Using UDP Packets to Detect P2P File Sharing

Tsang-Long Pao and Jian-Bo Chen

Tatung University, Taipei, Taiwan, R.O.C.

Summary
P2P file sharing is one of the major causes of network congestion. Because most of the P2P file sharing software do not bind to a specific port number, it is difficult to identify the P2P file sharing by using layer 3/4 header information. When we use the layer 7 information to find out P2P file sharing, the most difficult thing is to capture all the packets in the network because of the large traffic volume. In this paper, we focus on the feature of eMule and BitTorrent protocol, and using the layer 3/4 information such as UDP packet count and packet size to find out the suspected file sharing activities. When one IP address is suspected in performing file sharing, we only need to capture and analyze the layer 7 information for that IP address. When the payload is extracted, we can make sure that the IP address is running the P2P file sharing software. We do not need to capture all the packets in the network and can still find out the P2P file sharing efficiently and solve the network overload problem.

Key words:
P2P file sharing, NetFlow, eMule, BitTorrent

1. Introduction

Due to its ease of use and large install base with tremendous amount of shared objects, the Peer to Peer(P2P) file sharing is one of the most popular applications in the internet community. However, it is also the one of the major causes of network congestion problem in recent years [1]. The original ideal for P2P is to solve the bottleneck problem in the client-server architecture. The peer acts as both client and server. By using P2P technology, the communication no longer need a single server, thus, solving the single point of failure problem in the client-server architecture.

There are many examples for P2P applications such as VoIP or Instant Message [2]. When the connection is established, the peers can communicate to each other directly. Another example of P2P application is file sharing. When someone needs to download a large file, it can search and locate sites having the file and partition the file into segments and download the file segment by segment from different sites simultaneously [3].

Because of the convenience of P2P file sharing technology, many users begin to share large amount of files such as video, audio, and images to the internet community. This behavior has two problems. The first problem is that the bandwidth may be exhausted by the file sharing, which results in poor network performance for other network users. The second is the issue of intellectual property rights. Thus, it is necessary to limit the bandwidth consumption or even block the traffic of the P2P file sharing

The network administrator must develop an approach to identify the P2P file sharing and then try to limit their traffic [4]. Recently, most of the P2P file sharing software do not bind to a specific port number, thus makes it difficult to identify the P2P activities by only checking the port number. The packet capture software can help the network administrator to capture the payload of all packets, but the loading is extreme huge for a network like the university campus network. Thus, we proposed to use the of layer 3/4 feature to "guess" which IP address is suspected to run P2P file sharing application, and then use the packet capture software to capture the packets of that IP address. Then we can analyze the payload of that IP address to make sure that it is really sharing files using P2P technology.

Most P2P file sharing applications such as eMule[5] and BitTorrent[6] protocol use TCP to share files while use UDP to find the peer neighbor or shared file on the peer machine. In the practical internet traffic, if we ignore the normal UDP packets generated from applications such as DNS, VoIP, SNMP, NTP, Video conference, etc., we can find that there are only a few UDP packets in the Internet. Based on this feature, we can use the UDP packet count and packet size to guess which IP address is likely to be a file sharing host. If an IP address transmits or receives large amount of abnormal UDP packets, then we can capture its payload to verify that it is really using P2P technology to share file.

This remainder of this paper is organized as follows. In section 2, different types of P2P communications is addressed. In section 3, our system architecture is discussed in detail. The results and analysis are given in section 4. Finally, in section 5, the conclusion is given.

Manuscript received August 5, 2007
Manuscript revised August 20, 2007

2. P2P Communications

In the P2P network architecture, there are two mechanisms that peers can communicate to each other. The first is server-less architecture. In this architecture, peers use UDP packets to identify the files and file owners. If the peer discovers the file and file owners, it begins to download the files by establishing a new TCP connection. This architecture is illustrated in Fig. 1.

The second mechanism is server-based architecture. All the peers must search and identify the file and file owner through a centralized server. The server will response the information of the file owners to the requester. Then the requester must check that the peers are on-line and still hold that file. This may be done by sending out small UDP packets in order to improve the performance of P2P communication. If the peers still have the file, then the requester will establish another TCP connection to download that file. This architecture is illustrated in Fig. 2.

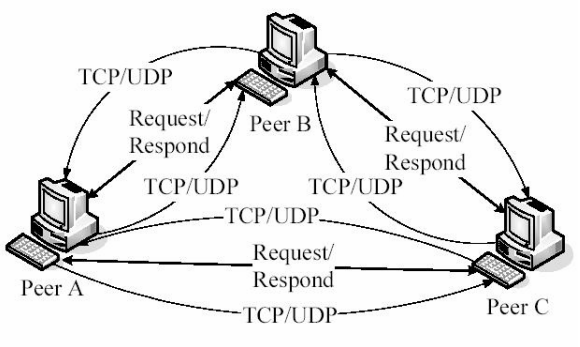

Figure 1. Server-less P2P architecture.

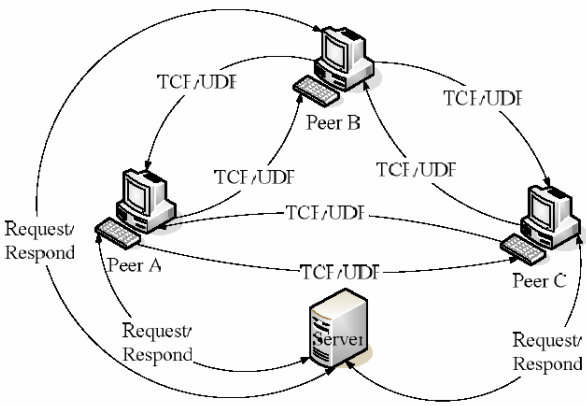

Figure 2. Server-based P2P architecture

3. System Architecture

In our system architecture, we focus on the outbound traffic of the campus network of an university. Generally, the outbound traffic for a university campus network is always large. This, it is hard to capture the entire packet payload by packet capture software and analyze the payload to identify the P2P file sharing. However, without analyzing the layer 7 payload, it is difficult to identify the P2P file sharing activities. In this paper, we propose to collect the outbound traffic in NetFlow format and identify the P2P-like feature. Based on the layer 3/4 information in the NetFlow flow data, we can guess which IP address is running P2P file sharing software. Then we capture the packets to and from that IP and analyze the payload to make sure it is really sharing files using P2P software.

3.1 NetFlow Traffic Collections

In network environments, NetFlow is probably the most useful standard for network traffic accounting. In our implementation, we use both a NetFlow probe (nProbe) and collector to monitor the inbound and outbound flows. The architecture of traffic collection is shown in Fig. 3.

When the nProbe activated, it will collect traffic data and emit flows in NetFlow format towards the specified collector. A set of packets of the same transaction (the same src ip, src port, dst ip, dst port, and protocol) is called a flow. Every flow, even a very long-standing ISO CD image download, has a limited lifetime. This is because the flow collector should periodically receive flow chunks for accounting traffic precisely. The collector is used to receive the NetFlow flow data and store all the information into a database.

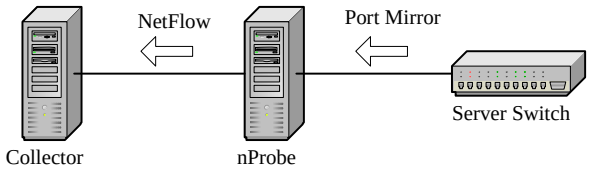

Figure 3. NetFlow traffic collections.

3.2 Packet Payload Capture

Ethereal is one of the most useful tools for network administrator to capture and analyze the packets[7]. It is an open source software that can be used to check the payload of an application. In our environment, the machine that runs the nProbe can also run the Ethereal software to capture the packets for a spechfied IP address.

A sample command for the Ethereal to capture the payload is

/usr/sbin/tethereal -f ip host 192.168.1.100 and udp
-a duration:300
-i eth1
-w cap_file.

In this example, the IP address we are tracking is 192.168.1.100. The duration:300 parameter means that we capture packets for 300 seconds for that IP address. The eth1 is the NIC interface that connects to the mirrored port of the switch. We store the payload in the cap_file for future analysis.

4. Experimental Results

In the experiments, we first observe the difference of UDP packet count for a single IP address that with or without running P2P file sharing application. The second experiment is to collect all the campus network traffic to analyze the TCP and UDP packet count. Then we analyze the size of UDP packets, and the payload information for the specific IP address.

4.1 UDP Packet Count Analysis

In the Internet, most of the network traffics are TCP-based because it is a reliable transmission protocol. But some protocols are UDP-based, such as DNS, SNMP, NTP, VoIP, Video conference, etc. In this experiment, we focus on a specific IP address to observe its UDP packet other than the normal UDP communications. In the first step, we collect all the flow data of a specific IP address for 20 hours and find that there is no any UDP packet other than those packets from normal UDP protocol. The second step, we enable the eMule software but we neither share any file nor download any file. In this case, the machine will transmit and receive UDP packet for finding the peer neighbor. The UDP packet count and distinct IP address count are shown in Fig. 4(a) and Fig. 4(b). Now we start to share file to other peers and download file from peers simultaneously. The UDP packet count and distinct IP address count are increasing as shown in Fig. 5(a) and Fig. 5(b).

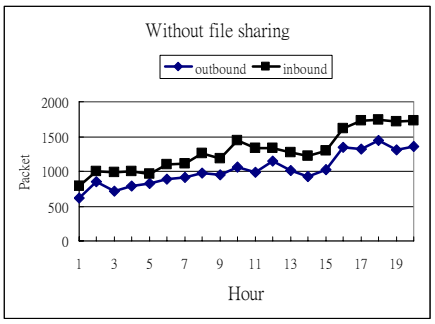

Figure 4(a). Packet count without file sharing

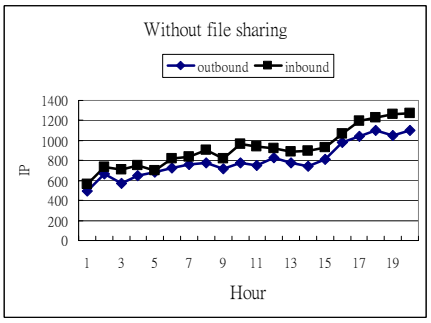

Figure 4(b). Contacting IP address count without file sharing

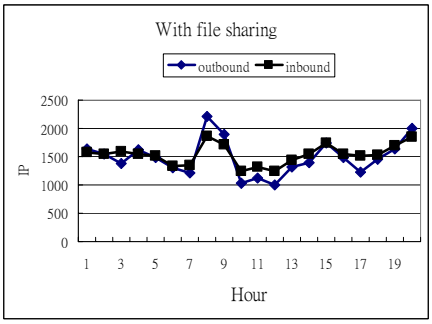

Figure 5(a). Packet count with file sharing

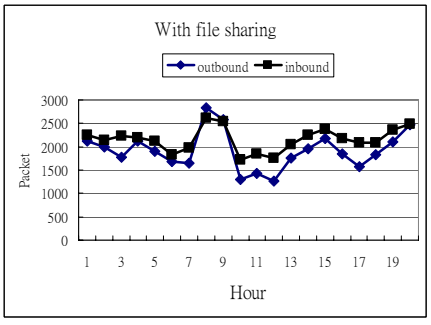

Figure 5(b). Contacting IP address count with file sharing

4.2 Campus Network Traffic Analysis

The collected campus network traffic flow data including the P2P file sharing information. We analyze the TCP and UDP packet count for ten hours. Figure 6 shows that when we do not block any IP addresses which are sharing P2P files, the TCP and UDP packet count are very high and nearly equal in number. In the next experiment, we use the UDP packet size and packet count to determine which IP address is sharing P2P files. We block the P2P file sharing traffic, the TCP and UDP packet count is shown in Fig. 7. In this figure, we can find the decrease of the UDP packet count.

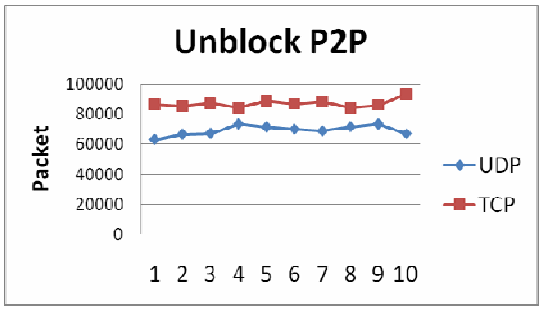

Figure 6. TCP/UDP packet count (P2P unblocked)

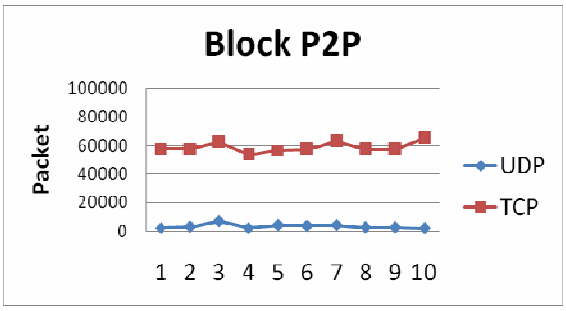

Figure 7. TCP/UDP packet count (P2P blocked)

4.3 UDP Packet Size Analysis

We want to examine the packet size for the UDP packets. Firstly, we exclude the normal UDP packets of protocol such as DNS, SNMP, NTP, etc. We analyze the remaining UPD packets. After analyzing the size of UDP packets of the gathered data, we learned that the P2P file sharing software uses small UDP packet to communicate or to identify the neighbor or to find out the shared files. We define a threshold for the packet count. When the number of UDP packets with nearly the same size is larger than the threshold, the IP address is a suspect of sharing file using P2P technology and probably need to be blocked or to limit its traffic volume. The distribution of UDP packet size is shown in Fig. 8.

4.4 UDP Payload Analysis

When we run the P2P software such as eMule or BitTorrent, we find that these software will continue to send UDP packets in order to verify the server status or to check the shared files of other peers. In the eMule protocol specification, the first byte of payload is always 0xC5, 0xD4, 0xE3, 0xE5 or 0xE4 depend on the versions of eMule software. On the other hand, the payload information of BitTorrent always has the keyword "info-hash" or "d1:ad2:id20". When the ethereal capture the payload and identify those features, it means that this IP address is sharing files using eMule or BitTorrent software. Now we can now apply our policy to that IP address.

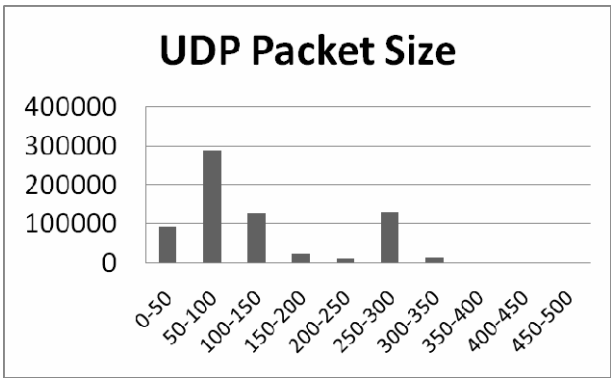

Figure 8. Distribution of UDP packet size

5. Conclusions

Based on the traffic collected by NetFlow, we can analyze the traffic without affecting the normal network traffic. The NetFlow flow information contains only the layer3/4 header information. We use the UDP packet count and packet size to identify which IP address is sharing file using P2P. In order to make sure that the IP address is really sharing files using P2P, we use ethereal to capture the payload of that IP address and analyze its feature. In this architecture, we can identify the P2P file sharing activities more efficiently and do not overwhelm the network loadings.

References

[1] Subhabrata Sen, and Jia Wang , "Analyzing Peer-To-Peer Traffic Across Large Networks," *in IEEE/ACM Transaction on Networking*, 2004

[2] Hamada, T., Chujo, K., Chujo, T., Yang, X., "Peer-to-peer traffic in metro networks: analysis, modeling, and policies,"

in Proceeding of Network Operations and Management Symposium, 2004
[3] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "A measurement study of peer-to-peer file sharing systems," *In Proceedings of Multimedia Computing and Networking*, 2002
[4] Angelo Spognardi, Alessandro Lucarelli, Roberto Di Pietro, "A Methodology for P2P File-Sharing Traffic Detection," *in Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems*, 2005
[5] The eMule protocol, http://www.cs.huji.ac.il/labs/danss/presentations/emule.pdf.
[6] The BitTorrent protocol, http://www.bittorrent.com/
[7] The Ethereal, http://www.ethereal.com/

Tsang-Long Pao received the BS degree in electrical engineering from Tatung Institute of Technology, Taipei, Taiwan, Republic of China, in 1982, and the MS and PhD degree in the School of Electrical and Computer Engineering from the Georgia Institute of Technology in 1990 and 1993, respectively. He is currently an Associate Professor in the department of computer science and engineering at the Tatung University, Taipei, Taiwan, Republic of China. His research interests include emotional speech recognition, ultrasound transducer array, ultrasound signal processing, digital image processing, and computer network management.

Jian-Bo Chen received the BS degree in the department of computer science and engineering in Tatung Institute of Technology, Taipei, Taiwan, Republic of China, in 1993, and the MS degree in the department of electrical engineering in National Taiwan University, Taipei, Taiwan, Republic of China, in 1995. He is currently a lecture in the department of information and telecommunications engineering, Taoyuan, Taiwan, Republic of China. His research interests include load balance and computer network management.