Analysis of P2P Traffic Identification Methods

Yan Liu, Yulong Yang
1, 2 Southwest University of Science and Technology

ABSTRACT

With the development of the P2P traffic, it brings us various kinds of services, but it also take over a large amount of network resources. Because of this, it affects other services which make it become the hot topic of academic research. As research continues, more and more identification methods are proposed. This paper introduces several popular identification methods from now on. We find P2P traffic shares about 67.79% network traffic by connection mode. Furthermore, we have tested out that some kinds of application software’s P2P traffic take over how much network traffic by the characteristic value of application level through mass data. And 29.19% is accurately found by us which is the software take over.

Keywords: P2P, dpi, characteristic code of application layer, traffic identification

1. INTRODUCTION

In recent years, peer-to-peer technology has become an important part of the Internet. Some people also think P2P will become the direction of development of the Internet in future. Obviously, it has covered most fields only by a few years [1]. Actually, many people have realized the immense business value of P2P. Everything has two sides, P2P does a lot of harm to the Internet at the same time. According to some statistics, P2P traffic takes over about 40%~60% of the net traffic, and 80%~90% is possible [2]. This has affected the other net services such as WEB, EMAIL etc. To solve this problem, it is useless to increase the bandwidth since the net traffic licked up by it. So, both the service provider and the network manager are likely to take control the P2P traffic [3]. It is necessary to supervise business of the P2P streaming media in order to limit or ensure P2P streaming media flow. Therefore, how to supervise business of the P2P streaming media has become everyday favorites in the field of net traffic menstruation.

2. CURRENT STATE

P2P traffic identification methods currently used are listed out as follow:

2.1 Port Identification Method

Port identification method is on the basis of the source port or destination port in the TCP head or the UDP head. Usually, this method only needs to analyze the protocol package of the transport layer. We find the port in the package to compare with the port we have already stored in advance. If the port we analyzed is always used in P2P traffic, the package is belongs to P2P traffic, otherwise it is belong to normal traffic. The advantages of this method are easy, costless, fast and accurate in the identification of the P2P traffic in advance works. And, this method is widely used [5].

2.2 The Recognition of Characteristic Value of Application Layer Method

This method is based on the fact that each kind of P2P packages always has specific message information. For example, the HTTP protocol packet has the specific information as GET, POST. Similarly, there is much similar information in a variety of P2P packages. According to this, people have brought forward the method by detecting the specific information. This is a very accurate way to do the job. Furthermore, it only needs to find the characteristic code in one package to determine the whole flow is P2P traffic. So, it is a good way to be used in the real-time monitoring [6].

2.3 Traffic Pattern Identification Method

P2P protocol, a new application of full use of client resources, is different from the traditional applications such as HTTP, FTP, DNS which are used in the transport layer in many ways. The basic principle of this method is based on detecting the packages of transport layer including TCP packages and UDP packages, and the characteristics of the P2P net to identify whether a network belongs to P2P traffic [7]. The flow characteristics are: the numbers of packages, the length of the package, the arrival time of the package, the ratio between upstream and downstream., the switching frequency of the packages, the ports used by the packages, time of persistent connection, the band width and so on. This method requires a lot of memory and processing speed.

2.4 Connection Pattern Identification Method

This method is based on recognizing the source of the data and the connection pattern of the destination IP address. Some of patterns are unique to P2P, so we can directly identify the P2P traffic. However, some of patterns belong to both P2P and other applications. We should drastically reduce the chances of false positives or false negatives by the history of the flow of the IP address and other characteristics [8].

3. IDENTIFY THE CHARACTERISTIC CODE OF THE APPLICATION LAYER METHOD

The method of identifying the characteristic code of the application level is the main idea of this paper. We find a fact that many characteristic codes of the application identified before is not able to identify the same P2P applications now by extensive literature review.
and collecting and processing characteristic codes found before. So, for a particular P2P application, characteristic code of the transport layer has changed as its software upgrades and updates. It means that we need to update our feature library or we will not be able to identify the corresponding application.

Since the fact that the characteristic codes we have found in the Internet and references are so lack, we need to find out new characteristic codes in order to identify more P2P applications. How do we do this job? Firstly, we open a kind of P2P applications on a local computer and shut down the applications useless. Secondly, we analyze the packages cached on the local computer. Then, we search the same string through different packages. But the result shows that the same string has not been found in different packages for many P2P applications which make us confused a lot. We realize that there are several different message types which used to transport different kinds of control information and data throughout the whole process of data transmission by finding and reading references. It is easy to figure out that different kinds of message types have different kinds of message formats, so does the characteristic code. Above all, we have changed the method. We sort the packages of the same P2P application based on some rules. Use the transport layer as an example. We classify the packages into TCP packages and UDP packages, and then we just need to analysis the classifications. According to the result, we find that most of TCP packages do not have characteristic codes, but UDP packages do. And the characteristic codes we found are most in the head of the data of the application layer. For example, the UDP packages of the QQlive application are always started with 0xfe.

So how about the PPS application? It is easy for us to figure out that the characteristic codes are different a lot whether we use it to watch TV live or do something other. The third byte and the fourth byte of the data of the application layer are 0x4300 when we use PPS to watch TV live. But the first byte and the second byte are not the same which makes us confused a lot. When we order the packages by their length, we find that the packages whose length is equal have the same data in the first byte and the second byte. Furthermore, the values of the two bytes increase as the length. So we can guess that the values of the two bytes have something to do with the length of the package. After calculation, the values of the two bytes are equal to the byte number of the application layer’s data minus 6. When we use it to watch something other, the first byte of the application layer’s data is not the same, but the value of the second byte is 0x80 or 0x84 mostly.

After a lot of material collection and the analysis of the data, the characteristic codes we have found for several P2P applications are list as fallow:

<table>
<thead>
<tr>
<th>Application</th>
<th>Protocol</th>
<th>feature string</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oicq[9]</td>
<td>UDP</td>
<td>0x02</td>
<td>0x2a</td>
</tr>
<tr>
<td>Thunder[6]</td>
<td>UDP</td>
<td>0x36000000</td>
<td>0x2a-0x2d</td>
</tr>
<tr>
<td>Bit Torrent[6]</td>
<td>TCP</td>
<td>0x36000000</td>
<td>0xb7-0xba</td>
</tr>
<tr>
<td>QQDownload</td>
<td>UDP</td>
<td>0xfe+length(payload-3)</td>
<td>0x2a-0x2c</td>
</tr>
<tr>
<td>QQlive</td>
<td>UDP</td>
<td>0xfe+length(payload-3)</td>
<td>0x2a-0x2c</td>
</tr>
<tr>
<td>PPLive[6]</td>
<td>UDP</td>
<td>0x010000</td>
<td>0x34-0x36</td>
</tr>
<tr>
<td>PPS</td>
<td>UDP</td>
<td>0x80/0x84</td>
<td>0x2b</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>length(payload-3)+0x4300</td>
<td>0x2a-0x2d</td>
</tr>
<tr>
<td>eDonKey</td>
<td>UDP</td>
<td>e3 or c5 or d4</td>
<td>0x2a</td>
</tr>
<tr>
<td>/Emule[10]</td>
<td>TCP</td>
<td>e3 or c5 or d4</td>
<td>0x36</td>
</tr>
</tbody>
</table>

Fallowing, we use the characteristic codes we have found to analyze the ratio of each P2P application. The new question comes without any warnings. Some characteristic codes just appear in the initial phase during the data exchange. For example the characteristic codes of the bit torrent protocol just appear in the information of TCP three-way handshake, and the value is 0x13426974. If we only do it by characteristic codes, the accuracy of recognition will be very low. So we do this job by not just characteristic codes but also the IP address and the port which are not changed. Once we find a connection belong to some P2P application, we note down the IP address and the port the connection used. If the packages coming latter have the same IP address and port, we clump the packages into the same application. Above all, we can get all traffic of the application.

However, another question is how we distinguish QQlive from QQDownload since them have the same format of application layer—0xfe+length(payload-3). After a lot of analysis of packages, we find that the length of the two packages is different. The second and the third byte of the application layer’s data have certain difference. QQlive has the value which is 0x2a00 in the second and the third byte, but QQdownload does not have.

In order to carry out the experiment, we first need to capture a large amount of data on the net. We do port mirroring on the switch Cisco-6509 in the center of the school LAN. We capture 20G data, and the data stored in the file which is used for further analysis. The school network topology is shown in Figure1.
Next is the analysis of the data, and the flow chart of the program used is shown in Figure 2.

```
Start
Open the data file
Read the data file
Analysis of the packet
Match the attribute code
P2P flow statistic
The end of the file?
No
Show the results
Yes
```

**Fig 2**: flow chart of the program

Result of the program is shown in Figure 3.

**Fig 3**: Result of the program

Through the data above shows that 27.19% of the total flow of the program identified is P2P traffic, but according to statistics of P2P traffic on the network accounted for 40%~60% of the total flow[3]. It is obvious that the program we used is not able to recognize the full P2P traffic. There are two main reasons, the program identifies a limited number of P2P applications, and does not recognize the encrypted traffic.

The experimental data shows that the recognition of characteristic code of the application layer method can effectively identify the P2P traffic on the network, but we can’t ignore some problems:

- Checking user’s packets often involves the issue of user privacy.
- Many P2P programs use private protocols. We need to use some manual analysis tools (such as Network packet capture) to get the characteristic code due to the lack of documentation. If the protocol updates, the characteristic will lose efficacy.
- The program only recognizes those pre-known characteristic codes analysis of P2P traffic. If the P2P traffic is a new one, the program can do nothing.
- If the P2P protocol uses encryption means of communication, the method will fail.
- It may affect the actual operating efficiency of the method since the large overhead of this method.

For all this, the method based on the recognition of characteristic codes has high precision ratio and accuracy in the practical application. The result of this method is always used as the reference standard for the evaluation of other P2P traffic identification method.

### 4. CONNECTION PATTERN IDENTIFICATION METHOD

There are two main current methods of the connection pattern identification:

- Identify Source-destination IP address used in TCP and UDP data transmission. According to some analysis, the majority of P2P protocols use both TCP and UDP protocols, and other applications using the protocols at the same time are only NetBIOS, games, videos and a few other applications. These applications can use other methods such as the identification of port to exclude.

- Identify the connection pattern by a pair of {IP, port}. Such as browsing the Web, a host typically uses multiple ports to receive data in parallel. So the number of IP addresses is far less than the number of ports. But the two figures of P2P protocol are more consistent due to the distributed network.

However, this method can't identify any specific kind of P2P applications. It can easily detect the proportion P2P traffic takes over because we do not need to speed a lot of time searching for characteristic codes compared to the characteristic code identification method.
We decide to use the second method to detect the proportion of P2P traffic since the high positive rate of the first method.

The flow chart of the experiment program is shown as Figure 4:

Fig 4: flow chart of the experiment program

Result of the program is shown in Figure 5

Fig 5: Result of the program

As we can see, P2P traffic accounts for 67.79% of the total traffic by {IP, port} analysis method. It is similar to the data with other literatures, and has a certain degree of reliability.

5. CONCLUSION

After the series of research, we can conclude that the characteristic code identification method is the major way to detect the P2P traffic since the high accuracy in recent years. We have also seen some shortcomings. We need to capture large amounts of statistics for getting the characteristic code. If the P2P application updates, we need to update the characteristic codes at the same time which may cost a lot of time and affect the actual performance of this method. The worse situation is that a growing number of P2P applications start using data encryption for the application layer, and the method is unable to identify encrypted traffic.

But some new methods are gradually improved, such as flow pattern recognition method. The behavior of the flow is relatively fixed for a specific P2P application. Through a lot of statistical analysis, we can draw the behavior of the flow, and we can use the flow pattern to identify P2P traffic. A great advantage of the flow pattern recognition method which is expected to become the next major P2P traffic identification method is that it can identify encrypted traffic.

REFERENCES


