2005－DSM－36（14）
2005／3／18

パッシブ/アクティブ検知を用いたP2Pトラヒック特定法

大坐畠 智　　鈴木 秀章　　萩原 洋一　　寺田 松昭　　川島 幸之助

東京農工大学

抄録　近年 P2P アプリケーションがファイル交換システムとして広く用いられている．このオーバレイネットワークでは，音楽，動画ファイルが主に交換されており，トラヒック量はこれまでのクライアント/サーバ型アプリケーションに比べ，はるかに大きいことが知られている．しかしながら，P2P トラヒックの匿名性が高まり，その実態はよく知られていない．特に近年の P2P アプリケーションの多くはデフォルトのサービスポート番号をもたず，通信路及び，交換するファイルも暗号化されおり，トラヒックの特定は難しい．この問題を解決するため，本論文では日本で最も人気のある P2P ファイル交換アプリケーションである Winny のトラヒックを Winny ピア間通信のサーバ，クライアント関係を用い特定し，方式の評価を行う．

A Traffic Identification Method for a Pure P2P Application with Passive and Active Measurement

Satoshi Ohzahata, Hideaki Suzuki, Yoichi Hagiwara, Matsuaki Terada and Konosuke Kawashima

Tokyo University of Agriculture and Technology

Abstract　Pure P2P applications are widely used nowadays as a file sharing system. In the overlay networks, music and video files are the main items exchanged, and it is known that the traffic volume is much larger than that of classical client/server applications. However, the current status of the P2P application traffic is not well known because of their anonymous communication architectures. In particular, in cases where the application does not use the default service port, and the communication route and the shared file are also encrypted, the identification traffic has not been feasible. To solve this problem, we have developed an identification method for pure Peer-to-Peer communication applications, especially for traffic for Winny, the most popular Peer-to-Peer application in Japan, by using server/client relationships among the peers. We will give some evaluation results for our proposed identification method.

1　Introduction

The Internet applications of end users are changing with the spread of high-performance PCs connected, with broadband links, through the Internet. The traffic volume is also increasing drastically increasing with the change in applications. In particular, the number of users of Peer-to-Peer (P2P) network applications is increasing rapidly since the users are easily able to use network resources over the overlay networks. The characteristic feature of a pure P2P network is that it is a distributed autonomous system which does not rely on a specific server for communications.

Because of this fact, such systems are expected to exhibit scalability in processing power and load balancing at the end computers. However, the traffic volume is becoming much larger than that of the previous Internet applications and the bottlenecks in processing power are shifting from the end computers to the network. In addition, traffic control is very difficult because there is no administrator in the overlay networks and on account of the anonymous nature of the traffic.

Consequently, we need to estimate the effect of P2P traffic to on other forms of traffic in order to construct networks and manage them appropriately. When we start evaluating the P2P traffic, we need first to identify the P2P traffic in the total Internet traffic. Much researches has been done to identify the application traffic and evaluate its characteristics.

The service port number in TCP or UDP is often used as a method of identifying the application traffic, since major Internet applications have use their well known service ports (0–1023) and the server has to use the TCP or UDP port number as the identification number [1]. If the identification number is used correctly by all applications, we can easily identify the application traffic.

Many P2P applications also have their default service port number, Gnutella [2] (6346, 6347), Kazaa

[3] (1214), BitTorrent [4] (6881–6889) and so on. In consequence, many research studies for P2P traffic use the default service port number identification methods in [5], [6] and [7]. However, some recent P2P applications, WinMX [8] and Winny [9], do not use a default service port number, which would allow their services to be identified. For these applications, this identification method does not work well.

Signature matching identification methods [10], [11] are effective when the applications exchange the specific characters in the payload of packets. This traffic identification method is widely applied for Intrusion Detection Systems (IDS) [12], [13] to manage traffic. In this method, every packet needs to be analyzed and it requires huge computation power. In [14], the authors propose a scalable signature matching identification system for P2P traffic, and compare identification methods their application level signature matching method with the default service port number identification method. In these signature matching methods, the application signatures need to be updated with changing the application protocols. There is further difficult problem in the case of Winny, which is one of the most popular pure P2P file sharing application in Japan. The payloads of the packets are encrypted and the protocol details are also not disclosed. These facts make it difficult to identify the Winny traffic since the signature matching is not also useful for the an encrypted payload.

This paper proposes an improved default service port number identification method specifically designed for pure P2P application traffic, to address the above problems. In our method, the service port number may be identified even in cases where the pure P2P applications do not use its their default service port numbers. In the Internet communication, each connection is identified by a tuple of the IP addresses, port numbers and a protocol number (TCP or UDP), and many classical Internet applications play function only as a server or client in the communications. In the classical client/server application, only one connection or relationship is used between the entities involved in the communication. Pure P2P peers, however, play function as both server and client between the peers, and two kinds of connection need to be established. In our method, the pure P2P traffic is identified by the patterns of connection to the server/client ports among the communicating entities. To realize our proposed method, we adopt active measurement and passive measurement for the pure P2P traffic. With the combination of the measurement logs, the service port of a peer is identified through a series of steps. We adopt have apply the proposed method to the Winny network and evaluate our proposed identification method.

The rest of this paper is structured as follows. Section 2 we describes our traffic measurement method. In Section 3, describes our proposed traffic identification method, and section 4 provides conclusions.

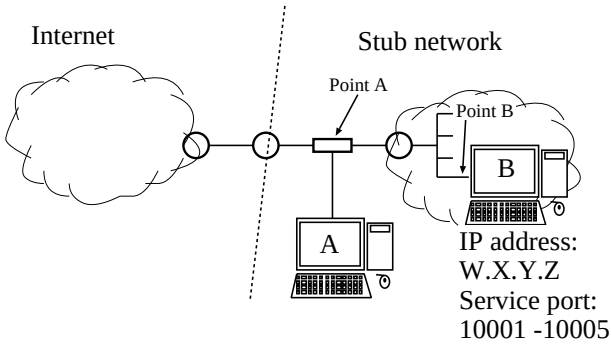

Figure 1: Traffic measurement points.

2 Measurement Methods

We adopt a combination of active and passive measurements to identify Winny traffic, and have two measurement points, as shown in Figure 1. At Point A, the back-bone traffic is measured with by means of a passive measurement, while Point B is placed inside the stub network and measures Winny traffic by acting as a decoy peer with an active measurement.

The Point A (which collects data in log A) is in a switching hub which is placed between an edge router of the Internet and an edge router of the stub network. The link speed is the 100Mbps of full duplex Ethernet but the transfer speed is restricted to 10Mbps at the edge router of the Internet for both directions. We can measure the traffic in the switching hub without affecting the backbone traffic itself by port mirroring. We measured the traffic for 24 hours, from 0:00–24:00 on January 11, 2005 and found 2461 unique IP addresses of the stub network in the traffic log. The combined total traffic volume for both directions was 166.1 GB.

We only logged information of the IP and TCP headers of all these packets. We obtained limited information from the log, but we can reduce the log size and still obtain enough information from it. We define a flow, in the following, as a connection which has the same tuple of IP addresses, port numbers and a protocol number (TCP) between the packet containing the SYN segment flag and the that containing the FIN segment flag. In the measurement, some flows were not evaluated since these flows had no SYN or FYN packet flag of packet in the log. These flows are ignored in the evaluations.

At the point B (log B), the network speed is the 100Mbps of full duplex Ethernet. We measured the traffic log for 13 days, from 0:00 January 5 to 0:00

January 17, 2005. We were able to directly measure the access log from/to the peers in the Winny network at the Point B because the PC B belongs to the Winny network, acting as a decoy peer. By repeatedly changing the point of connection to the Winny network at short intervals, we were able to collect about 40,000 of unique pairs of IP address and service port of the Winny peers per one day by using 5 decoy peers. We used a different measuring period of the log for each analysis.

Both traffic logs are necessary for our traffic identification method, and the log A is used for the backbone traffic evaluations. The detailed specifications of the PCs are below.

[Point A: for the Back-bone traffic]

- The PC is a Dell PRECISION 450 with dual Xeon 3.2Ghz CPUs and the main memory size is 2GB. The OS is FreeBSD.
- The traffic is measured by Snort version 2.0.

[Point B: for the Decoy peer traffic]

- The PC is a Dell PRECISION 450 with dual Xeon 3.2Ghz CPUs and the main memory size is 2GB. The OS is Windows XP professional.
- The version of the Winny is Winny2β6.6.
- We run 5 Winny programs in parallel in the PC in each user session, and the service port numbers is assigned are 10001–10005, respectively.
- All connections to the service port numbers are disconnected by a firewall in the PC B so as not to transfer any files to the Winny network.
- The traffic is measured by Snort version 2.0.

3 Proposed Method

Traditional Internet applications, WWW, FTP, E-mail, etc, are based on the client/server computing model. In thise computing model, each of the communication entitiesy is categorized by only one of the two roles, a server or a client. The server computer only supplies its service and the client computer only receives the service. When the communications start, the client computer accesses the service port of server computer with using its client port, and the server serves provides its service over the connection. Thus, only one identification of the connection between them identification (a tuple of source/destination IP addresses, source/destination port numbers and protocol number) between them is used in the communications. In pure P2P communications, however, one peer plays acts as both a server and a client simultaneously in the communications simultaneously, and there are two kinds of connections between the peers during their communications. Our traffic identification method focuses on the access relations to the ports among the peers.

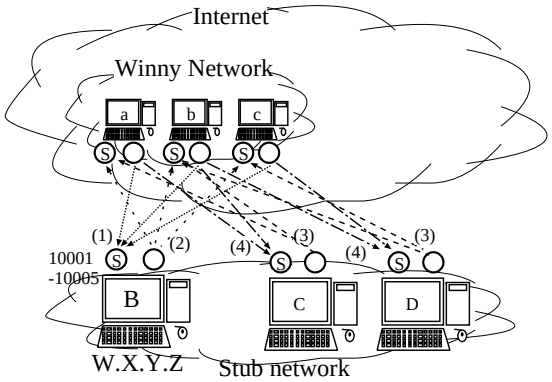

Figure 2: Procedures of our proposed identification method 1.

3.1 Proposed Method 1

The basic idea of our proposed identification method is a decoy peer which collects all pairs of IP addresses and service ports of the Winny peers. However, collecting all of these is difficult because of the restricted search capacity of the decoy peers. Therefore, we need to find the missing Winny peers by using a server/client relationship between the peers.

Figure 2 shows the procedures for our proposed identification method. We place the Peer B as a decoy peer in the stub network whose IP address is W.X.Y.Z and the service port numbers are assigned as 10001–10005 for each decoy peer. The service ports and client ports are depicted by the circles beside the PCs. As soon as the decoy peer joins the Winny network, the peers in the Winny network access the decoy peer and the decoy peer continuously accesses the peers in the Internet to configure the overlay networks. Each arrow corresponds to a connection made by one peer to another peers service port. Thus, these accesses are measured in the PCs A and B, as shown in Figure 1. The procedures are described below.

First we identify the service port number and IP address of the Winny peers connected to the Internet. In the procedures (1) and (2), only log B is used.

(1) When the decoy peer B joins the Winny network, some of the Winny peers in the Internet access the service port of the decoy peer B. The accesses

come from the client port, and we can only identify the IP address of the Winny peers. In this connection, the decoy peer B functions as the server. We add the IP addresses to database α (this applies peer a IP, peer b IP and peer c IP).

(2) Using its client port, the decoy peer B accesses the service ports of the Winny peers in the Internet. If the decoy peer B access the peers in database α, we can identify the service port and IP address of the Winny peers (including peers in the stub network). We add the IP addresses and service ports to database β (this applies peer a IP:service port number, peer b IP:service port number and peer c IP:service port number). In this connection, the peer B functions as a client then the two relations are established between the two peers.

Next, we identify the IP address and the service port number of the Winny users in the stub network, and define "Winny" and "Port 0" peers in the following procedures. Port 0 setting is originally prepared for the peers which are behind the firewall or NAT, but many of the Port 0 users use the setting not to upload any files to the other peers. This is because many shared files in the Winny network are illegal and these files are also automatically shared. In addition, in most cases a file is transferred via a "Winny" peer, and then such "Winny" peers will unintentionally upload and cache these illegal files.

The procedures (3) and (4) use log A and database β.

(3) In the case of a node inside the stub network which accesses a service port of a peer in database β, the node access has the capability of a Winny peer. However, we define a Winny peer in the stub network as a peer which accesses more than two peers in database β, to improve the identification probability. In addition, in this case, the access is initiated by a node inside the stub network, and the accessed port is a service port of the peer. Then, we find that the source IP address node is a Winny peer and add its IP address to database γ (this applies to peer C IP and peer D IP).

(4) The Winny peers in database β access the service ports of peers in database γ. If more than two peers in database β access an identical IP address (database γ) and port number in the stub network of IP and service port number may be identified. We define the peer as "Winny" in the stub network, for the following description, and add the peers to database δ (peer C IP:service port number and peer D IP:service port number). However, some peers in database β do not return to the peers in database γ by using their client ports.

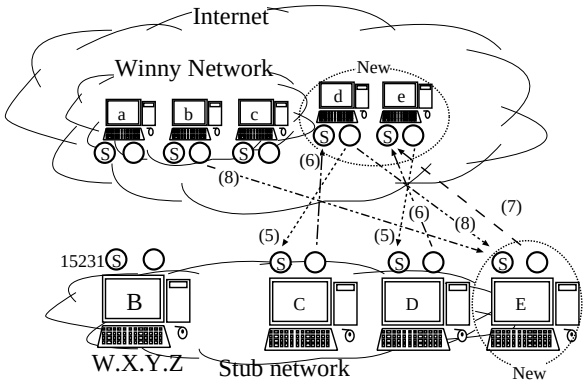

Figure 3: Procedures of our proposed identification method 2.

This is because some Winny peers do not prepare their service port in their setting. We call these peers in the stub network, which do not open their service port to the Winny networks, "Port 0" in the following, and add their IP addresses to the database ϵ.

In the identification procedures (3) and (4), the peers may not be Winny peers, but the probability of this is very low. This is because the value is factorial of the number of port in the TCP or UDP header (less than $1/65536^2$) and our method also considers the port access direction.

From these procedures, (1)–(4), we can find the IP addresses and service ports of Winny peers in the Internet and the stub network. Using databases β, γ and ϵ, we can select the Winny and Port 0 traffic from the log A with this improved port number based application traffic identification method.

3.2 Proposed Method 2

By extending the identification method proposed in the previous subsection, we can find new Winny peers one after another (Figure 3). In the following procedures, the service ports of Winny peers in the stub network play the same role as the decoy peer in the previous subsection. These procedures are described in below.

(5) When the peers d accesses to the service ports of peers C and D, which are not in database α, the peer d becomes newly found Winny peers. We add its IP addresses to database α.

(6) When peers C or D uses their client ports to access peer d, the service port of peer d is identified and the information is added to the database β . (Peer e is also found by the same procedures.)

(7) From inside the stub network, peer E, which is

not in databases δ and ϵ, accesses the newly found peers of these service ports (peers d and e), and so we identify their IP address and add them to database γ.

(8) If more than two peers in database β use their client ports to access peer E, the IP address and service port of the peer in the stub network are identified and the peers added to database δ. The "Port 0" peers are also found in database γ and we add them to database ϵ.

By repeating above procedures, we can eventually find new Winny peers even if these peers are not found in the first few implementations of the procedures.

The next section shows the results of analysis of our proposed methods.

4 Analysis Results

To identify Winny peers in the stub network, we used two traffic logs in section 2. First, we determined the same 24-hour measurement period for the point A and point B. The decoy peer is logged as "Winny" in the log A but we exclude it in the analysis results of this subsection. Some "IP address:service port" combinations in log B have not been identified as Winny peers in log A since a node address and service port number are changing at that time. However, our identification method ensures that the probability of false positive identification is small with these procedures.

The number of peers identified in each step is follows.

(1) The number of unique IP addresses of Winny peers is 67,984 (database α).

(2) The number of unique pair of IP addresses and service port of Winny peers is 45,873 (database β).

(3) The number of unique IP addresses of Winny peers in the stub network is 9 (database γ).

(4) The number of unique IP address and service port of Winny peers in the stub network is 0 (database δ). The number of the Port 0 peers is 9 (database ϵ).

(5) We cannot additionally find additional Winny peers in the stub network since there is no "Winny" peer in the stub network.

From (1) and (2), the service port of the decoy peer is accessed by many Winny peers in the Internet, when the decoy peer joins the Winny network. In the default setting of Winny, each peer has a few active file search connections to the other peers, but each the peer previously searches for further connectable peers to maintain the file search network. With these procedures, several hundred peer search connections are always maintained by keeping, in each Winny peer, the IP:service port number of other Winny peers (the default upper limit is 600.).

The number of IP addresses in (2) is lower than that in (1) since the information of (1) comes from the connection of the service port of the decoy peer but that in (2) is from the connection of the client port of the decoy peer into the result of (1). This fact depends on the search capacity of the decoy peers and the number of Port 0 peers, which do not have their own service port.

In (3), the Winny peers in the stub network (database γ) access 1–3216 peers of the service port in database β in Table 1. In our procedures, peers D and F are not identified as Winny peers since there are only one access to these peers. Since a Winny peer regularly accesses to the service port of the other peers to maintain the peer search connections, this identification method works well. However, we cannot find "Winny" in the definition (4). No connection between the service port of a Winny peer in the stub network and the client port of a Winny peer in the Internet is ever established. This is because that Winny users in the stub network will not upload any file to the other Winny peers.

Next we investigate the effect of the measurement period for log B in Table 2. The number of nodes identified does not vary but the number of identified flows is different. A longer measurement period finds many IP addresses and service ports of Winny peers in the Internet (database β), and many flows are also identified. Comparing (e) with (h), (h) gave better results in spite of the fact that the number of peers in database β is almost the same because earlyer logged peers were not joining the Winny network during the measurement period of the log A. However, the differences in the number of identified flows between (g) and (h) is small. This fact will depend on the connection period of each peer. In (a), (g) and (h), the average flow size becomes small, because the additionally identified flows are used for composing the Adjacent peer check/search network.

5 Conclusion

We have proposed an identification method for pure P2P traffic, Winny, and evaluated its the basic characteristics of it. Using the a decoy node, we identified the IP address and service port of Winny peers and can select the identified IP and service port number in the traffic log of the back-bone. Our iden-

Table 1: Number of accesses to database β peers from the stub network per day.

Suspected peer	A	B	C	D	E	F	G	H	I	J	K
Number of accesses	2446	623	166	1	1626	1	2753	2122	3216	3027	2899

Table 2: Relationship between log period at measurement point B and identified flows.

Measurement period	Databaseβ	Identified peers	Identified flows	Av. flow size
(a) 0:00 Jan. 11 – 0:00 Jan. 12	45873	9	111064	32.4KB
(b) 0:00 Jan. 10 – 0:00 Jan. 11	48434	9	57872	78.1KB
(c) 0:00 Jan. 9 – 0:00 Jan. 11	84129	9	71525	67.5KB
(d) 0:00 Jan. 8 – 0:00 Jan. 11	114715	9	78074	64.1KB
(e) 0:00 Jan. 4 – 0:00 Jan. 11	215557	10	87450	61.5KB
(f) 12:00 Jan. 10 – 12:00 Jan. 12	84129	9	120042	46.7KB
(g) 0:00 Jan. 10 – 0:00 Jan. 12	110097	9	123589	48.2KB
(h) 0:00 Jan. 8 – 0:00 Jan. 14	213968	9	128486	47.7KB

tification method will be effective for pure P2P applications which will appears in the future since our methodits depends on the basic relationships among in client/server computing in the Internet applications.

In the a stub network, the number of Winny users is small. We may not find "Winny" traffic since the Winny users in the stub network are use Port 0. We only a collect traffic log from the other stub networks which haves many Winny users, even if search capacity of the decoy peer is current one, characteristics of the traffic will be much clearly analyzed. The introduced identification method is one of an example, and we should improve the method with by analyzing the access patterns among the peers. Our identification method depends on the access number of accesses of the decoy peers from by peers in the Winny networks and the number of users in the stub network. As a result, some flows may not be identified by our method. If we prepare many decoy peers or there are many users in the stub network, our method improves the identification performance of our method improves.

When we control traffic, we should need know the status and deal them manage it in real time. Our proposed procedure will require this improvement for the usage application.

References

[1] M. St. Johns and G. Huston, "Considerations on the use of a Service Identifier in Packet Headers," RFC 3639, 2003.

[2] Gnutella, "http://www.gnutella.com/"

[3] Kazaa, "http://www.kazaa.com/"

[4] BitTorrent Protocol, "http://bitconjurer.org/BitTorrent"

[5] S. Saroiu, P. Gummadi and S. D. Gribble, "Measurement study of peer-to-peer file sharing systems," *Multimedia Computing and Networking 2002,* 2002.

[6] S. Sen and J. Wang, "Analyzing Peer-To-Peer Traffic Across Large Networks," *IEEE/ACM Trans. on Networking,* Vol. 12, No. 2, pp. 219–232, 2004.

[7] M. Kim, H. Kang and J. W. Hong, "Towards Peer-to-Peer Traffic Analysis Using Flows," *Proc. of 14h IFIP/IEEE Workshop Distributed Systems: Operations and Management,* 2003.

[8] WinMX, "http://www.winmx.com/"

[9] Winny, "http://www.nynode.info/"

[10] C. Dewes, A. Wichmann and A. Feldmann, "An Analysis of Internet Chat Systems," *Proc. of ACM SIGCOMM Internet Measurement Workshop 2003,* pp. 51–64, 2003.

[11] K. P. Gummadi, R. J. Dunn and S. Saroiu, "Measurement, Modeling and Analysis of a Peer-to-Peer File-Sharing Workload," *Proc. of ACM SOSP'03 2003,* pp. 314–329, 2003.

[12] Snort, "http://www.snort.org/"

[13] P. Barford, J. Kline, D. Plonka and A. Ron, "A Signal Analysis of Network Traffic Anomalies," *Proc. of ACM IMW'02,* pp. 71–82, 2002.

[14] S. Sen O. Spatscheck and D. Wang, "Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures," *Proc. of ACM WWW'04,* 2004.