Nanjing University of Posts and Telecommunications
Nanjing 210003, China, E-mail: wangrc@njupt.edu.cn

Abstract

Peer-to-peer (P2P) traffic has occupied major fraction of all internet traffic. Hence, P2P flow identification becomes an important problem for network management. In our work, we propose an ensemble classification approach for P2P traffic identification, which integrates six DTNB(combination of naive Bayes and decision tables) algorithm and dynamic weighted integration method. The proposed P2P identification scheme can be divided into three stages. In the first stage, we use feature selection algorithm to extract P2P flow characteristics. In the second stage, we use DTNB algorithm to learning the pattern of P2P traffic characteristics. In the third stage, we use dynamic weighted integration method to increase the detection accuracy and reduce false positive in classification. To verify the performance of the proposed P2P identification based on ensemble classification, we collect network traffic traces from NJUPT campus using NETMATE, and run WEKA experiments. The experimental results show that the ensemble classification approach for P2P flow identification can achieve at an average of 97% accuracy rate and 4% false positive rate. Through experiment and giving comparisons of precision, true positive, false positive and ROC curve between the proposed ensemble method and traditional methods such as naive Bayes(NB) , decision trees(DT) and single DTNB algorithm, we find that the proposed method has a better P2P traffic identification accuracy and stability.

Keywords: P2P, Traffic Identification, Ensemble Learning

1. Introduction

Since 2000, peer-to-peer (P2P) networking introduced a major shift in the application and traffic mix of the Internet and established itself as the main driver of increasing traffic volume[1, 2]. P2P applications utilize significant bandwidth and network resources, resulting in network congestion, affecting the availability, reliability and quality of services, and potentially reducing customer satisfaction. While allocating equipment for such significant network usage, telecom carriers and service providers do not gain proportional profits from the services they offer through their infrastructure. As such, telecommunication equipment vendors and Internet Service Providers are interested in efficient solutions to classify and filter P2P traffic for further control and regulation.

Traditionally, network traffic can be easily identified by detecting the port numbers of that traffic, as most of the traffic in the Internet uses standard port numbers. However, with the growth in Internet traffic, in terms of number and type of applications, traditional identification techniques such as port matching, protocol decoding or packet payload analysis are no longer effective. Especially, P2P applications may use randomly selected non-standard ports to communicate which makes it difficult to distinguish them from other types of traffic by inspecting only port numbers[3]. Thus, in recent years, several data mining techniques were proposed to identify the Internet traffic based on the statistical characteristics [4-7].
In this paper we present the results of our study where we captured Internet traffic data at the campus gateway in the Nanjing University of Posts and Telecommunications, performed preprocessing on the data set, and prepared a training data set to which the proposed method can be applied to. The attributes are extracted only from IP layer data streams by NETMATE software. That is to say, our approach relies only on the IP header of the packets, eliminating the privacy concern associated with the techniques that use deep packet inspection. The experiments’ results show that our approach achieved identification accuracy of higher than 97% and can detect P2P flow effectively.

The remainder of this paper is structured as follows. Section II describes related work. Section III presents our identification model and related theories. Section IV describes the data set used in this paper, flow properties, feature selection algorithm and analysis tools. Then experimental results and analysis will be presented in Section V. Finally, concluding remarks and ideas for future work end this paper.

2. Related Work

P2P has gained its popularity in many large scale distributed applications, such as P2P Grid[8] and P2P Live Media Streaming System[9]. And P2P traffic identification has recently gained much attention in both academic and industrial research communities. Various solutions have been developed for P2P traffic classification.

A popular approach is the TCP port based analysis where tools such as Netflow[10] and cflowd[11] are configured to read the service port numbers in the TCP/UDP packet headers, and compare them with the known(default) port numbers of the P2P applications. The packets are then classified as P2P if a match occurs. Although P2P applications have default port numbers, newer versions allow the user to change the port numbers, or choose a random port number within a specified range. Hence, port based analysis becomes inefficient and misleading.

A method using application signatures was developed by S. Sen, O. Spatscheck, and D. Wang in [12], noticing the fact that internet applications have a unique string(signature) located in the data portion of the packet(payload). They used the available information in the proprietary P2P protocol specifications in conjunction with information extracted from packet-level trace analysis to identify the signatures, and classify the packets accordingly. In this case, the traffic that passes through the network is monitored and the data payload of the packets is inspected according to some previously defined application signatures. This approach has been shown to work very well for Internet traffic including P2P applications[13, 14]. However, this technique also has some drawbacks: first, payload analysis poses privacy and security concerns; second, the technique typically requires increased processing and storage capacity; third, it is unable to cope with encrypted transmissions and, finally, this approach only identifies traffic for which signatures are available and are unable to classify previously unknown traffic.

In the past few years, many researchers turn their attentions to machine learning based approaches for P2P traffic classification, in general it can be divided into two categories: Clustering Approaches, in which EM[15], AutoClass[16], K-Means[17, 18] are typical ones; and Supervised Learning, in which naive Bayes[19-22], Bayesian Neural Network[23], Decision Tree[20] and Improved Neural Network[24] are most widely employed. In these researches the statistical characteristics of IP flows are concerned. Flow statistics, such as volume, duration and packet size, are extracted from the network data to establish the feature set. According to Erman[17], there are several reasons why the flow statistics are recommended. First, different applications have different behaviors and thus exhibit different flow statistics. For example, P2P applications would have larger average packet size while IM client would have a smaller one. Second, although obfuscation of flow statistics is possible, it is generally difficult to implement. Third, classification based on flow statistics can benefits from a lot of work on flow sampling/estimation techniques.

However, the currently published machine learning approaches for P2P traffic classification generally employ one classifier only. There are several limitations for single classifier. First, it is difficult to improve the classifier accuracy when it exceeds a certain level. Second, it may
achieve a fairly high classification accuracy rate in one network environment, while it is usually not high in another. Third, a large amount of labeled training data is needed when using supervised learning approaches. It is difficult to obtain a great deal of labeled samples, which is usually hand classified, in the real network environment with high bandwidth and diverse applications. In addition, P2P worms have already emerged recent years in P2P file sharing systems like Gnutella[25]. Through P2P traffic identification, we can detect abnormal worm flow early to help detect the P2P worms. This paper will show how to use ensemble learning method to identification P2P flow.

3. P2P Traffic Identification Model based on Ensemble Learning

Ensemble learning is a learning paradigm that constructs a set of classifiers and then classifies new examples by taking votes[26]. This article is based on a study first reported in the 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD’10). In the conference, we have proposed a P2P traffic identification method based on the combination of naive Bayes and decision tables[27]. Inspired by He et al[28, 29], we present a P2P traffic identification model based on ensemble learning. The ensemble method proposed for P2P traffic identification can be illustrated as follows. First the traffic feature selection algorithm is used to choose best P2P traffic features sets. Then ensemble method for learning P2P traffic features. At the end, Decision function is used in order to produce the final decision. In this paper, a novel P2P traffic identification approach based on ensemble learning using DTNB[30] algorithm, which combines naive Bayes with decision tables algorithms, and decision function using dynamic weighted ensemble method.

3.1. DTNB algorithm[30]

A decision tables (DT) stores the input data in condensed form based on a selected set of attributes and uses it as a lookup table when making predictions. Each entry in the table is associated with class probability estimates based on observed frequencies. The key to learning a DT is to select a subset of highly discriminative attributes. The standard approach is to choose a set by maximizing cross-validated performance. Cross-validation is efficient for DTs as the structure does not change when instances are added or deleted, only the class counts associated with the entries change. Similarly, cross-validation for naive Bayes (NB) is also efficient as frequency counts for discrete attributes can be updated in constant time.

The algorithm for learning the combined model (DTNB) proceeds in much the same way as the one for stand-alone DTs. At each point in the search it evaluates the merit associated with splitting the attributes into two disjoint subsets: one for the DT, the other for NB. It use a forward selection, where,
at each step, selected attributes are modeled by NB and the remainder by the DT, and all attributes are modeled by the DT initially. Leave-one-out cross-validated AUC is used to evaluate the quality of a split based on the probability estimates generated by the combined model. Note that AUC can easily be replaced by other performance measures. We chose AUC to enable a fair comparison to NB. AUC was also used to select attributes for the stand-alone DT.

The class probability estimates of the DT and NB must be combined to generate overall class probability estimates. Assuming \( X^T \) is the set of attributes in the DT and \( X^\perp \) the one in NB, the overall class probability is computed as

\[
Q(y \mid X) = \alpha \times Q_{\text{DT}}(y \mid X^T) \times Q_{\text{NB}}(y \mid X^\perp) / Q(y)
\]

where \( Q_{\text{DT}}(y \mid X^T) \) and \( Q_{\text{NB}}(y \mid X^\perp) \) are the class probability estimates obtained from the DT and NB respectively, \( \alpha \) is a normalization constant, and \( Q(y) \) is the prior probability of the class. All probabilities are estimated using Laplace corrected observed counts. In addition to the method described above, it also consider a variant that includes attribute selection, which can discard attributes entirely from the combined model. To this end, in each step of the forward selection, an attribute can be discarded rather than added to the NB model.

3.2. Dynamic Weighted Ensemble Method

If the output of a neural network \( y = f_i(x) \) can be interpreted as the probability that an instance \( x \) is in a class, then \( y \) as approaches 1, we feel more certain that the instance is in the class. As \( y \) approaches 0, we become more certain that the instance is not in the class. Let us quantify this notion; define the certainty \( c(y) \) of a neural network output:

\[
c(y) = \begin{cases} 
  y & \text{if } y \geq 0.5 \\
  1 - y & \text{otherwise} 
\end{cases}
\]

(2)

The certainty rises for outputs \( y < 0.5 \) as \( y \) falls, and for outputs \( y \geq 0.5 \) as \( y \) rises. We say one network output \( y_1 \) is less certain than another \( y_2 \) if \( c(y_1) < c(y_2) \). Note that the certainty behaves symmetrically with respect to positive and negative decisions; the certainty of an output of 0.1 is the same as that of an output of 0.9, but the decision they are certain about is different.

If instead of choosing static weights derived from \( f_i \)'s performance on a sample of the input space, we allow the weights to adjust to be proportional to the certainties of the respective network outputs, we might achieve better performance. Define the dynamically averaged network (DAN):

\[
f_{\text{DAN}} = \sum_{i=1}^{n} w_i f_i(x)
\]

Where the \( w_i \)'s are:

\[
w_i = \frac{c(f_i(x))}{\sum_{j=1}^{n} c(f_j(x))}
\]

(4)

The \( w_i \)'s sum to 1, so \( f_{\text{DAN}} \) is a weighted average of the network outputs. The difference is that the weight vector is recomputed each time the ensemble output is evaluated, to try to give the best decision for the particular instance under consideration, instead of statically choosing weights that give an optimal decision with respect to a cross validation set. Each network's
contribution to the sum is proportionate to its certainty. A value close to 0.5, for instance, would contribute very little to the sum while a very certain value of 0.99 (or 0.01) among many less certain values would dominate the sum. This method is similar to the idea of using agreement among a set of classifiers to obtain a measure of confidence in the decision[32], but the confidence level (certainty) of each classifier itself is used to obtain the final decision.

4. Experimental Setup

This section describes the data traces, flow features, feature selection algorithm and analysis tools used in the next section evaluation.

4.1. Data traces

This paper uses NETMATE[33] capture network traffic. NETMATE will automatically crawl the network traffic, which include the source address, destination address, source port number, destination port number, protocol type, grouping of packages, and then calculate the average byte of IP packet, the smallest segment size of IP packet and so on. There are 12 traffic characteristics property of IP packet. The network architecture of collecting the network traffic data by NETMATE is shown in Fig.2.

![Figure 2. Network traffic collecting architecture with NETMATE](image)

There are fifty machines in the LAN, including machine which is running a number of popular P2P applications, such as PPLive, Bittorrent, eMule, and Xunlei and so on. In the current network, we set up a flow collector, which uses NETMATE software to capture packets and calculate the network traffic characteristics. In the, According to source address or destination address is 192.168.0.5, P2P or Non-P2P are classified by NETMATE automatically. The results of the network traffic collected was shown in Table I. In our study, we only concerned the TCP data; UDP data are left in future work.

<table>
<thead>
<tr>
<th>Table 1. Collected data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Set</td>
</tr>
<tr>
<td>Day1</td>
</tr>
<tr>
<td>Day2</td>
</tr>
<tr>
<td>Day3</td>
</tr>
</tbody>
</table>
4.2. Flow Features

The best features of data set are 12 attributes in the literature [34]. According to the data features in table II, push_pkts_serv, act_data_pkt_clnt, data_bytes_var_serv, min_seg_size_clnt, and RTT_samples_clnt are selected as the best five feature attributes for P2P identification model by FCBF algorithm.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>push_pkts_serv</td>
<td>Count of all packets with push bit set in TCP header (server to client)</td>
</tr>
<tr>
<td>init_win_bytes_clnt</td>
<td>The total number of bytes sent in initial window.(client→server)</td>
</tr>
<tr>
<td>init_win_bytes_serv</td>
<td>The total number of bytes sent in initial window.(server→client)</td>
</tr>
<tr>
<td>avg_seg_size_serv</td>
<td>Average segment size: data bytes divided by packets. (server→client)</td>
</tr>
<tr>
<td>IP_bytes_med_clnt</td>
<td>Median of total bytes in IP packet (client to server)</td>
</tr>
<tr>
<td>act_data_pkt_clnt</td>
<td>Count of packets with at least 1 byte of TCP data payload (client to server)</td>
</tr>
<tr>
<td>data_bytes_var_serv</td>
<td>Variance of total bytes in packets (server to client)</td>
</tr>
<tr>
<td>min_seg_size_clnt</td>
<td>Minimum segment size observed. (client to server)</td>
</tr>
<tr>
<td>RTT_samples_clnt</td>
<td>Total numbers of RTT samples found (client to server)</td>
</tr>
<tr>
<td>push_pkts_clnt</td>
<td>Count of all packets with push bit set in TCP header (client to server)</td>
</tr>
<tr>
<td>serv_port</td>
<td>Server port.</td>
</tr>
<tr>
<td>clnt_port</td>
<td>Port Number at client</td>
</tr>
</tbody>
</table>

4.3. Feature Selection Algorithm

Feature selection algorithms are widely organized into the filter and wrapper model[35]. And the wrapper method evaluates the effect of different subsets using specific classification algorithms. In this paper, we will be using Fast Correlation-Based Filter (FCBF), described in [36], as well as a variation of a wrapper method in determining the value of the threshold.

As shown in Fig. 3, given a data set with N features and a class C, the algorithm finds a set of predominant features S’best for the class concept. It consists of two major parts. In the first part (line 2-7), it calculates the SU value for each feature, selects relevant features into S’list based on the predefined threshold $\delta$, and orders them in descending order according to their SU values. In the second part (line 8-20), it further processes the ordered list S’list to remove redundant features and only keeps predominant ones among all the selected relevant features.

The first part of the above algorithm has a linear time complexity in terms of the number of features N. As to the second part, in each iteration, using the predominant feature Fp identified in the previous round, FCBF can remove a large number of features that are redundant peers to Fp in the current iteration. The best case could be that all of the remaining features following Fp in the ranked list will be removed; the worst case could be none of them. On average, we can assume that half of the remaining features will be removed in each iteration. Therefore, the time complexity for the second part is $O(N \log N)$ in terms of N. Since the calculation of SU for a pair of features is linear in term of the number of instances M in a data set, the overall complexity of FCBF is $O(MN \log N)$. 

- 114 -
4.4. Analysis Tools

In our work, experiments were conducted using version 3.6.0 of the WEKA (Waikato Environment for Knowledge Analysis) software suite [37]. It is widely used in the Machine Learning community and implemented by Java language.

5. Performance Evaluation and Results Discussion

This section evaluates the effectiveness of the proposed method (6DTNB) using different dataset, and gives a comparison of accuracy and ROC curve with naive Bayes (NB), decision trees (DT) and single DTNB.

5.1. Evaluation Metrics

To measure the performance of the proposed method we use for metrics: precision, true positives, false positives and ROC (receiver operating characteristic) curve [38].

The True Positives, False Positives, and False Negatives defined as follows. The True Positives (TP) measures how many instances of a given class are correctly classified; the False Positives (FP) measures how many instances of other classes are confused with a given class; and the False Negatives (FN) measures the number of misclassified instances of a class. The precision of an algorithm is the ratio of True Positives over the sum of True Positives and False Positives, or the percentage of flows that are properly attributed to a given application by this algorithm.

The ROC curve was first developed by electrical engineers and radar engineers during World War II for detecting enemy objects in battle fields, also known as the signal detection theory, and was soon introduced in psychology to account for perceptual detection of signals. ROC analysis since then has been used in medicine, radiology, and other areas for many decades, and...
it has been introduced relatively recently in other areas like machine learning and data mining. In signal detection theory, a receiver operating characteristic, or simply ROC curve, is a graphical plot of the sensitivity vs. the specificity for a binary classifier system as its discrimination threshold is varied. The ROC can also be represented equivalently by plotting the fraction of true positives (TPR = true positive rate) vs. the fraction of false positives (FPR = false positive rate). Also it is well known as a Relative Operating Characteristic curve, because it is a comparison of two operating characteristics (TPR & FPR) as the criterion changes[39].

5.2. Precision

As shown in Fig. 4, it is the accuracy of the 4 experimental algorithms for P2P identification. D1/D2 represents a subset of the data set Day1 as the training set, a subset of the data set Day2 as test set method. From the graph we can see six DTNB algorithms performs better than other three kinds of algorithm, and reaches at an average of 97% accuracy rate.

5.3. TP & FP

Fig. 5 shows the true positive of P2P identification, and Fig. 6 shows the false positive of P2P identification. Through comparing with other algorithms, we find that the proposed P2P identification ensemble method performs better true positive, especially lower false positive.

5.4. ROC curve

Fig. 7 is the ROC curve of this paper’s experimental methods for P2P flow identification. As be seen, the proposed P2P identification combination algorithm based on naive Bayes and decision tables works in the upper left corner of the ROC curve and gets steep rising curve, which shows the best P2P traffic identification accuracy and stability. At the remaining algorithm, the P2P identification model based on single DTNB is optimal, followed by DT, and based on NB identification model is the worst.
6. Summery and Conclusions

In this paper, we propose a novel method for P2P traffic identification method based on ensemble learning. The proposed P2P identification scheme can be divided into three stages. In the first stage, we use feature selection algorithm to extract P2P flow characteristics. In the second stage, we use DTNB algorithm to learning the pattern of P2P traffic characteristics. In the third stage, we use dynamic weighted integration method to increase the detection accuracy and reduce false positive in classification. Through evaluating the effectiveness of our approach with precision, true positive, false positive and ROC curve, the results show that the DTNB-ensemble P2P traffic identification method produces the best accuracy and the lowest false-positive.

Although this ensemble P2P identification model performs much better than traditional methods, many problems are left for further work, such as 1) in the ensemble learning stage, how to maintain the diversity of P2P classifiers to the great extent, 2) in P2P flow characteristics extract stage, how to utilize the features selection algorithm to extract P2P traffic features more effectively, 3) because our experiment is offline, next step is to test this method online, which needs higher performance, 4) we also need more data to verify whether the five attributes are enough to identify unknown P2P. We plan to evaluate our approach with a larger number of flows and more applications. We believe there are worthy of being studied in the future.

7. Acknowledgments

This work is supported by the National Natural Science Foundation of P.R. China(60973139, 60773041, 61003039, 61003236); Science and Technology Support Program (Industry) Project of Jiangsu Province(BE2010197, BE2010198); Special Fund for Software Technology of Jiangsu Province; The Postdoctoral Foundation of China and Jiangsu Province (0801019C, 20090451240, 20090451241, 20100471353, 20100471355); The Science & Technology Innovation Fund for Higher Education Institutions of Jiangsu Province (CX09B_153Z, CX10B_196Z, CX10B_197Z, CX10B_198Z, CX10B_199Z); The Six Kinds of Top Talent of Jiangsu Province (2008118) and Natural Science Fund for Colleges and Universities in Jiangsu Province(09KJB510020). Special thanks to reviewers for their illuminating comments and editors for their kind efforts in improving our manuscript.

8. References


[33] NETMATE 0.9.4. http://www.ip-measurement.org/tools/netmate/


[37] WEKA. http://www.cs.waikato.ac.nz/ml/weka/
